
Amazon Elastic Compute Cloud

User Guide for Windows Instances



Amazon Elastic Compute Cloud: User Guide for Windows Instances

Copyright © 2016 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

What Is Amazon EC2?	1
Features of Amazon EC2	1
How to Get Started with Amazon EC2	2
Related Services	3
Accessing Amazon EC2	3
Pricing for Amazon EC2	4
PCI DSS Compliance	4
Basic Infrastructure	5
Amazon Machine Images and Instances	5
Regions and Availability Zones	6
Storage	6
Root Device Volume	8
Networking and Security	10
AWS Identity and Access Management	10
Differences between Windows Server and an Amazon EC2 Windows Instance	10
Designing Your Applications to Run on Amazon EC2 Windows Instances	12
Setting Up	13
Sign Up for AWS	13
Create an IAM User	13
Create a Key Pair	15
Create a Virtual Private Cloud (VPC)	17
Create a Security Group	17
Getting Started	20
Overview	20
Prerequisites	21
Step 1: Launch an Instance	21
Step 2: Connect to Your Instance	22
Step 3: Clean Up Your Instance	24
Next Steps	24
Best Practices	25
Tutorials	27
Tutorial: Deploy a WordPress Blog	27
Prerequisites	27
Installing the Microsoft Web Platform Installer	28
Installing WordPress	28
Configuring Security Keys	29
Configuring the Site Title and Administrator	30
Making Your WordPress Site Public	31
Next Steps	31
Tutorial: Installing a WAMP Server	32
Tutorial: Installing a WIMP Server	34
Tutorial: Increase the Availability of Your Application	38
Prerequisites	39
Scale and Load Balance Your Application	39
Test Your Load Balancer	41
Tutorial: Remotely Manage Your Instances	41
Launch a New Instance	41
Grant Your User Account Access to SSM	42
Send a Command Using the EC2 Console	42
Send a Command Using AWS Tools for Windows PowerShell	43
Tutorial: Set Up a Windows HPC Cluster	44
Prerequisites	44
Step 1: Set Up Your Active Directory Domain Controller	45
Step 2: Configure Your Head Node	47
Step 3: Set Up the Compute Node	49

Step 4: Scale Your HPC Compute Nodes (Optional)	50
Running the Lizard Performance Measurement Application	51
IP Permissions for the Active Directory Security Groups	51
IP Permissions for HPC Cluster Security Group	55
Amazon Machine Images	61
Using an AMI	61
Creating Your Own AMI	62
Buying, Sharing, and Selling AMIs	62
Deregistering Your AMI	62
AWS Windows AMIs	62
Selecting an Initial Windows AMI	62
Keeping Your AMIs Up-to-Date	63
AMI Types	63
Launch Permissions	64
Storage for the Root Device	64
Finding a Windows AMI	67
Finding a Windows AMI Using the Amazon EC2 Console	67
Finding an AMI Using the AWS CLI	68
Finding an AMI Using the AWS Tools for Windows PowerShell	68
Finding a Windows Server 2003 AMI	68
Shared AMIs	69
Finding Shared AMIs	69
Making an AMI Public	70
Sharing an AMI with Specific AWS Accounts	71
Using Bookmarks	73
Guidelines for Shared Windows AMIs	73
Paid AMIs	74
Selling Your AMI	74
Finding a Paid AMI	74
Purchase a Paid AMI	75
Getting the Product Code for Your Instance	76
Using Paid Support	76
Bills for Paid and Supported AMIs	77
Managing Your AWS Marketplace Subscriptions	77
Creating an Amazon EBS-Backed Windows AMI	77
Overview of Creating Amazon EBS-Backed AMIs	78
Creating a Windows AMI from a Running Instance	78
Creating an Instance Store-Backed Windows AMI	80
Instance Store-Backed Windows AMIs	81
Preparing to Create an Instance Store-Backed Windows AMI	82
Bundling an Instance Store-Backed Windows Instance	82
Registering an Instance Store-Backed Windows AMI	83
AMIs with Encrypted Snapshots	84
AMI Scenarios Involving Encrypted EBS Snapshots	84
Copying an AMI	87
Copying an AMI You Own	87
Copying an AMI Across AWS Accounts	87
Copying an AMI Across Regions	88
Copying to Encrypt	89
AMI Copying Scenarios	89
Copying an AMI Using the Console or Command Line	90
Stopping a Pending AMI Copy Operation	91
Deregistering Your AMI	92
Cleaning Up Your Amazon EBS-Backed AMI	92
Cleaning Up Your Instance Store-Backed AMI	93
Windows AMI Versions	94
Configuration Settings and Drivers	95
Updating Your Windows Instance	95

Upgrading or Migrating a Windows Server Instance	95
Determining Your Instance Version	95
Subscribing to Windows AMI Notifications	96
Image Changes	97
Details About AWS Windows AMI Versions	98
Changes in Windows Server 2016 AMIs	108
Create a Standard Amazon Machine Image Using Sysprep	111
Before You Begin	111
Using Sysprep with the EC2Config Service	112
Run Sysprep with the EC2Config Service	115
Troubleshooting Sysprep with EC2Config	116
Instances	117
Instance Types	117
Available Instance Types	118
Hardware Specifications	119
Networking and Storage Features	119
Instance Limits	121
T2 Instances	121
Memory Optimized Instances	124
Accelerated Computing Instances	127
C4 Instances	129
I2 Instances	131
D2 Instances	133
H1 Instances	134
HS1 Instances	135
T1 Micro Instances	136
Resizing Instances	147
Instance Purchasing Options	150
Determining the Instance Lifecycle	151
Reserved Instances	152
Scheduled Instances	176
Spot Instances	180
Dedicated Hosts	226
Dedicated Instances	236
Instance Lifecycle	241
Instance Launch	241
Instance Stop and Start (Amazon EBS-backed instances only)	241
Instance Reboot	242
Instance Retirement	242
Instance Termination	242
Differences Between Reboot, Stop, and Terminate	243
Launch	244
Connect	254
Stop and Start	259
Reboot	262
Retire	262
Terminate	264
Recover	269
Configure Instances	270
Instance Metadata and User Data	271
Using EC2Config	283
Using EC2Launch	319
Using SSM Config	326
PV Drivers	352
Setting the Password	370
Setting the Time	375
Configuring a Secondary Private IPv4 Address	378
Upgrading Instances	382

Identify EC2 Instances in a Mixed Computing Environment	390
Amazon EC2 Systems Manager	392
Getting Started	393
Prerequisites	394
Installing and Updating SSM Agent	396
Configuring SSM Agent to Use a Proxy	396
Setting Up Systems Manager in Hybrid Environments	397
Create an IAM Service Role	398
Create a Managed-Instance Activation	399
Install the SSM Agent on Servers and VMs in Your Hybrid Environment	400
Configuring Access	400
Use SSM Managed Policies	401
Configure Your Own Roles and Polices	401
Create EC2 Instances that Use the EC2 Instance Role	407
Shared Components	407
Cron Schedules	408
Maintenance Windows	410
Parameter Store	429
Remote Management	437
Components and Concepts	439
Executing Commands	442
Creating SSM Documents	478
Sharing SSM Documents	484
Walkthroughs	489
Command Status and Monitoring	501
Troubleshooting Run Command	513
Inventory Management	515
Getting Started with Inventory	515
Service Limits	515
About Systems Manager Inventory	516
Configuring Inventory Collection	517
Querying Inventory Collection	518
Inventory Manager Walkthrough	518
State Management	522
How It Works	522
Getting Started with State Manager	522
Creating a Document	523
About State Manager Associations	526
State Manager Walkthroughs	527
Maintenance and Deployment Automation	530
Configuring Access	530
Automation Walkthroughs	532
Actions Reference for Automation Documents	540
Automation System Variables	552
Monitoring	563
Automated and Manual Monitoring	565
Automated Monitoring Tools	565
Manual Monitoring Tools	566
Best Practices for Monitoring	566
Monitoring the Status of Your Instances	567
Instance Status Checks	567
Scheduled Events	571
Monitoring Your Instances Using CloudWatch	575
Enable Detailed Monitoring	575
List Available Metrics	577
Get Statistics for Metrics	583
Graph Metrics	590
Create an Alarm	591

Create Alarms That Stop, Terminate, Reboot, or Recover an Instance	592
Network and Security	601
Key Pairs	602
Creating Your Key Pair Using Amazon EC2	602
Importing Your Own Key Pair to Amazon EC2	603
Retrieving the Public Key for Your Key Pair on Windows	604
Verifying Your Key Pair's Fingerprint	605
Deleting Your Key Pair	605
Security Groups	606
Security Groups for EC2-Classic	606
Security Groups for EC2-VPC	606
Security Group Rules	607
Default Security Groups	609
Custom Security Groups	609
Working with Security Groups	609
Security Group Rules Reference	613
Controlling Access	619
Network Access to Your Instance	619
Amazon EC2 Permission Attributes	620
IAM and Amazon EC2	620
IAM Policies	621
IAM Roles	658
Network Access	663
Amazon VPC	665
Benefits of Using a VPC	666
Differences Between EC2-Classic and EC2-VPC	666
Sharing and Accessing Resources Between EC2-Classic and EC2-VPC	669
Instance Types Available Only in a VPC	671
Amazon VPC Documentation	671
Supported Platforms	672
ClassicLink	673
Migrating from EC2-Classic to a VPC	683
Instance IP Addressing	693
Private IPv4 Addresses and Internal DNS Hostnames	694
Public IPv4 Addresses and External DNS Hostnames	695
Elastic IP Addresses (IPv4)	696
Amazon DNS Server	696
IPv6 Addresses	696
IP Address Differences Between EC2-Classic and EC2-VPC	697
Working with IP Addresses for Your Instance	698
Multiple IP Addresses	702
Elastic IP Addresses	709
Elastic IP Address Basics	710
Elastic IP Address Differences for EC2-Classic and EC2-VPC	710
Working with Elastic IP Addresses	712
Using Reverse DNS for Email Applications	716
Elastic IP Address Limit	716
Network Interfaces	716
IP Addresses Per Network Interface Per Instance Type	717
Scenarios for Network Interfaces	720
Best Practices for Configuring Network Interfaces	722
Working with Network Interfaces	722
Placement Groups	731
Placement Group Limitations	731
Launching Instances into a Placement Group	732
Deleting a Placement Group	733
Network MTU	734
Jumbo Frames (9001 MTU)	734

Path MTU Discovery	734
Check the Path MTU Between Two Hosts	735
Check and Set the MTU on your Amazon EC2 Instance	735
Troubleshooting	737
Enhanced Networking	737
Enhanced Networking Types	737
Enabling Enhanced Networking on Your Instance	738
Enabling Enhanced Networking: Intel 82599 VF	738
Enabling Enhanced Networking: ENA	740
Storage	744
Amazon EBS	745
Features of Amazon EBS	746
EBS Volumes	747
EBS Snapshots	788
EBS Optimization	795
EBS Encryption	799
EBS Performance	803
EBS CloudWatch Events	816
Instance Store	822
Instance Store Lifetime	823
Instance Store Volumes	823
Add Instance Store Volumes	826
SSD Instance Store Volumes	828
Amazon EFS	829
Amazon S3	829
Amazon S3 and Amazon EC2	830
Instance Volume Limits	831
Linux-Specific Volume Limits	831
Windows-Specific Volume Limits	831
Bandwidth vs Capacity	832
Device Naming	832
Available Device Names	832
Device Name Considerations	833
Block Device Mapping	833
Block Device Mapping Concepts	834
AMI Block Device Mapping	836
Instance Block Device Mapping	838
Mapping Disks to Volumes	842
Listing the Disks Using Windows Disk Management	843
Listing the Disks Using Windows PowerShell	844
Disk Device to Device Name Mapping	846
Using Public Data Sets	848
Public Data Set Concepts	848
Finding Public Data Sets	848
Creating a Public Data Set Volume from a Snapshot	849
Attaching and Mounting the Public Data Set Volume	850
Resources and Tags	851
Resource Locations	851
Resource IDs	852
Working with Longer IDs	853
Controlling Access to Longer ID Settings	856
Listing and Filtering Your Resources	856
Advanced Search	856
Listing Resources Using the Console	857
Filtering Resources Using the Console	858
Listing and Filtering Using the CLI and API	859
Tagging Your Resources	859
Tag Basics	860

Tag Restrictions	861
Tagging Your Resources for Billing	862
Working with Tags Using the Console	863
Working with Tags Using the CLI or API	868
Service Limits	869
Viewing Your Current Limits	869
Requesting a Limit Increase	870
Usage Reports	870
Available Reports	871
Getting Set Up for Usage Reports	871
Granting IAM Users Access to the Amazon EC2 Usage Reports	872
Instance Usage	873
Reserved Instance Utilization	876
AWS Systems Manager for Microsoft System Center VMM	882
Features	882
Limitations	883
Requirements	883
Getting Started	883
Setting Up	883
Sign Up for AWS	883
Set Up Access for Users	884
Deploy the Add-In	886
Provide Your AWS Credentials	886
Managing EC2 Instances	887
Creating an EC2 Instance	887
Viewing Your Instances	889
Connecting to Your Instance	890
Rebooting Your Instance	890
Stopping Your Instance	891
Starting Your Instance	891
Terminating Your Instance	891
Importing Your VM	891
Prerequisites	892
Importing Your Virtual Machine	892
Checking the Import Task Status	893
Backing Up Your Imported Instance	893
Troubleshooting	894
Error: Add-in cannot be installed	894
Installation Errors	894
Checking the Log File	895
Errors Importing a VM	895
Uninstalling the Add-In	895
AWS Management Pack	896
Overview of AWS Management Pack for System Center 2012	897
Overview of AWS Management Pack for System Center 2007 R2	898
Downloading	899
System Center 2012	899
System Center 2007 R2	900
Deploying	900
Step 1: Installing the AWS Management Pack	901
Step 2: Configuring the Watcher Node	902
Step 3: Create an AWS Run As Account	902
Step 4: Run the Add Monitoring Wizard	905
Step 5: Configure Ports and Endpoints	909
Using	910
Views	910
Discoveries	924
Monitors	925

Rules	926
Events	926
Health Model	927
Customizing the AWS Management Pack	928
Upgrading	929
System Center 2012	929
System Center 2007 R2	929
Uninstalling	930
System Center 2012	930
System Center 2007 R2	930
Troubleshooting	931
Errors 4101 and 4105	931
Error 4513	931
Event 623	931
Events 2023 and 2120	932
Event 6024	932
General Troubleshooting for System Center 2012 — Operations Manager	932
General Troubleshooting for System Center 2007 R2	933
AWS Diagnostics for Windows Server	934
Analysis Rules	934
Analyzing the Current Instance	935
Collecting Data From an Offline Instance	937
Data File Storage	937
Troubleshooting	939
Troubleshoot an Unreachable Instance	939
How to Take a Screenshot of an Unreachable Instance	939
Common Screenshots	940
Common Issues	946
EBS volumes don't initialize on Windows Server 2016 AMIs	947
Boot an EC2 Windows Instance into Directory Services Restore Mode (DSRM)	947
High CPU usage shortly after Windows starts	950
No console output	950
Instance terminates immediately	951
Remote Desktop can't connect to the remote computer	951
RDP displays a black screen instead of the desktop	953
Instance loses network connectivity or scheduled tasks don't run when expected	954
Insufficient Instance Capacity	954
Instance Limit Exceeded	955
Windows Server 2012 R2 not available on the network	955
Common Messages	955
"Password is not available"	955
"Password not available yet"	956
"Cannot retrieve Windows password"	956
"Waiting for the metadata service"	956
"Unable to activate Windows"	959
"Windows is not genuine (0x80070005)"	960
"No Terminal Server License Servers available to provide a license"	960
Document History	962
AWS Glossary	978

What Is Amazon EC2?

Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) cloud. Using Amazon EC2 eliminates your need to invest in hardware up front, so you can develop and deploy applications faster. You can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage. Amazon EC2 enables you to scale up or down to handle changes in requirements or spikes in popularity, reducing your need to forecast traffic.

For more information about cloud computing, see [What is Cloud Computing?](#)

Features of Amazon EC2

Amazon EC2 provides the following features:

- Virtual computing environments, known as *instances*
- Preconfigured templates for your instances, known as *Amazon Machine Images (AMIs)*, that package the bits you need for your server (including the operating system and additional software)
- Various configurations of CPU, memory, storage, and networking capacity for your instances, known as *instance types*
- Secure login information for your instances using *key pairs* (AWS stores the public key, and you store the private key in a secure place)
- Storage volumes for temporary data that's deleted when you stop or terminate your instance, known as *instance store volumes*
- Persistent storage volumes for your data using Amazon Elastic Block Store (Amazon EBS), known as *Amazon EBS volumes*
- Multiple physical locations for your resources, such as instances and Amazon EBS volumes, known as *regions* and *Availability Zones*
- A firewall that enables you to specify the protocols, ports, and source IP ranges that can reach your instances using *security groups*
- Static IPv4 addresses for dynamic cloud computing, known as *Elastic IP addresses*
- Metadata, known as *tags*, that you can create and assign to your Amazon EC2 resources

- Virtual networks you can create that are logically isolated from the rest of the AWS cloud, and that you can optionally connect to your own network, known as *virtual private clouds* (VPCs)

For more information about the features of Amazon EC2, see the [Amazon EC2 product page](#).

Amazon EC2 enables you to run any compatible Windows-based solution on our high-performance, reliable, cost-effective, cloud computing platform. For more information, see [Amazon EC2 Running Windows Server & SQL](#).

For more information about running your website on AWS, see [Websites & Website Hosting](#).

How to Get Started with Amazon EC2

The first thing you need to do is get set up to use Amazon EC2. After you are set up, you are ready to complete the Getting Started tutorial for Amazon EC2. Whenever you need more information about a feature of Amazon EC2, you can read the technical documentation.

Get Up and Running

- [Setting Up with Amazon EC2](#) (p. 13)
- [Getting Started with Amazon EC2 Windows Instances](#) (p. 20)

Basics

- [Amazon EC2 Basic Infrastructure for Windows](#) (p. 5)
- [Instance Types](#) (p. 117)
- [Tags](#) (p. 859)

Networking and Security

- [Amazon EC2 Key Pairs and Windows Instances](#) (p. 602)
- [Security Groups](#) (p. 606)
- [Elastic IP Addresses](#) (p. 709)
- [Amazon EC2 and Amazon VPC](#) (p. 665)

Storage

- [Amazon EBS](#) (p. 745)
- [Instance Store](#) (p. 822)

Working with Windows Instances

- [Remote Management](#) (p. 437)
- [Differences between Windows Server and an Amazon EC2 Windows Instance](#) (p. 10)
- [Designing Your Applications to Run on Amazon EC2 Windows Instances](#) (p. 12)
- [Getting Started with AWS: Hosting a .NET Web App](#)

If you have questions about whether AWS is right for you, [contact AWS Sales](#). If you have technical questions about Amazon EC2, use the [Amazon EC2 forum](#).

Related Services

You can provision Amazon EC2 resources, such as instances and volumes, directly using Amazon EC2. You can also provision Amazon EC2 resources using other services in AWS. For more information, see the following documentation:

- [Auto Scaling User Guide](#)
- [AWS CloudFormation User Guide](#)
- [AWS Elastic Beanstalk Developer Guide](#)
- [AWS OpsWorks User Guide](#)

To automatically distribute incoming application traffic across multiple instances, use Elastic Load Balancing. For more information, see [Elastic Load Balancing User Guide](#).

To monitor basic statistics for your instances and Amazon EBS volumes, use Amazon CloudWatch. For more information, see the [Amazon CloudWatch User Guide](#).

To monitor the calls made to the Amazon EC2 API for your account, including calls made by the AWS Management Console, command line tools, and other services, use AWS CloudTrail. For more information, see the [AWS CloudTrail User Guide](#).

To get a managed relational database in the cloud, use Amazon Relational Database Service (Amazon RDS) to launch a database instance. Although you can set up a database on an EC2 instance, Amazon RDS offers the advantage of handling your database management tasks, such as patching the software, backing up, and storing the backups. For more information, see [Amazon Relational Database Service Developer Guide](#).

To import virtual machine (VM) images from your local environment into AWS and convert them into ready-to-use AMIs or instances, use VM Import/Export. For more information, see the [VM Import/Export User Guide](#).

Accessing Amazon EC2

Amazon EC2 provides a web-based user interface, the Amazon EC2 console. If you've signed up for an AWS account, you can access the Amazon EC2 console by signing into the AWS Management Console and selecting **EC2** from the console home page.

If you prefer to use a command line interface, you have the following options:

AWS Command Line Interface (CLI)

Provides commands for a broad set of AWS products, and is supported on Windows, Mac, and Linux. To get started, see [AWS Command Line Interface User Guide](#). For more information about the commands for Amazon EC2, see [ec2](#) in the *AWS Command Line Interface Reference*.

AWS Tools for Windows PowerShell

Provides commands for a broad set of AWS products for those who script in the PowerShell environment. To get started, see the [AWS Tools for Windows PowerShell User Guide](#). For more information about the cmdlets for Amazon EC2, see the [AWS Tools for Windows PowerShell Reference](#).

Amazon EC2 provides a Query API. These requests are HTTP or HTTPS requests that use the HTTP verbs GET or POST and a Query parameter named `Action`. For more information about the API actions for Amazon EC2, see [Actions](#) in the *Amazon EC2 API Reference*.

If you prefer to build applications using language-specific APIs instead of submitting a request over HTTP or HTTPS, AWS provides libraries, sample code, tutorials, and other resources for software

developers. These libraries provide basic functions that automate tasks such as cryptographically signing your requests, retrying requests, and handling error responses, making it is easier for you to get started. For more information, see [AWS SDKs and Tools](#).

Pricing for Amazon EC2

When you sign up for AWS, you can get started with Amazon EC2 for free using the [AWS Free Tier](#).

Amazon EC2 provides the following purchasing options for instances:

On-Demand instances

Pay for the instances that you use by the hour, with no long-term commitments or up-front payments.

Reserved Instances

Make a low, one-time, up-front payment for an instance, reserve it for a one- or three-year term, and pay a significantly lower hourly rate for these instances.

Spot instances

Specify the maximum hourly price that you are willing to pay to run a particular instance type. The Spot price fluctuates based on supply and demand, but you never pay more than the maximum price you specified. If the Spot price moves higher than your maximum price, Amazon EC2 shuts down your Spot instances.

For a complete list of charges and specific prices for Amazon EC2, see [Amazon EC2 Pricing](#).

To calculate the cost of a sample provisioned environment, see [AWS Economics Center](#).

To see your bill, go to your [AWS Account Activity page](#). Your bill contains links to usage reports that provide details about your bill. To learn more about AWS account billing, see [AWS Account Billing](#).

If you have questions concerning AWS billing, accounts, and events, [contact AWS Support](#).

For an overview of Trusted Advisor, a service that helps you optimize the costs, security, and performance of your AWS environment, see [AWS Trusted Advisor](#).

PCI DSS Compliance

Amazon EC2 supports the processing, storage, and transmission of credit card data by a merchant or service provider, and has been validated as being compliant with Payment Card Industry (PCI) Data Security Standard (DSS). For more information about PCI DSS, including how to request a copy of the AWS PCI Compliance Package, see [PCI DSS Level 1](#).

Amazon EC2 Basic Infrastructure for Windows

As you get started with Amazon EC2, you'll benefit from understanding the components of its basic infrastructure and how they compare or contrast with your own data centers.

Concepts

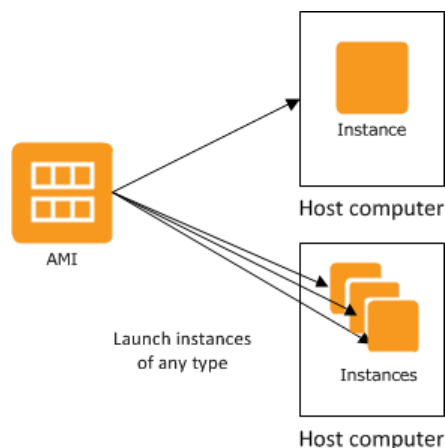
- [Amazon Machine Images and Instances \(p. 5\)](#)
- [Regions and Availability Zones \(p. 6\)](#)
- [Storage \(p. 6\)](#)
- [Root Device Volume \(p. 8\)](#)
- [Networking and Security \(p. 10\)](#)
- [AWS Identity and Access Management \(p. 10\)](#)
- [Differences between Windows Server and an Amazon EC2 Windows Instance \(p. 10\)](#)
- [Designing Your Applications to Run on Amazon EC2 Windows Instances \(p. 12\)](#)

Amazon Machine Images and Instances

An *Amazon Machine Image (AMI)* is a template that contains a software configuration (for example, an operating system, an application server, and applications). From an AMI, you launch *instances*, which are copies of the AMI running as virtual servers in the cloud.

Amazon publishes many AMIs that contain common software configurations for public use. In addition, members of the AWS developer community have published their own custom AMIs. You can also create your own custom AMI or AMIs; doing so enables you to quickly and easily start new instances that have everything you need. For example, if your application is a website or web service, your AMI could include a web server, the associated static content, and the code for the dynamic pages. As a result, after you launch an instance from this AMI, your web server starts, and your application is ready to accept requests.

You can launch different types of instances from a single AMI. An *instance type* essentially determines the hardware of the host computer used for your instance. Each instance type offers different compute and memory facilities. Select an instance type based on the amount of memory and computing power that you need for the applications or software that you plan to run on the instance. For more information about the hardware specifications for each Amazon EC2 instance type, see [Amazon EC2 Instances](#). You can also launch multiple instances from an AMI, as shown in the following figure.



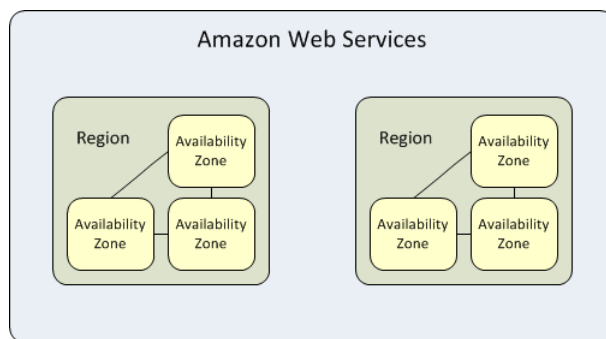
Your Windows instances keep running until you stop or terminate them, or until they fail. If an instance fails, you can launch a new one from the AMI.

Your AWS account has a limit on the number of instances that you can have running. For more information about this limit, and how to request an increase, see [How many instances can I run in Amazon EC2](#) in the Amazon EC2 General FAQ.

Regions and Availability Zones

Amazon has data centers in different areas of the world (for example, North America, Europe, and Asia). Correspondingly, Amazon EC2 is available to use in different *regions*. By launching instances in separate regions, you can design your application to be closer to specific customers or to meet legal or other requirements. Prices for Amazon EC2 usage vary by region (for more information about pricing by region, see [Amazon EC2 Pricing](#)).

Each region contains multiple distinct locations called *Availability Zones*. Each Availability Zone is engineered to be isolated from failures in other Availability Zones, and to provide inexpensive, low-latency network connectivity to other zones in the same region. By launching instances in separate Availability Zones, you can protect your applications from the failure of a single location.



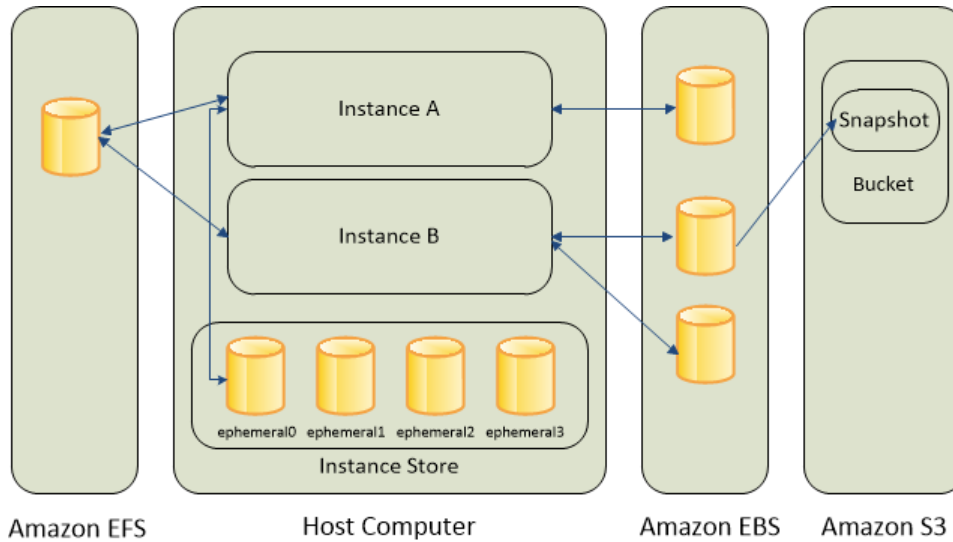
For more information about the available regions and Availability Zones, see [Using Regions and Availability Zones](#) in the *Amazon EC2 User Guide for Linux Instances*.

Storage

When using Amazon EC2, you may have data that you need to store. Amazon EC2 offers the following storage options:

- [Amazon Elastic Block Store \(Amazon EBS\)](#)
- [Amazon EC2 Instance Store \(p. 822\)](#)
- [Amazon Simple Storage Service \(Amazon S3\)](#)

The following figure shows the relationship between these types of storage.

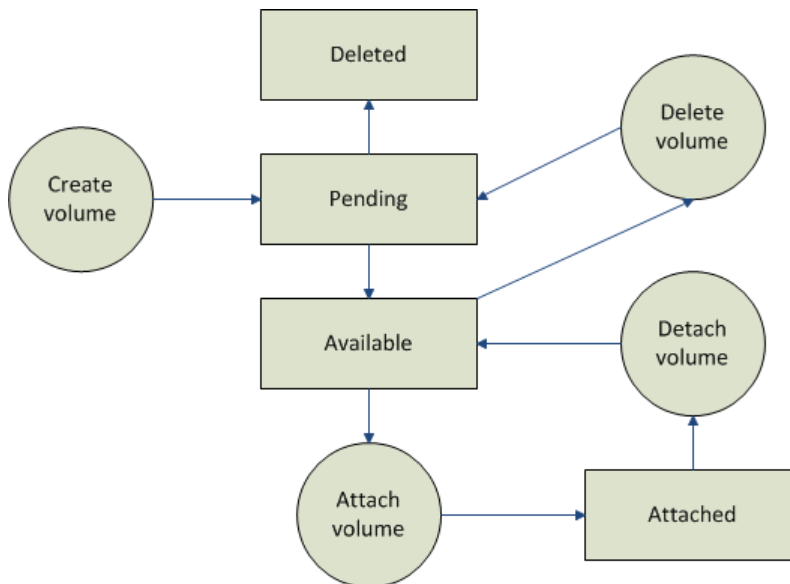


Amazon EBS Volumes

Amazon EBS volumes are the recommended storage option for the majority of use cases. Amazon EBS provides your instances with persistent, block-level storage. Amazon EBS volumes are essentially hard disks that you can attach to a running instance.

Amazon EBS is especially suited for applications that require a database, a file system, or access to raw block-level storage.

As illustrated in the previous figure, you can attach multiple volumes to an instance. Also, to keep a backup copy of your data, you can create a *snapshot* of an EBS volume, which is stored in Amazon S3. You can create a new Amazon EBS volume from a snapshot, and attach it to another instance. You can also detach a volume from an instance and attach it to a different instance. The following figure illustrates the life cycle of an EBS volume.



For more information about Amazon EBS volumes, see [Amazon Elastic Block Store \(Amazon EBS\)](#) (p. 745).

Instance Store

All instance types, with the exception of Micro instances, offer *instance store*, which provides your instances with temporary, block-level storage. This is storage that is physically attached to the host computer. The data on an instance store volume doesn't persist when the associated instance is stopped or terminated. For more information about instance store volumes, see [Amazon EC2 Instance Store \(p. 822\)](#).

Instance store is an option for inexpensive temporary storage. You can use instance store volumes if you don't require data persistence.

Amazon S3

Amazon S3 is storage for the Internet. It provides a simple web service interface that enables you to store and retrieve any amount of data from anywhere on the web. For more information about Amazon S3, see the [Amazon S3 product page](#).

Root Device Volume

When you launch an instance, the *root device volume* contains the image used to boot the instance. You can launch an Amazon EC2 Windows instance using an AMI backed either by instance store or by Amazon Elastic Block Store (Amazon EBS).

- **Instances launched from an AMI backed by Amazon EBS** use an Amazon EBS volume as the root device. The root device volume of an Amazon EBS-backed AMI is an Amazon EBS snapshot. When an instance is launched using an Amazon EBS-backed AMI, a root EBS volume is created from the EBS snapshot and attached to the instance. The root device volume is then used to boot the instance.
- **Instances launched from an AMI backed by instance store** use an instance store volume as the root device. The image of the root device volume of an instance store-backed AMI is initially stored in Amazon S3. When an instance is launched using an instance store-backed AMI, the image of its root device is copied from Amazon S3 to the root partition of the instance. The root device volume is then used to boot the instance.

Important

The only Windows AMIs that can be backed by instance store are those for Windows Server 2003. Instance store-backed instances don't have the available disk space required for later versions of Windows Server.

For a summary of the differences between instance store-backed AMIs and Amazon EBS-backed AMIs, see [Storage for the Root Device \(p. 64\)](#).

Determining the Root Device Type of an AMI

You can determine the root device type of an AMI using the console or the command line.

To determine the root device type of an AMI using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **AMIs**, and select the AMI.
3. Check the value of **Root Device Type** in the **Details** tab as follows:
 - If the value is `ebs`, this is an Amazon EBS-backed AMI.
 - If the value is `instance store`, this is an instance store-backed AMI.

To determine the root device type of an AMI using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [describe-images](#) (AWS CLI)
- [Get-EC2Image](#) (AWS Tools for Windows PowerShell)

Determining the Root Device Type of an Instance

You can determine the root device type of an instance using the console or the command line.

To determine the root device type of an instance using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **Instances**, and select the instance.
3. Check the value of **Root device type** in the **Description** tab as follows:
 - If the value is `ebs`, this is an Amazon EBS-backed instance.
 - If the value is `instance store`, this is an instance store-backed instance.

To determine the root device type of an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [describe-instances](#) (AWS CLI)
- [Get-EC2Instance](#) (AWS Tools for Windows PowerShell)

Changing the Root Device Volume to Persist

Using the console, you can change the `DeleteOnTermination` attribute when you launch an instance. To change this attribute for a running instance, you must use the command line.

To change the root device volume of an instance to persist at launch using the console

1. Open the Amazon EC2 console.
2. From the Amazon EC2 console dashboard, click **Launch Instance**.
3. On the **Choose an Amazon Machine Image (AMI)** page, choose the AMI to use and click **Select**.
4. Follow the wizard to complete the **Choose an Instance Type** and **Configure Instance Details** pages.
5. On the **Add Storage** page, deselect the **Delete On Termination** check box for the root volume.
6. Complete the remaining wizard pages, and then click **Launch**.

You can verify the setting by viewing details for the root device volume on the instance's details pane. Next to **Block devices**, click the entry for the root device volume. By default, **Delete on termination** is `True`. If you change the default behavior, **Delete on termination** is `False`.

To change the root device volume of an instance to persist using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [modify-instance-attribute](#) (AWS CLI)

- [Edit-EC2InstanceAttribute](#) (AWS Tools for Windows PowerShell)

Networking and Security

You can launch instances in one of two platforms: EC2-Classic and EC2-VPC. An instance that's launched into EC2-Classic is assigned a public IPv4 address. By default, an instance that's launched into EC2-VPC is assigned public IPv4 address only if it's launched into a default VPC. An instance that's launched into a nondefault VPC must be specifically assigned a public IPv4 address at launch, or you must modify your subnet's default public IPv4 addressing behavior. For more information about EC2-Classic and EC2-VPC, see [Supported Platforms](#) (p. 672).

Instances can fail or terminate for reasons outside of your control. If one fails and you launch a replacement instance, the replacement has a different public IPv4 address than the original. However, if your application needs a static IPv4 address, Amazon EC2 offers *Elastic IP addresses*. For more information, see [Amazon EC2 Instance IP Addressing](#) (p. 693).

You can use *security groups* to control who can access your instances. These are analogous to an inbound network firewall that enables you to specify the protocols, ports, and source IP ranges that are allowed to reach your instances. You can create multiple security groups and assign different rules to each group. You can then assign each instance to one or more security groups, and we use the rules to determine which traffic is allowed to reach the instance. You can configure a security group so that only specific IP addresses or specific security groups have access to the instance. For more information, see [Amazon EC2 Security Groups for Windows Instances](#) (p. 606).

AWS Identity and Access Management

AWS Identity and Access Management (IAM) enables you to do the following:

- Create users and groups under your AWS account
- Assign unique security credentials to each user under your AWS account
- Control each user's permissions to perform tasks using AWS resources
- Allow the users in another AWS account to share your AWS resources
- Create roles for your AWS account and define the users or services that can assume them
- Use existing identities for your enterprise to grant permissions to perform tasks using AWS resources

By using IAM with Amazon EC2, you can control whether users in your organization can perform a task using specific Amazon EC2 API actions and whether they can use specific AWS resources.

For more information about IAM, see the following:

- [Creating an IAM Group and Users](#) (p. 620)
- [IAM Policies for Amazon EC2](#) (p. 621)
- [IAM Roles for Amazon EC2](#) (p. 658)
- [Identity and Access Management \(IAM\)](#)
- [IAM User Guide](#)

Differences between Windows Server and an Amazon EC2 Windows Instance

After you launch your Amazon EC2 Windows instance, it behaves like a traditional server running Windows Server. For example, both Windows Server and an Amazon EC2 instance can be used to

Amazon Elastic Compute Cloud
User Guide for Windows Instances
Differences between Windows Server
and an Amazon EC2 Windows Instance

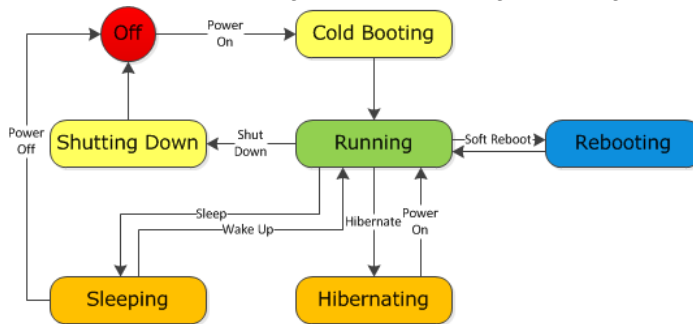
run your web applications, conduct batch processing, or manage applications requiring large-scale computations. However, there are important differences between the server hardware model and the cloud computing model. The way an Amazon EC2 instance runs is not the same as the way a traditional server running Windows Server runs.

Before you begin launching Amazon EC2 Windows instances, you should be aware that the architecture of applications running on cloud servers can differ significantly from the architecture for traditional application models running on your hardware. Implementing applications on cloud servers requires a shift in your design process.

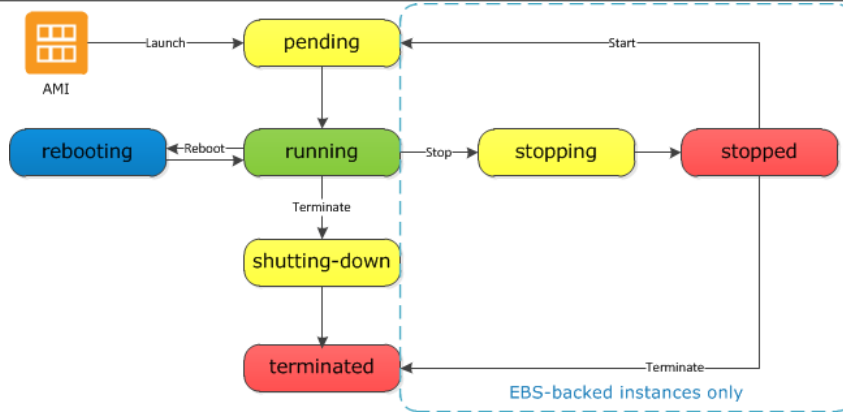
The following table describes some key differences between Windows Server and an Amazon EC2 Windows instance.

Windows Server	Amazon EC2 Windows Instance
Resources and capacity are physically limited.	Resources and capacity are scalable.
You pay for the infrastructure, even if you don't use it.	You pay for the usage of the infrastructure. We stop charging you for the instance as soon as you stop or terminate it.
Occupies physical space and must be maintained on a regular basis.	Doesn't occupy physical space and does not require regular maintenance.
Starts with push of the power button (known as <i>cold booting</i>).	Starts with the launch of the instance.
You can keep the server running until it is time to shut it down, or put it in a sleep or hibernation state (during which the server is powered down).	You can keep the server running, or stop and restart it (during which the instance is moved to a new host computer).
When you shut down the server, all resources remain intact and in the state they were in when you switched it off. Information you stored on the hard drives persists and can be accessed whenever it's needed. You can restore the server to the running state by powering it on.	When you terminate the instance, its infrastructure is no longer available to you. You can't connect to or restart an instance after you've terminated it. However, you can create an image from your instance while it's running, and launch new instances from the image at any time.

A traditional server running Windows Server goes through the states shown in the following diagram.



An Amazon EC2 Windows instance is similar to the traditional Windows Server, as you can see by comparing the following diagram with the previous diagram for Windows Server. After you launch an instance, it briefly goes into the pending state while registration takes place, then it goes into the running state. The instance remains active until you stop or terminate it. You can't restart an instance after you terminate it. You can create a backup image of your instance while it's running, and launch a new instance from that backup image.



Designing Your Applications to Run on Amazon EC2 Windows Instances

It is important that you consider the differences mentioned in the previous section when you design your applications to run on Amazon EC2 Windows instances.

Applications built for Amazon EC2 use the underlying computing infrastructure on an as-needed basis. They draw on necessary resources (such as storage and computing) on demand in order to perform a job, and relinquish the resources when done. In addition, they often dispose of themselves after the job is done. While in operation, the application scales up and down elastically based on resource requirements. An application running on an Amazon EC2 instance can terminate and recreate the various components at will in case of infrastructure failures.

When designing your Windows applications to run on Amazon EC2, you can plan for rapid deployment and rapid reduction of compute and storage resources, based on your changing needs.

When you run an Amazon EC2 Windows instance, you don't need to provision the exact system package of hardware, software, and storage, the way you do with Windows Server. Instead, you can focus on using a variety of cloud resources to improve the scalability and overall performance of your Windows application.

With Amazon EC2, designing for failure and outages is an integral and crucial part of the architecture. As with any scalable and redundant system, architecture of your system should account for computing, network, and storage failures. You have to build mechanisms in your applications that can handle different kinds of failures. The key is to build a modular system with individual components that are not tightly coupled, can interact asynchronously, and treat one another as black boxes that are independently scalable. Thus, if one of your components fails or is busy, you can launch more instances of that component without breaking your current system.

Another key element to designing for failure is to distribute your application geographically. Replicating your application across geographically distributed regions improves high availability in your system.

Amazon EC2 infrastructure is programmable and you can use scripts to automate the deployment process, to install and configure software and applications, and to bootstrap your virtual servers.

You should implement security in every layer of your application architecture running on an Amazon EC2 Windows instance. If you are concerned about storing sensitive and confidential data within your Amazon EC2 environment, you should encrypt the data before uploading it.

Setting Up with Amazon EC2

If you've already signed up for Amazon Web Services (AWS), you can start using Amazon EC2 immediately. You can open the Amazon EC2 console, click **Launch Instance**, and follow the steps in the launch wizard to launch your first instance.

If you haven't signed up for AWS yet, or if you need assistance launching your first instance, complete the following tasks to get set up to use Amazon EC2:

1. [Sign Up for AWS \(p. 13\)](#)
2. [Create an IAM User \(p. 13\)](#)
3. [Create a Key Pair \(p. 15\)](#)
4. [Create a Virtual Private Cloud \(VPC\) \(p. 17\)](#)
5. [Create a Security Group \(p. 17\)](#)

Sign Up for AWS

When you sign up for Amazon Web Services (AWS), your AWS account is automatically signed up for all services in AWS, including Amazon EC2. You are charged only for the services that you use.

With Amazon EC2, you pay only for what you use. If you are a new AWS customer, you can get started with Amazon EC2 for free. For more information, see [AWS Free Tier](#).

If you have an AWS account already, skip to the next task. If you don't have an AWS account, use the following procedure to create one.

To create an AWS account

1. Open <http://aws.amazon.com/>, and then choose **Create an AWS Account**.
2. Follow the online instructions.

Part of the sign-up procedure involves receiving a phone call and entering a PIN using the phone keypad.

Note your AWS account number, because you'll need it for the next task.

Create an IAM User

Services in AWS, such as Amazon EC2, require that you provide credentials when you access them, so that the service can determine whether you have permission to access its resources. The console

requires your password. You can create access keys for your AWS account to access the command line interface or API. However, we don't recommend that you access AWS using the credentials for your AWS account; we recommend that you use AWS Identity and Access Management (IAM) instead. Create an IAM user, and then add the user to an IAM group with administrative permissions or and grant this user administrative permissions. You can then access AWS using a special URL and the credentials for the IAM user.

If you signed up for AWS but have not created an IAM user for yourself, you can create one using the IAM console. If you aren't familiar with using the console, see [Working with the AWS Management Console](#) for an overview.

To create an IAM user for yourself and add the user to an Administrators group

1. Sign in to the Identity and Access Management (IAM) console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Users**, and then choose **Add user**.
3. For **User name**, type a user name, such as **Administrator**. The name can consist of letters, digits, and the following characters: plus (+), equal (=), comma (,), period (.), at (@), underscore (_), and hyphen (-). The name is not case sensitive and can be a maximum of 64 characters in length.
4. Select the check box next to **AWS Management Console access**, select **Custom password**, and then type the new user's password in the text box. You can optionally select **Require password reset** to force the user to select a new password the next time the user signs in.
5. Choose **Next: Permissions**.
6. On the **Set permissions for user** page, choose **Add user to group**.
7. Choose **Create group**.
8. In the **Create group** dialog box, type the name for the new group. The name can consist of letters, digits, and the following characters: plus (+), equal (=), comma (,), period (.), at (@), underscore (_), and hyphen (-). The name is not case sensitive and can be a maximum of 128 characters in length.
9. For **Filter**, choose **Job function**.
10. In the policy list, select the check box for **AdministratorAccess**. Then choose **Create group**.
11. Back in the list of groups, select the check box for your new group. Choose **Refresh** if necessary to see the group in the list.
12. Choose **Next: Review** to see the list of group memberships to be added to the new user. When you are ready to proceed, choose **Add permissions**.

You can use this same process to create more groups and users, and to give your users access to your AWS account resources. To learn about using policies to restrict users' permissions to specific AWS resources, go to [Access Management](#) and [Example Policies for Administering AWS Resources](#).

To sign in as this new IAM user, sign out of the AWS console, then use the following URL, where *your_aws_account_id* is your AWS account number without the hyphens (for example, if your AWS account number is 1234-5678-9012, your AWS account ID is 123456789012):

```
https://your_aws_account_id.signin.aws.amazon.com/console/
```

Enter the IAM user name (not your email address) and password that you just created. When you're signed in, the navigation bar displays "*your_user_name* @ *your_aws_account_id*".

If you don't want the URL for your sign-in page to contain your AWS account ID, you can create an account alias. From the IAM console, click **Dashboard** in the navigation pane. From the dashboard, click **Customize** and enter an alias such as your company name. To sign in after you create an account alias, use the following URL:

```
https://your_account_alias.signin.aws.amazon.com/console/
```

To verify the sign-in link for IAM users for your account, open the IAM console and check under **IAM users sign-in link** on the dashboard.

For more information about IAM, see [IAM and Amazon EC2 \(p. 620\)](#).

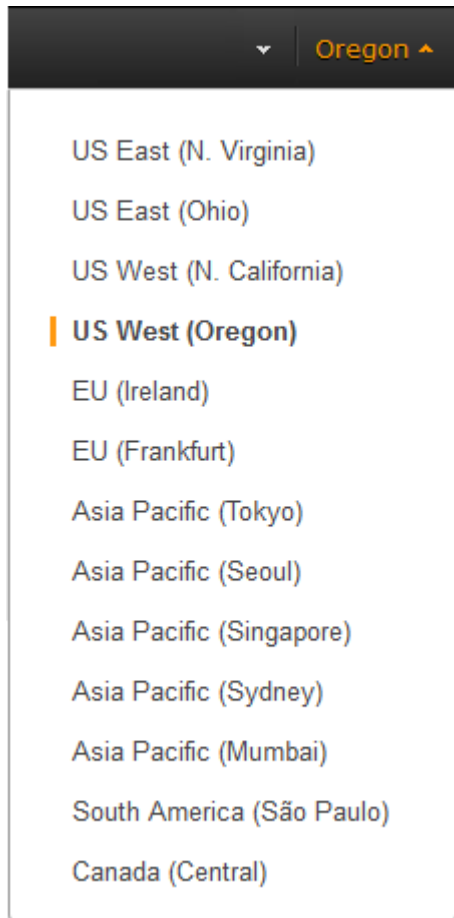
Create a Key Pair

AWS uses public-key cryptography to secure the login information for your instance. You specify the name of the key pair when you launch your instance, then provide the private key to obtain the administrator password for your Windows instance so you can log in using RDP.

If you haven't created a key pair already, you can create one using the Amazon EC2 console. Note that if you plan to launch instances in multiple regions, you'll need to create a key pair in each region. For more information about regions, see [Regions and Availability Zones \(p. 6\)](#).

To create a key pair

1. Sign in to AWS using the URL that you created in the previous section.
2. From the AWS dashboard, choose **EC2** to open the Amazon EC2 console.
3. From the navigation bar, select a region for the key pair. You can select any region that's available to you, regardless of your location. However, key pairs are specific to a region; for example, if you plan to launch an instance in the US West (Oregon) Region, you must create a key pair for the instance in the US West (Oregon) Region.



4. In the navigation pane, under **NETWORK & SECURITY**, click **Key Pairs**.

Tip

The navigation pane is on the left side of the console. If you do not see the pane, it might be minimized; click the arrow to expand the pane. You may have to scroll down to see the **Key Pairs** link.



5. Click **Create Key Pair**.
6. Enter a name for the new key pair in the **Key pair name** field of the **Create Key Pair** dialog box, and then click **Create**. Choose a name that is easy for you to remember, such as your IAM user name, followed by `-key-pair`, plus the region name. For example, `me-key-pair-uswest2`.

7. The private key file is automatically downloaded by your browser. The base file name is the name you specified as the name of your key pair, and the file name extension is `.pem`. Save the private key file in a safe place.

Important

This is the only chance for you to save the private key file. You'll need to provide the name of your key pair when you launch an instance and the corresponding private key each time you connect to the instance.

For more information, see [Amazon EC2 Key Pairs and Windows Instances \(p. 602\)](#).

Create a Virtual Private Cloud (VPC)

Amazon VPC enables you to launch AWS resources into a virtual network that you've defined. If you have a default VPC, you can skip this section and move to the next task, [Create a Security Group \(p. 17\)](#). To determine whether you have a default VPC, see [Supported Platforms in the Amazon EC2 Console \(p. 672\)](#). Otherwise, you can create a nondefault VPC in your account using the steps below.

Important

If your account supports EC2-Classic in a region, then you do not have a default VPC in that region. T2 instances must be launched into a VPC.

To create a nondefault VPC

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. From the navigation bar, select a region for the VPC. VPCs are specific to a region, so you should select the same region in which you created your key pair.
3. On the VPC dashboard, click **Start VPC Wizard**.
4. On the **Step 1: Select a VPC Configuration** page, ensure that **VPC with a Single Public Subnet** is selected, and click **Select**.
5. On the **Step 2: VPC with a Single Public Subnet** page, enter a friendly name for your VPC in the **VPC name** field. Leave the other default configuration settings, and click **Create VPC**. On the confirmation page, click **OK**.

For more information about Amazon VPC, see [What is Amazon VPC?](#) in the *Amazon VPC User Guide*.

Create a Security Group

Security groups act as a firewall for associated instances, controlling both inbound and outbound traffic at the instance level. You must add rules to a security group that enable you to connect to your instance from your IP address using RDP. You can also add rules that allow inbound and outbound HTTP and HTTPS access from anywhere.

Note that if you plan to launch instances in multiple regions, you'll need to create a security group in each region. For more information about regions, see [Regions and Availability Zones \(p. 6\)](#).

Prerequisites

You'll need the public IPv4 address of your local computer. The security group editor in the Amazon EC2 console can automatically detect the public IPv4 address for you. Alternatively, you can use the search phrase "what is my IP address" in an Internet browser, or use the following service: <http://checkip.amazonaws.com/>. If you are connecting through an Internet service provider (ISP) or from

behind a firewall without a static IP address, you need to find out the range of IP addresses used by client computers.

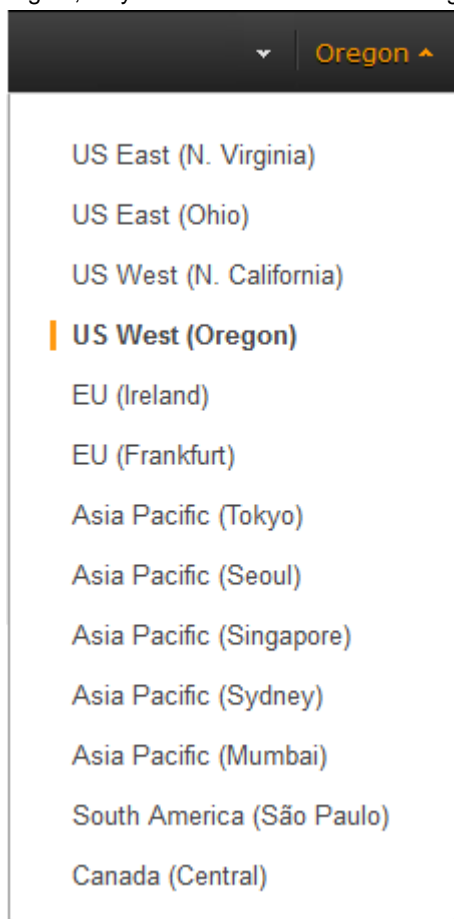
To create a security group with least privilege

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

Tip

Alternatively, you can use the Amazon VPC console to create a security group. However, the instructions in this procedure don't match the Amazon VPC console. Therefore, if you switched to the Amazon VPC console in the previous section, either switch back to the Amazon EC2 console and use these instructions, or use the instructions in [Set Up a Security Group for Your VPC](#) in the *Amazon VPC Getting Started Guide*.

2. From the navigation bar, select a region for the security group. Security groups are specific to a region, so you should select the same region in which you created your key pair.



3. Click **Security Groups** in the navigation pane.
4. Click **Create Security Group**.
5. Enter a name for the new security group and a description. Choose a name that is easy for you to remember, such as your IAM user name, followed by `_SG_`, plus the region name. For example, `me_SG_uswest2`.
6. In the **VPC** list, select your VPC. If you have a default VPC, it's the one that is marked with an asterisk (*).

Note

If your account supports EC2-Classic, select the VPC that you created in the previous task.

7. On the **Inbound** tab, create the following rules (click **Add Rule** for each new rule), and then click **Create**:
 - Select **HTTP** from the **Type** list, and make sure that **Source** is set to **Anywhere** (0.0.0.0/0).
 - Select **HTTPS** from the **Type** list, and make sure that **Source** is set to **Anywhere** (0.0.0.0/0).
 - Select **RDP** from the **Type** list. In the **Source** box, choose **My IP** to automatically populate the field with the public IPv4 address of your local computer. Alternatively, choose **Custom** and specify the public IPv4 address of your computer or network in CIDR notation. To specify an individual IP address in CIDR notation, add the routing suffix /32, for example, 203.0.113.25/32. If your company allocates addresses from a range, specify the entire range, such as 203.0.113.0/24.

Caution

For security reasons, we don't recommend that you allow RDP access from all IPv4 addresses (0.0.0.0/0) to your instance, except for testing purposes and only for a short time.

For more information, see [Amazon EC2 Security Groups for Windows Instances \(p. 606\)](#).

Getting Started with Amazon EC2 Windows Instances

Let's get started with Amazon Elastic Compute Cloud (Amazon EC2) by launching, connecting to, and using a Windows instance. An *instance* is a virtual server in the AWS cloud. With Amazon EC2, you can set up and configure the operating system and applications that run on your instance.

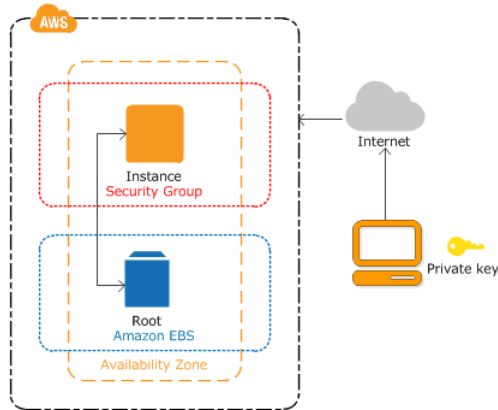
When you sign up for AWS, you can get started with Amazon EC2 for free using the [AWS Free Tier](#). If you created your AWS account less than 12 months ago, and have not already exceeded the free tier benefits for Amazon EC2, it will not cost you anything to complete this tutorial, because we help you select options that are within the free tier benefits. Otherwise, you'll incur the standard Amazon EC2 usage fees from the time that you launch the instance until you terminate the instance (which is the final task of this tutorial), even if it remains idle.

Contents

- [Overview \(p. 20\)](#)
- [Prerequisites \(p. 21\)](#)
- [Step 1: Launch an Instance \(p. 21\)](#)
- [Step 2: Connect to Your Instance \(p. 22\)](#)
- [Step 3: Clean Up Your Instance \(p. 24\)](#)
- [Next Steps \(p. 24\)](#)

Overview

The instance is an Amazon EBS-backed instance (meaning that the root volume is an EBS volume). You can either specify the Availability Zone in which your instance runs, or let Amazon EC2 select an Availability Zone for you. When you launch your instance, you secure it by specifying a key pair and security group. When you connect to your instance, you must specify the private key of the key pair that you specified when launching your instance.



Tasks

To complete this tutorial, perform the following tasks:

1. [Launch an Instance](#) (p. 21)
2. [Connect to Your Instance](#) (p. 22)
3. [Clean Up Your Instance](#) (p. 24)

Related Tutorials

- If you'd prefer to launch a Linux instance, see this tutorial in the *Amazon EC2 User Guide for Linux Instances*: [Getting Started with Amazon EC2 Linux Instances](#).
- If you'd prefer to use the command line, see this tutorial in the *AWS Command Line Interface User Guide*: [Using Amazon EC2 through the AWS CLI](#).

Prerequisites

Before you begin, be sure that you've completed the steps in [Setting Up with Amazon EC2](#) (p. 13).

Step 1: Launch an Instance

You can launch a Windows instance using the AWS Management Console as described in the following procedure. This tutorial is intended to help you launch your first instance quickly, so it doesn't cover all possible options. For more information about the advanced options, see [Launching an Instance](#).

To launch an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the console dashboard, choose **Launch Instance**.
3. The **Choose an Amazon Machine Image (AMI)** page displays a list of basic configurations, called *Amazon Machine Images (AMIs)*, that serve as templates for your instance. Select the AMI for Windows Server 2012 R2 Base or Windows Server 2008 R2 Base. Notice that these AMIs are marked "Free tier eligible."
4. On the **Choose an Instance Type** page, you can select the hardware configuration of your instance. Select the `t2.micro` type, which is selected by default. Notice that this instance type is eligible for the free tier.

Note

[T2 instances](#), such as `t2.micro`, must be launched into a VPC. If your AWS account supports EC2-Classic and you do not have a VPC in the selected region, the launch wizard creates a VPC for you and you can continue to the next step. Otherwise, the **Review and Launch** button is disabled and you must choose **Next: Configure Instance Details** and follow the directions to select a subnet.

5. Choose **Review and Launch** to let the wizard complete the other configuration settings for you.
6. On the **Review Instance Launch** page, under **Security Groups**, you'll see that the wizard created and selected a security group for you. You can use this security group, or alternatively you can select the security group that you created when getting set up using the following steps:
 - a. Choose **Edit security groups**.
 - b. On the **Configure Security Group** page, ensure that **Select an existing security group** is selected.
 - c. Select your security group from the list of existing security groups, and then choose **Review and Launch**.
7. On the **Review Instance Launch** page, choose **Launch**.
8. When prompted for a key pair, select **Choose an existing key pair**, then select the key pair that you created when getting set up.

Alternatively, you can create a new key pair. Select **Create a new key pair**, enter a name for the key pair, and then choose **Download Key Pair**. This is the only chance for you to save the private key file, so be sure to download it. Save the private key file in a safe place. You'll need to provide the name of your key pair when you launch an instance and the corresponding private key each time you connect to the instance.

Caution

Don't select the **Proceed without a key pair** option. If you launch your instance without a key pair, then you can't connect to it.

When you are ready, select the acknowledgement check box, and then choose **Launch Instances**.

9. A confirmation page lets you know that your instance is launching. Choose **View Instances** to close the confirmation page and return to the console.
10. On the **Instances** screen, you can view the status of the launch. It takes a short time for an instance to launch. When you launch an instance, its initial state is `pending`. After the instance starts, its state changes to `running` and it receives a public DNS name. (If the **Public DNS (IPv4)** column is hidden, choose the Show/Hide icon in the top right corner of the page and then select **Public DNS (IPv4)**.)
11. It can take a few minutes for the instance to be ready so that you can connect to it. Check that your instance has passed its status checks; you can view this information in the **Status Checks** column.

Step 2: Connect to Your Instance

To connect to a Windows instance, you must retrieve the initial administrator password and then specify this password when you connect to your instance using Remote Desktop.

Note

If you've joined your instance to a domain, you can connect to your instance using domain credentials you've defined in AWS Directory Service. For more information about connecting to an instance in a domain, see [Connecting To Your Instance Using Domain Credentials](#) (p. 338).

The name of the administrator account depends on the language of the operating system. For example, for English, it's Administrator, for French it's Administrateur, and for Portuguese it's Administrador. For more information, see [Localized Names for Administrator Account in Windows](#) in the Microsoft TechNet Wiki.

The license for the Windows Server operating system (OS) allows two simultaneous remote connections for administrative purposes. The license for Windows Server is included in the price of your EC2 instance. If you need more than two simultaneous remote connections you must purchase a Remote Desktop Services (RDS) license. If you attempt a third connection, an error occurs. For more information, see [Configure the Number of Simultaneous Remote Connections Allowed for a Connection](#).

To connect to your Windows instance using an RDP client

1. In the Amazon EC2 console, select the instance, and then choose **Connect**.
2. In the **Connect To Your Instance** dialog box, choose **Get Password** (it will take a few minutes after the instance is launched before the password is available).
3. Choose **Browse** and navigate to the private key file you created when you launched the instance. Select the file and choose **Open** to copy the entire contents of the file into the **Contents** field.
4. Choose **Decrypt Password**. The console displays the default administrator password for the instance in the **Connect To Your Instance** dialog box, replacing the link to **Get Password** shown previously with the actual password.
5. Record the default administrator password, or copy it to the clipboard. You need this password to connect to the instance.
6. Choose **Download Remote Desktop File**. Your browser prompts you to either open or save the .rdp file. Either option is fine. When you have finished, you can choose **Close** to dismiss the **Connect To Your Instance** dialog box.
 - If you opened the .rdp file, you'll see the **Remote Desktop Connection** dialog box.
 - If you saved the .rdp file, navigate to your downloads directory, and open the .rdp file to display the dialog box.
7. You may get a warning that the publisher of the remote connection is unknown. If you are using **Remote Desktop Connection** from a Windows PC, choose **Connect** to connect to your instance. If you are using **Microsoft Remote Desktop** on a Mac, skip the next step.
8. When prompted, log in to the instance, using the administrator account for the operating system and the password that you recorded or copied previously. If your **Remote Desktop Connection** already has an administrator account set up, you might have to choose the **Use another account** option and enter the user name and password manually.

Note

Sometimes copying and pasting content can corrupt data. If you encounter a "Password Failed" error when you log in, try typing in the password manually.

9. Due to the nature of self-signed certificates, you may get a warning that the security certificate could not be authenticated. Use the following steps to verify the identity of the remote computer, or simply choose **Yes** or **Continue** to continue if you trust the certificate.
 - a. If you are using **Remote Desktop Connection** from a Windows PC, choose **View certificate**. If you are using **Microsoft Remote Desktop** on a Mac, choose **Show Certificate**.
 - b. Choose the **Details** tab, and scroll down to the **Thumbprint** entry on a Windows PC, or the **SHA1 Fingerprints** entry on a Mac. This is the unique identifier for the remote computer's security certificate.
 - c. In the Amazon EC2 console, select the instance, choose **Actions**, and then choose **Get System Log**.
 - d. In the system log output, look for an entry labeled `RDPCERTIFICATE-THUMBPRINT`. If this value matches the thumbprint or fingerprint of the certificate, you have verified the identity of the remote computer.

- e. If you are using **Remote Desktop Connection** from a Windows PC, return to the **Certificate** dialog box and choose **OK**. If you are using **Microsoft Remote Desktop** on a Mac, return to the **Verify Certificate** and choose **Continue**.
- f. If you are using **Remote Desktop Connection** from a Windows PC, choose **Yes** in the **Remote Desktop Connection** window to connect to your instance. If you are using **Microsoft Remote Desktop** on a Mac, log in to the instance as prompted, using the default **Administrator** account and the default administrator password that you recorded or copied previously.

Note

On a Mac, you may need to switch spaces to see the **Microsoft Remote Desktop** login screen. For more information on spaces, see <http://support.apple.com/kb/PH14155>.

Step 3: Clean Up Your Instance

After you've finished with the instance that you created for this tutorial, you should clean up by terminating the instance. If you want to do more with this instance before you clean up, see [Next Steps](#) (p. 24).

Important

Terminating an instance effectively deletes it; you can't reconnect to an instance after you've terminated it.

If you launched an instance that is not within the [AWS Free Tier](#), you'll stop incurring charges for that instance as soon as the instance status changes to `shutting down` or `terminated`. If you'd like to keep your instance for later, but not incur charges, you can stop the instance now and then start it again later. For more information, see [Stopping Instances](#).

To terminate your instance

1. In the navigation pane, choose **Instances**. In the list of instances, select the instance.
2. Choose **Actions**, then **Instance State**, and then choose **Terminate**.
3. Choose **Yes, Terminate** when prompted for confirmation.

Amazon EC2 shuts down and terminates your instance. After your instance is terminated, it remains visible on the console for a short while, and then the entry is deleted.

Next Steps

After you start your instance, you might want to try some of the following exercises:

- Learn how to remotely manage you EC2 instance using Run Command. For more information, see [Tutorial: Remotely Manage Your Amazon EC2 Instances](#) (p. 41) and [Remote Management](#) (p. 437).
- Configure a CloudWatch alarm to notify you if your usage exceeds the Free Tier. For more information, see [Create a Billing Alarm](#) in the *AWS Billing and Cost Management User Guide*.
- Add an EBS volume. For more information, see [Creating an Amazon EBS Volume](#) (p. 761) and [Attaching an Amazon EBS Volume to an Instance](#) (p. 766).
- Install the WAMP or WIMP stack. For more information, see [Tutorial: Installing a WAMP Server on an Amazon EC2 Instance Running Windows Server](#) (p. 32) and [Tutorial: Installing a WIMP Server on an Amazon EC2 Instance Running Windows Server](#) (p. 34).

Best Practices for Amazon EC2

This checklist is intended to help you get the maximum benefit from and satisfaction with Amazon EC2.

Security and Network

- Manage access to AWS resources and APIs using identity federation, IAM users, and IAM roles. Establish credential management policies and procedures for creating, distributing, rotating, and revoking AWS access credentials. For more information, see [IAM Best Practices](#) in the *IAM User Guide*.
- Implement the least permissive rules for your security group. For more information, see [Security Group Rules \(p. 607\)](#).
- Regularly patch, update, and secure the operating system and applications on your instance. For more information about updating Amazon Linux, see [Managing Software on Your Linux Instance](#) in the *Amazon EC2 User Guide for Linux Instances*. For more information about updating your Windows instance, see [Updating Your Windows Instance](#).
- Launch your instances into a VPC instead of EC2-Classic. Note that if you created your AWS account after 2013-12-04, we automatically launch your instances into a VPC. For more information about the benefits, see [Amazon EC2 and Amazon Virtual Private Cloud \(p. 665\)](#).

Storage

- Understand the implications of the root device type for data persistence, backup, and recovery. For more information, see [Storage for the Root Device \(p. 64\)](#).
- Use separate Amazon EBS volumes for the operating system versus your data. Ensure that the volume with your data persists after instance termination. For more information, see [Preserving Amazon EBS Volumes on Instance Termination \(p. 268\)](#).
- Use the instance store available for your instance to store temporary data. Remember that the data stored in instance store is deleted when you stop or terminate your instance. If you use instance store for database storage, ensure that you have a cluster with a replication factor that ensures fault tolerance.

Resource Management

- Use instance metadata and custom resource tags to track and identify your AWS resources. For more information, see [Instance Metadata and User Data \(p. 271\)](#) and [Tagging Your Amazon EC2 Resources \(p. 859\)](#).
- View your current limits for Amazon EC2. Plan to request any limit increases in advance of the time that you'll need them. For more information, see [Amazon EC2 Service Limits \(p. 869\)](#).

Backup and Recovery

- Regularly back up your instance using [Amazon EBS snapshots \(p. 788\)](#) or a backup tool.
- Deploy critical components of your application across multiple Availability Zones, and replicate your data appropriately.
- Design your applications to handle dynamic IP addressing when your instance restarts. For more information, see [Amazon EC2 Instance IP Addressing \(p. 693\)](#).
- Monitor and respond to events. For more information, see [Monitoring Amazon EC2 \(p. 563\)](#).
- Ensure that you are prepared to handle failover. For a basic solution, you can manually attach a network interface or Elastic IP address to a replacement instance. For more information, see [Elastic Network Interfaces \(p. 716\)](#). For an automated solution, you can use Auto Scaling. For more information, see the [Auto Scaling User Guide](#).
- Regularly test the process of recovering your instances and Amazon EBS volumes if they fail.

Tutorials for Amazon EC2 Instances Running Windows Server

The following tutorials show you how to perform common tasks using EC2 instances running Windows Server.

Tutorials

- [Tutorial: Deploying a WordPress Blog on Your Amazon EC2 Instance Running Windows Server \(p. 27\)](#)
- [Tutorial: Installing a WAMP Server on an Amazon EC2 Instance Running Windows Server \(p. 32\)](#)
- [Tutorial: Installing a WIMP Server on an Amazon EC2 Instance Running Windows Server \(p. 34\)](#)
- [Tutorial: Increase the Availability of Your Application on Amazon EC2 \(p. 38\)](#)
- [Tutorial: Remotely Manage Your Amazon EC2 Instances \(p. 41\)](#)
- [Tutorial: Setting Up a Windows HPC Cluster on Amazon EC2 \(p. 44\)](#)

Tutorial: Deploying a WordPress Blog on Your Amazon EC2 Instance Running Windows Server

This tutorial will help you install and deploy a WordPress blog on an Amazon EC2 instance running Windows Server.

If you'd prefer to host your WordPress blog on a Linux instance, see [Tutorial: Hosting a WordPress Blog with Amazon EC2](#) in the *Amazon EC2 User Guide for Linux Instances*.

Prerequisites

Before you get started, be sure that you do the following:

1. Launch an Amazon EC2 instance from a Windows Server AMI. For information, see [Getting Started with Amazon EC2 Windows Instances \(p. 20\)](#).

2. Use the AWS free usage tier (if eligible) to launch and use the free Windows *t2.micro* instance for 12 months. You can use the AWS free usage tier for launching new applications, testing existing applications, or simply gaining hands-on experience with AWS. For more information about eligibility and the highlights, see the [AWS Free Usage Tier](#) product page.

Important

If you've launched a regular instance and use it to deploy the WordPress website, you will incur the standard Amazon EC2 usage fees for the instance until you terminate it. For more information about Amazon EC2 usage rates, go to the [Amazon EC2 product page](#).

3. Ensure that the security group in which you're launching your instance has ports 80 (HTTP), 443 (HTTPS), and 3389 (RDP) open for inbound traffic. Ports 80 and 443 allow computers outside of the instance to connect with HTTP and HTTPS. If these ports are not open, the WordPress site can't be accessed from outside the instance. Port 3389 allows you to connect to the instance with Remote Desktop Protocol.
4. Connect to your instance.

Installing the Microsoft Web Platform Installer

You can use the Microsoft Web Platform Installer to install and configure WordPress on your server. This tool simplifies deployment of Web applications and Web sites to IIS servers. For more information, see [Microsoft Web Platform Installer](#).

1. Verify that you've met the conditions in [Prerequisites](#) (p. 27).
2. Disable Internet Explorer Enhanced Security Configuration so that you can download and install required software from the web.
 - a. In your Windows Server 2008 or 2012 instance, open Server Manager.
 - On Windows Server 2008 R2, under **Server Summary**, in the **Security Information** section, click **Configure IE ESC**.
 - On Windows Server 2012 R2, click **Local Server** in the left pane. In the **Properties** pane, locate **IE Enhanced Security Configuration**. Click **On**.
 - b. Under **Administrators**, click **Off**, and then click **OK**.
 - c. Close Server Manager.

Note

Make a note to re-enable Internet Explorer Enhanced Security Configuration when you have finished installing software from the web.

3. Download and install the latest version of the [Microsoft Web Platform Installer](#).

Installing WordPress

Now you'll use the Web Platform Installer to deploy WordPress on your server.

To install WordPress

1. [Download](#) and install Visual C++ Redistributable for Visual Studio 2012 Update 4 or later.
2. Open the **Web Platform Installer** and click **Applications**.
3. Select **WordPress**, click **Add**, and then click **Install**.
4. On the **Prerequisites** page, select **MySQL** for the database to use. Enter the desired administrator password for your MySQL database in the **Password** and **Re-type Password** boxes, and then click **Continue**.

Note

For more information about creating a secure password, see <http://www.pctools.com/guides/password/>. Do not reuse an existing password, and make sure to store this password in a safe place.

5. Click **I Accept** for the list of third-party application software, Microsoft products (including the IIS web server), and components. After the Web Platform Installer finishes installing the software, you are prompted to configure your new site.
6. On the **Configure** page, clear the default application name in the **'WordPress' application name:** box and leave it blank, then leave the default information in the other boxes and click **Continue**.
7. Click **Yes** to accept that the contents of the folder will be overwritten.

Configuring Security Keys

WordPress allows you to generate and enter unique authentication keys and salts for your site. These key and salt values provide a layer of encryption to the browser cookies that WordPress users store on their local machines. Basically, adding long, random values here makes your site more secure.

For more information about security keys, see http://codex.wordpress.org/Editing_wp-config.php#Security_Keys.

To configure security keys

1. Visit <https://api.wordpress.org/secret-key/1.1/salt/> to randomly generate a set of key values that you can copy and paste into the installation wizard. The following steps will show you how to modify these values in Notepad to work with a Windows installation.
2. Copy all of the text in that page to your clipboard. It should look similar to the example below.

Note

The values below are for example purposes only; do not use these values for your installation.

```
define('AUTH_KEY',          '3#U$${[RXN8:b^-L 0(WU_+ c+WFkI~c]o]-bHw+)/  
Aj[wTwSiZ<Qb[mghEXcRh-');  
define('SECURE_AUTH_KEY',  'Zsz._P=1/  
y.Lq)XjlkwS1y5NJ76E6EJ.AV0pCKZZB,*~*r ?6OP$eJT@;+(ndLg');  
define('LOGGED_IN_KEY',    'ju}qwre3V*+8f_zOWF?{L1GsQ]Ye@2Jh^,8x>)Y |;  
(^[Iw]Pi+LG#A4R?7N`YB3');  
define('NONCE_KEY',        'P(g62HeZxEes|LnI^i=H,[Xwk9I&[2s|: ?ON}VJM  
%?;v2v]v+;+^9eXUahg@::Cj');  
define('AUTH_SALT',        'C$DpB4Hj[JK: ?{ql`sRVa: { :7yShy(9A@5wg+`JJVb1fk  
%_-Bx*M4(qc[Qg%JT!h');  
define('SECURE_AUTH_SALT', 'd!uRu#)+q#{f$Z?Z9uFPG.${+S{n~1M&  
%~gL>U>NV<zpD-@2-Es7Q10-bp28EKv');  
define('LOGGED_IN_SALT',   ' ;j{00P*owZf)kVD+FVLn~>. |Y  
%Ug4#I^*LVd9QeZ^&XmK|e(76miC+&W&+^0P/');  
define('NONCE_SALT',       '-97r*v/cgxLmp?Zy4zUU4r99QQ_rGs2LTd%P;  
_e1tS)8_B/, .6[=UK<J_y9?JWG');
```

3. Open a Notepad window by clicking **Start, All Programs, Accessories**, and then **Notepad**.
4. Paste the copied text into the Notepad window.
5. Windows WordPress installations do not accept the dollar sign (\$) in key and salt values, so they need to be replaced with another character (such as s). In the Notepad window, click **Edit**, then click **Replace**.
6. In the **Find what** box, type \$.
7. In the **Replace with** box, type s.

- Click **Replace All** to replace all of the dollar signs with `s` characters.
- Close the **Replace** window.
- Paste the modified key and salt values from the Notepad window into their corresponding boxes in the installation wizard. For example, the `AUTH_KEY` value in the Notepad window should be pasted into the **Authentication Key** box in the wizard.

Do not include the single quotes or other text surrounding the values, just the actual value as in the example shown below.

The modified `AUTH_KEY` line from the Notepad window:

```
define('AUTH_KEY', '3#USS+[RXN8:b^-L 0(WU_+ c+WFkI~c]o]-bHw+)/Aj[wTwSiZ<Qb[mghEXcRh- ');
```

Paste this text into the **Authentication Key** box of the wizard:

```
3#USS+[RXN8:b^-L 0(WU_+ c+WFkI~c]o]-bHw+)/Aj[wTwSiZ<Qb[mghEXcRh-
```

- Click **Continue** and **Finish** to complete the Web Platform Installer wizard.

Configuring the Site Title and Administrator

When you complete the Web Platform Installer wizard, a browser window opens to your WordPress installation at `http://localhost/wp-admin/install.php`. On this page, you configure the title for your site and an administrative user to moderate your blog.

To complete the installation

- On the WordPress **Welcome** page, enter the following information and click **Install WordPress**.

Field	Value
Site Title	Enter a name for your WordPress site.
Username	Enter a name for your WordPress administrator. For security purposes you should choose a unique name for this user, since this will be more difficult to exploit than the default user name, <code>admin</code> .
Password	Enter a strong password, and then enter it again to confirm. Do not reuse an existing password, and make sure to store this password in a safe place.
Your E-mail	Enter the email address you want to use for notifications.
Privacy	Check to allow search engines to index your site.

- Click **Log In**.
- On the **Log In** page, enter your user name for **Username** and the site password you entered previously for **Password**.

Making Your WordPress Site Public

Now that you can see your WordPress blog on your local host, you can publish this website as the default site on your instance so that other people can see it. The next procedure walks you through the process of modifying your WordPress settings to point to the public DNS name of your instance instead of your local host.

To configure the default settings for your WordPress site

1. Open the WordPress dashboard by opening a browser on your instance and going to `http://localhost/wp-admin`. If prompted for your credentials, enter your user name for the **Username** and your site password for **Password**.
2. In the **Dashboard** pane, click **Settings**.
3. On the **General Settings** page, enter the following information and click **Save Changes**.
 - **WordPress address (URL)**—The public DNS address of your instance. For example, your URL may look something like `http://ec2-203-0-113-25.compute-1.amazonaws.com`.

You can get the public DNS for your instance using the Amazon EC2 console (select the instance and check the **Public DNS** column; if this column is hidden, click the **Show/Hide** icon and select **Public DNS**).
 - **Site address (URL)**—The same public DNS address of your instance that you set in **WordPress address (URL)**.
4. To see your new site, open a browser on a computer other than the instance hosting WordPress and type the public DNS address of your instance in the web address field. Your WordPress site appears.

Congratulations! You have just deployed a WordPress site on a Windows instance.

Next Steps

If you no longer need this instance, you can remove it to avoid incurring charges. For more information, see [Clean Up Your Instance \(p. 24\)](#).

If your WordPress blog becomes popular and you need more compute power or storage, consider the following steps:

- Expand the storage space on your instance. For more information, see [Expanding the Storage Space of an EBS Volume on Windows \(p. 783\)](#).
- Move your MySQL database to [Amazon RDS](#) to take advantage of the service's ability to scale automatically.
- Migrate to a larger instance type. For more information, see [Resizing Your Instance \(p. 147\)](#).
- Add additional instances. For more information, see [Tutorial: Increase the Availability of Your Application on Amazon EC2 \(p. 38\)](#).

For information about WordPress, see the WordPress Codex help documentation at <http://codex.wordpress.org/>. For more information about troubleshooting your installation, see http://codex.wordpress.org/Installing_WordPress#Common_Installation_Problems. For information about making your WordPress blog more secure, see http://codex.wordpress.org/Hardening_WordPress. For information about keeping your WordPress blog up-to-date, see http://codex.wordpress.org/Updating_WordPress.

Tutorial: Installing a WAMP Server on an Amazon EC2 Instance Running Windows Server

This tutorial shows you how to install an Apache web server with PHP and MySQL on an EC2 instance running Windows Server. This software configuration is sometimes called a WAMP server or WAMP stack (Windows, Apache, MySQL, PHP). For information about how to create a similar server on Linux, see [Tutorial: Installing a LAMP Web Server](#) in the *Amazon EC2 User Guide for Linux Instances*.

A WAMP stack is designed for easy installation to help developers get up and running quickly. It is *not* designed for production environments for the following reasons:

- The default configurations do not meet security requirements for most production environments.
- Upgrading and patching the different software components on a single production server would affect server availability.
- The WAMP one-click installers do not place files in standard locations, which can make it difficult to locate important configuration files.

You can, however, create a WAMP stack on an EC2 instance to prototype a web project in a controlled test environment. For example, you can host a static website or deploy a dynamic PHP application that reads and writes information to a database.

There are many third-party solutions that you can use to install a WAMP stack; this tutorial uses the Bitnami WAMP stack. For more information, see [Review: WAMP stacks for Web developers](#).

Prerequisites

Before you begin:

- Provision a Windows Server 2008 R2 or 2012 R2 base instance. You must configure the base instance with a public domain name system (DNS) name that is reachable from the Internet. For more information, see [Getting Started with Amazon EC2 Windows Instances \(p. 20\)](#). Optionally, you might be eligible to configure the base instance on the AWS free tier. The free tier is designed for users with new AWS accounts who want to gain experience with AWS. For more information about the free tier and eligibility requirements, see [AWS Free Tier](#).

Important

If you launch a non-free tier instance and use it to deploy your stack, you will incur the standard Amazon EC2 usage fees for the instance until you terminate it. For more information, see [Amazon EC2 Pricing](#).

- Verify that the security group for your instance has the following ports open:
 - 80 (HTTP inbound and outbound) - Port 80 allows computers outside of the instance to connect by using HTTP.
 - 443 (HTTPS inbound and outbound) - Port 443 allows computers outside of the instance to connect by using HTTPS.
 - 3389 (RDP inbound only) - Port 3389 allows you to connect to the instance with Remote Desktop Protocol (RDP). As a security best practice, restrict RDP access to a range of IP addresses in your organization.

For more information about these prerequisites, see [Setting Up with Amazon EC2 \(p. 13\)](#).

To install a WAMP server

1. Connect to your instance using Microsoft Remote Desktop. For more information, see [Connecting to Your Windows Instance \(p. 254\)](#).

2. Disable Internet Explorer Enhanced Security Configuration so that you can download and install required software from the web.
 - a. In your Windows Server 2008 or 2012 instance, open Server Manager.
 - On Windows Server 2008 R2, under **Server Summary**, in the **Security Information** section, click **Configure IE ESC**.
 - On Windows Server 2012 R2, click **Local Server** in the left pane. In the **Properties** pane, locate **IE Enhanced Security Configuration**. Click **On**.
 - b. Under **Administrators**, click **Off**, and then click **OK**.
 - c. Close Server Manager.

Note

Make a note to re-enable Internet Explorer Enhanced Security Configuration when you have finished installing software from the web.

3. Install software updates to ensure that the instance has the latest security updates and bug fixes.
 - a. **EC2Config - Download** and install the latest version of the EC2Config service. For more information about how to install this service, see [Installing the Latest Version of EC2Config \(p. 295\)](#).
 - b. **Windows Update** - Run Windows Update to ensure that the latest security and software updates are installed on the instance. In Control Panel, click **System and Security**. In the **Windows Update** section, click **Check for updates**.
4. Download and install the WAMP stack. For the purposes of this tutorial, we suggest that you download and install [this WAMP stack](#). You can, however, download and install [other Bitnami WAMP stacks](#). Regardless of which stack you install, the Bitnami site prompts you to either create a free Bitnami account or log in by using a social media account. After you log in, run the Bitnami setup wizard.
5. After setup completes, verify that the Apache web server is configured properly and running by browsing to a test page. Open a web browser on a different computer and enter either the public DNS address of the WAMP server or the public IP address. The public DNS address for your instance is listed on the Amazon EC2 console in the **Public DNS** column. If this column is hidden, click the **Show/Hide** icon and select **Public DNS**.

Important

If you do not see the Bitnami test page, use Windows Firewall with Advanced Security to create a custom rule that allows the HTTP protocol through port 80 and the HTTPS protocol through port 443. For more information, see [Windows Firewall with Advanced Security Overview](#) on Microsoft TechNet. Also verify that the security group you are using contains a rule to allow HTTP (port 80) connections. For information about adding an HTTP rule to your security group, see [Adding Rules to a Security Group](#).

6. Test your WAMP server by viewing a PHP file from the web. You must be logged onto the instance as an administrator to perform the following steps.
 - a. Create a file called phpinfo.php containing the code below and place this file in the Apache root directory. By default, the path is: C:\Bitnami\wampstack-version_number\apache2\htdocs.

```
<?php phpinfo(); ?>
```

- b. In a web browser, enter the URL of the file you just created. This URL is the public DNS address of your instance followed by a forward slash and the file name. For example:

`http://my.public.dns.amazonaws.com/phpinfo.php`
 - c. Verify that the PHP information page is displayed. If the page does not display, verify that you entered the correct public DNS address. Also verify that Windows folder options are configured to show known file extensions. By default, folder options hide known

file extensions. If you created the file in Notepad and saved it in the root directory your `phpinfo.php` file might incorrectly be saved as `phpinfo.php.txt`.

- d. As a security best practice, delete the `phpinfo.php` file when you finish testing the WAMP server.
7. Enhance MySQL security by disabling default features and by setting a root password. The **mysql_secure_installation** Perl script can perform these tasks for you. To run the script, you must install Perl.
- a. Download and install Perl from the [Perl Programming Language](#) website.
 - b. In the `C:\Bitnami\wampstack-version_number\mysql\bin` directory, double-click **mysql_secure_installation**.
 - c. When prompted, enter the MySQL root account password that you entered when you ran the Bitnami WAMP stack installer, and then press **Enter**.
 - d. Type **n** to skip changing the password.
 - e. Type **Y** to remove the anonymous user accounts.
 - f. Type **Y** to disable remote root login.
 - g. Type **Y** to remove the test database.
 - h. Type **Y** to reload the privilege tables and save your changes.

If you successfully completed the steps in this tutorial, then your WAMP server is functioning properly. To continue testing, you can add more content to the `C:\Bitnami\wampstack-version_number\apache2\htdocs` folder and view the content by using the public DNS address for your instance.

Important

As a best practice, stop the MySQL server if you do not plan to use it right away. You can restart the server when you need it again.

Tutorial: Installing a WIMP Server on an Amazon EC2 Instance Running Windows Server

This tutorial shows you how to install a Microsoft Internet Information Services (IIS) web server with PHP and MySQL on an EC2 instance running Windows Server. This software configuration is sometimes called a WIMP server or WIMP stack (Windows, IIS, MySQL, PHP).

A WIMP stack is designed for easy installation to help developers get up and running quickly. It is *not* designed for production environments for the following reasons:

- The default configurations do not meet security requirements for most production environments.
- Upgrading and patching the different software components on a single production server would affect server availability.
- The WAMP one-click installers do not place files in standard locations, which can make it difficult to locate important configuration files.

You can, however, create a WIMP stack on an EC2 instance to prototype a web project in a controlled test environment. For example, you can host a static website or deploy a dynamic PHP application that reads and writes information to a database.

Prerequisites

Before you begin:

- Provision a Windows Server 2008 R2 or 2012 R2 base instance. You must configure the base instance with a public domain name system (DNS) name that is reachable from the Internet. For more information, see [Getting Started with Amazon EC2 Windows Instances \(p. 20\)](#). Optionally, you might be eligible to configure the base instance using the AWS free tier. The free tier is designed for users with new AWS accounts who want to gain experience with AWS. For more information about the free tier and eligibility requirements, see [AWS Free Tier](#).

Important

If you launch a non-free tier instance and use it to deploy your stack, you will incur the standard Amazon EC2 usage fees for the instance until you terminate it. For more information, see [Amazon EC2 Pricing](#).

- Verify that the security group for your instance has the following ports open:
 - 80 (HTTP inbound and outbound) - Port 80 allows computers outside of the instance to connect by using HTTP.
 - 443 (HTTPS inbound and outbound) - Port 443 allows computers outside of the instance to connect by using HTTPS.
 - 3389 (RDP inbound only) - Port 3389 allows you to connect to the instance with Remote Desktop Protocol (RDP). As a security best practice, restrict RDP access to a range of IP addresses in your organization.

For more information about these prerequisites, see [Setting Up with Amazon EC2 \(p. 13\)](#).

- Read the best practices for installing PHP on the [Microsoft web platform](#).

To install a WIMP server

1. Connect to your instance using Microsoft Remote Desktop. For more information, see [Connecting to Your Windows Instance \(p. 254\)](#).
2. Disable Internet Explorer Enhanced Security Configuration so that you can download and install required software from the web.

Note

Make a note to re-enable Internet Explorer Enhanced Security Configuration when you have finished installing software from the web.

3. Install software updates to ensure that the instance has the latest security updates and bug fixes.
 - a. **EC2Config - Download** and install the latest version of the EC2Config service. For more information about how to install this service, see [Installing the Latest Version of EC2Config \(p. 295\)](#).
 - b. **Windows Update** - Run Windows Update to ensure that the latest security and software updates are installed on the instance. In Control Panel, click **System and Security**. In the **Windows Update** section, click **Check for updates**.

Install the IIS web server

IIS is a feature of Windows Server and is installed by using Server Manager. This section includes procedures for installing IIS on either Windows Server 2008 or 2012.

Install IIS on Windows Server 2012

1. In **Server Manager** click **Add roles and features**.
2. On the **Before you begin** page, click **Next**.
3. On the **Select installation type** page, select **Role-based or feature-based installation**, and then click **Next**.
4. On the **Select destination server** page, select your instance from the server pool, and then click **Next**.

5. On the **Select server roles** page, select **Web Server (IIS)**, click **Add features**, and then click **Next**.
6. On the **Select features** page, retain the default features and expand **.NET Framework 4.5 Features**, select **ASP.NET 4.5**, and then click **Next**.
7. On the **Web Server Role (IIS)** page, click **Next**.
8. On the **Select role services** page, retain the default services and select **Application Development**.
9. Expand **Application Development**, and then select the following features. When selecting these features, if you are prompted, click **Add features**:
 - a. .NET Extensibility 3.5
 - b. .NET Extensibility 4.5
 - c. Application Initialization
 - d. ASP.NET 3.5
 - e. ASP.NET 4.5
 - f. CGI
10. Click **Next**.
11. On the **Confirm installation selections** page, select **Restart the destination server automatically if required**. When prompted for confirmation, click **Yes**.
12. Click **Install**, and then after the installation is complete, click **Close**.
13. Run Windows update again.

Install IIS on Windows Server 2008

1. In **Server Manager**, click **Roles**.
2. Click **Add Roles**.
3. On the **Before You Begin** page, click **Next**.
4. On the **Select Server Roles** page, click **Web Server (IIS)**.
5. On the **Select Role Services** page under **Application Development**, click **ASP.NET**.
 - a. When prompted, click **Add Required Role Services**.
 - b. Click **CGI**.
 - c. Click **Next**.
6. On the **Confirm Installation Selections**, click **Install**.
7. Run Windows update again.

Verify that the web server is running

After setup completes, verify that the IIS web server is configured properly and running by going to the IIS welcome page. Open a web browser on a different computer and enter either the public DNS address of the WIMP server or the public IP address. The public DNS address for your instance is listed on the Amazon EC2 console in the **Public DNS** column. If this column is hidden, click the **Show/Hide** icon and select **Public DNS**.

Important

If you do not see the IIS welcome page, use Windows Firewall with Advanced Security to create a custom rule that allows the HTTP protocol through port 80 and the HTTPS protocol through port 443. For more information, see [Windows Firewall with Advanced Security Overview](#) on Microsoft TechNet. Also verify that the security group you are using contains a rule to allow HTTP (port 80) connections. For information about adding an HTTP rule to your security group, see [Adding Rules to a Security Group](#).

Install MySQL and PHP

You can download and install MySQL and PHP by using the Microsoft Web Platform Installer, as described in this section.

To install MySQL and PHP

1. In your Windows Server instance, download and install the latest version of the [Microsoft Web Platform Installer 5.0](#).
2. In the Microsoft Web Platform Installer click the **Products** tab.
3. Select **MySQL Windows 5.5** and click **Add**.
4. Select **PHP 5.6.0** and click **Add**.
5. Click **Install**.
6. On the **Prerequisites** page, enter a password for the MySQL default database administrator account, and then click **Continue**.
7. When the installation is complete, click **Finish**, and then click **Exit** to close the Web Platform Installer.

Test your WIMP server

Test your WIMP server by viewing a PHP file from the web. You must be logged onto the instance as an administrator to perform the following steps.

To test your WIMP server

1. Download and install the [Visual C++ Redistributable for Visual Studio 2012 Update 4 x86 package](#). Even if your server is a 64-bit server, you must install the x86 package.
2. Create a file called `phpinfo.php` that contains the following code and place this file in the IIS root directory. By default, the path is: `C:\inetpub\wwwroot`.

```
<?php phpinfo(); ?>
```

3. In a web browser, enter the URL of the file you just created. This URL is the public DNS address of your instance followed by a forward slash and the file name, as in the following example:

`http://my.public.dns.amazonaws.com/phpinfo.php`
4. Verify that the PHP information page is displayed. If the page does not display, verify that you entered the correct public DNS address. Also verify that Windows folder options are configured to show known file extensions. By default, folder options hide known file extensions. If you created the file in Notepad and saved it in the root directory your `phpinfo.php` file might incorrectly be saved as `phpinfo.php.txt`.
5. As a security best practice, delete the `phpinfo.php` file when you finish testing the WAMP server.
6. Enhance MySQL security by disabling default features and by setting a root password. The **mysql_secure_installation** Perl script can perform these tasks for you. To run the script, you must install Perl.
 - a. Download and install Perl from the [Perl Programming Language](#) website.
 - b. In the `C:\Program Files\MySQL\MySQL Server 5.5\bin` directory, double-click **mysql_secure_installation**.
 - c. When prompted, enter the current root password and press **Enter**.
 - d. Type **n** to skip changing the password.
 - e. Type **Y** to remove the anonymous user accounts.
 - f. Type **Y** to disable remote root login.

- g. Type **Y** to remove the test database.
- h. Type **Y** to reload the privilege tables and save your changes.

You should now have a fully functional WIMP web server. If you add content to the IIS document root at `C:\inetpub\wwwroot`, you can view that content at the public DNS address for your instance.

Important

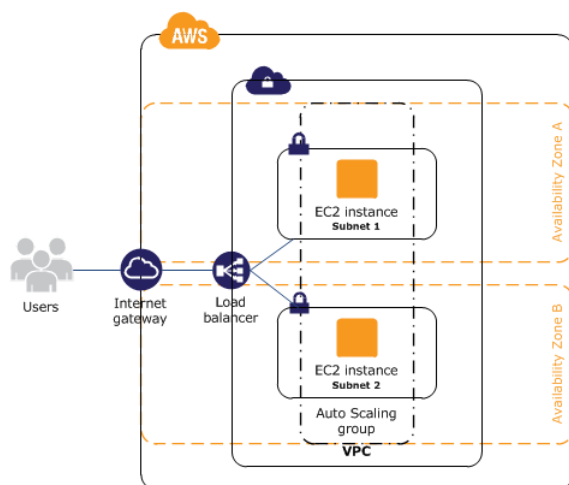
As a best practice, stop the MySQL server if you do not plan to use it right away. You can restart the server when you need it again.

Tutorial: Increase the Availability of Your Application on Amazon EC2

Suppose that you start out running your app or website on a single EC2 instance, and over time, traffic increases to the point that you require more than one instance to meet the demand. You can launch multiple EC2 instances from your AMI and then use Elastic Load Balancing to distribute incoming traffic for your application across these EC2 instances. This increases the availability of your application. Placing your instances in multiple Availability Zones also improves the fault tolerance in your application. If one Availability Zone experiences an outage, traffic is routed to the other Availability Zone.

You can use Auto Scaling to maintain a minimum number of running instances for your application at all times. Auto Scaling can detect when your instance or application is unhealthy and replace it automatically to maintain the availability of your application. You can also use Auto Scaling to scale your Amazon EC2 capacity up or down automatically based on demand, using criteria that you specify.

In this tutorial, we use Auto Scaling with Elastic Load Balancing to ensure that you maintain a specified number of healthy EC2 instances behind your load balancer. Note that these instances do not need public IP addresses, because traffic goes to the load balancer and is then routed to the instances. For more information, see [Auto Scaling](#) and [Elastic Load Balancing](#).



Contents

- [Prerequisites \(p. 39\)](#)
- [Scale and Load Balance Your Application \(p. 39\)](#)

- [Test Your Load Balancer \(p. 41\)](#)

Prerequisites

This tutorial assumes that you have already done the following:

1. If you don't have a default virtual private cloud (VPC), create a VPC with one public subnet in two or more Availability Zones. For more information, see [Create a Virtual Private Cloud \(VPC\) \(p. 17\)](#).
2. Launch an instance in the VPC.
3. Connect to the instance and customize it. For example, you can install software and applications, copy data, and attach additional EBS volumes. For information about setting up a web server on your instance, see [Tutorial: Installing a WAMP Server on an Amazon EC2 Instance Running Windows Server \(p. 32\)](#) or [Tutorial: Installing a WIMP Server on an Amazon EC2 Instance Running Windows Server \(p. 34\)](#).
4. Test your application on your instance to ensure that your instance is configured correctly.
5. Create a custom Amazon Machine Image (AMI) from your instance. For more information, see [Creating an Amazon EBS-Backed Windows AMI \(p. 77\)](#).
6. (Optional) Terminate the instance if you no longer need it.
7. Create an IAM role that grants your application the access to AWS that it needs. For more information, see [Creating an IAM Role Using the Console \(p. 660\)](#).

Scale and Load Balance Your Application

Use the following procedure to create a load balancer, create a launch configuration for your instances, create an Auto Scaling group with two or more instances, and associate the load balancer with the Auto Scaling group.

To scale and load-balance your application

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the navigation pane, under **LOAD BALANCING**, choose **Load Balancers**.
3. Choose **Create Load Balancer**.
4. Choose **Application Load Balancer**, and then choose **Continue**.
5. On the **Configure Load Balancer** page, do the following:
 - a. For **Name**, type a name for your load balancer. For example, `my-1b`.
 - b. For **Scheme**, keep the default value, **internet-facing**.
 - c. For **Listeners**, keep the default, which is a listener that accepts HTTP traffic on port 80.
 - d. For **VPC**, select the same VPC that you used for your instances.
 - e. For **Available subnets**, select at least two public subnets using their add icons. The subnets are moved under **Selected subnets**. Note that you can select only one subnet per Availability Zone. If you select a subnet from an Availability Zone where there is already a selected subnet, this subnet replaces the currently selected subnet for the Availability Zone.
 - f. Choose **Next: Configure Security Settings**.
6. For this tutorial, you are not using a secure listener. Choose **Next: Configure Security Groups**.
7. On the **Configure Security Groups** page, do the following:
 - a. Choose **Create a new security group**.
 - b. Type a name and description for the security group, or keep the default name and description. This new security group contains a rule that allows traffic to the port configured for the listener.

- c. Choose **Next: Configure Routing**.
8. On the **Configure Routing** page, do the following:
 - a. For **Target group**, keep the default, **New target group**.
 - b. For **Name**, type a name for the target group.
 - c. Keep **Protocol** as HTTP and **Port** as 80.
 - d. For **Health checks**, keep the default protocol and path.
 - e. Choose **Next: Register Targets**.
9. On the **Register Targets** page, choose **Next: Review** to continue to the next page, as we'll use Auto Scaling to add EC2 instances to the target group.
10. On the **Review** page, choose **Create**. After the load balancer is created, choose **Close**.
11. On the navigation pane, under **AUTO SCALING**, choose **Launch Configurations**.
 - If you are new to Auto Scaling, you see a welcome page. Choose **Create Auto Scaling group** to start the Create Auto Scaling Group wizard, and then choose **Create launch configuration**.
 - Otherwise, choose **Create launch configuration**.
12. On the **Choose AMI** page, select the **My AMIs** tab, and then select the AMI that you created in [Prerequisites \(p. 39\)](#).
13. On the **Choose Instance Type** page, select an instance type, and then choose **Next: Configure details**.
14. On the **Configure details** page, do the following:
 - a. For **Name**, type a name for your launch configuration (for example, `my-launch-config`).
 - b. For **IAM role**, select the IAM role that you created in [Prerequisites \(p. 39\)](#).
 - c. (Optional) If you need to run a startup script, expand **Advanced Details** and type the script in **User data**.
 - d. Choose **Skip to review**.
15. On the **Review** page, choose **Edit security groups**. You can select an existing security group or create a new one. This security group must allow HTTP traffic and health checks from the load balancer. If your instances will have public IP addresses, you can optionally allow RDP traffic if you need to connect to the instances. When you are finished, choose **Review**.
16. On the **Review** page, choose **Create launch configuration**.
17. When prompted, select an existing key pair, create a new key pair, or proceed without a key pair. Select the acknowledgment check box, and then choose **Create launch configuration**.
18. After the launch configuration is created, you must create an Auto Scaling group.
 - If you are new to Auto Scaling and you are using the Create Auto Scaling group wizard, you are taken to the next step automatically.
 - Otherwise, choose **Create an Auto Scaling group using this launch configuration**.
19. On the **Configure Auto Scaling group details** page, do the following:
 - a. For **Group name**, type a name for the Auto Scaling group. For example, `my-asg`.
 - b. For **Group size**, type the number of instances (for example, `2`). Note that we recommend that you maintain approximately the same number of instances in each Availability Zone.
 - c. Select your VPC from **Network** and your two public subnets from **Subnet**.
 - d. Under **Advanced Details**, select **Receive traffic from one or more load balancers**. Select your target group from **Target Groups**.
 - e. Choose **Next: Configure scaling policies**.
20. On the **Configure scaling policies** page, choose **Review**, as we will let Auto Scaling maintain the group at the specified size. Note that later on, you can manually scale this Auto Scaling group, configure the group to scale on a schedule, or configure the group to scale based on demand.

21. On the **Review** page, choose **Create Auto Scaling group**.
22. After the group is created, choose **Close**.

Test Your Load Balancer

When a client sends a request to your load balancer, the load balancer routes the request to one of its registered instances.

To test your load balancer

1. Verify that your instances are ready. From the **Auto Scaling Groups** page, select your Auto Scaling group, and then choose the **Instances** tab. Initially, your instances are in the `Pending` state. When their states are `InService`, they are ready for use.
2. Verify that your instances are registered with the load balancer. From the **Target Groups** page, select your target group, and then choose the **Targets** tab. If the state of your instances is `initial`, it's possible that they are still registering. When the state of your instances is `healthy`, they are ready for use. After your instances are ready, you can test your load balancer as follows.
3. From the **Load Balancers** page, select your load balancer.
4. On the **Description** tab, locate the DNS name. This name has the following form:

```
my-lb-xxxxxxxxxx.us-west-2.elb.amazonaws.com
```

5. In a web browser, paste the DNS name for the load balancer into the address bar and press Enter. You'll see your website displayed.

Tutorial: Remotely Manage Your Amazon EC2 Instances

This tutorial shows you how to remotely manage an Amazon EC2 instance using Amazon Elastic Compute Cloud (Amazon EC2) Run Command from your local machine. In this tutorial, you will learn how to do the following tasks:

- Launch a new instance that is configured for Run Command.
- Configure your user account for Run Command.
- Use Run Command to send a command from your local machine and retrieve a list of services running on the instance.

This tutorial includes procedures for executing commands using either the Amazon EC2 console or AWS Tools for Windows PowerShell.

Note

With Run Command, you can also manage your servers and virtual machines (VMs) in your on-premises environment or in an environment provided by other cloud providers. For more information, see [Setting Up Systems Manager in Hybrid Environments](#) (p. 397).

Launch a New Instance

Instances require an AWS Identity and Access Management (IAM) role that enables the instance to communicate with Amazon EC2 Simple Systems Manager (SSM). You must assign the IAM role when

you create the new instance. You can't assign a role to an instance that is already running. For existing instances, you must create an image of the instance, launch an instance from that image, and assign the IAM role as you launch the instance. For more information, see [Creating an Amazon EBS-Backed Windows AMI \(p. 77\)](#).

To create an instance that uses an SSM-supported role

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Select a supported [region](#).
3. Choose **Launch Instance** and select a Windows Server Amazon Machine Image (AMI).
4. Choose your instance type and then choose **Next: Configure Instance Details**.
5. In **Auto-assign Public IP**, choose **Enable**.
6. Beside **IAM role** choose **Create new IAM role**. The IAM console opens in a new tab.
 - a. Choose **Create New Role**.
 - b. In **Step 1: Set Role Name**, enter a name that identifies this role as a Run Command role.
 - c. In **Step 2: Select Role Type**, choose **Amazon EC2 Role for Simple Systems Manager**. The system skips **Step 3: Establish Trust** because this is a managed policy.
 - d. In **Step 4: Attach Policy**, choose **AmazonEC2RoleforSSM**.
 - e. Choose **Next Step**, and then choose **Create Role**.
 - f. Close the tab with the IAM console.
7. In the Amazon EC2 console, choose the **Refresh** button beside **Create New IAM role**.
8. From **IAM role**, choose the role you just created.
9. Complete the wizard to launch the new instance. Make a note of the instance ID. You will need to specify this ID later in this tutorial.

Grant Your User Account Access to SSM

Your user account must be configured to communicate with the SSM API. Use the following procedure to attach a managed IAM policy to your user account that grants you full access to SSM API actions.

To create the IAM policy for your user account

1. Open the Identity and Access Management (IAM) console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Policies**. (If this is your first time using IAM, choose **Get Started**, and then choose **Create Policy**.)
3. In the **Filter** field, type **AmazonSSMFullAccess** and press Enter.
4. Select the check box next to **AmazonSSMFullAccess** and then choose **Policy Actions, Attach**.
5. On the **Attach Policy** page, choose your user account and then choose **Attach Policy**.

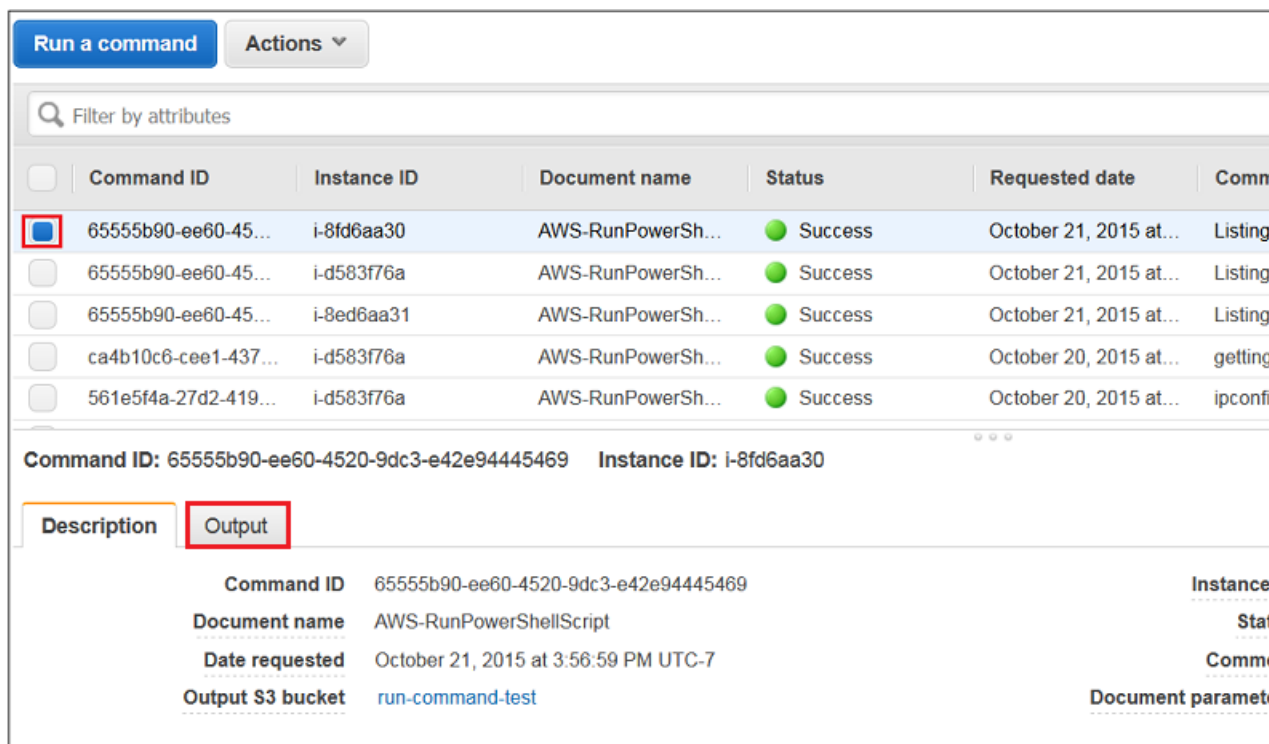
Send a Command Using the EC2 Console

Use the following procedure to list all services running on the instance by using Run Command from the Amazon EC2 console.

To execute a command using Run Command from the console

1. In the [Amazon EC2 console](#), in the navigation pane choose **Command History**, and then choose **Run a Command**.

2. In the **Command document** list, choose **AWS-RunPowerShellScript**.
3. Choose **Select instances**, and then choose the instance you just created. If you don't see the instance, verify that you are currently in the same region as the instance you created. Also verify that you configured the IAM role and trust policies as described earlier in this topic.
4. In the **Commands** field, type `Get-Service`. You can specify a **Working Directory** and **Execution Timeout**, if you want. The **Execution Timeout** is the number of seconds the EC2Config service will attempt to run the command before it is considered to have failed. We recommend entering a comment in the **Comments** field. A comment will help you identify the command in the list of pending commands and make it easier to view the output.
5. In the **Timeout (seconds)** field, type the number of seconds Run Command should attempt to reach instances before an instance is considered unreachable and the command execution fails.
6. Choose **Run** to execute the command. Run Command displays a status screen.
7. Choose **View results**.
8. Choose the command invocation for the command you just ran. Choose the **Output** tab, and then choose **View Output**.



Send a Command Using AWS Tools for Windows PowerShell

Use the following procedure to list all services running on the instance by using Run Command from AWS Tools for Windows PowerShell.

To execute a command

1. On your local computer, download the latest version of [AWS Tools for Windows PowerShell](#).
2. Open **AWS Tools for Windows PowerShell** on your local computer and execute the following command to specify your credentials.

```
Set-AWSCredentials -AccessKey key -SecretKey key
```

- Execute the following command to set the region for your PowerShell session. Specify the region where you created the instance in the previous procedure. This example uses the us-west-2 region.

```
Set-DefaultAWSRegion -Region us-west-2
```

- Execute the following command to retrieve the services running on the instance.

```
Send-SSMCommand -InstanceId 'Instance-ID' -DocumentName AWS-  
RunPowerShellScript -Comment 'listing services on the instance' -Parameter  
@{'commands'=@('Get-Service')}
```

The command returns a command ID, which you will use to view the results.

- The following command returns the output of the original Send-SSMCommand. The output is truncated after 2500 characters. To view the full list of services, specify an Amazon S3 bucket in the command using the -OutputS3BucketName *bucket_name* parameter.

```
Get-SSMCommandInvocation -CommandId Command-ID -Details $true | select -  
ExpandProperty CommandPlugins
```

For more examples of how to execute commands using Run Command with Tools for Windows PowerShell and the AWS Management Console, see [Amazon EC2 Run Command Walkthroughs \(p. 489\)](#). For more information about Run Command, see [Remote Management \(p. 437\)](#).

Tutorial: Setting Up a Windows HPC Cluster on Amazon EC2

You can launch a scalable Windows High Performance Computing (HPC) cluster using Amazon EC2 instances. A Windows HPC cluster requires an Active Directory domain controller, a DNS server, a head node, and one or more compute nodes.

To set up a Windows HPC cluster on Amazon EC2, complete the following tasks:

- [Step 1: Set Up Your Active Directory Domain Controller \(p. 45\)](#)
- [Step 2: Configure Your Head Node \(p. 47\)](#)
- [Step 3: Set Up the Compute Node \(p. 49\)](#)
- [Step 4: Scale Your HPC Compute Nodes \(Optional\) \(p. 50\)](#)

For more information about high performance computing, see [High Performance Computing \(HPC\) on AWS](#).

Prerequisites

- Install the AWS Command Line Interface tools, and set the region you'll be using as the default region. For more information, see [Installing the AWS Command Line Interface](#) in the *AWS Command Line Interface User Guide*.

- These procedures assume that you have a VPC in which to launch your instances. You can use your default VPC, or configure a nondefault VPC. For more information, see [What is Amazon VPC?](#) in the *Amazon VPC User Guide*.

Step 1: Set Up Your Active Directory Domain Controller

The Active Directory domain controller provides authentication and centralized resource management of the HPC environment and is required for the installation. To set up your Active Directory, complete these steps:

1. Create the security groups required for Active Directory.
2. Create the instance that serves as the domain controller for your HPC cluster.
3. Configure the domain controller for your HPC cluster.

Creating Security Groups for Active Directory

Use the AWS CLI to create security groups for the domain controller and domain members.

To create the required security groups for Active Directory

1. Create a security group in your VPC for the domain controller. In the output, take note of the security group ID.

```
aws ec2 create-security-group --vpc-id vpc-id --group-name "SG - Domain
  Controller" --description "Active Directory Domain Controller"

{
  "GroupId": "dc-security-group-id"
}
```

2. Create a security group in your VPC for the domain members. In the output, take note of the security group ID.

```
aws ec2 create-security-group --vpc-id vpc-id --group-name "SG - Domain
  Member" --description "Active Directory Domain Member"

{
  "GroupId": "dm-security-group-id"
}
```

3. Copy the contents of the first file in [IP Permissions for the Active Directory Security Groups \(p. 51\)](#) to a text editor. Replace the `dm-security-group-id` values with the ID of the your domain member security group. Save the file, using the file name `dc-sg-rules.json`.
4. Add the rules to the domain controller security group.

```
aws ec2 authorize-security-group-ingress --group-id dc-security-group-id
  --ip-permissions file://dc-sg-rules.json
```

Note

If the JSON file is located in a different directory from which you're working, you must include the path to the file after `file://`.

5. Copy the contents of the second file in [IP Permissions for the Active Directory Security Groups \(p. 51\)](#) to a text editor. Replace the `dc-security-group-id` values with the ID of the your domain controller security group. Save the file, using the file name `dm-sg-rules.json`.
6. Add the rules to the domain member security group.

```
aws ec2 authorize-security-group-ingress --group-id dm-security-group-id --ip-permissions file://dm-sg-rules.json
```

7. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
8. In the navigation pance, choose **Security Groups**. Verify that the following security groups appear in the list, and are populated with the required rules:
 - SG - Domain Controller
 - SG - Domain Member

Alternatively, manually set up the firewall to allow traffic on the required ports. For more information, go to [How to configure a firewall for domains and trusts](#) on the Microsoft website.

Creating the Domain Controller for your HPC cluster

Launch an instance that will serve as the domain controller for your HPC cluster.

To create a domain controller for your HPC cluster

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
Choose the same region in which you created your security groups.
2. On the console dashboard, choose **Launch Instance**.
3. On the **Choose an AMI** page, select an AMI for Windows Server, and choose **Select**.
4. On the next page of the wizard, select an instance type, then choose **Next: Configure Instance Details**.
5. On the **Configure Instance Details** page, select your VPC from **Network** and a subnet from **Subnet**. On the next page of the wizard, you can specify additional storage for your instance.
6. On the **Tag Instance** page, enter `Domain Controller` as the value for the Name tag and then choose **Next: Configure Security Group**.
7. On the **Configure Security Group** page, choose **Select an existing security group**, select `SG - Domain Controller` from the list of security groups, and then choose **Review and Launch**.
8. Choose **Launch**.

After you've launched your instance, associate an Elastic IP with the instance.

To associate an Elastic IP address with an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Choose **Allocate New Address**.
4. When prompted, choose **Yes, Allocate**, and then close the confirmation dialog box.

Note

If your account supports EC2-Classic, first choose **VPC** from the list.

5. Select the Elastic IP address you created, choose **Actions**, and then choose **Associate Address**.
6. In the **Instance** list, select the `Domain Controller` instance and then choose **Associate**.

Configuring the Domain Controller for Your HPC Cluster

Connect to the instance you created, and configure the server as a domain controller for the HPC cluster.

To configure your instance as a domain controller

1. Connect to your `Domain Controller` instance. For more information, see [Connecting to Your Windows Instance \(p. 254\)](#).
2. Open **Server Manager**, and add the **Active Directory Domain Services** role.
3. Promote the server to a domain controller using Server Manager or by running **DCPromo.exe**.
4. Create a new domain in a new forest.
5. Enter `hpc.local` as the fully qualified domain name (FQDN).
6. Select Forest Functional Level as **Windows Server 2008 R2**.
7. Ensure that the **DNS Server** option is selected, and then choose **Next**.
8. Select **Yes, the computer will use an IP address automatically assigned by a DHCP server (not recommended)**.
9. In the warning box, choose **Yes** to continue.
10. Complete the wizard and then select **Reboot on Completion**.
11. Connect to the instance as `hpc.local\administrator`.
12. Create a domain user `hpc.local\hpcuser`.

Step 2: Configure Your Head Node

An HPC client connects to the head node. The head node facilitates the scheduled jobs. You configure your head node by completing the following steps:

1. Create security groups for your HPC cluster.
2. Launch an instance for your head node.
3. Install the HPC Pack.
4. Configure your HPC cluster.

Creating Security Groups for Your HPC Cluster

Use the AWS CLI to create a security group for the HPC cluster.

To create the security group for your HPC cluster

1. Create a security group in your VPC for the HPC cluster. In the output, take note of the security group ID.

```
aws ec2 create-security-group --vpc-id vpc-id --group-name "SG - Windows  
HPC Cluster" --description "Windows HPC Server 2008 R2 Cluster Nodes"  
  
{  
  "GroupId": "hpc-security-group-id"  
}
```

2. Copy the contents of the JSON file in [IP Permissions for HPC Cluster Security Group \(p. 55\)](#) to a text editor. Replace the `hpc-security-group-id` value with the ID of your HPC security group. Save the file, using the file name `hpc-sg-rules.json`.
3. Add the rules to your HPC cluster security group.

```
aws ec2 authorize-security-group-ingress --group-id hpc-security-group-id --ip-permissions file://hpc-sg-rules.json
```

4. Open the Amazon EC2 console, select **Security Groups** from the navigation pane, and verify that the `SG - Windows HPC Cluster` security group appears in the list, and is populated with the required security group rules.

Alternatively, manually configure the firewall with the port requirements for HPC cluster members to communicate. For more information, see [Windows Firewall configuration](#) on the Microsoft website.

Launch an Instance for the HPC Head Node

Launch an instance and then configure it as a member of the `hpc.local` domain and with the necessary user accounts.

To configure an instance as your head node

1. Launch an instance and name it `HPC-Head`. When you launch the instance, select both of these security groups:
 - `SG - Windows HPC Cluster`
 - `SG - Domain Member`
2. Connect to the instance and get the existing DNS server address from **HPC-Head** using the following command:

```
C:\> IPConfig /all
```

3. Update the TCP/IPv4 properties of the `HPC-Head` NIC to include the Elastic IP address for the `Domain Controller` instance as the primary DNS, and then add the additional DNS IP address from the previous step.
4. Join the machine to the `hpc.local` domain using the credentials for `hpc.local\administrator` (the domain administrator account).
5. Add `hpc.local\hpcuser` as the local administrator. When prompted for credentials, use `hpc.local\administrator`, and then restart the instance.
6. Connect to **HPC-Head** as `hpc.local\hpcuser`.

Install the HPC Pack

To install the HPC Pack

1. Connect to your **HPC-Head** instance using the `hpc.local\hpcuser` account.
2. Using **Server Manager**, turn off Internet Explorer Enhanced Security Configuration (IE ESC) for Administrators.
 - a. In **Server Manager**, under **Security Information**, choose **Configure IE ESC**.
 - b. Turn off IE ESC for administrators.
3. Install the HPC Pack on **HPC-Head**.
 - a. Download the HPC Pack to `HPC-Head` from the [Microsoft Download Center](#). Choose the HPC Pack for the version of Windows Server on `HPC-Head`.
 - b. Extract the files to a folder, open the folder, and double-click **setup.exe**.
 - c. On the Installation page, select **Create a new HPC cluster by creating a head node**, and then choose **Next**.

- d. Accept the default settings to install all the databases on the Head Node, and then choose **Next**.
- e. Complete the wizard.

Configure Your HPC Cluster on the Head Node

To configure your HPC cluster on the head node

1. Start **HPC Cluster Manager**.
2. In the **Deployment To-Do List**, select **Configure your network**.
 - a. In the wizard, select the default option (5), and then choose **Next**.
 - b. Complete the wizard accepting default values on all screens, and choose how you want to update the server and participate in customer feedback.
 - c. Choose **Configure**.
3. Select **Provide Network Credentials**, then supply the `hpc.local\hpcuser` credentials.
4. Select **Configure the naming of new nodes**, and then choose **OK**.
5. Select **Create a node template**.
 - a. Select the **Compute node template**, and then choose **Next**.
 - b. Select **Without operating system**, and then continue with the defaults.
 - c. Choose **Create**.

Step 3: Set Up the Compute Node

Setting up the compute node involves the following steps:

1. Launch an instance for your compute node.
2. Install the HPC Pack on the instance.
3. Add the compute node to your cluster.

Launch an Instance for the HPC Compute Node

Configure your compute node by launching an instance, and then configuring the instance as a member of the `hpc.local` domain with the necessary user accounts.

To configure an instance for your compute node

1. Launch an instance and name it `HPC-Compute`. When you launch the instance, select the following security groups: **SG - Windows HPC Cluster** and **SG - Domain Member**.
2. Log in to the instance and get the existing DNS server address from **HPC-Compute** using the following command:

```
C:\> IPConfig /all
```

3. Update the TCP/IPv4 properties of the `HPC-Compute` NIC to include the Elastic IP address of the `Domain Controller` instance as the primary DNS. Then add the additional DNS IP address from the previous step.
4. Join the machine to the `hpc.local` domain using the credentials for `hpc.local\administrator` (the domain administrator account).

5. Add `hpc.local\hpcuser` as the local administrator. When prompted for credentials, use `hpc.local\administrator`, and then restart.
6. Connect to HPC-Compute as `hpc.local\hpcuser`.

Install the HPC Pack on the Compute Node

To install the HPC Pack on the compute node

1. Connect to your HPC-Compute instance using the `hpc.local\hpcuser` account.
2. Using **Server Manager**, turn off Internet Explorer Enhanced Security Configuration (IE ESC) for Administrators.
 - a. In **Server Manager**, under **Security Information**, choose **Configure IE ESC**.
 - b. Turn off IE ESC for administrators.
3. Install the HPC Pack on HPC-Compute.
 - a. Download the HPC Pack to HPC-Compute from the [Microsoft Download Center](#). Choose the HPC Pack for the version of Windows Server on HPC-Compute.
 - b. Extract the files to a folder, open the folder, and double-click **setup.exe**.
 - c. On the **Installation** page, select **Join an existing HPC cluster by creating a new compute node**, and then choose **Next**.
 - d. Specify the fully-qualified name of the HPC-Head instance, and then choose the defaults.
 - e. Complete the wizard.

Add the Compute Node to Your HPC Cluster

To complete your cluster configuration, from the head node, add the compute node to your cluster.

To add the compute node to your cluster

1. Connect to the HPC-Head instance as `hpc.local\hpcuser`.
2. Open **HPC Cluster Manager**.
3. Select **Node Management**.
4. If the compute node displays in the **Unapproved** bucket, right-click the node that is listed and select **Add Node**.
 - a. Select **Add compute nodes or broker nodes that have already been configured**.
 - b. Select the check box next to the node and choose **Add**.
5. Right-click the node and choose **Bring Online**.

Step 4: Scale Your HPC Compute Nodes (Optional)

To scale your compute nodes

1. Connect to the HPC-Compute instance as `hpc.local\hpcuser`.
2. Delete any files you downloaded locally from the HP Pack installation package. (You have already run setup and created these files on your image so they do not need to be cloned for an AMI.)
3. From `C:\Program Files\Amazon\Ec2ConfigService` open the file `sysprep2008.xml`.
4. At the bottom of `<settings pass="specialize">`, add the following section. Make sure to replace `hpc.local`, `password`, and `hpcuser` to match your environment.

```
<component name="Microsoft-Windows-UnattendedJoin"
  processorArchitecture="amd64" publicKeyToken="31bf3856ad364e35"
  language="neutral" versionScope="nonSxS" xmlns:wcm="http://
schemas.microsoft.com/WMIConfig/2002/State"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Identification>
    <UnsecureJoin>false</UnsecureJoin>
    <Credentials>
      <Domain>hpc.local</Domain>
      <Password>password</Password>
      <Username>hpcuser</Username>
    </Credentials>
    <JoinDomain>hpc.local</JoinDomain>
  </Identification>
</component>
```

5. Save `sysprep2008.xml`.
6. Choose **Start, All Programs, EC2ConfigService Settings**.
 - a. Choose the **General** tab, and clear the **Set Computer Name** check box.
 - b. Choose the **Bundle** tab, and then choose **Run Sysprep and Shutdown Now**.
7. Open the Amazon EC2 console.
8. In the navigation pane, choose **Instances**.
9. Wait for the instance status to show **stopped**.
10. Select the instance, choose **Actions, Image, Create Image**.
11. Specify an image name and image description, and then choose **Create Image** to create an AMI from the instance.
12. Start the original `HPC-Compute` instance that was shut down.
13. Connect to the head node using the `hpc.local\hpcuser` account.
14. From **HPC Cluster Manager**, delete the old node that now appears in an error state.
15. In the Amazon EC2 console, in the navigation pane, choose **AMIs**.
16. Use the AMI you created to add additional nodes to the cluster.

You can launch additional compute nodes from the AMI that you created. These nodes are automatically joined to the domain, but you must add them to the cluster as already configured nodes in **HPC Cluster Manager** using the head node and then bring them online.

Running the Lizard Performance Measurement Application

You can optionally run the Lizard application, which measures the computational performance and efficiency that can be achieved by your HPC cluster. Go to <http://www.microsoft.com/download/en/details.aspx?id=8433>, download the `lizard_x64.msi` installer, and run the installer directly on your head node as `hpc.local\hpcuser`.

IP Permissions for the Active Directory Security Groups

The following JSON contains the IP permissions structures for the security groups for your Active Directory environment: one group for Active Directory domain controllers and one for Active Directory domain member servers.

For more information about these security group rules, go to the following Microsoft article: <http://support.microsoft.com/kb/179442>.

1. Security group rules for the domain controller security group

The following rules apply to the domain controller security group. Replace the `dm-security-group-id` value with the ID of your domain member security group. Replace the `cidr_block` value with the CIDR block of your local network.

```
[
  {
    "IpProtocol": "UDP",
    "FromPort": 123,
    "ToPort": 123,
    "UserIdGroupPairs": [
      {
        "GroupId": "dm-security-group-id"
      }
    ]
  },
  {
    "IpProtocol": "TCP",
    "FromPort": 135,
    "ToPort": 135,
    "UserIdGroupPairs": [
      {
        "GroupId": "dm-security-group-id"
      }
    ]
  },
  {
    "IpProtocol": "UDP",
    "FromPort": 138,
    "ToPort": 138,
    "UserIdGroupPairs": [
      {
        "GroupId": "dm-security-group-id"
      }
    ]
  },
  {
    "IpProtocol": "TCP",
    "FromPort": 49152,
    "ToPort": 65535,
    "UserIdGroupPairs": [
      {
        "GroupId": "dm-security-group-id"
      }
    ]
  },
  {
    "IpProtocol": "TCP",
    "FromPort": 389,
    "ToPort": 389,
    "UserIdGroupPairs": [
      {
        "GroupId": "dm-security-group-id"
      }
    ]
  }
]
```

```
    },  
    {  
      "IpProtocol": "UDP",  
      "FromPort": 389,  
      "ToPort": 389,  
      "UserIdGroupPairs": [  
        {  
          "GroupId": "dm-security-group-id"  
        }  
      ]  
    },  
    {  
      "IpProtocol": "TCP",  
      "FromPort": 636,  
      "ToPort": 636,  
      "UserIdGroupPairs": [  
        {  
          "GroupId": "dm-security-group-id"  
        }  
      ]  
    },  
    {  
      "IpProtocol": "TCP",  
      "FromPort": 3268,  
      "ToPort": 3269,  
      "UserIdGroupPairs": [  
        {  
          "GroupId": "dm-security-group-id"  
        }  
      ]  
    },  
    {  
      "IpProtocol": "TCP",  
      "FromPort": 53,  
      "ToPort": 53,  
      "UserIdGroupPairs": [  
        {  
          "GroupId": "dm-security-group-id"  
        }  
      ]  
    },  
    {  
      "IpProtocol": "UDP",  
      "FromPort": 53,  
      "ToPort": 53,  
      "UserIdGroupPairs": [  
        {  
          "GroupId": "dm-security-group-id"  
        }  
      ]  
    },  
    {  
      "IpProtocol": "UDP",  
      "FromPort": 88,  
      "ToPort": 88,  
      "UserIdGroupPairs": [  
        {  
          "GroupId": "dm-security-group-id"  
        }  
      ]  
    }  
  ]  
}
```

```
]
},
{
  "IpProtocol": "TCP",
  "FromPort": 88,
  "ToPort": 88,
  "UserIdGroupPairs": [
    {
      "GroupId": "dm-security-group-id"
    }
  ]
},
{
  "IpProtocol": "TCP",
  "FromPort": 445,
  "ToPort": 445,
  "UserIdGroupPairs": [
    {
      "GroupId": "dm-security-group-id"
    }
  ]
},
{
  "IpProtocol": "UDP",
  "FromPort": 445,
  "ToPort": 445,
  "UserIdGroupPairs": [
    {
      "GroupId": "dm-security-group-id"
    }
  ]
},
{
  "IpProtocol": "ICMP",
  "FromPort": -1,
  "ToPort": -1,
  "UserIdGroupPairs": [
    {
      "GroupId": "dm-security-group-id"
    }
  ]
},
{
  "IpProtocol": "UDP",
  "FromPort": 53,
  "ToPort": 53,
  "IpRanges": [
    {
      "CidrIp": "cidr_block"
    }
  ]
},
{
  "IpProtocol": "TCP",
  "FromPort": 3389,
  "ToPort": 3389,
  "IpRanges": [
    {
      "CidrIp": "cidr_block"
    }
  ]
}
```

```
    }  
  ]  
}
```

2. Security group rules for the domain member security group

The following rules apply to the domain member security group. Replace the `dc-security-group-id` value with the ID of your domain controller security group.

```
[  
  {  
    "IpProtocol": "TCP",  
    "FromPort": 49152,  
    "ToPort": 65535,  
    "UserIdGroupPairs": [  
      {  
        "GroupId": "dc-security-group-id"  
      }  
    ]  
  },  
  {  
    "IpProtocol": "UDP",  
    "FromPort": 49152,  
    "ToPort": 65535,  
    "UserIdGroupPairs": [  
      {  
        "GroupId": "dc-security-group-id"  
      }  
    ]  
  },  
  {  
    "IpProtocol": "TCP",  
    "FromPort": 53,  
    "ToPort": 53,  
    "UserIdGroupPairs": [  
      {  
        "GroupId": "dc-security-group-id"  
      }  
    ]  
  },  
  {  
    "IpProtocol": "UDP",  
    "FromPort": 53,  
    "ToPort": 53,  
    "UserIdGroupPairs": [  
      {  
        "GroupId": "dc-security-group-id"  
      }  
    ]  
  }  
]
```

IP Permissions for HPC Cluster Security Group

The following JSON file contains the IP permissions to create a security group for your HPC cluster nodes. Replace the `hpc-security-group-id` value with the ID of the SG - Windows HPC

Cluster security group. The last rule enables you to connect to your instance via RDP. Replace the `cidr_block` value with the CIDR block for your network.

For more information about these security group rules, go to the following Microsoft article: http://technet.microsoft.com/en-us/library/ff919486.aspx#BKMK_Firewall

```
[
  {
    "IpProtocol": "TCP",
    "FromPort": 80,
    "ToPort": 80,
    "UserIdGroupPairs": [
      {
        "GroupId": "hpc-security-group-id"
      }
    ]
  },
  {
    "IpProtocol": "TCP",
    "FromPort": 443,
    "ToPort": 443,
    "UserIdGroupPairs": [
      {
        "GroupId": "hpc-security-group-id"
      }
    ]
  },
  {
    "IpProtocol": "TCP",
    "FromPort": 1856,
    "ToPort": 1856,
    "UserIdGroupPairs": [
      {
        "GroupId": "hpc-security-group-id"
      }
    ]
  },
  {
    "IpProtocol": "TCP",
    "FromPort": 5800,
    "ToPort": 5801,
    "UserIdGroupPairs": [
      {
        "GroupId": "hpc-security-group-id"
      }
    ]
  },
  {
    "IpProtocol": "TCP",
    "FromPort": 5801,
    "ToPort": 5801,
    "UserIdGroupPairs": [
      {
        "GroupId": "hpc-security-group-id"
      }
    ]
  },
  {
    "IpProtocol": "TCP",
```

```
"FromPort": 5969,  
"ToPort": 5969,  
"UserIdGroupPairs": [  
  {  
    "GroupId": "hpc-security-group-id"  
  }  
]  
},  
{  
  "IpProtocol": "TCP",  
  "FromPort": 5970,  
  "ToPort": 5970,  
  "UserIdGroupPairs": [  
    {  
      "GroupId": "hpc-security-group-id"  
    }  
  ]  
},  
{  
  "IpProtocol": "TCP",  
  "FromPort": 5974,  
  "ToPort": 5974,  
  "UserIdGroupPairs": [  
    {  
      "GroupId": "hpc-security-group-id"  
    }  
  ]  
},  
{  
  "IpProtocol": "TCP",  
  "FromPort": 5999,  
  "ToPort": 5999,  
  "UserIdGroupPairs": [  
    {  
      "GroupId": "hpc-security-group-id"  
    }  
  ]  
},  
{  
  "IpProtocol": "TCP",  
  "FromPort": 6729,  
  "ToPort": 6730,  
  "UserIdGroupPairs": [  
    {  
      "GroupId": "hpc-security-group-id"  
    }  
  ]  
},  
{  
  "IpProtocol": "TCP",  
  "FromPort": 7997,  
  "ToPort": 7997,  
  "UserIdGroupPairs": [  
    {  
      "GroupId": "hpc-security-group-id"  
    }  
  ]  
},  
{
```



```
"IpProtocol": "TCP",
"FromPort": 8677,
"ToPort": 8677,
"UserIdGroupPairs": [
  {
    "GroupId": "hpc-security-group-id"
  }
]
},
{
  "IpProtocol": "TCP",
  "FromPort": 9087,
  "ToPort": 9087,
  "UserIdGroupPairs": [
    {
      "GroupId": "hpc-security-group-id"
    }
  ]
},
{
  "IpProtocol": "TCP",
  "FromPort": 9090,
  "ToPort": 9092,
  "UserIdGroupPairs": [
    {
      "GroupId": "hpc-security-group-id"
    }
  ]
},
{
  "IpProtocol": "TCP",
  "FromPort": 9100,
  "ToPort": 9163,
  "UserIdGroupPairs": [
    {
      "GroupId": "hpc-security-group-id"
    }
  ]
},
{
  "IpProtocol": "TCP",
  "FromPort": 9200,
  "ToPort": 9263,
  "UserIdGroupPairs": [
    {
      "GroupId": "hpc-security-group-id"
    }
  ]
},
{
  "IpProtocol": "TCP",
  "FromPort": 9794,
  "ToPort": 9794,
  "UserIdGroupPairs": [
    {
      "GroupId": "hpc-security-group-id"
    }
  ]
},
}
```

```
{
  "IpProtocol": "TCP",
  "FromPort": 9892,
  "ToPort": 9893,
  "UserIdGroupPairs": [
    {
      "GroupId": "hpc-security-group-id"
    }
  ]
},
{
  "IpProtocol": "UDP",
  "FromPort": 9893,
  "ToPort": 9893,
  "UserIdGroupPairs": [
    {
      "GroupId": "hpc-security-group-id"
    }
  ]
},
{
  "IpProtocol": "TCP",
  "FromPort": 6498,
  "ToPort": 6498,
  "UserIdGroupPairs": [
    {
      "GroupId": "hpc-security-group-id"
    }
  ]
},
{
  "IpProtocol": "TCP",
  "FromPort": 7998,
  "ToPort": 7998,
  "UserIdGroupPairs": [
    {
      "GroupId": "hpc-security-group-id"
    }
  ]
},
{
  "IpProtocol": "TCP",
  "FromPort": 8050,
  "ToPort": 8050,
  "UserIdGroupPairs": [
    {
      "GroupId": "hpc-security-group-id"
    }
  ]
},
{
  "IpProtocol": "TCP",
  "FromPort": 5051,
  "ToPort": 5051,
  "UserIdGroupPairs": [
    {
      "GroupId": "hpc-security-group-id"
    }
  ]
}
]
```

```
    },  
    {  
      "IpProtocol": "TCP",  
      "FromPort": 3389,  
      "ToPort": 3389,  
      "IpRanges": [  
        {  
          "CidrIp": "cidr_block"  
        }  
      ]  
    }  
  ]
```

Amazon Machine Images (AMI)

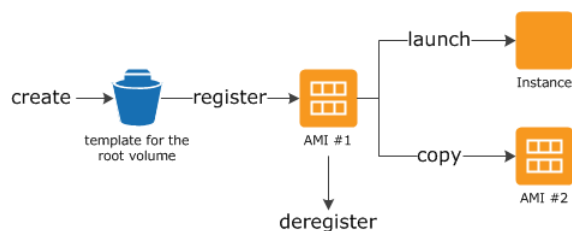
An Amazon Machine Image (AMI) provides the information required to launch an instance, which is a virtual server in the cloud. You specify an AMI when you launch an instance, and you can launch as many instances from the AMI as you need. You can also launch instances from as many different AMIs as you need.

An AMI includes the following:

- A template for the root volume for the instance (for example, an operating system, an application server, and applications)
- Launch permissions that control which AWS accounts can use the AMI to launch instances
- A block device mapping that specifies the volumes to attach to the instance when it's launched

Using an AMI

The following diagram summarizes the AMI lifecycle. After you create and register an AMI, you can use it to launch new instances. (You can also launch instances from an AMI if the AMI owner grants you launch permissions.) You can copy an AMI to the same region or to different regions. When you are finished launching instance from an AMI, you can deregister the AMI.



You can search for an AMI that meets the criteria for your instance. You can search for AMIs provided by AWS or AMIs provided by the community. For more information, see [AMI Types \(p. 63\)](#) and [Finding a Windows AMI \(p. 67\)](#).

When you are connected to an instance, you can use it just like you use any other server. For information about launching, connecting, and using your instance, see [Amazon EC2 Instances \(p. 117\)](#).

Creating Your Own AMI

You can customize the instance that you launch from a public AMI and then save that configuration as a custom AMI for your own use. Instances that you launch from your AMI use all the customizations that you've made.

The root storage device of the instance determines the process you follow to create an AMI. The root volume of an instance is either an Amazon EBS volume or an instance store volume. For information, see [Root Device Volume](#) (p. 8).

To create an Amazon EBS-backed AMI, see [Creating an Amazon EBS-Backed Windows AMI](#) (p. 77). To create an instance store-backed AMI, see [Creating an Instance Store-Backed Windows AMI](#) (p. 80).

To help categorize and manage your AMIs, you can assign custom *tags* to them. For more information, see [Tagging Your Amazon EC2 Resources](#) (p. 859).

Buying, Sharing, and Selling AMIs

After you create an AMI, you can keep it private so that only you can use it, or you can share it with a specified list of AWS accounts. You can also make your custom AMI public so that the community can use it. Building a safe, secure, usable AMI for public consumption is a fairly straightforward process, if you follow a few simple guidelines. For information about how to create and use shared AMIs, see [Shared AMIs](#) (p. 69).

You can purchase an AMIs from a third party, including AMIs that come with service contracts from organizations such as Red Hat. You can also create an AMI and sell it to other Amazon EC2 users. For more information about buying or selling AMIs, see [Paid AMIs](#) (p. 74).

Deregistering Your AMI

You can deregister an AMI when you have finished with it. After you deregister an AMI, you can't use it to launch new instances. For more information, see [Deregistering Your AMI](#) (p. 92).

AWS Windows AMIs

AWS provides a set of publicly available AMIs that contain software configurations specific to the Windows platform. Using these AMIs, you can quickly start building and deploying your applications using Amazon EC2. First choose the AMI that meets your specific requirements, and then launch an instance using that AMI. You retrieve the password for the administrator account and then log in to the instance using Remote Desktop Connection, just as you would with any other Windows server. The name of the administrator account depends on the language of the operating system. For example, for English, it's Administrator, for French it's Administrateur, and for Portuguese it's Administrador. For more information, see [Localized Names for Administrator Account in Windows](#) in the Microsoft TechNet Wiki.

Selecting an Initial Windows AMI

To view the Windows AMIs provided by AWS using the Amazon EC2 console, click this link to filter the list of public AMIs: [Windows AMIs](#). If you launch an instance using the Amazon EC2 console, the first

page of the wizard includes a **Quick Start** tab that lists some of the most popular AMIs provided by AWS, including AMIs that are eligible for the free tier.

AWS currently provides AMIs based on the following versions of Windows:

- Microsoft Windows Server 2016 (64-bit)
- Windows Server 2012 R2 (64-bit)
- Windows Server 2012 (64-bit)
- Windows Server 2008 R2 (64-bit)
- Windows Server 2008 (64-bit)
- Windows Server 2008 (32-bit)
- Windows Server 2003 R2 (64-bit)
- Windows Server 2003 R2 (32-bit)

Some of these AMIs also include an edition of Microsoft SQL Server (SQL Enterprise Edition, SQL Server Standard, SQL Server Express, or SQL Server Web). Launching an instance from an AWS Windows AMI with Microsoft SQL Server enables you to run the instance as a database server. Alternatively, you can launch an instance from any Windows AMI and then install the database software that you need on the instance. To view Windows Server AMIs with SQL Server, see [Windows AMIs](#) on the AWS Marketplace.

Some AMIs come with Internet Information Services (IIS) and ASP.NET already configured, to help you get started quickly. Alternatively, you can launch an instance from any Windows AMI and then install IIS and ASP.NET. For step-by-step directions, see [Configure Your EC2 Instance](#) in *Getting Started with AWS: Hosting a .NET Web App*.

In addition to the public AMIs provided by AWS, AMIs published by the AWS developer community are available for your use. We highly recommend that you use only those Windows AMIs that AWS or other reputable sources provide. To learn how to find a list of Windows AMIs approved by Amazon, see [Finding a Windows AMI](#) (p. 67).

You can also create an AMI from your own Windows computer. For more information, see the [VM Import/Export User Guide](#).

Keeping Your AMIs Up-to-Date

AWS provides updated, fully-patched Windows AMIs within five business days of Microsoft's patch Tuesday (the second Tuesday of each month). For more information, see [Details About AWS Windows AMI Versions](#) (p. 98).

At their initial launch, your Windows instances contain all the latest security updates. We recommend that you run the Windows Update service as a first step after you launch a Windows, and before you create an AMI. After you launch an instance or create an AMI, you are responsible for keeping them up-to-date. You can use the Windows Update service, or the Automatic Updates tool available on your instance to deploy Microsoft updates to your instance. You must also keep any other software that you deploy to your instance up-to-date using whatever mechanism is appropriate for that software. After you update your Windows instance, you can create an AMI that replaces any previous AMIs that you created. For more information, see [Updating Your Windows Instance](#) (p. 95).

AMI Types

You can select an AMI to use based on the following characteristics:

- Region (see [Regions and Availability Zones \(p. 6\)](#))
- Operating system
- Architecture (32-bit or 64-bit)
- [Launch Permissions \(p. 64\)](#)
- [Storage for the Root Device \(p. 64\)](#)

Launch Permissions

The owner of an AMI determines its availability by specifying launch permissions. Launch permissions fall into the following categories.

Launch Permission	Description
public	The owner grants launch permissions to all AWS accounts.
explicit	The owner grants launch permissions to specific AWS accounts.
implicit	The owner has implicit launch permissions for an AMI.

Amazon and the Amazon EC2 community provide a large selection of public AMIs. For more information, see [Shared AMIs \(p. 69\)](#). Developers can charge for their AMIs. For more information, see [Paid AMIs \(p. 74\)](#).

Storage for the Root Device

All AMIs are categorized as either *backed by Amazon EBS* or *backed by instance store*. The former means that the root device for an instance launched from the AMI is an Amazon EBS volume created from an Amazon EBS snapshot. The latter means that the root device for an instance launched from the AMI is an instance store volume created from a template stored in Amazon S3. For more information, see [Root Device Volume \(p. 8\)](#).

This section summarizes the important differences between the two types of AMIs. The following table provides a quick summary of these differences.

Characteristic	Amazon EBS-Backed	Amazon Instance Store-Backed
Boot time	Usually less than 1 minute	Usually less than 5 minutes
Size limit	16 TiB	10 GiB
Root device volume	Amazon EBS volume	Instance store volume
Data persistence	By default, the root volume is deleted when the instance terminates.* Data on any other Amazon EBS volumes persists after instance termination by default. Data on any instance store volumes persists only during the life of the instance.	Data on any instance store volumes persists only during the life of the instance. Data on any Amazon EBS volumes persists after instance termination by default.
Upgrading	The instance type, kernel, RAM disk, and user data can be changed while the instance is stopped.	Instance attributes are fixed for the life of an instance.

Characteristic	Amazon EBS-Backed	Amazon Instance Store-Backed
Charges	You're charged for instance usage, Amazon EBS volume usage, and storing your AMI as an Amazon EBS snapshot.	You're charged for instance usage and storing your AMI in Amazon S3.
AMI creation/bundling	Uses a single command/call	Requires installation and use of AMI tools
Stopped state	Can be placed in stopped state where instance is not running, but the root volume is persisted in Amazon EBS	Cannot be in stopped state; instances are running or terminated

* By default, Amazon EBS-backed instance root volumes have the `DeleteOnTermination` flag set to `true`. For information about how to change this flag so that the volume persists after termination, see [Root Device Volume \(p. 8\)](#).

Determining the Root Device Type of Your AMI

To determine the root device type of an AMI using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **AMIs**, and select the AMI.
3. Check the value of **Root Device Type** in the **Details** tab as follows:
 - If the value is `ebs`, this is an Amazon EBS-backed AMI.
 - If the value is `instance store`, this is an instance store-backed AMI.

To determine the root device type of an AMI using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- `describe-images` (AWS CLI)
- `Get-EC2Image` (AWS Tools for Windows PowerShell)

Size Limit

Amazon EC2 instance store-backed AMIs are limited to 10 GiB storage for the root device, whereas Amazon EBS-backed AMIs are limited to 1 TiB. Many Windows AMIs come close to the 10 GiB limit, so you'll find that Windows AMIs are often backed by an Amazon EBS volume.

Note

All Windows Server 2008, Windows Server 2008 R2, and Windows Server 2012 AMIs are backed by an Amazon EBS volume by default because of their larger size.

Stopped State

You can stop an Amazon EBS-backed instance, but not an Amazon EC2 instance store-backed instance. Stopping causes the instance to stop running (its status goes from `running` to `stopping` to `stopped`). A stopped instance persists in Amazon EBS, which allows it to be restarted. Stopping is different from terminating; you can't restart a terminated instance. Because Amazon EC2 instance

store-backed AMIs can't be stopped, they're either running or terminated. For more information about what happens and what you can do while an instance is stopped, see [Stop and Start Your Instance \(p. 259\)](#).

Default Data Storage and Persistence

Instances that use an instance store volume for the root device automatically have instance store available (the root volume contains the root partition and you can store additional data). Any data on an instance store volume is deleted when the instance fails or terminates (except for data on the root device). You can add persistent storage to your instance by attaching one or more Amazon EBS volumes.

Instances that use Amazon EBS for the root device automatically have an Amazon EBS volume attached. The volume appears in your list of volumes like any other. The instances don't use any available instance store volumes by default. You can add instance storage or additional Amazon EBS volumes using a block device mapping. For more information, see [Block Device Mapping \(p. 833\)](#). For information about what happens to the instance store volumes when you stop an instance, see [Stop and Start Your Instance \(p. 259\)](#).

Boot Times

Amazon EBS-backed AMIs launch faster than Amazon EC2 instance store-backed AMIs. When you launch an Amazon EC2 instance store-backed AMI, all the parts have to be retrieved from Amazon S3 before the instance is available. With an Amazon EBS-backed AMI, only the parts required to boot the instance need to be retrieved from the snapshot before the instance is available. However, the performance of an instance that uses an Amazon EBS volume for its root device is slower for a short time while the remaining parts are retrieved from the snapshot and loaded into the volume. When you stop and restart the instance, it launches quickly, because the state is stored in an Amazon EBS volume.

AMI Creation

To create Windows AMIs backed by instance store, there's an API action that creates an AMI and another API action that registers the AMI.

AMI creation is much easier for AMIs backed by Amazon EBS. The `CreateImage` API action creates your Amazon EBS-backed AMI and registers it. There's also a button in the AWS Management Console that lets you create an AMI from a running instance. For more information, see [Creating an Amazon EBS-Backed Windows AMI \(p. 77\)](#).

How You're Charged

With AMIs backed by instance store, you're charged for AMI storage and instance usage. With AMIs backed by Amazon EBS, you're charged for volume storage and usage in addition to the AMI and instance usage charges.

With Amazon EC2 instance store-backed AMIs, each time you customize an AMI and create a new one, all of the parts are stored in Amazon S3 for each AMI. So, the storage footprint for each customized AMI is the full size of the AMI. For Amazon EBS-backed AMIs, each time you customize an AMI and create a new one, only the changes are stored. So the storage footprint for subsequent AMIs you customize after the first is much smaller, resulting in lower AMI storage charges.

When an Amazon EBS-backed instance is stopped, you're not charged for instance usage; however, you're still charged for volume storage. We charge a full instance hour for every transition from a stopped state to a running state, even if you transition the instance multiple times within a single hour. For example, let's say the hourly instance charge for your instance is \$0.10. If you were to run that

instance for one hour without stopping it, you would be charged \$0.10. If you stopped and restarted that instance twice during that hour, you would be charged \$0.30 for that hour of usage (the initial \$0.10, plus 2 x \$0.10 for each restart).

Finding a Windows AMI

Before you can launch an instance, you must select an AMI to use. As you select an AMI, consider the following requirements you might have for the instances that you'll launch:

- The region
- The operating system (see [AWS Windows AMIs \(p. 62\)](#))
- The architecture: 32-bit (`i386`) or 64-bit (`x86_64`)
- The root device type: Amazon EBS or instance store
- The provider: Amazon Web Services, Oracle, IBM, Microsoft, or the community

If you need to find a Linux AMI, see [Finding a Linux AMI](#) in the *Amazon EC2 User Guide for Linux Instances*.

Contents

- [Finding a Windows AMI Using the Amazon EC2 Console \(p. 67\)](#)
- [Finding an AMI Using the AWS CLI \(p. 68\)](#)
- [Finding an AMI Using the AWS Tools for Windows PowerShell \(p. 68\)](#)
- [Finding a Windows Server 2003 AMI \(p. 68\)](#)

Finding a Windows AMI Using the Amazon EC2 Console

You can find Windows AMIs using the Amazon EC2 console. You can search through all available AMIs using the **Images** page, or select from commonly used AMIs on the **Quick Launch** tab when you use the console to launch an instance.

To find a Windows AMI using the Images page

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select a region. You can select any region that's available to you, regardless of your location. This is the region in which you'll launch your instance.
3. In the navigation pane, choose **AMIs**.
4. (Optional) Use the **Filter** options to scope the list of displayed AMIs to see only the AMIs that interest you. For example, to list all Windows AMIs provided by AWS, select **Public images**. Choose the Search bar and select **Owner** from the menu, then select **Amazon images**. Choose the Search bar again to select **Platform** and then the operating system from the list provided.
5. (Optional) Choose the **Show/Hide Columns** icon to select which image attributes to display, such as the root device type. Alternatively, you can select an AMI from the list and view its properties in the **Details** tab.
6. To launch an instance from this AMI, select it and then choose **Launch**. For more information about launching an instance using the console, see [Launching Your Instance from an AMI \(p. 246\)](#). If you're not ready to launch the instance now, write down the AMI ID (`ami-xxxxxxx`) for later.

To find a Windows AMI when you launch an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the console dashboard, choose **Launch Instance**.
3. On the **Choose an Amazon Machine Image (AMI)** page, on the **Quick Start** tab, select from one of the commonly used AMIs in the list. If you don't see the AMI that you need, select the **AWS Marketplace** or **Community AMIs** tab to find additional AMIs.

Finding an AMI Using the AWS CLI

You can use command line parameters to list only the types of AMIs that interest you. For example, you can use the [describe-images](#) command as follows to find public AMIs owned by you or Amazon.

```
C:\> aws ec2 describe-images --owners self amazon
```

Add the following filter to the previous command to display only Windows AMIs:

```
--filters "Name=platform,Values=windows"
```

After locating an AMI that meets your needs, write down its ID (ami-xxxxxxx). You can use this AMI to launch instances. For more information, see [Launching an Instance Using the AWS CLI](#) in the *AWS Command Line Interface User Guide*.

Finding an AMI Using the AWS Tools for Windows PowerShell

You can use command-line parameters to list only the types of AMIs that interest you. For more information, see [Find an AMI Using Windows PowerShell](#) in the *AWS Tools for Windows PowerShell User Guide*.

After locating an AMI that meets your needs, write down its ID (ami-xxxxxxx). You can use this AMI to launch instances. For more information, see [Launch an Instance Using Windows PowerShell](#) in the *AWS Tools for Windows PowerShell User Guide*.

Finding a Windows Server 2003 AMI

As of July 14, 2015, Microsoft [no longer supports](#) Windows Server 2003. If your business or organization is currently running Windows Server 2003 EC2 instances, we recommend that you upgrade those instances to Windows Server 2008. For more information, see [Upgrading a Windows Server EC2 Instance to a Newer Version of Windows Server](#).

To find a Windows Server 2003 AMI

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **AMIs**.
3. Choose **Owned by me**, and then choose **Public images**.
4. In the **Search** field, add the following filters.
 - a. Owner : Amazon images
 - b. AMI Name : Windows_Server-2003

Note

The **Search** field is case sensitive.

Shared AMIs

A *shared AMI* is an AMI that a developer created and made available for other developers to use. One of the easiest ways to get started with Amazon EC2 is to use a shared AMI that has the components you need and then add custom content. You can also create your own AMIs and share them with others.

You use a shared AMI at your own risk. Amazon can't vouch for the integrity or security of AMIs shared by other Amazon EC2 users. Therefore, you should treat shared AMIs as you would any foreign code that you might consider deploying in your own data center and perform the appropriate due diligence. We recommend that you get an AMI from a trusted source. If you have questions or observations about a shared AMI, use the [AWS forums](#).

Amazon's public images have an aliased owner, which appears as `amazon` in the account field. This enables you to find AMIs from Amazon easily. Other users can't alias their AMIs.

For information about creating an AMI, see [Creating an Instance Store-Backed Windows AMI](#) or [Creating an Amazon EBS-Backed Windows AMI](#). For more information about building, delivering, and maintaining your applications on the AWS Marketplace, see the [AWS Marketplace User Guide](#) and [AWS Marketplace Seller Guide](#).

Contents

- [Finding Shared AMIs](#) (p. 69)
- [Making an AMI Public](#) (p. 70)
- [Sharing an AMI with Specific AWS Accounts](#) (p. 71)
- [Using Bookmarks](#) (p. 73)
- [Guidelines for Shared Windows AMIs](#) (p. 73)

Finding Shared AMIs

You can use the Amazon EC2 console or the command line to find shared AMIs.

Finding a Shared AMI (Console)

To find a shared private AMI using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **AMIs**.
3. In the first filter, choose **Private images**. All AMIs that have been shared with you are listed. To granulate your search, choose the Search bar and use the filter options provided in the menu.

To find a shared public AMI using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **AMIs**.
3. In the first filter, choose **Public images**. To granulate your search, choose the Search bar and use the filter options provided in the menu.

4. Use filters to list only the types of AMIs that interest you. For example, choose **Owner :** and then choose **Amazon images** to display only Amazon's public images.

Finding a Shared AMI (Command Line)

To find a shared public AMI using the command line tools

Use the [describe-images](#) command (AWS CLI) to list AMIs. You can scope the list to the types of AMIs that interest you, as shown in the following examples.

The following command lists all public AMIs using the `--executable-users` option. This list includes any public AMIs that you own.

```
C:\> aws ec2 describe-images --executable-users all
```

The following command lists the AMIs for which you have explicit launch permissions. This list excludes any such AMIs that you own.

```
C:\> aws ec2 describe-images --executable-users self
```

The following command lists the AMIs owned by Amazon. Amazon's public AMIs have an aliased owner, which appears as `amazon` in the `account` field. This enables you to find AMIs from Amazon easily. Other users can't alias their AMIs.

```
C:\> aws ec2 describe-images --owners amazon
```

The following command lists the AMIs owned by the specified AWS account.

```
C:\> aws ec2 describe-images --owners 123456789012
```

To reduce the number of displayed AMIs, use a filter to list only the types of AMIs that interest you. For example, use the following filter to display only EBS-backed AMIs.

```
--filters "Name=root-device-type,Values=ebs"
```

Alternatively, you can use the following AWS Tools for Windows PowerShell command: [Get-EC2Image](#).

Making an AMI Public

Amazon EC2 enables you to share your AMIs with other AWS accounts. You can allow all AWS accounts to launch the AMI (make the AMI public), or only allow a few specific accounts to launch the AMI (see [Sharing an AMI with Specific AWS Accounts \(p. 71\)](#)). You are not billed when your AMI is launched by other AWS accounts; only the accounts launching the AMI are billed.

AMIs are a regional resource. Therefore, sharing an AMI makes it available in that region. To make an AMI available in a different region, copy the AMI to the region and then share it. For more information, see [Copying an AMI \(p. 87\)](#).

Note

If an AMI has a product code, you can't make it public. You must share the AMI with only specific AWS accounts.

Sharing an AMI with all AWS Accounts (Console)

After you make an AMI public, it is available in **Community AMIs** when you launch an instance in the same region using the console. Note that it can take a short while for an AMI to appear in **Community AMIs** after you make it public. It can also take a short while for an AMI to be removed from **Community AMIs** after you make it private again.

To share a public AMI using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **AMIs**.
3. Select your AMI from the list, and then choose **Actions, Modify Image Permissions**.
4. Choose **Public** and choose **Save**.

Sharing an AMI with all AWS Accounts (Command Line)

Each AMI has a `launchPermission` property that controls which AWS accounts, besides the owner's, are allowed to use that AMI to launch instances. By modifying the `launchPermission` property of an AMI, you can make the AMI public (which grants launch permissions to all AWS accounts) or share it with only the AWS accounts that you specify.

You can add or remove account IDs from the list of accounts that have launch permissions for an AMI. To make the AMI public, specify the `all` group. You can specify both public and explicit launch permissions.

To make an AMI public

Use the `modify-image-attribute` command (AWS CLI) as follows to add the `all` group to the `launchPermission` list for the specified AMI.

```
C:\> aws ec2 modify-image-attribute --image-id ami-12345678 --launch-  
permission "{\"Add\": [{\"Group\": \"all\"}]}"
```

To verify the launch permissions of the AMI, use the following `describe-image-attribute` command.

```
C:\> aws ec2 describe-image-attribute --image-id ami-12345678 --attribute  
launchPermission
```

(Optional) To make the AMI private again, remove the `all` group from its launch permissions. Note that the owner of the AMI always has launch permissions and is therefore unaffected by this command.

```
C:\> aws ec2 modify-image-attribute --image-id ami-12345678 --launch-  
permission "{\"Remove\": [{\"Group\": \"all\"}]}"
```

Alternatively, you can use the following AWS Tools for Windows PowerShell commands: [Edit-EC2ImageAttribute](#) and [Get-EC2ImageAttribute](#).

Sharing an AMI with Specific AWS Accounts

You can share an AMI with specific AWS accounts without making the AMI public. All you need are the AWS account IDs.

AMIs are a regional resource. Therefore, sharing an AMI makes it available in that region. To make an AMI available in a different region, copy the AMI to the region and then share it. For more information, see [Copying an AMI \(p. 87\)](#).

Sharing an AMI (Console)

To grant explicit launch permissions using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **AMIs**.
3. Select your AMI in the list, and then choose **Actions, Modify Image Permissions**.
4. Specify the AWS account number of the user with whom you want to share the AMI in the **AWS Account Number** field, then choose **Add Permission**.

To share this AMI with multiple users, repeat the above step until you have added all the required users.

5. To allow create volume permissions for snapshots, select **Add "create volume" permissions to the following associated snapshots when creating permissions**.

Note

You do not need to share the Amazon EBS snapshots that an AMI references in order to share the AMI. Only the AMI itself needs to be shared; the system automatically provides the instance access to the referenced Amazon EBS snapshots for the launch.

6. Choose **Save** when you are done.

Sharing an AMI (Command Line)

Use the `modify-image-attribute` command (AWS CLI) to share an AMI as shown in the following examples.

To grant explicit launch permissions

The following command grants launch permissions for the specified AMI to the specified AWS account.

```
C:\> aws ec2 modify-image-attribute --image-id ami-12345678 --launch-  
permission "{\"Add\": [{\"UserId\": \"123456789012\"}]}"
```

The following command grants create volume permission for a snapshot.

```
C:\> aws ec2 modify-snapshot-attribute --snapshot-id snap-1234567890abcdef0  
--attribute createVolumePermission --operation-type add --user-  
ids 123456789012
```

To remove launch permissions for an account

The following command removes launch permissions for the specified AMI from the specified AWS account:

```
C:\> aws ec2 modify-image-attribute --image-id ami-12345678 --launch-  
permission "{\"Remove\": [{\"UserId\": \"123456789012\"}]}"
```

The following command removes create volume permission for a snapshot.

```
C:\> aws ec2 modify-snapshot-attribute --snapshot-id snap-1234567890abcdef0  
--attribute createVolumePermission --operation-type remove --user-  
ids 123456789012
```

To remove all launch permissions

The following command removes all public and explicit launch permissions from the specified AMI. Note that the owner of the AMI always has launch permissions and is therefore unaffected by this command.

```
C:\> aws ec2 reset-image-attribute --image-id ami-12345678 --attribute  
launchPermission
```

Alternatively, you can use the following AWS Tools for Windows PowerShell command: [Edit-EC2ImageAttribute](#).

Using Bookmarks

If you have created a public AMI, or shared an AMI with another AWS user, you can create a *bookmark* that allows a user to access your AMI and launch an instance in their own account immediately. This is an easy way to share AMI references, so users don't have to spend time finding your AMI in order to use it.

Note that your AMI must be public, or you must have shared it with the user to whom you want to send the bookmark.

To create a bookmark for your AMI

1. Type a URL with the following information, where *<region>* is the region in which your AMI resides, and *<ami_id>* is the ID of the AMI:

```
https://console.aws.amazon.com/ec2/v2/home?  
region=<region>#LaunchInstanceWizard:ami=<ami_id>
```

For example, this URL launches an instance from the ami-12345678 AMI in the us-east-1 region:

```
https://console.aws.amazon.com/ec2/v2/home?region=us-  
east-1#LaunchInstanceWizard:ami=ami-12345678
```

2. Distribute the link to users who want to use your AMI.
3. To use a bookmark, choose the link or copy and paste it into your browser. The launch wizard opens, with the AMI already selected.

Guidelines for Shared Windows AMIs

Use the following guidelines to reduce the attack surface and improve the reliability of the AMIs you create.

Note

No list of security guidelines can be exhaustive. Build your shared AMIs carefully and take time to consider where you might expose sensitive data.

- Develop a repeatable process for building, updating, and republishing AMIs.
- Build AMIs using the most up-to-date operating systems, packages, and software.
- [Download](#) and install the latest version of the EC2Config service. For more information about installing this service, see [Installing the Latest Version of EC2Config \(p. 295\)](#).
- Verify that Ec2SetPassword, Ec2WindowsActivate and Ec2HandleUserData are enabled.
- Verify that no guest accounts or Remote Desktop user accounts are present.
- Disable or remove unnecessary services and programs to reduce the attack surface of your AMI.

- Remove instance credentials, such as your key pair, from the AMI (if you saved them on the AMI). Store the credentials in a safe location.
- Ensure that the administrator password and passwords on any other accounts are set to an appropriate value for sharing. These passwords are available for anyone who launches your shared AMI.
- Test your AMI before you share it.

Paid AMIs

A *paid AMI* is an AMI that you can purchase from a developer.

Amazon EC2 integrates with AWS Marketplace, enabling developers to charge other Amazon EC2 users for the use of their AMIs or to provide support for instances.

The AWS Marketplace is an online store where you can buy software that runs on AWS; including AMIs that you can use to launch your EC2 instance. The AWS Marketplace AMIs are organized into categories, such as Developer Tools, to enable you to find products to suit your requirements. For more information about AWS Marketplace, see the [AWS Marketplace](#) site.

Launching an instance from a paid AMI is the same as launching an instance from any other AMI. No additional parameters are required. The instance is charged according to the rates set by the owner of the AMI, as well as the standard usage fees for the related web services; for example, the hourly rate for running a m1.small instance type in Amazon EC2. Additional taxes may also apply. The owner of the paid AMI can confirm whether a specific instance was launched using that paid AMI.

Important

Amazon DevPay is no longer accepting new sellers or products. AWS Marketplace is now the single, unified e-commerce platform for selling software and services through AWS. For information about how to deploy and sell software from AWS Marketplace, see [Selling on AWS Marketplace](#). AWS Marketplace supports AMIs backed by Amazon EBS.

Topics

- [Selling Your AMI \(p. 74\)](#)
- [Finding a Paid AMI \(p. 74\)](#)
- [Purchase a Paid AMI \(p. 75\)](#)
- [Getting the Product Code for Your Instance \(p. 76\)](#)
- [Using Paid Support \(p. 76\)](#)
- [Bills for Paid and Supported AMIs \(p. 77\)](#)
- [Managing Your AWS Marketplace Subscriptions \(p. 77\)](#)

Selling Your AMI

You can sell your AMI using AWS Marketplace. AWS Marketplace offers an organized shopping experience. Additionally, AWS Marketplace also supports AWS features such as Amazon EBS-backed AMIs, Reserved Instances, and Spot instances.

For information about how to sell your AMI on AWS Marketplace, see [Selling on AWS Marketplace](#).

Finding a Paid AMI

There are several ways that you can find AMIs that are available for you to purchase. For example, you can use [AWS Marketplace](#), the Amazon EC2 console, or the command line. Alternatively, a developer might let you know about a paid AMI themselves.

Finding a Paid AMI Using the Console

To find a paid AMI using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **AMIs**.
3. Select **Public images** from the first **Filter** list. Click the Search bar and select **Product Code**, then **Marketplace**. Click the Search bar again, select **Platform** and then choose the operating system from the list.

Finding a Paid AMI Using AWS Marketplace

To find a paid AMI using AWS Marketplace

1. Open [AWS Marketplace](#).
2. Enter the name of the operating system in the search box, and click **Go**.
3. To scope the results further, use one of the categories or filters.
4. Each product is labeled with its product type: either `AMI` or `Software as a Service`.

Finding a Paid AMI Using the Command Line

You can find a paid AMI using the [describe-images](#) command (AWS CLI) as follows.

```
C:\> aws ec2 describe-images --owners aws-marketplace
```

This command returns numerous details that describe each AMI, including the product code for a paid AMI. The output from `describe-images` includes an entry for the product code like the following:

```
"ProductCodes": [  
  {  
    "ProductCodeId": "product_code",  
    "ProductCodeType": "marketplace"  
  }  
],
```

Alternatively, you can use the following AWS Tools for Windows PowerShell command: [Get-EC2Image](#).

Purchase a Paid AMI

You must sign up for (purchase) a paid AMI before you can launch an instance using the AMI.

Typically a seller of a paid AMI presents you with information about the AMI, including its price and a link where you can buy it. When you click the link, you're first asked to log into AWS, and then you can purchase the AMI.

Purchasing a Paid AMI Using the Console

You can purchase a paid AMI by using the Amazon EC2 launch wizard. For more information, see [Launching an AWS Marketplace Instance](#) (p. 251).

Subscribing to a Product Using AWS Marketplace

To use the AWS Marketplace, you must have an AWS account. To launch instances from AWS Marketplace products, you must be signed up to use the Amazon EC2 service, and you must be subscribed to the product from which to launch the instance. There are two ways to subscribe to products in the AWS Marketplace:

- **AWS Marketplace website:** You can launch preconfigured software quickly with the 1-Click deployment feature.
- **Amazon EC2 launch wizard:** You can search for an AMI and launch an instance directly from the wizard. For more information, see [Launching an AWS Marketplace Instance \(p. 251\)](#).

Purchasing a Paid AMI From a Developer

The developer of a paid AMI can enable you to purchase a paid AMI that isn't listed in AWS Marketplace. The developer provides you with a link that enables you to purchase the product through Amazon. You can sign in with your Amazon.com credentials and select a credit card that's stored in your Amazon.com account to use when purchasing the AMI.

Getting the Product Code for Your Instance

You can retrieve the AWS Marketplace product code for your instance using its instance metadata. For more information about retrieving metadata, see [Instance Metadata and User Data \(p. 271\)](#).

To retrieve a product code, use the following query:

```
C:\> GET http://169.254.169.254/latest/meta-data/product-codes
```

If the instance has a product code, Amazon EC2 returns it. For example:

```
774F4FF8
```

Using Paid Support

Amazon EC2 also enables developers to offer support for software (or derived AMIs). Developers can create support products that you can sign up to use. During sign-up for the support product, the developer gives you a product code, which you must then associate with your own AMI. This enables the developer to confirm that your instance is eligible for support. It also ensures that when you run instances of the product, you are charged according to the terms for the product specified by the developer.

Important

You can't use a support product with Reserved Instances. You always pay the price that's specified by the seller of the support product.

To associate a product code with your AMI, use one of the following commands, where *ami_id* is the ID of the AMI and *product_code* is the product code:

- [modify-image-attribute](#) (AWS CLI)

```
C:\> aws ec2 modify-image-attribute --image-id ami_id --product-codes  
"product_code"
```

- [Edit-EC2ImageAttribute](#) (AWS Tools for Windows PowerShell)

```
C:\> Edit-EC2ImageAttribute -ImageId ami_id -ProductCode product_code
```

After you set the product code attribute, it cannot be changed or removed.

Bills for Paid and Supported AMIs

At the end of each month, you receive an email with the amount your credit card has been charged for using any paid or supported AMIs during the month. This bill is separate from your regular Amazon EC2 bill. For more information, see [Paying For AWS Marketplace Products](#).

Managing Your AWS Marketplace Subscriptions

On the AWS Marketplace website, you can check your subscription details, view the vendor's usage instructions, manage your subscriptions, and more.

To check your subscription details

1. Log in to the [AWS Marketplace](#).
2. Click **Your Account**.
3. Click **Manage Your Software Subscriptions**.
4. All your current subscriptions are listed. Click **Usage Instructions** to view specific instructions for using the product, for example, a user name for connecting to your running instance.

To cancel an AWS Marketplace subscription

1. Ensure that you have terminated any instances running from the subscription.
 - a. Open the Amazon EC2 console.
 - b. In the navigation pane, click **Instances**.
 - c. Select the instance, click **Actions**, select **Instance State**, and select **Terminate**. When prompted, click **Yes, Terminate**.
2. Log in to the [AWS Marketplace](#), and click **Your Account**, then **Manage Your Software Subscriptions**.
3. Click **Cancel subscription**. You are prompted to confirm your cancellation.

Note

After you've canceled your subscription, you are no longer able to launch any instances from that AMI. To use that AMI again, you need to resubscribe to it, either on the AWS Marketplace website, or through the launch wizard in the Amazon EC2 console.

Creating an Amazon EBS-Backed Windows AMI

To create an Amazon EBS-backed Windows AMI, you launch and customize a Windows instance, then you create the AMI.

If you need to create an Amazon EBS-backed Linux AMI, see [Creating an Amazon EBS-Backed Linux AMI](#) in the *Amazon EC2 User Guide for Linux Instances*.

The AMI-creation process is different for instance-store-backed AMIs. For more information about the differences between Amazon EBS-backed and instance-store-backed instances, and how to determine the root device type for your instance, see [Root Device Volume \(p. 8\)](#). If you need to create an instance store-backed Windows AMI, see [Creating an Instance Store-Backed Windows AMI \(p. 80\)](#).

Overview of Creating Amazon EBS-Backed AMIs

First, launch an instance from an AMI that's similar to the AMI that you'd like to create. You can connect to your instance and customize it. When the instance is set up the way you want it, ensure data integrity by stopping the instance before you create an AMI and then create the image. When you create an Amazon EBS-backed AMI, we automatically register it for you.

During the AMI-creation process, Amazon EC2 creates snapshots of your instance's root volume and any other EBS volumes attached to your instance. If any volumes attached to the instance are encrypted, the new AMI only launches successfully on instances that support Amazon EBS encryption. For more information, see [Amazon EBS Encryption \(p. 799\)](#).

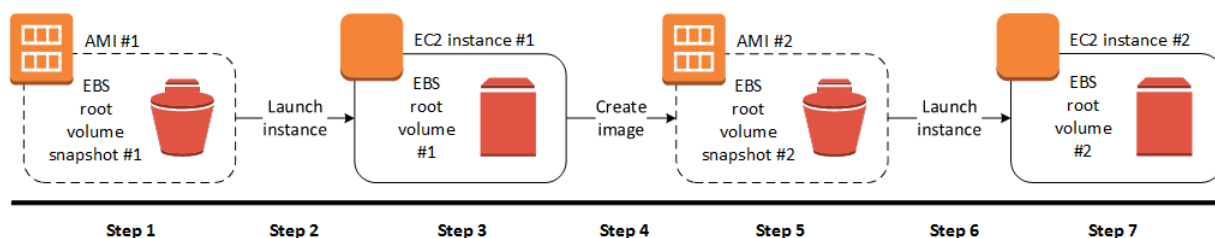
Depending on the size of the volumes, it can take several minutes for the AMI-creation process to complete (sometimes up to 24 hours). You may find it more efficient to create snapshots of your volumes prior to creating your AMI. This way, only small, incremental snapshots need to be created when the AMI is created, and the process completes more quickly (the total time for snapshot creation remains the same). For more information, see [Creating an Amazon EBS Snapshot \(p. 789\)](#).

After the process completes, you have a new AMI and snapshot created from the root volume of the instance. When you launch an instance using the new AMI, we create a new EBS volume for its root volume using the snapshot. Both the AMI and the snapshot incur charges to your account until you delete them. For more information, see [Deregistering Your AMI \(p. 92\)](#).

If you add instance-store volumes or Amazon EBS volumes to your instance in addition to the root device volume, the block device mapping for the new AMI contains information for these volumes, and the block device mappings for instances that you launch from the new AMI automatically contain information for these volumes. The instance-store volumes specified in the block device mapping for the new instance are new and don't contain any data from the instance store volumes of the instance you used to create the AMI. The data on EBS volumes persists. For more information, see [Block Device Mapping \(p. 833\)](#).

Creating a Windows AMI from a Running Instance

You can create an AMI using the AWS Management Console or the command line. The following diagram summarizes the process for creating an Amazon EBS-backed AMI from a running EC2 instance. Start with an existing AMI, launch an instance, customize it, create a new AMI from it, and finally launch an instance of your new AMI. The steps in the following diagram match the steps in the procedure below. If you already have a running Amazon EBS-backed instance, you can go directly to step 4.



To create an AMI from an instance using the console

1. Select an appropriate EBS-backed AMI to serve as a starting point for your new AMI. To view the EBS-backed Windows AMIs, choose the following options from the **Filter lists**: **Public images**, **EBS images**, and then **Windows**.

You can select any public AMI that uses the version of Windows Server that you want for your AMI. However, you must select an EBS-backed AMI; don't start with an instance store-backed AMI.

2. Choose **Launch** to launch an instance of the EBS-backed AMI that you've selected. Accept the default values as you step through the wizard. For more information, see [Launching an Instance \(p. 244\)](#).
3. While the instance is running, connect to it.

You can perform any of the following actions on your instance to customize it for your needs:

- Install software and applications
- Copy data
- Reduce start time by deleting temporary files, defragmenting your hard drive, and zeroing out free space
- Attach additional EBS volumes
- Create a new user account and add it to the Administrators group

Tip

If you are sharing your AMI, these credentials can be supplied for RDP access without disclosing your default administrator password.

- Configure settings using EC2Config. If you want your AMI to generate a random password at launch time, you need to enable the `Ec2SetPassword` plugin; otherwise, the current administrator password is used. For more information, see [Configuring a Windows Instance Using the EC2Config Service \(p. 283\)](#).
- If the instance uses RedHat drivers to access Xen virtualized hardware, upgrade to Citrix drivers before you create an AMI. For more information, see [Upgrading PV Drivers on Your Windows AMI \(p. 356\)](#).

(Optional) When the instance is configured correctly, it is best to stop the instance before you create the AMI, to ensure data integrity. You can use EC2Config to stop the instance, or select the instance in the Amazon EC2 console and choose **Actions, Instance State, Stop**.

4. In the navigation pane, choose **Instances** and select your instance. Choose **Actions, Image, and Create Image**.

Tip

If this option is disabled, your instance isn't an Amazon EBS-backed instance.

In the **Create Image** dialog box, specify values for the following fields, and then choose **Create Image**.

Name

A unique name for the image.

Description

(Optional) A description of the image, up to 255 characters.

By default, Amazon EC2 shuts down the instance, takes snapshots of any attached volumes, creates and registers the AMI, and then reboots the instance. Choose **No reboot** if you don't want your instance to be shut down.

Warning

If you choose **No reboot**, we can't guarantee the file system integrity of the created image.

You can modify the root volume, Amazon EBS volumes, and instance store volumes as follows:

- To change the size of the root volume, locate the **Root** volume in the **Type** column, and fill in the **Size** field.
- To suppress an Amazon EBS volume specified by the block device mapping of the AMI used to launch the instance, locate the EBS volume in the list and choose **Delete**.

- To add an Amazon EBS volume, choose **Add New Volume, Type**, and **EBS**, and fill in the fields. When you then launch an instance from your new AMI, these additional volumes are automatically attached to the instance. Empty volumes must be formatted and mounted. Volumes based on a snapshot must be mounted.
 - To suppress an instance store volume specified by the block device mapping of the AMI used to launch the instance, locate the volume in the list and choose **Delete**.
 - To add an instance store volume, choose **Add New Volume, Type**, and **Instance Store**, and select a device name from the **Device** list. When you launch an instance from your new AMI, these additional volumes are automatically initialized and mounted. These volumes don't contain data from the instance store volumes of the running instance from which you based your AMI.
5. While your AMI is being created, you can choose **AMIs** in the navigation pane to view its status. Initially, this is `pending`. After a few minutes, the status should change to `available`.

(Optional) Choose **Snapshots** in the navigation pane to view the snapshot that was created for the new AMI. When you launch an instance from this AMI, we use this snapshot to create its root device volume.

6. Launch an instance from your new AMI. For more information, see [Launching an Instance \(p. 244\)](#). The new running instance contains all of the customizations you applied in previous steps.

Note

You can specify scripts to execute when an instance starts. You enter the script in the **User data** section of the Instance Configuration Wizard. For example, you could specify a PowerShell script to rename an instance when it starts. For more information, see [Configuring Instances with User Data \(p. 273\)](#).

To create an AMI from an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- `create-image` (AWS CLI)
- `New-EC2Image` (AWS Tools for Windows PowerShell)

Creating an Instance Store-Backed Windows AMI

To create an instance store-backed Windows AMI, first launch and customize a Windows instance, then bundle the instance, and register an AMI from the manifest that's created during the bundling process.

Important

The only Windows AMIs that can be backed by instance store are those for Windows Server 2003. Instance store-backed instances don't have the available disk space required for later versions of Windows Server.

You can only bundle an instance store-backed Windows instance using this procedure. If you need to create an instance store-backed Linux AMI, see [Creating an Instance Store-Backed Linux AMI](#) in the *Amazon EC2 User Guide for Linux Instances*.

The AMI creation process is different for Amazon EBS-backed AMIs. For more information about the differences between Amazon EBS-backed and instance store-backed instances, and how to determine the root device type for your instance, see [Root Device Volume \(p. 8\)](#). If you need to create an Amazon EBS-backed Windows AMI, see [Creating an Amazon EBS-Backed Windows AMI \(p. 77\)](#).

Contents

- [Instance Store-Backed Windows AMIs \(p. 81\)](#)
- [Preparing to Create an Instance Store-Backed Windows AMI \(p. 82\)](#)
- [Bundling an Instance Store-Backed Windows Instance \(p. 82\)](#)
- [Registering an Instance Store-Backed Windows AMI \(p. 83\)](#)

Instance Store-Backed Windows AMIs

Instances launched from an AMI backed by instance store use an instance store volume as the root device volume. The image of the root device volume of an instance store-backed AMI is initially stored in Amazon S3. When an instance is launched using an instance store-backed AMI, the image of its root device volume is copied from Amazon S3 to the root partition of the instance. The root device volume is then used to boot the instance.

When you create an instance store-backed AMI, it must be uploaded to Amazon S3. Amazon S3 stores data objects in buckets, which are similar in concept to directories. Buckets have globally unique names and are owned by unique AWS accounts.

Bundling Process

The bundling process comprises the following tasks:

- Compress the image to minimize bandwidth usage and storage requirements.
- Encrypt and sign the compressed image to ensure confidentiality and authenticate the image against its creator.
- Split the encrypted image into manageable parts for upload.
- Run `Sysprep` to strip computer-specific information (for example, the MAC address and computer name) from the Windows AMI to prepare it for virtualization.
- Create a manifest file that contains a list of the image parts with their checksums.
- Put all components of the AMI in the Amazon S3 bucket that you specify when making the bundle request.

Storage Volumes

It is important to remember the following details about the storage for your instance when you create an instance store-backed AMI:

- The root device volume (C:) is automatically attached when a new instance is launched from your new AMI. The data on any other instance store volumes is deleted when the instance is bundled.
- The instance store volumes other than the root device volume (for example, D:) are temporary and should be used only for short-term storage.
- You can add Amazon EBS volumes to your instance store-based instance. Amazon EBS volumes are stored within Amazon S3 buckets and remain intact when the instance is bundled. Therefore, we recommend that you store all the data that must persist on Amazon EBS volumes, not instance store volumes.

For more information about Amazon EC2 storage options, see [Storage \(p. 744\)](#).

Preparing to Create an Instance Store-Backed Windows AMI

When you create an AMI, you start by basing it on an instance. You can customize the instance to include the data and software that you need. As a result, any instance that you launch from your AMI has everything that you need.

To launch an instance store-backed Windows instance

1. Open the Amazon EC2 console.
2. In the navigation pane, click **AMIs**. Select an instance store-backed AMI that is similar to the AMI that you want to create. To view the instance store-backed Windows AMIs, select the following options from the **Filter** lists: **Public images**, **Instance store images**, and then **Windows**.

You can select any public AMI that uses the version of Windows Server that you want for your AMI. However, you must select an instance store-backed AMI; don't start with an Amazon EBS-backed AMI.

3. Click **Launch** to launch an instance of the instance store-backed AMI that you've selected. Accept the default values as you step through the wizard.
4. While the instance is running, connect to it and customize it. For example, you can perform any of the following on your instance:
 - Install software and applications.
 - Copy data.
 - Reduce start time by deleting temporary files, defragmenting your hard drive, and zeroing out free space.
 - Create a new user account and add it to the Administrators group.

Tip

If you are sharing your AMI, these credentials can be provided for RDP access without disclosing your default Administrator password.

- Configure settings using EC2Config. For example, to generate a random password for your instance when you launch it from this AMI, enable the Ec2SetPassword plugin; otherwise, the current Administrator password is used. For more information, see [Configuring a Windows Instance Using the EC2Config Service \(p. 283\)](#).
5. If the instance uses RedHat drivers to access Xen virtualized hardware, upgrade to Citrix drivers before you create an AMI. For more information, see [Upgrading PV Drivers on Your Windows AMI \(p. 356\)](#).

Bundling an Instance Store-Backed Windows Instance

Now that you've customized your instance, you can bundle the instance to create an AMI, using either the AWS Management Console or the command line.

To bundle an instance store-backed Windows instance using the console

1. Determine whether you'll use an existing Amazon S3 bucket for your new AMI or create a new one. To create a new Amazon S3 bucket, use the following steps:
 - a. Open the Amazon S3 console.
 - b. Click **Create Bucket**.
 - c. Specify a name for the bucket and click **Create**.

2. Open the Amazon EC2 console.
3. In the navigation pane, click **Instances**. Right-click the instance you set up in the previous procedure, and select **Bundle Instance (instance store AMI)**.
4. In the **Bundle Instance** dialog box, fill in the requested information, and then click **OK**:
 - **Amazon S3 bucket name**: Specify the name of an S3 bucket that you own. The bundle files and manifest will be stored in this bucket.
 - **Amazon S3 key name**: Specify a prefix for the files that are generated by the bundle process.

The **Bundle Instance** dialog box confirms that the request to bundle the instance has succeeded, and also provides the ID of the bundle task. Click **Close**.

To view the status of the bundle task, click **Bundle Tasks** in the navigation pane. The bundle task progresses through several states, including `waiting-for-shutdown`, `bundling`, and `storing`. If the bundle task can't be completed successfully, the status is `failed`.

To bundle an instance store-backed Windows instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- `bundle-instance` (AWS CLI)
- `New-EC2InstanceBundle` (AWS Tools for Windows PowerShell)

Registering an Instance Store-Backed Windows AMI

Finally, you must register your AMI so that Amazon EC2 can locate it and launch instances from it.

Your new AMI is stored in Amazon S3. You'll incur charges for this storage until you deregister the AMI and delete the bundle in Amazon S3.

If you make any changes to the source AMI stored in Amazon S3, you must deregister and reregister the AMI before the changes take effect.

To register an instance store-backed Windows AMI from the AMI page in the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **AMIs**. By default, the console displays the AMIs that you own.
3. Click **Actions** and select **Register new AMI**.
4. In the **Register Image** dialog box, provide the **AMI Manifest Path** and then click **Register**.

To register an instance store-backed Windows AMI from the Bundle Tasks page in the console

1. On the navigation pane, click **Bundle Tasks**.
2. Select the bundle task, and click **Register as an AMI**.
3. A dialog displays the AMI manifest path. Click **Register**, and then click **Close** in the confirmation dialog box.

To register an instance store-backed Windows AMI using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [register-image](#) (AWS CLI)
- [Register-EC2Image](#) (AWS Tools for Windows PowerShell)

To view your new AMI, click **AMIs** in the navigation pane, and ensure the **Owned by me** filter option is selected.

AMIs with Encrypted Snapshots

AMIs that are backed by Amazon EBS snapshots can take advantage of Amazon EBS encryption. Snapshots of both data and root volumes can be encrypted and attached to an AMI.

EC2 instances with encrypted volumes are launched from AMIs in the same way as other instances.

The `CopyImage` action can be used to create an AMI with encrypted snapshots from an AMI with unencrypted snapshots. By default, `CopyImage` preserves the encryption status of source snapshots when creating destination copies. However, you can configure the parameters of the copy process to also encrypt the destination snapshots.

Snapshots can be encrypted with either your default AWS Key Management Service customer master key (CMK), or with a custom key that you specify. You must in all cases have permission to use the selected key. If you have an AMI with encrypted snapshots, you can choose to re-encrypt them with a different encryption key as part of the `CopyImage` action. `CopyImage` accepts only one key at a time and encrypts all of an image's snapshots (whether root or data) to that key. However, it is possible to manually build an AMI with snapshots encrypted to multiple keys.

Support for creating AMIs with encrypted snapshots is accessible through the Amazon EC2 console, Amazon EC2 API, or the AWS CLI.

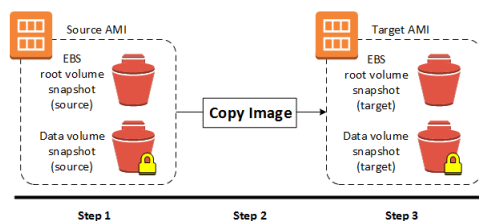
The encryption parameters of `CopyImage` are available in all regions where AWS KMS is available.

AMI Scenarios Involving Encrypted EBS Snapshots

You can copy an AMI and simultaneously encrypt its associated EBS snapshots using the AWS Management Console or the command line.

Copying an AMI with an Encrypted Data Snapshot

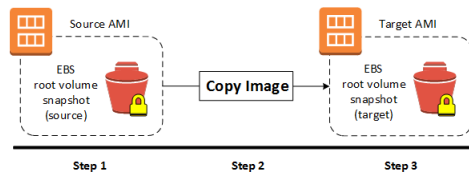
In this scenario, an EBS-backed AMI has an unencrypted root snapshot and an encrypted data snapshot, shown in step 1. The `CopyImage` action is invoked in step 2 without encryption parameters. As a result, the encryption status of each snapshot is preserved, so that the destination AMI, in step 3, is also backed by an unencrypted root snapshot and an encrypted data snapshot. Though the snapshots contain the same data, they are distinct from each other and you will incur storage costs for the snapshots in both AMIs, as well as charges for any instances you launch from either AMI.



You can perform a simple copy such as this using either the Amazon EC2 console or the command line. For more information, see [Copying an AMI](#) (p. 87).

Copying an AMI Backed by An Encrypted Root Snapshot

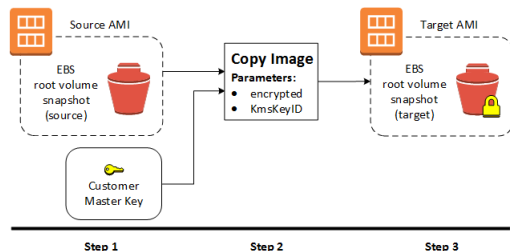
In this scenario, an Amazon EBS-backed AMI has an encrypted root snapshot, shown in step 1. The `CopyImage` action is invoked in step 2 without encryption parameters. As a result, the encryption status of the snapshot is preserved, so that the destination AMI, in step 3, is also backed by an encrypted root snapshot. Though the root snapshots contain identical system data, they are distinct from each other and you will incur storage costs for the snapshots in both AMIs, as well as charges for any instances you launch from either AMI.



You can perform a simple copy such as this using either the Amazon EC2 console or the command line. For more information, see [Copying an AMI \(p. 87\)](#).

Creating an AMI with Encrypted Root Snapshot from an Unencrypted AMI

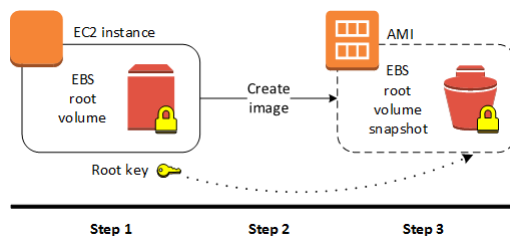
In this scenario, an Amazon EBS-backed AMI has an unencrypted root snapshot, shown in step 1, and an AMI is created with an encrypted root snapshot, shown in step 3. The `CopyImage` action in step 2 is invoked with two encryption parameters, including the choice of a CMK. As a result, the encryption status of the root snapshot changes, so that the target AMI is backed by a root snapshot containing the same data as the source snapshot, but encrypted using the specified key. You will incur storage costs for the snapshots in both AMIs, as well as charges for any instances you launch from either AMI.



You can perform a copy and encrypt operation such as this using either the Amazon EC2 console or the command line. For more information, see [Copying an AMI \(p. 87\)](#).

Creating an AMI with an Encrypted Root Snapshot from a Running Instance

In this scenario, an AMI is created from a running EC2 instance. The running instance in step 1 has an encrypted root volume, and the created AMI in step 3 has a root snapshot encrypted to the same key as the source volume. The `CreateImage` action has exactly the same behavior whether or not encryption is present.



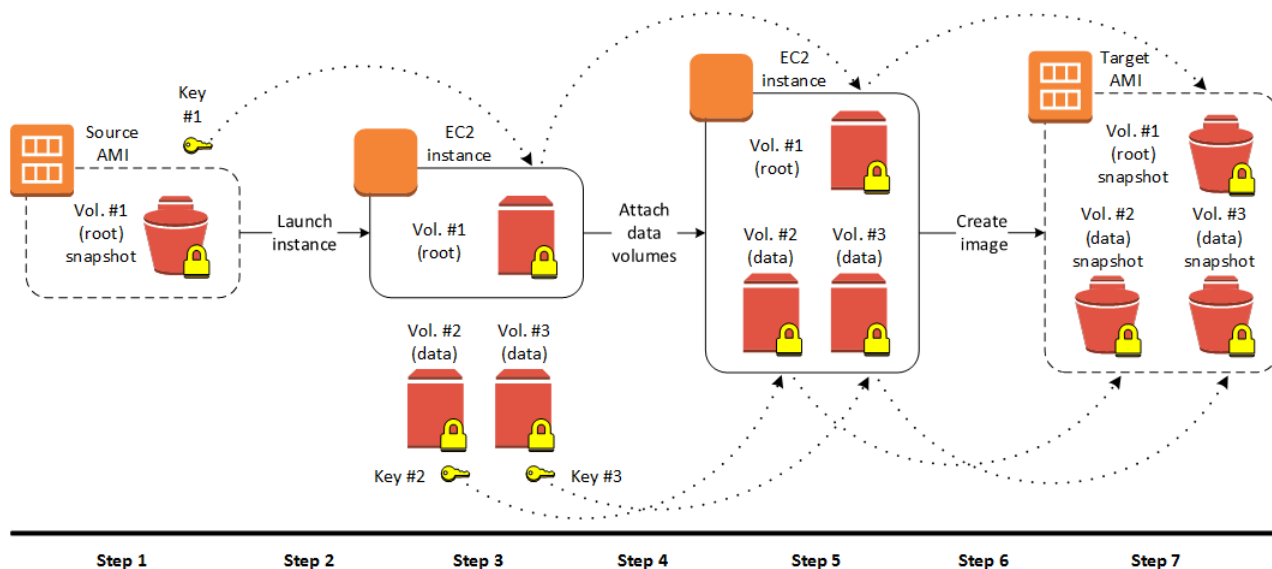
You can create an AMI from a running Amazon EC2 instance (with or without encrypted volumes) using either the Amazon EC2 console or the command line. For more information, see [Creating an Amazon EBS-Backed Windows AMI \(p. 77\)](#).

Creating an AMI with Unique CMKs for Each Encrypted Snapshot

This scenario starts with an AMI backed by a root-volume snapshot (encrypted to key #1), and finishes with an AMI that has two additional data-volume snapshots attached (encrypted to key #2 and key #3). The `CopyImage` action cannot apply more than one encryption key in a single operation. However, you can create an AMI from an instance that has multiple attached volumes encrypted to different keys. The resulting AMI has snapshots encrypted to those keys and any instance launched from this new AMI also has volumes encrypted to those keys.

The steps of this example procedure correspond to the following diagram.

1. Start with the source AMI backed by vol. #1 (root) snapshot, which is encrypted with key #1.
2. Launch an EC2 instance from the source AMI.
3. Create EBS volumes vol. #2 (data) and vol. #3 (data), encrypted to key #2 and key #3 respectively.
4. Attach the encrypted data volumes to the EC2 instance.
5. The EC2 instance now has an encrypted root volume as well as two encrypted data volumes, all using different keys.
6. Use the `CreateImage` action on the EC2 instance.
7. The resulting target AMI contains encrypted snapshots of the three EBS volumes, all using different keys.



You can carry out this procedure using either the Amazon EC2 console or the command line. For more information, see the following topics:

- [Launch Your Instance \(p. 244\)](#)
- [Creating an Amazon EBS-Backed Windows AMI \(p. 77\)](#).
- [Amazon EBS Volumes \(p. 747\)](#)
- [AWS Key Management](#) in the *AWS Key Management Service Developer Guide*

Copying an AMI

You can copy an Amazon Machine Image (AMI) within or across an AWS region using the AWS Management Console, the command line, or the Amazon EC2 API, all of which support the `CopyImage` action. Both Amazon EBS-backed AMIs and instance store-backed AMIs can be copied.

Copying a source AMI results in an identical but distinct target AMI with its own unique identifier. In the case of an Amazon EBS-backed AMI, each of its backing snapshots is, by default, copied to an identical but distinct target snapshot. (The one exception is when you choose to encrypt the snapshot, as described below.) The source AMI can be changed or deregistered with no effect on the target AMI. The reverse is also true.

There are no charges for copying an AMI. However, standard storage and data transfer rates apply.

AWS does not copy launch permissions, user-defined tags, or Amazon S3 bucket permissions from the source AMI to the new AMI. After the copy operation is complete, you can apply launch permissions, user-defined tags, and Amazon S3 bucket permissions to the new AMI.

Copying an AMI You Own

You can copy any AMI that belongs to your AWS account using the `CopyImage` action. This includes AMIs with encrypted snapshots and encrypted AMIs.

Copying an AMI Across AWS Accounts

You can copy an AMI across AWS accounts. This includes AMIs with encrypted snapshots, but does not include encrypted AMIs.

When an AMI is copied, the owner of the source AMI is charged standard Amazon EBS or Amazon S3 transfer fees, and the owner of the target AMI is charged for storage in the destination region.

Permissions

- The owner of the account must grant read permissions on the storage that backs the AMI, whether it is an associated EBS snapshot (for an Amazon EBS-backed AMI) or an associated Amazon S3 bucket (for an instance-store-backed AMI). To allow other accounts to copy your AMIs, you must grant read permissions on your associated snapshot or bucket using the Amazon EBS or Amazon S3 access management tools.
- If you use an IAM user to copy an instance-store-backed AMI, the user must have the following Amazon S3 permissions: `s3:CreateBucket`, `s3:GetBucketAcl`, `s3:ListBuckets`, `s3:CopyObject`, `s3:GetObject`, and `s3:PutObject`.

Limits

- You can't copy an encrypted AMI between accounts. Instead, if the underlying snapshot and encryption key have been shared with you, you can copy the snapshot to another account while re-encrypting it with a key of your own, and then register this privately owned snapshot as a new AMI.
- You can't directly copy an AMI that has a `billingProduct` code associated with it. This includes Windows AMIs and other AMIs from the AWS Marketplace that are owned and shared by another AWS account.

To create a private copy of an AMI that has a `billingProduct` code associated with it, we recommend that you launch an EC2 instance in the target account using the shared AMI and then

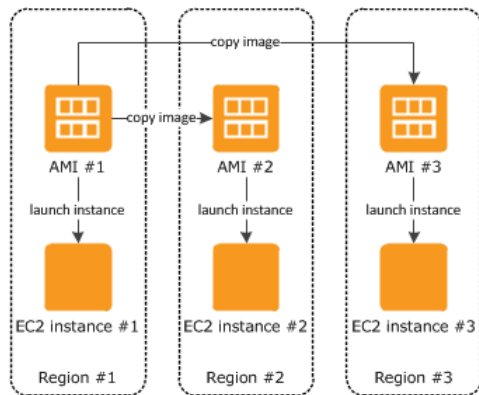
create an image from the instance. The result is a private AMI that you own and can customize. For example, if you create a private copy of an EBS-backed AMI, you can use `CopyImage` to create an AMI with an encrypted root volume. For more information, see [Creating an Amazon EBS-Backed Windows AMI \(p. 77\)](#).

Copying an AMI Across Regions

Copying an AMI across geographically diverse regions provides the following benefits:

- **Consistent global deployment:** Copying an AMI from one region to another enables you to launch consistent instances based from the same AMI into different regions.
- **Scalability:** You can more easily design and build world-scale applications that meet the needs of your users, regardless of their location.
- **Performance:** You can increase performance by distributing your application, as well as locating critical components of your application in closer proximity to your users. You can also take advantage of region-specific features, such as instance types or other AWS services.
- **High availability:** You can design and deploy applications across AWS regions, to increase availability.

The following diagram shows the relations among a source AMI and two copied AMIs in different regions, as well as the EC2 instances launched from each. When you launch an instance from an AMI, it resides in the same region where the AMI resides. If you make changes to the source AMI and want those changes to be reflected in the AMIs in the target regions, you must recopy the source AMI to the target regions.



When you first copy an instance store-backed AMI to a region, we create an Amazon S3 bucket for the AMIs copied to that region. All instance store-backed AMIs that you copy to that region are stored in this bucket. The names of these buckets have the following format: `amis-for-account-in-region-hash`. For example: `amis-for-123456789012-in-us-west-2-yhjmxvp6`.

Note

Destination regions are limited to 50 concurrent AMI copies at a time, with no more than 25 of those coming from a single source region. To request an increase to this limit, see [Amazon EC2 Service Limits \(p. 869\)](#).

Prior to copying an AMI, you must ensure that the contents of the source AMI are updated to support running in a different region. For example, you should update any database connection strings or similar application configuration data to point to the appropriate resources. Otherwise, instances launched from the new AMI in the destination region may still use the resources from the source region, which can impact performance and cost.

Copying to Encrypt

Encrypting during copying applies only to Amazon EBS-backed AMIs. Because an instance-store-backed AMIs does not rely on snapshots, the `CopyImage` action cannot be used to change its encryption status.

The `CopyImage` action can also be used to create a new AMI backed by encrypted Amazon EBS snapshots. If you invoke encryption while copying an AMI, each snapshot taken of its associated Amazon EBS volumes—including the root volume—will be encrypted using a key that you specify. For more information about using AMIs with encrypted snapshots, see [AMIs with Encrypted Snapshots \(p. 84\)](#).

By default, the backing snapshot of an AMI will be copied with its original encryption status. Copying an AMI backed by an unencrypted snapshot will result in an identical target snapshot that is also unencrypted. If the source AMI is backed by an encrypted snapshot, copying it will result in a target snapshot encrypted to the specified key. Copying an AMI backed by multiple snapshots preserves the source encryption status in each target snapshot. For more information about copying AMIs with multiple snapshots, see [AMIs with Encrypted Snapshots \(p. 84\)](#).

The following table shows encryption support for various scenarios. Note that while it is possible to copy an unencrypted snapshot to yield an encrypted snapshot, you cannot copy an encrypted snapshot to yield an unencrypted one.

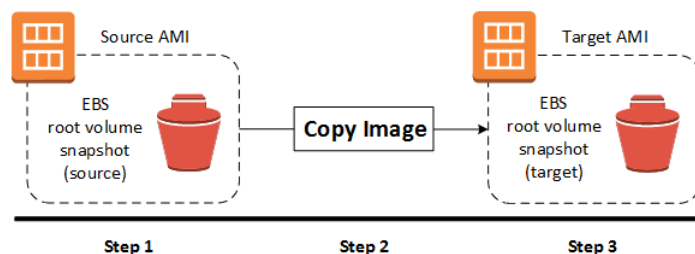
Scenario	Description	Supported
1	Unencrypted-to-unencrypted	Yes
2	Encrypted-to-encrypted	Yes
3	Unencrypted-to-encrypted	Yes
4	Encrypted-to-unencrypted	No

AMI Copying Scenarios

This section describes basic scenarios for copying AMIs and provides copy procedures using the Amazon EC2 console and the command line.

Copy an unencrypted source AMI to an unencrypted target AMI

In the simplest case, a copy of an AMI with an unencrypted single backing snapshot is created in the specified geographical region (not shown).



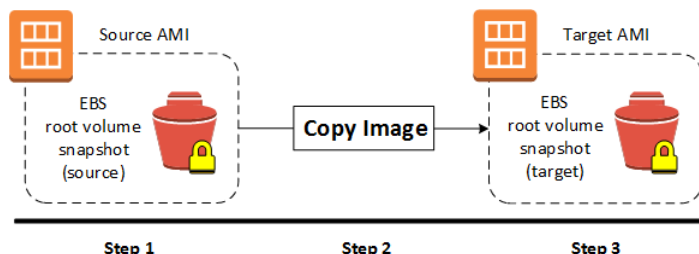
Note

Although the diagram above shows an AMI with a single backing snapshot, the `CopyImage` action also works for AMIs with multiple snapshots. The encryption status of each snapshot is preserved. This means that an unencrypted snapshot in the source AMI will cause an

unencrypted snapshot to be created in the target AMI, and an encrypted snapshot in the source AMI will cause an encrypted snapshot to be created in the target AMI.

Copy an encrypted source AMI to an encrypted target AMI

Although this scenario involves encrypted snapshots, it is functionally equivalent to the previous scenario.

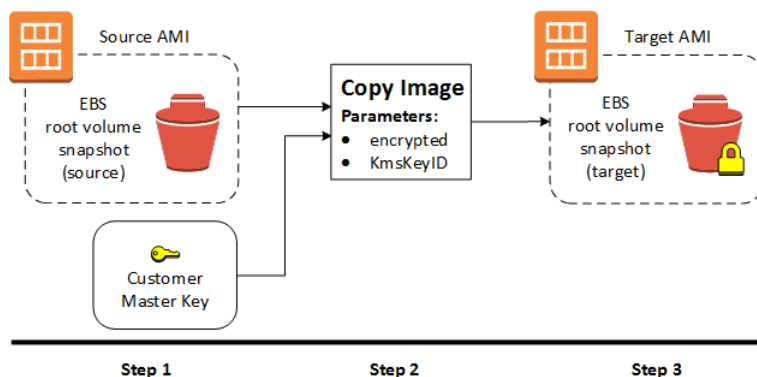


Note

If you apply encryption while copying a multi-snapshot AMI, all of the target snapshots are encrypted using the specified key or the default key if none is specified. For information about creating an AMI with multiple snapshots encrypted to multiple keys, see [AMIs with Encrypted Snapshots](#) (p. 84).

Copy an unencrypted source AMI to an encrypted target AMI

In this last scenario, the `CopyImage` action changes the encryption status of the destination image, for instance, by encrypting an unencrypted snapshot, or re-encrypting an encrypted snapshot with a different key. To apply encryption during the copy, you must supply encryption parameters: an encryption flag and a key. Volumes created from the target snapshot are accessible only if you supply this key. For more information about supported encryption scenarios for AMIs, see [AMIs with Encrypted Snapshots](#) (p. 84).



Copying an AMI Using the Console or Command Line

The steps in the following procedure correspond to the three steps in each scenario diagram. Apart from the configuration of encryption options, the procedure for implementing the `CopyImage` action is identical in all cases.

To copy an AMI using the console

1. Create or obtain an AMI backed by an Amazon EBS snapshot. For more information, see [Creating an Amazon EBS-Backed Windows AMI](#) (p. 77). A wide variety of AWS-supplied AMIs are available through the Amazon EC2 console.

From the console navigation bar, select the region that contains the AMI you wish to copy. In the navigation pane, expand **Images** and select **AMIs** to display the list of AMIs available to you in the selected region.

2. Select the AMI to copy and choose **Actions** and **Copy AMI**.

In the **AMI Copy** page, set the following fields and choose **Copy AMI**:

- **Destination region:** Choose the region into which to copy the AMI.
 - **Name:** Provide a name for the new AMI. You may want to include operating system information in the name, as we do not provide this information when displaying details about the AMI.
 - **Description:** By default, the description includes information about the source AMI so that you can distinguish a copy from its original. You can change this description as needed.
 - **Encryption:** Select this field to encrypt the target Amazon EBS snapshots, or to re-encrypt them using a different key.
 - **Master Key:** The KMS key that will be used to encrypt the target Amazon EBS snapshots if **Encryption** has been chosen.
3. We display a confirmation page to let you know that the copy operation has been initiated and to provide you with the ID of the new AMI.

To check on the progress of the copy operation immediately, follow the provided link. To check on the progress later, choose **Done**, and then when you are ready, use the navigation bar to switch to the target region (if applicable) and locate your AMI in the list of AMIs.

The initial status of the target AMI is `pending` and the operation is complete when the status is `available`.

To copy an AMI using the command line

Copying an AMI using the command line requires that you specify both the source and destination regions. You specify the source region using the `--source-region` parameter. For the destination region, you have two options:

- Use the `--region` parameter.
- Set an environmental variable. For more information, see [Configuring the AWS Command Line Interface](#).

When you encrypt a target snapshot during copying, you will need to supply two additional parameters:

- A Boolean, `--encrypted`
- A string, `--kms-key-id`, providing the master encryption key ID

You can copy an AMI using one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- `copy-image` (AWS CLI)
- `Copy-EC2Image` (AWS Tools for Windows PowerShell)

Stopping a Pending AMI Copy Operation

You can stop a pending AMI copy using the AWS Management Console or the command line.

To stop an AMI copy operation using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select the destination region from the region selector.
3. In the navigation pane, choose **AMIs**.
4. Select the AMI to stop copying and choose **Actions** and **Deregister**.
5. When asked for confirmation, choose **Continue**.

To stop an AMI copy operation using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [deregister-image](#) (AWS CLI)
- [Unregister-EC2Image](#) (AWS Tools for Windows PowerShell)

Deregistering Your AMI

You can deregister an AMI when you have finished using it. After you deregister an AMI, you can't use it to launch new instances.

When you deregister an AMI, it doesn't affect any instances that you've already launched from the AMI. You'll continue to incur usage costs for these instances. Therefore, if you are finished with these instances, you should terminate them.

The procedure that you'll use to clean up your AMI depends on whether it is backed by Amazon EBS or instance store. (Note that the only Windows AMIs that can be backed by instance store are those for Windows Server 2003.)

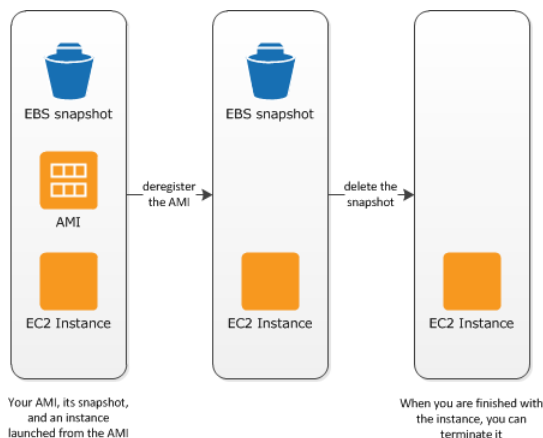
Contents

- [Cleaning Up Your Amazon EBS-Backed AMI \(p. 92\)](#)
- [Cleaning Up Your Instance Store-Backed AMI \(p. 93\)](#)

Cleaning Up Your Amazon EBS-Backed AMI

When you deregister an Amazon EBS-backed AMI, it doesn't affect the snapshot that was created for the root volume of the instance during the AMI creation process. You'll continue to incur storage costs for this snapshot. Therefore, if you are finished with the snapshot, you should delete it.

The following diagram illustrates the process for cleaning up your Amazon EBS-backed AMI.



To clean up your Amazon EBS-backed AMI

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **AMIs**. Select the AMI, and take note of its ID — this can help you find the correct snapshot in the next step. Choose **Actions**, and then **Deregister**. When prompted for confirmation, choose **Continue**.

The AMI status is now `unavailable`.

Note

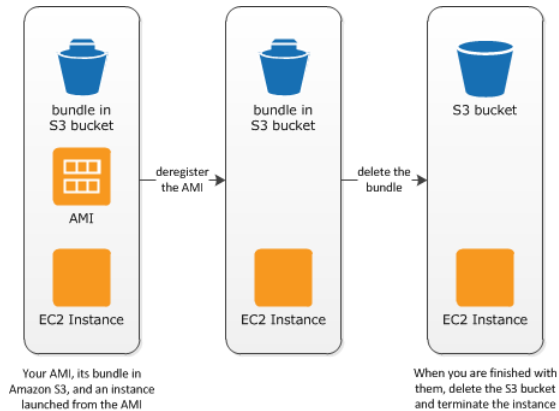
It may take a few minutes before the console changes the status from `available` to `unavailable`, or removes the AMI from the list altogether. Choose **Refresh** to refresh the status.

3. In the navigation pane, choose **Snapshots**, and select the snapshot (look for the AMI ID in the **Description** column). Choose **Actions**, and then choose **Delete Snapshot**. When prompted for confirmation, choose **Yes, Delete**.
4. (Optional) If you are finished with an instance that you launched from the AMI, terminate it. In the navigation pane, choose **Instances**. Select the instance, choose **Actions**, then **Instance State**, and then **Terminate**. When prompted for confirmation, choose **Yes, Terminate**.

Cleaning Up Your Instance Store-Backed AMI

When you deregister an instance store-backed AMI, it doesn't affect the files that you uploaded to Amazon S3 when you created the AMI. You'll continue to incur usage costs for these files in Amazon S3. Therefore, if you are finished with these files, you should delete them.

The following diagram illustrates the process for cleaning up your instance store-backed AMI.



To clean up your instance store-backed AMI

1. Deregister the AMI using the `deregister-image` command as follows.

```
aws ec2 deregister-image --image-id ami_id
```

The AMI status is now `unavailable`.

2. Delete the bundle in Amazon S3 using the `rm` command. For example, this command recursively removes the files that start with `mybundle` (assume this is the S3 key you used when you created the bundle, and assume you don't have other important objects in this bucket that use this key).

```
aws s3 rm s3://myawsbucket/myami --recursive --exclude "*" --include "mybundle.*"
```

3. (Optional) If you are finished with an instance that you launched from the AMI, you can terminate it using the `terminate-instances` command as follows.

```
aws ec2 terminate-instances --instance-ids instance_id
```

4. (Optional) If you are finished with the Amazon S3 bucket that you uploaded the bundle to, you can delete the bucket. To delete an Amazon S3 bucket, open the Amazon S3 console, select the bucket, choose **Actions**, and then choose **Delete**.

AWS Windows AMI Version History

AWS provides Amazon Machine Images (AMIs) that contain versions of Windows Server, known as the *AWS Windows AMIs*. Some AWS Windows AMIs also come configured with Microsoft SQL Server or Internet Information Services (IIS). You can use an AMI with Microsoft SQL Server and IIS already configured, or you can start from a basic Windows AMI, and then install Microsoft SQL Server and enable IIS on the instance. For more information, see [AWS Windows AMIs \(p. 62\)](#).

Contents

- [Configuration Settings and Drivers \(p. 95\)](#)
- [Updating Your Windows Instance \(p. 95\)](#)
- [Upgrading or Migrating a Windows Server Instance \(p. 95\)](#)
- [Determining Your Instance Version \(p. 95\)](#)
- [Subscribing to Windows AMI Notifications \(p. 96\)](#)

- [Image Changes \(p. 97\)](#)
- [Details About AWS Windows AMI Versions \(p. 98\)](#)
- [Changes in Windows Server 2016 AMIs \(p. 108\)](#)

Configuration Settings and Drivers

The AWS Windows AMIs are generally configured the same way as a Windows Server that you install from Microsoft-issued media. There are, however, a few differences in the installation defaults.

AWS Windows AMIs come with an additional service installed, the EC2Config service. The EC2Config service runs in the local system account and is primarily used during the initial setup. For information about the tasks that EC2Config performs, see [Overview of EC2Config Tasks \(p. 284\)](#).

After you launch your Windows instance with its initial configuration, you can use the EC2Config service to change the configuration settings as part of the process of customizing and creating your own AMI. Instances launched from your customized AMI are launched with the new configuration.

AWS Windows AMIs contain a set of drivers to permit access to Xen virtualized hardware. These drivers are used by Amazon EC2 to map instance store and Amazon EBS volumes to their devices. For more information, see [Paravirtual Drivers \(p. 352\)](#).

Updating Your Windows Instance

After you launch a Windows instance, you are responsible for installing updates on it. You can manually install only the updates that interest you, or you can start from a current AWS Windows AMI and build a new Windows instance. For information about finding the current AWS Windows AMIs, see [Finding a Windows AMI \(p. 67\)](#).

For Windows instances, you can install updates to the following services or applications:

- [Windows](#)
- [Microsoft SQL Server](#)
- [Windows PowerShell](#)
- [EC2Config service \(p. 295\)](#)
- [PV Drivers \(p. 356\)](#)
- [AWS Tools for Windows PowerShell](#)
- [AWS CloudFormation helper scripts](#)

You can reboot a Windows instance after installing updates. For more information, see [Reboot Your Instance \(p. 262\)](#).

Upgrading or Migrating a Windows Server Instance

For information about how to upgrade or migrate an instance to a newer version of Windows, see [Upgrading a Windows Server EC2 Instance to a Newer Version of Windows Server](#).

Determining Your Instance Version

The AWS Management Console provides details about the AMI that you use to create an Amazon EC2 instance. The **AMI ID** field on the **Description** tab contains information including the Windows Server SKU, the architecture (32-bit or 64-bit), the date the AMI was created, and an AMI ID.

Description	Status Checks	Monitoring	Tags
Instance ID			
Instance state	stopped		
Instance type	t2.micro		
Private DNS			
Private IPs			
Secondary private IPs			
VPC ID			
Subnet ID			
			Public DNS
			Public IP
			Elastic IP
			Availability zone us-west-2a
			Security groups
			Scheduled events
			AMI ID Windows_Server-2012-R2_RTM-Englis
			Platform windows

If an AMI has been made private or replaced by later versions and is no longer listed in the catalog, the **AMI ID** field states, "Cannot load detail for ami-xxxxx. You may not be permitted to view it." To determine which AMI was used to create the instance, you must open the system log. In the EC2 console, choose an instance, and from the context-menu (right-click) choose **Instance Settings** and then choose **Get System Log**. The date the AMI was created and the SKU are listed in the **AMI Origin Version** and **AMI Origin Name** fields.

```
2015/04/23 17:19:172: EC2ConfigMonitorState: 0
2015/04/23 17:19:182: AMI Origin Version: 2015.04.15
2015/04/23 17:19:182: AMI Origin Name: Windows_Server-2003-R2_SP2-English-32Bit-Base
2015/04/23 17:19:182: OS: Microsoft Windows NT 5.2.3790
2015/04/23 17:19:182: OsVersion: 5.2
2015/04/23 17:19:182: OsProductName: Microsoft Windows Server 2003 R2
2015/04/23 17:19:182: OsBuildLabEx: NotFound
2015/04/23 17:19:182: Language: en-US
2015/04/23 17:19:182: EC2 Agent: Ec2Config service v3.3.174
2015/04/23 17:19:192: Message: Waiting for meta-data accessibility...
2015/04/23 17:19:202: Message: Meta-data is now available.
2015/04/23 17:19:202: Driver: Citrix EV Ethernet Adapter v5.9.960.49119
2015/04/23 17:19:202: Driver: Citrix EV SCSI Host Adapter v6.0.2.56921
2015/04/23 17:19:332: AMI-ID: ami-a3c9e393
```

Note

The **AMI Origin Version** and **AMI Origin Name** are displayed in the system log only if the EC2Config service is running version 2.1.19 or later and the AMI was created after 2013.11.13.

Subscribing to Windows AMI Notifications

If you want to be notified when new AMIs are released or when the previous AMIs are made private, you can subscribe to these notifications using Amazon SNS.

To subscribe to Windows AMI notifications

1. Open the Amazon SNS console.
2. In the navigation bar, change the region to **US East (N. Virginia)**, if necessary. You must select this region because the SNS notifications that you are subscribing to were created in this region.
3. In the navigation pane, click **Subscriptions**.
4. Click **Create Subscription**.
5. In the **Create Subscription** dialog box, do the following:
 - a. In **TopicARN**, enter one of the following Amazon Resource Names (ARNs):
 - arn:aws:sns:us-east-1:801119661308:ec2-windows-ami-update
 - arn:aws:sns:us-east-1:801119661308:ec2-windows-ami-private
 - b. In **Protocol**, select **Email**.
 - c. In **Endpoint**, enter an email address that you can use to receive the notifications.
 - d. Click **Subscribe**.
6. You'll receive a confirmation email with the subject line **AWS Notification - Subscription Confirmation**. Open the email and click **Confirm subscription** to complete your subscription.

Whenever new Windows AMIs are released, we send notifications to subscribers of the `ec2-windows-ami-update` topic. Whenever new Windows AMIs are made private, we send notifications

to subscribers of the `ec2-windows-ami-private` topic. If you no longer want to receive these notifications, use the following procedure to unsubscribe.

To unsubscribe from Windows AMI notifications

1. Open the Amazon SNS console.
2. In the navigation pane, click **Subscriptions**.
3. Select the subscription and then click **Delete Subscriptions**. When prompted for confirmation, click **Yes, Delete**.

Image Changes

The following changes are applied to each Amazon-provided image.

- Allow Internet Control Message Protocol (ICMP) traffic through firewall
- Set performance options for best performance
- Set power setting to high performance
- Disable screensaver password
- Disable hibernation
- Disable clearing page file at shutdown
- Add links to desktop EC2 Windows Guide (<http://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/concepts.html>) and EC2 Feedback (https://aws.qualtrics.com/se/?sid=sv_e5mofjvhv18gtayw)
- Set timezone to UTC
- Configure page file (512 MB to 8 GB)
- Install PowerShell tools (<http://aws.amazon.com/powershell>)
- Install the latest version of the EC2Config service
- Disable Windows network location profile selection prompt
- Install Cloud Formation tools (<http://aws.amazon.com/developertools/aws-cloudformation/4026240853893296>)
- Disable IPv6 in network adapters
- Disable NetBIOS in network adapters
- Install PowerShell 3.0 for images earlier than Windows Server 2012
- Enable remote PowerShell
- Enable file and printer sharing
- Open port 1433 for images that include SQL Server
- Enable notification of Windows updates
- Sync time daily via NTP
- Disable Windows Internet Explorer RunOnce
- Apply the following hotfixes for Windows Server 2008 or Server 2008 R2 images:
 - GARP (<http://support.microsoft.com/kb/2582281>)
 - Microsoft DST (<http://support.microsoft.com/kb/2800213>)
 - Microsoft RTIU clock sync (<http://support.microsoft.com/kb/2922223>)
 - ELB (<http://support.microsoft.com/kb/2634328>)
 - TCP scaling (<http://support.microsoft.com/kb/2780879>)
 - SMB2 (<http://support.microsoft.com/kb/2394911>)
- Attach instance storage volumes to extended mount points (25)
- Install latest Windows updates

Details About AWS Windows AMI Versions

AWS provides updated, fully-patched Windows AMIs within five business days of Microsoft's patch Tuesday (the second Tuesday of each month). The new AMIs are available immediately through the **Images** page in the Amazon EC2 console. The new AMIs are available in the AWS Marketplace and the **Quick Start** tab of the launch instance wizard within a few days of their release. AWS makes the previously published Windows AMIs private within 10 business days after publishing updated Windows AMIs, to ensure that customers have the latest security updates by default.

The Windows AMIs in each release have new AMI IDs. Therefore, we recommend that you write scripts that locate the latest AWS Windows AMIs by their names, rather than by their IDs. For more information, see [Get-EC2ImageByName](#) in the *AWS Tools for Windows PowerShell User Guide*. You can also create a Lambda function to perform this task with Amazon EC2 and other services such as AWS CloudFormation. For more information, see [Create a Lambda Function](#).

Contents

- [Latest AMIs \(p. 98\)](#)
- [AMIs Released in 2015 \(p. 101\)](#)
- [AMIs Released in 2014 \(p. 103\)](#)
- [AMIs Released in 2013 \(p. 105\)](#)
- [AMIs Released in 2012 \(p. 107\)](#)
- [AMIs Released in 2011 and earlier \(p. 108\)](#)

The following tables summarize the changes to each release of the AWS Windows AMIs. Note that some changes apply to all AWS Windows AMIs while others apply to only a subset of these AMIs.

Latest AMIs

Release	Changes
2016.11.23	<ul style="list-style-type: none">• Released EC2Config version 4.1.1378.• The Windows Server 2003-2012 R2 AMIs released this month, and going forward, use the EC2Config service to process boot-time configurations and Amazon EC2 Simple Systems Manager (SSM) Agent to process Amazon EC2 Run Command and SSM Config requests. EC2Config no longer processes requests for Run Command and SSM Config. The latest EC2Config installer installs SSM Agent side-by-side with the EC2Config service. For more information, see EC2Config and Amazon EC2 Simple Systems Manager (SSM) (p. 285).
2016.11.09	<p>All AMIs</p> <ul style="list-style-type: none">• Released AWS PV driver, version 7.4.3.0 for Windows 2008 R2 and newer• Windows Server 2016 added to patch release cycle.• Microsoft security updates current to November 8 2016.• Current AWS Tools for Windows PowerShell
2016.10.18	Released new AMIs for Windows Server 2016. AMIs that use Windows Server 2016 include significant changes. For example, these AMIs don't include the EC2Config service and you can't connect to Windows Server 2016 Nano Server by using Remote Desktop. You must remotely administer Nano Server by using Windows PowerShell. Before you use a Windows Server 2016 AMI, read about all of the changes and how to work

Release	Changes
	<p>with these AMIs. For more information, see Changes in Windows Server 2016 AMIs (p. 108).</p> <p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to October 12, 2016. • Current AWS Tools for Windows PowerShell
2016.9.14	<p>ALL AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to September 13, 2016. • Current AWS Tools for Windows PowerShell • Renamed AMI: Windows_Server-2012-RTM-Japanese-64Bit-SQL_2008_R3_SP2_Standard to Windows_Server-2012-RTM-Japanese-64Bit-SQL_2008_R2_SP3_Standard
2016.8.26	<p>All Windows Server 2008 R2 AMIs dated 2016.08.11 were updated to fix a known issue. New AMIs are dated 2016.08.25.</p>
2016.8.11	<p>ALL AMIs</p> <ul style="list-style-type: none"> • Ec2Config v3.19.1153 • Microsoft security updates current to August 10, 2016. • Enabled the registry key User32 exception handler hardening feature in Internet Explorer for MS15-124. <p>Server 2008 R2, Server 2012 RTM, and Server 2012 R2 AMIs</p> <ul style="list-style-type: none"> • Elastic Network Adapter (ENA) Driver 1.0.8.0 • ENA AMI property set to enabled. <p>Note AWS PV Driver for Windows Server 2008 R2 was re-released this month because of a known issue. Windows Server 2008 R2 AMI's were removed in July because of this issue.</p>
2016.8.2	<p>Windows Server 2008 R2 AMIs</p> <p>All Windows Server 2008 R2 AMIs for July were removed and rolled back to AMIs dated 2016.06.15, because of an issue discovered in the AWS PV driver. The AWS PV driver issue has been fixed. The August AMI release will include Windows Server 2008 R2 AMIs with the fixed AWS PV driver and July/August Windows updates.</p>
2016.7.26	<p>ALL AMIs</p> <ul style="list-style-type: none"> • Ec2Config v3.18.1118 • Microsoft security updates current to July 2016. <p>2016.07.13 AMIs were missing security patches. AMIs were re-patched. Additional processes were put in place to verify successful patch installations going forward.</p>

Amazon Elastic Compute Cloud
User Guide for Windows Instances
Details About AWS Windows AMI Versions

Release	Changes
2016.7.13	<p>ALL AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to July 2016 • Current AWS Tools for Windows PowerShell • Updated AWS PV Driver 7.4.2.0 • AWS PV Driver for Windows Server 2008 R2
2016.6.16	<ul style="list-style-type: none"> • Microsoft security updates current to June 2016 • Current AWS Tools for Windows PowerShell • EC2Config service version 3.17.1032 • Released 10 new AMIs that include 64-bit versions of Microsoft SQL Server 2016. You can launch an instance from one of these AMIs from the EC2 console, CLI, or API. If using the console, navigate to EC2 > Images > AMIs, choose Public Images, and enter "Windows_Server-2012-R2_RTM-English-64Bit-SQL_2016_Standard" in the search bar. For more information about SQL Server 2016, see What's New in SQL Server 2016 on MSDN.
2016.5.11	<p>ALL AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to May 2016 • Current AWS Tools for Windows PowerShell • EC2Config service version 3.16.930 • MS15-011 Active Directory patch installed • Intel SRIOV driver for Windows Server 2012 R2 based AMIs. Version 1.0.16.1 (03/04/2014)
2016.4.13	<p>ALL AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to April 2016 • Current AWS Tools for Windows PowerShell • EC2Config service version 3.15.880
2016.3.9	<p>ALL AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to March 2016 • Current AWS Tools for Windows PowerShell • EC2Config service version 3.14.786
2016.2.10	<p>ALL AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to February 2016 • Current AWS Tools for Windows PowerShell • EC2Config service version 3.13.727
2016.1.25	<p>ALL AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to January 2016 • Current AWS Tools for Windows PowerShell • EC2Config service version 3.12.649

Release	Changes
2016.1.5	ALL AMIs <ul style="list-style-type: none"> • Current AWS Tools for Windows PowerShell

AMIs Released in 2015

Release	Changes
2015.12.15	ALL AMIs <ul style="list-style-type: none"> • Microsoft security updates current to December 2015 • Current AWS Tools for Windows PowerShell
2015.11.11	ALL AMIs <ul style="list-style-type: none"> • Microsoft security updates current to November 2015 • Current AWS Tools for Windows PowerShell • EC2Config service version 3.11.521 • CFN Agent updated to latest version
2015.10.26	Corrected boot volume sizes of base AMIs to be 30GB instead of 35GB
2015.10.14	ALL AMIs <ul style="list-style-type: none"> • Microsoft security updates current to October 2015 • EC2Config service version 3.10.442 • Current AWS Tools for Windows PowerShell • Updated SQL Service Packs to latest versions for all SQL variants • Removed old entries in Event Logs • AMI Names have been changed to reflect the latest service pack. For example, the latest AMI with Server 2012 and SQL 2014 Standard is named "Windows_Server-2012-RTM-English-64Bit-SQL_2014_SP1_Standard-2015.10.26", not "Windows_Server-2012-RTM-English-64Bit-SQL_2014_RTM_Standard-2015.10.26".
2015.9.9	ALL AMIs <ul style="list-style-type: none"> • Microsoft security updates current to September 2015 • EC2Config service version 3.9.359 • Current AWS Tools for Windows PowerShell • Current AWS CloudFormation helper scripts

Release	Changes
2015.8.18	<p>ALL AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to August 2015 • EC2Config service version 3.8.294 • Current AWS Tools for Windows PowerShell <p>Only AMIs with Windows Server 2012 and Windows Server 2012 R2</p> <ul style="list-style-type: none"> • AWS PV Driver 7.3.2
2015.7.21	<p>ALL AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to July 2015 • EC2Config service version 3.7.308 • Current AWS Tools for Windows PowerShell • Modified AMI descriptions of SQL images for consistency
2015.6.10	<p>ALL AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to June 2015 • EC2Config service version 3.6.269 • Current AWS Tools for Windows PowerShell • Current AWS CloudFormation helper scripts <p>Only AMIs with Windows Server 2012 R2</p> <ul style="list-style-type: none"> • AWS PV Driver 7.3.1
2015.5.13	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to May 2015 • EC2Config service version 3.5.228 • Current AWS Tools for Windows PowerShell
2015.04.15	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to April 2015 • EC2Config service version 3.3.174 • Current AWS Tools for Windows PowerShell
2015.03.11	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to March 2015 • EC2Config service version 3.2.97 • Current AWS Tools for Windows PowerShell <p>Only AMIs with Windows Server 2012 R2</p> <ul style="list-style-type: none"> • AWS PV Driver 7.3.0

Release	Changes
2015.02.11	All AMIs <ul style="list-style-type: none">• Microsoft security updates current to February 2015• EC2Config service version 3.0.54• Current AWS Tools for Windows PowerShell• Current AWS CloudFormation helper scripts
2015.01.14	All AMIs <ul style="list-style-type: none">• Microsoft security updates current to January 2015• EC2Config service version 2.3.313• Current AWS Tools for Windows PowerShell• Current AWS CloudFormation helper scripts

AMIs Released in 2014

Release	Changes
2014.12.10	All AMIs <ul style="list-style-type: none">• Microsoft security updates current to December 2014• EC2Config service version 2.2.12• Current AWS Tools for Windows PowerShell
2014.11.19	All AMIs <ul style="list-style-type: none">• Microsoft security updates current to November 2014• EC2Config service version 2.2.11• Current AWS Tools for Windows PowerShell
2014.10.15	All AMIs <ul style="list-style-type: none">• Microsoft security updates current to October 2014• EC2Config service version 2.2.10• Current AWS Tools for Windows PowerShell Only AMIs with Windows Server 2012 R2 <ul style="list-style-type: none">• AWS PV Driver 7.2.4.1 (resolves the issues with Plug and Play Cleanup, which is now enabled by default)

Release	Changes
2014.09.10	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to September 2014 • EC2Config service version 2.2.8 • Current AWS Tools for Windows PowerShell <p>Only AMIs with Windows Server 2012 R2</p> <ul style="list-style-type: none"> • Disable Plug and Play Cleanup (see Important information) • AWS PV Driver 7.2.2.1 (resolves issues with the uninstaller)
2014.08.13	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to August 2014 • EC2Config service version 2.2.7 • Current AWS Tools for Windows PowerShell <p>Only AMIs with Windows Server 2012 R2</p> <ul style="list-style-type: none"> • AWS PV Driver 7.2.2.1 (improves disk performance, resolves issues with reconnecting multiple network interfaces and lost network settings)
2014.07.10	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to July 2014 • EC2Config service version 2.2.5 • Current AWS Tools for Windows PowerShell
2014.06.12	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to June 2014 • EC2Config service version 2.2.4 • Removed NVIDIA drivers (except for Windows Server 2012 R2 AMIs) • Current AWS Tools for Windows PowerShell
2014.05.14	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to May 2014 • EC2Config service version 2.2.2 • Current AWS Tools for Windows PowerShell • AWS CloudFormation helper scripts version 1.4.0
2014.04.09	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to April 2014 • Current AWS Tools for Windows PowerShell • Current AWS CloudFormation helper scripts
2014.03.12	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to March 2014

Release	Changes
2014.02.12	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to February 2014 • EC2Config service version 2.2.1 • Current AWS Tools for Windows PowerShell • KB2634328 • Remove the BCDEdit useplatformclock value <p>Only AMIs with Microsoft SQL Server</p> <ul style="list-style-type: none"> • Microsoft SQL Server 2012 SP1 cumulative update package 8 • Microsoft SQL Server 2008 R2 cumulative update package 10

AMIs Released in 2013

Release	Changes
2013.11.13	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to November 2013 • EC2Config service version 2.1.19 • Current AWS Tools for Windows PowerShell • Configure NTP to synchronize the time once a day (the default is every seven days) <p>Only AMIs with Windows Server 2012</p> <ul style="list-style-type: none"> • Clean up the WinSXS folder using the following command: <code>dism /online /cleanup-image /StartComponentCleanup</code>
2013.09.11	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to September 2013 • EC2Config service version 2.1.18 • Current AWS Tools for Windows PowerShell • AWS CloudFormation helper scripts version 1.3.15
2013.07.10	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to July 2013 • EC2Config service version 2.1.16 • Expanded the root volume to 50 GB • Set the page file to 512 MB, expanding to 8 GB as needed • Current AWS Tools for Windows PowerShell

Release	Changes
2013.06.12	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to June 2013 • Current AWS Tools for Windows PowerShell <p>Only AMIs with Microsoft SQL Server</p> <ul style="list-style-type: none"> • Microsoft SQL Server 2012 SP1 with cumulative update package 4
2013.05.15	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to May 2013 • EC2Config service version 2.1.15 • All instance store volumes attached by default • Remote PowerShell enabled by default • Current AWS Tools for Windows PowerShell
2013.04.14	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to April 2013 • Current AWS Tools for Windows PowerShell • AWS CloudFormation helper scripts version 1.3.14
2013.03.14	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to March 2013 • EC2Config service version 2.1.14 • Citrix Agent with CPU heartbeat fix • Current AWS Tools for Windows PowerShell • AWS CloudFormation helper scripts version 1.3.11
2013.02.22	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to February 2013 • KB2800213 • Windows PowerShell 3.0 upgrade • EC2Config service version 2.1.13 • Citrix Agent with time fix • Citrix PV drivers dated 2011.07.19 • Current AWS Tools for Windows PowerShell • AWS CloudFormation helper scripts version 1.3.8 <p>Only AMIs with Microsoft SQL Server</p> <ul style="list-style-type: none"> • Microsoft SQL Server 2012 cumulative update package 5

AMIs Released in 2012

Release	Changes
2012.12.12	All AMIs <ul style="list-style-type: none">• Microsoft security updates current to December 2012• Set the ActiveTimeBias registry value to 0• Disable IPv6 for the network adapter• EC2Config service version 2.1.9• Add AWS Tools for Windows PowerShell and set the policy to allow import-module
2012.11.15	All AMIs <ul style="list-style-type: none">• Microsoft security updates current to November 2012• EC2Config service version 2.1.7
2012.10.10	All AMIs <ul style="list-style-type: none">• Microsoft security updates current to October 2012
2012.08.15	All AMIs <ul style="list-style-type: none">• Microsoft security updates current to August 2012• EC2Config service version 2.1.2• KB2545227
2012.07.11	All AMIs <ul style="list-style-type: none">• Microsoft security updates current to July 2012
2012.06.12	All AMIs <ul style="list-style-type: none">• Microsoft security updates current to June 2012• Set page file to 4 GB• Remove installed language packs• Set performance option to "Adjust for best performance"• Set the screen saver to no longer display the logon screen on resume• Remove previous RedHat driver versions using pnputil• Remove duplicate bootloaders and set bootstatuspolicy to ignoreallfailures using bcdedit
2012.05.10	All AMIs <ul style="list-style-type: none">• Microsoft security updates current to May 2012• EC2Config service version 2.1.0
2012.04.11	All AMIs <ul style="list-style-type: none">• Microsoft security updates current to April 2012• KB2582281• Current version of EC2Config• System time in UTC instead of GMT

Release	Changes
2012.03.13	All AMIs <ul style="list-style-type: none"> Microsoft security updates current to March 2012
2012.02.24	All AMIs <ul style="list-style-type: none"> Microsoft security updates current to February 2012 Standardize AMI names and descriptions
2012.01.12	All AMIs <ul style="list-style-type: none"> Microsoft security updates current to January 2012 RedHat PV driver version 1.3.10

AMIs Released in 2011 and earlier

Release	Changes
2011.09.11	All AMIs <ul style="list-style-type: none"> Microsoft security updates current to September 2011
1.04	All AMIs <ul style="list-style-type: none"> Current Microsoft security updates Update network driver Fix issue with instances in a VPC losing connectivity when changing the time zone of the instance
1.02	All AMIs <ul style="list-style-type: none"> Current Microsoft security updates Update network driver Add support for licensing activation for instances in a VPC
1.01	All AMIs <ul style="list-style-type: none"> Current Microsoft security updates Fix issue with password improperly generated while waiting for network availability
1.0	All AMIs <ul style="list-style-type: none"> Initial release

Changes in Windows Server 2016 AMIs

AWS provides AMIs for Windows Server 2016. These AMIs include the following high-level changes from earlier Windows AMIs.

- To accommodate the change from .NET Framework to .NET Core, the EC2Config service has been deprecated on Windows Server 2016 AMIs and replaced by EC2Launch. EC2Launch is a bundle of

Windows PowerShell scripts that perform many of the tasks performed by the EC2Config service. For more information, see [Configuring a Windows Instance Using EC2Launch](#) (p. 319).

- The Windows Server 2016 Nano Server installation option (Nano Server) does not support Remote Desktop connections. The **Connection** option is available in the EC2 console, but the connection fails. You must remotely connect to your instance using Windows PowerShell. For more information, see [Connect to a Windows Server 2016 Nano Server Instance](#) (p. 257).
- On earlier versions of Windows Server AMIs, you can use the EC2Config service to join an EC2 instance to a domain and configure integration with Amazon CloudWatch. On Windows Server 2016 AMIs, the Amazon EC2 Simple Systems Manager (SSM) agent performs these tasks. This means that you must use either Amazon EC2 Run Command or SSM Config to join an EC2 instance to a domain or configure integration with Amazon CloudWatch on Windows Server 2016 instances. For more information, see the following topics.

Run Command

- [Joining EC2 Instances to a Domain Using Amazon EC2 Run Command](#) (p. 451)
- [Uploading Logs from EC2 Instances to Amazon CloudWatch Using Amazon EC2 Run Command](#) (p. 454)

SSM Config

- [Joining a Windows Instance to an AWS Directory Service Domain](#) (p. 331)
- [Sending Performance Counters to CloudWatch and Logs to CloudWatch Logs Using Amazon EC2 Simple Systems Manager](#) (p. 340)

Other Differences

Note these additional important differences for instances created from Windows Server 2016 AMIs.

- By default, EC2Launch does not initialize secondary EBS volumes. You can configure EC2Launch to initialize disks automatically by either scheduling the script to run or by calling EC2Launch in user data. For the procedure to initialize disks using EC2Launch, see "Initialize Drives and Drive Letter Mappings" in [Configuring EC2Launch](#) (p. 320).
- Nano Server does not support online domain joining. You must perform an offline domain join instead. For more information, see [Offline Domain Join \(Djoin.exe\) Step-by-Step Guide](#) on Microsoft TechNet.
- If you previously enabled CloudWatch integration on your instances by using a local configuration file (AWS.EC2.Windows.CloudWatch.json), you can configure the file to work with the SSM agent on instances created from Windows Server 2016 AMIs. For more information, see [Using a Local Configuration File for CloudWatch Integration on Windows Server 2016 Instances](#) (p. 109).

For more information about Windows Server 2016, see [What's New with Windows Server 2016](#) and [Getting Started with Nano Server](#) on Microsoft.com.

Contents

- [Using a Local Configuration File for CloudWatch Integration on Windows Server 2016 Instances](#) (p. 109)
- [Docker Container Conflict on Windows Server 2016 Instances](#) (p. 110)

Using a Local Configuration File for CloudWatch Integration on Windows Server 2016 Instances

If you previously enabled Amazon CloudWatch integration on your instances by using a local configuration file (AWS.EC2.Windows.CloudWatch.json), you can use the following procedure to configure the file to work with the SSM agent on Windows Server 2016 instances.

If you did not previously configure CloudWatch integration using a local configuration file, you can do this on Windows Server 2016. Create a local configuration file using the instructions for the EC2Config service and then return to this procedure to make the final changes for Windows Server 2016. For more information, see [Sending Performance Counters to CloudWatch and Logs to CloudWatch Logs Using EC2Config](#) (p. 307).

1. Locate the `AWS.EC2.Windows.CloudWatch.json` file on your earlier instance. The file is located in the following directory:

```
C:\Program Files\Amazon\SSM\Plugins\awsCloudWatch
```

2. Add the `IsEnabled` section to your existing file. The `IsEnabled` section must be located on the same level as the `EngineConfiguration` section. The following example illustrates this:

```
{  
  "IsEnabled": true,  
  "EngineConfiguration": {  
    "PollInterval": "00:00:15",  
    "Components": [  
      {  
        "Id": "OsCpuUtilization",  
  
        "FullName": "AWS.EC2.Windows.CloudWatch.PerformanceCounterComponent.PerformanceCounterIn  
        "Parameters": {  
          "CategoryName": "Process",  
  
[Sample JSON truncated]
```

3. Save the file with the same name in the following folder on your Windows Server 2016 instance: `C:\Program Files\Amazon\SSM\Plugins\awsCloudWatch\`
4. Start or restart the SSM agent (`AmazonSSMAgent.exe`) using the Windows Services control panel or by sending the following command in PowerShell:

```
Restart-Service AmazonSSMAgent.exe
```

After the SSM agent restarts, it detects the local configuration file and configures the instance for CloudWatch integration. If you change parameters and settings in the local configuration file, you need to restart the SSM agent to pick up the changes. If you want to disable CloudWatch integration on the instance, change `IsEnabled` to `false` and save your changes in the configuration file.

Docker Container Conflict on Windows Server 2016 Instances

If you run the Docker service on Windows Server 2016 AMIs, the service is configured to use a different CIDR value than the default internal IP address prefix value. The default value is `172.16.0.0/12`. Windows Server 2016 AMIs use `172.17.0.0/16` to avoid a conflict with the default Amazon EC2 VPC/subnet. If you don't change VPC/subnet settings for your EC2 instances, then you don't need to do anything. The conflict is essentially avoided because of the different CIDR values. If you do change VPC/subnet settings, be aware of these internal IP address prefix values and avoid creating a conflict. For more information, read the following section.

Important

If you plan to run Docker on a Windows Server 2016 instance, you must create the instance from the following Amazon Machine Image (AMI) or an AMI based on this image:

```
"Windows_Server-2016-English-Full-Containers-2016.10.18"
```

If you create the instance from another Windows Server 2016 AMI, instances fail to boot correctly after installing Docker and then running Sysprep.

Technical Details About the Conflict

In the networking context, Windows containers function like virtual machines. Each container has a virtual network adapter that is connected to a virtual switch. Inbound and outbound traffic is forwarded over this switch. Windows Server containers use a host virtual network interface controller (vNIC) to attach to the virtual switch.

When the Docker service starts for the first time on Windows Server 2016, the Docker engine creates a network address translation (NAT) network. By default, all container endpoints are connected to the default NAT network. The Docker internal IP address prefix is 172.16.0.0/12. If the container host IP address is in this same prefix, then NAT network creation fails because of the conflict between overlapping IP address spaces.

On Amazon EC2, default VPCs are assigned a CIDR range of 172.31.0.0/16. Default subnets within a default VPC are assigned /20 netblocks within the VPC CIDR range. There is an address space overlap between the default Amazon EC2 VPC and the default internal prefix used by Docker. Therefore, AWS embeds a new CIDR value of 172.17.0.0/16 in the Docker config file `daemon.json`. This file is located in the following directory: `C:\ProgramData\Docker\config\daemon.json`. The `daemon.json` file uses the `fixed-cidr: < IP Prefix > / Mask` option to create the default NAT network with the IP address prefix and mask specified, thereby avoiding any address space conflicts. If you change your VPC and subnet settings, you must stop the Docker service, update the `daemon.json` file with the new CIDR range, and restart the service.

Create a Standard Amazon Machine Image Using Sysprep

The Microsoft System Preparation (Sysprep) tool simplifies the process of duplicating a customized installation of Windows. We recommend that you use Sysprep to create a standardized Amazon Machine Image (AMI). You can then create new Amazon EC2 instances for Windows from this standardized image and deploy these across your organization.

We also recommend that you run Sysprep with the EC2Config service, which automates and secures the image-preparation process on your AMI by using an answer file. The file is located in the following directory, by default: `C:\Program Files\Amazon\Ec2ConfigService\sysprep2008.xml`

Important

- The EC2Config service is replaced by EC2Launch on Windows Server 2016. You can use Sysprep with EC2Launch on the full version of Windows Server 2016 (with a desktop experience). Sysprep with EC2Launch is not supported on the Nano installation of Windows Server 2016. For more information, see [Using Sysprep with EC2Launch \(p. 322\)](#).
- Don't use Sysprep to create an instance backup. Sysprep removes system-specific information; removing this information might have unintended consequences for an instance backup.

Contents

- [Before You Begin \(p. 111\)](#)
- [Using Sysprep with the EC2Config Service \(p. 112\)](#)
- [Run Sysprep with the EC2Config Service \(p. 115\)](#)
- [Troubleshooting Sysprep with EC2Config \(p. 116\)](#)

Before You Begin

- Learn more about [Sysprep](#) on Microsoft TechNet.

- Learn which [server roles are supported for Sysprep](#).

Using Sysprep with the EC2Config Service

Learn the details of the different Sysprep execution phases and the tasks performed by the EC2Config service as the image is prepared.

Sysprep Phases

Sysprep runs through the following phases:

1. **Generalize:** The tool removes image-specific information and configurations. For example, Sysprep removes the security identifier (SID), the computer name, the event logs, and specific drivers, to name a few. After this phase is completed, the operating system (OS) is ready to create an AMI.

Note

When you run Sysprep with the EC2Config service, the system prevents drivers from being removed because the `PersistAllDeviceInstalls` setting is set to true by default.

2. **Specialize:** Plug and Play scans the computer and installs drivers for any detected devices. The tool generates OS requirements like the computer name and SID. Optionally, you can execute commands in this phase.
3. **Out-of-Box Experience (OOBE):** The system runs an abbreviated version of Windows Setup and asks the user to enter information such as a system language, the time zone, and a registered organization. When you run Sysprep with EC2Config, the answer file automates this phase.

Sysprep Actions

Sysprep and the EC2Config service perform the following actions when preparing an image.

1. When you choose **Shutdown with Sysprep** in the **EC2 Service Properties** dialog box, the system runs the `ec2config.exe –sysprep` command.
2. The EC2Config service reads the content of the `BundleConfig.xml` file. This file is located in the following directory, by default: `C:\Program Files\Amazon\Ec2ConfigService\Settings`.

The `BundleConfig.xml` file includes the following settings. You can change these settings:

- **AutoSysprep:** Indicates whether to use Sysprep automatically. You do not need to change this value if you are running Sysprep from the EC2 Service Properties dialog box. The default value is No.
 - **SetRDPCertificate:** Sets a self-signed certificate for the Remote Desktop server running on Windows Server 2003. This enables you to securely use the Remote Desktop Protocol (RDP) to connect to the instance. Change the value to Yes if new instances should use a certificate. This setting is not used with Windows Server 2008 or Windows Server 2012 instances because these operating systems can generate their own certificates. The default value is No.
 - **SetPasswordAfterSysprep:** Sets a random password on a newly launched instance, encrypts it with the user launch key, and outputs the encrypted password to the console. Change the value to No if new instances should not be set to a random encrypted password. The default value is Yes.
 - **PreSysprepRunCmd:** The location of the command to run. The command is located in the following directory, by default: `C:\Program Files\Amazon\Ec2ConfigService\Scripts\BeforeSysprep.cmd`
3. The system executes the `BeforeSysprep.cmd`. This command creates the following registry key:

```
reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /v  
fDenyTSConnections /t REG_DWORD /d 1 /f"
```

The registry key disables RDP connections until they are re-enabled. Disabling RDP connections is a necessary security measure because, during the first boot session after Sysprep has run, there is a short period of time where RDP allows connections and the Administrator password is blank.

4. The EC2Config service calls sysprep.exe by executing the following command:

```
sysprep.exe /unattend: "C:\Program Files\Amazon\Ec2ConfigService\sysprep2008.xml" /  
oobe /generalize /shutdown
```

Generalize Phase

1. The tool removes image-specific information and configurations such as the computer name and the SID. If the instance is a member of a domain, it is removed from the domain. The sysprep2008.xml answer file includes the following settings which affect this phase:
 - **PersistAllDeviceInstalls:** This setting prevents Windows Setup from removing and reconfiguring devices, which speeds up the image preparation process because Amazon AMIs require certain drivers to run and re-detection of those drivers would take time.
 - **DoNotCleanUpNonPresentDevices:** This setting retains Plug and Play information for devices that are not currently present.
2. Sysprep.exe shuts down the OS as it prepares to create the AMI. The system either launches a new instance or starts the original instance.

Specialize Phase

The system generates OS specific requirements such as a computer name and a SID. The system also performs the following actions based on configurations that you specify in the sysprep2008.xml answer file.

- **CopyProfile:** Sysprep can be configured to delete all user profiles, including the built-in Administrator profile. This setting retains the built-in Administrator account so that any customizations you made to that account are carried over to the new image. The default value is True.

If you don't have specific user-profile customizations that you want to carry over to the new image then change this setting to False. Sysprep will remove all user profiles; this saves time and disk space.

- **TimeZone:** The time zone is set to Coordinate Universal Time (UTC) by default.
- **Synchronous command with order 1:** The system executes the following command that enables the administrator account and specifies the password requirement.

```
net user Administrator /ACTIVE:YES /LOGONPASSWORDCHG:NO /EXPIRES:NEVER /  
PASSWORDREQ:YES
```

- **Synchronous command with order 2:** The system scrambles the administrator password. This security measure is designed to prevent the instance from being accessible after Sysprep completes if you did not enable the ec2setpassword setting.

```
C:\Program Files\Amazon\Ec2ConfigService\ScramblePassword.exe" -u Administrator
```

- **Synchronous command with order 3:** The system executes the following command:

```
C:\Program Files\Amazon\Ec2ConfigService\Scripts\SysprepSpecializePhase.cmd
```

This command adds the following registry key, which re-enables RDP:

```
reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /v  
fDenyTSConnections /t REG_DWORD /d 0 /f
```


OOBE Phase

1. Using the EC2Config service answer file, the system specifies the following configurations:

- <InputLocale>en-US</InputLocale>
- <SystemLocale>en-US</SystemLocale>
- <UILanguage>en-US</UILanguage>
- <UserLocale>en-US</UserLocale>
- <HideEULAPage>true</HideEULAPage>
- <HideWirelessSetupInOOBE>true</HideWirelessSetupInOOBE>
- <NetworkLocation>Other</NetworkLocation>
- <ProtectYourPC>3</ProtectYourPC>
- <BluetoothTaskbarIconEnabled>false</BluetoothTaskbarIconEnabled>
- <TimeZone>UTC</TimeZone>
- <RegisteredOrganization>Amazon.com</RegisteredOrganization>
- <RegisteredOwner>Amazon</RegisteredOwner>

Note

During the generalize and specialize phases the EC2Config service monitors the status of the OS. If EC2Config detects that the OS is in a Sysprep phase, then it publishes the following message to the system log:

```
"EC2ConfigMonitorState: 0 Windows is being configured.  
SysprepState=IMAGE_STATE_UNDEPLOYABLE"
```

2. After the OOBE phase completes, the system executes the SetupComplete.cmd from the following location: C:\Windows\Setup\Scripts\SetupComplete.cmd. In Amazon public AMIs before April 2015 this file was empty and executed nothing on the image. In public AMIs dated after April 2015, the file includes the following value: **call "C:\Program Files\Amazon\Ec2ConfigService\Scripts\PostSysprep.cmd"**.
3. The system executes the PostSysprep.cmd, which performs the following operations:
- Sets the local Administrator password to not expire. If the password expired, Administrators might not be able to log on.
 - Sets the MSSQLServer machine name (if installed) so that the name will be in sync with the AMI.

Post Sysprep

After Sysprep completes, the EC2Config services sends the following message to the console output:
"Windows sysprep configuration complete. Message: Sysprep Start Message:
Sysprep End"

EC2Config then performs the following actions:

1. Reads the content of the config.xml file and lists all enabled plug-ins.
2. Executes all "Before Windows is ready" plug-ins at the same time.
 - Ec2SetPassword
 - Ec2SetComputerName
 - Ec2InitializeDrives
 - Ec2EventLog
 - Ec2ConfigureRDP
 - Ec2OutputRDPcert
 - Ec2SetDriveLetter
 - Ec2WindowsActivate
 - Ec2DynamicBootVolumeSize

3. After it is finished, sends a “Windows is ready” message to the instance system logs.
4. Runs all “After Windows is ready” plug-ins at the same time.
 - AWS CloudWatch logs
 - UserData
 - Simple Systems Manager (SSM)

For more information about Windows plug-ins, see [Configuring a Windows Instance Using the EC2Config Service](#).

Run Sysprep with the EC2Config Service

Use the following procedure to create a standardized AMI using Sysprep and the EC2Config service.

1. In the Amazon EC2 console locate or [create](#) an AMI that you want to duplicate.
2. Launch and connect to your Windows instance.
3. Customize it.
4. Specify configuration settings in the EC2Config service answer file:

```
C:\Program Files\Amazon\Ec2ConfigService\syprep2008.xml
```
5. From the Windows **Start** menu, choose **All Programs**, and then choose **EC2ConfigService Settings**.
6. Choose the **Image** tab in the **Ec2 Service Properties** dialog box. For more information about the options and settings in the Ec2 Service Properties dialog box, see [Ec2 Service Properties](#).
7. Select an option for the Administrator password, and then click **Shutdown with Sysprep** or **Shutdown without Sysprep**. EC2Config edits the settings files based on the password option that you selected.
 - **Random**: EC2Config generates a password, encrypts it with user's key, and displays the encrypted password to the console. We disable this setting after the first launch so that this password persists if the instance is rebooted or stopped and started.
 - **Specify**: The password is stored in the Sysprep answer file in unencrypted form (clear text). When Sysprep runs next, it sets the Administrator password. If you shut down now, the password is set immediately. When the service starts again, the Administrator password is removed. It's important to remember this password, as you can't retrieve it later.
 - **Keep Existing**: The existing password for the Administrator account doesn't change when Sysprep is run or EC2Config is restarted. It's important to remember this password, as you can't retrieve it later.
8. Choose **OK**.

When you are asked to confirm that you want to run Sysprep and shut down the instance, click **Yes**. You'll notice that EC2Config runs Sysprep. Next, you are logged off the instance, and the instance is shut down. If you check the **Instances** page in the Amazon EC2 console, the instance state changes from *running* to *stopping*, and then finally to *stopped*. At this point, it's safe to create an AMI from this instance.

You can manually invoke the Sysprep tool from the command line using the following command:

```
C:\> %ProgramFiles%\Amazon\Ec2ConfigService\ec2config.exe -sysprep
```

However, you must be very careful that the XML file options specified in the `Ec2ConfigService\Settings` folder are correct; otherwise, you might not be able to connect to the instance. For more information about the settings files, see [EC2Config Settings Files \(p. 289\)](#). For an example of

configuring and then running Sysprep from the command line, see `Ec2ConfigService\Scripts\InstallUpdates.ps1`.

Troubleshooting Sysprep with EC2Config

If you experience problems or receive error messages during image preparations, review the following logs:

- `%WINDIR%\Panther\Unattendgc`
- `%WINDIR%\System32\Sysprep\Panther`
- `"C:\Program Files\Amazon\Ec2ConfigService\Log\Ec2ConfigLog.txt"`

If you receive an error message during image preparation with Sysprep, the OS might not be reachable. To review the log files, you must stop the instance, attach its root volume to another healthy instance as a secondary volume, and then review the logs mentioned earlier on the secondary volume.

If you locate errors in the Unattendgc log file, use the [Microsoft Error Lookup Tool](#) to get more details about the error. The following issue reported in the Unattendgc log file is typically the result of one or more corrupted user profiles on the instance:

```
Error [Shell Unattend] _FindLatestProfile failed (0x80070003)
[gle=0x00000003] Error [Shell Unattend] CopyProfile failed (0x80070003)
[gle=0x00000003]
```

There are two options for resolving this issue:

Option 1: Use Regedit on the instance to search for the following key. Verify that there are no profile registry keys for a deleted user:

```
[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\ProfileList\
```

Option 2: Edit the EC2Config answer file (`C:\Program Files\Amazon\Ec2ConfigService\sysprep2008.xml`) and change `<CopyProfile>true</CopyProfile>` to `<CopyProfile>>false</CopyProfile>`. Run Sysprep again. Note that this configuration change will delete the built-in administrator user profile after Sysprep completes.

Amazon EC2 Instances

If you're new to Amazon EC2, see the following topics to get started:

- [What Is Amazon EC2? \(p. 1\)](#)
- [Setting Up with Amazon EC2 \(p. 13\)](#)
- [Getting Started with Amazon EC2 Windows Instances \(p. 20\)](#)
- [Instance Lifecycle \(p. 241\)](#)

Before you launch a production environment, you need to answer the following questions.

Q. What instance type best meets my needs?

Amazon EC2 provides different instance types to enable you to choose the CPU, memory, storage, and networking capacity that you need to run your applications. For more information, see [Instance Types \(p. 117\)](#).

Q. What purchasing option best meets my needs?

Amazon EC2 supports On-Demand instances (the default), Spot instances, and Reserved Instances. For more information, see [Instance Purchasing Options \(p. 150\)](#).

Q. Which type of root volume meets my needs?

Each instance is backed by Amazon EBS or backed by instance store. Select an AMI based on which type of root volume you need. For more information, see [Storage for the Root Device \(p. 64\)](#).

Q. Would I benefit from using a virtual private cloud?

If you can launch instances in either EC2-Classic or EC2-VPC, you'll need to decide which platform meets your needs. For more information, see [Supported Platforms \(p. 672\)](#) and [Amazon EC2 and Amazon Virtual Private Cloud \(p. 665\)](#).

Q. Can I remotely manage a fleet of EC2 instances *and* machines in my hybrid environment?

Amazon Elastic Compute Cloud (Amazon EC2) Run Command lets you remotely and securely manage the configuration of your Amazon EC2 instances, virtual machines (VMs) and servers in hybrid environments, or VMs from other cloud providers. For more information, see [Remote Management \(p. 437\)](#).

Instance Types

When you launch an instance, the *instance type* that you specify determines the hardware of the host computer used for your instance. Each instance type offers different compute, memory, and storage

capabilities and are grouped in instance families based on these capabilities. Select an instance type based on the requirements of the application or software that you plan to run on your instance.

Amazon EC2 provides each instance with a consistent and predictable amount of CPU capacity, regardless of its underlying hardware.

Amazon EC2 dedicates some resources of the host computer, such as CPU, memory, and instance storage, to a particular instance. Amazon EC2 shares other resources of the host computer, such as the network and the disk subsystem, among instances. If each instance on a host computer tries to use as much of one of these shared resources as possible, each receives an equal share of that resource. However, when a resource is under-utilized, an instance can consume a higher share of that resource while it's available.

Each instance type provides higher or lower minimum performance from a shared resource. For example, instance types with high I/O performance have a larger allocation of shared resources. Allocating a larger share of shared resources also reduces the variance of I/O performance. For most applications, moderate I/O performance is more than enough. However, for applications that require greater or more consistent I/O performance, consider an instance type with higher I/O performance.

Contents

- [Available Instance Types \(p. 118\)](#)
- [Hardware Specifications \(p. 119\)](#)
- [Networking and Storage Features \(p. 119\)](#)
- [Instance Limits \(p. 121\)](#)

Available Instance Types

Amazon EC2 provides the instance types listed in the following tables.

Current Generation Instances

For the best performance, we recommend that you use the current generation instance types when you launch new instances. For more information about the current generation instance types, see [Amazon EC2 Instances](#).

Instance Family	Current Generation Instance Types
General purpose	t2.nano t2.micro t2.small t2.medium t2.large t2.xlarge t2.2xlarge m4.large m4.xlarge m4.2xlarge m4.4xlarge m4.10xlarge m4.16xlarge m3.medium m3.large m3.xlarge m3.2xlarge
Compute optimized	c4.large c4.xlarge c4.2xlarge c4.4xlarge c4.8xlarge c3.large c3.xlarge c3.2xlarge c3.4xlarge c3.8xlarge
Memory optimized	r3.large r3.xlarge r3.2xlarge r3.4xlarge r3.8xlarge r4.large r4.xlarge r4.2xlarge r4.4xlarge r4.8xlarge r4.16xlarge x1.16xlarge x1.32xlarge
Storage optimized	i2.xlarge i2.2xlarge i2.4xlarge i2.8xlarge d2.xlarge d2.2xlarge d2.4xlarge d2.8xlarge
Accelerated computing	p2.xlarge p2.8xlarge p2.16xlarge g2.2xlarge g2.8xlarge

Previous Generation Instances

Amazon Web Services offers previous generation instances for users who have optimized their applications around these instances and have yet to upgrade. We encourage you to use the latest generation of instances to get the best performance, but we will continue to support these previous generation instances. If you are currently using a previous generation instance, you can see which current generation instance would be a suitable upgrade. For more information, see [Previous Generation Instances](#).

Instance Family	Previous Generation Instance Types
General purpose	m1.small m1.medium m1.large m1.xlarge
Compute optimized	c1.medium c1.xlarge cc2.8xlarge
Memory optimized	m2.xlarge m2.2xlarge m2.4xlarge cr1.8xlarge
Storage optimized	hi1.4xlarge hs1.8xlarge
Accelerated computing	cg1.4xlarge
Micro instances	t1.micro

Hardware Specifications

For more information about the hardware specifications for each Amazon EC2 instance type, see [Amazon EC2 Instances](#).

To determine which instance type best meets your needs, we recommend that you launch an instance and use your own benchmark application. Because you pay by the instance hour, it's convenient and inexpensive to test multiple instance types before making a decision.

Even after you make a decision, if your needs change, you can resize your instance later on. For more information, see [Resizing Your Instance \(p. 147\)](#).

Note

Amazon EC2 instances run on 64-bit virtual Intel processors as specified in the instance type product pages. For more information about the hardware specifications for each Amazon EC2 instance type, see [Amazon EC2 Instances](#). However, confusion may result from industry naming conventions for 64-bit CPUs. Chip manufacturer Advanced Micro Devices (AMD) introduced the first commercially successful 64-bit architecture based on the Intel x86 instruction set. Consequently, the architecture is widely referred to as AMD64 regardless of the chip manufacturer. Windows and several Linux distributions follow this practice. This explains why the internal system information on an Ubuntu or Windows EC2 instance displays the CPU architecture as AMD64 even though the instances are running on Intel hardware.

Networking and Storage Features

When you select an instance type, this determines the networking and storage features that are available.

Networking features

- Some instance types are not available in EC2-Classic, so you must launch them in a VPC. By launching an instance in a VPC, you can leverage features that are not available in EC2-Classic, such as enhanced networking, assigning multiple private IPv4 addresses to an instance, assigning

IPv6 addresses to an instance, and changing the security groups assigned to an instance. For more information, see [Instance Types Available Only in a VPC \(p. 671\)](#).

- To maximize the networking and bandwidth performance of your instance type, you can do the following:
 - Launch supported instance types into a placement group to optimize your instances for high performance computing (HPC) applications. Instances in a common placement group can benefit from high-bandwidth (10 Gbps), low-latency networking. For more information, see [Placement Groups \(p. 731\)](#). Instance types that support 10 Gbps network speeds can only take advantage of those network speeds when launched in a placement group.
 - Enable enhanced networking for supported current generation instance types to get significantly higher packet per second (PPS) performance, lower network jitter, and lower latencies. For more information, see [Enhanced Networking on Windows \(p. 737\)](#).
- The maximum supported MTU varies across instance types. All Amazon EC2 instance types support standard Ethernet V2 1500 MTU frames. All current generation instances support 9001 MTU, or jumbo frames, and some previous generation instances support them as well. For more information, see [Network Maximum Transmission Unit \(MTU\) for Your EC2 Instance \(p. 734\)](#).

Storage features

- Some instance types support EBS volumes and instance store volumes, while other instance types support only EBS volumes. Some instances that support instance store volumes use solid state drives (SSD) to deliver very high random I/O performance. For more information, see [Storage \(p. 744\)](#).
- To obtain additional, dedicated capacity for Amazon EBS I/O, you can launch some instance types as EBS-optimized instances. Some instance types are EBS-optimized by default. For more information, see [Amazon EBS-Optimized Instances \(p. 795\)](#).

The following table summarizes the networking and storage features supported by the current generation instance types.

	VPC only	EBS only	SSD volumes	Placement group	HVM only	Enhanced networking	IPv6 support (VPC only)
C3			Yes	Yes		Intel 82599 VF	Yes
C4	Yes	Yes		Yes	Yes	Intel 82599 VF	Yes
D2				Yes	Yes	Intel 82599 VF	Yes
G2			Yes	Yes	Yes		
I2			Yes	Yes	Yes	Intel 82599 VF	Yes
M3			Yes				
M4	Yes	Yes		Yes	Yes	m4.16xlarge ENA All other sizes: Intel 82599 VF	Yes

	VPC only	EBS only	SSD volumes	Placement group	HVM only	Enhanced networking	IPv6 support (VPC only)
P2	Yes	Yes		Yes	Yes	ENA	Yes
R3			Yes	Yes	Yes	Intel 82599 VF	Yes
R4	Yes	Yes		Yes	Yes	ENA	Yes
T2	Yes	Yes			Yes		Yes
X1	Yes		Yes	Yes	Yes	ENA	Yes

Instance Limits

There is a limit on the total number of instances that you can launch in a region, and there are additional limits on some instance types.

For more information about the default limits, see [How many instances can I run in Amazon EC2?](#)

For more information about viewing your current limits or requesting an increase in your current limits, see [Amazon EC2 Service Limits \(p. 869\)](#).

T2 Instances

T2 instances are designed to provide moderate baseline performance and the capability to burst to significantly higher performance as required by your workload. They are intended for workloads that don't use the full CPU often or consistently, but occasionally need to burst. T2 instances are well suited for general purpose workloads, such as web servers, developer environments, and small databases. For more information about T2 instance pricing and additional hardware details, see [Amazon EC2 Instances](#).

If your account is less than 12 months old, you can use a `t2.micro` instance for free within certain usage limits. For more information, see [AWS Free Tier](#).

Contents

- [Hardware Specifications \(p. 121\)](#)
- [T2 Instance Requirements \(p. 121\)](#)
- [CPU Credits \(p. 122\)](#)
- [Monitoring Your CPU Credits \(p. 124\)](#)

Hardware Specifications

For more information about the hardware specifications for each Amazon EC2 instance type, see [Amazon EC2 Instances](#).

T2 Instance Requirements

The following are the requirements for T2 instances:

- You must launch your T2 instances into a virtual private cloud (VPC); they are not supported on the EC2-Classical platform. Amazon VPC enables you to launch AWS resources into a virtual network

that you've defined. You cannot change the instance type of an existing instance in EC2-Classic to a T2 instance type. For more information about EC2-Classic and EC2-VPC, see [Supported Platforms \(p. 672\)](#) For more information about launching a VPC-only instance, see [Instance Types Available Only in a VPC \(p. 671\)](#).

- T2 instances are available as On-Demand instances and Reserved Instances, but they are not available as Spot instances, Scheduled Instances, or Dedicated instances. They are also not supported on a Dedicated Host. For more information about these options, see [Instance Purchasing Options \(p. 150\)](#).
- There is a limit on the total number of instances that you can launch in a region, and there are additional limits on some instance types. By default, you can run up to 20 T2 instances simultaneously. If you need more T2 instances, you can request them using the [Amazon EC2 Instance Request Form](#).
- Ensure that the T2 instance size you choose passes the minimum memory requirements of your operating system and applications. Operating systems with graphical user interfaces that consume significant memory and CPU resources (for example, Windows) may require a `t2.micro`, or larger, instance size for many use cases. As the memory and CPU requirements of your workload grows over time, you can scale to larger T2 instance sizes, or other EC2 instance types.

CPU Credits

A CPU Credit provides the performance of a full CPU core for one minute. Traditional Amazon EC2 instance types provide fixed performance, while T2 instances provide a baseline level of CPU performance with the ability to burst above that baseline level. The baseline performance and ability to burst are governed by CPU credits.

What is a CPU credit?

One CPU credit is equal to one vCPU running at 100% utilization for one minute. Other combinations of vCPUs, utilization, and time are also equal to one CPU credit; for example, one vCPU running at 50% utilization for two minutes or two vCPUs running at 25% utilization for two minutes.

How are CPU credits earned?

Each T2 instance starts with a healthy initial CPU credit balance and then continuously (at a millisecond-level resolution) receives a set rate of CPU credits per hour, depending on instance size. The accounting process for whether credits are accumulated or spent also happens at a millisecond-level resolution, so you don't have to worry about overspending CPU credits; a short burst of CPU takes a small fraction of a CPU credit.

When a T2 instance uses fewer CPU resources than its base performance level allows (such as when it is idle), the unused CPU credits (or the difference between what was earned and what was spent) are stored in the credit balance for up to 24 hours, building CPU credits for bursting. When your T2 instance requires more CPU resources than its base performance level allows, it uses credits from the CPU credit balance to burst up to 100% utilization. The more credits your T2 instance has for CPU resources, the more time it can burst beyond its base performance level when more performance is needed.

The following table lists the initial CPU credit allocation received at launch, the rate at which CPU credits are received, the baseline performance level as a percentage of a full core performance, and the maximum earned CPU credit balance that an instance can accrue.

Instance type	Initial CPU credit*	CPU credits earned per hour	Base performance (CPU utilization)	Maximum earned CPU credit balance***
<code>t2.nano</code>	30	3	5%	72

Instance type	Initial CPU credit*	CPU credits earned per hour	Base performance (CPU utilization)	Maximum earned CPU credit balance***
t2.micro	30	6	10%	144
t2.small	30	12	20%	288
t2.medium	60	24	40%**	576
t2.large	60	36	60%**	864
t2.xlarge	120	54	90%**	1296
t2.2xlarge	240	81	135%**	1944

* There are limits to how many T2 instances will launch or start with the initial CPU credit, which by default is set to 100 launches or starts of any T2 instance per account, per 24-hour period, per region. If you'd like to increase this limit, you can file a customer support limit increase request by using the [Amazon EC2 Credit Based Instances Launch Credits Form](#). If your account does not launch or start more than 100 T2 instances in 24 hours, this limit will not affect you.

** t2.medium and larger instances have more than one vCPU. The base performance is an aggregate of the available vCPUs. For example, if a t2.large uses 100% of one vCPU and a small amount of the other, the CloudWatch metrics will show over 50% utilization.

*** This maximum does not include the initial CPU credits, which are used first and do not expire. For example, a t2.micro instance that was launched and then remained idle for over 24 hours could reach a credit balance of up to 174 (30 initial CPU credits + 144 earned credits). However, after the instance uses the initial 30 CPU credits, the credit balance can never exceed 144 unless a new initial CPU credit balance is issued by stopping and starting the instance.

The initial credit balance is designed to provide a good startup experience. The maximum earned credit balance for an instance is equal to the number of CPU credits received per hour times 24 hours. For example, a t2.micro instance earns 6 CPU credits per hour and can accumulate a maximum earned CPU credit balance of 144 CPU credits.

Do CPU credits expire?

Initial CPU credits do not expire, but they are used first when an instance uses CPU credits. Unused earned credits from a given 5 minute interval expire 24 hours after they are earned, and any expired credits are removed from the CPU credit balance at that time, before any newly earned credits are added. Additionally, the CPU credit balance for an instance does not persist between instance stops and starts; stopping an instance causes it to lose its credit balance entirely, but when it restarts it will receive its initial credit balance again.

For example, if a t2.small instance had a CPU utilization of 5% for the hour, it would have used 3 CPU credits (5% of 60 minutes), but it would have earned 12 CPU credits during the hour, so the difference of 9 CPU credits would be added to the CPU credit balance. Any CPU credits in the balance that reached their 24 hour expiration date during that time (which could be as many as 12 credits if the instance was completely idle 24 hours ago) would also be removed from the balance. If the amount of credits expired is greater than those earned, the credit balance will go down; conversely, if the amount of credits expired is fewer than those earned, the credit balance will go up.

What happens if I use all of my credits?

If your instance uses all of its CPU credit balance, performance remains at the baseline performance level. If your instance is running low on credits, your instance's CPU credit consumption (and therefore CPU performance) is gradually lowered to the base performance level over a 15-minute interval, so

you will not experience a sharp performance drop-off when your CPU credits are depleted. If your instance consistently uses all of its CPU credit balance, we recommend a larger T2 size or a fixed performance instance type such as M3 or C3.

Monitoring Your CPU Credits

You can see the credit balance for each T2 instance presented in the Amazon EC2 per-instance metrics of the CloudWatch console. T2 instances have two metrics, `CPUCreditUsage` and `CPUCreditBalance`. The `CPUCreditUsage` metric indicates the number of CPU credits used during the measurement period. The `CPUCreditBalance` metric indicates the number of unused CPU credits a T2 instance has earned. This balance is depleted during burst time as CPU credits are spent more quickly than they are earned.

The following table describes the new available CloudWatch metrics. For more information about using these metrics in CloudWatch, see [List the Available CloudWatch Metrics for Your Instances \(p. 577\)](#).

Metric	Description
<code>CPUCreditUsage</code>	<p>[T2 instances] The number of CPU credits consumed during the specified period.</p> <p>This metric identifies the amount of time during which physical CPUs were used for processing instructions by virtual CPUs allocated to the instance.</p> <p>CPU Credit metrics are available at a 5 minute frequency.</p> <p>Units: Count</p>
<code>CPUCreditBalance</code>	<p>[T2 instances] The number of CPU credits that an instance has accumulated.</p> <p>This metric determines how long an instance can burst beyond its baseline performance level at a given rate.</p> <p>CPU Credit metrics are available at a 5 minute frequency.</p> <p>Units: Count</p>

Memory Optimized Instances

Memory optimized instances are designed to deliver fast performance for workloads that process large data sets in memory.

R4 Instances

R4 instances are well suited for the following applications:

- High performance relational (MySQL) and NoSQL (MongoDB, Cassandra) databases.
- Distributed web scale cache stores that provide in-memory caching of key-value type data (Memcached and Redis).
- In-memory databases using optimized data storage formats and analytics for business intelligence (for example, SAP HANA).
- Applications performing real-time processing of big unstructured data (financial services, Hadoop/Spark clusters).
- High-performance computing (HPC) and Electronic Design Automation (EDA) applications.

X1 Instances

X1 instances are well suited for the following applications:

- In-memory databases such as SAP HANA, including SAP-certified support for Business Suite S/4HANA, Business Suite on HANA (SoH), Business Warehouse on HANA (BW), and Data Mart Solutions on HANA. For more information, see [SAP HANA on the AWS Cloud](#).
- Big-data processing engines such as Apache Spark or Presto.
- High-performance computing (HPC) applications.

R3 Instances

R3 instances are well suited for the following applications:

- High performance relational (MySQL) and NoSQL (MongoDB, Cassandra) databases.
- In-memory analytics.
- Genome assembly and analysis.
- Enterprise applications (for example, Microsoft SharePoint).

Hardware Specifications

For more information about the hardware specifications for each Amazon EC2 instance type, see [Amazon EC2 Instances](#).

Memory Performance

R4 instances enable up to 488 GiB of RAM.

X1 instances include Intel Scalable Memory Buffers, providing 300 GiB/s of sustainable memory-read bandwidth and 140 GiB/s of sustainable memory-write bandwidth.

R3 instances enable up to 244 GiB of RAM.

Memory optimized instances have high-memory and require 64-bit HVM AMIs to take advantage of that capacity. HVM AMIs provide superior performance in comparison to paravirtual (PV) AMIs on high-memory instance types. .

Compute Performance

R4 instances feature up to 64 vCPUs and are powered by two AWS-customized Intel XEON processors based on E5-2686v4 that feature high-memory bandwidth and larger L3 caches to boost the performance of in-memory applications.

X1 instances feature up to 128 vCPUs and are powered by four Intel Xeon E7-8880 v3 processors that feature high-memory bandwidth and larger L3 caches to boost the performance of in-memory applications.

Memory optimized instances enable increased cryptographic performance through the latest Intel AES-NI feature, support Intel Transactional Synchronization Extensions (TSX) to boost the performance of in-memory transactional data processing, and support Advanced Vector Extensions 2 (Intel AVX2) processor instructions to expand most integer commands to 256 bits.

Network Performance

To increase network performance of your memory optimized instances, enable enhanced networking. For more information, see [Enhanced Networking on Windows \(p. 737\)](#).

R4 instances deliver high packet per second performance with consistently low latencies using Elastic Network Adapter (ENA). Most applications do not consistently need a high level of network performance, but can benefit from having access to increased bandwidth when they send or receive data. The smaller R4 instance sizes offer peak throughput of 10 Gbps. These instances use a network I/O credit mechanism to allocate network bandwidth to instances based on average bandwidth utilization. These instances accrue credits when their network throughput is below their baseline limits, and can use these credits when they perform network data transfers. For workloads that require access to 10 Gbps or higher bandwidth on a sustained basis, we recommend using `r4.8xlarge` and `r4.16xlarge` instances, which can utilize up to 10 Gbps and 20 Gbps of network bandwidth, respectively.

Instance Features

The following is a summary of features for memory optimized instances.

	VPC only	EBS only	SSD volumes	Placement group	Enhanced networking
R3			Yes	Yes	Intel 82599 VF
R4	Yes	Yes		Yes	ENA
X1	Yes		Yes	Yes	ENA

For more information, see the following:

- [Instance Types Available Only in a VPC \(p. 671\)](#)
- [Amazon EBS–Optimized Instances \(p. 795\)](#)
- [Amazon EC2 Instance Store \(p. 822\)](#)
- [Placement Groups \(p. 731\)](#)
- [Enhanced Networking on Windows \(p. 737\)](#)

High Availability and Reliability (X1)

X1 instances support Single Device Data Correction (SDDC +1), which detects and corrects multi-bit errors. SDDC +1 uses error checking and correction code to identify and disable a failed single DRAM device.

In addition, you can implement high availability (HA) and disaster recovery (DR) solutions to meet recovery point objective (RPO), recovery time objective (RTO), and cost requirements by leveraging [Amazon CloudFormation](#) and [Recover Your Instance \(p. 269\)](#). For more information about implementing HA and DR solutions, see the [Using AWS for Disaster Recovery](#) whitepaper.

If you run an SAP HANA production environment, you also have the option of using HANA System Replication (HSR) on X1 instances. For more information about architecting HA and DR solutions on X1 instances, see [SAP HANA on the Amazon Web Services Cloud: Quick Start Reference Deployment](#).

Support for vCPUs

Memory optimized instances provide a high number of vCPUs, which can cause launch issues with operating systems that have a lower vCPU limit. We strongly recommend that you use the latest AMIs when you launch memory optimized instances.

The following AMIs support launching memory optimized instances:

- Amazon Linux AMI 2016.03 (HVM) or later
- Ubuntu Server 14.04 LTS (HVM)
- Red Hat Enterprise Linux 7.1 (HVM)
- SUSE Linux Enterprise Server 12 SP1 (HVM)
- Windows Server 2016
- Windows Server 2012 R2
- Windows Server 2012
- Windows Server 2008 R2 64-bit
- Windows Server 2008 SP2 64-bit
- Windows Server 2003 R2 64-bit

Instance Limits

- You can't launch R4 instances as Spot Instances or Dedicated Instances.
- You can't launch `r4.large` and `r4.4xlarge` instances using a Windows Server 2008 R2 64-bit AMI.
- You can't launch X1 instances using a Windows Server 2008 SP2 64-bit AMI or a Windows Server 2003 R2 64-bit AMI, except for `x1.16xlarge` instances.
- There is a limit on the total number of instances that you can launch in a region, and there are additional limits on some instance types. For more information, see [How many instances can I run in Amazon EC2?](#). To request a limit increase, use the [Amazon EC2 Instance Request Form](#).

Windows Accelerated Computing Instances

If you require high parallel processing capability, you'll benefit from using accelerated computing instances, which provide access to NVIDIA GPUs. You can use accelerated computing instances to accelerate many scientific, engineering, and rendering applications by leveraging the CUDA or Open Computing Language (OpenCL) parallel computing frameworks. You can also use them for graphics applications, including game streaming, 3-D application streaming, and other graphics workloads.

Accelerated computing instances run as HVM-based instances. Hardware virtual machine (HVM) virtualization uses hardware-assist technology provided by the AWS platform. With HVM virtualization, the guest VM runs as if it were on a native hardware platform, which enables Amazon EC2 to provide dedicated access to one or more discrete GPUs in each accelerated computing instance.

You can cluster accelerated computing instances into a placement group. Placement groups provide low latency and high-bandwidth connectivity between the instances within a single Availability Zone. For more information, see [Placement Groups \(p. 731\)](#).

Contents

- [Accelerated Computing Instance Families \(p. 128\)](#)
- [GPU Hardware Specifications \(p. 128\)](#)
- [Accelerated Computing Instance Limitations \(p. 128\)](#)
- [AMIs for Accelerated Computing Instances \(p. 128\)](#)
- [Installing the NVIDIA Driver on Windows \(p. 129\)](#)

For information about Linux accelerated computing instances, see [Linux Accelerated Computing Instances](#) in the *Amazon EC2 User Guide for Linux Instances*.

Accelerated Computing Instance Families

Accelerated computing instance families use hardware accelerators, or co-processors, to perform some functions, such as floating point number calculation and graphics processing, more efficiently than is possible in software running on CPUs. The following accelerated computing instance families are available for you to launch in Amazon EC2.

P2 Instances

P2 instances use NVIDIA Tesla K80 GPUs and are designed for general purpose GPU computing using the CUDA or OpenCL programming models. P2 instances provide high bandwidth networking, powerful single and double precision floating-point capabilities, and 12 GiB of memory per GPU, which makes them ideal for deep learning, graph databases, high performance databases, computational fluid dynamics, computational finance, seismic analysis, molecular modeling, genomics, rendering, and other server-side GPU compute workloads.

- P2 instances support enhanced networking with the Elastic Network Adapter. For more information, see [Enabling Enhanced Networking with the Elastic Network Adapter \(ENA\) on Windows Instances in a VPC \(p. 740\)](#).
- P2 instances are EBS-optimized by default. For more information, see [Amazon EBS–Optimized Instances \(p. 795\)](#).
- P2 instances support NVIDIA GPUDirect peer to peer transfers. For more information, see [NVIDIA GPUDirect](#).

G2 Instances

G2 instances use NVIDIA GRID K520 GPUs and provide a cost-effective, high-performance platform for graphics applications using DirectX or OpenGL. NVIDIA GRID GPUs also support NVIDIA's fast capture and encode API operations. Example applications include video creation services, 3D visualizations, streaming graphics-intensive applications, and other server-side graphics workloads.

CG1 Instances

CG1 instances use NVIDIA Tesla M2050 GPUs and are designed for general purpose GPU computing using the CUDA or OpenCL programming models. CG1 instances provide customers with high bandwidth networking, double precision floating-point capabilities, and error-correcting code (ECC) memory, making them ideal for high performance computing (HPC) applications.

GPU Hardware Specifications

For more information about the hardware specifications for each Amazon EC2 instance type, see [Amazon EC2 Instances](#).

Accelerated Computing Instance Limitations

Accelerated computing instances have the following limitations:

- You must launch the instance using an HVM AMI.
- The instance can't access the GPU unless the NVIDIA drivers are installed.
- There is a limit on the number of instances that you can run. For more information, see [How many instances can I run in Amazon EC2?](#) in the Amazon EC2 FAQ. To request an increase in these limits, use the following form: [Request to Increase Amazon EC2 Instance Limit](#).

AMIs for Accelerated Computing Instances

To help you get started, NVIDIA provides AMIs for accelerated computing instances. These reference AMIs include the NVIDIA driver, which enables full functionality and performance of the NVIDIA GPUs.

For a list of AMIs with the NVIDIA driver, see [AWS Marketplace \(NVIDIA GRID\)](#).

You can launch accelerated computing instances using any HVM AMI.

Installing the NVIDIA Driver on Windows

To install the NVIDIA driver on your Windows instance, log on to your instance as the administrator using Remote Desktop. You can download NVIDIA drivers from <http://www.nvidia.com/Download/Find.aspx>. Select the appropriate driver for your instance type:

- P2 instances: K-Series K-80
- G2 instances: GRID K520
- CG1 instances: Tesla M-Class M2050

Open the folder where you downloaded the driver and double-click the installation file to launch it. Follow the instructions to install the driver and reboot your instance as required. To verify that the GPU is working properly, check Device Manager.

Note

If you launch a multi-GPU instance with a Windows AMI that was created on a single-GPU instance, Windows does not automatically install the NVIDIA driver for all GPUs. You must authorize the driver installation for the new GPU hardware.

You can correct this manually in the Device Manager by opening the **Other** device category (the inactive GPUs do not appear under **Display Adapters**). For each inactive GPU, open the context (right-click) menu and choose **Update Driver Software**, and then choose the default **Automatic Update** option.

When using Remote Desktop, GPUs that use the WDDM driver model are replaced with a non-accelerated Remote Desktop display driver. To access your GPU hardware, you must use a different remote access tool, such as VNC. You can also use one of the GPU AMIs from the AWS Marketplace because they provide remote access tools that support 3-D acceleration.

C4 Instances

C4 instances are ideal for compute-bound applications that benefit from high performance processors. C4 instances are well suited for the following applications:

- Batch processing workloads
- Media transcoding
- High-traffic web servers, massively multiplayer online (MMO) gaming servers, and ad serving engines
- High performance computing (HPC) and other compute-intensive applications

Contents

- [Hardware Specifications \(p. 129\)](#)
- [C4 Instance Features \(p. 130\)](#)
- [C4 Instance Requirements \(p. 131\)](#)

Hardware Specifications

C4 instances are based on custom 2.9 GHz Intel Xeon E5-2666 v3 (Haswell) processors, optimized specifically for Amazon EC2. With Intel Turbo Boost Technology, the processor clock speed in C4 instances can reach as high as 3.5GHz with 1 or 2 core Turbo Boost on `c4.8xlarge` instances.

The following table highlights the feature set of the Intel Xeon E5-2666 v3 processor. For more information, see [Intel and Amazon Web Services](#).

Feature	Specification
Processor Number	E5-2666 v3
Intel Smart Cache	25 MiB
Instruction Set	64-bit
Instruction Set Extensions	AVX 2.0
Lithography	22 nm
Processor Base Frequency	2.9 GHz
Max All Core Turbo Frequency	3.2 GHz
Max Turbo Frequency	3.5 GHz (available on c4.8xlarge)
Intel Turbo Boost Technology	2.0
Intel vPro Technology	Yes
Intel Hyper-Threading Technology	Yes
Intel 64	Yes
Idle States	Yes
Enhanced Intel SpeedStep Technology	Yes
Thermal Monitoring Technologies	Yes
AES New Instructions	Yes
Secure Key	Yes
Execute Disable Bit	Yes

For more information about the hardware specifications for each Amazon EC2 instance type, see [Amazon EC2 Instances](#).

C4 Instance Features

The following is a summary of the features for C4 instances:

- C4 instances are EBS-optimized by default, and deliver dedicated block storage throughput to Amazon EBS ranging from 500 Mbps to 4,000 Mbps at no additional cost. EBS-optimized instances enable you to get consistently high performance for your EBS volumes by eliminating contention between Amazon EBS I/O and other network traffic from your C4 instance. For more information, see [Amazon EBS–Optimized Instances \(p. 795\)](#).
- You can enable enhanced networking capabilities. Enhanced networking provides significantly higher packet per second (PPS) performance, lower network jitter, and lower latencies. For more information, see [Enhanced Networking on Windows \(p. 737\)](#).
- You can cluster C4 instances in a placement group. Placement groups provide low latency and high-bandwidth connectivity between the instances within a single Availability Zone. For more information, see [Placement Groups \(p. 731\)](#).

C4 Instance Requirements

The following are the requirements for C4 instances:

- C4 instances require 64-bit HVM AMIs. They have high-memory (up to 60 GiB of RAM), and require a 64-bit operating system to take advantage of that capacity. HVM AMIs provide superior performance in comparison to paravirtual (PV) AMIs on high-memory instance types. In addition, you must use an HVM AMI to take advantage of enhanced networking.
- You must launch your C4 instances into a virtual private cloud (VPC); they are not supported on the EC2-Classic platform. Amazon VPC enables you to launch AWS resources into a virtual network that you've defined. For more information about EC2-Classic and EC2-VPC, see [Supported Platforms \(p. 672\)](#) For more information about launching a VPC-only instance, see [Instance Types Available Only in a VPC \(p. 671\)](#).
- There is a limit on the total number of instances that you can launch in a region, and there are additional limits on some C4 instance types. For more information, see [How many instances can I run in Amazon EC2?](#)

If you need more C4 instances, you can request them using the [Amazon EC2 Instance Request Form](#).

I2 Instances

I2 instances are optimized to deliver tens of thousands of low-latency, random I/O operations per second (IOPS) to applications. They are well suited for the following scenarios:

- NoSQL databases (for example, Cassandra and MongoDB)
- Clustered databases
- Online transaction processing (OLTP) systems

Contents

- [Hardware Specifications \(p. 131\)](#)
- [I2 Instance Features \(p. 131\)](#)
- [I2 Instance Requirements \(p. 132\)](#)
- [SSD I/O Performance \(p. 132\)](#)

Hardware Specifications

For more information about the hardware specifications for each Amazon EC2 instance type, see [Amazon EC2 Instances](#).

I2 Instance Features

The following is a summary of the features for I2 instances:

- The primary data storage is SSD-based instance storage. Like all instance storage, these volumes persist only for the life of the instance. When you stop or terminate an instance, the applications and data in its instance store are erased. We recommend that you regularly back up or replicate the data that you've stored in instance storage. For more information, see [SSD Instance Store Volumes \(p. 828\)](#).
- You can enable enhanced networking capabilities. Enhanced networking provides significantly higher packet per second (PPS) performance, lower network jitter, and lower latencies. For more information, see [Enhanced Networking on Windows \(p. 737\)](#).

- You can cluster I2 instances in a placement group. Placement groups provide low latency and high-bandwidth connectivity between the instances within a single Availability Zone. For more information, see [Placement Groups \(p. 731\)](#).
- You can enable EBS–optimization to obtain additional, dedicated capacity for Amazon EBS I/O. For more information, see [Amazon EBS–Optimized Instances \(p. 795\)](#).

I2 Instance Requirements

The following are the requirements for I2 instances:

- You must launch an I2 instance using an HVM AMI.
- There is a limit on the total number of instances that you can launch in a region, and there are additional limits on some I2 instance types. For more information, see [How many instances can I run in Amazon EC2?](#)

If you need more I2 instances, you can request them using the [Amazon EC2 Instance Request Form](#).

SSD I/O Performance

If you use a Linux AMI with kernel version 3.8 or later and utilize all the SSD-based instance store volumes available to the instance, you can get at least the minimum random IOPS (4,096 byte block size) listed in the following table. Otherwise, you'll get lower IOPS performance than what is shown in the table.

Instance Size	Read IOPS	First Write IOPS
i2.xlarge	35,000	35,000
i2.2xlarge	75,000	75,000
i2.4xlarge	175,000	155,000
i2.8xlarge	365,000	315,000

As you fill the SSD-based instance storage for your instance, the number of write IOPS that you can achieve decreases. This is due to the extra work the SSD controller must do to find available space, rewrite existing data, and erase unused space so that it can be rewritten. This process of garbage collection results in internal write amplification to the SSD, expressed as the ratio of SSD write operations to user write operations. This decrease in performance is even larger if the write operations are not in multiples of 4,096 bytes or not aligned to a 4,096-byte boundary. If you write a smaller amount of bytes or bytes that are not aligned, the SSD controller must read the surrounding data and store the result in a new location. This pattern results in significantly increased write amplification, increased latency, and dramatically reduced I/O performance.

SSD controllers can use several strategies to reduce the impact of write amplification. One such strategy is to reserve space in the SSD instance storage so that the controller can more efficiently manage the space available for write operations. This is called *over-provisioning*. The SSD-based instance store volumes provided to an I2 instance don't have any space reserved for over-provisioning. To reduce write amplification, you should leave 10% of the volume unpartitioned so that the SSD controller can use it for over-provisioning. This decreases the storage that you can use, but increases performance.

I2 instance store–backed volumes support TRIM. You can use the TRIM command to notify the SSD controller whenever you no longer need data that you've written. This provides the controller with more

free space, which can reduce write amplification and increase performance. For more information, see [Instance Store Volume TRIM Support \(p. 828\)](#).

D2 Instances

D2 instances are designed for workloads that require high sequential read and write access to very large data sets on local storage. D2 instances are well suited for the following applications:

- Massive parallel processing (MPP) data warehouse
- MapReduce and Hadoop distributed computing
- Log or data processing applications

Contents

- [Hardware Specifications \(p. 133\)](#)
- [D2 Instance Features \(p. 133\)](#)
- [D2 Instance Requirements \(p. 133\)](#)

Hardware Specifications

For more information about the hardware specifications for each Amazon EC2 instance type, see [Amazon EC2 Instances](#).

D2 Instance Features

The following is a summary of the features for D2 instances:

- The primary data storage for D2 instances is HDD-based instance storage. Like all instance storage, these volumes persist only for the life of the instance. For more information about instance store volumes, see [Amazon EC2 Instance Store \(p. 822\)](#).
- D2 instances are EBS-optimized by default, and deliver dedicated block storage throughput to Amazon EBS ranging from 750 Mbps to 4,000 Mbps at no additional cost. EBS-optimized instances enable you to get consistently high performance for your EBS volumes by eliminating contention between Amazon EBS I/O and other network traffic from your D2 instance. For more information, see [Amazon EBS-Optimized Instances \(p. 795\)](#).
- You can enable enhanced networking capabilities. Enhanced networking provides significantly higher packet per second (PPS) performance, lower network jitter, and lower latencies. For more information, see [Enhanced Networking on Windows \(p. 737\)](#).
- You can cluster D2 instances in a placement group. Placement groups provide low latency and high-bandwidth connectivity between the instances within a single Availability Zone. For more information, see [Placement Groups \(p. 731\)](#).

D2 Instance Requirements

The following are the requirements for D2 instances:

- D2 instances require 64-bit HVM AMIs. They have high-memory (up to 244 GiB of RAM), and require a 64-bit operating system to take advantage of that capacity. HVM AMIs provide superior performance in comparison to paravirtual (PV) AMIs on high-memory instance types. In addition, you must use an HVM AMI to take advantage of enhanced networking.
- There is a limit on the total number of instances that you can launch in a region, and there are additional limits on some D2 instance types. For more information, see [How many instances can I run in Amazon EC2?](#)

If you need more D2 instances, you can request them using the [Amazon EC2 Instance Request Form](#).

- Your `d2.8xlarge` instances are capable of providing up to 3.5 GB/s read performance and 3.1 GB/s write performance with a 2 MiB block size.

HI1 Instances

HI1 instances (`hi1.4xlarge`) can deliver tens of thousands of low-latency, random I/O operations per second (IOPS) to applications. They are well suited for the following scenarios:

- NoSQL databases (for example, Cassandra and MongoDB)
- Clustered databases
- Online transaction processing (OLTP) systems

You can cluster HI1 instances in a placement group. For more information, see [Placement Groups \(p. 731\)](#).

By default, you can run up to two `hi1.4xlarge` instances. If you need more than two `hi1.4xlarge` instances, contact <http://aws.amazon.com/premiumsupport/>.

Contents

- [Hardware Specifications \(p. 134\)](#)
- [Disk I/O Performance \(p. 134\)](#)
- [SSD Storage \(p. 134\)](#)

Hardware Specifications

The `hi1.4xlarge` instance type is based on solid-state drive (SSD) technology.

For more information about the hardware specifications for each Amazon EC2 instance type, see [Amazon EC2 Instances](#).

Disk I/O Performance

Using Linux paravirtual (PV) AMIs, HI1 instances can deliver more than 120,000 4 KB random read IOPS and between 10,000 and 85,000 4 KB random write IOPS (depending on active logical block addressing span) to applications across two SSD data volumes. Using hardware virtual machine (HVM) AMIs, performance is approximately 90,000 4 KB random read IOPS and between 9,000 and 75,000 4 KB random write IOPS.

HI1 Windows instances deliver approximately 90,000 4 KB random read IOPS and between 9,000 and 75,000 4 KB random write IOPS.

The maximum sequential throughput is approximately 2 GB read per second and 1.1 GB write per second.

SSD Storage

With SSD storage on HI1 instances:

- The primary data source is an instance store with SSD storage.
- Read performance is consistent and write performance can vary.

- Write amplification can occur.
- The TRIM command is not currently supported.

Instance Store with SSD Storage

The `hi1.4xlarge` instances use an Amazon EBS-backed root device. However, their primary data storage is provided by the SSD volumes in the instance store. Like other instance store volumes, these instance store volumes persist only for the life of the instance. Because the root device of the `hi1.4xlarge` instance is Amazon EBS-backed, you can still start and stop your instance. When you stop an instance, your application persists, but your production data in the instance store does not persist. For more information about instance store volumes, see [Amazon EC2 Instance Store \(p. 822\)](#).

Variable Write Performance

Write performance depends on how your applications utilize logical block addressing (LBA) space. If your applications use the total LBA space, write performance can degrade by about 90 percent. Benchmark your applications and monitor the queue length (the number of pending I/O requests for a volume) and I/O size.

Write Amplification

Write amplification refers to an undesirable condition associated with flash memory and SSDs, where the actual amount of physical information written is a multiple of the logical amount intended to be written. Because flash memory must be erased before it can be rewritten, the process to perform these operations results in moving (or rewriting) user data and metadata more than once. This multiplying effect increases the number of writes required over the life of the SSD, which shortens the time that it can reliably operate. The `hi1.4xlarge` instances are designed with a provisioning model intended to minimize write amplification.

Random writes have a much more severe impact on write amplification than serial writes. If you are concerned about write amplification, allocate less than the full tebibyte of storage for your application (also known as over provisioning).

The TRIM Command

The TRIM command enables the operating system to notify an SSD that blocks of previously saved data are considered no longer in use. TRIM limits the impact of write amplification.

TRIM support is not available for HI1 instances. For information about instances that support TRIM, see [Instance Store Volume TRIM Support \(p. 828\)](#).

HS1 Instances

HS1 instances (`hs1.8xlarge`) provide very high storage density and high sequential read and write performance per instance. They are well suited for the following scenarios:

- Data warehousing
- Hadoop/MapReduce
- Parallel file systems

You can cluster HS1 instances in a placement group. For more information, see [Placement Groups \(p. 731\)](#).

By default, you can run up to two HS1 instances. If you need more than two HS1 instances, you can request more using the [Amazon EC2 Instance Request Form](#).

Contents

- [Hardware Specifications \(p. 136\)](#)
- [Instance Store \(p. 136\)](#)
- [Disk Initialization \(p. 136\)](#)

Hardware Specifications

HS1 instances support both Amazon Elastic Block Store (Amazon EBS)-backed and instance store-backed Amazon Machine Images (AMIs). HS1 instances support both paravirtual (PV) and hardware virtual machine (HVM) AMIs.

HS1 instances provide high bandwidth networking and can also be used with Provisioned IOPS SSD (io1) volumes for improved consistency and performance.

For more information about the hardware specifications for each Amazon EC2 instance type, see [Amazon EC2 Instances](#).

Instance Store

HS1 instances support both instance store and Amazon EBS root device volumes. However, even when using an Amazon EBS-backed instance, primary data storage is provided by the hard disk drives in the instance store. Like other instance store volumes, these instance store volumes persist only for the life of the instance. For more information about instance store volumes, see [Amazon EC2 Instance Store \(p. 822\)](#).

Disk Initialization

If you plan to run an HS1 instance in a steady state for long periods of time, we recommend that you zero the hard disks first for improved performance. This process can take as long as six hours to complete.

T1 Micro Instances

T1 Micro instances (`t1.micro`) provide a small amount of consistent CPU resources and allow you to increase CPU capacity in short bursts when additional cycles are available. They are well suited for lower throughput applications and websites that require additional compute cycles periodically.

Note

The `t1.micro` is a previous generation instance and it has been replaced by the `t2.micro`, which has a much better performance profile. We recommend using the `t2.micro` instance type instead of the `t1.micro`. For more information, see [T2 Instances \(p. 121\)](#).

The `t1.micro` instance is available as an Amazon EBS-backed instance only.

This documentation describes how `t1.micro` instances work so that you can understand how to apply them. It's not our intent to specify exact behavior, but to give you visibility into the instance's behavior so you can understand its performance.

Topics

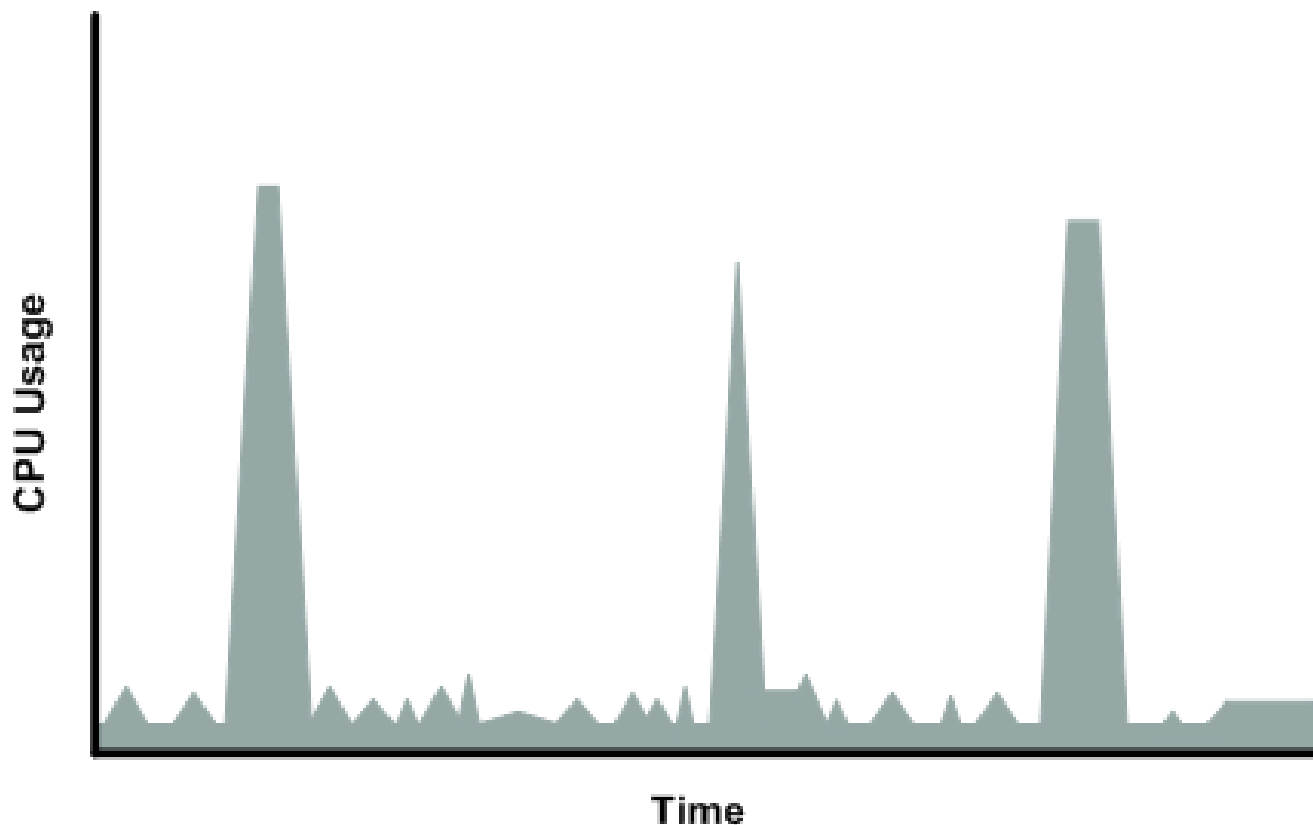
- [Hardware Specifications \(p. 137\)](#)
- [Optimal Application of T1 Micro Instances \(p. 137\)](#)
- [Available CPU Resources During Spikes \(p. 140\)](#)
- [When the Instance Uses Its Allotted Resources \(p. 140\)](#)
- [Comparison with the m1.small Instance Type \(p. 143\)](#)
- [AMI Optimization for Micro Instances \(p. 146\)](#)

Hardware Specifications

For more information about the hardware specifications for each Amazon EC2 instance type, see [Amazon EC2 Instances](#).

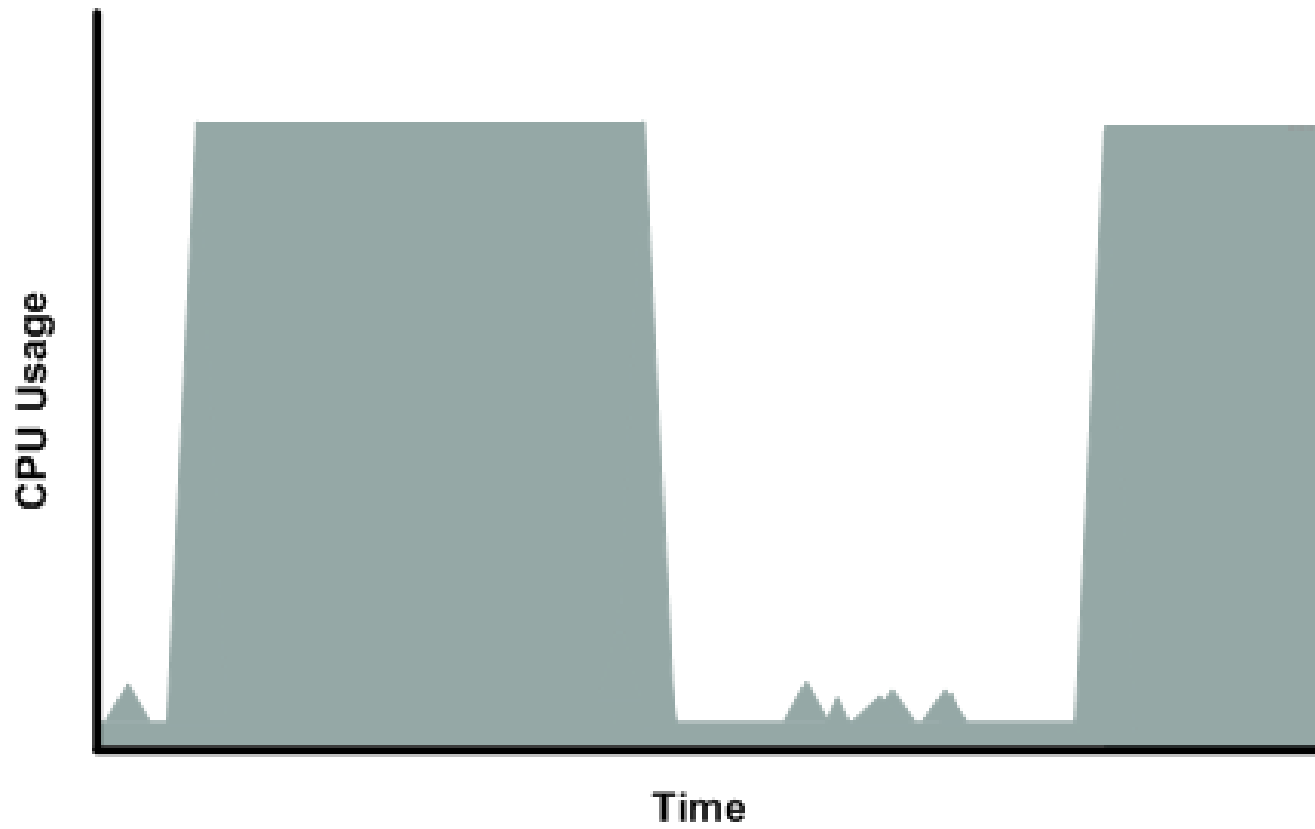
Optimal Application of T1 Micro Instances

A `t1.micro` instance provides spiky CPU resources for workloads that have a CPU usage profile similar to what is shown in the following figure.

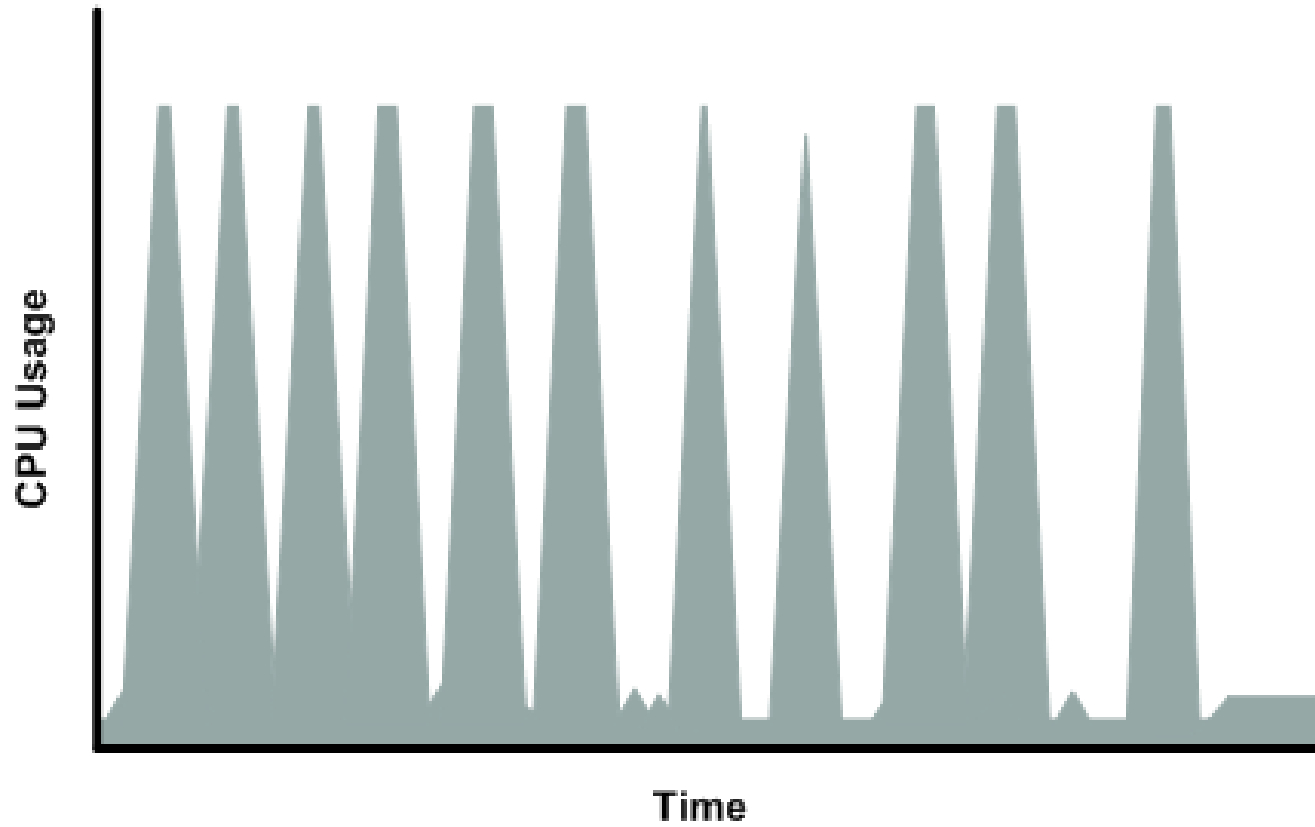


The instance is designed to operate with its CPU usage at essentially only two levels: the normal low background level, and then at brief spiked levels much higher than the background level. We allow the instance to operate at up to 2 EC2 compute units (ECUs) (one ECU provides the equivalent CPU capacity of a 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor). The ratio between the maximum level and the background level is designed to be large. We designed `t1.micro` instances to support tens of requests per minute on your application. However, actual performance can vary significantly depending on the amount of CPU resources required for each request on your application.

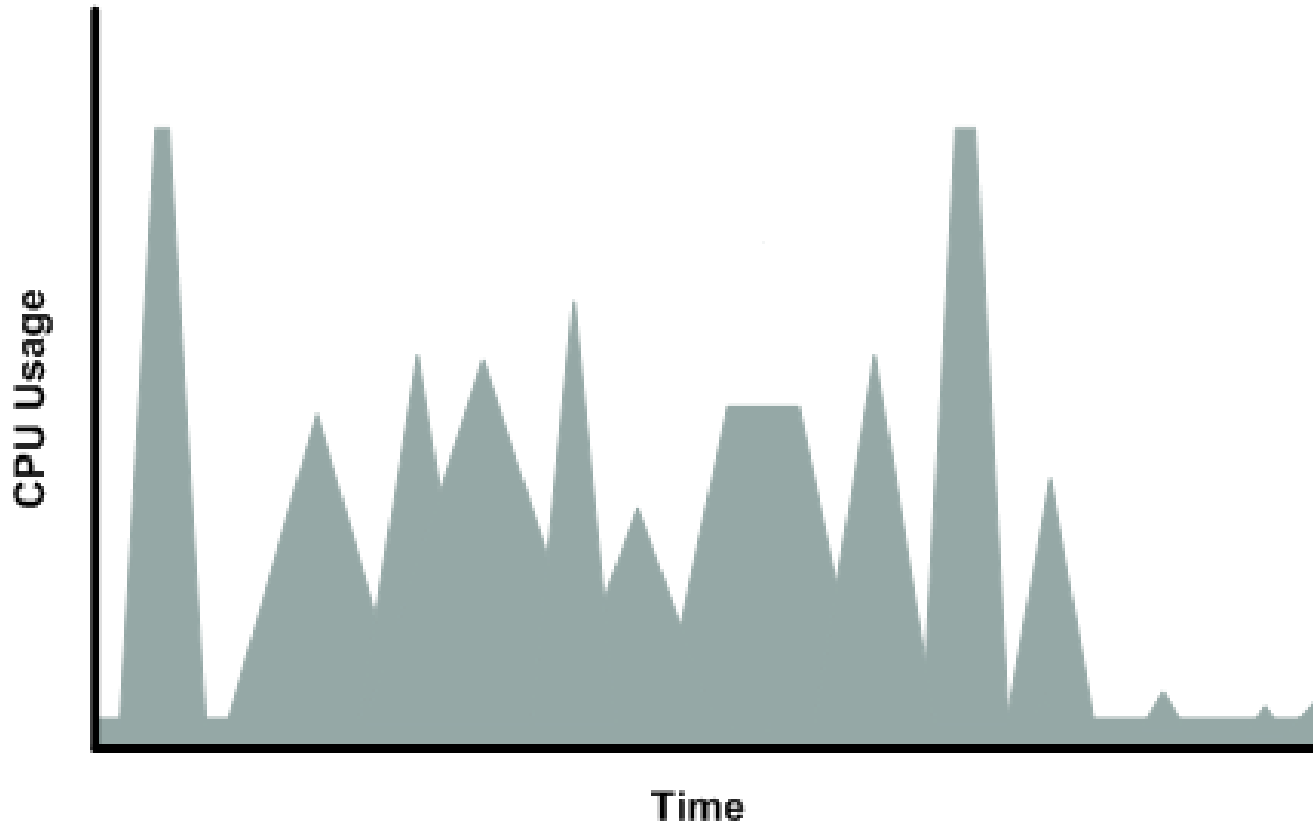
Your application might have a different CPU usage profile than that described in the preceding section. The next figure shows the profile for an application that isn't appropriate for a `t1.micro` instance. The application requires continuous data-crunching CPU resources for each request, resulting in plateaus of CPU usage that the `t1.micro` instance isn't designed to handle.



The next figure shows another profile that isn't appropriate for a `t1.micro` instance. Here the spikes in CPU use are brief, but they occur too frequently to be serviced by a micro instance.



The next figure shows another profile that isn't appropriate for a `t1.micro` instance. Here the spikes aren't too frequent, but the background level between spikes is too high to be serviced by a `t1.micro` instance.



In each of the preceding cases of workloads not appropriate for a `t1.micro` instance, we recommend that you consider using a different instance type. For more information about instance types, see [Instance Types \(p. 117\)](#).

Available CPU Resources During Spikes

When your instance *bursts* to accommodate a spike in demand for compute resources, it uses unused resources on the host. The amount available depends on how much contention there is when the spike occurs. The instance is never left with zero CPU resources, whether other instances on the host are spiking or not.

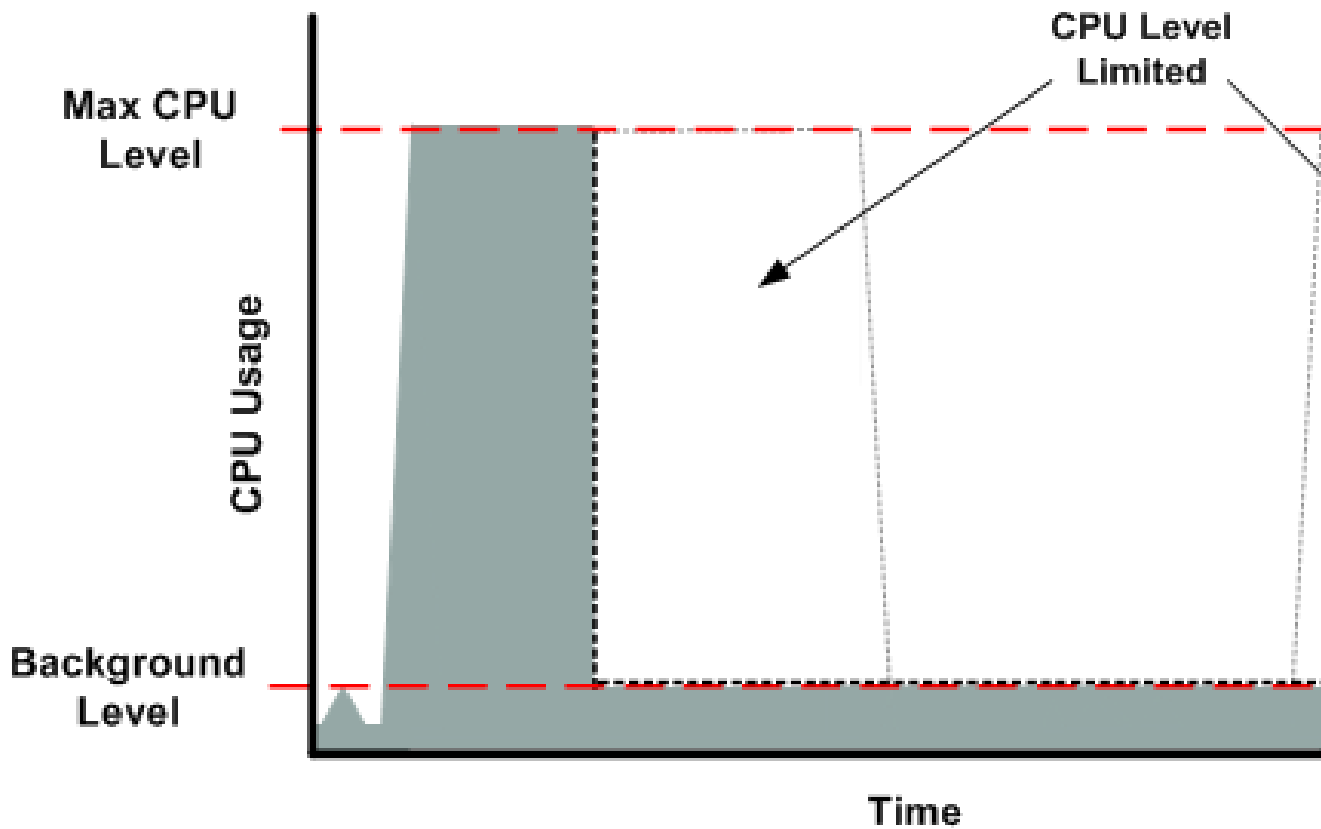
When the Instance Uses Its Allotted Resources

We expect your application to consume only a certain amount of CPU resources in a period of time. If the application consumes more than your instance's allotted CPU resources, we temporarily limit the instance so it operates at a low CPU level. If your instance continues to use all of its allotted resources, its performance will degrade. We will increase the time that we limit its CPU level, thus increasing the time before the instance is allowed to burst again.

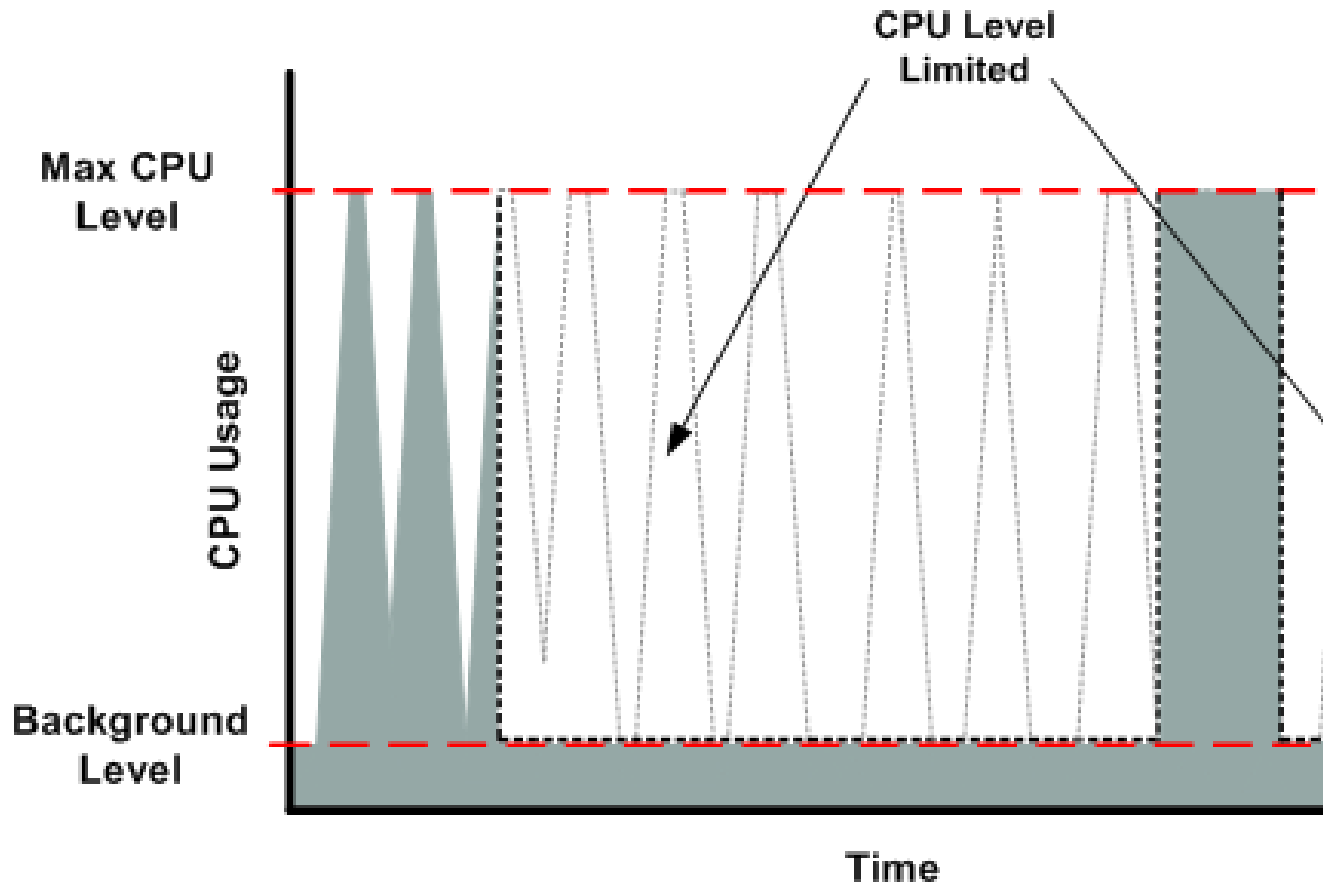
If you enable CloudWatch monitoring for your `t1.micro` instance, you can use the "Avg CPU Utilization" graph in the AWS Management Console to determine whether your instance is regularly using all its allotted CPU resources. We recommend that you look at the maximum value reached during each given period. If the maximum value is 100%, we recommend that you use Auto Scaling to scale out (with additional `t1.micro` instances and a load balancer), or move to a larger instance type. For more information, see the [Auto Scaling User Guide](#).

The following figures show the three suboptimal profiles from the preceding section and what it might look like when the instance consumes its allotted resources and we have to limit its CPU level. If the instance consumes its allotted resources, we restrict it to the low background level.

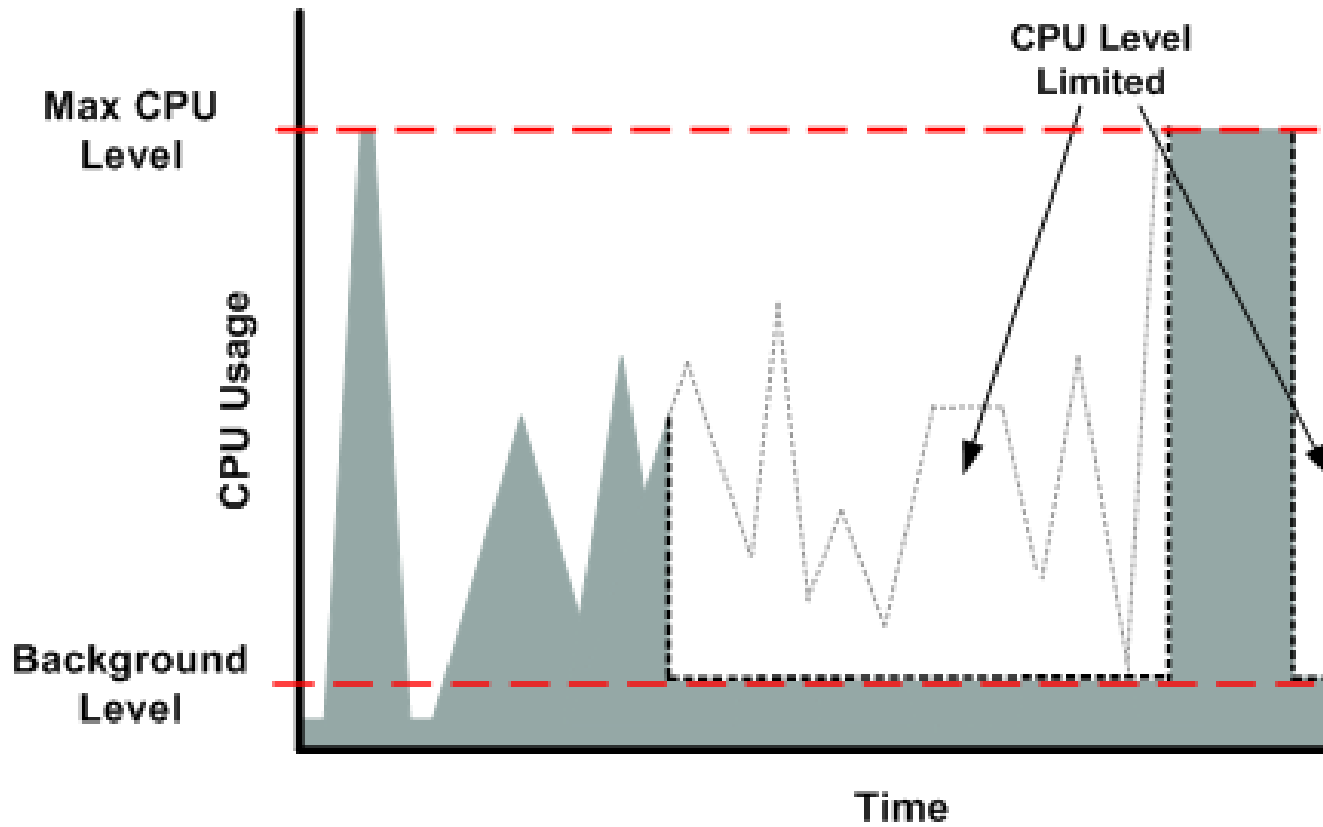
The next figure shows the situation with the long plateaus of data-crunching CPU usage. The CPU hits the maximum allowed level and stays there until the instance's allotted resources are consumed for the period. At that point, we limit the instance to operate at the low background level, and it operates there until we allow it to burst above that level again. The instance again stays there until the allotted resources are consumed and we limit it again (not seen on the graph).



The next figure shows the situation where the requests are too frequent. The instance uses its allotted resources after only a few requests and so we limit it. After we lift the restriction, the instance maxes out its CPU usage trying to keep up with the requests, and we limit it again.

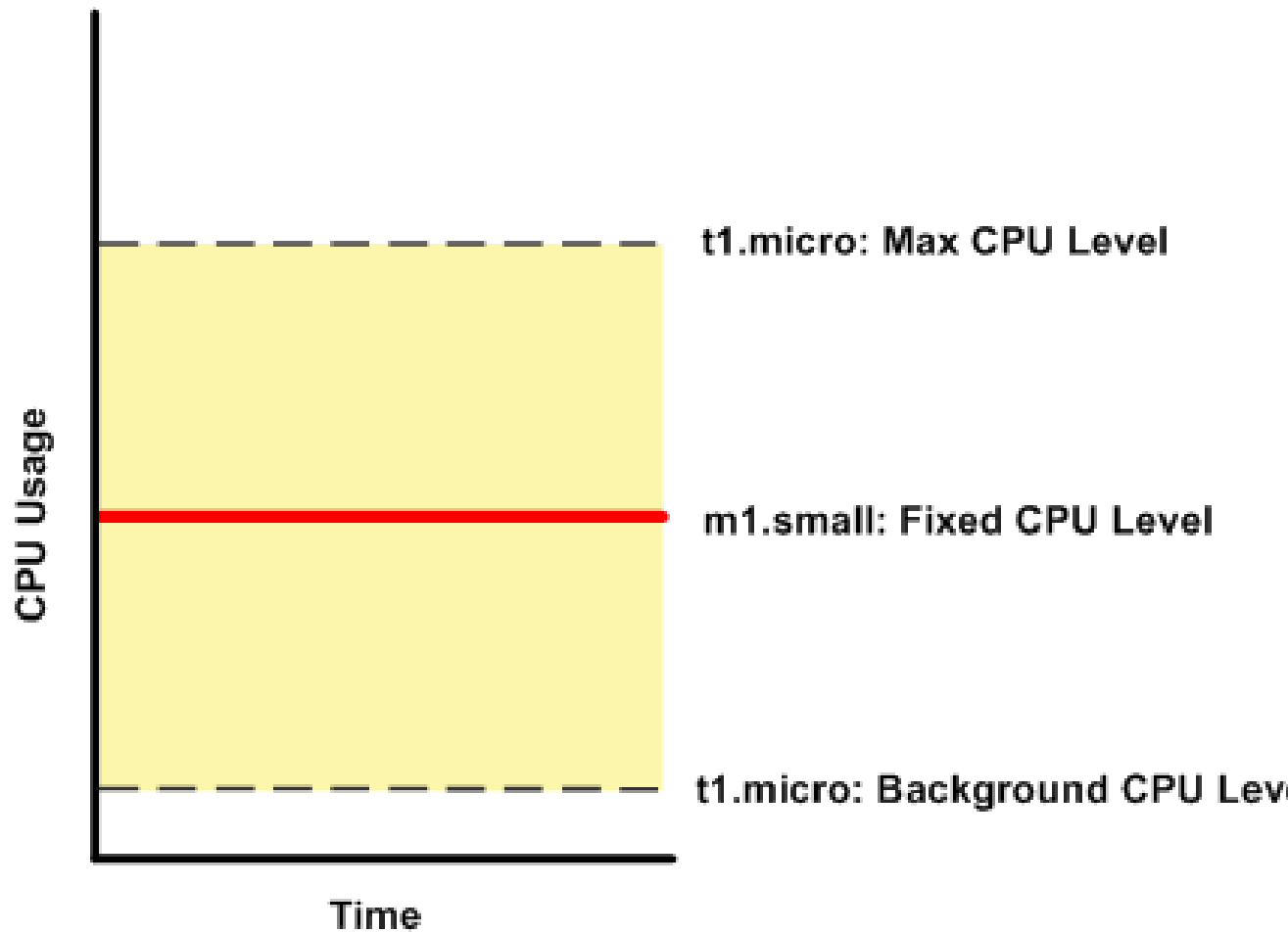


The next figure shows the situation where the background level is too high. Notice that the instance doesn't have to be operating at the maximum CPU level for us to limit it. We limit the instance when it's operating above the normal background level and has consumed its allotted resources for the given period. In this case (as in the preceding one), the instance can't keep up with the work, and we limit it again.



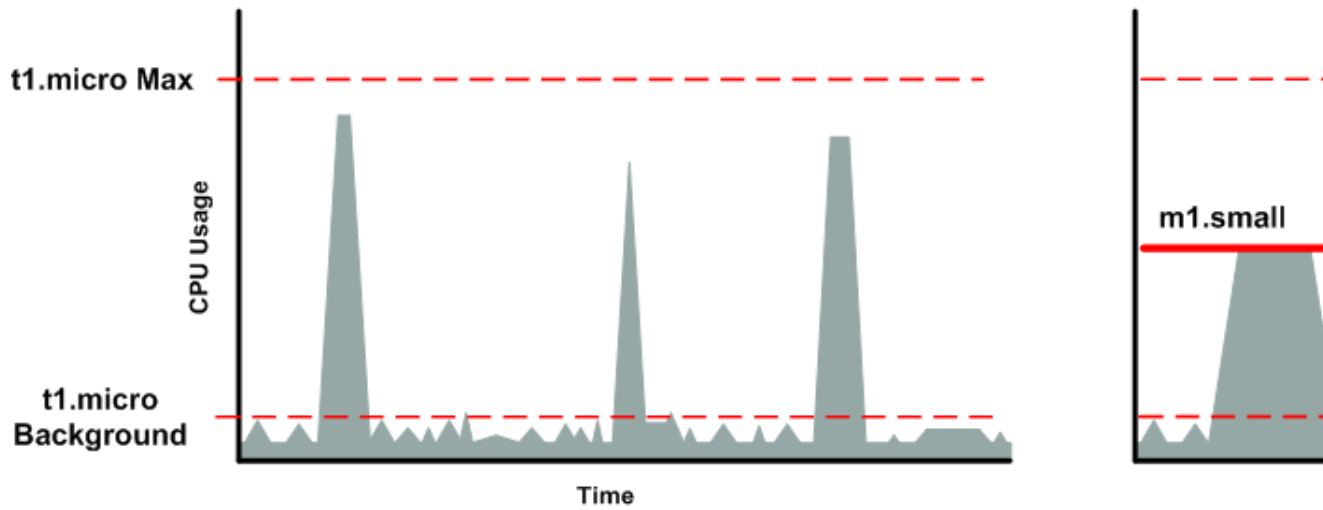
Comparison with the m1.small Instance Type

The `t1.micro` instance provides different levels of CPU resources at different times (up to 2 ECUs). By comparison, the `m1.small` instance type provides 1 ECU at all times. The following figure illustrates the difference.

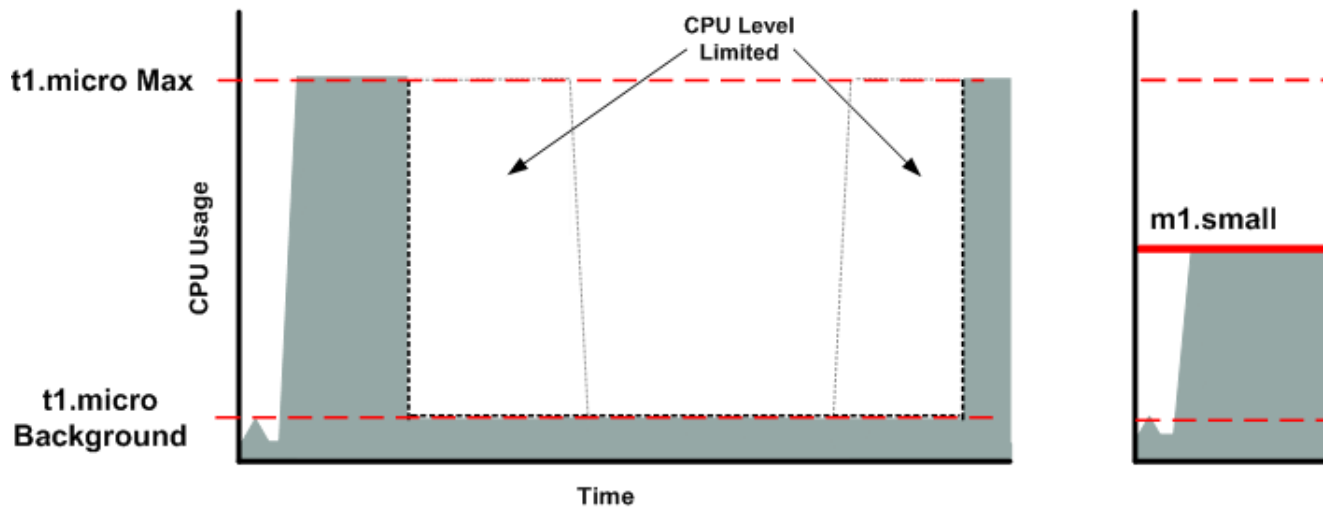


The following figures compare the CPU usage of a `t1.micro` instance with an `m1.small` instance for the various scenarios we've discussed in the preceding sections.

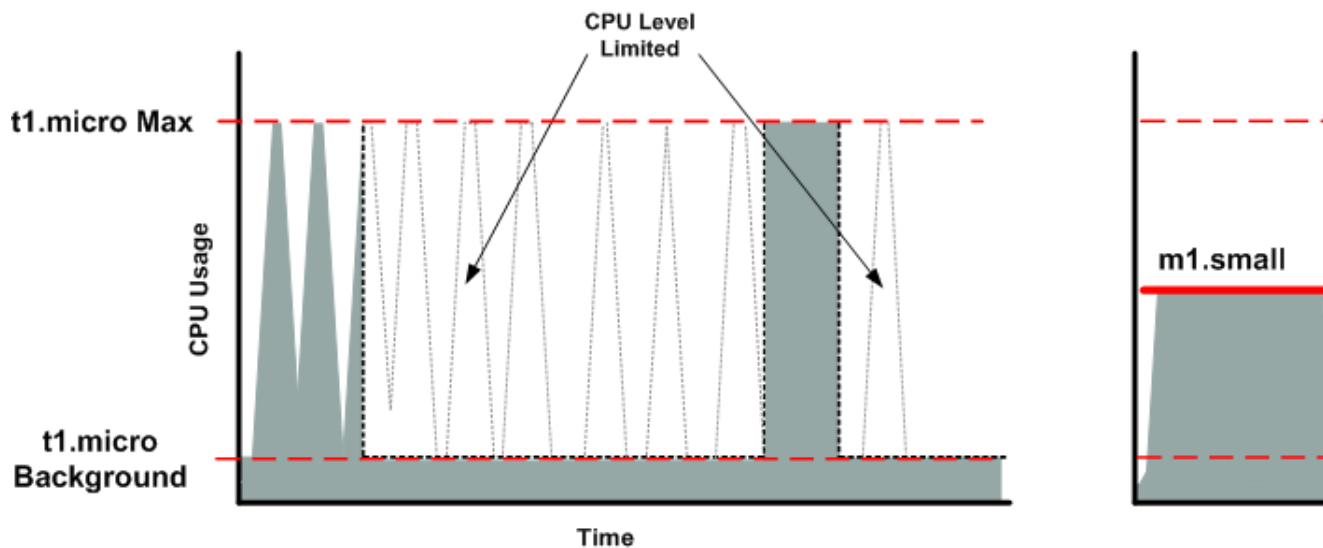
The first figure that follows shows an optimal scenario for a `t1.micro` instance (the left graph) and how it might look for an `m1.small` instance (the right graph). In this case, we don't need to limit the `t1.micro` instance. The processing time on the `m1.small` instance would be longer for each spike in CPU demand compared to the `t1.micro` instance.



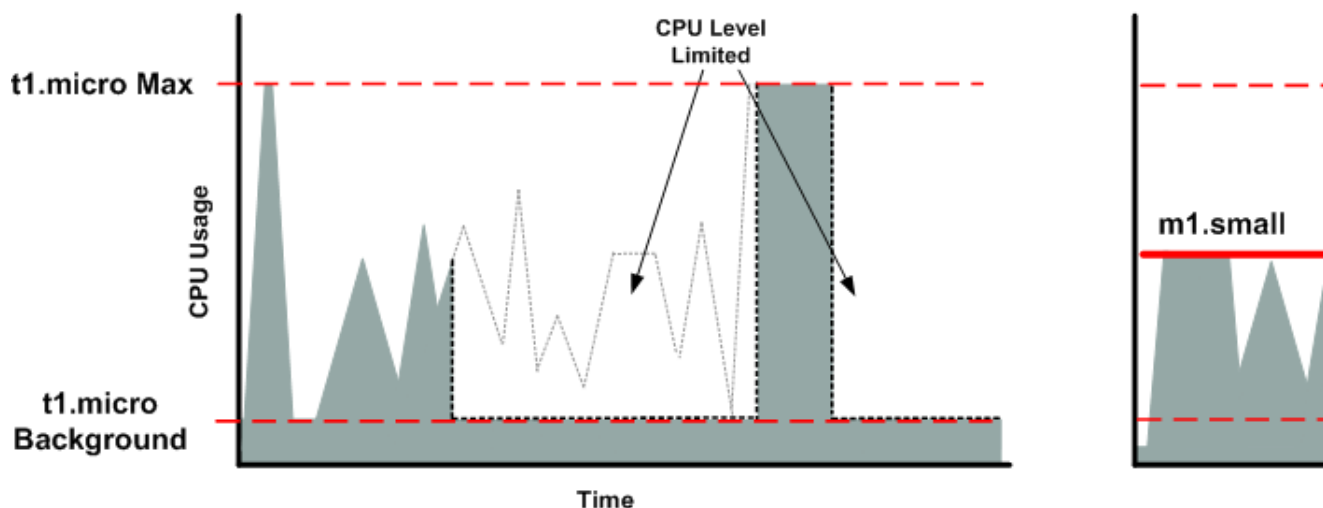
The next figure shows the scenario with the data-crunching requests that used up the allotted resources on the `t1.micro` instance, and how they might look with the `m1.small` instance.



The next figure shows the frequent requests that used up the allotted resources on the `t1.micro` instance, and how they might look on the `m1.small` instance.



The next figure shows the situation where the background level used up the allotted resources on the `t1.micro` instance, and how it might look on the `m1.small` instance.



AMI Optimization for Micro Instances

We recommend that you follow these best practices when optimizing an AMI for the `t1.micro` instance type:

- Design the AMI to run on 600 MB of RAM
- Limit the number of recurring processes that use CPU time (for example, cron jobs, daemons)

When you perform significant AMI or instance configuration changes (for example, enable server roles or install large applications), you might see limited instance performance, because these changes can be memory intensive and require long-running CPU resources. We recommend that you first use a

larger instance type when performing these changes to the AMI, and then run the AMI on a `t1.micro` instance for normal operations.

Resizing Your Instance

As your needs change, you might find that your instance is over-utilized (the instance type is too small) or under-utilized (the instance type is too large). If this is the case, you can change the size of your instance. For example, if your `t2.micro` instance is too small for its workload, you can change it to an `m3.medium` instance.

If the root device for your instance is an EBS volume, you can change the size of the instance simply by changing its instance type, which is known as *resizing* it. If the root device for your instance is an instance store volume, you must migrate your application to a new instance with the instance type that you want. For more information about root device volumes, see [Storage for the Root Device \(p. 64\)](#).

When you resize an instance, you must select an instance type that is compatible with the configuration of the instance. If the instance type that you want is not compatible with the instance configuration you have, then you must migrate your application to a new instance with the instance type that you want.

Important

When you resize an instance, the resized instance usually has the same number of instance store volumes that you specified when you launched the original instance. If you want to add instance store volumes, you must migrate your application to a completely new instance with the instance type and instance store volumes that you want. An exception to this rule is when you resize to a storage-intensive instance type that by default contains a higher number of volumes. For more information about instance store volumes, see [Amazon EC2 Instance Store \(p. 822\)](#).

Contents

- [Compatibility for Resizing Instances \(p. 147\)](#)
- [Resizing an Amazon EBS-backed Instance \(p. 148\)](#)
- [Migrating an Instance Store-backed Instance \(p. 149\)](#)
- [Migrating to a New Instance Configuration \(p. 150\)](#)

Compatibility for Resizing Instances

You can resize an instance only if its current instance type and the new instance type that you want are compatible in the following ways:

- **Network.** Some instance types are not supported in EC2-Classic and must be launched in a VPC. Therefore, you can't resize an instance in EC2-Classic to a instance type that is available only in a VPC unless you have a nondefault VPC. For more information, see [Instance Types Available Only in a VPC \(p. 671\)](#).
- **Platform.** All Amazon EC2 instance types support 64-bit AMIs, but only the following instance types support 32-bit AMIs: `t2.nano`, `t2.micro`, `t2.small`, `t2.medium`, `c3.large`, `t1.micro`, `m1.small`, `m1.medium`, and `c1.medium`. If you are resizing a 32-bit instance, you are limited to these instance types.

For example, T2 instances are not supported in EC2-Classic and they are HVM only. Therefore, you can't resize a T1 instance to a T2 instance because T1 instances do not support HVM and must be launched from PV AMIs. If you want to resize a T2 instance to a larger instance type, you can select any current generation instance type, such as M3, because all current generation instance types support HVM AMIs. For more information, see [Available Instance Types \(p. 118\)](#).

Resizing an Amazon EBS-backed Instance

You must stop your Amazon EBS-backed instance before you can change its instance type. When you stop and start an instance, be aware of the following:

- We move the instance to new hardware; however, the instance ID does not change.
- If your instance is running in a VPC and has a public IPv4 address, we release the address and give it a new public IPv4 address. The instance retains its private IPv4 addresses, any Elastic IP addresses, and any IPv6 addresses.
- If your instance is running in EC2-Classic, we give it new public and private IP addresses, and disassociate any Elastic IP address that's associated with the instance. Therefore, to ensure that your users can continue to use the applications that you're hosting on your instance uninterrupted, you must re-associate any Elastic IP address after you restart your instance.
- If your instance is in an Auto Scaling group, the Auto Scaling service marks the stopped instance as unhealthy, and may terminate it and launch a replacement instance. To prevent this, you can suspend the Auto Scaling processes for the group while you're resizing your instance. For more information, see [Suspend and Resume Auto Scaling Processes](#) in the *Auto Scaling User Guide*.

For more information, see [Stop and Start Your Instance \(p. 259\)](#).

Use the following procedure to resize an Amazon EBS-backed instance using the AWS Management Console.

To resize an Amazon EBS-backed instance

1. Open the Amazon EC2 console.
2. In the navigation pane, choose **Instances**, and select the instance.
3. [EC2-Classic] If the instance has an associated Elastic IP address, write down the Elastic IP address and the instance ID shown in the details pane.
4. Choose **Actions**, select **Instance State**, and then choose **Stop**.
5. In the confirmation dialog box, choose **Yes, Stop**. It can take a few minutes for the instance to stop.

[EC2-Classic] When the instance state becomes `stopped`, the **Elastic IP**, **Public DNS (IPv4)**, **Private DNS**, and **Private IPs** fields in the details pane are blank to indicate that the old values are no longer associated with the instance.

6. With the instance still selected, choose **Actions**, select **Instance Settings**, and then choose **Change Instance Type**. Note that this action is disabled if the instance state is not `stopped`.
7. In the **Change Instance Type** dialog box, do the following:
 - a. From **Instance Type**, select the instance type that you want. If the instance type that you want does not appear in the list, then it is not compatible with the configuration of your instance (for example, because of virtualization type).
 - b. (Optional) If the instance type that you selected supports EBS-optimization, select **EBS-optimized** to enable EBS-optimization or deselect **EBS-optimized** to disable EBS-optimization. Note that if the instance type that you selected is EBS-optimized by default, **EBS-optimized** is selected and you can't deselect it.
 - c. Choose **Apply** to accept the new settings.
8. To restart the stopped instance, select the instance, choose **Actions**, select **Instance State**, and then choose **Start**.
9. In the confirmation dialog box, choose **Yes, Start**. It can take a few minutes for the instance to enter the `running` state.

10. [EC2-Classical] When the instance state is `running`, the **Public DNS (IPv4)**, **Private DNS**, and **Private IPs** fields in the details pane contain the new values that we assigned to the instance. If your instance had an associated Elastic IP address, you must reassociate it as follows:
 - a. In the navigation pane, choose **Elastic IPs**.
 - b. Select the Elastic IP address that you wrote down before you stopped the instance.
 - c. Choose **Actions** and then choose **Associate Address**.
 - d. From **Instance**, select the instance ID that you wrote down before you stopped the instance, and then choose **Associate**.

Migrating an Instance Store-backed Instance

When you want to move your application from one instance store-backed instance to an instance store-backed instance with a different instance type, you must migrate it by creating an image from your instance, and then launching a new instance from this image with the instance type that you need. To ensure that your users can continue to use the applications that you're hosting on your instance uninterrupted, you must take any Elastic IP address that you've associated with your original instance and associate it with the new instance. Then you can terminate the original instance.

To migrate an instance store-backed instance

1. [EC2-Classical] If the instance you are migrating has an associated Elastic IP address, record the Elastic IP address now so that you can associate it with the new instance later.
2. Back up any data on your instance store volumes that you need to keep to persistent storage. To migrate data on your EBS volumes that you need to keep, take a snapshot of the volumes (see [Creating an Amazon EBS Snapshot \(p. 789\)](#)) or detach the volume from the instance so that you can attach it to the new instance later (see [Detaching an Amazon EBS Volume from an Instance \(p. 781\)](#)).
3. Create an AMI from your instance store-backed instance by satisfying the prerequisites and following the procedures in [Creating an Instance Store-Backed Windows AMI \(p. 80\)](#). When you are finished creating an AMI from your instance, return to this procedure.
4. Open the Amazon EC2 console and in the navigation pane, select **AMIs**. From the filter lists, select **Owned by me**, and select the image that you created in the previous step. Notice that **AMI Name** is the name that you specified when you registered the image and **Source** is your Amazon S3 bucket.

Note

If you do not see the AMI that you created in the previous step, make sure that you have selected the region in which you created your AMI.

5. Choose **Launch**. When you specify options for the instance, be sure to select the new instance type that you want. If the instance type that you want can't be selected, then it is not compatible with configuration of the AMI that you created (for example, because of virtualization type). You can also specify any EBS volumes that you detached from the original instance.

Note that it can take a few minutes for the instance to enter the `running` state.

6. [EC2-Classical] If the instance that you started with had an associated Elastic IP address, you must associate it with the new instance as follows:
 - a. In the navigation pane, choose **Elastic IPs**.
 - b. Select the Elastic IP address that you recorded at the beginning of this procedure.
 - c. Choose **Actions** and then choose **Associate Address**.
 - d. From **Instance**, select the new instance, and then choose **Associate**.
7. (Optional) You can terminate the instance that you started with, if it's no longer needed. Select the instance and verify that you are about to terminate the original instance, not the new instance

(for example, check the name or launch time). Choose **Actions**, select **Instance State**, and then choose **Terminate**.

Migrating to a New Instance Configuration

If the current configuration of your instance is incompatible with the new instance type that you want, then you can't resize the instance to that instance type. Instead, you can migrate your application to a new instance with a configuration that is compatible with the new instance type that you want.

If you want to move from an instance launched from a PV AMI to an instance type that is HVM only, the general process is as follows:

1. Back up any data on your instance store volumes that you need to keep to persistent storage. To migrate data on your EBS volumes that you need to keep, create a snapshot of the volumes (see [Creating an Amazon EBS Snapshot \(p. 789\)](#)) or detach the volume from the instance so that you can attach it to the new instance later (see [Detaching an Amazon EBS Volume from an Instance \(p. 781\)](#)).
2. Launch a new instance, selecting the following:
 - An HVM AMI.
 - The HVM only instance type.
 - [EC2-VPC] If you are using an Elastic IP address, select the VPC that the original instance is currently running in.
 - Any EBS volumes that you detached from the original instance and want to attach to the new instance, or new EBS volumes based on the snapshots that you created.
 - If you want to allow the same traffic to reach the new instance, select the security group that is associated with the original instance.
3. Install your application and any required software on the instance.
4. Restore any data that you backed up from the instance store volumes of the original instance.
5. If you are using an Elastic IP address, assign it to the newly launched instance as follows:
 - a. In the navigation pane, choose **Elastic IPs**.
 - b. Select the Elastic IP address that is associated with the original instance, choose **Actions**, and then choose **Disassociate Address**. When prompted for confirmation, choose **Yes, Disassociate**.
 - c. With the Elastic IP address still selected, choose **Actions**, and then choose **Associate Address**.
 - d. From **Instance**, select the new instance, and then choose **Associate**.
6. (Optional) You can terminate the original instance if it's no longer needed. Select the instance and verify that you are about to terminate the original instance, not the new instance (for example, check the name or launch time). Choose **Actions**, select **Instance State**, and then choose **Terminate**.

For information about migrating an application from an instance in EC2-Classic to an instance in a VPC, see [Migrating from a Windows Instance in EC2-Classic to a Windows Instance in a VPC \(p. 683\)](#).

Instance Purchasing Options

Amazon EC2 provides the following purchasing options to enable you to optimize your costs based on your needs:

- **On-Demand instances** — Pay, by the hour, for the instances that you launch.
- **Reserved Instances** — Purchase, at a significant discount, instances that are always available, for a term from one to three years.
- **Scheduled Instances** — Purchase instances that are always available on the specified recurring schedule, for a one-year term.
- **Spot instances** — Bid on unused instances, which can run as long as they are available and your bid is above the Spot price, at a significant discount.
- **Dedicated hosts** — Pay for a physical host that is fully dedicated to running your instances, and bring your existing per-socket, per-core, or per-VM software licenses to reduce costs.
- **Dedicated instances** — Pay, by the hour, for instances that run on single-tenant hardware.

If you require a capacity reservation, consider Reserved Instances or Scheduled Instances. Spot instances are a cost-effective choice if you can be flexible about when your applications run and if they can be interrupted. Dedicated hosts can help you address compliance requirements and reduce costs by using your existing server-bound software licenses. For more information, see [Amazon EC2 Instance Purchasing Options](#).

Contents

- [Determining the Instance Lifecycle \(p. 151\)](#)
- [Reserved Instances \(p. 152\)](#)
- [Scheduled Reserved Instances \(p. 176\)](#)
- [Spot Instances \(p. 180\)](#)
- [Dedicated Hosts \(p. 226\)](#)
- [Dedicated Instances \(p. 236\)](#)

Determining the Instance Lifecycle

The lifecycle of an instance starts when it is launched and ends when it is terminated. The purchasing option that you choose effects the lifecycle of the instance. For example, an On-Demand instance runs when you launch it and ends when you terminate it. A Spot instance runs as long as its capacity is available and your bid price is higher than the Spot price. You can launch a Scheduled Instance during its scheduled time period; Amazon EC2 launches the instances and then terminates them three minutes before the time period ends.

Use the following procedure to determine the lifecycle of an instance.

To determine the instance lifecycle using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance.
4. On the **Description** tab, find **Tenancy**. If the value is `host`, the instance is running on a Dedicated Host. If the value is `dedicated`, the instance is a Dedicated Instance.
5. On the **Description** tab, find **Lifecycle**. If the value is `spot`, the instance is a Spot instance. If the value is `scheduled`, the instance is a Scheduled Instance. If the value is `normal`, the instance is either an On-Demand instance or a Reserved Instance.
6. (Optional) If you have purchased a Reserved Instance and want to verify that it is being applied, you can check the usage reports for Amazon EC2. For more information, see [Reserved Instance Utilization Reports \(p. 876\)](#).

To determine the instance lifecycle using the AWS CLI

Use the following [describe-instances](#) command:

```
aws ec2 describe-instances --instance-ids i-1234567890abcdef0
```

If the instance is running on a Dedicated host, the output contains the following information:

```
"Tenancy": "host "
```

If the instance is a Dedicated instance, the output contains the following information:

```
"Tenancy": "dedicated"
```

If the instance is a Spot instance, the output contains the following information:

```
"InstanceLifecycle": "spot "
```

If the instance is a Scheduled Instance, the output contains the following information:

```
"InstanceLifecycle": "scheduled"
```

Otherwise, the output contains the following information:

```
"InstanceLifecycle": "normal "
```

Reserved Instances

Reserved Instances provide you with a significant discount compared to On-Demand instance pricing. In addition, when Reserved Instances are assigned to a specific Availability Zone, they provide a capacity reservation. For more information, see [Choosing a Reserved Instance Payment Option](#) (p. 156).

Reserved Instances are automatically applied to running EC2 instances that match the Reserved Instance attributes. So, if you purchased two `c4.xlarge, default` tenancy, Linux Reserved Instances for the US East (N. Virginia) region, up to two `c4.xlarge, default` tenancy, Linux instances running in the US East (N. Virginia) region can benefit from the Reserved Instance discount. The same logic applies if you've purchased a Reserved Instance for a specific Availability Zone; however, each Reserved Instance that is specific to an Availability Zone can also provide a capacity reservation.

When you purchase a Reserved Instance, choose a payment option, term, and an offering class that suits your needs. Generally speaking, you can save more money choosing Reserved Instances with a higher upfront payment. There are three payment options (No Upfront, Partial Upfront, All Upfront) and two term lengths (one-year or three-years). You can find Reserved Instances offered by third-party sellers at shorter term lengths and lower prices as well. The offering class is used to differentiate Convertible Reserved Instances and Standard Reserved Instances. Convertible Reserved Instances can be exchanged during the term for Convertible Reserved Instances with new attributes including instance type. Standard Reserved Instances can be modified during the term, but the instance type is fixed throughout the term.

When you purchase a Reserved Instance, the reservation is automatically applied to running instances that match your specified parameters. Alternatively, you can launch an On-Demand EC2 instance with the same configuration as the reservation, and the billing benefit is applied automatically. No Upfront

and Partial Upfront Reserved Instances are billed for usage on an hourly basis, regardless of whether or not they are being used. All Upfront Reserved Instances have no additional hourly charges.

Reserved Instances do not renew automatically; you can continue using the EC2 instance without interruption, but you will be charged On-Demand rates. New Reserved Instances can have the same parameters as the expired ones, or you can purchase Reserved Instances with different parameters.

You can use Auto Scaling or other AWS services to launch the On-Demand instances that use your Reserved Instance benefits. For information about launching On-Demand instances, see [Launch Your Instance](#). For information about launching instances using Auto Scaling, see the [Auto Scaling User Guide](#).

For product pricing information, see the following pages:

- [AWS Service Pricing Overview](#)
- [Amazon EC2 On-Demand Instances Pricing](#)
- [Amazon EC2 Reserved Instance Pricing](#)
- For information about the Reserved Instance pricing tiers, see [Understanding Reserved Instance Discount Pricing Tiers](#) (p. 157).

Note

Light, Medium, and Heavy Utilization Reserved Instances are no longer available for purchase. For more information about how these options are affected by changes to the Reserved Instances pricing model, see [Reserved Instances FAQ](#).

Topics

- [Types of Reserved Instances](#) (p. 153)
- [How Reserved Instances Work](#) (p. 154)
- [Billing Benefits and Payment Options](#) (p. 156)
- [Buying Reserved Instances](#) (p. 160)
- [Selling in the Reserved Instance Marketplace](#) (p. 163)
- [Modifying Your Standard Reserved Instances](#) (p. 169)
- [Exchanging Convertible Reserved Instances](#) (p. 174)
- [Troubleshooting Modification Requests](#) (p. 176)

Types of Reserved Instances

Standard Reserved Instances can be purchased for one-year or three-year terms and applied to a single instance family, platform, scope, and tenancy over a term.

Convertible Reserved Instances can be purchased for a three-year term and exchanged for Convertible Reserved Instances with different instance families, platform, tenancy, or scope during the term.

Both Standard and Convertible Reserved Instances can be purchased to apply to instances in a specific Availability Zone, or to instances in a region. Reserved Instances purchased for a specific Availability Zone can be modified to apply to a region—but doing so removes the associated capacity reservation.

Convertible Reserved Instances can be exchanged for other Convertible Reserved Instances with entirely different configurations, including instance type, platform, scope, or tenancy. It is not possible to exchange Standard Reserved Instances in this way. It is not possible to modify the scope of a Convertible Reserved Instance once it has been purchased. For more information, see [Modifying Your Standard Reserved Instances](#) (p. 169) and [Exchanging Convertible Reserved Instances](#) (p. 174).

How Reserved Instances Work

Amazon EC2 Reserved Instances and the Reserved Instance Marketplace can be a powerful, cost-saving strategy for running your business. However, before you use Reserved Instances or the Reserved Instance Marketplace, ensure that you meet the requirements for purchase and sale. You also must understand the details and restrictions on certain elements of Reserved Instances and the Reserved Instance Marketplace—including seller registration, banking, using the AWS free tier, dealing with cancelled instances, and so on. Use this topic as a checklist for buying and selling Reserved Instances, and for buying and selling in the Reserved Instance Marketplace.

Note

To purchase and modify Reserved Instances, ensure that your IAM user account has the appropriate permissions, such as the ability to describe Availability Zones. For information, see [Example Policies for Working With the AWS CLI or an AWS SDK](#) and [Example Policies for Working in the Amazon EC2 Console](#).

Getting Started

- **AWS account**—You need to have an AWS account in order to purchase Reserved Instances. If you don't have an AWS account, read and complete the instructions described in [Setting Up with Amazon EC2 \(p. 13\)](#), which provide information on signing up for your Amazon EC2 account and credentials.
- **AWS free tier**—The AWS free usage tier is available for new AWS accounts. If you are using the AWS free usage tier to run Amazon EC2 instances, and then you purchase a Reserved Instance, you are charged under standard pricing guidelines. For information about applicable services and usage amounts, see [AWS Free Tier](#).

Buying Reserved Instances

- **Usage fee**—With Reserved Instances, you pay for the entire term regardless of whether or not you use it.
- **Tiered discounts on purchases**—The Reserved Instance pricing tier discounts only apply to purchases made from AWS. These discounts do not apply to purchases of third-party Reserved Instances. For information, see [Understanding Reserved Instance Discount Pricing Tiers \(p. 157\)](#).
- **Cancellation of purchase**—Before you confirm your purchase, review the details of the Reserved Instances that you plan to buy, and make sure that all the parameters are accurate. After you purchase a Reserved Instance (either from a third-party seller in the Reserved Instance Marketplace or from AWS), you cannot cancel your purchase. However, you may be able to sell the Reserved Instance if your needs change. For information, see [Listing Your Reserved Instances \(p. 166\)](#).

Selling Reserved Instances and the Reserved Instance Marketplace

- **Convertible Reserved Instances**—Only Amazon EC2 Standard Reserved Instances can be sold in the Reserved Instance Marketplace. Convertible Reserved Instances cannot be sold.
- **Reserved Instances scope**—Only Standard Reserved Instances with a capacity reservation can be sold in the Reserved Instance Marketplace. Reserved Instances with a regional benefit cannot be sold.
- **Seller requirement**—To become a seller in the Reserved Instance Marketplace, you must register as a seller. For information, see [Listing Your Reserved Instances \(p. 166\)](#).
- **Bank requirement**—AWS must have your bank information in order to disburse funds collected when you sell your reservations. The bank you specify must have a US address. For more information, see [Bank Accounts \(p. 165\)](#).
- **Tax requirement**—Sellers who have 50 or more transactions or who plan to sell \$20,000 or more in Standard Reserved Instances will have to provide additional information about their business for tax reasons. For information, see [Tax Information \(p. 165\)](#).

- **Minimum selling price**—The minimum price allowed in the Reserved Instance Marketplace is \$0.00.
- **When Standard Reserved Instances can be sold**—Standard Reserved Instances can be sold only after AWS has received the upfront payment and the reservation has been active (you've owned it) for at least 30 days. In addition, there must be at least one month remaining in the term of the Standard Reserved Instance you are listing.
- **Modifying your listing**—It is not possible to modify your listing in the Reserved Instance Marketplace directly. However, you can change your listing by first cancelling it and then creating another listing with new parameters. For information, see [Pricing Your Reserved Instances \(p. 169\)](#). You can also modify your Reserved Instances before listing them. For information, see [Modifying Your Standard Reserved Instances \(p. 169\)](#).
- **Selling discounted Standard Reserved Instances**—Amazon EC2 Standard Reserved Instances purchased at a reduced cost resulting from a tiering discount cannot be sold in the Reserved Instance Marketplace. For more information, see [Reserved Instance Marketplace \(p. 155\)](#).
- **Service fee**—AWS charges a service fee of 12 percent of the total upfront price of each Standard Reserved Instance you sell in the Reserved Instance Marketplace. (The upfront price is the price the seller is charging for the Standard Reserved Instance.)
- **Other AWS Reserved Instances**—Only Amazon EC2 Standard Reserved Instances can be sold in the Reserved Instance Marketplace. Other AWS Reserved Instances, such as Amazon RDS and Amazon ElastiCache Reserved Instances cannot be sold in the Reserved Instance Marketplace.

Using Reserved Instances in a VPC

You can launch instances into an VPC and benefit from your Standard and Convertible Reserved Instances. For more information see [What is Amazon VPC?](#) in the *Amazon VPC User Guide*.

If you have an EC2-Classic account, you can purchase Reserved Instances to apply to instances launched into a nondefault VPC by selecting a platform that includes *Amazon VPC* in its name. For more information, see [Detecting Your Supported Platforms and Whether You Have a Default VPC](#).

If you have an EC2-VPC-only account, the listed platforms available do not include *Amazon VPC* in its name because all platforms have default subnets. When you launch an instance with the same configuration as the capacity you reserved, and that instance is launched into your default or nondefault VPC, the capacity reservation and billing benefits are automatically applied to your instance. For more information, see [Your Default VPC and Subnets](#) in the *Amazon VPC User Guide*.

You can also choose to purchase Reserved Instances that are physically isolated at the host hardware level by specifying *dedicated* as the instance tenancy. For more information, see [Dedicated Instances \(p. 236\)](#).

Reserved Instance Marketplace

The Reserved Instance Marketplace is a platform that supports the sale of third-party and AWS customers' unused Standard Reserved Instances, which vary in term lengths and pricing options. For example, an AWS customer may want to sell capacity after moving instances to a new AWS region, changing to a new instance type, or ending projects before the term expiration.

The Reserved Instance Marketplace provides increased selection and flexibility by allowing you to address your specific business and searching for Reserved Instances that most closely match your preferred combination of instance type, region, and duration. Amazon EC2 instances purchased on the Reserved Instance Marketplace offer the same capacity reservations as Standard Reserved Instances purchased directly from AWS.

Note

Only Amazon EC2 Standard Reserved Instances can be sold in the Reserved Instance Marketplace. Other types, such as Amazon RDS and Amazon ElastiCache Reserved Instances, cannot be sold on the Reserved Instance Marketplace.

Billing Benefits and Payment Options

You can purchase Reserved Instances with a billing and capacity benefit by specifying an Availability Zone, or with a regional billing benefit by purchasing a Reserved Instance for a region.

The billing benefit is automatically applied to matching running instances in that Availability Zone or region. You can also purchase Reserved Instances and then launch On-Demand instances with matching specifications—the billing benefit is automatically applied to those instances.

When you purchase a Reserved Instance for a specific Availability Zone, by default EC2 will create a capacity reservation in that Availability Zone matching the configuration of your Reserved Instance. Your capacity reservation will be used by the first instance launched from the account that owns the Reserved Instance with a matching configuration.

Choosing a Reserved Instance Payment Option

There are three payment options for Reserved Instances:

- **No Upfront**—You are billed a discounted hourly rate for every hour within the term, regardless of usage, and no upfront payment is required. This option is only available as a 1-year reservation for Standard Reserved Instances and a 3-year reservation for Convertible Reserved Instances.

Note

No Upfront Reserved Instances are based on a contractual obligation to pay monthly for the entire term of the reservation. For this reason, a successful billing history is required before an account is eligible to purchase No Upfront Reserved Instances.

- **Partial Upfront**—A portion of the cost must be paid upfront and the remaining hours in the term are billed at a discounted hourly rate, regardless of usage.
- **All Upfront**—Full payment is made at the start of the term, with no other costs incurred for the remainder of the term regardless of the number of hours used.

Understanding hourly billing

Reserved Instances are billed for every clock-hour during the term that you select, regardless of whether an instance is running or not. It's important to understand the difference between instance states and how these impact billing hours. For more information, see [Instance Lifecycle \(p. 241\)](#).

Reserved Instance billing benefits only apply to one instance-hour per clock-hour. An instance-hour begins when an instance is started and continues for 60 minutes or until the instance is stopped or terminated—whichever happens first. A clock-hour is defined as the standard 24-hour clock that runs from midnight to midnight, and is divided into 24 hours (for example, 1:00:00 to 1:59:59 is one clock-hour).

A new instance-hour begins after an instance has run for 60 continuous minutes, or if an instance is stopped and then started. Rebooting an instance does not reset the running instance-hour.

For example, if an instance is stopped and then started again during a clock-hour and continues running for two more clock-hours, the first instance-hour (before the restart) is charged at the discounted Reserved Instance rate. The next instance-hour (after restart) is charged at the On-Demand rate and the next two instance-hours are charged at the discounted Reserved Instance rate.

The [Reserved Instance Utilization Reports \(p. 876\)](#) section includes sample reports which illustrate the savings against running On-Demand instances. The [Reserved Instances FAQ](#) includes a sample list value calculation.

How to apply a Reserved Instance

Reserved Instances apply to usage in the same manner, irrespective of the type of Reserved Instance you use (Standard or Convertible).

To apply a Reserved Instance to a running instance, you can either modify an existing Reserved Instance or purchase a Reserved Instance by selecting the Availability Zone (such as us-east-1b) or region, instance type (such as `m3.large`), platform (such as Amazon Linux VPC), and tenancy (such as default) to match the configuration of the running instance.

Below is an example scenario where a customer is running the following On-Demand instances in account A:

- 4 x `m3.large` Linux, default tenancy instances in Availability Zone us-east-1a
- 2 x `c4.xlarge` Linux, default tenancy instances in Availability Zone us-east-1b
- 2 x `c4.xlarge` Linux, default tenancy instances in Availability Zone us-east-1c

The customer then purchases the following Reserved Instances in account A:

- 4 x `m3.large` Linux, default tenancy Reserved Instances in Availability Zone us-east-1a (capacity is reserved)
- 4 x `c4.xlarge` Linux, default tenancy Reserved Instances in us-east-1
- 1 x `d2.xlarge` Linux, default tenancy Reserved Instances in us-east-1c

The Reserved Instance benefits are applied in the following way:

- The discount and capacity reservation of the four `m3.large` Reserved Instances will be used by the four `m3.large` instances because the attributes (instance size, region, platform, tenancy) between them match.
- The discount of the four `c4.xlarge` Reserved Instances will be utilized by the 4 `c4.xlarge` instances because the attributes (instance size, region, platform, tenancy) between them match even though two different Availability Zones (us-east-1b and us-east-1c) are being used.
- As there are no `d2.xlarge` instances matching the configuration of the `d2.xlarge` Reserved Instance, the capacity reservation will be held for future use and the discount will not apply to usage.

Understanding Reserved Instance Discount Pricing Tiers

When your account qualifies for a discount pricing tier, it automatically receives discounts on upfront and hourly usage fees for all Reserved Instance purchases that you make within that tier level from that point on. To qualify for a discount, the list value of your Reserved Instances in the region must be \$500,000 USD or more.

Note

Discount pricing tiers are not currently applicable to Convertible Reserved Instance purchases.

Topics

- [Calculating Reserved Instance Pricing Discounts \(p. 157\)](#)
- [Consolidated Billing for Pricing Tiers \(p. 158\)](#)
- [Buying with a Discount Tier \(p. 158\)](#)
- [Current Pricing Tier Limits \(p. 159\)](#)
- [Crossing Pricing Tiers \(p. 159\)](#)

Calculating Reserved Instance Pricing Discounts

You can determine the pricing tier for your account by calculating the list value for all of your Reserved Instances in a region. Multiply the hourly recurring price for each reservation by the hours left in each term and add the undiscounted upfront price (known as the fixed price) listed on the [AWS marketing website](#) at the time of purchase. Because the list value is based on undiscounted (public) pricing, it

is not affected if you qualify for a volume discount or if the price drops after you buy your Reserved Instances.

```
List value = fixed price + (undiscounted recurring hourly price * hours in term)
```

To view the fixed price values for Reserved Instances using the AWS Management Console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Turn on the display of the **Fixed Price column** by choosing **Show/Hide** in the top right corner.

To view the fixed price values for Reserved Instances using the command line

- Using the AWS CLI, see [describe-reserved-instances](#)
- Using the AWS Tools for Windows PowerShell, see [Get-EC2ReservedInstance](#)
- Using the Amazon EC2 API, see [DescribeReservedInstances](#)

Consolidated Billing for Pricing Tiers

A consolidated billing account aggregates the list value of member accounts within a region. When the list value of all active Reserved Instances for the consolidated billing account reaches a discount pricing tier, any Reserved Instances purchased after this point by any member of the consolidated billing account are charged at the discounted rate (as long as the list value for that consolidated account stays above the discount pricing tier threshold). For more information, see [Reserved Instances and Consolidated Billing](#) (p. 159).

Buying with a Discount Tier

When you buy Reserved Instances, Amazon EC2 automatically applies any discounts to the part of your purchase that falls within a discount pricing tier. You don't need to do anything differently, and you can buy using any of the Amazon EC2 tools. For more information, see [Buying in the Reserved Instance Marketplace](#) (p. 163).

Note

Reserved Instance purchases are the only purchases that determine your discount pricing tiers, and the discounts apply only to Amazon EC2 Reserved Instance purchases.

After the list value of your active Reserved Instances in a region crosses into a discount pricing tier, any future purchase of Reserved Instances in that region are charged at a discounted rate. If a single purchase of Reserved Instances in a region takes you over the threshold of a discount tier, then the portion of the purchase that is above the price threshold is charged at the discounted rate. For more information about temporary Reserved Instance IDs created during the purchase process, see [Crossing Pricing Tiers](#) (p. 159).

If your list value falls below the price point for that discount pricing tier—for example, if some of your Reserved Instances expire—future purchases of Reserved Instances in the region are not discounted. However, you continue to get the discount applied against any Reserved Instances that were originally purchased within the discount pricing tier.

When you buy Reserved Instances, one of four possible scenarios occurs:

- **No discount**—Your purchase within a region is still below the discount threshold.
- **Partial discount**—Your purchase within a region crosses the threshold of the first discount tier. No discount is applied to one or more reservations and the discounted rate is applied to the remaining reservations.
- **Full discount**—Your entire purchase within a region falls within one discount tier and is discounted appropriately.

- **Two discount rates**—Your purchase within a region crosses from a lower discount tier to a higher discount tier. You are charged two different rates: one or more reservation at the lower discounted rate, and the remaining reservations at the higher discounted rate.

Current Pricing Tier Limits

The following limitations currently apply to Reserved Instance pricing tiers:

- Reserved Instance pricing tiers and related discounts apply only to purchases of Amazon EC2 Reserved Instances.
- Reserved Instance pricing tiers do not apply to Reserved Instances for Windows with SQL Server Standard or Windows with SQL Server Web.
- Reserved Instances purchased as part of a tiered discount cannot be sold in the Reserved Instance Marketplace. For more information, see the [Reserved Instance Marketplace \(p. 155\)](#) page.

Crossing Pricing Tiers

If your purchase crosses into a discounted pricing tier, you see multiple entries for that purchase: one for that part of the purchase charged at the regular price, and another for that part of the purchase charged at the applicable discounted rate.

The Reserved Instance service generates several Reserved Instance IDs because your purchase crossed from an undiscounted tier, or from one discounted tier to another. There is an ID for each set of reservations in a tier. Consequently, the ID returned by your purchase CLI command or API action will be different from the actual ID of the new Reserved Instances.

Reserved Instances and Consolidated Billing

The pricing benefits of Reserved Instances are shared when the purchasing account is part of a set of accounts billed under one consolidated billing payer account. The hourly usage across all sub-accounts is aggregated in the payer account every month. This is typically useful for companies in which there are different functional teams or groups; then, the normal Reserved Instance logic is applied to calculate the bill. For more information, see *Consolidated Billing* in [AWS Billing and Cost Management User Guide](#).

For more information about how the discounts of the Reserved Instance pricing tiers apply to consolidated billing accounts, see [Amazon EC2 Reserved Instances](#).

Reading Your Statement (Invoice)

You can find out about the charges and fees to your account by viewing the **Billing & Cost Management** page in the AWS Management Console. Choose the arrow beside your account name to access it.

- The **Dashboard** page displays charges against your account—such as upfront and one-time fees, and recurring charges. You can get both a summary and detailed list of your charges.
- The upfront charges from your purchase of third-party Reserved Instances in the Reserved Instance Marketplace are listed in the **AWS Marketplace Charges** section, with the name of the seller displayed beside it. All recurring or usage charges for these Reserved Instances are listed in the **AWS Service Charges** section.
- The **Detail** section contains information about the Reserved Instance—such as the Availability Zone, instance type, cost, and number of instances.

You can view the charges online, and you can also download a PDF rendering of the charge information.

Buying Reserved Instances

You can search for specific types of Reserved Instances to buy, adjusting your parameters until you find the exact match that you're looking for.

It's important to note the following for any Reserved Instance purchase:

- **Usage fee**—With Reserved Instances, you pay for the entire term regardless of actual use.
- **Tiered discounts on purchases**—Pricing tier discounts apply only to AWS Standard Reserved Instances purchases. These discounts do not apply to purchases of third-party Reserved Instances or to Convertible Reserved Instances. For more information, see [Understanding Reserved Instance Discount Pricing Tiers \(p. 157\)](#).
- **Cancellation of purchase**—After the purchase is confirmed, it cannot be cancelled. Before you confirm, review the details of the Reserved Instances that you plan to buy, and make sure that all the parameters are accurate. However, you may be able to sell the Reserved Instances if your needs change and you meet the requirements. For more information, see [Selling in the Reserved Instance Marketplace \(p. 163\)](#).

After you select Reserved Instances to buy, you receive a quote on the total cost of your selections. When you decide to proceed with the purchase, AWS automatically places a limit price on the purchase price, so that the total cost of your Reserved Instances does not exceed the amount that you were quoted.

If the price rises or changes for any reason, you are returned to the previous screen and the purchase is not completed. If, at the time of purchase, there are offerings similar to your choice but at a lower price, AWS sells you the offerings at the lower price.

Buying Standard Reserved Instances Using the AWS Management Console

You can buy Standard Reserved Instances with or without a capacity reservation. The default view lists Reserved Instances with a regional benefit. To purchase a capacity reservation choose **Only show offerings that reserve capacity** in the top-right corner of the purchase screen.

To buy Standard Reserved Instances with no capacity reservation using the AWS Management Console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the **Navigation** pane, choose **Reserved Instances**.
3. On the **Reserved Instances** page, choose **Purchase Reserved Instances**.
4. Select **Offering Class** and choose **Standard** to display Standard Reserved Instances.
5. Select other configurations as needed and choose **Search**.

Note

The **Seller** column in the search results indicates whether the seller is a third-party. If so, the **Term** column displays non-standard terms.

6. Select the Reserved Instances to purchase, enter the quantity, and choose **Add to Cart**.
7. To see a summary of the Reserved Instances that you selected, choose **View Cart**.
8. To complete the order, choose **Purchase**.

Note

If, at the time of purchase, there are offerings similar to your choice but with a lower price, AWS sells you the offerings at the lower price.

To apply your reservation, launch an On-Demand instance, ensuring that you match the same criteria that you specified for your Reserved Instance. AWS automatically charges you the lower hourly rate. You do not have to restart your instances.

To view transaction status using the AWS Management Console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose the **Reserved Instances** page. The status of your purchase is listed in the **State** column. When your order is complete, the **State** value changes from `payment-pending` to `active`.

Buying Convertible Reserved Instances Using the AWS Management Console

You can buy Convertible Reserved Instances with or without a capacity reservation. The default view lists Reserved Instances with a regional benefit. To purchase a capacity reservation choose **Only show offerings that reserve capacity** in the top-right corner of the purchase screen.

To buy Convertible Reserved Instances with no capacity reservation using the AWS Management Console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the **Navigation** pane, choose **Reserved Instances**.
3. On the **Reserved Instances** page, choose **Purchase Reserved Instances**.
4. Select **Offering Class** and choose **Convertible** to display Convertible Reserved Instances.
5. Select other configurations as needed and choose **Search**.
6. Select the Convertible Reserved Instances to purchase, enter the quantity, and choose **Add to Cart**.
7. To see a summary of your selection, choose **View Cart**.
8. To complete the order, choose **Purchase**.

Note

If, at the time of purchase, there are offerings similar to your choice but with a lower price, AWS sells you the offerings at the lower price.

The billing benefit is automatically applied to matching On-Demand instances with matching specifications, in the specified region. AWS automatically charges you the lower hourly rate. You do not have to restart your instances.

To view transaction status using the AWS Management Console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose the **Reserved Instances** page. The status of your purchase is listed in the **State** column. When your order is complete, the **State** value changes from `payment-pending` to `active`.

Buying Reserved Instances Using the Command Line Interface or API

To buy Reserved Instances using the command line or API

1. Using the AWS CLI, see [purchase-reserved-instances-offering](#)
2. Using the AWS Tools for Windows PowerShell, see [New-EC2ReservedInstance](#)
3. Using the Amazon EC2 API, see [PurchaseReservedInstancesOffering](#)

To view transaction status using the command line or API

1. Using the AWS CLI, see [describe-reserved-instances](#)
2. Using the AWS Tools for Windows PowerShell, see [Get-EC2ReservedInstance](#)

3. Using the Amazon EC2 API, see [DescribeReservedInstances](#)

Applying Reserved Instances

Reserved Instances are automatically applied to running On-Demand instances provided that the specifications match. You can use the AWS Management Console, a command line tool, or the Amazon EC2 API to perform any of these tasks.

Note

To purchase and modify Reserved Instances, ensure that your IAM user account has the appropriate permissions, such as the ability to describe Availability Zones. For information, see [Example Policies for Working With the AWS CLI or an AWS SDK](#) and [Example Policies for Working in the Amazon EC2 Console](#).

Purchase—Determine how much capacity to reserve. Specify the following criteria:

- Platform (for example, Linux).

Note

To use your Reserved Instance on a specific platform (e.g., Windows, Linux/Unix), you must identify the platform when you purchase the reserved capacity. Then, when you launch your instance with the intention of using the capacity you purchased, you must choose the Amazon Machine Image (AMI) that runs that specific platform, along with any other specifications that you identified during the purchase.

- Instance type (for example, `m1.small`).
- Scope of the reservation (**Region** or **Availability Zone**).
- Term (time period) over which to reserve capacity.
- Tenancy. You can reserve capacity for your instance to run in single-tenant hardware (`dedicated` tenancy, as opposed to `shared`). The tenancy you select must match the tenancy of the On-Demand instance to which you're applying, or plan to apply, the Reserved Instance. For more information, see [Dedicated Instances \(p. 236\)](#).
- Offering Class (**Standard** or **Convertible**).
- Offering (No Upfront, Partial Upfront, All Upfront).

Use—To use your Reserved Instance, launch an On-Demand instance with the same specifications as the reservation you purchased. The pricing benefits and capacity reservations automatically apply to any matching instances you have that aren't already covered by a reservation.

For more information, see [Launch Your Instance \(p. 244\)](#).

Reserved Instance States

Reserved Instances can be in one of the following states:

- `active`—The Reserved Instance is available for use.
- `payment-pending`—AWS is processing your payment for the Reserved Instance. You can use the Reserved Instance when the state becomes **active**.
- `retired`—The Reserved Instance has been terminated for any of the following reasons:
 - AWS did not receive your payment. For example, the credit card transaction did not go through.
 - The Reserved Instance term expired.

It's important to note that status information displayed in the **State** column in the **Reserved Instance** page is different from the status information displayed in the **Listing State** in the **My Listings** tab.

If you are a seller in the Reserved Instance Marketplace the **Listing State** displays the status of a reservation that's been listed in the Reserved Instance Marketplace. For more information, see [Reserved Instance Listing States \(p. 167\)](#).

Buying in the Reserved Instance Marketplace

Note

Convertible Reserved Instances are not available for purchase in the Reserved Instance Marketplace.

You can purchase Amazon EC2 Reserved Instances from AWS or you can purchase from third-party sellers who own Reserved Instances that they no longer need.

For a buyer, the Reserved Instance Marketplace provides increased selection and flexibility by allowing you to search for Reserved Instances that most closely match your preferred combination of instance type, region, and duration.

For more information about the Reserved Instance Marketplace, see [Selling in the Reserved Instance Marketplace \(p. 163\)](#).

There are a few differences between Reserved Instances purchased in the Reserved Instance Marketplace and Reserved Instances purchased directly from AWS:

- **Term**—Reserved Instances that you purchase from third-party sellers have less than a full standard term remaining. Full standard terms from AWS run for one year or three years.
- **Upfront price**—Third-party Reserved Instances can be sold at different upfront prices. The usage or recurring fees remain the same as the fees set when the Reserved Instances were originally purchased from AWS.

Basic information about you is shared with the seller, for example, your ZIP code and country information.

This information enables sellers to calculate any necessary transaction taxes that they have to remit to the government (such as sales tax or value-added tax) and is provided in the form of a disbursement report. In rare circumstances, AWS might have to provide the seller with your email address, so that they can contact you regarding questions related to the sale (for example, tax questions).

For similar reasons, AWS shares the legal entity name of the seller on the buyer's purchase invoice. If you need additional information about the seller for tax or related reasons, you can [contact AWS Support](#).

Selling in the Reserved Instance Marketplace

Note

Convertible Reserved Instances cannot be listed in the Reserved Instance Marketplace.

Selling unused reservations in the Reserved Instance Marketplace provides you with the flexibility to move to new configurations when your business needs change or if you have capacity you no longer need.

As soon as you list your Reserved Instances in the Reserved Instance Marketplace, they are available for potential buyers to find. All Reserved Instances are grouped according to the duration of the term remaining and the hourly price.

To fulfill a buyer's request, AWS first sells the Reserved Instance with the lowest upfront price in the specified grouping; then it sells the Reserved Instance with the next lowest price, until the buyer's entire order is fulfilled. AWS then processes the transactions and transfers ownership of the Reserved Instances to the buyer.

You own your Reserved Instance until it's sold. After the sale, you've given up the capacity reservation and the discounted recurring fees. If you continue to use your instance, AWS charges you the On-Demand price starting from the time that your Reserved Instance was sold. You can buy more reserved capacity, or terminate your instances when your capacity reservation is sold.

Contents

- [Getting Paid \(p. 164\)](#)
- [Registering as a Seller \(p. 164\)](#)
- [Listing Your Reserved Instances \(p. 166\)](#)
- [Pricing Your Reserved Instances \(p. 169\)](#)

Getting Paid

As soon as AWS receives funds from the buyer, a message is sent to the email address associated with the account that is registered as owner of the Reserved Instance that was sold.

AWS sends an Automated Clearing House (ACH) wire transfer to your specified bank account. Typically, this transfer occurs between one to three days after your Reserved Instance has been sold. You can view the state of this disbursement by viewing your Reserved Instance disbursement report. Disbursements take place once a day. Keep in mind that you will not be able to receive disbursements until AWS has received verification from your bank. This period can take up to two weeks.

The Reserved Instance you sold will continue to appear in the results of `DescribeReservedInstances` calls you make.

You receive a cash disbursement for your Reserved Instances through a wire transfer directly into your bank account. AWS charges a service fee of 12 percent of the total upfront price of each Reserved Instance you sell in the Reserved Instance Marketplace.

Note

Only Amazon EC2 Reserved Instances can be sold in the Reserved Instance Marketplace. Other types, such as Amazon RDS and Amazon ElastiCache Reserved Instances, cannot be sold on the Reserved Instance Marketplace.

The following are important limits to note:

- **Reserved Instances can be sold after 30 days**—Reserved Instances can only be sold when you've owned them for at least 30 days. In addition, there must be at least a month remaining in the term of the Reserved Instance you are listing.
- **Listings cannot be modified**—You cannot modify your listing in the Reserved Instance Marketplace. However, you can change your listing by first cancelling it and then creating another listing with new parameters. For information, see [Listing Your Reserved Instances \(p. 166\)](#). You can also modify your Reserved Instances before listing them. For information, see [Modifying Your Standard Reserved Instances \(p. 169\)](#).
- **Discounted Reserved instances cannot be sold**—Reserved Instances purchased at a reduced cost resulting from a tiering discount cannot be sold in the Reserved Instance Marketplace. For more information, see [Reserved Instance Marketplace \(p. 155\)](#).

Registering as a Seller

To be able to sell in the Reserved Instance Marketplace, your first task is to register as a seller. During registration, you need to provide the name of your business, information about your bank, and your business's tax identification number.

After AWS receives your completed seller registration, you will receive an email confirming your registration and informing you that you can get started selling in the Reserved Instance Marketplace.

Topics

- [Bank Accounts \(p. 165\)](#)
- [Tax Information \(p. 165\)](#)
- [Sharing Information with the Buyer \(p. 166\)](#)

Bank Accounts

AWS must have your bank information in order to disburse funds collected when you sell your Reserved Instance. The bank you specify must have a US address.

To register a default bank account for disbursements

1. On the [Reserved Instance Marketplace Seller Registration](#) page, sign in. If you do not yet have an AWS account you can also create one via this page.
2. On the **Manage Bank Account** page, provide the following information about the bank through which you will receive payment:

- Bank account holder name
- Routing number
- Account number
- Bank account type

Note

If you are using a corporate bank account, you are prompted to send the information about the bank account via fax (1-206-765-3424).

After registration, the bank account provided is set as the default, pending verification with the bank. It can take up to two weeks to verify a new bank account, during which time you will not be able to receive disbursements. For an established account, it usually takes about two days for disbursements to complete.

To change the default bank account for disbursement

1. On the [Reserved Instance Marketplace Seller Registration](#) page, sign in with the account that you used when you registered.
2. On the **Manage Bank Account** page, add a new bank account or modify the default bank account as needed.

Tax Information

Your sale of Reserved Instances might be subject to a transactional tax, such as sales tax or value-added tax. You should check with your business's tax, legal, finance, or accounting department to determine if transaction-based taxes are applicable. You are responsible for collecting and sending the transaction-based taxes to the appropriate tax authority.

As part of the seller registration process, you have the option of completing a tax interview. We encourage you to complete this process if any of the following apply:

- You want AWS to generate a Form 1099-K.
- You anticipate having either 50 or more transactions or \$20,000 or more in sales of Reserved Instances in a calendar year. A transaction can involve one or more Reserved Instances. If you choose to skip this step during registration, and later you reach transaction 49, you will get a message saying, "You have reached the transaction limit for pre-tax. Please complete the tax interview in the [Seller Registration Portal](#)." Once the tax interview is completed, the account limit is automatically increased.

- You are a non-US seller. In this case, you must electronically complete Form W-8BEN.

For more information about IRS requirements and the Form 1099-K, see the [IRS website](#).

The tax information you enter as part of the tax interview will differ depending on whether your business is a US or non-US legal entity. As you fill out the tax interview, keep in mind the following:

- Information provided by AWS, including the information in this topic, does not constitute tax, legal, or other professional advice. To find out how the IRS reporting requirements might affect your business, or if you have other questions, please contact your tax, legal, or other professional advisor.
- To fulfill the IRS reporting requirements as efficiently as possible, answer all questions and enter all information requested during the interview.
- Check your answers. Avoid misspellings or entering incorrect tax identification numbers. They can result in an invalidated tax form.

After you complete the tax registration process, AWS files Form 1099-K. You will receive a copy of it through the US mail on or before January 31 in the year following the year that your tax account reaches the threshold levels. For example, if your tax account reaches the threshold in 2016, you will receive the form in 2017.

Sharing Information with the Buyer

When you sell in the Reserved Instance Marketplace, AWS shares your company's legal name on the buyer's statement in accordance with US regulations. In addition, if the buyer calls AWS Customer Support because the buyer needs to contact you for an invoice or for some other tax-related reason, AWS may need to provide the buyer with your email address so that the buyer can contact you directly.

For similar reasons, the buyer's ZIP code and country information are provided to the seller in the disbursement report. As a seller, you might need this information to accompany any necessary transaction taxes that you remit to the government (such as sales tax and value-added tax).

AWS cannot offer tax advice, but if your tax specialist determines that you need specific additional information, [contact AWS Support](#).

Listing Your Reserved Instances

As a registered seller, you can choose to sell one or more of your Reserved Instances, and you can choose to sell all of them in one listing or in portions. In addition, you can list any type of Reserved Instance—including any configuration of instance type, platform, region, and Availability Zone.

If you decide to cancel your listing and a portion of that listing has already been sold, the cancellation is not effective on the portion that has been sold. Only the portion of the listing not yet sold will no longer be available in the Reserved Instance Marketplace.

Lifecycle of a Listing

Now that you have created a listing, let's walk through what happens when your listing sells.

When all the instances in your listing are matched and sold, the **My Listings** tab shows that your **Total instance count** matches the count listed under **Sold**, there are no **Available** instances left for your listing, and its **Status** is `closed`.

When only a portion of your listing is sold, AWS retires the Reserved Instances in the listing and creates the number of Reserved Instances equal to the Reserved Instances remaining in the count. So, the listing ID and the listing that it represents, which now has fewer reservations for sale, is still active.

Any future sales of Reserved Instances in this listing are processed this way. When all the Reserved Instances in the listing are sold, AWS marks the listing as `closed`.

For example, let's say you created a listing *Reserved Instances listing ID 5ec28771-05ff-4b9b-aa31-9e57dexample* with a listing count of 5.

Your **My Listings** tab in the **Reserved Instance** page of the AWS Management Console will display the listing this way:

Reserved Instance listing ID 5ec28771-05ff-4b9b-aa31-9e57dexample

- Total reservation count = 5
- Sold = 0
- Available = 5
- Status = active

Let's say that a buyer purchases two of the reservations, which leaves a count of three reservations still available for sale. As a result of this partial sale, AWS creates a new reservation with a count of three to represent the remaining reservations that are still for sale.

This is how your listing will look in your **My Listings** tab:

Reserved Instance listing ID 5ec28771-05ff-4b9b-aa31-9e57dexample

- Total reservation count = 5
- Sold = 2
- Available = 3
- Status = active

If you decide to cancel your listing and a portion of that listing has already sold, the cancellation is not effective on the portion that has been sold. Only the portion of the listing not yet sold will no longer be available in the Reserved Instance Marketplace.

[After Your Reserved Instance Is Sold](#)

When your Reserved Instance is sold, AWS will send you an email notification. Each day that there is any kind of activity (for example, you create a listing; you sell a listing; or AWS sends funds to your account), you will receive one email notification capturing all the activities of the day.

You can track the status of your Reserved Instance listings by looking at the **My Listings** tab of the selected Reserved Instance on the **Reserved Instance** page in the AWS Management Console. The tab contains the **Listing State** as well as information about the term, listing price, and a breakdown of how many instances in the listing are available, pending, sold, and cancelled. You can also use the `ec2-describe-reserved-instances-listings` CLI command or the `DescribeReservedInstancesListings` API call, with the appropriate filter to obtain information about your listings.

[Reserved Instance Listing States](#)

Listing State displays the current status of your listings:

The information displayed by **Listing State** is about the status of your listing in the Reserved Instance Marketplace. It is different from the status information that is displayed by the **State** column in the **Reserved Instances** page. This **State** information is about your reservation. For more information, see [Reserved Instance States](#) (p. 162).

[Listing your Reserved Instances using the AWS CLI](#)

To list a Reserved Instance in the Reserved Instance Marketplace using the AWS CLI

1. Get a list of your Reserved Instances by calling `aws ec2 describe-reserved-instances`.
2. Specify the ID of the Reserved Instance you want to list and call `aws ec2 create-reserved-instances-listing`. You have to specify the following required parameters:

- Reserved Instance ID
- Instance count
- MONTH:PRICE

To view your listing

- Run `aws ec2 describe-reserved-instances-listings` to get details about your listing.

To cancel and change your listing

- Run `aws ec2 cancel-reserved-instances-listings` to cancel your listing.

Listing Your Reserved Instances using the Amazon EC2 API

To list a Reserved Instance in the Reserved Instance Marketplace using the Amazon EC2 API

1. Get a list of your Reserved Instances by calling `DescribeReservedInstances`. Note the ID of the Reserved Instance to list in the Reserved Instance Marketplace.
2. Create a listing using `CreateReservedInstancesListing`.

To view your listing

1. Call `DescribeReservedInstancesListings` to get details about your listing.

To cancel your listing

1. Run `CancelReservedInstancesListing`.
2. Confirm that it's cancelled by calling `DescribeReservedInstancesListings`.

Listing Your Reserved Instance using the AWS Management Console

You can list the Reserved Instances you want to sell in the Reserved Instance Marketplace by using the AWS Management Console.

To list a Reserved Instance in the Reserved Instance Marketplace using the AWS Management Console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Reserved Instances**.
3. Select the Reserved Instances to list, and choose **Sell Reserved Instances**.
4. On the **Configure Your Reserved Instance Listing** page, set the number of instances to sell and the upfront price for the remaining term in the relevant columns. You can see how the value of your reservation will change over the remainder of the term by clicking the arrow next to the **Months Remaining** column.
5. If you are an advanced user and you want to customize the pricing, you can enter different values for the subsequent months. To return to the default linear price drop, choose **Reset**.
6. Choose **Continue** when you are finished configuring your listing.

7. Confirm the details of your listing, on the **Confirm Your Reserved Instance Listing** page and if you're satisfied, choose **List Reserved Instance**.

Listing State displays the current status of your listings:

- **active**—The listing is available for purchase.
- **cancelled**—The listing is cancelled and won't be available for purchase in the Reserved Instance Marketplace.
- **closed**—The Reserved Instance is not listed. A Reserved Instance might be `closed` because the sale of the listing was completed.

Pricing Your Reserved Instances

The upfront fee is the only fee that you can specify for the Reserved Instance that you're selling. The upfront fee is the one-time fee that the buyer pays when they purchase a Reserved Instance. You cannot specify the usage fee or the recurring fee; The buyer will pay the same usage or recurring fees that were set when the reservations were originally purchased.

The following are important limits to note:

- **You can sell up to \$50,000 in Reserved Instances per year.** If you need to sell more, complete the [Request to Raise Sales Limit on Amazon EC2 Reserved Instances](#) form.
- **The minimum price is \$0.** The minimum allowed price allowed in the Reserved Instance Marketplace is \$0.00.

You cannot modify your listing directly. However, you can change your listing by first cancelling it and then creating another listing with new parameters.

You can cancel your listing at any time, as long as it's in the `actived` state. You cannot cancel the listing if it's already matched or being processed for a sale. If some of the instances in your listing are matched and you cancel the listing, only the remaining unmatched instances are removed from the listing.

Setting a Pricing Schedule

Because the value of Reserved Instances decreases over time, by default, AWS can set prices to decrease in equal increments month over month. However, you can set different upfront prices based on when your reservation sells.

For example, if your Reserved Instance has nine months of its term remaining, you can specify the amount you would accept if a customer were to purchase that Reserved Instance with nine months remaining, and you could set another price with five months remaining, and yet another price with one month remaining.

Modifying Your Standard Reserved Instances

When your computing needs change, you can modify your Standard Reserved Instances and continue to benefit from the billing benefit. Convertible Reserved Instances can be adjusted using the exchange process. For more information, see [Exchanging Convertible Reserved Instances \(p. 174\)](#).

The following topics guide you through the modification process for Standard Reserved Instances:

Topics

- [Requirements for Modification \(p. 170\)](#)
- [Modifying the Instance Size of Your Reservations \(p. 171\)](#)
- [Submitting Modification Requests \(p. 172\)](#)

Modification does not change the remaining term of your Standard Reserved Instances; their end dates remain the same. There is no fee, and you do not receive any new bills or invoices. Modification is separate from purchasing and does not affect how you use, purchase, or sell Standard Reserved Instances. You can modify your whole reservation, or just a subset, in one or more of the following ways:

- Change Availability Zones within the same region
- Change the scope of the reservation from Availability Zone to Region (and vice-versa)
- Change between EC2-VPC and EC2-Classic
- Change the instance size within the same instance type

Availability Zone, scope, and network platform modifications are supported for all product platform types (Linux and Windows). Instance type modifications are supported only for the Linux platform types. However, due to licensing differences, it is not possible to change the instance type of RedHat or SUSE Linux Standard Reserved Instances. For more information about RedHat and SUSE pricing, see [Amazon EC2 Reserved Instance Pricing](#).

If you change the Availability Zone of a reservation, the capacity reservation and pricing benefits are automatically applied to instance usage in the new Availability Zone. If you modify the network platform of a Reserved Instance (for example, from EC2-Classic to EC2-VPC) the capacity reservation is automatically applied to instance usage on the new network platform.

If you change the scope of a reservation from Availability Zone to Region, you no longer receive a capacity reservation benefit. The billing benefit of the reservation is applied to all applicable instances in that region.

After modification, the pricing benefit of the Reserved Instances is applied only to instances that match the new parameters. Instances that no longer match the new parameters are charged at the On-Demand rate unless your account has other applicable reservations. Pricing benefits apply to both EC2-Classic and EC2-VPC instances that match the specifications of the reservation.

Requirements for Modification

Amazon EC2 processes your modification request if there is sufficient capacity for your target configuration, and if the following conditions are met.

Your modified Reserved Instances must be:

- Active
- Not pending another modification request
- Not listed in the Reserved Instance Marketplace
- Terminating in the same hour (but not minutes or seconds)

Your modification request must be:

- A unique combination of scope, instance type, instance size, offering class, and network platform attributes
- A match between the instance size footprint of the active reservation and the target configuration

Limitations

- Only Standard Reserved Instances can be modified.

If your Reserved Instances are not in the active state or cannot be modified, the **Modify Reserved Instances** button in the AWS Management Console is not enabled. If you select multiple Reserved

Instances for modification and one or more are for a product platform that does not allow instance type modification, the **Modify Reserved Instances** page does not show the option of changing the instance type of any of the selected Reserved Instances. For more information, see [Modifying the Instance Size of Your Reservations](#) (p. 171).

You may modify your reservations as frequently as you like; however, you cannot submit a modification request for reservations that are pending a previous modification request. Also, you cannot change or cancel a pending modification request after you submit it. After the modification has completed successfully, you can submit another modification request to roll back any changes you made. For more information, see [Determining the Status of Your Modification](#) (p. 173).

To modify Reserved Instances that are listed in the Reserved Instance Marketplace, cancel the listing, request modification, and then list them again. In addition, you cannot modify an offering before or at the same time that you purchase it. For more information, see [Reserved Instance Marketplace](#) (p. 155).

Modifying the Instance Size of Your Reservations

You can adjust the instance size of your Reserved Instances if you have Amazon Linux reservations in an instance type with multiple sizes. Keep in mind that instance size modifications are allowed only if other attributes—such as region, utilization type, tenancy, product, end date, and hour—match and if capacity is available. It is not possible to modify the instance size of Windows Reserved Instances.

Note

Instances are grouped by family (based on storage, or CPU capacity); type (designed for specific use cases); and size. For example, the `c4` instance type is in the Compute optimized instance family and is available in multiple sizes. While `c3` instances are in the same family, you can't modify `c4` instances into `c3` instances because they have different hardware specifications. For more information, see [Amazon EC2 Instance Types](#).

For information about the modification process and steps, see [Submitting Modification Requests](#) (p. 172).

The following instances cannot be modified because there are no other sizes available.

- `t1.micro`
- `cc1.4xlarge`
- `cc2.8xlarge`
- `cg1.8xlarge`
- `cr1.8xlarge`
- `hi1.4xlarge`
- `hs1.8xlarge`
- `g2.2xlarge`

Your request is successful if the capacity exists and the modification does not change the instance size footprint of your Reserved Instances.

Understanding the Instance Size Footprint

Each Reserved Instance has an instance size footprint, which is determined by the normalization factor of the instance type and the number of instances in the reservation. In the Amazon EC2 console, the footprint is measured in units.

The normalization factor is based on size within the instance type (e.g., the `m1` instance family), and is only meaningful within the same instance type; instance types cannot be modified from one type to another. The following table illustrates the normalization factor that applies within an instance type.

Instance size	Normalization factor
nano	0.25
micro	0.5
small	1
medium	2
large	4
xlarge	8
2xlarge	16
4xlarge	32
8xlarge	64
10xlarge	80
16xlarge	128
32xlarge	256

A modification request is not processed if the footprint of the target configuration does not match the size of the original configuration.

To calculate the instance size footprint of a Reserved Instance, multiply the number of instances by the normalization factor. For example, an `m1.medium` has a normalization factor of 2 so a reservation for four `m1.medium` instances has a footprint of 8 units.

You can allocate your reservations into different instance sizes across the same instance type as long as the instance size footprint of your reservation remains the same. For example, you can divide a reservation for one `m1.large` (1 x 4) instance into four `m1.small` (4 x 1) instances, or you can combine a reservation for four `m1.small` instances into one `m1.large` instance. However, you cannot change your reservation for two `m1.small` (2 x 1) instances into one `m1.large` (1 x 4) instance because the existing instance size footprint of your current reservation is smaller than the proposed reservation.

For more information, see [Amazon EC2 Instance Types](#).

Submitting Modification Requests

AWS provides you with several ways to view and work with modification requests: You can use the AWS Management Console, interact directly with the Amazon EC2 API, or use the command line interface.

Topics

- [AWS Management Console](#) (p. 172)
- [Command Line Interface](#) (p. 173)
- [Amazon EC2 API](#) (p. 173)
- [Determining the Status of Your Modification](#) (p. 173)

AWS Management Console

Each target configuration row on the **Modify Reserved Instances** page keeps track of the number of instances for the current instance type (**Count**) and the instance size footprint of your reservation

relative to its instance type (**Units**). For more information, see [Understanding the Instance Size Footprint](#) (p. 171).

The allocated total is displayed in red if you have specified either more or fewer Reserved Instances than are available for modification. The total changes to green and you can choose **Continue** after you have specified changes for all the Reserved Instances that were available for modification.

When you modify a subset of your reservation, Amazon EC2 splits your original Reserved Instances into two or more new Reserved Instances. For example, if you have reservations for 10 instances in us-east-1a, and decide to move 5 instances to us-east-1b, the modification request results in two new reservations—one for 5 instances in us-east-1a (the original Availability Zone), and the other for 5 instances in us-east-1b.

To modify your Reserved Instances using the AWS Management Console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the **Reserved Instances** page, select one or more Reserved Instances to modify, and choose **Modify Reserved Instances**.

Note

The first entry in the modification table is the original, unmodified reservation. To modify the attributes of all reservations, choose new specifications from the menus. To modify or split only some of your reservations, add an additional line for each change.

3. Choose **Add** for each additional attribute change and enter the number of reservations to modify for **Count**.
 - To change the Availability Zone, select a value in the **Availability Zone** list.
 - To change the network platform, select a value in the **Network** list.
 - To change the instance type, select a value in the **Instance Type** list.
4. To delete a specified attribute, choose **X** for that row.

Note

If the **Modify Reserved Instances** page contains only one row for attribute changes, you cannot delete that row. To modify multiple Reserved Instance attributes, first add a row for the new specifications and then delete the original row.

5. Choose **Continue**.
6. To confirm your modification choices when you finish specifying your target configurations, choose **Submit Modifications**. If you change your mind at any point, choose **Cancel** to exit the wizard.

Command Line Interface

You can complete modification tasks programmatically by using the AWS CLI ([modify-reserved-instances](#)), the AWS Tools for Windows PowerShell ([Edit-EC2ReservedInstance](#)) the Amazon EC2 API ([ModifyReservedInstances](#)), and the [AWS SDK for Java](#).

Amazon EC2 API

You can use the [ModifyReservedInstances](#) action to modify your Reserved Instances. For more information, see [Amazon EC2 API Reference](#).

Determining the Status of Your Modification

You can determine the status of your modification request by looking at the **state** of the Reserved Instances that you are modifying. The state returned shows your request as `in-progress`, `fulfilled`, or `failed`. Use the following resources to get this information:

- The **State** field in the AWS Management Console
- The [DescribeReservedInstancesModifications](#) API action
- The [describe-reserved-instances-modifications](#) AWS CLI command

- The [Get-EC2ReservedInstancesModifications](#) AWS Tools for Windows PowerShell command

The following table illustrates the possible **State** values in the AWS Management Console.

State	Description
active (pending modification)	Transition state for original Reserved Instances.
retired (pending modification)	Transition state for original Reserved Instances while new Reserved Instances are being created.
retired	Reserved Instances successfully modified and replaced.
active	New Reserved Instances created from a successful modification request. -Or- Original Reserved Instances after a failed modification request.

Note

If you use the [DescribeReservedInstancesModifications](#) API action, the status of your modification request should show *processing*, *fulfilled*, or *failed*.

If your modification request succeeds:

- The modified reservation becomes effective immediately and the pricing benefit is applied to the new instances beginning at the hour of the modification request. For example, if you successfully modify your reservations at 9:15PM, the pricing benefit transfers to your new instance at 9:00PM. (You can get the `effective_date` of the modified Reserved Instances by using the [DescribeReservedInstances](#) API action or the `describe-reserved-instances` command (AWS CLI).
- The original reservation is retired. Its end date is the start date of the new reservation, and the end date of the new reservation is the same as the end date of the original Reserved Instance. If you modify a three-year reservation that had 16 months left in its term, the resulting modified reservation is a 16-month reservation with the same end date as the original one.
- The modified reservation lists a \$0 fixed price and not the fixed price of the original reservation.

Note

The fixed price of the modified reservation does not affect the discount pricing tier calculations applied to your account, which are based on the fixed price of the original reservation.

If your modification request fails:

- Your Reserved Instances maintain their original configuration.
- Your Reserved Instances are immediately available for another modification request.

For more information about why some Reserved Instances cannot be modified, see [Requirements for Modification](#) (p. 170).

Exchanging Convertible Reserved Instances

You can exchange Convertible Reserved Instances for other Convertible Reserved Instances with different configurations, including instance family. There are no limits to how many times you perform

an exchange, as long as the target Convertible Reserved Instances are of a higher value than the Convertible Reserved Instances that you are exchanging.

Requirements for Exchanging Convertible Reserved Instances

Amazon EC2 processes your exchange request if the following conditions are met.

Your Convertible Reserved Instances must be:

- Active
- Not pending another exchange request
- Terminating in the same hour (but not minutes or seconds)

Limitations:

- Convertible Reserved Instances can only be exchanged for other Convertible Reserved Instances currently offered by AWS.
- Convertible Reserved Instances cannot be modified. To change the reservation's configuration, you need to exchange it for another one.
- Convertible Reserved Instances can only be exchanged with the same or higher payment option. For example, Partial Upfront Convertible Reserved Instances can be exchanged for All Upfront Convertible Reserved Instances—but they cannot be exchanged for No Upfront Convertible Reserved Instances.

If your Convertible Reserved Instances are not in the active state or cannot be exchanged, the **Exchange Reserved Instances** button in the AWS Management Console is not enabled.

You may exchange your reservations as frequently as you like; however, you cannot submit an exchange request for reservations that are pending a previous exchange request.

Calculating Convertible Reserved Instances Exchanges

Exchanging Convertible Reserved Instances is free; however, you may be required to pay a true-up cost, which is a prorated upfront cost of the difference between the Convertible Reserved Instances that you had and the Convertible Reserved Instances that you receive as a result of the exchange.

Each Convertible Reserved Instance has a list value. This list value is compared to the list value of the Convertible Reserved Instances that you want in order to determine how many reservations you can receive as a result of the exchange.

For example: You have 1 x \$35-list value Convertible Reserved Instance which you want to exchange for a new instance type with a list value of \$10.

$$\$35/\$10 = 3.5$$

You can exchange your Convertible Reserved Instance for three \$10 Convertible Reserved Instances. It's not possible to purchase half reservations, so in this scenario you need to purchase an additional Convertible Reserved Instance to cover the remainder:

$$3.5 = 3 \text{ whole Convertible Reserved Instances} + 1 \text{ additional Convertible Reserved Instance.}$$

The fourth Convertible Reserved Instance has the same end date as the other three, and you pay the true-up cost for the fourth reservation if you are exchanging Partial or All Upfront Convertible Reserved

Instances. If the remaining upfront cost of your Convertible Reserved Instances is \$500, and the target reservation would normally cost \$600 on a prorated basis, you are charged \$100.

`$600 prorated upfront cost of new reservations - $500 remaining upfront cost of original reservations = $100 difference.`

Troubleshooting Modification Requests

If the target configuration settings that you requested were unique, you receive a message that your request is being processed. At this point, Amazon EC2 has only determined that the parameters of your modification request are valid. Your modification request can still fail during processing due to unavailable capacity.

In some situations, you might get a message indicating incomplete or failed modification requests instead of a confirmation. Use the information in such messages as a starting point for resubmitting another modification request.

Not all selected Reserved Instances can be processed for modification

Amazon EC2 identifies and lists the Reserved Instances that cannot be modified. If you receive a message like this, go to the **Reserved Instances** page in the AWS Management Console and check the information details about these capacity reservations.

Error in processing your modification request

You submitted one or more Reserved Instances for modification and none of your requests can be processed. Depending on the number of reservations you are modifying, you can get different versions of the message.

Amazon EC2 displays the reasons why your request cannot be processed. For example, you might have specified the same target configuration—a combination of Availability Zone and platform—for one or more subsets of the Reserved Instances you are modifying. Try submitting these modification requests again, but ensure that the instance details of the reservations match, and that the target configurations for all subsets being modified are unique.

Scheduled Reserved Instances

Scheduled Reserved Instances (Scheduled Instances) enable you to purchase capacity reservations that recur on a daily, weekly, or monthly basis, with a specified start time and duration, for a one-year term. You reserve the capacity in advance, so that you know it is available when you need it. You pay for the time that the instances are scheduled, even if you do not use them.

Scheduled Instances are a good choice for workloads that do not run continuously, but do run on a regular schedule. For example, you can use Scheduled Instances for an application that runs during business hours or for batch processing that runs at the end of the week.

If you require a capacity reservation on a continuous basis, Reserved Instances might meet your needs and decrease costs. For more information, see [Reserved Instances \(p. 152\)](#). If you are flexible about when your instances run, Spot instances might meet your needs and decrease costs. For more information, see [Spot Instances \(p. 180\)](#).

Contents

- [How Scheduled Instances Work \(p. 177\)](#)
- [Purchasing a Scheduled Instance \(p. 177\)](#)
- [Launching a Scheduled Instance \(p. 178\)](#)

- [Scheduled Instance Limits \(p. 179\)](#)

How Scheduled Instances Work

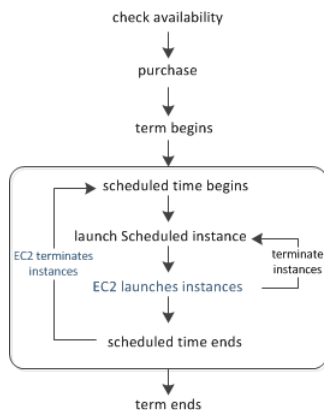
Amazon EC2 sets aside pools of EC2 instances in each Availability Zone for use as Scheduled Instances. Each pool supports a specific combination of instance type, operating system, and network (EC2-Classic or EC2-VPC).

To get started, you must search for an available schedule. You can search across multiple pools or a single pool. After you locate a suitable schedule, purchase it.

You must launch your Scheduled Instances during their scheduled time periods, using a launch configuration that matches the following attributes of the schedule that you purchased: instance type, Availability Zone, network, and platform. When you do so, Amazon EC2 launches EC2 instances on your behalf, based on the specified launch specification. Amazon EC2 must ensure that the EC2 instances have terminated by the end of the current scheduled time period so that the capacity is available for any other Scheduled Instances it is reserved for. Therefore, Amazon EC2 terminates the EC2 instances three minutes before the end of the current scheduled time period.

You can't stop or reboot Scheduled Instances, but you can terminate them manually as needed. If you terminate a Scheduled Instance before its current scheduled time period ends, you can launch it again after a few minutes. Otherwise, you must wait until the next scheduled time period.

The following diagram illustrates the lifecycle of a Scheduled Instance.



Purchasing a Scheduled Instance

To purchase a Scheduled Instance, you can use the Scheduled Reserved Instances Reservation Wizard.

Warning



After you purchase a Scheduled Instance, you can't cancel, modify, or resell your purchase.

To purchase a Scheduled Instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **INSTANCES**, choose **Scheduled Instances**.
3. Choose **Purchase Scheduled Instances**.
4. On the **Find available schedules** page, do the following:
 - a. Under **Create a schedule**, select the starting date from **Starting on**, the schedule recurrence (daily, weekly, or monthly) from **Recurring**, and the minimum duration from **for duration**.

Note that the console ensures that you specify a value for the minimum duration that meets the minimum required utilization for your Scheduled Instance (1,200 hours per year).

Create a schedule

Starting on  for duration  hours

+/- 2 hours

Recurring

- b. Under **Instance details**, select the operating system and network from **Platform**. To narrow the results, select one or more instance types from **Instance type** or one or more Availability Zones from **Availability Zone**.

Instance details

Platform Instance type

Availability Zone

- c. Choose **Find schedules**.
 - d. Under **Available schedules**, select one or more schedules. For each schedule that you select, set the quantity of instances and choose **Add to Cart**.
 - e. Your cart is displayed at the bottom of the page. When you are finished adding and removing schedules from your cart, choose **Review and purchase**.
5. On the **Review and purchase** page, verify your selections and edit them as needed. When you are finished, choose **Purchase**.

To purchase a Scheduled Instance using the AWS CLI

Use the [describe-scheduled-instance-availability](#) command to list the available schedules that meet your needs, and then use the [purchase-scheduled-instances](#) command to complete the purchase.

Launching a Scheduled Instance

After you purchase a Scheduled Instance, it is available for you to launch during its scheduled time periods.

To launch a Scheduled Instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **INSTANCES**, choose **Scheduled Instances**.
3. Select the Scheduled Instance and choose **Launch Scheduled Instances**.
4. On the **Configure** page, complete the launch specification for your Scheduled Instances and choose **Review**.

Important

The launch specification must match the instance type, Availability Zone, network, and platform of the schedule that you purchased.

5. On the **Review** page, verify the launch configuration and modify it as needed. When you are finished, choose **Launch**.

To launch a Scheduled Instance using the AWS CLI

Use the [describe-scheduled-instances](#) command to list your Scheduled Instances, and then use the [run-scheduled-instances](#) command to launch each Scheduled Instance during its scheduled time periods.

Scheduled Instance Limits

Scheduled Instances are subject to the following limits:

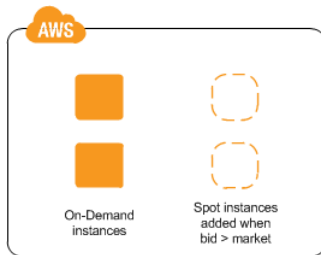
- The following are the only supported instance types: C3, C4, M4, and R3.
- The required term is 365 days (one year).
- The minimum required utilization is 1,200 hours per year.
- You can purchase a Scheduled Instance up to three months in advance.

Spot Instances

Spot instances enable you to bid on unused EC2 instances, which can lower your Amazon EC2 costs significantly. The hourly price for a Spot instance (of each instance type in each Availability Zone) is set by Amazon EC2, and fluctuates depending on the supply of and demand for Spot instances. Your Spot instance runs whenever your bid exceeds the current market price.

Spot instances are a cost-effective choice if you can be flexible about when your applications run and if your applications can be interrupted. For example, Spot instances are well-suited for data analysis, batch jobs, background processing, and optional tasks. For more information, see [Amazon EC2 Spot Instances](#).

The key differences between Spot instances and On-Demand instances are that Spot instances might not start immediately, the hourly price for Spot instances varies based on demand, and Amazon EC2 can terminate an individual Spot instance as the hourly price for or availability of Spot instances changes. One strategy is to launch a core group of On-Demand instances to maintain a minimum level of guaranteed compute resources for your applications, and supplement them with Spot instances when the opportunity arises.



Another strategy is to launch Spot instances with a required duration (also known as Spot blocks), which are not interrupted due to changes in the Spot price. For more information, see [Specifying a Duration for Your Spot Instances \(p. 192\)](#).

Concepts

Before you get started with Spot instances, you should be familiar with the following concepts:

- **Spot instance pool**—A set of unused EC2 instances with the same instance type, operating system, Availability Zone, and network platform (EC2-Classic or EC2-VPC).
- **Spot price**—The current market price of a Spot instance per hour, which is set by Amazon EC2 based on the last fulfilled bid. You can also retrieve the Spot price history.
- **Spot instance request (or Spot bid)**—Provides the maximum price (bid price) that you are willing to pay per hour for a Spot instance. When your bid price exceeds the Spot price, Amazon EC2 fulfills your request. Note that a Spot instance request is either *one-time* or *persistent*. Amazon EC2 automatically resubmits a persistent Spot request after the Spot instance associated with the request is terminated. Your Spot instance request can optionally specify a duration for the Spot instances.
- **Spot fleet**—A set of Spot instances that is launched based on criteria that you specify. The Spot fleet selects the Spot instance pools that meet your needs and launches Spot instances to meet the target capacity for the fleet. By default Spot fleets are set to *maintain* target capacity by launching replacement instances after Spot instances in the fleet are terminated. They can also be submitted as a one-time *request* which does not persist once instances have been terminated.
- **Spot instance interruption**—Amazon EC2 terminates your Spot instance when the Spot price exceeds your bid price or there are no longer any unused EC2 instances. Amazon EC2 marks the Spot instance for termination and provides a Spot instance termination notice, which gives the instance a two-minute warning before it terminates.

- *Bid status*—Provides detailed information about the current state of your Spot bid.

How to Get Started

The first thing you need to do is get set up to use Amazon EC2. It can also be helpful to have experience launching On-Demand instances before launching Spot instances.

Get Up and Running

- [Setting Up with Amazon EC2](#) (p. 13)
- [Getting Started with Amazon EC2 Windows Instances](#) (p. 20)

Spot Basics

- [How Spot Instances Work](#) (p. 182)
- [How Spot Fleet Works](#) (p. 185)

Working with Spot Instances

- [Preparing for Interruptions](#) (p. 222)
- [Creating a Spot Instance Request](#) (p. 193)
- [Getting Bid Status Information](#) (p. 220)

Working with Spot Fleets

- [Spot Fleet Prerequisites](#) (p. 199)
- [Creating a Spot Fleet Request](#) (p. 201)

Related Services

You can provision Spot instances directly using Amazon EC2. You can also provision Spot instances using other services in AWS. For more information, see the following documentation.

Auto Scaling and Spot instances

You can create launch configurations with a bid price so that Auto Scaling can launch Spot instances. For more information, see [Launching Spot instances in Your Auto Scaling Group](#) in the *Auto Scaling User Guide*.

Amazon EMR and Spot instances

There are scenarios where it can be useful to run Spot instances in an Amazon EMR cluster. For more information, see [Lower Costs with Spot Instances](#) in the *Amazon EMR Developer Guide*.

AWS CloudFormation Templates

AWS CloudFormation enables you to create and manage a collection of AWS resources using a template in JSON format. AWS CloudFormation templates can include a Spot price. For more information, see [EC2 Spot Instance Updates - Auto Scaling and CloudFormation Integration](#).

AWS SDK for Java

You can use the Java programming language to manage your Spot instances. For more information, see [Tutorial: Amazon EC2 Spot Instances](#) and [Tutorial: Advanced Amazon EC2 Spot Request Management](#).

AWS SDK for .NET

You can use the .NET programming environment to manage your Spot instances. For more information, see [Tutorial: Amazon EC2 Spot instances](#).

Pricing

You pay the Spot price for Spot instances, which is set by Amazon EC2 and fluctuates periodically depending on the supply of and demand for Spot instances. If your bid price exceeds the current Spot price, Amazon EC2 fulfills your request and your Spot instances run until either you terminate them or the Spot price increases above your bid price.

Everyone pays that same Spot price for that period, regardless of whether their bid price was higher. You never pay more than your bid price per hour, and often pay less per hour. For example, if you bid \$0.25 per hour, and the Spot price is \$0.20 per hour, you only pay \$0.20 per hour. If the Spot price drops, you pay the new, lower price. If the Spot price rises, you pay the new price if it is equal to or less than your bid price. If the Spot price rises above your bid price, then your Spot instance is interrupted.

At the start of each instance hour, you are charged based on the Spot price. If your Spot instance is interrupted in the middle of an instance hour because the Spot price exceeded your bid, you are not charged for the hour of use that was interrupted. However, if you terminate your Spot instance in the middle of an instance hour, you are charged for the hour.

Note that Spot instances with a predefined duration use a fixed hourly price that remains in effect for the Spot instance while it runs.

View Prices

To view the current (updated every five minutes) lowest Spot price per region and instance type, see the [Spot Instances Pricing](#) page.

To view the Spot price history for the past three months, use the Amazon EC2 console or the [describe-spot-price-history](#) command (AWS CLI). For more information, see [Spot Instance Pricing History](#) (p. 190).

Note that we independently map Availability Zones to codes for each AWS account. Therefore, you can get different results for the same Availability Zone code (for example, `us-west-2a`) between different accounts.

View Billing

To review your bill, go to your [AWS Account Activity page](#). Your bill contains links to usage reports that provide details about your bill. For more information, see [AWS Account Billing](#).

If you have questions concerning AWS billing, accounts, and events, [contact AWS Support](#).

How Spot Instances Work

To use Spot instances, create a *Spot instance request* or a *Spot fleet request*. The request includes the maximum price that you are willing to pay per hour per instance (your bid price), and other constraints such as the instance type and Availability Zone. If your bid price is greater than the current Spot price for the specified instance, and the specified instance is available, your request is fulfilled immediately. Otherwise, the request is fulfilled whenever the Spot price falls below your bid price or the specified instance becomes available. Spot instances run until you terminate them or until Amazon EC2 must terminate them (also known as a *Spot instance interruption*).

When you use Spot instances, you must be prepared for interruptions. Amazon EC2 can interrupt your Spot instance when the Spot price rises above your bid price, when the demand for Spot instances rises, or when the supply of Spot instances decreases. When Amazon EC2 marks a Spot instance for termination, it provides a Spot instance termination notice, which gives the instance a two-minute warning before it terminates. Note that you can't enable termination protection for Spot instances. For more information, see [Spot Instance Interruptions](#) (p. 221).

Note that you can't stop and start an Amazon EBS-backed instance if it is a Spot instance, but you can reboot or terminate it.

Shutting down a Spot instance on OS-level results in the Spot instance being terminated. It is not possible to change this behavior.

Contents

- [Supply and Demand in the Spot Market \(p. 183\)](#)
- [Launching Spot Instances in a Launch Group \(p. 184\)](#)
- [Launching Spot Instances in an Availability Zone Group \(p. 184\)](#)
- [Launching Spot Instances in a VPC \(p. 185\)](#)

Supply and Demand in the Spot Market

AWS continuously evaluates how many Spot instances are available in each Spot instance pool, monitors the bids that have been made for each pool, and provisions the available Spot instances to the highest bidders. The Spot price for a pool is set to the lowest fulfilled bid for that pool. Therefore, the Spot price is the price above which you must bid to fulfill a Spot request for a single Spot instance immediately.

For example, suppose that you create a Spot instance request, and that the corresponding Spot instance pool has only five Spot instances for sale. Your bid price is \$0.10, which is also the current Spot price. The following table shows the current bids, ranked in descending order. Bids 1-5 are fulfilled. Bid 5, being the last fulfilled bid, sets the Spot price at \$0.10. Bid 6 is unfulfilled. Bids 3-5, which share the same bid price of \$0.10, are ranked in random order.

Bid	Bid price	Current Spot price	Notes
1	\$1.00	\$0.10	
2	\$1.00	\$0.10	
3	\$0.10	\$0.10	
4	\$0.10	\$0.10	Your bid
5	\$0.10	\$0.10	Last fulfilled bid, which sets the Spot price. Everyone pays the same Spot price for the period.
---	---		Spot capacity cutoff
6	\$0.05		

Now, let's say that the size of this pool drops to 3. Bids 1-3 are fulfilled. Bid 3, the last fulfilled bid, sets the Spot price at \$0.10. Bids 4-5, which also are \$0.10, are unfulfilled. As you can see, even though the Spot price didn't change, two of the bids, including your bid, are no longer fulfilled because the Spot supply decreased.

Bid	Bid price	Current Spot price	Notes
1	\$1.00	\$0.10	
2	\$1.00	\$0.10	
3	\$0.10	\$0.10	Last fulfilled bid, which sets the Spot price. Everyone pays the

Bid	Bid price	Current Spot price	Notes
			same Spot price for the period.
— — —	— — —		Spot capacity cutoff
4	\$0.10		Your bid
5	\$0.10		
6	\$0.05		

To fulfill a Spot request for a single instance from this pool, you must bid above the current Spot price of \$0.10. If you bid \$0.101, your request will be fulfilled, the Spot instance for bid 3 would be interrupted, and the Spot price would become \$0.101. If you bid \$2.00, the Spot instance for bid 3 would be interrupted and the Spot price would become \$1.00 (the price for bid 2).

Keep in mind that no matter how high you bid, you can never get more than the available number of Spot instances in a Spot instance pool. If the size of the pool drops to zero, then all the Spot instances from that pool would be interrupted.

Launching Spot Instances in a Launch Group

Specify a launch group in your Spot instance request to tell Amazon EC2 to launch a set of Spot instances only if it can launch them all. In addition, if the Spot service must terminate one of the instances in a launch group (for example, if the Spot price rises above your bid price), it must terminate them all. However, if you terminate one or more of the instances in a launch group, Amazon EC2 does not terminate the remaining instances in the launch group.

Note that although this option can be useful, adding this constraint can lower the chances that your Spot instance request is fulfilled. It can also increase the chance that your Spot instances will be terminated.

If you create another successful Spot instance request that specifies the same (existing) launch group as an earlier successful request, then the new instances are added to the launch group. Subsequently, if an instance in this launch group is terminated, all instances in the launch group are terminated, which includes instances launched by the first and second requests.

Launching Spot Instances in an Availability Zone Group

Specify an Availability Zone group in your Spot instance request to tell the Spot service to launch a set of Spot instances in the same Availability Zone. Note that Amazon EC2 need not terminate all instances in an Availability Zone group at the same time. If Amazon EC2 must terminate one of the instances in an Availability Zone group, the others remain running.

Note that although this option can be useful, adding this constraint can lower the chances that your Spot instance request is fulfilled.

If you specify an Availability Zone group but don't specify an Availability Zone in the Spot instance request, the result depends on whether you specified the EC2-Classic network, a default VPC, or a nondefault VPC. For more information about EC2-Classic and EC2-VPC, see [Supported Platforms \(p. 672\)](#).

EC2-Classic

Amazon EC2 finds the lowest-priced Availability Zone in the region and launches your Spot instances in that Availability Zone if the lowest bid for the group is higher than the current Spot price in that

Availability Zone. Amazon EC2 waits until there is enough capacity to launch your Spot instances together, as long as the Spot price remains lower than the lowest bid for the group.

Default VPC

Amazon EC2 uses the Availability Zone for the specified subnet, or if you don't specify a subnet, it selects an Availability Zone and its default subnet, but it might not be the lowest-priced Availability Zone. If you deleted the default subnet for an Availability Zone, then you must specify a different subnet.

Nondefault VPC

Amazon EC2 uses the Availability Zone for the specified subnet.

Launching Spot Instances in a VPC

To take advantage of the features of EC2-VPC when you use Spot instances, specify in your Spot request that your Spot instances are to be launched in a VPC. You specify a subnet for your Spot instances the same way that you specify a subnet for your On-Demand instances.

The process for making a Spot instance request that launches Spot instances in a VPC is the same as the process for making a Spot instance request that launches Spot instances in EC2-Classical—except for the following differences:

- You should base your bid on the Spot price history of Spot instances in a VPC.
- [Default VPC] If you want your Spot instance launched in a specific low-priced Availability Zone, you must specify the corresponding subnet in your Spot instance request. If you do not specify a subnet, Amazon EC2 selects one for you, and the Availability Zone for this subnet might not have the lowest Spot price.
- [Nondefault VPC] You must specify the subnet for your Spot instance.

How Spot Fleet Works

A *Spot fleet* is a collection, or fleet, of Spot instances. The Spot fleet attempts to launch the number of Spot instances that are required to meet the target capacity that you specified in the Spot fleet request. The Spot fleet also attempts to maintain its target capacity fleet if your Spot instances are interrupted due to a change in Spot prices or available capacity.

A *Spot instance pool* is a set of unused EC2 instances with the same instance type, operating system, Availability Zone, and network platform (EC2-Classical or EC2-VPC). When you make a Spot fleet request, you can include multiple launch specifications, that vary by instance type, AMI, Availability Zone, or subnet. The Spot fleet selects the Spot instance pools that are used to fulfill the request, based on the launch specifications included in your Spot fleet request, and the configuration of the Spot fleet request. The Spot instances come from the selected pools.

Contents

- [Spot Fleet Allocation Strategy \(p. 185\)](#)
- [Spot Price Overrides \(p. 186\)](#)
- [Spot Fleet Instance Weighting \(p. 186\)](#)
- [Walkthrough: Using Spot Fleet with Instance Weighting \(p. 188\)](#)

Spot Fleet Allocation Strategy

The allocation strategy for your Spot fleet determines how it fulfills your Spot fleet request from the possible Spot instance pools represented by its launch specifications. The following are the allocation strategies that you can specify in your Spot fleet request:

`lowestPrice`

The Spot instances come from the pool with the lowest price. This is the default strategy.

`diversified`

The Spot instances are distributed across all pools.

Choosing an Allocation Strategy

You can optimize your Spot fleets based on your use case.

If your fleet is small or runs for a short time, the probability that your Spot instances will be interrupted is low, even with all the instances in a single Spot instance pool. Therefore, the `lowestPrice` strategy is likely to meet your needs while providing the lowest cost.

If your fleet is large or runs for a long time, you can improve the availability of your fleet by distributing the Spot instances across multiple pools. For example, if your Spot fleet request specifies 10 pools and a target capacity of 100 instances, the Spot fleet launches 10 Spot instances in each pool. If the Spot price for one pool increases above your bid price for this pool, only 10% of your fleet is affected. Using this strategy also makes your fleet less sensitive to increases in the Spot price in any one pool over time.

Note that with the `diversified` strategy, the Spot fleet does not launch Spot instances into any pools with a Spot price that is higher than the [On-Demand price](#).

Maintaining Target Capacity

After Spot instances are terminated due to a change in the Spot price or available capacity of a Spot instance pool, the Spot fleet launches replacement Spot instances. If the allocation strategy is `lowestPrice`, the Spot fleet launches replacement instances in the pool where the Spot price is currently the lowest. If the allocation strategy is `diversified`, the Spot fleet distributes the replacement Spot instances across the remaining pools.

Spot Price Overrides

Each Spot fleet request must include a global Spot price. By default, the Spot fleet uses this price as the bid price for each of its launch specifications.

You can optionally specify a Spot price in one or more launch specifications. This bid price is specific to the launch specification. If a launch specification includes a specific Spot price, the Spot fleet uses this price as the bid price for that launch specification, overriding the global Spot price. Note that any other launch specifications that do not include a specific Spot price still use the global Spot price.

Spot Fleet Instance Weighting

When you request a fleet of Spot instances, you can define the capacity units that each instance type would contribute to your application's performance, and adjust your bid price for each Spot instance pool accordingly using *instance weighting*.

By default, the Spot price that you specify represents your bid price *per instance hour*. When you use the instance weighting feature, the Spot price that you specify represents your bid price *per unit hour*. You can calculate your bid price per unit hour by dividing your bid price for an instance type by the number of units that it represents. The Spot fleet calculates the number of Spot instances to launch by dividing the target capacity by the instance weight. If the result isn't an integer, the Spot fleet rounds it up to the next integer, so that the size of your fleet is not below its target capacity. Note that Spot fleet can select any pool that you specify in your launch specification, even if the capacity of the instances launched exceeds the requested target capacity.

The following table includes examples of calculations to determine the bid price per unit for a Spot fleet request with a target capacity of 10.

Instance type	Instance weight	Spot price per instance hour	Spot price per unit hour	Number of instances launched
r3.xlarge	2	\$0.05	.025 (.05 divided by 2)	5 (10 divided by 2)
r3.8xlarge	8	\$0.10	.0125 (.10 divided by 8)	2 (10 divided by 8, result rounded up)

Use Spot fleet instance weighting as follows to provision the target capacity you want in the pools with the lowest price per unit at the time of fulfillment:

1. Set the target capacity for your Spot fleet either in instances (the default) or in the units of your choice, such as virtual CPUs, memory, storage, or throughput.
2. Set the bid price per unit.
3. For each launch configuration, specify the weight, which is the number of units that the instance type represents toward the target capacity.

Instance Weighting Example

Consider a Spot fleet request with the following configuration:

- A target capacity of 24
- A launch specification with an instance type `r3.2xlarge` and a weight of 6
- A launch specification with an instance type `c3.xlarge` and a weight of 5

The weights represent the number of units that instance type represents toward the target capacity. If the first launch specification provides the lowest Spot price per unit (Spot price for `r3.2xlarge` per instance hour divided by 6), the Spot fleet would launch four of these instances (24 divided by 6).

If the second launch specification provides the lowest Spot price per unit (Spot price for `c3.xlarge` per instance hour divided by 5), the Spot fleet would launch five of these instances (24 divided by 5, result rounded up).

Instance Weighting and Allocation Strategy

Consider a Spot fleet request with the following configuration:

- A target capacity of 30
- A launch specification with an instance type `c3.2xlarge` and a weight of 8
- A launch specification with an instance type `m3.xlarge` and a weight of 8
- A launch specification with an instance type `r3.xlarge` and a weight of 8

The Spot fleet would launch four instances (30 divided by 8, result rounded up). With the `lowestPrice` strategy, all four instances come from the pool that provides the lowest Spot price per unit. With the `diversified` strategy, the Spot fleet launches 1 instance in each of the three pools, and the fourth instance in whichever of the three pools provides the lowest Spot price per unit.

Walkthrough: Using Spot Fleet with Instance Weighting

This walkthrough uses a fictitious company called Example Corp to illustrate the process of bidding for a Spot fleet using instance weighting.

Objective

Example Corp, a pharmaceutical company, wants to leverage the computational power of Amazon EC2 for screening chemical compounds that might be used to fight cancer.

Planning

Example Corp first reviews [Spot Best Practices](#). Next, Example Corp determines the following requirements for their Spot fleet.

Instance Types

Example Corp has a compute- and memory-intensive application that performs best with at least 60 GB of memory and eight virtual CPUs (vCPUs). They want to maximize these resources for the application at the lowest possible price. Example Corp decides that any of the following EC2 instance types would meet their needs:

Instance type	Memory (GiB)	vCPUs
r3.2xlarge	61	8
r3.4xlarge	122	16
r3.8xlarge	244	32

Target Capacity in Units

With instance weighting, target capacity can equal a number of instances (the default) or a combination of factors such as cores (vCPUs), memory (GiBs), and storage (GBs). By considering the base for their application (60 GB of RAM and eight vCPUs) as 1 unit, Example Corp decides that 20 times this amount would meet their needs. So the company sets the target capacity of their Spot fleet request to 20.

Instance Weights

After determining the target capacity, Example Corp calculates instance weights. To calculate the instance weight for each instance type, they determine the units of each instance type that are required to reach the target capacity as follows:

- r3.2xlarge (61.0 GB, 8 vCPUs) = 1 unit of 20
- r3.4xlarge (122.0 GB, 16 vCPUs) = 2 units of 20
- r3.8xlarge (244.0 GB, 32 vCPUs) = 4 units of 20

Therefore, Example Corp assigns instance weights of 1, 2, and 4 to the respective launch configurations in their Spot fleet request.

Bid Price Per Unit Hour

Example Corp uses the [On-Demand price](#) per instance hour as a starting point for their bid price. They could also use recent Spot prices, or a combination of the two. To calculate bid price per unit hour, they divide their starting bid price per instance hour by the weight. For example:

Instance type	On-Demand price	Instance weight	Price per unit hour
r3.2xLarge	\$0.7	1	\$0.7

Instance type	On-Demand price	Instance weight	Price per unit hour
r3.4xLarge	\$1.4	2	\$0.7
r3.8xLarge	\$2.8	4	\$0.7

Example Corp could enter a global bid price per unit hour of \$0.7 and be competitive for all three instance types. They could also enter a global bid price per unit hour of \$0.7 and a specific bid price per unit hour of \$0.9 in the `r3.8xlarge` launch specification. Depending on the strategy for provisioning their Spot fleet, Example Corp could bid lower to further reduce costs, or bid higher to reduce the probability of interruption.

Verifying Permissions

Before creating a Spot fleet request, Example Corp verifies that it has an IAM role with the required permissions. For more information, see [Spot Fleet Prerequisites \(p. 199\)](#).

Creating the Request

Example Corp creates a file, `config.json`, with the following configuration for its Spot fleet request:

```
{
  "SpotPrice": "0.70",
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
  "LaunchSpecifications": [
    {
      "ImageId": "ami-1a2b3c4d",
      "InstanceType": "r3.2xlarge",
      "SubnetId": "subnet-482e4972",
      "WeightedCapacity": 1
    },
    {
      "ImageId": "ami-1a2b3c4d",
      "InstanceType": "r3.4xlarge",
      "SubnetId": "subnet-482e4972",
      "WeightedCapacity": 2
    },
    {
      "ImageId": "ami-1a2b3c4d",
      "InstanceType": "r3.8xlarge",
      "SubnetId": "subnet-482e4972",
      "SpotPrice": "0.90",
      "WeightedCapacity": 4
    }
  ]
}
```

Example Corp creates the Spot fleet request using the following `request-spot-fleet` command:

```
aws ec2 request-spot-fleet --spot-fleet-request-config file://config.json
```

For more information, see [Spot Fleet Requests \(p. 198\)](#).

Fulfillment

The allocation strategy determines which Spot instance pools your Spot instances come from.

With the `lowestPrice` strategy (which is the default strategy), the Spot instances come from the pool with the lowest Spot price per unit at the time of fulfillment. To provide 20 units of capacity, the Spot fleet launches either 20 `r3.2xlarge` instances (20 divided by 1), 10 `r3.4xlarge` instances (20 divided by 2), or 5 `r3.8xlarge` instances (20 divided by 4).

If Example Corp used the `diversified` strategy, the Spot instances would come from all three pools. The Spot fleet would launch 6 `r3.2xlarge` instances (which provide 6 units), 3 `r3.4xlarge` instances (which provide 6 units), and 2 `r3.8xlarge` instances (which provide 8 units), for a total of 20 units.

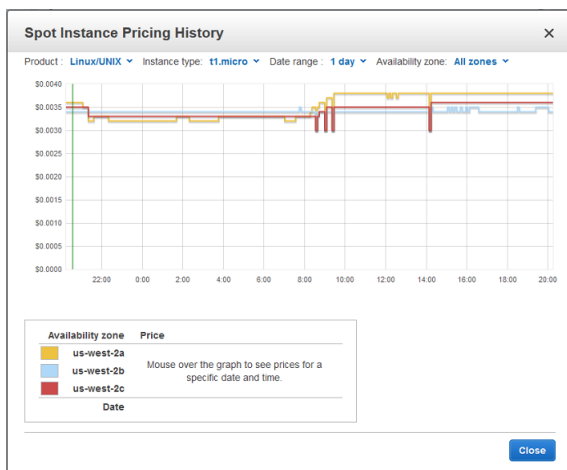
Spot Instance Pricing History

The Spot price represents the price above which you have to bid to guarantee that a single Spot request is fulfilled. When your bid price is above the Spot price, Amazon EC2 launches your Spot instance, and when the Spot price rises above your bid price, Amazon EC2 terminates your Spot instance. You can bid above the current Spot price so that your Spot request is fulfilled quickly. However, before you specify a bid price for your Spot instance, we recommend that you review the Spot price history. You can view the Spot price history for the last 90 days, filtering by instance type, operating system, and Availability Zone.

Using the Spot price history as a guide, you can select a bid price that would have met your needs in the past. For example, you can determine which bid price that would have provided 75 percent uptime in the time range you viewed. However, keep in mind that the historical trends are not a guarantee of future results. Spot prices vary based on real-time supply and demand, and the conditions that generated certain patterns in the Spot price might not occur in the future.

To view the Spot price history using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the navigation pane, choose **Spot Requests**.
3. If you are new to Spot instances, you see a welcome page; choose **Get started**, scroll to the bottom of the screen, and then choose **Cancel**.
4. Choose **Pricing History**. By default, the page displays a graph of the data for Linux `t1.micro` instances in all Availability Zones over the past day. Move your mouse over the graph to display the prices at specific times in the table below the graph.



5. (Optional) To review the Spot price history for a specific Availability Zone, select an Availability Zone from the list. You can also select a different product, instance type, or date range.

To view the Spot price history using the command line

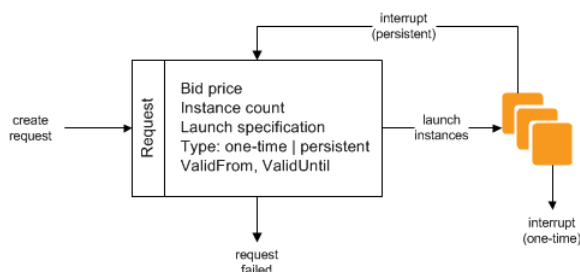
You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [describe-spot-price-history](#) (AWS CLI)
- [Get-EC2SpotPriceHistory](#) (AWS Tools for Windows PowerShell)

Spot Instance Requests

To use Spot instances, you create a Spot instance request that includes the number of instances, the instance type, the Availability Zone, and the maximum price that you are willing to pay per instance hour (your bid). If your bid exceeds the current Spot price, Amazon EC2 fulfills your request immediately. Otherwise, Amazon EC2 waits until your request can be fulfilled or until you cancel the request.

The following illustration shows how Spot requests work. Notice that the action taken for a Spot instance interruption depends on the request type (one-time or persistent). If the request is a persistent request, the request is opened again after your Spot instance is terminated.



Contents

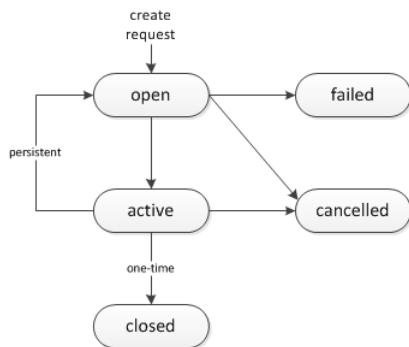
- [Spot Instance Request States \(p. 191\)](#)
- [Specifying a Duration for Your Spot Instances \(p. 192\)](#)
- [Creating a Spot Instance Request \(p. 193\)](#)
- [Finding Running Spot Instances \(p. 195\)](#)
- [Tagging Spot Instance Requests \(p. 195\)](#)
- [Cancelling a Spot Instance Request \(p. 196\)](#)
- [Spot Request Example Launch Specifications \(p. 196\)](#)

Spot Instance Request States

A Spot instance request can be in one of the following states:

- **open**—The request is waiting to be fulfilled.
- **active**—The request is fulfilled and has an associated Spot instance.
- **failed**—The request has one or more bad parameters.
- **closed**—The Spot instance was interrupted or terminated.
- **cancelled**—You cancelled the request, or the request expired.

The following illustration represents the transitions between the request states. Notice that the transitions depend on the request type (one-time or persistent).



A one-time Spot instance request remains active until Amazon EC2 launches the Spot instance, the request expires, or you cancel the request. If the Spot price rises above your bid price, your Spot instance is terminated and the Spot instance request is closed.

A persistent Spot instance request remains active until it expires or you cancel it, even if the request is fulfilled. For example, if you create a persistent Spot instance request for one instance when the Spot price is \$0.25, Amazon EC2 launches your Spot instance if your bid price is above \$0.25. If the Spot price rises above your bid price, your Spot instance is terminated; however, the Spot instance request is open again and Amazon EC2 launches a new Spot instance when the Spot price falls below your bid price.

You can track the status of your Spot instance requests, as well as the status of the Spot instances launched, through the bid status. For more information, see [Spot Bid Status \(p. 217\)](#).

Specifying a Duration for Your Spot Instances

Amazon EC2 does not terminate Spot instances with a specified duration (also known as Spot blocks) when the Spot price changes. This makes them ideal for jobs that take a finite time to complete, such as batch processing, encoding and rendering, modeling and analysis, and continuous integration.

You can specify a duration of 1, 2, 3, 4, 5, or 6 hours. The price that you pay depends on the specified duration. To view the current prices for a 1 hour duration or a 6 hour duration, see [Spot Instance Prices](#). You can use these prices to estimate the cost of the 2, 3, 4, and 5 hour durations. When a request with a duration is fulfilled, the price for your Spot instance is fixed, and this price remains in effect until the instance terminates.

When you specify a duration in your Spot request, the duration period for each Spot instance starts as soon as the instance receives its instance ID. The Spot instance runs until you terminate it or the duration period ends. At the end of the duration period, Amazon EC2 marks the Spot instance for termination and provides a Spot instance termination notice, which gives the instance a two-minute warning before it terminates.

To launch Spot instances with a specified duration using the console

Select the appropriate request type. For more information, see [Creating a Spot Instance Request \(p. 193\)](#).

To launch Spot instances with a specified duration using the AWS CLI

To specify a duration for your Spot instances, include the `--block-duration-minutes` option with the `request-spot-instances` command. For example, the following command creates a Spot request that launches Spot instances that run for two hours:

```
aws ec2 request-spot-instances --spot-price "0.050" --instance-count 5
--block-duration-minutes 120 --type "one-time" --launch-specification
file://specification.json
```

To retrieve the cost for Spot instances with a specified duration using the AWS CLI

Use the `describe-spot-instance-requests` command to retrieve the fixed cost for your Spot instances with a specified duration. The information is in the `actualBlockHourlyPrice` field.

Creating a Spot Instance Request

The process for requesting a Spot instance is similar to the process for launching an On-Demand instance. Note that you can't change the parameters of your Spot request, including the bid price, after you've submitted the request.

If you request multiple Spot instances at one time, Amazon EC2 creates separate Spot instance requests so that you can track the status of each request separately. For more information about tracking Spot requests, see [Spot Bid Status \(p. 217\)](#).

Prerequisites

Before you begin, decide on your bid price, how many Spot instances you'd like, and what instance type to use. To review Spot price trends, see [Spot Instance Pricing History \(p. 190\)](#).

To create a Spot instance request using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the navigation pane, choose **Spot Requests**.
3. If you are new to Spot instances, you see a welcome page; choose **Get started**. Otherwise, choose **Request Spot Instances**.
4. On the **Find instance types** page, do the following:
 - a. For **Request type**, the default is a one-time Spot request created using a Spot fleet. For more information, see [Spot Fleet Requests \(p. 198\)](#). To use Spot blocks instead, select **Reserve for duration**.
 - b. For **Target capacity**, enter the number of units to request. You can choose instances or performance characteristics that are important to your application workload, such as vCPUs, memory, and storage.
 - c. [Spot block] For **Reserved duration**, select the number of hours for the job to complete.
 - d. For **AMI**, choose one of the basic Amazon Machine Images (AMI) provided by AWS, or choose **Use custom AMI** to specify your own AMI.
 - e. For **Instance type(s)**, choose **Select**. Select the instance types that have the minimum hardware specifications that you need (vCPUs, memory, and storage).
 - f. [Spot fleet] For **Allocation strategy**, choose the strategy that meets your needs. For more information, see [Spot Fleet Allocation Strategy \(p. 185\)](#).
 - g. For **Network**, your account supports either the EC2-Classic and EC2-VPC platforms, or the EC2-VPC platform only. To find out which platforms your account supports, see [Supported Platforms \(p. 672\)](#).
 - [Existing VPC] Select the VPC.
 - [New VPC] Select **Create new VPC** to go the Amazon VPC console. When you are done, return to the wizard and refresh the list.
 - [EC2-Classic] Select **EC2-Classic**.
 - h. (Optional) For **Availability Zones**, the default is to let AWS choose the Availability Zones for your Spot instances. If you prefer specific Availability Zones, do the following:
 - [EC2-VPC] Select one or more Availability Zones. If you have more than one subnet in an Availability Zone, select the appropriate subnet from **Subnet**. To add subnets, select **Create new subnet** to go to the Amazon VPC console. When you are done, return to the wizard and refresh the list.

- [EC2-Classical] Select **Select specific zone/subnet**, and then select one or more Availability Zones.
 - i. [Spot fleet] For **Maximum price**, you can use automated bidding or specify a bid price. Your Spot instances are not launched if your bid price is lower than the Spot price for the instance types that you selected.
 - j. Choose **Next**.
5. On the **Configure** page, do the following:
 - a. (Optional) If you need to connect to your instances, specify your key pair using **Key pair name**.
 - b. (Optional) If you need to launch your Spot instances with an IAM role, specify the role using **IAM instance profile**.
 - c. (Optional) If you have any start-up scripts to run, specify them using **User data**.
 - d. For **Security groups**, choose one or more security groups.
 - e. [EC2-VPC] If you need to connect to your instances in a VPC, select **auto-assign at launch** for **Public IP**.
 - f. By default, the request remains in effect until it is fulfilled or you cancel it. To create a request that is valid only during a specific time period, edit **Request valid from** and **Request valid to**.
 - g. [Spot fleet] By default, we terminate your Spot instances when the request expires. To keep them running after your request expires, clear **Terminate instances at expiration**.
 - h. Choose **Review**.
 6. On the **Review** page, verify the launch configuration. To make changes, choose **Previous**. To download a copy of the launch configuration for use with the AWS CLI, choose **JSON config**. When you are ready, choose **Launch**.
 7. On the confirmation page, choose **OK**.

[Spot fleet] The request type is `fleet`. When the request is fulfilled, requests of type `instance` are added, where the state is `active` and the status is `fulfilled`.

[Spot block] The request type is `block` and the initial state is `open`. When the request is fulfilled, the state is `active` and the status is `fulfilled`.

To create a Spot instance request using the AWS CLI

Use the following [request-spot-instances](#) command to create a one-time request:

```
aws ec2 request-spot-instances --spot-price "0.05" --instance-count 5 --type "one-time" --launch-specification file://specification.json
```

Use the following [request-spot-instances](#) command to create a persistent request:

```
aws ec2 request-spot-instances --spot-price "0.05" --instance-count 5 --type "persistent" --launch-specification file://specification.json
```

For example launch specification files, see [Spot Request Example Launch Specifications](#) (p. 196).

Amazon EC2 launches your Spot instance when the Spot price is below your bid. The Spot instance runs until either it is interrupted, or you terminate it yourself. Use the following [describe-spot-instance-requests](#) command to monitor your Spot instance request:

```
aws ec2 describe-spot-instance-requests --spot-instance-request-ids sir-08b93456
```

Finding Running Spot Instances

Amazon EC2 launches a Spot instance when the Spot price is below your bid. A Spot instance runs until either its bid price is no longer higher than the Spot price, or you terminate it yourself. (If your bid price is exactly equal to the Spot price, there is a chance that your Spot instance will remain running, depending on demand.)

To find running Spot instances using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Spot Requests**.

You can see both Spot instance requests and Spot fleet requests. If a Spot instance request has been fulfilled, **Capacity** is the ID of the Spot instance. For a Spot fleet, **Capacity** indicates how much of the requested capacity has been fulfilled. To view the IDs of the instances in a Spot fleet, choose the expand arrow, or select the fleet and then select the **Instances** tab.

3. Alternatively, in the navigation pane, choose **Instances**. In the top right corner, choose the **Show/Hide** icon, and then select **Lifecycle**. For each instance, **Lifecycle** is either `normal`, `spot`, or `scheduled`.

To find running Spot instances using the AWS CLI

To enumerate your Spot instances, use the `describe-spot-instance-requests` command with the `--query` option as follows:

```
aws ec2 describe-spot-instance-requests --query SpotInstanceRequests[*].  
{ID:InstanceId}
```

The following is example output:

```
[  
  {  
    "ID": "i-1234567890abcdef0"  
  },  
  {  
    "ID": "i-0598c7d356eba48d7"  
  }  
]
```

Alternatively, you can enumerate your Spot instances using the `describe-instances` command with the `--filters` option as follows:

```
aws ec2 describe-instances --filters "Name=instance-lifecycle,Values=spot"
```

Tagging Spot Instance Requests

To help categorize and manage your Spot instance requests, you can tag them with metadata of your choice. You tag your Spot instance requests in the same way that you tag other any other Amazon EC2 resource. For more information, see [Tagging Your Amazon EC2 Resources \(p. 859\)](#).

You can assign a tag to the request after you create it.

The tags that you create for your Spot instance requests only apply to the requests. These tags are not added automatically to the Spot instance that the Spot service launches to fulfill the request. You must add tags to a Spot instance yourself after the Spot instance is launched.

To add a tag to your Spot instance request or Spot instance using the AWS CLI

Use the following [create-tags](#) command to tag your resources:

```
aws ec2 create-tags --resources sir-08b93456 i-1234567890abcdef0 --tags  
Key=purpose,Value=test
```

Cancelling a Spot Instance Request

If you no longer want your Spot request, you can cancel it. You can only cancel Spot instance requests that are `open` or `active`. Your Spot request is `open` when your request has not yet been fulfilled and no instances have been launched. Your Spot request is `active` when your request has been fulfilled, and Spot instances have launched as a result. If your Spot request is `active` and has an associated running Spot instance, cancelling the request does not terminate the instance; you must terminate the running Spot instance manually.

If the Spot request is a persistent Spot request, it returns to the `open` state so that a new Spot instance can be launched. To cancel a persistent Spot request and terminate its Spot instances, you must cancel the Spot request first and then terminate the Spot instances. Otherwise, the Spot request can launch a new instance.

To cancel a Spot instance request using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Spot Requests**, and then select the Spot request.
3. Choose **Actions**, and then choose **Cancel spot request**.
4. (Optional) If you are finished with the associated Spot instances, you can terminate them. In the navigation pane, choose **Instances**, select the instance, choose **Actions**, choose **Instance State**, and then choose **Terminate**.

To cancel a Spot instance request using the AWS CLI

Use the following [cancel-spot-instance-requests](#) command to cancel the specified Spot request:

```
aws ec2 cancel-spot-instance-requests --spot-instance-request-  
ids sir-08b93456
```

If you are finished with the associated Spot instances, you can terminate them manually using the following [terminate-instances](#) command:

```
aws ec2 terminate-instances --instance-  
ids i-1234567890abcdef0 i-0598c7d356eba48d7
```

Spot Request Example Launch Specifications

The following examples show launch configurations that you can use with the [request-spot-instances](#) command to create a Spot instance request. For more information, see [Creating a Spot Instance Request](#) (p. 193).

1. [Launch Spot instances](#) (p. 196)
2. [Launch Spot instances in the specified Availability Zone](#) (p. 197)
3. [Launch Spot instances in the specified subnet](#) (p. 197)

Example 1: Launch Spot Instances

The following example does not include an Availability Zone or subnet. Amazon EC2 selects an Availability Zone for you. If your account supports EC2-VPC only, Amazon EC2 launches the instances

in the default subnet of the selected Availability Zone. If your account supports EC2-Classic, Amazon EC2 launches the instances in EC2-Classic in the selected Availability Zone.

```
{
  "ImageId": "ami-1a2b3c4d",
  "KeyName": "my-key-pair",
  "SecurityGroupIds": [ "sg-1a2b3c4d" ],
  "InstanceType": "m3.medium",
  "IamInstanceProfile": {
    "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
  }
}
```

Note that you can specify security groups for EC2-Classic either by ID or by name (using the `SecurityGroups` field). You must specify security groups for EC2-VPC by ID.

Example 2: Launch Spot Instances in the Specified Availability Zone

The following example includes an Availability Zone. If your account supports EC2-VPC only, Amazon EC2 launches the instances in the default subnet of the specified Availability Zone. If your account supports EC2-Classic, Amazon EC2 launches the instances in EC2-Classic in the specified Availability Zone.

```
{
  "ImageId": "ami-1a2b3c4d",
  "KeyName": "my-key-pair",
  "SecurityGroupIds": [ "sg-1a2b3c4d" ],
  "InstanceType": "m3.medium",
  "Placement": {
    "AvailabilityZone": "us-west-2a"
  },
  "IamInstanceProfile": {
    "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
  }
}
```

Example 3: Launch Spot Instances in the Specified Subnet

The following example includes a subnet. Amazon EC2 launches the instances in the specified subnet. If the VPC is a nondefault VPC, the instance does not receive a public IPv4 address by default.

```
{
  "ImageId": "ami-1a2b3c4d",
  "SecurityGroupIds": [ "sg-1a2b3c4d" ],
  "InstanceType": "m3.medium",
  "SubnetId": "subnet-1a2b3c4d",
  "IamInstanceProfile": {
    "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
  }
}
```

To assign a public IPv4 address to an instance in a nondefault VPC, specify the `AssociatePublicIpAddress` field as shown in the following example. Note that when you specify a network interface, you must include the subnet ID and security group ID using the network interface, rather than using the `SubnetId` and `SecurityGroupIds` fields shown in example 3.

```
{
```

```
"ImageId": "ami-1a2b3c4d",
"KeyName": "my-key-pair",
"InstanceType": "m3.medium",
"NetworkInterfaces": [
  {
    "DeviceIndex": 0,
    "SubnetId": "subnet-1a2b3c4d",
    "Groups": [ "sg-1a2b3c4d" ],
    "AssociatePublicIpAddress": true
  }
],
"IamInstanceProfile": {
  "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
}
}
```

Spot Fleet Requests

To use a Spot fleet, you create a Spot fleet request that includes the target capacity, one or more launch specifications for the instances, and the bid price that you are willing to pay. Amazon EC2 attempts to maintain your Spot fleet's target capacity as Spot prices change. For more information, see [How Spot Fleet Works \(p. 185\)](#).

You can create a Spot fleet to submit a one-time `request` for your desired capacity, or require it to `maintain` a target capacity over time. Both types of requests benefit from Spot fleet's allocation strategy.

When you `request` a target capacity, Spot fleet places the required bids but will not attempt to replenish Spot instances if capacity is diminished. If capacity is not available, Spot fleet will not submit bids in alternative Spot pools.

When you want to `maintain` a target capacity, Spot fleet will place the required bids to meet this target capacity and automatically replenish any interrupted instances. By default, Spot fleets are set to `maintain` the requested target capacity.

It is not possible to modify the target capacity of a one-time `request` once it's been submitted. To change the target capacity, cancel the request and submit a new one.

A Spot fleet request remains active until it expires or you cancel it. When you cancel a Spot fleet request, you may specify whether cancelling your Spot fleet request terminates the Spot instances in your Spot fleet.

Each launch specification includes the information that Amazon EC2 needs to launch an instance—such as an AMI, an instance type, a subnet or Availability Zone, and one or more security groups.

Contents

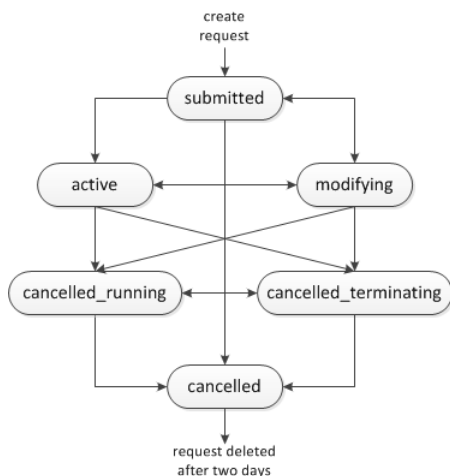
- [Spot Fleet Request States \(p. 199\)](#)
- [Spot Fleet Prerequisites \(p. 199\)](#)
- [Spot Fleet and IAM Users \(p. 200\)](#)
- [Planning a Spot Fleet Request \(p. 201\)](#)
- [Creating a Spot Fleet Request \(p. 201\)](#)
- [Monitoring Your Spot Fleet \(p. 202\)](#)
- [Modifying a Spot Fleet Request \(p. 203\)](#)
- [Cancelling a Spot Fleet Request \(p. 204\)](#)
- [Spot Fleet Example Configurations \(p. 205\)](#)

Spot Fleet Request States

A Spot fleet request can be in one of the following states:

- `submitted`—The Spot fleet request is being evaluated and Amazon EC2 is preparing to launch the target number of Spot instances.
- `active`—The Spot fleet has been validated and Amazon EC2 is attempting to maintain the target number of running Spot instances. The request remains in this state until it is modified or cancelled.
- `modifying`—The Spot fleet request is being modified. The request remains in this state until the modification is fully processed or the Spot fleet is cancelled. A one-time `request` cannot be modified, and this state does not apply to such Spot requests.
- `cancelled_running`—The Spot fleet is cancelled and will not launch additional Spot instances, but its existing Spot instances continue to run until they are interrupted or terminated. The request remains in this state until all instances are interrupted or terminated.
- `cancelled_terminating`—The Spot fleet is cancelled and its Spot instances are terminating. The request remains in this state until all instances are terminated.
- `cancelled`—The Spot fleet is cancelled and has no running Spot instances. The Spot fleet request is deleted two days after its instances were terminated.

The following illustration represents the transitions between the request states. Note that if you exceed your Spot fleet limits, the request is cancelled immediately.



Spot Fleet Prerequisites

If you use the AWS Management Console to create a Spot fleet, it creates a role named `aws-ec2-spot-fleet-role` that grants the Spot fleet permission to bid on, launch, and terminate instances on your behalf, and specifies it in your Spot fleet request. If you create a Spot fleet using the AWS CLI or an API, you can use this role if it exists, or manually create your own role for this purpose as follows.

To manually create an IAM role with the `AmazonEC2SpotFleetRole` policy

1. Open the Identity and Access Management (IAM) console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Roles**.
3. Choose **Create New Role**.
4. On the **Set Role Name** page, type a name for the role and then choose **Next Step**.

5. On the **Select Role Type** page, choose **Select** next to **Amazon EC2 Spot Fleet Role**.
6. On the **Attach Policy** page, select the `AmazonEC2SpotFleetRole` policy, and then choose **Next Step**.
7. On the **Review** page, choose **Create Role**.

Spot Fleet and IAM Users

If IAM users will be creating or managing Spot fleet, be sure to grant them the required permissions as follows.

To grant an IAM user permissions for Spot fleet

1. Open the Identity and Access Management (IAM) console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Policies**, and then choose **Create Policy**.
3. On the **Create Policy** page, choose **Select** next to **Create Your Own Policy**.
4. On the **Review Policy** page, enter a policy name and copy the following text into the **Policy Document** section.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole",
        "iam:ListRoles",
        "iam:ListInstanceProfiles"
      ],
      "Resource": "*"
    }
  ]
}
```

The `ec2:*` enables an IAM user to call all Amazon EC2 API actions. To limit the user to specific API actions, specify those actions instead.

The `iam:PassRole` action enables the user to specify the Spot fleet role in a Spot fleet request. The `iam:ListRoles` action enables the user to enumerate existing roles. The `iam:ListInstanceProfiles` action enables the user to enumerate existing instance profiles. The Amazon EC2 console uses `iam:ListRoles` to populate the **IAM role** list and `iam:ListInstanceProfiles` to populate the **IAM instance profile** list. To enable the user to create roles or instance profiles using the console, you must add the following actions: `iam:CreateRole`, `iam:CreateInstanceProfile`, and `iam:AddRoleToInstanceProfile`.

5. Choose **Create Policy**.
6. In the navigation pane, choose **Users**, and then choose the user who will submit the Spot fleet request.
7. On the **Permissions** tab, choose **Add permissions**.

8. Choose **Attach existing policies directly**. Select the policy you created above, choose **Next: Review**, then **Add permissions**.

Planning a Spot Fleet Request

Before you create a Spot fleet request, review [Spot Best Practices](#). Use these best practices when you plan your Spot fleet request so that you can provision the type of instances you want at the lowest possible price. We also recommend that you do the following:

- Determine whether you want to create a Spot fleet that submits a one-time `request` for the desired target capacity, or one that will `maintain` a target capacity over time.
- Determine the instance types that meet your application requirements.
- Determine the target capacity for your Spot fleet request. You can set target capacity in instances or in custom units. For more information, see [Spot Fleet Instance Weighting \(p. 186\)](#).
- Determine your bid price per instance hour. Bidding lower can further reduce costs, while bidding higher can reduce the probability of interruption.
- Determine your bid price per unit, if you are using instance weighting. To calculate the bid price per unit, divide the bid price per instance hour by the number of units (or weight) that this instance represents. (If you are not using instance weighting, the default bid price per unit is the bid price per instance hour.)
- Review the possible options for your Spot fleet request. For more information, see the [request-spot-fleet](#) command in the *AWS Command Line Interface Reference*. For additional examples, see [Spot Fleet Example Configurations \(p. 205\)](#).

Creating a Spot Fleet Request

When you create a Spot fleet request, you must specify information about the Spot instances to launch, such as the instance type and the Spot price.

To create a Spot fleet request using the console

1. Open the Spot console at <https://console.aws.amazon.com/ec2spot>.
2. If you are new to Spot, you see a welcome page; choose **Get started**. Otherwise, choose **Request Spot Instances**.
3. On the **Find instance types** page, do the following:
 - a. For **Request type**, select either **Request** or **Request and Maintain**.
 - b. For **Target capacity**, enter the number of units to request. You can choose instances or performance characteristics that are important to your application workload, such as vCPUs, memory, and storage.
 - c. For **AMI**, choose one of the basic Amazon Machine Images (AMI) provided by AWS, or choose **Use custom AMI** to use an AMI from our user community, the AWS Marketplace, or one of your own.
 - d. For **Instance type(s)**, choose **Select**. Select the instance types that have the minimum hardware specifications that you need (vCPUs, memory, and storage).
 - e. For **Allocation strategy**, choose the strategy that meets your needs. For more information, see [Spot Fleet Allocation Strategy \(p. 185\)](#).
 - f. For **Network**, your account supports either the EC2-Classical and EC2-VPC platforms, or the EC2-VPC platform only. To find out which platforms your account supports, see [Supported Platforms \(p. 672\)](#).
 - [Existing VPC] Select the VPC.
 - [New VPC] Select **Create new VPC** to go the Amazon VPC console. When you are done, return to the wizard and refresh the list.

- [EC2-Classic] Select **EC2-Classic**.
- g. (Optional) For **Availability Zones**, the default is to let AWS choose the Availability Zones for your Spot instances. If you prefer specific Availability Zones, do the following:
 - [EC2-VPC] Select one or more Availability Zones. If you have more than one subnet in an Availability Zone, select the appropriate subnet from **Subnet**. To add subnets, select **Create new subnet** to go to the Amazon VPC console. When you are done, return to the wizard and refresh the list.
 - [EC2-Classic] Select **Select specific zone/subnet**, and then select one or more Availability Zones.
 - h. For **Maximum price**, you can use automated bidding or specify a bid price. Your Spot instances are not launched if your bid price is lower than the Spot price for the instance types that you selected.
 - i. Choose **Next**.
4. On the **Configure** page, do the following:
 - a. (Optional) If you need to connect to your instances, specify your key pair using **Key pair name**.
 - b. (Optional) If you need to launch your Spot instances with an IAM role, specify the role using **IAM instance profile**.
 - c. (Optional) If you have any start-up scripts to run, specify them using **User data**.
 - d. For **Security groups**, choose one or more security groups.
 - e. [EC2-VPC] If you need to connect to your instances in a VPC, select **auto-assign at launch** for **Public IP**.
 - f. By default, the request remains in effect until it is fulfilled or you cancel it. To create a request that is valid only during a specific time period, edit **Request valid from** and **Request valid to**.
 - g. (Optional) By default, we terminate your Spot instances when the request expires. To keep them running after your request expires, clear **Terminate instances at expiration**.
 - h. Choose **Review**.
 5. On the **Review** page, verify the launch configuration. To make changes, choose **Previous**. To download a copy of the launch configuration for use with the AWS CLI, choose **JSON config**. When you are ready, choose **Launch**.
 6. On the confirmation page, choose **OK**. The request type is `fleet`. When the request is fulfilled, requests of type `instance` are added, where the state is `active` and the status is `fulfilled`.

To create a Spot fleet request using the AWS CLI

Use the following [request-spot-fleet](#) command to create a Spot fleet request:

```
aws ec2 request-spot-fleet --spot-fleet-request-config file://config.json
```

For example configuration files, see [Spot Fleet Example Configurations \(p. 205\)](#).

The following is example output:

```
{  
  "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE"  
}
```

Monitoring Your Spot Fleet

The Spot fleet launches Spot instances when the Spot price is below your bid. The Spot instances run until either the bid price is no longer higher than the Spot price, or you terminate them yourself.

To monitor your Spot fleet using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Spot Requests**.
3. Select your Spot fleet request. The configuration details are available in the **Description** tab.
4. To list the Spot instances for the Spot fleet, choose the **Instances** tab.
5. To view the history for the Spot fleet, choose the **History** tab.

To monitor your Spot fleet using the AWS CLI

Use the following [describe-spot-fleet-requests](#) command to describe your Spot fleet requests:

```
aws ec2 describe-spot-fleet-requests
```

Use the following [describe-spot-fleet-instances](#) command to describe the Spot instances for the specified Spot fleet:

```
aws ec2 describe-spot-fleet-instances --spot-fleet-request-id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

Use the following [describe-spot-fleet-request-history](#) command to describe the history for the specified Spot fleet request:

```
aws ec2 describe-spot-fleet-request-history --spot-fleet-request-id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE --start-time 2015-05-18T00:00:00Z
```

Modifying a Spot Fleet Request

You can modify an active Spot fleet request to complete the following tasks:

- Increase the target capacity
- Decrease the target capacity

Note

It is not possible to modify a one-time Spot fleet request.

When you increase the target capacity, the Spot fleet launches the additional Spot instances according to the allocation strategy for its Spot fleet request. If the allocation strategy is `lowestPrice`, the Spot fleet launches the instances from the lowest-priced Spot instance pool in the Spot fleet request. If the allocation strategy is `diversified`, the Spot fleet distributes the instances across the pools in the Spot fleet request.

When you decrease the target capacity, the Spot fleet cancels any open bids that exceed the new target capacity. You can request that the Spot fleet terminate Spot instances until the size of the fleet reaches the new target capacity. If the allocation strategy is `lowestPrice`, the Spot fleet terminates the instances with the highest price per unit. If the allocation strategy is `diversified`, the Spot fleet terminates instances across the pools. Alternatively, you can request that the Spot fleet keep the fleet at its current size, but not replace any Spot instances that are interrupted or that you terminate manually.

Note that when a Spot fleet terminates an instance because the target capacity was decreased, the instance receives a Spot instance termination notice.

To modify a Spot fleet request using the console

1. Open the Spot console at <https://console.aws.amazon.com/ec2spot/home/fleet>.
2. Select your Spot fleet request.
3. Choose **Actions**, and then choose **Modify target capacity**.
4. In **Modify target capacity**, do the following:
 - a. Enter the new target capacity.
 - b. (Optional) If you are decreasing the target capacity but want to keep the fleet at its current size, deselect **Terminate instances**.
 - c. Choose **Submit**.

To modify a Spot fleet request using the AWS CLI

Use the following [modify-spot-fleet-request](#) command to update the target capacity of the specified Spot fleet request:

```
aws ec2 modify-spot-fleet-request --spot-fleet-request-id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE --target-capacity 20
```

You can modify the previous command as follows to decrease the target capacity of the specified Spot fleet without terminating any Spot instances as a result:

```
aws ec2 modify-spot-fleet-request --spot-fleet-request-id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE --target-capacity 10 --excess-capacity-termination-policy NoTermination
```

Cancelling a Spot Fleet Request

When you are finished using your Spot fleet, you can cancel the Spot fleet request. This cancels all Spot requests associated with the Spot fleet, so that no new Spot instances are launched for your Spot fleet. You must specify whether the Spot fleet should terminate its Spot instances. If you terminate the instances, the Spot fleet request enters the `cancelled_terminating` state. Otherwise, the Spot fleet request enters the `cancelled_running` state and the instances continue to run until they are interrupted or you terminate them manually.

To cancel a Spot fleet request using the console

1. Open the Spot console at <https://console.aws.amazon.com/ec2spot/home/fleet>.
2. Select your Spot fleet request.
3. Choose **Actions**, and then choose **Cancel spot request**.
4. In **Cancel spot request**, verify that you want to cancel the Spot fleet. To keep the fleet at its current size, deselect **Terminate instances**. When you are ready, choose **Confirm**.

To cancel a Spot fleet request using the AWS CLI

Use the following [cancel-spot-fleet-requests](#) command to cancel the specified Spot fleet request and terminate the instances:

```
aws ec2 cancel-spot-fleet-requests --spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE --terminate-instances
```

The following is example output:

```
{
  "SuccessfulFleetRequests": [
    {
      "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
      "CurrentSpotFleetRequestState": "cancelled_terminating",
      "PreviousSpotFleetRequestState": "active"
    }
  ],
  "UnsuccessfulFleetRequests": []
}
```

You can modify the previous command as follows to cancel the specified Spot fleet request without terminating the instances:

```
aws ec2 cancel-spot-fleet-requests --spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE --no-terminate-instances
```

The following is example output:

```
{
  "SuccessfulFleetRequests": [
    {
      "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
      "CurrentSpotFleetRequestState": "cancelled_running",
      "PreviousSpotFleetRequestState": "active"
    }
  ],
  "UnsuccessfulFleetRequests": []
}
```

Spot Fleet Example Configurations

The following examples show launch configurations that you can use with the `request-spot-fleet` command to create a Spot fleet request. For more information, see [Creating a Spot Fleet Request](#) (p. 201).

1. [Launch Spot instances using the lowest-priced Availability Zone or subnet in the region](#) (p. 205)
2. [Launch Spot instances using the lowest-priced Availability Zone or subnet in a specified list](#) (p. 206)
3. [Launch Spot instances using the lowest-priced instance type in a specified list](#) (p. 207)
4. [Override the Spot price for the request](#) (p. 209)
5. [Launch a Spot fleet using the diversified allocation strategy](#) (p. 210)
6. [Launch a Spot fleet using instance weighting](#) (p. 211)

Example 1: Launch Spot Instances Using the Lowest-priced Availability Zone or Subnet in the Region

The following example specifies a single launch specification without an Availability Zone or subnet. If your account supports EC2-VPC only, the Spot fleet launches the instances in the lowest-priced Availability Zone that has a default subnet. If your account supports EC2-Classical, the Spot fleet launches the instances in EC2-Classical in the lowest-priced Availability Zone. Note that the price you pay will not exceed the specified Spot price for the request.

```
{
  "SpotPrice": "0.07",
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
  "LaunchSpecifications": [
    {
      "ImageId": "ami-1a2b3c4d",
      "KeyName": "my-key-pair",
      "SecurityGroups": [
        {
          "GroupId": "sg-1a2b3c4d"
        }
      ],
      "InstanceType": "m3.medium",
      "IamInstanceProfile": {
        "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
      }
    }
  ]
}
```

Example 2: Launch Spot Instances Using the Lowest-priced Availability Zone or Subnet in a Specified List

The following examples specify two launch specifications with different Availability Zones or subnets, but the same instance type and AMI.

Availability Zones

If your account supports EC2-VPC only, the Spot fleet launches the instances in the default subnet of the lowest-priced Availability Zone that you specified. If your account supports EC2-Classic, the Spot fleet launches the instances in the lowest-priced Availability Zone that you specified.

```
{
  "SpotPrice": "0.07",
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
  "LaunchSpecifications": [
    {
      "ImageId": "ami-1a2b3c4d",
      "KeyName": "my-key-pair",
      "SecurityGroups": [
        {
          "GroupId": "sg-1a2b3c4d"
        }
      ],
      "InstanceType": "m3.medium",
      "Placement": {
        "AvailabilityZone": "us-west-2a, us-west-2b"
      },
      "IamInstanceProfile": {
        "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
      }
    }
  ]
}
```

Subnets

You can specify default subnets or nondefault subnets, and the nondefault subnets can be from a default VPC or a nondefault VPC. The Spot service launches the instances in whichever subnet is in the lowest-priced Availability Zone.

Note that you can't specify different subnets from the same Availability Zone in a Spot fleet request.

```
{
  "SpotPrice": "0.07",
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
  "LaunchSpecifications": [
    {
      "ImageId": "ami-1a2b3c4d",
      "KeyName": "my-key-pair",
      "SecurityGroups": [
        {
          "GroupId": "sg-1a2b3c4d"
        }
      ],
      "InstanceType": "m3.medium",
      "SubnetId": "subnet-a61dafcf, subnet-65ea5f08",
      "IamInstanceProfile": {
        "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
      }
    }
  ]
}
```

If the instances are launched in a default VPC, they receive a public IPv4 address by default. If the instances are launched in a nondefault VPC, they do not receive a public IPv4 address by default. Use a network interface in the launch specification to assign a public IPv4 address to instances launched in a nondefault VPC. Note that when you specify a network interface, you must include the subnet ID and security group ID using the network interface.

```
...
{
  "ImageId": "ami-1a2b3c4d",
  "KeyName": "my-key-pair",
  "InstanceType": "m3.medium",
  "NetworkInterfaces": [
    {
      "DeviceIndex": 0,
      "SubnetId": "subnet-1a2b3c4d",
      "Groups": [ "sg-1a2b3c4d" ],
      "AssociatePublicIpAddress": true
    }
  ],
  "IamInstanceProfile": {
    "Arn": "arn:aws:iam::880185128111:instance-profile/my-iam-role"
  }
}
...
```

Example 3: Launch Spot Instances Using the Lowest-priced Instance Type in a Specified List

The following examples specify two launch configurations with different instance types, but the same AMI and Availability Zone or subnet. The Spot fleet launches the instances using the specified instance type with the lowest price.

Availability Zone

```
{
  "SpotPrice": "2.80",
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
  "LaunchSpecifications": [
    {
      "ImageId": "ami-1a2b3c4d",
      "SecurityGroups": [
        {
          "GroupId": "sg-1a2b3c4d"
        }
      ],
      "InstanceType": "cc2.8xlarge",
      "Placement": {
        "AvailabilityZone": "us-west-2b"
      }
    },
    {
      "ImageId": "ami-1a2b3c4d",
      "SecurityGroups": [
        {
          "GroupId": "sg-1a2b3c4d"
        }
      ],
      "InstanceType": "r3.8xlarge",
      "Placement": {
        "AvailabilityZone": "us-west-2b"
      }
    }
  ]
}
```

Subnet

```
{
  "SpotPrice": "2.80",
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
  "LaunchSpecifications": [
    {
      "ImageId": "ami-1a2b3c4d",
      "SecurityGroups": [
        {
          "GroupId": "sg-1a2b3c4d"
        }
      ],
      "InstanceType": "cc2.8xlarge",
      "SubnetId": "subnet-1a2b3c4d"
    },
    {
      "ImageId": "ami-1a2b3c4d",
      "SecurityGroups": [
        {
          "GroupId": "sg-1a2b3c4d"
        }
      ]
    }
  ],
}
```

```
        "InstanceType": "r3.8xlarge",  
        "SubnetId": "subnet-1a2b3c4d"  
    }  
]  
}
```

Example 4. Override the Spot Price for the Request

The ability to specify Spot prices for individual launch specifications provides you with additional control over the bidding process. The following examples override the Spot price for the request (0.070) with individual Spot prices for two of the three launch specifications. Note that the Spot price for the request is used for any launch specification that does not specify an individual Spot price. The Spot fleet launches the instances using the instance type with the lowest price.

Availability Zone

```
{  
  "SpotPrice": "1.68",  
  "TargetCapacity": 30,  
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",  
  "LaunchSpecifications": [  
    {  
      "ImageId": "ami-1a2b3c4d",  
      "InstanceType": "c3.2xlarge",  
      "Placement": {  
        "AvailabilityZone": "us-west-2b"  
      },  
      "SpotPrice": "0.04"  
    },  
    {  
      "ImageId": "ami-1a2b3c4d",  
      "InstanceType": "c3.4xlarge",  
      "Placement": {  
        "AvailabilityZone": "us-west-2b"  
      },  
      "SpotPrice": "0.06"  
    },  
    {  
      "ImageId": "ami-1a2b3c4d",  
      "InstanceType": "c3.8xlarge",  
      "Placement": {  
        "AvailabilityZone": "us-west-2b"  
      }  
    }  
  ]  
}
```

Subnet

```
{  
  "SpotPrice": "1.68",  
  "TargetCapacity": 30,  
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",  
  "LaunchSpecifications": [  
    {  
      "ImageId": "ami-1a2b3c4d",  
      "InstanceType": "c3.2xlarge",  
      "SubnetId": "subnet-1a2b3c4d",  
    }  
  ]  
}
```



```
    "SpotPrice": "0.04"  
  },  
  {  
    "ImageId": "ami-1a2b3c4d",  
    "InstanceType": "c3.4xlarge",  
    "SubnetId": "subnet-1a2b3c4d",  
    "SpotPrice": "0.06"  
  },  
  {  
    "ImageId": "ami-1a2b3c4d",  
    "InstanceType": "c3.8xlarge",  
    "SubnetId": "subnet-1a2b3c4d"  
  }  
]  
}
```

Example 5: Launch a Spot Fleet Using the Diversified Allocation Strategy

The following example uses the `diversified` allocation strategy. The launch specifications have different instance types but the same AMI and Availability Zone or subnet. The Spot fleet distributes the 30 instances across the 3 launch specifications, such that there are 10 instances of each type. For more information, see [Spot Fleet Allocation Strategy \(p. 185\)](#).

Availability Zone

```
{  
  "SpotPrice": "0.70",  
  "TargetCapacity": 30,  
  "AllocationStrategy": "diversified",  
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",  
  "LaunchSpecifications": [  
    {  
      "ImageId": "ami-1a2b3c4d",  
      "InstanceType": "c4.2xlarge",  
      "Placement": {  
        "AvailabilityZone": "us-west-2b"  
      }  
    },  
    {  
      "ImageId": "ami-1a2b3c4d",  
      "InstanceType": "m3.2xlarge",  
      "Placement": {  
        "AvailabilityZone": "us-west-2b"  
      }  
    },  
    {  
      "ImageId": "ami-1a2b3c4d",  
      "InstanceType": "r3.2xlarge",  
      "Placement": {  
        "AvailabilityZone": "us-west-2b"  
      }  
    }  
  ]  
}
```

Subnet

```
{
```

```
"SpotPrice": "0.70",
"TargetCapacity": 30,
"AllocationStrategy": "diversified",
"IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
"LaunchSpecifications": [
  {
    "ImageId": "ami-1a2b3c4d",
    "InstanceType": "c4.2xlarge",
    "SubnetId": "subnet-1a2b3c4d"
  },
  {
    "ImageId": "ami-1a2b3c4d",
    "InstanceType": "m3.2xlarge",
    "SubnetId": "subnet-1a2b3c4d"
  },
  {
    "ImageId": "ami-1a2b3c4d",
    "InstanceType": "r3.2xlarge",
    "SubnetId": "subnet-1a2b3c4d"
  }
]
}
```

Example 6: Launch a Spot Fleet Using Instance Weighting

The following examples use instance weighting, which means that the bid price is per unit hour instead of per instance hour. Each launch configuration lists a different instance type and a different weight. The Spot fleet selects the instance type with the lowest price per unit hour. The Spot fleet calculates the number of Spot instances to launch by dividing the target capacity by the instance weight. If the result isn't an integer, the Spot fleet rounds it up to the next integer, so that the size of your fleet is not below its target capacity.

If the `r3.2xlarge` bid is successful, Spot provisions 4 of these instances. (Divide 20 by 6 for a total of 3.33 instances, then round up to 4 instances.)

If the `c3.xlarge` bid is successful, Spot provisions 7 of these instances. (Divide 20 by 3 for a total of 6.66 instances, then round up to 7 instances.)

For more information, see [Spot Fleet Instance Weighting \(p. 186\)](#).

Availability Zone

```
{
  "SpotPrice": "0.70",
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
  "LaunchSpecifications": [
    {
      "ImageId": "ami-1a2b3c4d",
      "InstanceType": "r3.2xlarge",
      "Placement": {
        "AvailabilityZone": "us-west-2b"
      },
      "WeightedCapacity": 6
    },
    {
      "ImageId": "ami-1a2b3c4d",
      "InstanceType": "c3.xlarge",

```

```
        "Placement": {
            "AvailabilityZone": "us-west-2b"
        },
        "WeightedCapacity": 3
    }
]
}
```

Subnet

```
{
  "SpotPrice": "0.70",
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
  "LaunchSpecifications": [
    {
      "ImageId": "ami-1a2b3c4d",
      "InstanceType": "r3.2xlarge",
      "SubnetId": "subnet-1a2b3c4d",
      "WeightedCapacity": 6
    },
    {
      "ImageId": "ami-1a2b3c4d",
      "InstanceType": "c3.xlarge",
      "SubnetId": "subnet-1a2b3c4d",
      "WeightedCapacity": 3
    }
  ]
}
```

Priority

You can also use instance weighting to give priority to an Availability Zone or subnet. For example, the following launch specifications are nearly identical, except that they specify different subnets and weights. The Spot fleet finds the specification with the highest value for `WeightedCapacity`, and attempts to provision the request in the least expensive Spot instance pool in that subnet. (Note that the second launch specification does not include a weight, so it defaults to 1.)

```
{
  "SpotPrice": "0.42",
  "TargetCapacity": 40,
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
  "LaunchSpecifications": [
    {
      "ImageId": "ami-1a2b3c4d",
      "InstanceType": "c3.2xlarge",
      "SubnetId": "subnet-482e4972",
      "WeightedCapacity": 2
    },
    {
      "ImageId": "ami-1a2b3c4d",
      "InstanceType": "c3.2xlarge",
      "SubnetId": "subnet-bb3337d"
    }
  ]
}
```

CloudWatch Metrics for Spot Fleet

Amazon EC2 provides Amazon CloudWatch metrics that you can use to monitor your Spot fleet.

Important

To ensure accuracy, we recommend that you enable detailed monitoring when using these metrics. For more information, see [Enable or Disable Detailed Monitoring for Your Instances \(p. 575\)](#).

For more information about CloudWatch metrics provided by Amazon EC2, see [Monitoring Your Instances Using CloudWatch \(p. 575\)](#).

Spot Fleet Metrics

The `AWS/EC2Spot` namespace includes the following metrics, plus the CloudWatch metrics for the Spot instances in your fleet. For more information, see [Instance Metrics \(p. 577\)](#).

The `AWS/EC2Spot` namespace includes the following metrics.

Metric	Description
<code>AvailableInstancePoolsCount</code>	The Spot Instance pools specified in the Spot Fleet request. Units: Count
<code>BidsSubmittedForCapacity</code>	The capacity for which Amazon EC2 has submitted bids. Units: Count
<code>EligibleInstancePoolCount</code>	The Spot Instance pools specified in the Spot Fleet request where Amazon EC2 can fulfill bids. Amazon EC2 will not fulfill bids in pools where your bid price is less than the Spot price or the Spot price is greater than the price for On-Demand instances. Units: Count
<code>FulfilledCapacity</code>	The capacity that Amazon EC2 has fulfilled. Units: Count
<code>MaxPercentCapacityAllocation</code>	The maximum value of <code>PercentCapacityAllocation</code> across all Spot Instance pools specified in the Spot Fleet request. Units: Percent
<code>PendingCapacity</code>	The difference between <code>TargetCapacity</code> and <code>FulfilledCapacity</code> . Units: Count
<code>PercentCapacityAllocation</code>	The capacity allocated for the Spot Instance pool for the specified dimensions. To get the maximum value recorded across all Spot Instance pools, use <code>MaxPercentCapacityAllocation</code> . Units: Percent
<code>TargetCapacity</code>	The target capacity of the Spot Fleet request. Units: Count
<code>TerminatingCapacity</code>	The capacity that is being terminated due to Spot Instance interruptions.

Metric	Description
	Units: Count

If the unit of measure for a metric is `Count`, the most useful statistic is `Average`.

Spot Fleet Dimensions

To filter the data for your Spot fleet, you can use the following dimensions.

Dimensions	Description
<code>AvailabilityZone</code>	Filter the data by Availability Zone.
<code>FleetRequestId</code>	Filter the data by Spot Fleet request.
<code>InstanceType</code>	Filter the data by instance type.

View the CloudWatch Metrics for Your Spot Fleet

You can view the CloudWatch metrics for your Spot fleet using the Amazon CloudWatch console. These metrics are displayed as monitoring graphs. These graphs show data points if the Spot fleet is active.

Metrics are grouped first by namespace, and then by the various combinations of dimensions within each namespace. For example, you can view all Spot fleet metrics, or Spot fleet metrics groups by Spot fleet request ID, instance type, or Availability Zone.

To view Spot fleet metrics

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, under **Metrics**, choose the **EC2 Spot** namespace.
3. (Optional) To filter the metrics by dimension, select one of the following:
 - **Fleet Request Metrics** — Group by Spot fleet request
 - **By Availability Zone** — Group by Spot fleet request and Availability Zone
 - **By Instance Type** — Group by Spot fleet request and instance type
 - **By Availability Zone/Instance Type** — Group by Spot fleet request, Availability Zone, and instance type
4. To view the data for a metric, select the check box next to the metric.

The screenshot shows the Amazon CloudWatch console interface. At the top, there is a search bar with 'EC2 Spot' selected and a search icon. Below the search bar, there are filter options: 'Fleet Request Metrics' (selected), 'By Availability Zone', 'By Instance Type', and 'By Availability Zone/Instance Type'. The main content area displays a list of metrics for the selected namespace and filter. The metrics are grouped under 'EC2 Spot > Fleet Request Metrics'. The list includes columns for 'FleetRequestId' and 'Metric Name'. The 'CPUUtilization' metric is highlighted with a blue background and has a checked checkbox next to it. Other metrics listed are 'AvailableInstancePoolsCount', 'BidsSubmittedForCapacity', and 'DiskReadBytes'.

Automatic Scaling for Spot Fleet

Automatic scaling is the ability to increase or decrease the target capacity of your Spot fleet automatically based on demand. A Spot fleet can either launch instances (scale out) or terminate instances (scale in), within the range that you choose, in response to one or more scaling policies. We recommend that you create two policies, one for scaling out and one for scaling in.

A *scaling policy* uses CloudWatch alarms to trigger the scaling process. For example, if you want to scale out when CPU utilization reaches a certain level, create an alarm using the `CPUUtilization` metric provided by Amazon EC2.

When you create a scaling policy, you must specify one of the following scaling adjustment types:

- **Add** — Increase the target capacity of the fleet by a specified number of capacity units or a specified percentage of the current capacity.
- **Remove** — Decrease the target capacity of the fleet by a specified number of capacity units or a specified percentage of the current capacity.
- **Set to** — Set the target capacity of the fleet to the specified number of capacity units.

You can also configure the cooldown period for a scaling policy. This is the number of seconds after a scaling activity completes where previous trigger-related scaling activities can influence future scaling events. For scale out policies, while the cooldown period is in effect, the capacity that has been added by the previous scale out event that initiated the cooldown is calculated as part of the desired capacity for the next scale out. The intention is to continuously (but not excessively) scale out. For scale in policies, the cooldown period is used to block subsequent scale in requests until it has expired. The intention is to scale in conservatively to protect your application's availability. However, if another alarm triggers a scale out policy during the cooldown period after a scale-in, auto scaling scales out your scalable target immediately.

Note that when a Spot fleet terminates an instance because the target capacity was decreased, the instance receives a Spot instance termination notice.

Limits

- The Spot fleet request must have a request type of `maintain`. Automatic scaling is not supported for one-time requests or Spot blocks.

Prerequisites

- Consider which CloudWatch metrics are important to your application. You can create CloudWatch alarms based on metrics provided by AWS or your own custom metrics.
- For the AWS metrics that you will use in your scaling policies, enable CloudWatch metrics collection if the service that provides the metrics does not enable it by default.
- If you use the AWS Management Console to enable automatic scaling for your Spot fleet, it creates a role named `aws-ec2-spot-fleet-autoscale-role` that grants Auto Scaling permission to describe the alarms for your policies, monitor the current capacity of the fleet, and modify the capacity of the fleet. If you configure automatic scaling using the AWS CLI or an API, you can use this role if it exists, or manually create your own role for this purpose as follows.
 1. Open the Identity and Access Management (IAM) console at <https://console.aws.amazon.com/iam/>.
 2. In the navigation pane, choose **Roles**.
 3. Choose **Create New Role**.
 4. On the **Set Role Name** page, type a name for the role and then choose **Next Step**.
 5. On the **Select Role Type** page, choose **Select** next to **Amazon EC2**.

6. On the **Attach Policy** page, select the `AmazonEC2SpotFleetAutoscaleRole` policy and then choose **Next Step**.
7. On the **Review** page, choose **Create Role**.
8. Select the role that you just created.
9. On the **Trust Relationships** tab, choose **Edit Trust Relationship**.
10. Change `ec2.amazonaws.com` to `application-autoscaling.amazonaws.com` and then choose **Update Trust Policy**.

To create a CloudWatch alarm

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Alarms**.
3. Choose **Create Alarm**.
4. For **CloudWatch Metrics by Category**, choose a category. For example, choose **EC2 Spot Metrics**, **Fleet Request Metrics**.
5. Select a metric, and then choose **Next**.
6. For **Alarm Threshold**, type a name and description for the alarm, and set the threshold value and number of time periods for the alarm.
7. (Optional) To receive notification of a scaling event, for **Actions**, choose **New list** and type your email address. Otherwise, you can delete the notification now and add one later if needed.
8. Choose **Create Alarm**.

To configure automatic scaling for your Spot fleet using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Spot Requests**.
3. Select your Spot fleet request, and then choose the **Auto Scaling** tab.
4. If automatic scaling is not configured, choose **Configure**.
5. Use **Scale capacity between** to set the minimum and maximum capacity for your fleet. Automatic scaling will not scale your fleet below the minimum capacity or above the maximum capacity.
6. Initially, **Scaling policies** contains policies named `ScaleUp` and `ScaleDown`. You can complete these policies, or choose **Remove policy** to delete them. You can also choose **Add policy** to add a policy.
7. To define a policy, do the following:
 - a. For **Policy name**, type a name for the policy.
 - b. For **Policy trigger**, select an existing alarm or choose **Create new alarm** to open the Amazon CloudWatch console and create an alarm.
 - c. For **Modify capacity**, select a scaling adjustment type, select a number, and select a unit.
 - d. (Optional) To perform step scaling, choose **Define steps**. By default, an add policy has a lower bound of -infinity and an upper bound of the alarm threshold. By default, a remove policy has a lower bound of the alarm threshold and an upper bound of +infinity. To add another step, choose **Add step**.
 - e. (Optional) To modify the default value for the cooldown period, select a number from **Cooldown period**.
8. Choose **Save**.

To configure automatic scaling for your Spot fleet using the AWS CLI

1. Register the Spot fleet request as a scalable target using the `register-scalable-target` command.

2. Create a scaling policy using the `put-scaling-policy` command.
3. Create an alarm that will trigger the scaling policy using the `put-metric-alarm` command.

Spot Bid Status

To help you track your Spot instance requests, plan your use of Spot instances, and bid strategically, Amazon EC2 provides a *bid status*. For example, a bid status can tell you the reason why your Spot request isn't fulfilled yet, or list the constraints that are preventing the fulfillment of your Spot request.

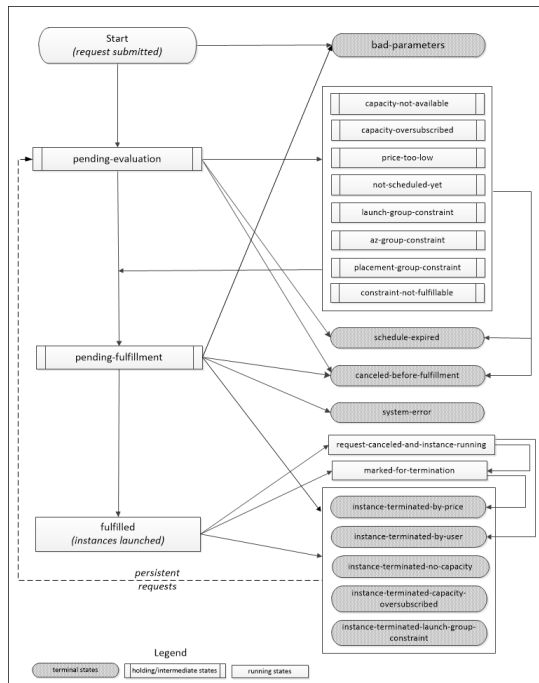
At each step of the process—also called the Spot request *life cycle*, specific events determine successive request states.

Contents

- [Life Cycle of a Spot Request](#) (p. 217)
- [Getting Bid Status Information](#) (p. 220)
- [Spot Bid Status Codes](#) (p. 220)

Life Cycle of a Spot Request

The following diagram shows you the paths that your Spot request can follow throughout its life cycle, from submission to termination. Each step is depicted as a node, and the status code for each node describes the status of the Spot request and Spot instance.



Pending evaluation

As soon as you make a Spot instance request, it goes into the `pending-evaluation` state unless one or more request parameters is not valid (`bad-parameters`).

Status Code	Request State	Instance State
<code>pending-evaluation</code>	<code>open</code>	n/a

Status Code	Request State	Instance State
bad-parameters	closed	n/a

Holding

If one or more request constraints are valid but can't be met yet, or if there is not enough capacity, the request goes into a holding state waiting for the constraints to be met. The request options affect the likelihood of the request being fulfilled. For example, if you specify a bid price below the current Spot price, your request stays in a holding state until the Spot price goes below your bid price. If you specify an Availability Zone group, the request stays in a holding state until the Availability Zone constraint is met.

Status Code	Request State	Instance State
capacity-not-available	open	n/a
capacity-oversubscribed	open	n/a
price-too-low	open	n/a
not-scheduled-yet	open	n/a
launch-group-constraint	open	n/a
az-group-constraint	open	n/a
placement-group-constraint	open	n/a
constraint-not-fulfillable	open	n/a

Pending evaluation/fulfillment-terminal

Your Spot instance request can go to a `terminal` state if you create a request that is valid only during a specific time period and this time period expires before your request reaches the pending fulfillment phase, you cancel the request, or a system error occurs.

Status Code	Request State	Instance State
schedule-expired	closed	n/a
canceled-before-fulfillment*	cancelled	n/a
bad-parameters	failed	n/a
system-error	closed	n/a

* If you cancel the request.

Pending fulfillment

When the constraints you specified (if any) are met and your bid price is equal to or higher than the current Spot price, your Spot request goes into the `pending-fulfillment` state.

At this point, Amazon EC2 is getting ready to provision the instances that you requested. If the process stops at this point, it is likely to be because it was cancelled by the user before a Spot instance was launched, or because an unexpected system error occurred.

Status Code	Request State	Instance State
pending-fulfillment	open	n/a

Fulfilled

When all the specifications for your Spot instances are met, your Spot request is fulfilled. Amazon EC2 launches the Spot instances, which can take a few minutes.

Status Code	Request State	Instance State
fulfilled	active	pending → running

Fulfilled-terminal

Your Spot instances continue to run as long as your bid price is at or above the Spot price, there is spare Spot capacity for your instance type, and you don't terminate the instance. If a change in Spot price or available capacity requires Amazon EC2 to terminate your Spot instances, the Spot request goes into a terminal state. For example, if your bid equals the Spot price but Spot instances are oversubscribed at that price, the status code is `instance-terminated-capacity-oversubscribed`. A request also goes into the terminal state if you cancel the Spot request or terminate the Spot instances.

Status Code	Request State	Instance State
request-canceled-and-instance-running	cancelled	running
marked-for-termination	closed	running
instance-terminated-by-price	closed (one-time), open (persistent)	terminated
instance-terminated-by-user	closed or cancelled *	terminated
instance-terminated-no-capacity	closed (one-time), open (persistent)	terminated
instance-terminated-capacity-oversubscribed	closed (one-time), open (persistent)	terminated
instance-terminated-launch-group-constraint	closed (one-time), open (persistent)	terminated

* The request state is `closed` if you terminate the instance but do not cancel the bid. The request state is `cancelled` if you terminate the instance and cancel the bid. Note that even if you terminate a Spot instance before you cancel its request, there might be a delay before Amazon EC2 detects that your Spot instance was terminated. In this case, the request state can either be `closed` or `cancelled`.

Persistent requests

When your Spot instances are terminated (either by you or Amazon EC2), if the Spot request is a persistent request, it returns to the `pending-evaluation` state and then Amazon EC2 can launch a new Spot instance when the constraints are met.

Getting Bid Status Information

You can get bid status information using the AWS Management Console or a command line tool.

To get bid status information using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Spot Requests**, and then select the Spot request.
3. Check the value of **Status** in the **Description** tab.

To get bid status information using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [describe-spot-instance-requests](#) (AWS CLI)
- [Get-EC2SpotInstanceRequest](#) (AWS Tools for Windows PowerShell)

Spot Bid Status Codes

Spot bid status information is composed of a bid status code, the update time, and a status message. Together, they help you determine the disposition of your Spot request.

The following list describes the Spot bid status codes:

`az-group-constraint`

Amazon EC2 cannot launch all the instances you requested in the same Availability Zone.

`bad-parameters`

One or more parameters for your Spot request are not valid (for example, the AMI you specified does not exist). The bid status message indicates which parameter is not valid.

`cancelled-before-fulfillment`

The user cancelled the Spot request before it was fulfilled.

`capacity-not-available`

There is not enough capacity available for the instances that you requested.

`capacity-oversubscribed`

The number of Spot requests with bid prices equal to or higher than your bid price exceeds the available capacity in this Spot instance pool.

`constraint-not-fulfillable`

The Spot request can't be fulfilled because one or more constraints are not valid (for example, the Availability Zone does not exist). The bid status message indicates which constraint is not valid.

`fulfilled`

The Spot request is `active`, and Amazon EC2 is launching your Spot instances.

`instance-terminated-by-price`

The Spot price rose above your bid price. If your request is a persistent bid, the process restarts, so your bid is pending evaluation.

`instance-terminated-by-user` or `spot-instance-terminated-by-user`

You terminated a Spot instance that had been fulfilled, so the bid state is `closed` (unless it's a persistent bid) and the instance state is `terminated`.

`instance-terminated-capacity-oversubscribed`

Your instance is terminated because the number of Spot requests with bid prices equal to or higher than your bid price exceeded the available capacity in this Spot instance pool. (Note that the Spot price might not have changed.) The Spot service randomly selects instances to be terminated.

`instance-terminated-launch-group-constraint`

One or more of the instances in your launch group was terminated, so the launch group constraint is no longer fulfilled.

`instance-terminated-no-capacity`

There is no longer enough Spot capacity available for the instance.

`launch-group-constraint`

Amazon EC2 cannot launch all the instances that you requested at the same time. All instances in a launch group are started and terminated together.

`marked-for-termination`

The Spot instance is marked for termination.

`not-scheduled-yet`

The Spot request will not be evaluated until the scheduled date.

`pending-evaluation`

After you make a Spot instance request, it goes into the `pending-evaluation` state while the system evaluates the parameters of your request.

`pending-fulfillment`

Amazon EC2 is trying to provision your Spot instances.

`placement-group-constraint`

The Spot request can't be fulfilled yet because a Spot instance can't be added to the placement group at this time.

`price-too-low`

The bid request can't be fulfilled yet because the bid price is below the Spot price. In this case, no instance is launched and your bid remains `open`.

`request-cancelled-and-instance-running`

You canceled the Spot request while the Spot instances are still running. The request is cancelled, but the instances remain `running`.

`schedule-expired`

The Spot request expired because it was not fulfilled before the specified date.

`system-error`

There was an unexpected system error. If this is a recurring issue, please contact customer support for assistance.

Spot Instance Interruptions

Demand for Spot instances can vary significantly from moment to moment, and the availability of Spot instances can also vary significantly depending on how many unused EC2 instances are available. In addition, no matter how high you bid, it is still possible that your Spot instance will be interrupted. Therefore, you must ensure that your application is prepared for a Spot instance interruption. We strongly recommend that you do not use Spot instances for applications that can't be interrupted.

The following are the possible reasons that Amazon EC2 will terminate your Spot instances:

- **Price**—The Spot price is greater than your bid price.
- **Capacity**—If there are not enough unused EC2 instances to meet the demand for Spot instances, Amazon EC2 terminates Spot instances, starting with those instances with the lowest bid prices. If there are several Spot instances with the same bid price, the order in which the instances are terminated is determined at random.

- Constraints—If your request includes a constraint such as a launch group or an Availability Zone group, these Spot instances are terminated as a group when the constraint can no longer be met.

Preparing for Interruptions

Here are some best practices to follow when you use Spot instances:

- Choose a reasonable bid price. Your bid price should be high enough to make it likely that your request will be fulfilled, but not higher than you are willing to pay. This is important because if the supply is low for an extended period of time, the Spot price can remain high during that period because it is based on the highest bid prices. We strongly recommend against bidding above the price for On-Demand instances.
- Ensure that your instance is ready to go as soon as the request is fulfilled by using an Amazon Machine Image (AMI) that contains the required software configuration. You can also use user data to run commands at start-up.
- Store important data regularly in a place that won't be affected when the Spot instance terminates. For example, you can use Amazon S3, Amazon EBS, or DynamoDB.
- Divide the work into small tasks (using a Grid, Hadoop, or queue-based architecture) or use checkpoints so that you can save your work frequently.
- Use Spot instance termination notices to monitor the status of your Spot instances.
- Test your application to ensure that it handles an unexpected instance termination gracefully. You can do so by running the application using an On-Demand instance and then terminating the On-Demand instance yourself.

Spot Instance Termination Notices

The best way to protect against Spot instance interruption is to architect your application to be fault tolerant. In addition, you can take advantage of *Spot instance termination notices*, which provide a two-minute warning before Amazon EC2 must terminate your Spot instance.

This warning is made available to the applications on your Spot instance using an item in the instance metadata. For example, you can check for this warning in the instance metadata periodically (we recommend every 5 seconds) using the following query:

```
C:\> invoke-restmethod -uri http://169.254.169.254/latest/meta-data/spot/termination-time
```

For information about other ways to retrieve instance metadata, see [Retrieving Instance Metadata \(p. 271\)](#).

If your Spot instance is marked for termination by Amazon EC2, the `termination-time` item is present and it specifies the approximate time in UTC when the instance will receive the shutdown signal. For example:

```
2015-01-05T18:02:00Z
```

If Amazon EC2 is not preparing to terminate the instance, or if you terminated the Spot instance yourself, the `termination-time` item is either not present (so you receive an HTTP 404 error) or contains a value that is not a time value.

Note that while we make every effort to provide this warning the moment that your Spot instance is marked for termination by Amazon EC2, it is possible that your Spot instance will be terminated before Amazon EC2 can make the warning available. Therefore, you must ensure that your application

is prepared to handle an unexpected Spot instance interruption even if you are checking for Spot instance termination notices.

If Amazon EC2 fails to terminate the instance, the Spot bid status is set to `fulfilled`. Note that `termination-time` remains in the instance metadata with the original approximate time, which is now in the past.

Spot Instance Data Feed

To help you understand the charges for your Spot instances, Amazon EC2 provides a data feed that describes your Spot instance usage and pricing. This data feed is sent to an Amazon S3 bucket that you specify when you subscribe to the data feed.

Data feed files arrive in your bucket typically once an hour, and each hour of usage is typically covered in a single data file. These files are compressed (gzip) before they are delivered to your bucket. Amazon EC2 can write multiple files for a given hour of usage where files are very large (for example, when file contents for the hour exceed 50 MB before compression).

Note

If you don't have a Spot instance running during a certain hour, you won't receive a data feed file for that hour.

Contents

- [Data Feed File Name and Format \(p. 223\)](#)
- [Amazon S3 Bucket Requirements \(p. 224\)](#)
- [Subscribing to Your Spot instance Data Feed \(p. 224\)](#)
- [Deleting Your Spot Instance Data Feed \(p. 224\)](#)

Data Feed File Name and Format

The Spot instance data feed file name uses the following format (with the date and hour in UTC):

```
bucket-name.s3.amazonaws.com/{optional prefix}/aws-account-id.YYYY-MM-DD-  
HH.n.unique-id.gz
```

For example, if your bucket name is `myawsbucket` and your prefix is `myprefix`, your file names are similar to the following:

```
myawsbucket.s3.amazonaws.com/myprefix/  
111122223333.2014-03-17-20.001.pwBdGTJG.gz
```

The Spot instance data feed files are tab-delimited. Each line in the data file corresponds to one instance hour and contains the fields listed in the following table.

Field	Description
Timestamp	The timestamp used to determine the price charged for this instance hour.
UsageType	The type of usage and instance type being charged for. For <code>m1.small</code> Spot instances, this field is set to <code>SpotUsage</code> . For all other instance types, this field is set to <code>SpotUsage:{instance-type}</code> . For example, <code>SpotUsage:c1.medium</code> .
Operation	The product being charged for. For Linux Spot instances, this field is set to <code>RunInstances</code> . For Windows Spot instances, this field is set to <code>RunInstances:0002</code> . Spot usage is grouped according to Availability Zone.
InstanceID	The ID of the Spot instance that generated this instance hour.

Field	Description
MyBidID	The ID for the Spot instance request that generated this instance hour.
MyMaxPrice	The maximum price specified for this Spot instance request.
MarketPrice	The Spot price at the time specified in the <code>Timestamp</code> field.
Charge	The price charged for this instance hour.
Version	The version included in the data feed file name for this record.

Amazon S3 Bucket Requirements

When you subscribe to the data feed, you must specify an Amazon S3 bucket to store the data feed files. Before you choose an Amazon S3 bucket for the data feed, consider the following:

- You must use a bucket from the US East (N. Virginia) (`us-east-1`) region.
- You must have `FULL_CONTROL` permission to the bucket.

If you're the bucket owner, you have this permission by default. Otherwise, the bucket owner must grant your AWS account this permission.

- When you create your data feed subscription, Amazon S3 updates the ACL of the specified bucket to allow the AWS data feed account read and write permissions.
- Removing the permissions for the data feed account does not disable the data feed. If you remove those permissions but don't disable the data feed, we restore those permissions the next time that the data feed account needs to write to the bucket.
- Each data feed file has its own ACL (separate from the ACL for the bucket). The bucket owner has `FULL_CONTROL` permission to the data files. The data feed account has read and write permissions.
- If you delete your data feed subscription, Amazon EC2 doesn't remove the read and write permissions for the data feed account on either the bucket or the data files. You must remove these permissions yourself.

Subscribing to Your Spot instance Data Feed

To subscribe to your data feed, use the following [create-spot-datafeed-subscription](#) command:

```
C:\> aws ec2 create-spot-datafeed-subscription --bucket myawsbucket [--  
prefix myprefix]
```

The following is example output:

```
{  
  "SpotDatafeedSubscription": {  
    "OwnerId": "111122223333",  
    "Prefix": "myprefix",  
    "Bucket": "myawsbucket",  
    "State": "Active"  
  }  
}
```

Deleting Your Spot Instance Data Feed

To delete your data feed, use the following [delete-spot-datafeed-subscription](#) command:

```
C:\> aws ec2 delete-spot-datafeed-subscription
```

Spot Instance Limits

Spot instance requests are subject to the following limits:

Limits

- [Unsupported Instance Types](#) (p. 225)
- [Spot Request Limits](#) (p. 225)
- [Spot Bid Price Limit](#) (p. 225)
- [Spot Fleet Limits](#) (p. 225)
- [Amazon EBS Encryption Unsupported](#) (p. 226)

Unsupported Instance Types

The following instance types are not supported for Spot:

- T2
- HS1

Some Spot instance types aren't available in every region. To view the supported instance types for a region, go to [Spot Instance Pricing](#) and select the region.

Spot Request Limits

By default, there is an account limit of 20 Spot instances per region. If you terminate your Spot instance but do not cancel the request, the request counts against this limit until Amazon EC2 detects the termination and closes the request.

Spot instance limits are dynamic. When your account is new, your limit might be lower than 20 to start, but increase over time. In addition, your account might have limits on specific Spot instance types. If you submit a Spot instance request and you receive the error `Max spot instance count exceeded`, you can go to [AWS Support Center](#) and submit a limit increase request form. For **Use Case Description**, indicate that you need an increase in your limits for Spot instance requests.

Spot Bid Price Limit

The bid price limit for Spot instances is ten times the On-Demand price. This limit is designed to help you control costs.

Spot Fleet Limits

The usual Amazon EC2 limits apply to instances launched by a Spot fleet, such as Spot bid price limits, instance limits, and volume limits. In addition, the following limits apply:

- The number of active Spot fleets per region: 1,000
- The number of launch specifications per fleet: 50
- The size of the user data in a launch specification: 16 KB
- The target capacity per Spot fleet: 3,000
- The target capacity across all Spot fleets in a region: 5,000
- A Spot fleet request can't span regions.
- A Spot fleet request can't span different subnets from the same Availability Zone.

Amazon EBS Encryption Unsupported

You can specify encrypted EBS volumes in the launch specification for your Spot instances, but these volumes are not encrypted.

Dedicated Hosts

An Amazon EC2 Dedicated Host is a physical server with EC2 instance capacity fully dedicated to your use. Dedicated Hosts allow you to use your existing per-socket, per-core, or per-VM software licenses, including Windows Server, Microsoft SQL Server, SUSE, Linux Enterprise Server, and so on.

Contents

- [Differences between Dedicated Hosts and Dedicated Instances \(p. 226\)](#)
- [Pricing and Billing \(p. 226\)](#)
- [Dedicated Hosts Limitations and Restrictions \(p. 228\)](#)
- [Dedicated Host Configurations \(p. 228\)](#)
- [Using Dedicated Hosts \(p. 228\)](#)
- [Monitoring Dedicated Hosts \(p. 236\)](#)

Differences between Dedicated Hosts and Dedicated Instances

Dedicated Hosts and Dedicated Instances can both be used to launch Amazon EC2 instances onto physical servers that are dedicated for your use.

There are no performance, security, or physical differences between Dedicated Instances and instances on Dedicated Hosts. However, Dedicated Hosts give you additional visibility and control over how instances are placed on a physical server.

When you use Dedicated Hosts, you have control over instance placement on the host using the Host Affinity and Instance Auto-placement settings. With Dedicated Instances, you don't have control over which host your instance launches and runs on. If your organization wants to use AWS, but has an existing software license with hardware compliance requirements, this allows visibility into the host's hardware so you can meet those requirements.

For more information about the differences between Dedicated Hosts and Dedicated Instances, see [Amazon EC2 Dedicated Hosts](#).

For more information about working with Dedicated Hosts and Dedicated Instances, see [Modifying Instance Tenancies \(p. 232\)](#).

Pricing and Billing

On-Demand Dedicated Hosts

On-Demand billing is automatically activated when you allocate a Dedicated Host to your account.

You are billed an hourly On-Demand rate. Rates vary based on the instance type that the Dedicated Host supports and the region in which the Dedicated Host is running. The instance type size or the number of instances that are running on the Dedicated Host do not have an impact on the cost of the host.

To terminate On-Demand billing, you must first stop instances running on the Dedicated Host and then release it. For more information, see [Managing and Releasing Dedicated Hosts \(p. 233\)](#).

Dedicated Host Reservations

Dedicated Host Reservations provide a billing discount compared to running On-Demand Dedicated Hosts. Reservations are available in three payment options:

- **No Upfront**—No Upfront Reservations provide you with a discount on your Dedicated Host usage over a term and do not require an upfront payment. Available for a one-year term only.
- **Partial Upfront**—A portion of the reservation must be paid upfront and the remaining hours in the term are billed at a discounted rate. Available in one-year and three-year terms.
- **All Upfront**—Provides the lowest effective price. Available in one-year and three-year terms and covers the entire cost of the term upfront, with no additional charges going forward.

You must have active Dedicated Hosts in your account before you can purchase reservations. Each reservation covers a single, specific Dedicated Host in your account. Reservations are applied to the instance family on the host, not the instance size. If you have three Dedicated Hosts with different instance sizes (`m4.xlarge`, `m4.medium`, and `m4.large`) you can associate a single `m4` reservation with all those Dedicated Hosts. The instance family and region of the reservation must match that of the Dedicated Hosts you want to associate it with.

Note

When a reservation is associated with a Dedicated Host, the Dedicated Host can't be released until the reservation's term is over.

Purchasing Dedicated Host Reservations

You can purchase Dedicated Host Reservations using the console or the API.

To purchase Dedicated Host Reservations using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the **Dedicated Hosts** page choose **Dedicated Host Reservations**.
3. Choose **Purchase Dedicated Host Reservation**.
4. On the **Purchase Dedicated Host Reservation** screen, you can search for offerings using the default settings or you can specify a configuration for the offering.
 - **Host instance family**—The options listed correspond with the Dedicated Hosts in your account that are not assigned to a reservation.
 - **Availability Zone**—The Availability Zone of the Dedicated Hosts in your account that aren't assigned to a reservation.
 - **Payment Option**—The payment option for the offering.
 - **Term**—The term of the reservation. Can be one or three years.
5. Choose **Find offering**.
6. Select an offering.
7. Choose the Dedicated Hosts to associate with the Dedicated Host Reservation.
8. Choose **Review**.
9. Review your order and choose **Purchase** to complete the transaction.

Viewing Dedicated Host Reservations

You can view information about the Dedicated Hosts associated with your reservation, the term of the reservation, the payment option selected, and the start and end dates of the reservation.

View details of Dedicated Host Reservations

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

2. On the Dedicated Hosts page, choose **Dedicated Host Reservations**.
3. Choose the reservation from the list provided.
4. Select **Details** for information about the reservation.
5. Select **Hosts** for information about the Dedicated Hosts the reservation is associated with.

Dedicated Hosts Limitations and Restrictions

Before you allocate Dedicated Hosts, take note of the following limitations and restrictions.

- Only BYOL RHEL, SUSE Linux, or Windows AMIs offered by AWS or on the AWS Marketplace can be used with Dedicated Hosts.
- Amazon EC2 instance auto recovery is not supported.
- Up to two On-Demand Dedicated Hosts per instance family, per region can be allocated. It is possible to request a limit increase: [Request to Raise Allocation Limit on Amazon EC2 Dedicated Hosts](#).
- The instances that run on a Dedicated Host can only be launched in a VPC.
- Host limits are independent from instance limits. Instances that you are running on Dedicated Hosts do not count towards your instance limits.
- Auto Scaling groups are not supported.
- Amazon RDS instances are not supported.
- The AWS Free Usage tier is not available for Dedicated Hosts.
- Instance placement control refers to managing instance launches onto Dedicated Hosts. Placement groups are not supported for Dedicated Hosts.

Dedicated Host Configurations

Dedicated Hosts are configured to support a single instance type and size capacity. The number of instances you can launch onto a Dedicated Host depends on the instance type that the Dedicated Host is configured to support. For example, if you allocated a `c3.xlarge` Dedicated Host, you'd have the right to launch up to 8 `c3.xlarge` instances on the Dedicated Host. To determine the number of instance type sizes that you can run on a particular Dedicated Host, see [Amazon EC2 Dedicated Hosts Pricing](#).

Using Dedicated Hosts

To use a Dedicated Host, you first *allocate* hosts for use in your account. You then *launch* instances onto the hosts by specifying `host` tenancy for the instance. The *instance auto-placement* setting allows you to control whether an instance can launch onto a particular host. When an instance is stopped and restarted, the *Host affinity* setting determines whether it's restarted on the same, or a different, host. If you no longer need an On-Demand host, you can stop the instances running on the host, direct them to launch on a different host, and then *release* the Dedicated Host.

Contents

- [Bring Your Own License \(p. 229\)](#)
- [Allocating Dedicated Hosts \(p. 229\)](#)
- [Launching Instances onto Dedicated Hosts \(p. 229\)](#)
- [Understanding Instance Placement and Host Affinity \(p. 231\)](#)
- [Modifying Instance Tenancies \(p. 232\)](#)
- [Managing and Releasing Dedicated Hosts \(p. 233\)](#)
- [API and CLI Command Overview \(p. 234\)](#)
- [Tracking Configuration Changes with AWS Config \(p. 234\)](#)

Bring Your Own License

You can use your own software licenses on Dedicated Hosts. These are the general steps you need to follow in order to bring your own volume licensed machine image into Amazon EC2.

1. Verify that the license terms controlling the use of your machine images (AMIs) allow the usage of a machine image in a virtualized cloud environment. For more information about Microsoft Licensing, see [Amazon Web Services and Microsoft Licensing](#).
2. After you have verified that your machine image can be used within Amazon EC2, import your machine images using the `ImportImage` API operation made available by the VM Import/Export tools. For information about restrictions and limitations, see [VM Import/Export Prerequisites](#). For information about how to import your VM using `ImportImage`, see [Importing a VM into Amazon EC2 Using ImportImage](#).
3. If you need a mechanism to track how your images were used in AWS, enable host recording in the AWS Config service. You can use AWS Config to record configuration changes to a Dedicated Host and use the output as a data source for license reporting. For more information, see [Tracking Configuration Changes with AWS Config \(p. 234\)](#).
4. After you've imported your machine image, you can launch instances from this image onto active Dedicated Hosts in your account.
5. When you run these instances, depending on the operating system, you may be required to activate these instances against your own KMS server (for example, Windows Server or Windows SQL Server). You cannot activate your imported Windows AMI against the Amazon Windows KMS server.

Allocating Dedicated Hosts

To begin using Dedicated Hosts, they need to be allocated to your account. You can use the AWS Management Console, interact directly with the API, or use the command line interface to perform these tasks. Follow these steps every time you allocate a Dedicated Host.

To allocate Dedicated Hosts to your account

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the **Dedicated Hosts** page, choose **Allocate Dedicated Host**.
3. Configure your host using the options provided:
 - a. **Instance type**—Instance type that will be available on the Dedicated Host.
 - b. **Availability Zone**—The Availability Zone for the Dedicated Host.
 - c. **Allow instance auto-placement**—The default setting is **Off**. The Dedicated Host accepts `host` tenancy instance launches only (provided capacity is available). When instance auto-placement is **On**, any instances with the tenancy of `host`, and matching the Dedicated Host's configuration, can be launched onto the host.
 - d. **Quantity**—The number of hosts to allocate with these settings.
4. Choose **Allocate host**.

The Dedicated Host capacity is made available in your account immediately.

If you launch instances with `tenancy host` but do not have any active Dedicated Hosts in your account, you receive an error and the instance launch fails.

Launching Instances onto Dedicated Hosts

After you have allocated a Dedicated Host, you can launch instances onto it. Instances with the `tenancy host` can be launched onto a specific Dedicated Host or Amazon EC2 can select the appropriate Dedicated Hosts for you (auto-placement). You cannot launch instances with the `tenancy`

`host` if you do not have active Dedicated Hosts in your account with available capacity matching the instance type configuration of the instances you are launching.

Note

The instances launched onto Dedicated Hosts can only be launched in a VPC. For more information, see [Introduction to VPC](#).

Before you launch your instances, take note of the limitations. For more information, see [Dedicated Hosts Limitations and Restrictions \(p. 228\)](#).

Launching instances onto a Dedicated Host from the Dedicated Hosts page

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the **Dedicated Hosts** page, select a host, choose **Actions** and then choose **Launch Instance(s) onto Host**.
3. Select the AMI to use. If you have imported your own AMI, choose **My AMIs** on the left sidebar and select the relevant AMI.
4. Choose the instance type for the Dedicated Host; this is the only instance type you can launch onto the host.
5. On the **Configure Instance Details** page, the **Tenancy** and **Host** options are pre-selected. You can toggle the **Affinity** setting to **On** or **Off**.
 - **On**—If stopped, the instance always restarts on that specific host.
 - **Off**—The instance launches onto the specified Dedicated Host, but is not guaranteed to restart on it if stopped.
6. Complete the rest of the steps and choose **Launch Instances**.

The instance is automatically launched onto the Dedicated Host that you specified. To view the instances on a Dedicated Host, go to the **Dedicated Hosts** page, and select the Dedicated Host that you specified when you launched the instance.

Launching instances onto a specific Dedicated Host from the Instances page

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the **Instances** page, choose **Launch Instance**.
3. Select an AMI from the list. If you have imported your own AMI, choose **My AMIs** and select the imported image. Not all AMIs can be used with Dedicated Hosts.
4. Select the type of instance to launch.
5. On the **Configure Instance Details** page, the Dedicated Host settings are:
 - **Tenancy—Dedicated host — Launch this instance on a Dedicated host**. If you're not able to choose this, check whether you have selected an incompatible AMI or instance type.
 - **Host**—Select a host. If you are unable to select a Dedicated Host, check:
 - Whether the selected subnet is in a different Availability Zone to the host.
 - That the instance type you've selected matches the instance type that the Dedicated Host supports. If you don't have matching, running hosts, the only option available is **Use auto-placement** but the instance launch fails unless there is available, matching Dedicated Host capacity in your account.
 - **Affinity**—The default setting for this is **Off**. The instance launches onto the specified Dedicated Host, but is not guaranteed to restart on it if stopped.

Note

If you are unable to see these settings, check that you have selected a VPC in the **Network** menu.

6. Complete the rest of the configuration steps. Choose **Review and Launch**.

7. Choose **Launch** to launch your instance.
8. Select an existing key pair, or create a new one. Choose **Launch Instances**.

Launching instances onto any Dedicated Host from the Instances page

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the **Instances** page, choose **Launch Instance**.
3. Select an AMI from the list. If you have imported your own AMI, choose **My AMIs** and select the imported image. Not all AMIs can be used with Dedicated Hosts.
4. Select the type of instance to launch.
5. On the **Configure Instance Details** page, the Dedicated Host settings are:
 - **Tenancy—Dedicated host — Launch this instance on a Dedicated host** If you're not able to choose this, check whether you have selected an incompatible AMI or instance type.
 - **Host**—For this type of launch, keep the setting as **Use auto-placement**.
 - **Affinity**—The default setting for this is **Off**. The instance launches onto any available Dedicated Host in your account, but is not guaranteed to restart on that host if stopped.

If you are unable to see these settings, check that you have selected a VPC in the **Network** menu.

6. Complete the rest of the configuration steps. Choose **Review and Launch**.
7. Choose **Launch** to launch your instance.
8. Select an existing key pair, or create a new one. Choose **Launch Instances**.

Understanding Instance Placement and Host Affinity

Placement control happens on both the instance level and host level.

Contents

- [Instance Auto-Placement \(p. 231\)](#)
- [Host Affinity \(p. 231\)](#)
- [Modifying Instance Auto-Placement and Host Affinity \(p. 232\)](#)
- [Modifying Instance Host Affinity \(p. 232\)](#)

Instance Auto-Placement

Auto-placement allows you to manage whether instances that you launch are launched onto a specific host, or onto any host that has matching configurations. The default setting for this is **Off**. This means that the Dedicated Host you are allocating only accepts `host` tenancy instances launches that specify the unique host ID. Instances launched without a host ID specified are not able to launch onto a host that have instance auto-placement set to **Off**.

Host Affinity

Host Affinity establishes a launch relationship between an instance and a Dedicated Host. When affinity is set to `host`, an instance launched onto a specific host always restarts on the same host if stopped. This applies to both targeted and untargeted launches.

If affinity is set to `default`, and you stop and restart the instance, it can be restarted on any available host but tries to launch back onto the last Dedicated Host it ran on (on a best-effort basis).

You can modify the relationship between an instance and a Dedicated Host by changing the affinity from `host` to `default` and vice-versa. For more information, see [Modifying Instance Tenancies \(p. 232\)](#).

Modifying Instance Auto-Placement and Host Affinity

You can manage instance placement controls using the Amazon EC2 console, the API, or CLI.

To modify the instance placement settings of your instances, first stop the instances and then edit the instance placement settings.

Note

If the instance is stopped and restarted, it is not guaranteed to restart on the same Dedicated Host.

To edit an instance's placement settings (any available hosts)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the **Instances** page, select the instance to edit.
3. Choose **Actions, Instance State**, and **Stop**.
4. Choose **Actions, Instance Settings**, and **Modify Instance Placement**.
5. Change the instance tenancy to **Launch this instance on a Dedicated host**.
6. Choose **This instance can run on any one of my Hosts**. The instance launches onto any Dedicated Host that has auto-placement enabled.
7. Choose **Save** to continue.
8. Open the context (right-click) menu on the instance and choose **Instance State, Start**.

To edit an instance's placement settings (specific Dedicated Host)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the **Instances** page, select the instance to edit.
3. Choose **Actions, Instance State**, and **Stop**.
4. Choose **Actions, Instance Settings**, and **Modify Instance Placement**.
5. Change the instance tenancy to **Launch this instance on a Dedicated host**.
6. Choose **This instance can only run on the selected Host**. Then select a value for **Target Host** and choose whether you want the instance to be placed on any available host, or a specific host.
7. Choose **Save** to continue.
8. Open the context (right-click) menu on the instance and choose **Instance State, Start**.

Modifying Instance Host Affinity

If you no longer want an instance to have affinity with a host, you can stop the instance and change its affinity to `default`. This removes the persistence between the instance and the host. However, when you restart the instance, it may launch back onto the same Dedicated Host (depending on Dedicated Host availability in your account, and on a best-effort basis). However, if it is stopped again, it will not restart on the same host.

Modifying Instance Tenancies

You can modify the tenancy of a Dedicated Instance from `dedicated` to `host`, and vice-versa if it is not using a Windows, SUSE, or RHEL AMI provided by Amazon EC2. You need to stop your Dedicated Instance in order to do this. Instances with `shared` tenancy cannot be modified to `host` tenancy.

Modify instance tenancy from `dedicated` to `host`

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

2. Choose **Instances**, then select the Dedicated Instances to modify.
3. Choose **Actions**, **Instance State**, and **Stop**.
4. Open the context (right-click) menu on the instance and choose **Instance Settings**, **Modify Instance Placement**.
5. On the **Modify Instance Placement** page, do the following:
 - **Tenancy**—Choose **Launch this instance on a Dedicated host**.
 - **Affinity**—Choose either **This instance can run on any one of my Hosts** or **This instance can only run on the selected Host**.

If you choose **This instance can run on any one of my Hosts**, the instance launches onto any available, compatible Dedicated Hosts in your account.

If you choose **This instance can only run on the selected Host**, select a value for **Target Host**. If no target host is listed, you may not have available, compatible Dedicated Hosts in your account.
6. Choose **Save**.
7. When you restart your instance Amazon EC2 places your instance on an available Dedicated Host in your account, provided it supports the instance type that you're launching.

Managing and Releasing Dedicated Hosts

You can use the console, interact directly with the API, or use the command line interface to view details about individual instances on a host and release an On-Demand Dedicated Host.

To view details of instances on a Dedicated Host

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the **Dedicated Hosts** page, select the host to view more information about.
3. Choose the **Description** tab for information about the host. Choose the **Instances** tab for information about instances running on your host.

To release a Dedicated Host

Any running instances on the Dedicated Host need to be stopped before you can release the host. These instances can be migrated to other Dedicated Hosts in your account so that you can continue to use them. For more information, see [Modifying Instance Auto-Placement and Host Affinity \(p. 232\)](#). These steps apply only to On-Demand Dedicated Hosts.

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the **Dedicated Hosts** page, select the Dedicated Host to release.
3. Choose **Actions**, **Release Hosts**.
4. Confirm your choice by choosing **Release**.

After you release a Dedicated Host, you cannot reuse the same host or host ID again.

When the Dedicated Host is released you are no longer charged On-Demand billing rates for it. The Dedicated Host status is changed to `released` and you are not able to launch any instances onto that host.

If you've recently released Dedicated Hosts, it may take some time for them to stop counting towards your limit. During this time, you may experience `LimitExceeded` errors when trying to allocate new Dedicated Hosts. If this is the case, try allocating new hosts again after a few minutes.

The instances that were stopped are still available for use and are listed on the **Instances** page. They retain their `host` tenancy setting.

API and CLI Command Overview

You can perform the tasks described in this section using an API or the command line.

To allocate Dedicated Hosts to your account

- [allocate-hosts](#) (AWS CLI)
- [AllocateHosts](#) (Amazon EC2 Query API)
- [New-EC2Hosts](#) (AWS Tools for Windows PowerShell)

To describe your Dedicated Hosts

- [describe-hosts](#) (AWS CLI)
- [DescribeHosts](#) (Amazon EC2 Query API)
- [Get-EC2Hosts](#) (AWS Tools for Windows PowerShell)

To modify your Dedicated Hosts

- [modify-hosts](#) (AWS CLI)
- [ModifyHosts](#) (Amazon EC2 Query API)
- [Edit-EC2Hosts](#) (AWS Tools for Windows PowerShell)

To modify instance auto-placement

- [modify-instance-placement](#) (AWS CLI)
- [ModifyInstancePlacement](#) (Amazon EC2 Query API)
- [Edit-EC2InstancePlacement](#) (AWS Tools for Windows PowerShell)

To release your Dedicated Hosts

- [release-hosts](#) (AWS CLI)
- [ReleaseHosts](#) (Amazon EC2 Query API)
- [Remove-EC2Hosts](#) (AWS Tools for Windows PowerShell)

Tracking Configuration Changes with AWS Config

You can use AWS Config to record configuration changes for Dedicated Hosts, and instances that are launched, stopped, or terminated on them. You can then use the information captured by AWS Config as a data source for license reporting.

AWS Config records configuration information for Dedicated Hosts and instances individually and pairs this information through relationships. There are three reporting conditions.

- **AWS Config recording status**—When **On**, AWS Config is recording one or more AWS resource types, which can include Dedicated Hosts and Dedicated Instances. To capture the information required for license reporting, verify that hosts and instances are being recorded with the following fields.
- **Host recording status**—When **Enabled**, the configuration information for Dedicated Hosts is recorded.

- **Instance recording status**—When **Enabled**, the configuration information for Dedicated Instances is recorded.

If any of these three conditions are disabled, the icon in the **Edit Config Recording** button is red. To derive the full benefit of this tool, ensure that all three recording methods are enabled. When all three are enabled, the icon is green. To edit the settings, choose **Edit Config Recording**. You are directed to the **Set up AWS Config** page in the AWS Config console, where you can set up AWS Config and start recording for your hosts, instances, and other supported resource types. For more information, see [Setting up AWS Config using the Console](#) in the *AWS Config Developer Guide*.

Note

AWS Config records your resources after it discovers them, which might take several minutes.

After AWS Config starts recording configuration changes to your hosts and instances, you can get the configuration history of any host that you have allocated or released and any instance that you have launched, stopped, or terminated. For example, at any point in the configuration history of a Dedicated Host, you can look up how many instances are launched on that host alongside the number of sockets and cores on the host. For any of those instances, you can also look up the ID of its Amazon Machine Image (AMI). You can use this information to report on licensing for your own server-bound software that is licensed per-socket or per-core.

You can view configuration histories in any of the following ways.

- By using the AWS Config console. For each recorded resource, you can view a timeline page, which provides a history of configuration details. To view this page, choose the grey icon in the **Config Timeline** column of the **Dedicated Hosts** page. For more information, see [Viewing Configuration Details in the AWS Config Console](#) in the *AWS Config Developer Guide*.
- By running AWS CLI commands. First, you can use the `list-discovered-resources` command to get a list of all hosts and instances. Then, you can use the `get-resource-config-history` command to get the configuration details of a host or instance for a specific time interval. For more information, see [View Configuration Details Using the CLI](#) in the *AWS Config Developer Guide*.
- By using the AWS Config API in your applications. First, you can use the `ListDiscoveredResources` action to get a list of all hosts and instances. Then, you can use the `GetResourceConfigHistory` action to get the configuration details of a host or instance for a specific time interval.

For example, to get a list of all of your Dedicated Hosts from AWS Config, run a CLI command such as the following:

```
aws configservice list-discovered-resources --resource-type  
AWS::EC2::Host
```

To obtain the configuration history of a Dedicated Host from AWS Config, run a CLI command such as the following:

```
aws configservice get-resource-config-history --resource type  
AWS::EC2::Instance --resource-id i-36a47fdF
```

To manage AWS Config settings using the AWS Management Console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the **Dedicated Hosts** page, choose **Edit Config Recording**.
3. In the AWS Config console, follow the steps provided to turn on recording. For more information, see [Setting up AWS Config using the Console](#).

For more information, see [Viewing Configuration Details in the AWS Config Console](#).

To activate AWS Config using the command line or API

- Using the AWS CLI, see [Viewing Configuration Details in the AWS Config Console](#) in the *AWS Config Developer Guide*.
- Using the Amazon EC2 API, see [GetResourceConfigHistory](#).

Monitoring Dedicated Hosts

Amazon EC2 constantly monitors the state of your Dedicated Hosts; updates are communicated on the Amazon EC2 console. You can also obtain information about your Dedicated Hosts using the API or CLI.

The following table illustrates the possible **State** values in the console.

State	Description
available	AWS hasn't detected an issue with the Dedicated Host; no maintenance or repairs are scheduled. Instances can be launched onto this Dedicated Host.
released	The Dedicated Host has been released. The host ID is no longer in use. Released hosts cannot be reused.
under-assessment	AWS is exploring a possible issue with the Dedicated Host. If action needs to be taken, you will be notified via the AWS Management Console or email. Instances cannot be launched onto a Dedicated Host in this state.
permanent-failure	An unrecoverable failure has been detected. You will receive an eviction notice through your instances and by email. Your instances may continue to run. If you stop or terminate all instances on a Dedicated Host with this state, AWS retires the host. Instances cannot be launched onto Dedicated Hosts in this state.
released-permanent-failure	AWS permanently releases Dedicated Hosts that have failed and no longer have running instances on them. The Dedicated Host ID is no longer available for use.

Dedicated Instances

Dedicated instances are Amazon EC2 instances that run in a virtual private cloud (VPC) on hardware that's dedicated to a single customer. Your Dedicated instances are physically isolated at the host hardware level from instances that belong to other AWS accounts. Dedicated instances may share hardware with other instances from the same AWS account that are not Dedicated instances.

Note

A *Dedicated Host* is also a physical server that's dedicated for your use. With a Dedicated Host, you have visibility and control over how instances are placed on the server. For more information, see [Dedicated Hosts](#) (p. 226).

Topics

- [Dedicated Instance Basics](#) (p. 237)
- [Working with Dedicated Instances](#) (p. 238)
- [API and Command Overview](#) (p. 240)

Dedicated Instance Basics

Each instance that you launch into a VPC has a tenancy attribute. This attribute has the following values.

Value	Description
default	Your instance runs on shared hardware.
dedicated	Your instance runs on single-tenant hardware.
host	Your instance runs on a Dedicated Host, which is an isolated server with configurations that you can control.

You cannot change the tenancy of a default instance after you've launched it. You can change the tenancy of an instance from `dedicated` to `host` after you've launched it, and vice versa. For more information, see [Changing the Tenancy of an Instance](#) (p. 239).

Each VPC has a related instance tenancy attribute. You can't change the instance tenancy of a VPC after you create it. This attribute has the following values.

Value	Description
default	An instance launched into the VPC runs on shared hardware by default, unless you explicitly specify a different tenancy during instance launch.
dedicated	An instance launched into the VPC is a Dedicated instance by default, unless you explicitly specify a tenancy of <code>host</code> during instance launch. You cannot specify a tenancy of <code>default</code> during instance launch.

To create Dedicated instances, you can do the following:

- Create the VPC with the instance tenancy set to `dedicated` (all instances launched into this VPC are Dedicated instances).
- Create the VPC with the instance tenancy set to `default`, and specify a tenancy of `dedicated` for any instances when you launch them.

Dedicated Instances Limitations

Some AWS services or their features won't work with a VPC with the instance tenancy set to `dedicated`. Check the service's documentation to confirm if there are any limitations.

Some instance types cannot be launched into a VPC with the instance tenancy set to `dedicated`. For more information about supported instances types, see [Amazon EC2 Dedicated Instances](#).

Amazon EBS with Dedicated Instances

When you launch an Amazon EBS-backed Dedicated instance, the EBS volume doesn't run on single-tenant hardware.

Reserved Instances with Dedicated Tenancy

To guarantee that sufficient capacity will be available to launch Dedicated instances, you can purchase Dedicated Reserved Instances. For more information, see [Reserved Instances \(p. 152\)](#).

When you purchase a Dedicated Reserved Instance, you are purchasing the capacity to launch a Dedicated instance into a VPC at a much reduced usage fee; the price break in the hourly charge applies only if you launch an instance with dedicated tenancy. However, if you purchase a Reserved Instance with a default tenancy value, you won't get a Dedicated Reserved Instance if you launch an instance with `dedicated` instance tenancy.

In addition, you can't change the tenancy of a Reserved Instance after you've purchased it.

Auto Scaling of Dedicated Instances

For information about using Auto Scaling to launch Dedicated instances, see [Auto Scaling in Amazon Virtual Private Cloud](#) in the *Auto Scaling User Guide*.

Pricing for Dedicated Instances

Pricing for Dedicated instances is different to pricing for On-Demand instances. For more information, see the [Amazon EC2 Dedicated Instances product page](#).

Working with Dedicated Instances

You can create a VPC with an instance tenancy of `dedicated` to ensure that all instances launched into the VPC are Dedicated instances. Alternatively, you can specify the tenancy of the instance during launch.

Topics

- [Creating a VPC with an Instance Tenancy of Dedicated \(p. 238\)](#)
- [Launching Dedicated Instances into a VPC \(p. 239\)](#)
- [Displaying Tenancy Information \(p. 239\)](#)
- [Changing the Tenancy of an Instance \(p. 239\)](#)

Creating a VPC with an Instance Tenancy of Dedicated

When you create a VPC, you have the option of specifying its instance tenancy. You can create a VPC using the VPC wizard or the **Your VPCs** page in the Amazon VPC console.

To create a VPC with an instance tenancy of dedicated (VPC Wizard)

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. From the dashboard, choose **Start VPC Wizard**.
3. Select a VPC configuration, and then choose **Select**.
4. On the next page of the wizard, choose **Dedicated** from the **Hardware tenancy** list.
5. Choose **Create VPC**.

To create a VPC with an instance tenancy of dedicated (Create VPC dialog box)

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Your VPCs**, and then **Create VPC**.
3. For **Tenancy**, choose **Dedicated**. Specify the CIDR block, and choose **Yes, Create**.

If you launch an instance into a VPC that has an instance tenancy of `dedicated`, your instance is automatically a Dedicated instance, regardless of the tenancy of the instance.

Launching Dedicated Instances into a VPC

You can launch a Dedicated instance using the Amazon EC2 launch instance wizard.

To launch an instance with a tenancy of `dedicated` into a VPC with a tenancy of `default`

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Launch Instance**.
3. On the **Choose an Amazon Machine Image (AMI)** page, select an AMI and choose **Select**.
4. On the **Choose an Instance Type** page, select the instance type and choose **Next: Configure Instance Details**.

Note

Ensure that you choose an instance type that's supported as a Dedicated instance. For more information, see [Amazon EC2 Dedicated Instances](#).

5. On the **Configure Instance Details** page, select a VPC and subnet. Choose **Dedicated - Run a dedicated instance** from the **Tenancy** list, and then **Next: Add Storage**.
6. Continue as prompted by the wizard. When you've finished reviewing your options on the **Review Instance Launch** page, choose **Launch** to choose a key pair and launch the Dedicated instance.

For more information about launching an instance with a tenancy of `host`, see [Launching Instances onto Dedicated Hosts \(p. 229\)](#).

Displaying Tenancy Information

To display tenancy information for your VPC

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Your VPCs**.
3. Check the instance tenancy of your VPC in the **Tenancy** column.
4. If the **Tenancy** column is not displayed, choose **Edit Table Columns** (the gear-shaped icon), **Tenancy** in the **Show/Hide Columns** dialog box, and then **Close**.

To display tenancy information for your instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Check the tenancy of your instance in the **Tenancy** column.
4. If the **Tenancy** column is not displayed, do one of the following:
 - Choose **Edit Table Columns** (the gear-shaped icon), **Tenancy** in the **Show/Hide Columns** dialog box, and then **Close**.
 - Select the instance. The **Description** tab in the details pane displays information about the instance, including its tenancy.

Changing the Tenancy of an Instance

Depending on your instance type and platform, you can change the tenancy of a stopped Dedicated instance to `host` after launching it. The next time the instance starts, it's started on a Dedicated Host that's allocated to your account. For more information about allocating and working with

Dedicated hosts, and the instance types that can be used with Dedicated hosts, see [Using Dedicated Hosts \(p. 228\)](#). Similarly, you can change the tenancy of a stopped Dedicated Host instance to `dedicated` after launching it. The next time the instance starts, it's started on single-tenant hardware that we control.

To change the tenancy of an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**, and then select your instance.
3. Choose **Actions**, then **Instance State**, and then choose **Stop**.
4. Choose **Actions**, then **Instance Settings**, and then choose **Modify Instance Placement**.
5. In the **Tenancy** list, choose whether to run your instance on dedicated hardware or on a Dedicated Host. Choose **Save**.

API and Command Overview

You can perform the tasks described on this page using the command line or an API.

Set the tenancy option when you create a VPC

- [create-vpc](#) (AWS CLI)
- [New-EC2Vpc](#) (AWS Tools for Windows PowerShell)

Describe the supported tenancy options for instances launched into the VPC

- [describe-vpcs](#) (AWS CLI)
- [Get-EC2Vpc](#) (AWS Tools for Windows PowerShell)

Set the tenancy option for an instance during launch

- [run-instances](#) (AWS CLI)
- [New-EC2Instance](#) (AWS Tools for Windows PowerShell)

Describe the tenancy value of an instance

- [describe-instances](#) (AWS CLI)
- [Get-EC2Instance](#) (AWS Tools for Windows PowerShell)

Describe the tenancy value of a Reserved Instance

- [describe-reserved-instances](#) (AWS CLI)
- [Get-EC2ReservedInstance](#) (AWS Tools for Windows PowerShell)

Describe the tenancy value of a Reserved Instance offering

- [describe-reserved-instances-offerings](#) (AWS CLI)
- [Get-EC2ReservedInstancesOffering](#) (AWS Tools for Windows PowerShell)

Modify the tenancy value of an instance

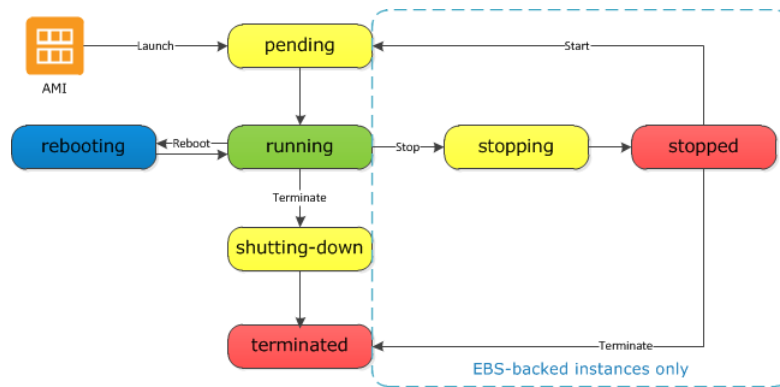
- [modify-instance-placement](#) (AWS CLI)

- [Edit-EC2InstancePlacement](#) (AWS Tools for Windows PowerShell)

Instance Lifecycle

By working with Amazon EC2 to manage your instances from the moment you launch them through their termination, you ensure that your customers have the best possible experience with the applications or sites that you host on your instances.

The following illustration represents the transitions between instance states. Notice that you can't stop and start an instance store-backed instance. For more information about instance store-backed instances, see [Storage for the Root Device](#) (p. 64).



Instance Launch

When you launch an instance, it enters the `pending` state. The instance type that you specified at launch determines the hardware of the host computer for your instance. We use the Amazon Machine Image (AMI) you specified at launch to boot the instance. After the instance is ready for you, it enters the `running` state. You can connect to your running instance and use it the way that you'd use a computer sitting in front of you.

As soon as your instance transitions to the `running` state, you're billed for each hour or partial hour that you keep the instance running; even if the instance remains idle and you don't connect to it.

For more information, see [Launch Your Instance](#) (p. 244) and [Connecting to Your Windows Instance](#) (p. 254).

Instance Stop and Start (Amazon EBS-backed instances only)

If your instance fails a status check or is not running your applications as expected, and if the root volume of your instance is an Amazon EBS volume, you can stop and start your instance to try to fix the problem.

When you stop your instance, it enters the `stopping` state, and then the `stopped` state. We don't charge hourly usage or data transfer fees for your instance after you stop it, but we do charge for the storage for any Amazon EBS volumes. While your instance is in the `stopped` state, you can modify certain attributes of the instance, including the instance type.

When you start your instance, it enters the `pending` state, and in most cases, we move the instance to a new host computer. (Your instance may stay on the same host computer if there are no problems

with the host computer.) When you stop and start your instance, you'll lose any data on the instance store volumes on the previous host computer.

If your instance is running in EC2-Classic, it receives a new private IPv4 address, which means that an Elastic IP address (EIP) associated with the private IPv4 address is no longer associated with your instance. If your instance is running in EC2-VPC, it retains its private IPv4 address, which means that an EIP associated with the private IPv4 address or network interface is still associated with your instance. If your instance has an IPv6 address, it retains its IPv6 address.

Each time you transition an instance from `stopped` to `running`, we charge a full instance hour, even if these transitions happen multiple times within a single hour.

For more information, see [Stop and Start Your Instance \(p. 259\)](#).

Instance Reboot

You can reboot your instance using the Amazon EC2 console, a command line tool, and the Amazon EC2 API. We recommend that you use Amazon EC2 to reboot your instance instead of running the operating system reboot command from your instance.

Rebooting an instance is equivalent to rebooting an operating system; the instance remains on the same host computer and maintains its public DNS name, private IP address, and any data on its instance store volumes. It typically takes a few minutes for the reboot to complete, but the time it takes to reboot depends on the instance configuration.

Rebooting an instance doesn't start a new instance billing hour.

For more information, see [Reboot Your Instance \(p. 262\)](#).

Instance Retirement

An instance is scheduled to be retired when AWS detects irreparable failure of the underlying hardware hosting the instance. When an instance reaches its scheduled retirement date, it is stopped or terminated by AWS. If your instance root device is an Amazon EBS volume, the instance is stopped, and you can start it again at any time. If your instance root device is an instance store volume, the instance is terminated, and cannot be used again.

For more information, see [Instance Retirement \(p. 262\)](#).

Instance Termination

When you've decided that you no longer need an instance, you can terminate it. As soon as the status of an instance changes to `shutting-down` or `terminated`, you stop incurring charges for that instance.

Note that if you enable termination protection, you can't terminate the instance using the console, CLI, or API.

After you terminate an instance, it remains visible in the console for a short while, and then the entry is automatically deleted. You can also describe a terminated instance using the CLI and API. Resources (such as tags) are gradually disassociated from the terminated instance, therefore may no longer be visible on the terminated instance after a short while. You can't connect to or recover a terminated instance.

Each Amazon EBS-backed instance supports the `InstanceInitiatedShutdownBehavior` attribute, which controls whether the instance stops or terminates when you initiate a shutdown from

within the instance itself. The default behavior is to stop the instance. You can modify the setting of this attribute while the instance is running or stopped.

Each Amazon EBS volume supports the `DeleteOnTermination` attribute, which controls whether the volume is deleted or preserved when you terminate the instance it is attached to. The default is to delete the root device volume and preserve any other EBS volumes.

For more information, see [Terminate Your Instance](#) (p. 264).

Differences Between Reboot, Stop, and Terminate

The following table summarizes the key differences between rebooting, stopping, and terminating your instance.

Characteristic	Reboot	Stop/start (Amazon EBS-backed instances only)	Terminate
Host computer	The instance stays on the same host computer	The instance runs on a new host computer	None
Private and public IPv4 addresses	These addresses stay the same	EC2-Classic: The instance gets new private and public IPv4 addresses EC2-VPC: The instance keeps its private IPv4 address. The instance gets a new public IPv4 address, unless it has an Elastic IP address (EIP), which doesn't change during a stop/start.	None
Elastic IP addresses (IPv4)	The Elastic IP remains associated with the instance	EC2-Classic: The Elastic IP is disassociated from the instance EC2-VPC: The Elastic IP remains associated with the instance	The Elastic IP is disassociated from the instance
IPv6 address (EC2-VPC only)	The address stays the same	The instance keeps its IPv6 address	None
Instance store volumes	The data is preserved	The data is erased	The data is erased
Root device volume	The volume is preserved	The volume is preserved	The volume is deleted by default
Billing	The instance billing hour doesn't change.	You stop incurring charges for an instance as soon as its state changes to <code>stopping</code> . Each time an instance transitions from <code>stopped</code>	You stop incurring charges for an instance as soon as its state changes to <code>shutting-down</code> .

Characterist	Reboot	Stop/start (Amazon EBS-backed instances only)	Terminate
		to <code>running</code> , we start a new instance billing hour.	

Note that operating system shutdown commands always terminate an instance store-backed instance. You can control whether operating system shutdown commands stop or terminate an Amazon EBS-backed instance. For more information, see [Changing the Instance Initiated Shutdown Behavior \(p. 267\)](#).

Launch Your Instance

An instance is a virtual server in the AWS cloud. You launch an instance from an Amazon Machine Image (AMI). The AMI provides the operating system, application server, and applications for your instance.

When you sign up for AWS, you can get started with Amazon EC2 for free using the [AWS Free Tier](#). You can either leverage the free tier to launch and use a micro instance for free for 12 months. If you launch an instance that is not within the free tier, you incur the standard Amazon EC2 usage fees for the instance. For more information, see the [Amazon EC2 Pricing](#).

You can launch an instance using the following methods.

Method	Documentation
Use the Amazon EC2 console with an AMI that you select	Launching an Instance (p. 244)
Use the Amazon EC2 console to launch an instance using an existing instance as a template	Launching an Instance Using an Existing Instance as a Template (p. 250)
Use the Amazon EC2 console with an AMI that you purchased from the AWS Marketplace	Launching an AWS Marketplace Instance (p. 251)
Use the AWS CLI with an AMI that you select	Using Amazon EC2 through the AWS CLI
Use the AWS Tools for Windows PowerShell with an AMI that you select	Amazon EC2 from the AWS Tools for Windows PowerShell

After you launch your instance, you can connect to it and use it. To begin, the instance state is `pending`. When the instance state is `running`, the instance has started booting. There might be a short time before you can connect to the instance. The instance receives a public DNS name that you can use to contact the instance from the Internet. The instance also receives a private DNS name that other instances within the same Amazon EC2 network (EC2-Classic or EC2-VPC) can use to contact the instance. For more information about connecting to your instance, see [Connecting to Your Windows Instance \(p. 254\)](#).

When you are finished with an instance, be sure to terminate it. For more information, see [Terminate Your Instance \(p. 264\)](#).

Launching an Instance

Before you launch your instance, be sure that you are set up. For more information, see [Setting Up with Amazon EC2 \(p. 13\)](#).

Your AWS account might support both the EC2-Classic and EC2-VPC platforms, depending on when you created your account and which regions you've used. To find out which platform your account

supports, see [Supported Platforms \(p. 672\)](#). If your account supports EC2-Classic, you can launch an instance into either platform. If your account supports EC2-VPC only, you can launch an instance into a VPC only.

Important

When you launch an instance that's not within the [AWS Free Tier](#), you are charged for the time that the instance is running, even if it remains idle.

Launching Your Instance from an AMI

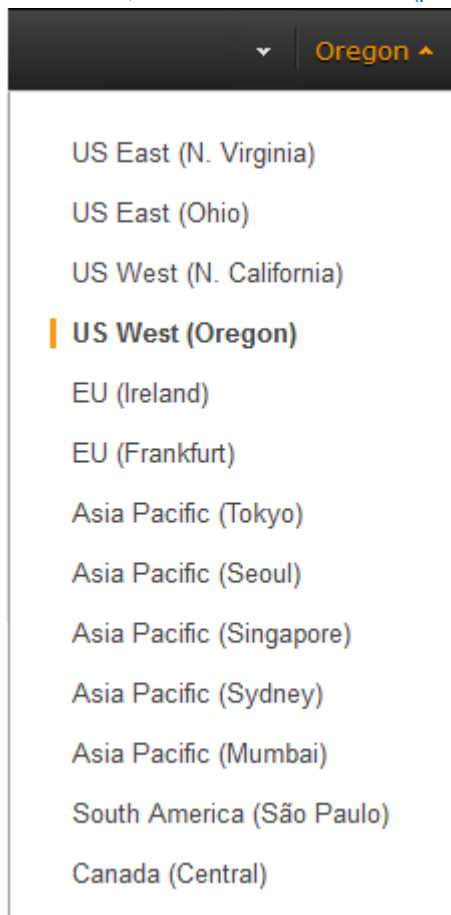
When you launch an instance, you must select a configuration, known as an Amazon Machine Image (AMI). An AMI contains the information required to create a new instance. For example, an AMI might contain the software required to act as a web server: for example, Windows, Apache, and your web site.

Tip

To ensure faster instance launches, break up large requests into smaller batches. For example, create five separate launch requests for 100 instances each instead of one launch request for 500 instances.

To launch an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation bar at the top of the screen, the current region is displayed. Select the region for the instance. This choice is important because some Amazon EC2 resources can be shared between regions, while others can't. Select the region that meets your needs. For more information, see [Resource Locations \(p. 851\)](#).



3. From the Amazon EC2 console dashboard, choose **Launch Instance**.
4. On the **Choose an Amazon Machine Image (AMI)** page, choose an AMI as follows:
 - a. Select the type of AMI to use in the left pane:
 - Quick Start**
A selection of popular AMIs to help you get started quickly. To ensure that you select an AMI that is eligible for the free tier, choose **Free tier only** in the left pane. (Notice that these AMIs are marked **Free tier eligible**.)
 - My AMIs**
The private AMIs that you own, or private AMIs that have been shared with you.
 - AWS Marketplace**
An online store where you can buy software that runs on AWS, including AMIs. For more information about launching an instance from the AWS Marketplace, see [Launching an AWS Marketplace Instance \(p. 251\)](#).
 - Community AMIs**
The AMIs that AWS community member have made available for others to use. To filter the list of AMIs by operating system, choose the appropriate check box under **Operating system**. You can also filter by architecture and root device type.
 - b. Check the **Root device type** listed for each AMI. Notice which AMIs are the type that you need, either `ebs` (backed by Amazon EBS) or `instance-store` (backed by instance store). For more information, see [Storage for the Root Device \(p. 64\)](#).
 - c. Check the **Virtualization type** listed for each AMI. Notice which AMIs are the type that you need, either `hvm` or `paravirtual`. For example, some instance types require HVM.
 - d. Choose an AMI that meets your needs, and then choose **Select**.
5. On the **Choose an Instance Type** page, select the hardware configuration and size of the instance to launch. Larger instance types have more CPU and memory. For more information, see [Instance Types \(p. 117\)](#).

To remain eligible for the free tier, choose the **t2.micro** instance type. For more information, see [T2 Instances \(p. 121\)](#).

By default, the wizard displays current generation instance types, and selects the first available instance type based on the AMI that you selected. To view previous generation instance types, choose **All generations** from the filter list.

Note

If you are new to AWS and would like to set up an instance quickly for testing purposes, you can choose **Review and Launch** at this point to accept default configuration settings, and launch your instance. Otherwise, to configure your instance further, choose **Next: Configure Instance Details**.

6. On the **Configure Instance Details** page, change the following settings as necessary (expand **Advanced Details** to see all the settings), and then choose **Next: Add Storage**:
 - **Number of instances:** Enter the number of instances to launch.
 - Note**
To help ensure that you maintain the correct number of instances to handle your application, you can choose **Launch into Auto Scaling Group** to create a launch configuration and an Auto Scaling group. Auto Scaling scales the number of instances in the group according to your specifications. For more information, see the [Auto Scaling User Guide](#).
 - **Purchasing option:** Select **Request Spot instances** to launch a Spot instance. For more information, see [Spot Instances \(p. 180\)](#).
 - Your account may support the EC2-Classic and EC2-VPC platforms, or EC2-VPC only. To find out which platform your account supports, see [Supported Platforms \(p. 672\)](#). If your account

supports EC2-VPC only, you can launch your instance into your default VPC or a nondefault VPC. Otherwise, you can launch your instance into EC2-Classic or a nondefault VPC.

Note

Some instance types must be launched into a VPC. If you don't have a VPC, you can let the wizard create one for you.

To launch into EC2-Classic:

- **Network:** Select **Launch into EC2-Classic**.
- **Availability Zone:** Select the Availability Zone to use. To let AWS choose an Availability Zone for you, select **No preference**.

To launch into a VPC:

- **Network:** Select the VPC, or to create a new VPC, choose **Create new VPC** to go the Amazon VPC console. When you have finished, return to the wizard and choose **Refresh** to load your VPC in the list.
- **Subnet:** Select the subnet into which to launch your instance. If your account is EC2-VPC only, select **No preference** to let AWS choose a default subnet in any Availability Zone. To create a new subnet, choose **Create new subnet** to go to the Amazon VPC console. When you are done, return to the wizard and choose **Refresh** to load your subnet in the list.
- **Auto-assign Public IP:** Specify whether your instance receives a public IPv4 address. By default, instances in a default subnet receive a public IPv4 address and instances in a nondefault subnet do not. You can select **Enable** or **Disable** to override the subnet's default setting. For more information, see [Public IPv4 Addresses and External DNS Hostnames \(p. 695\)](#).
- **Auto-assign IPv6 IP:** Specify whether your instance receives an IPv6 address from the range of the subnet. Select **Enable** or **Disable** to override the subnet's default setting. This option is only available if you've associated an IPv6 CIDR block with your VPC and subnet. For more information, see [Your VPC and Subnets](#) in the *Amazon VPC User Guide*.
- **Domain join directory:** Select the AWS Directory Service directory (domain) to which your Windows instance is joined. The directory must be in the same VPC that you selected for your instance. If you select a domain, you must select an IAM role. For more information, see [Joining a Windows Instance to an AWS Directory Service Domain \(p. 331\)](#).
- **IAM role:** Select an AWS Identity and Access Management (IAM) role to associate with the instance. For more information, see [IAM Roles for Amazon EC2 \(p. 658\)](#).
- **Shutdown behavior:** Select whether the instance should stop or terminate when shut down. For more information, see [Changing the Instance Initiated Shutdown Behavior \(p. 267\)](#).
- **Enable termination protection:** Select this check box to prevent accidental termination. For more information, see [Enabling Termination Protection for an Instance \(p. 266\)](#).
- **Monitoring:** Select this check box to enable detailed monitoring of your instance using Amazon CloudWatch. Additional charges apply. For more information, see [Monitoring Your Instances Using CloudWatch \(p. 575\)](#).
- **EBS-Optimized instance:** An Amazon EBS-optimized instance uses an optimized configuration stack and provides additional, dedicated capacity for Amazon EBS I/O. If the instance type supports this feature, select this check box to enable it. Additional charges apply. For more information, see [Amazon EBS-Optimized Instances \(p. 795\)](#).
- **Tenancy:** If you are launching your instance into a VPC, you can choose to run your instance on isolated, dedicated hardware (**Dedicated**) or on a Dedicated host (**Dedicated host**). Additional charges may apply. For more information, see [Dedicated Instances \(p. 236\)](#) and [Dedicated Hosts \(p. 226\)](#).
- **Network interfaces:** If you selected a specific subnet, you can specify up to two network interfaces for your instance:
 - For **Network Interface**, select **New network interface** to let AWS create a new interface, or select an existing, available network interface.

- For **Primary IP**, enter a private IPv4 address from the range of your subnet, or leave **Auto-assign** to let AWS choose a private IPv4 address for you.
- For **Secondary IP addresses**, choose **Add IP** to assign more than one private IPv4 address to the selected network interface.
- (IPv6-only) For **IPv6 IPs**, choose **Add IP**, and enter an IPv6 address from the range of the subnet, or leave **Auto-assign** to let AWS choose one for you.
- Choose **Add Device** to add a secondary network interface. A secondary network interface can reside in a different subnet of the VPC, provided it's in the same Availability Zone as your instance.

For more information, see [Elastic Network Interfaces \(p. 716\)](#). If you specify more than one network interface, your instance cannot receive a public IPv4 address. Additionally, you cannot override the subnet's public IPv4 setting using **Auto-assign Public IP** if you specify an existing network interface for eth0. For more information, see [Assigning a Public IPv4 Address During Instance Launch \(p. 700\)](#).

- **Kernel ID:** (Only valid for paravirtual (PV) AMIs) Select **Use default** unless you want to use a specific kernel.
 - **RAM disk ID:** (Only valid for paravirtual (PV) AMIs) Select **Use default** unless you want to use a specific RAM disk. If you have selected a kernel, you may need to select a specific RAM disk with the drivers to support it.
 - **Placement group:** A placement group is a logical grouping for your cluster instances. Select an existing placement group, or create a new one. This option is only available if you've selected an instance type that supports placement groups. For more information, see [Placement Groups \(p. 731\)](#).
 - **User data:** You can specify user data to configure an instance during launch, or to run a configuration script. To attach a file, select the **As file** option and browse for the file to attach.
7. On the **Add Storage** page, you can specify volumes to attach to the instance besides the volumes specified by the AMI (such as the root device volume). You can change the following options, then choose **Next: Tag Instance** when you have finished:
- **Type:** Select instance store or Amazon EBS volumes to associate with your instance. The type of volume available in the list depends on the instance type you've chosen. For more information, see [Amazon EC2 Instance Store \(p. 822\)](#) and [Amazon EBS Volumes \(p. 747\)](#).
 - **Device:** Select from the list of available device names for the volume.
 - **Snapshot:** Enter the name or ID of the snapshot from which to restore a volume. You can also search for public snapshots by typing text into the **Snapshot** field. Snapshot descriptions are case-sensitive.
 - **Size:** For Amazon EBS-backed volumes, you can specify a storage size. Note that even if you have selected an AMI and instance that are eligible for the free tier, you need to keep under 30 GiB of total storage to stay within the free tier.

Note

The following Amazon EBS volume considerations apply to Windows boot volumes:

- Windows 2003 instances do not boot if the boot volume is 2 TiB (2048 GiB) or greater.
 - Windows boot volumes must use an MBR partition table, which limits the usable space to 2 TiB, regardless of volume size.
 - Windows boot volumes of 2 TiB (2048 GiB) that have been converted to use a dynamic MBR partition table display an error when examined with Disk Manager.
- The following Amazon EBS volume considerations apply to Windows data (non-boot) volumes:
- Windows volumes 2 TiB (2048 GiB) or greater must use a GPT partition table to access the entire volume.
 - Amazon EBS volumes over 2048 GiB that are attached to Windows instances at launch are automatically formatted with a GPT partition table.

- Amazon EBS volumes attached to Windows instances after launch must be manually initialized with a GPT partition table. For more information, see [Making an Amazon EBS Volume Available for Use](#).

Note

If you increase the size of your root volume at this point (or any other volume created from a snapshot), you need to extend the file system on that volume in order to use the extra space. For more information about extending your file system after your instance has launched, see [Expanding the Storage Space of an EBS Volume on Windows \(p. 783\)](#).

- **Volume Type:** For Amazon EBS volumes, select either a General Purpose SSD, Provisioned IOPS SSD, or Magnetic volume. For more information, see [Amazon EBS Volume Types \(p. 749\)](#).

Note

If you select a Magnetic boot volume, you'll be prompted when you complete the wizard to make General Purpose SSD volumes the default boot volume for this instance and future console launches. (This preference persists in the browser session, and does not affect AMIs with Provisioned IOPS SSD boot volumes.) We recommend that you make General Purpose SSD volumes the default because they provide a much faster boot experience and they are the optimal volume type for most workloads. For more information, see [Amazon EBS Volume Types \(p. 749\)](#).

Note

Some AWS accounts created before 2012 might have access to Availability Zones in us-east-1, us-west-1, or ap-northeast-1 that do not support Provisioned IOPS SSD (io1) volumes. If you are unable to create an io1 volume (or launch an instance with an io1 volume in its block device mapping) in one of these regions, try a different Availability Zone in the region. You can verify that an Availability Zone supports io1 volumes by creating a 4 GiB io1 volume in that zone.

- **IOPS:** If you have selected a Provisioned IOPS SSD volume type, then you can enter the number of I/O operations per second (IOPS) that the volume can support.
 - **Delete on Termination:** For Amazon EBS volumes, select this check box to delete the volume when the instance is terminated. For more information, see [Preserving Amazon EBS Volumes on Instance Termination \(p. 268\)](#).
 - **Encrypted:** Select this check box to encrypt new Amazon EBS volumes. Amazon EBS volumes that are restored from encrypted snapshots are automatically encrypted. Encrypted volumes may only be attached to [supported instance types \(p. 800\)](#).
8. On the **Tag Instance** page, specify [tags \(p. 859\)](#) for the instance by providing key and value combinations. Choose **Create Tag** to add more than one tag to your resource. Choose **Next: Configure Security Group** when you are done.
 9. On the **Configure Security Group** page, use a security group to define firewall rules for your instance. These rules specify which incoming network traffic is delivered to your instance. All other traffic is ignored. (For more information about security groups, see [Amazon EC2 Security Groups for Windows Instances \(p. 606\)](#).) Select or create a security group as follows, and then choose **Review and Launch**.

To select an existing security group:

1. Choose **Select an existing security group**. Your security groups are displayed. (If you are launching into EC2-Classic, these are security groups for EC2-Classic. If you are launching into a VPC, these are security group for that VPC.)
2. Select a security group from the list.
3. (Optional) You can't edit the rules of an existing security group, but you can copy them to a new group by choosing **Copy to new**. Then you can add rules as described in the next procedure.

To create a new security group:

1. Choose **Create a new security group**. The wizard automatically defines the launch-wizard-x security group.
2. (Optional) You can edit the name and description of the security group.
3. The wizard automatically defines an inbound rule to allow to you connect to your instance over SSH (port 22) for Linux or RDP (port 3389) for Windows.

Caution

This rule enables all IP addresses (0.0.0.0/0) to access your instance over the specified port. This is acceptable for this short exercise, but it's unsafe for production environments. You should authorize only a specific IP address or range of addresses to access your instance.

4. You can add rules to suit your needs. For example, if your instance is a web server, open ports 80 (HTTP) and 443 (HTTPS) to allow Internet traffic.

To add a rule, choose **Add Rule**, select the protocol to open to network traffic, and then specify the source. Choose **My IP** from the **Source** list to let the wizard add your computer's public IP address. However, if you are connecting through an ISP or from behind your firewall without a static IP address, you need to find out the range of IP addresses used by client computers.

10. On the **Review Instance Launch** page, check the details of your instance, and make any necessary changes by choosing the appropriate **Edit** link.

When you are ready, choose **Launch**.

11. In the **Select an existing key pair or create a new key pair** dialog box, you can choose an existing key pair, or create a new one. For example, choose **Choose an existing key pair**, then select the key pair you created when getting set up.

To launch your instance, select the acknowledgment check box, then choose **Launch Instances**.

Important

If you choose the **Proceed without key pair** option, you won't be able to connect to the instance unless you choose an AMI that is configured to allow users another way to log in.

12. (Optional) You can create a status check alarm for the instance (additional fees may apply). (If you're not sure, you can always add one later.) On the confirmation screen, choose **Create status check alarms** and follow the directions. For more information, see [Creating and Editing Status Check Alarms \(p. 570\)](#).
13. If the instance state immediately goes to `terminated` instead of `running`, you can get information about why the instance didn't launch. For more information, see [Instance terminates immediately \(p. 951\)](#).

Launching an Instance Using an Existing Instance as a Template

The Amazon EC2 console provides a **Launch More Like This** wizard option that enables you to use a current instance as a template for launching other instances. This option automatically populates the Amazon EC2 launch wizard with certain configuration details from the selected instance.

Note

The **Launch More Like This** wizard option does not clone your selected instance; it only replicates some configuration details. To create a copy of your instance, first create an AMI from it, then launch more instances from the AMI.

The following configuration details are copied from the selected instance into the launch wizard:

- AMI ID
- Instance type
- Availability Zone, or the VPC and subnet in which the selected instance is located
- Public IPv4 address. If the selected instance currently has a public IPv4 address, the new instance receives a public IPv4 address - regardless of the selected instance's default public IPv4 address setting. For more information about public IPv4 addresses, see [Public IPv4 Addresses and External DNS Hostnames \(p. 695\)](#).
- Placement group, if applicable
- IAM role associated with the instance, if applicable
- Shutdown behavior setting (stop or terminate)
- Termination protection setting (true or false)
- CloudWatch monitoring (enabled or disabled)
- Amazon EBS-optimization setting (true or false)
- Tenancy setting, if launching into a VPC (shared or dedicated)
- Kernel ID and RAM disk ID, if applicable
- User data, if specified
- Tags associated with the instance, if applicable
- Security groups associated with the instance
- Association information. If the selected instance is associated with a configuration file, the same file is automatically associated with new instance. If the configuration file includes a domain join configuration, the new instance will be joined to the same domain. For more information about joining a domain, see [Joining a Windows Instance to an AWS Directory Service Domain \(p. 331\)](#).

The following configuration details are not copied from your selected instance; instead, the wizard applies their default settings or behavior:

- (VPC only) Number of network interfaces: The default is one network interface, which is the primary network interface (eth0).
- Storage: The default storage configuration is determined by the AMI and the instance type.

To use your current instance as a template

1. On the Instances page, select the instance you want to use.
2. Choose **Actions**, and then **Launch More Like This**.
3. The launch wizard opens on the **Review Instance Launch** page. You can check the details of your instance, and make any necessary changes by clicking the appropriate **Edit** link.

When you are ready, choose **Launch** to select a key pair and launch your instance.

Launching an AWS Marketplace Instance

You can subscribe to an AWS Marketplace product and launch an instance from the product's AMI using the Amazon EC2 launch wizard. For more information about paid AMIs, see [Paid AMIs \(p. 74\)](#). To cancel your subscription after launch, you first have to terminate all instances running from it. For more information, see [Managing Your AWS Marketplace Subscriptions \(p. 77\)](#).

To launch an instance from the AWS Marketplace using the launch wizard

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

- From the Amazon EC2 dashboard, choose **Launch Instance**.
- On the **Choose an Amazon Machine Image (AMI)** page, choose the **AWS Marketplace** category on the left. Find a suitable AMI by browsing the categories, or using the search functionality. Choose **Select** to choose your product.
- A dialog displays an overview of the product you've selected. You can view the pricing information, as well as any other information that the vendor has provided. When you're ready, choose **Continue**.

Note

You are not charged for using the product until you have launched an instance with the AMI. Take note of the pricing for each supported instance type, as you will be prompted to select an instance type on the next page of the wizard. Additional taxes may also apply to the product.

- On the **Choose an Instance Type** page, select the hardware configuration and size of the instance to launch. When you're done, choose **Next: Configure Instance Details**.
- On the next pages of the wizard, you can configure your instance, add storage, and add tags. For more information about the different options you can configure, see [Launching an Instance \(p. 244\)](#). Choose **Next** until you reach the **Configure Security Group** page.

The wizard creates a new security group according to the vendor's specifications for the product. The security group may include rules that allow all IPv4 addresses (0.0.0.0/0) access on SSH (port 22) on Linux or RDP (port 3389) on Windows. We recommend that you adjust these rules to allow only a specific address or range of addresses to access your instance over those ports.

When you are ready, choose **Review and Launch**.

- On the **Review Instance Launch** page, check the details of the AMI from which you're about to launch the instance, as well as the other configuration details you set up in the wizard. When you're ready, choose **Launch** to select or create a key pair, and launch your instance.
- Depending on the product you've subscribed to, the instance may take a few minutes or more to launch. You are first subscribed to the product before your instance can launch. If there are any problems with your credit card details, you will be asked to update your account details. When the launch confirmation page displays, choose **View Instances** to go to the Instances page.

Note

You are charged the subscription price as long as your instance is running, even if it is idle. If your instance is stopped, you may still be charged for storage.

- When your instance is in the **running** state, you can connect to it. To do this, select your instance in the list and choose **Connect**. Follow the instructions in the dialog. For more information about connecting to your instance, see [Connecting to Your Windows Instance \(p. 254\)](#).

Important

Check the vendor's usage instructions carefully, as you may need to use a specific user name to log in to the instance. For more information about accessing your subscription details, see [Managing Your AWS Marketplace Subscriptions \(p. 77\)](#).

Launching an AWS Marketplace AMI Instance Using the API and CLI

To launch instances from AWS Marketplace products using the API or command line tools, first ensure that you are subscribed to the product. You can then launch an instance with the product's AMI ID using the following methods:

Method	Documentation
AWS CLI	Use the run-instances command, or see the following topic for more information: Launching an Instance .

Amazon Elastic Compute Cloud
User Guide for Windows Instances
Launch

Method	Documentation
AWS Tools for Windows PowerShell	Use the New-EC2Instance command, or see the following topic for more information: Launch an Amazon EC2 Instance Using Windows PowerShell
Query API	Use the RunInstances request.

Connecting to Your Windows Instance

Amazon EC2 instances created from most Windows Amazon Machine Images (AMIs) enable you to connect using Remote Desktop. Remote Desktop uses the Remote Desktop Protocol (RDP) and enables you to connect to and use your instance in the same way you use a computer sitting in front of you. This topic describes how to connect using Remote Desktop Connection, which is available on most editions of Windows.

Important

The Windows Server 2016 Nano installation option (Nano Server) does *not* include an RDP option for connecting. You must use Windows PowerShell. For more information, see [Connect to a Windows Server 2016 Nano Server Instance \(p. 257\)](#).

For information about connecting to a Linux instance, see [Connect to Your Linux Instance](#) in the *Amazon EC2 User Guide for Linux Instances*. If you receive an error while attempting to connect to your instance, see [Remote Desktop can't connect to the remote computer \(p. 951\)](#).

Contents

- [Prerequisites \(p. 254\)](#)
- [Connect to Your Windows Instance \(p. 255\)](#)
- [Connect to a Windows Instance Using Its IPv6 Address \(p. 256\)](#)
- [Connect to a Windows Server 2016 Nano Server Instance \(p. 257\)](#)
- [Transfer Files to Windows Server Instances \(p. 258\)](#)

Prerequisites

• Install an RDP client

Your Windows computer includes an RDP client by default. You can check for an RDP client by typing `mstsc` at a Command Prompt window. If your computer doesn't recognize this command, see the [Windows home page](#) and search for the download for Remote Desktop Connection. For Mac OS X, you can use the Microsoft Remote Desktop app from the Apple App Store, or the [Microsoft's Remote Desktop Connection Client](#) from the Microsoft website. For Linux, you can use `rdesktop`.

Important

Mac OS X users: If you are connecting to a Windows 2012 R2 instance, the Remote Desktop Connection client from the Microsoft website may not work. Use the Microsoft Remote Desktop app from the Apple App Store instead.

• Get the ID of the instance

You can get the ID of your instance using the Amazon EC2 console (from the **Instance ID** column). If you prefer, you can use the [describe-instances](#) (AWS CLI) or [Get-EC2Instance](#) (AWS Tools for Windows PowerShell) command.

• Get the public DNS name of the instance

You can get the public DNS for your instance using the Amazon EC2 console (check the **Public DNS (IPv4)** column; if this column is hidden, choose the **Show/Hide** icon and select **Public DNS (IPv4)**). If you prefer, you can use the [describe-instances](#) (AWS CLI) or [Get-EC2Instance](#) (AWS Tools for Windows PowerShell) command.

• (IPv6 only) Get the IPv6 address of the instance

If you've assigned an IPv6 address to your instance, you can optionally connect to the instance using its IPv6 address instead of a public IPv4 address or public IPv4 DNS hostname. Your local computer must have an IPv6 address and must be configured to use IPv6. You can get the IPv6 address of your instance using the Amazon EC2 console (check the **IPv6 IPs** field). If you prefer, you can use the [describe-instances](#) (AWS CLI) or [Get-EC2Instance](#) (AWS Tools for Windows PowerShell) command. For more information about IPv6, see [IPv6 Addresses \(p. 696\)](#).

• Locate the private key

You'll need the fully qualified path of the `.pem` file for the key pair that you specified when you launched the instance.

- **Enable inbound RDP traffic from your IP address to your instance**
Ensure that the security group associated with your instance allows incoming RDP traffic from your IP address. For more information, see [Authorizing Inbound Traffic for Your Windows Instances \(p. 663\)](#).

Important

Your default security group does not allow incoming RDP traffic by default.

- For the best experience using Internet Explorer, run the latest version.

Connect to Your Windows Instance

To connect to a Windows instance, you must retrieve the initial administrator password and then specify this password when you connect to your instance using Remote Desktop.

Note

If you've joined your instance to a domain, you can connect to your instance using domain credentials you've defined in AWS Directory Service. For more information about connecting to an instance in a domain, see [Connecting To Your Instance Using Domain Credentials \(p. 338\)](#).

The name of the administrator account depends on the language of the operating system. For example, for English, it's Administrator, for French it's Administrateur, and for Portuguese it's Administrador. For more information, see [Localized Names for Administrator Account in Windows](#) in the Microsoft TechNet Wiki.

The license for the Windows Server operating system (OS) allows two simultaneous remote connections for administrative purposes. The license for Windows Server is included in the price of your EC2 instance. If you need more than two simultaneous remote connections you must purchase a Remote Desktop Services (RDS) license. If you attempt a third connection, an error occurs. For more information, see [Configure the Number of Simultaneous Remote Connections Allowed for a Connection](#).

To connect to your Windows instance using an RDP client

1. In the Amazon EC2 console, select the instance, and then choose **Connect**.
2. In the **Connect To Your Instance** dialog box, choose **Get Password** (it will take a few minutes after the instance is launched before the password is available).
3. Choose **Browse** and navigate to the private key file you created when you launched the instance. Select the file and choose **Open** to copy the entire contents of the file into the **Contents** field.
4. Choose **Decrypt Password**. The console displays the default administrator password for the instance in the **Connect To Your Instance** dialog box, replacing the link to **Get Password** shown previously with the actual password.
5. Record the default administrator password, or copy it to the clipboard. You need this password to connect to the instance.
6. Choose **Download Remote Desktop File**. Your browser prompts you to either open or save the `.rdp` file. Either option is fine. When you have finished, you can choose **Close** to dismiss the **Connect To Your Instance** dialog box.
 - If you opened the `.rdp` file, you'll see the **Remote Desktop Connection** dialog box.
 - If you saved the `.rdp` file, navigate to your downloads directory, and open the `.rdp` file to display the dialog box.
7. You may get a warning that the publisher of the remote connection is unknown. If you are using **Remote Desktop Connection** from a Windows PC, choose **Connect** to connect to your instance. If you are using **Microsoft Remote Desktop** on a Mac, skip the next step.

- When prompted, log in to the instance, using the administrator account for the operating system and the password that you recorded or copied previously. If your **Remote Desktop Connection** already has an administrator account set up, you might have to choose the **Use another account** option and enter the user name and password manually.

Note

Sometimes copying and pasting content can corrupt data. If you encounter a "Password Failed" error when you log in, try typing in the password manually.

- Due to the nature of self-signed certificates, you may get a warning that the security certificate could not be authenticated. Use the following steps to verify the identity of the remote computer, or simply choose **Yes** or **Continue** to continue if you trust the certificate.
 - If you are using **Remote Desktop Connection** from a Windows PC, choose **View certificate**. If you are using **Microsoft Remote Desktop** on a Mac, choose **Show Certificate**.
 - Choose the **Details** tab, and scroll down to the **Thumbprint** entry on a Windows PC, or the **SHA1 Fingerprints** entry on a Mac. This is the unique identifier for the remote computer's security certificate.
 - In the Amazon EC2 console, select the instance, choose **Actions**, and then choose **Get System Log**.
 - In the system log output, look for an entry labeled `RDPCERTIFICATE-THUMBPRINT`. If this value matches the thumbprint or fingerprint of the certificate, you have verified the identity of the remote computer.
 - If you are using **Remote Desktop Connection** from a Windows PC, return to the **Certificate** dialog box and choose **OK**. If you are using **Microsoft Remote Desktop** on a Mac, return to the **Verify Certificate** and choose **Continue**.
 - If you are using **Remote Desktop Connection** from a Windows PC, choose **Yes** in the **Remote Desktop Connection** window to connect to your instance. If you are using **Microsoft Remote Desktop** on a Mac, log in to the instance as prompted, using the default **Administrator** account and the default administrator password that you recorded or copied previously.

Note

On a Mac, you may need to switch spaces to see the **Microsoft Remote Desktop** login screen. For more information on spaces, see <http://support.apple.com/kb/PH14155>.

After you connect, we recommend that you do the following:

- Change the administrator password from the default value. You change the password while logged on to the instance itself, just as you would on any other Windows Server.
- Create another user account with administrator privileges on the instance. Another account with administrator privileges is a safeguard if you forget the administrator password or have a problem with the administrator account.

Connect to a Windows Instance Using Its IPv6 Address

If you've enabled your VPC for IPv6 and assigned an IPv6 address to your Windows instance, you can use an RDP client to connect to your instance using its IPv6 address instead of a public IPv4 address or public DNS hostname. For more information, see [IPv6 Addresses](#) (p. 696).

To connect to your Windows instance using its IPv6 address

- In the Amazon EC2 console, select the instance, and then choose **Connect**.
- In the **Connect To Your Instance** dialog box, choose **Get Password** (it will take a few minutes after the instance is launched before the password is available).

3. Choose **Browse** and navigate to the private key file you created when you launched the instance. Select the file and choose **Open** to copy the entire contents of the file into the **Contents** field.
 4. Choose **Decrypt Password**.
 5. Copy the default administrator password. You need this password to connect to the instance.
 6. Open the RDP client on your computer.
 7. (Windows) For the RDP client on a Windows computer, choose **Show Options** and do the following:
 - For **Computer**, enter the IPv6 address of your Windows instance, for example, `2001:db8:1234:1a00:9691:9503:25ad:1761`.
 - For **User name**, enter **Administrator**.
 - Choose **Connect**.
- (OS X) For the Microsoft Remote Desktop app, choose **New** and do the following:
- For **PC Name**, enter the IPv6 address of your Windows instance; for example, `2001:db8:1234:1a00:9691:9503:25ad:1761`.
 - For **User name**, enter **Administrator**.
 - Close the dialog box. Under **My Desktops**, select the connection and choose **Start**.
8. Due to the nature of self-signed certificates, you may get a warning that the security certificate could not be authenticated. Use the following steps to verify the identity of the remote computer, or simply choose **Yes** or **Continue** to continue if you trust the certificate.
 9. When prompted, enter the password that you recorded or copied previously.

Connect to a Windows Server 2016 Nano Server Instance

Windows Server 2016 Nano Server is a remotely administered server operating system that is optimized for private clouds and data centers. It is similar to Windows Server in Server Core mode, but it is significantly smaller, has no local logon capability, and only supports 64-bit applications, tools, and agents. It takes up far less disk space, sets up significantly faster, and requires far fewer updates and restarts than Windows Server.

Windows Server 2016 Nano Server does not support Remote Desktop connections. To connect to a Windows Server 2016 Nano Server instance, you must connect using PowerShell, as described in the following procedure.

Connecting to a Nano Server instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Launch a Windows Server 2016 Nano Server instance. For more information about launching an instance from the Amazon EC2 console, see [Launching Your Instance from an AMI \(p. 245\)](#).

Important

You must either edit the security group for the instance and specify a **Custom TCP Rule** over HTTP that uses TCP port 5985, or specify this custom rule in Step 6 of the Amazon EC2 Launch Wizard, as shown below.

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can

Assign a security group: Create a new security group
 Select an existing security group

Security group name:

Description:

Type (i)	Protocol (i)
Custom TCP Rule ▼	TCP

Add Rule

3. After the instance starts, locate the private IP address for the instance and the administrator password. If you use the private IP address, you must connect to the instance from another virtual machine on the same VPC.
4. Execute the following command in Windows PowerShell.

```
$ip = "Your instance IP address"  
Set-Item WSMan:\localhost\Client\TrustedHosts $ip  
$user = "$ip\Administrator"  
Enter-PSSession -ComputerName $ip -Credential $user
```

The following procedure describes how to remotely copy files to Windows Server 2016 Nano Server.

Copying files to Nano Server

1. On the instance you are using to connect to Nano Server, [download](#) and install version 5.0 or later of the Windows Management Framework. The installation requires a restart.
2. Use the `Copy-Item` command to copy files to the Nano Server instance.

```
$ip = "Your instance IP address"  
Set-Item WSMan:\localhost\Client\TrustedHosts $ip  
$user = "$ip\Administrator"  
$cs = New-PSSession -ComputerName $ip -Credential $user  
Copy-Item -Path Path to files -Destination Path to destination on the Nano  
Server instance -ToSession $cs -Recurse
```

Transfer Files to Windows Server Instances

You can work with your Windows instance the same way that you would work with any Windows server. For example, you can transfer files between a Windows instance and your local computer using the local file sharing feature of the Microsoft Remote Desktop Connection software. If you enable this option, you can access your local files from your Windows instances. You can access local files on

hard disk drives, DVD drives, portable media drives, and mapped network drives. For more information about this feature, go to the following articles:

- [How to gain access to local files in a remote desktop session to a Windows XP-based or to a Windows Server 2003-based host computer](#)
- [Make Local Devices and Resources Available in a Remote Session](#)
- [Getting Started with Remote Desktop Client on Mac](#)

Stop and Start Your Instance

You can stop and restart your instance if it has an Amazon EBS volume as its root device. The instance retains its instance ID, but can change as described in the Overview section.

When you stop an instance, we shut it down. We don't charge hourly usage for a stopped instance, or data transfer fees, but we do charge for the storage for any Amazon EBS volumes. Each time you start a stopped instance we charge a full instance hour, even if you make this transition multiple times within a single hour.

While the instance is stopped, you can treat its root volume like any other volume, and modify it (for example, repair file system problems or update software). You just detach the volume from the stopped instance, attach it to a running instance, make your changes, detach it from the running instance, and then reattach it to the stopped instance. Make sure that you reattach it using the storage device name that's specified as the root device in the block device mapping for the instance.

If you decide that you no longer need an instance, you can terminate it. As soon as the state of an instance changes to `shutting-down` or `terminated`, we stop charging for that instance. For more information, see [Terminate Your Instance](#) (p. 264).

Contents

- [Overview](#) (p. 259)
- [Stopping and Starting Your Instances](#) (p. 260)
- [Modifying a Stopped Instance](#) (p. 261)
- [Troubleshooting](#) (p. 262)

Overview

You can only stop an Amazon EBS-backed instance. To verify the root device type of your instance, describe the instance and check whether the device type of its root volume is `ebs` (Amazon EBS-backed instance) or `instance store` (instance store-backed instance). For more information, see [Determining the Root Device Type of Your AMI](#) (p. 65).

When you stop a running instance, the following happens:

- The instance performs a normal shutdown and stops running; its status changes to `stopping` and then `stopped`.
- Any Amazon EBS volumes remain attached to the instance, and their data persists.
- Any data stored in the RAM of the host computer or the instance store volumes of the host computer is gone.
- In most cases, the instance is migrated to a new underlying host computer when it's started.
- EC2-Classic: We release the public and private IPv4 addresses for the instance when you stop the instance, and assign new ones when you restart it.

EC2-VPC: The instance retains its private IPv4 addresses and any IPv6 addresses when stopped and restarted. We release the public IPv4 address and assign a new one when you restart it.

- **EC2-Classic:** We disassociate any Elastic IP address that's associated with the instance. You're charged for Elastic IP addresses that aren't associated with an instance. When you restart the instance, you must associate the Elastic IP address with the instance; we don't do this automatically.

EC2-VPC: The instance retains its associated Elastic IP addresses. You're charged for any Elastic IP addresses associated with a stopped instance.

- When you stop and start a Windows instance, the EC2Config service performs tasks on the instance such as changing the drive letters for any attached Amazon EBS volumes. For more information about these defaults and how you can change them, see [Configuring a Windows Instance Using the EC2Config Service \(p. 283\)](#) in the *Amazon EC2 User Guide for Windows Instances*.
- If you've registered the instance with a load balancer, it's likely that the load balancer won't be able to route traffic to your instance after you've stopped and restarted it. You must de-register the instance from the load balancer after stopping the instance, and then re-register after starting the instance. For more information, see [Register or Deregister EC2 Instances for Your Classic Load Balancer](#) in the *Classic Load Balancer Guide*.
- If your instance is in an Auto Scaling group, the Auto Scaling service marks the stopped instance as unhealthy, and may terminate it and launch a replacement instance. For more information, see [Health Checks for Auto Scaling Instances](#) in the *Auto Scaling User Guide*.
- When you stop a ClassicLink instance, it's unlinked from the VPC to which it was linked. You must link the instance to the VPC again after restarting it. For more information about ClassicLink, see [ClassicLink \(p. 673\)](#).

For more information, see [Differences Between Reboot, Stop, and Terminate \(p. 243\)](#).

You can modify the following attributes of an instance only when it is stopped:

- Instance type
- User data
- Kernel
- RAM disk

If you try to modify these attributes while the instance is running, Amazon EC2 returns the `IncorrectInstanceState` error.

Stopping and Starting Your Instances

You can start and stop your Amazon EBS-backed instance using the console or the command line.

By default, when you initiate a shutdown from an Amazon EBS-backed instance (using the **shutdown**, **halt**, or **poweroff** command), the instance stops. You can change this behavior so that it terminates instead. For more information, see [Changing the Instance Initiated Shutdown Behavior \(p. 267\)](#).

To stop and start an Amazon EBS-backed instance using the console

1. In the navigation pane, choose **Instances**, and select the instance.
2. [EC2-Classic] If the instance has an associated Elastic IP address, write down the Elastic IP address and the instance ID shown in the details pane.
3. Choose **Actions**, select **Instance State**, and then choose **Stop**. If **Stop** is disabled, either the instance is already stopped or its root device is an instance store volume.

Warning

When you stop an instance, the data on any instance store volumes is erased. Therefore, if you have any data on instance store volumes that you want to keep, be sure to back it up to persistent storage.

4. In the confirmation dialog box, choose **Yes, Stop**. It can take a few minutes for the instance to stop.

[EC2-Classic] When the instance state becomes `stopped`, the **Elastic IP**, **Public DNS (IPv4)**, **Private DNS**, and **Private IPs** fields in the details pane are blank to indicate that the old values are no longer associated with the instance.

5. While your instance is stopped, you can modify certain instance attributes. For more information, see [Modifying a Stopped Instance](#) (p. 261).
6. To restart the stopped instance, select the instance, choose **Actions**, select **Instance State**, and then choose **Start**.
7. In the confirmation dialog box, choose **Yes, Start**. It can take a few minutes for the instance to enter the `running` state.

[EC2-Classic] When the instance state becomes `running`, the **Public DNS (IPv4)**, **Private DNS**, and **Private IPs** fields in the details pane contain the new values that we assigned to the instance.

8. [EC2-Classic] If your instance had an associated Elastic IP address, you must reassociate it as follows:
 - a. In the navigation pane, choose **Elastic IPs**.
 - b. Select the Elastic IP address that you wrote down before you stopped the instance.
 - c. Choose **Actions**, and then select **Associate Address**.
 - d. Select the instance ID that you wrote down before you stopped the instance, and then choose **Associate**.

To stop and start an Amazon EBS-backed instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2](#) (p. 3).

- [stop-instances](#) and [start-instances](#) (AWS CLI)
- [Stop-EC2Instance](#) and [Start-EC2Instance](#) (AWS Tools for Windows PowerShell)

Modifying a Stopped Instance

You can change the instance type, user data, and EBS-optimization attributes of a stopped instance using the AWS Management Console or the command line interface. You can't use the AWS Management Console to modify the `DeleteOnTermination`, `kernel`, or `RAM disk` attributes.

To modify an instance attribute

- To change the instance type, see [Resizing Your Instance](#) (p. 147).
- To change the user data for your instance, see [Configuring Instances with User Data](#) (p. 273).
- To enable or disable EBS-optimization for your instance, see [Modifying EBS-Optimization](#) (p. 798).
- To change the `DeleteOnTermination` attribute of the root volume for your instance, see [Updating the Block Device Mapping of a Running Instance](#) (p. 840).

To modify an instance attribute using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2](#) (p. 3).

- [modify-instance-attribute](#) (AWS CLI)

- [Edit-EC2InstanceAttribute](#) (AWS Tools for Windows PowerShell)

Troubleshooting

If you have stopped your Amazon EBS-backed instance and it appears "stuck" in the `stopping` state, you can forcibly stop it. For more information, see [Troubleshooting Stopping Your Instance](#) in the *Amazon EC2 User Guide for Linux Instances*.

Reboot Your Instance

An instance reboot is equivalent to an operating system reboot. In most cases, it takes only a few minutes to reboot your instance. When you reboot an instance, it remains on the same physical host, so your instance keeps its public DNS name (IPv4), private IPv4 address, IPv6 address (if applicable), and any data on its instance store volumes.

Rebooting an instance doesn't start a new instance billing hour, unlike stopping and restarting your instance.

We might schedule your instance for a reboot for necessary maintenance, such as to apply updates that require a reboot. No action is required on your part; we recommend that you wait for the reboot to occur within its scheduled window. For more information, see [Scheduled Events for Your Instances](#) (p. 571).

We recommend that you use Amazon EC2 to reboot your instance instead of running the operating system reboot command from your instance. If you use Amazon EC2 to reboot your instance, we perform a hard reboot if the instance does not cleanly shut down within four minutes. If you use AWS CloudTrail, then using Amazon EC2 to reboot your instance also creates an API record of when your instance was rebooted.

To reboot an instance using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, choose **Instances**.
3. Select the instance, choose **Actions**, select **Instance State**, and then select **Reboot**.
4. Choose **Yes, Reboot** when prompted for confirmation.

To reboot an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2](#) (p. 3).

- [reboot-instances](#) (AWS CLI)
- [Restart-EC2Instance](#) (AWS Tools for Windows PowerShell)

Instance Retirement

An instance is scheduled to be retired when AWS detects irreparable failure of the underlying hardware hosting the instance. When an instance reaches its scheduled retirement date, it is stopped or terminated by AWS. If your instance root device is an Amazon EBS volume, the instance is stopped, and you can start it again at any time. Starting the stopped instance migrates it to new hardware. If your instance root device is an instance store volume, the instance is terminated, and cannot be used again.

Topics

- [Identifying Instances Scheduled for Retirement \(p. 263\)](#)
- [Working with Instances Scheduled for Retirement \(p. 263\)](#)

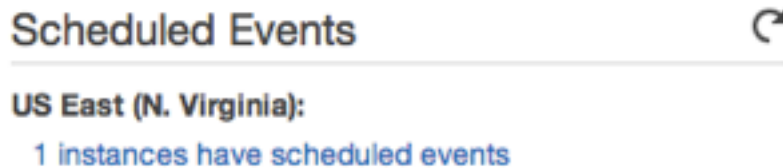
For more information about types of instance events, see [Scheduled Events for Your Instances \(p. 571\)](#).

Identifying Instances Scheduled for Retirement

If your instance is scheduled for retirement, you'll receive an email prior to the event with the instance ID and retirement date. This email is sent to the address that's associated with your account; the same email address that you use to log in to the AWS Management Console. If you use an email account that you do not check regularly, then you can use the Amazon EC2 console or the command line to determine if any of your instances are scheduled for retirement. To update the contact information for your account, go to the [Account Settings](#) page.

To identify instances scheduled for retirement using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, choose **EC2 Dashboard**. Under **Scheduled Events**, you can see the events associated with your Amazon EC2 instances and volumes, organized by region.



3. If you have an instance with a scheduled event listed, select its link below the region name to go to the **Events** page.
4. The **Events** page lists all resources with events associated with them. To view instances that are scheduled for retirement, select **Instance resources** from the first filter list, and then **Instance retirement** from the second filter list.
5. If the filter results show that an instance is scheduled for retirement, select it, and note the date and time in the **Start time** field in the details pane. This is your instance retirement date.

To identify instances scheduled for retirement using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- `describe-instance-status` (AWS CLI)
- `Get-EC2InstanceStatus` (AWS Tools for Windows PowerShell)

Working with Instances Scheduled for Retirement

There are a number of actions available to you when your instance is scheduled for retirement. The action you take depends on whether your instance root device is an Amazon EBS volume, or an instance store volume. If you do not know what your instance root device type is, you can find out using the Amazon EC2 console or the command line.

Determining Your Instance Root Device Type

To determine your instance root device type using the console

1. In the navigation pane, select **Events**. Use the filter lists to identify retiring instances, as demonstrated in the procedure above, [Identifying instances scheduled for retirement \(p. 263\)](#).
2. In the **Resource ID** column, select the instance ID to go to the **Instances** page.
3. Select the instance and locate the **Root device type** field in the **Description** tab. If the value is `ebs`, then your instance is EBS-backed. If the value is `instance-store`, then your instance is instance store-backed.

To determine your instance root device type using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [describe-instances](#) (AWS CLI)
- [Get-EC2Instance](#) (AWS Tools for Windows PowerShell)

Managing Instances Scheduled for Retirement

You can perform one of the actions listed below in order to preserve the data on your retiring instance. It's important that you take this action before the instance retirement date, to prevent unforeseen downtime and data loss.

Warning

If your instance store-backed instance passes its retirement date, it's terminated and you cannot recover the instance or any data that was stored on it. Regardless of the root device of your instance, the data on instance store volumes is lost when the instance is retired, even if they are attached to an EBS-backed instance.

Instance Root Device Type	Action
EBS	Wait for the scheduled retirement date - when the instance is stopped - or stop the instance yourself before the retirement date. You can start the instance again at any time. For more information about stopping and starting your instance, and what to expect when your instance is stopped, such as the effect on public, private and Elastic IP addresses associated with your instance, see Stop and Start Your Instance (p. 259) .
EBS	Create an EBS-backed AMI from your instance, and launch a replacement instance. For more information, see Creating an Amazon EBS-Backed Windows AMI (p. 77) .
Instance store	Bundle your instance, and then create an instance store-backed AMI from the manifest that's created during bundling. You can launch a replacement instance from your new AMI. For more information, see Creating an Instance Store-Backed Windows AMI (p. 80) .

Terminate Your Instance

When you've decided that you no longer need an instance, you can terminate it. As soon as the state of an instance changes to `shutting-down` or `terminated`, you stop incurring charges for that instance.

You can't connect to or restart an instance after you've terminated it. However, you can launch additional instances using the same AMI. If you'd rather stop and restart your instance, see [Stop and Start Your Instance](#) (p. 259). For more information, see [Differences Between Reboot, Stop, and Terminate](#) (p. 243).

Topics

- [Instance Termination](#) (p. 265)
- [Terminating an Instance](#) (p. 265)
- [Enabling Termination Protection for an Instance](#) (p. 266)
- [Changing the Instance Initiated Shutdown Behavior](#) (p. 267)
- [Preserving Amazon EBS Volumes on Instance Termination](#) (p. 268)

Instance Termination

After you terminate an instance, it remains visible in the console for a short while, and then the entry is automatically deleted. You cannot delete the terminated instance entry yourself. After an instance is terminated, resources such as tags and volumes are gradually disassociated from the instance, therefore may no longer be visible on the terminated instance after a short while.

When an instance terminates, the data on any instance store volumes associated with that instance is deleted.

By default, Amazon EBS root device volumes are automatically deleted when the instance terminates. However, by default, any additional EBS volumes that you attach at launch, or any EBS volumes that you attach to an existing instance persist even after the instance terminates. This behavior is controlled by the volume's `DeleteOnTermination` attribute, which you can modify. For more information, see [Preserving Amazon EBS Volumes on Instance Termination](#) (p. 268).

You can prevent an instance from being terminated accidentally by someone using the AWS Management Console, the CLI, and the API. This feature is available for both Amazon EC2 instance store-backed and Amazon EBS-backed instances. Each instance has a `DisableApiTermination` attribute with the default value of `false` (the instance can be terminated through Amazon EC2). You can modify this instance attribute while the instance is running or stopped (in the case of Amazon EBS-backed instances). For more information, see [Enabling Termination Protection for an Instance](#) (p. 266).

You can control whether an instance should stop or terminate when shutdown is initiated from the instance using an operating system command for system shutdown. For more information, see [Changing the Instance Initiated Shutdown Behavior](#) (p. 267).

If you run a script on instance termination, your instance might have an abnormal termination, because we have no way to ensure that shutdown scripts run. Amazon EC2 attempts to shut an instance down cleanly and run any system shutdown scripts; however, certain events (such as hardware failure) may prevent these system shutdown scripts from running.

Terminating an Instance

You can terminate an instance using the AWS Management Console or the command line.

To terminate an instance using the console

1. Before you terminate the instance, verify that you won't lose any data by checking that your Amazon EBS volumes won't be deleted on termination and that you've copied any data that you need from your instance store volumes to Amazon EBS or Amazon S3.
2. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
3. In the navigation pane, select **Instances**.

4. Select the instance, choose **Actions**, select **Instance State**, and then select **Terminate**.
5. Select **Yes, Terminate** when prompted for confirmation.

To terminate an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [terminate-instances](#) (AWS CLI)
- [Stop-EC2Instance](#) (AWS Tools for Windows PowerShell)

Enabling Termination Protection for an Instance

By default, you can terminate your instance using the Amazon EC2 console, command line interface, or API. If you want to prevent your instance from being accidentally terminated using Amazon EC2, you can enable *termination protection* for the instance. The `DisableApiTermination` attribute controls whether the instance can be terminated using the console, CLI, or API. By default, termination protection is disabled for your instance. You can set the value of this attribute when you launch the instance, while the instance is running, or while the instance is stopped (for Amazon EBS-backed instances).

The `DisableApiTermination` attribute does not prevent you from terminating an instance by initiating shutdown from the instance (using an operating system command for system shutdown) when the `InstanceInitiatedShutdownBehavior` attribute is set. For more information, see [Changing the Instance Initiated Shutdown Behavior \(p. 267\)](#).

You can't enable termination protection for Spot instances — a Spot instance is terminated when the Spot price exceeds your bid price. However, you can prepare your application to handle Spot instance interruptions. For more information, see [Spot Instance Interruptions \(p. 221\)](#).

The `DisableApiTermination` attribute does not prevent Auto Scaling from terminating an instance. For instances in an Auto Scaling group, use the following Auto Scaling features instead of Amazon EC2 termination protection:

- To prevent instances that are part of an Auto Scaling group from terminating on scale in, use instance protection. For more information, see [Instance Protection](#) in the *Auto Scaling User Guide*.
- To prevent Auto Scaling from terminating unhealthy instances, suspend the `ReplaceUnhealthy` process. For more information, see [Suspending and Resuming Auto Scaling Processes](#) in the *Auto Scaling User Guide*.
- To specify which instances Auto Scaling should terminate first, choose a termination policy. For more information, see [Customizing the Termination Policy](#) in the *Auto Scaling User Guide*.

You can enable or disable termination protection using the AWS Management Console or the command line.

To enable termination protection for an instance at launch time

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the dashboard, choose **Launch Instance** and follow the directions in the wizard.
3. On the **Configure Instance Details** page, select the **Enable termination protection** check box.

To enable termination protection for a running or stopped instance

1. Select the instance, choose **Actions**, **Instance Settings**, and then choose **Change Termination Protection**.

2. Select **Yes, Enable**.

To disable termination protection for a running or stopped instance

1. Select the instance, select **Actions**, select **Instance Settings**, and then choose **Change Termination Protection**.
2. Select **Yes, Disable**.

To enable or disable termination protection using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- `modify-instance-attribute` (AWS CLI)
- `Edit-EC2InstanceAttribute` (AWS Tools for Windows PowerShell)

Changing the Instance Initiated Shutdown Behavior

By default, when you initiate a shutdown from an Amazon EBS-backed instance (using a command such as **shutdown**, **halt**, or **poweroff**), the instance stops. You can change this behavior using the `InstanceInitiatedShutdownBehavior` attribute for the instance so that it terminates instead. You can update this attribute while the instance is running or stopped.

You can update the `InstanceInitiatedShutdownBehavior` attribute using the Amazon EC2 console or the command line. The `InstanceInitiatedShutdownBehavior` attribute only applies when you perform a shutdown from the operating system of the instance itself; it does not apply when you stop an instance using the `StopInstances` API or the Amazon EC2 console.

To change the shutdown behavior of an instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance, select **Actions**, **Instance Settings**, and then choose **Change Shutdown Behavior**. The current behavior is already selected.
4. To change the behavior, select an option from the **Shutdown behavior** list, and then select **Apply**.



To change the shutdown behavior of an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [modify-instance-attribute](#) (AWS CLI)
- [Edit-EC2InstanceAttribute](#) (AWS Tools for Windows PowerShell)

Preserving Amazon EBS Volumes on Instance Termination

When an instance terminates, Amazon EC2 uses the value of the `DeleteOnTermination` attribute for each attached Amazon EBS volume to determine whether to preserve or delete the volume.

By default, the `DeleteOnTermination` attribute for the root volume of an instance is set to `true`. Therefore, the default is to delete the root volume of an instance when the instance terminates.

By default, when you attach an EBS volume to an instance, its `DeleteOnTermination` attribute is set to `false`. Therefore, the default is to preserve these volumes. After the instance terminates, you can take a snapshot of the preserved volume or attach it to another instance.

To verify the value of the `DeleteOnTermination` attribute for an EBS volume that is in-use, look at the instance's block device mapping. For more information, see [Viewing the EBS Volumes in an Instance Block Device Mapping \(p. 841\)](#).

You can change value of the `DeleteOnTermination` attribute for a volume when you launch the instance or while the instance is running.

Examples

- [Changing the Root Volume to Persist at Launch Using the Console \(p. 268\)](#)
- [Changing the Root Volume to Persist at Launch Using the Command Line \(p. 268\)](#)
- [Changing the Root Volume of a Running Instance to Persist Using the Command Line \(p. 269\)](#)

Changing the Root Volume to Persist at Launch Using the Console

Using the console, you can change the `DeleteOnTermination` attribute when you launch an instance. To change this attribute for a running instance, you must use the command line.

To change the root volume of an instance to persist at launch using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the console dashboard, select **Launch Instance**.
3. On the **Choose an Amazon Machine Image (AMI)** page, choose an AMI and choose **Select**.
4. Follow the wizard to complete the **Choose an Instance Type** and **Configure Instance Details** pages.
5. On the **Add Storage** page, deselect the **Delete On Termination** check box for the root volume.
6. Complete the remaining wizard pages, and then choose **Launch**.

You can verify the setting by viewing details for the root device volume on the instance's details pane. Next to **Block devices**, click the entry for the root device volume. By default, **Delete on termination** is `True`. If you change the default behavior, **Delete on termination** is `False`.

Changing the Root Volume to Persist at Launch Using the Command Line

When you launch an instance, you can use one of the following commands to change the root device volume to persist. The root device is typically `/dev/sda1`. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [run-instances](#) (AWS CLI)
- [New-EC2Instance](#) (AWS Tools for Windows PowerShell)

For example, add the following option to your `run-instances` command:

```
--block-device-mappings file://mapping.json
```

Specify the following in `mapping.json`:

```
[
  {
    "DeviceName": "/dev/sda1",
    "Ebs": {
      "DeleteOnTermination": false,
      "SnapshotId": "snap-1234567890abcdef0",
      "VolumeType": "gp2"
    }
  }
]
```

Changing the Root Volume of a Running Instance to Persist Using the Command Line

You can use one of the following commands to change the root device volume of a running instance to persist. The root device is typically `/dev/sda1`. For more information about these command line interfaces, see [Accessing Amazon EC2](#) (p. 3).

- [modify-instance-attribute](#) (AWS CLI)
- [Edit-EC2InstanceAttribute](#) (AWS Tools for Windows PowerShell)

For example, use the following command:

```
C:\> aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --
block-device-mappings file://mapping.json
```

Specify the following in `mapping.json`:

```
[
  {
    "DeviceName": "/dev/sda1",
    "Ebs": {
      "DeleteOnTermination": false
    }
  }
]
```

Recover Your Instance

You can create an Amazon CloudWatch alarm that monitors an Amazon EC2 instance and automatically recovers the instance if it becomes impaired due to an underlying hardware failure or a problem that requires AWS involvement to repair. Terminated instances cannot be recovered. A recovered instance is identical to the original instance, including the instance ID, private IP

addresses, Elastic IP addresses, and all instance metadata. For more information about using Amazon CloudWatch alarms to recover an instance, see [Create Alarms That Stop, Terminate, Reboot, or Recover an Instance \(p. 592\)](#). To troubleshoot issues with instance recovery failures, see [Troubleshooting Instance Recovery Failures](#) in the *Amazon EC2 User Guide for Linux Instances*.

When the `StatusCheckFailed_System` alarm is triggered, and the recover action is initiated, you will be notified by the Amazon SNS topic that you selected when you created the alarm and associated the recover action. During instance recovery, the instance is migrated during an instance reboot, and any data that is in-memory is lost. When the process is complete, information is published to the SNS topic you've configured for the alarm. Anyone who is subscribed to this SNS topic will receive an email notification that includes the status of the recovery attempt and any further instructions. You will notice an instance reboot on the recovered instance.

Examples of problems that cause system status checks to fail include:

- Loss of network connectivity
- Loss of system power
- Software issues on the physical host
- Hardware issues on the physical host

The recover action can also be triggered when an instance is scheduled by AWS to stop or retire due to degradation of the underlying hardware. For more information about scheduled events, see [Scheduled Events for Your Instances \(p. 571\)](#).

The recover action is supported only on instances with the following characteristics:

- Use a C3, C4, M3, M4, R3, R4, T2, or X1 instance type
- Run in a VPC (not EC2-Classic)
- Use shared tenancy (the `tenancy` attribute is set to `default`)
- Use EBS volumes, including encrypted EBS volumes (not instance store volumes)

If your instance has a public IPv4 address, it retains the public IPv4 address after recovery.

Configuring Your Windows Instance

A Windows instance is a virtual server running Windows Server in the cloud.

After you have successfully launched and logged into your instance, you can make changes to it so that it's configured to meet the needs of a specific application. The following are some common tasks to help you get started.

Contents

- [Instance Metadata and User Data \(p. 271\)](#)
- [Configuring a Windows Instance Using the EC2Config Service \(p. 283\)](#)
- [Configuring a Windows Instance Using EC2Launch \(p. 319\)](#)
- [Configuring a Windows Instance Using SSM Config \(p. 326\)](#)
- [Paravirtual Drivers \(p. 352\)](#)
- [Setting Passwords for Windows Instances \(p. 370\)](#)
- [Setting the Time for a Windows Instance \(p. 375\)](#)
- [Configuring a Secondary Private IPv4 Address for Your Windows Instance in a VPC \(p. 378\)](#)

- [Upgrading a Windows Server EC2 Instance to a Newer Version of Windows Server \(p. 382\)](#)

Instance Metadata and User Data

Instance metadata is data about your instance that you can use to configure or manage the running instance. Instance metadata is divided into categories. For more information, see [Instance Metadata Categories \(p. 277\)](#).

EC2 instances can also include *dynamic data*, such as an instance identity document that is generated when the instance is launched. For more information, see [Dynamic Data Categories \(p. 282\)](#).

You can also access the *user data* that you supplied when launching your instance. For example, you can specify parameters for configuring your instance, or attach a simple script. You can also use this data to build more generic AMIs that can be modified by configuration files supplied at launch time. For example, if you run web servers for various small businesses, they can all use the same AMI and retrieve their content from the Amazon S3 bucket you specify in the user data at launch. To add a new customer at any time, simply create a bucket for the customer, add their content, and launch your AMI. If you launch more than one instance at the same time, the user data is available to all instances in that reservation.

Important

Although you can only access instance metadata and user data from within the instance itself, the data is not protected by cryptographic methods. Anyone who can access the instance can view its metadata. Therefore, you should take suitable precautions to protect sensitive data (such as long-lived encryption keys). You should not store sensitive data, such as passwords, as user data.

Contents

- [Retrieving Instance Metadata \(p. 271\)](#)
- [Configuring Instances with User Data \(p. 273\)](#)
- [Retrieving User Data \(p. 276\)](#)
- [Retrieving Dynamic Data \(p. 277\)](#)
- [Instance Metadata Categories \(p. 277\)](#)
- [Instance Identity Documents \(p. 282\)](#)

Retrieving Instance Metadata

Because your instance metadata is available from your running instance, you do not need to use the Amazon EC2 console or the AWS CLI. This can be helpful when you're writing scripts to run from your instance. For example, you can access the local IP address of your instance from instance metadata to manage a connection to an external application.

To view all categories of instance metadata from within a running instance, use the following URI:

```
http://169.254.169.254/latest/meta-data/
```

Note that you are not billed for HTTP requests used to retrieve instance metadata and user data.

You can install a tool such as GNU Wget or cURL to retrieve instance metadata at the command line, or you can copy and paste the URI into a browser. If you do not want to install any third-party tools, you can use PowerShell cmdlets to retrieve the URI. For example, if you are running version 3.0 or later of PowerShell, use the following cmdlet:

```
PS C:\> invoke-restmethod -uri http://169.254.169.254/latest/meta-data/
```

Important

If you do install a third-party tool on a Windows instance, ensure that you read the accompanying documentation carefully, as the method of calling the HTTP and the output format might be different from what is documented here.

All metadata is returned as text (content type text/plain). A request for a specific metadata resource returns the appropriate value, or a 404 - Not Found HTTP error code if the resource is not available.

A request for a general metadata resource (the URI ends with a /) returns a list of available resources, or a 404 - Not Found HTTP error code if there is no such resource. The list items are on separate lines, terminated by line feeds (ASCII 10).

Examples of Retrieving Instance Metadata

This example gets the available versions of the instance metadata. These versions do not necessarily correlate with an Amazon EC2 API version. The earlier versions are available to you in case you have scripts that rely on the structure and information present in a previous version.

```
C:\> curl http://169.254.169.254/
1.0
2007-01-19
2007-03-01
2007-08-29
2007-10-10
2007-12-15
2008-02-01
2008-09-01
2009-04-04
2011-01-01
2011-05-01
2012-01-12
2014-02-25
latest
```

This example gets the top-level metadata items. Some items are only available for instances in a VPC. For more information about each of these items, see [Instance Metadata Categories \(p. 277\)](#).

```
C:\> curl http://169.254.169.254/latest/meta-data/
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
hostname
instance-action
instance-id
instance-type
kernel-id
local-hostname
local-ipv4
mac
network/
placement/
public-hostname
public-ipv4
public-keys/
reservation-id
security-groups
services/
```

These examples get the value of some of the metadata items from the preceding example.

```
C:\> curl http://169.254.169.254/latest/meta-data/ami-id  
ami-12345678
```

```
C:\> curl http://169.254.169.254/latest/meta-data/reservation-id  
r-fea54097
```

```
C:\> curl http://169.254.169.254/latest/meta-data/local-hostname  
ip-10-251-50-12.ec2.internal
```

```
C:\> curl http://169.254.169.254/latest/meta-data/public-hostname  
ec2-203-0-113-25.compute-1.amazonaws.com
```

This example shows the information available for a specific network interface (indicated by the MAC address) on a NAT instance in the EC2-Classical platform.

```
C:\> curl http://169.254.169.254/latest/meta-data/network/interfaces/  
macs/02:29:96:8f:6a:2d/  
device-number  
local-hostname  
local-ipv4s  
mac  
owner-id  
public-hostname  
public-ipv4s
```

This example gets the subnet ID for an instance launched into a VPC.

```
C:\> curl http://169.254.169.254/latest/meta-data/network/interfaces/  
macs/02:29:96:8f:6a:2d/subnet-id  
subnet-be9b61d7
```

Throttling

We throttle queries to the instance metadata service on a per-instance basis, and we place limits on the number of simultaneous connections from an instance to the instance metadata service.

If you're using the instance metadata service to retrieve AWS security credentials, avoid querying for credentials during every transaction or concurrently from a high number of threads or processes, as this may lead to throttling. Instead, we recommend that you cache the credentials until they start approaching their expiry time.

If you're throttled while accessing the instance metadata service, retry your query with an exponential backoff strategy.

Configuring Instances with User Data

When you specify user data, note the following:

- User data is treated as opaque data: what you give is what you get back. It is up to the instance to be able to interpret it.
- User data is limited to 16 KB. This limit applies to the data in raw form, not base64-encoded form.

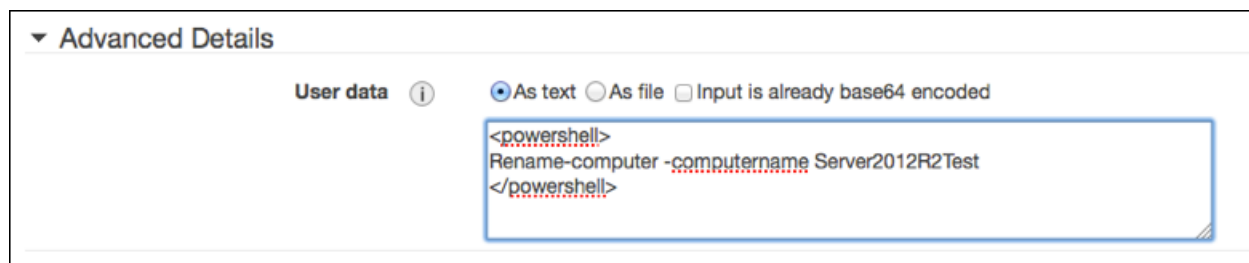
- User data must be base64-encoded before being submitted to the API. The EC2 command line tools perform the base64 encoding for you. The data is decoded before being presented to the instance. For more information about base64 encoding, see <http://tools.ietf.org/html/rfc4648>.
- User data is executed only at launch. If you stop an instance, modify the user data, and start the instance, the new user data is not executed automatically.

Topics

- [Executing Scripts with User Data \(p. 274\)](#)
- [Overriding the Initialize Drives Setting with User Data \(p. 276\)](#)
- [Modify User Data for a Running Instance \(p. 276\)](#)

Executing Scripts with User Data

You can specify scripts to execute when an instance starts. You enter the script in the **User data** section of the Instance Configuration Wizard. The **User data** option is located on the **Step 3: Configure Instance Details** page in the **Advanced Details** section. The example in the following image would change the name of the instance to *Server2012R2Test* when the instance booted.



For EC2Config to execute user data scripts, you must enclose the lines of the specified script within one of the following special tags:

```
<script></script>
```

Run any command that you can run in a Command Prompt window.

Example: `<script>dir > c:\test.log</script>`

```
<powershell></powershell>
```

Run any command that you can run at the Windows PowerShell command prompt.

If you use an AMI that includes the [AWS Tools for Windows PowerShell](#), you can also use those cmdlets. If you specify an IAM role when you launch your instance, then you don't need to specify credentials to the cmdlets, as applications that run on the instance can use the role's credentials to access AWS resources such as Amazon S3 buckets.

Example: `<powershell>Read-S3Object -BucketName myS3Bucket -Key myFolder/myFile.zip -File c:\destinationFile.zip</powershell>`

You can separate the commands in a script using line breaks.

If EC2Config finds `script` or `powershell` tags, it saves the script to a batch or PowerShell file in its `/Scripts` folder. It runs these files when the instance starts. If both `script` and `powershell` tags are present, it runs the batch script first and the PowerShell script next, regardless of the order in which they appear.

The `/Logs` folder contains output from the standard output and standard error streams.

EC2Config expects the user data to be available in base64 encoding. If the user data is not available in base64 encoding, EC2Config logs an error about being unable to find `script` or `powershell` tags

to execute. If your encoding is not correct, the following is an example that sets the encoding using PowerShell.

```
$UserData =  
[System.Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes($Script))
```

Initial Boot

By default, all Amazon AMIs have user data execution enabled for the initial boot. If you click **Shutdown with Sysprep** in EC2Config, user data script execution is enabled, regardless of the setting of the **User Data** check box.

User data script execution happens under the local administrator user only when a random password is generated. This is because EC2Config generates the password and is aware of the credentials briefly (prior to sending to the console). EC2Config doesn't store or track password changes, so when you don't generate a random password, user data execution is performed by the EC2Config service account.

Subsequent Boots

Because Amazon AMIs automatically disable user data script execution after the initial boot, you must do one of the following to make user data persist across reboots:

- Programmatically create a scheduled task to run at system start using `schtasks.exe /Create`, and point the scheduled task to the user data script (or another script) at `C:\Program Files\Amazon\Ec2ConfigService\Scripts\UserScript.ps1`.
- Programmatically enable the user data plug-in in `Config.xml` using a script similar to the following:

```
<powershell>  
$EC2SettingsFile="C:\Program Files\Amazon\Ec2ConfigService\Settings  
\Config.xml"  
$xml = [xml](get-content $EC2SettingsFile)  
$xmlElement = $xml.get_DocumentElement()  
$xmlElementToModify = $xmlElement.Plugins  
  
foreach ($element in $xmlElementToModify.Plugin)  
{  
    if ($element.name -eq "Ec2SetPassword")  
    {  
        $element.State="Enabled"  
    }  
    elseif ($element.name -eq "Ec2HandleUserData")  
    {  
        $element.State="Enabled"  
    }  
}  
$xml.Save($EC2SettingsFile)  
</powershell>
```

- Starting with EC2Config version 2.1.10, you can use `<persist>>true</persist>` to enable the plug-in after user data execution.

```
<powershell>  
    insert script here  
</powershell>  
<persist>true</persist>
```

Overriding the Initialize Drives Setting with User Data

Use the following to override the initialize drives setting with user data. These settings will be used every time you reboot the instance.

```
<InitializeDrivesSettings>  
  <SettingsGroup>FormatWithTRIM</SettingsGroup>  
</InitializeDrivesSettings>
```

Use a settings group to specify how you want to initialize drives.

Important

In EC2Config version 3.18 or later, the TRIM command is disabled for the duration of the disk format operation, by default. This change improves formatting times in Windows. To enable TRIM for the duration of the disk format operation, specify `FormatWithTRIM` for EC2Config version 3.18 or later. `FormatWithoutTRIM` is still available for earlier versions of EC2Config. Use `FormatWithoutTRIM` to disable TRIM.

- `FormatWithTRIM` (v3.18 and above): This setting enables the TRIM command when formatting drives. After a drive has been formatted and initialized, the system restores TRIM configuration.
- `FormatWithoutTRIM`: This setting disables the TRIM command when formatting drives and improves formatting times in Windows. After a drive has been formatted and initialized, the system restores TRIM configuration.
- `DisableInitializeDrives`: This setting disables formatting for new drives. Use this setting to initialize drives manually.

Modify User Data for a Running Instance

You can modify user data for an instance that previously had user data assigned and is currently running. The new user data is available on your instance after you reboot it.

To modify the user data for an Amazon EBS-backed instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**, and select the instance.
3. Click **Actions**, select **Instance State**, and then choose **Stop**.

Warning

When you stop an instance, the data on any instance store volumes is erased. Therefore, if you have any data on instance store volumes that you want to keep, be sure to back it up to persistent storage.

4. In the confirmation dialog box, click **Yes, Stop**. It can take a few minutes for the instance to stop.
5. With the instance still selected, choose **Actions**, select **Instance Settings**, and then choose **View/Change User Data**. Note that you can't change the user data if the instance is running, but you can view it.
6. In the **View/Change User Data** dialog box, update the user data, and then choose **Save**.

Retrieving User Data

To retrieve user data, use the following URI:

```
http://169.254.169.254/latest/user-data
```

Requests for user data returns the data as it is (content type `application/x-octetstream`).

This shows an example of returning comma-separated user data.

```
C:\> curl http://169.254.169.254/latest/user-data
1234,john,reboot,true | 4512,richard, | 173,,,
```

This shows an example of returning line-separated user data.

```
C:\> curl http://169.254.169.254/latest/user-data
[general]
instances: 4

[instance-0]
s3-bucket: <user_name>

[instance-1]
reboot-on-error: yes
```

Retrieving Dynamic Data

To retrieve dynamic data from within a running instance, use the following URI:

```
http://169.254.169.254/latest/dynamic/
```

This example shows how to retrieve the high-level instance identity categories:

```
C:\> curl http://169.254.169.254/latest/dynamic/instance-identity/
pkcs7
signature
document
```

For more information about dynamic data and examples of how to retrieve it, see [Instance Identity Documents \(p. 282\)](#).

Instance Metadata Categories

The following table lists the categories of instance metadata.

Data	Description	Version Introduced
ami-id	The AMI ID used to launch the instance.	1.0
ami-launch-index	If you started more than one instance at the same time, this value indicates the order in which the instance was launched. The value of the first instance launched is 0.	1.0
ami-manifest-path	The path to the AMI manifest file in Amazon S3. If you used an Amazon EBS-backed AMI to launch the instance, the returned result is unknown.	1.0
ancestor-ami-ids	The AMI IDs of any instances that were rebundled to create this	2007-10-10

Amazon Elastic Compute Cloud
User Guide for Windows Instances
Instance Metadata and User Data

Data	Description	Version Introduced
	AMI. This value will only exist if the AMI manifest file contained an <code>ancestor-ami</code> key.	
<code>block-device-mapping/ami</code>	The virtual device that contains the root/boot file system.	2007-12-15
<code>block-device-mapping/ebs</code> <i>N</i>	The virtual devices associated with Amazon EBS volumes, if any are present. Amazon EBS volumes are only available in metadata if they were present at launch time or when the instance was last started. The <i>N</i> indicates the index of the Amazon EBS volume (such as <code>ebs1</code> or <code>ebs2</code>).	2007-12-15
<code>block-device-mapping/ephemeral</code> <i>N</i>	The virtual devices associated with ephemeral devices, if any are present. The <i>N</i> indicates the index of the ephemeral volume.	2007-12-15
<code>block-device-mapping/root</code>	The virtual devices or partitions associated with the root devices, or partitions on the virtual device, where the root (/ or C:) file system is associated with the given instance.	2007-12-15
<code>block-device-mapping/swap</code>	The virtual devices associated with <code>swap</code> . Not always present.	2007-12-15
<code>hostname</code>	The private IPv4 DNS hostname of the instance. In cases where multiple network interfaces are present, this refers to the <code>eth0</code> device (the device for which the device number is 0).	1.0
<code>iam/info</code>	If there is an IAM role associated with the instance at launch, contains information about the last time the instance profile was updated, including the instance's <code>LastUpdated</code> date, <code>InstanceProfileArn</code> , and <code>InstanceProfileId</code> . Otherwise, not present.	2012-01-12
<code>iam/security-credentials/role-name</code>	If there is an IAM role associated with the instance at launch, <code>role-name</code> is the name of the role, and <code>role-name</code> contains the temporary security credentials associated with the role (for more information, see Retrieving Security Credentials from Instance Metadata (p. 659)). Otherwise, not present.	2012-01-12

Amazon Elastic Compute Cloud
User Guide for Windows Instances
Instance Metadata and User Data

Data	Description	Version Introduced
<code>instance-action</code>	Notifies the instance that it should reboot in preparation for bundling. Valid values: <code>none</code> <code>shutdown</code> <code>bundle-pending</code> .	2008-09-01
<code>instance-id</code>	The ID of this instance.	1.0
<code>instance-type</code>	The type of instance. For more information, see Instance Types (p. 117) .	2007-08-29
<code>kernel-id</code>	The ID of the kernel launched with this instance, if applicable.	2008-02-01
<code>local-hostname</code>	The private IPv4 DNS hostname of the instance. In cases where multiple network interfaces are present, this refers to the <code>eth0</code> device (the device for which the device number is 0).	2007-01-19
<code>local-ipv4</code>	The private IPv4 address of the instance. In cases where multiple network interfaces are present, this refers to the <code>eth0</code> device (the device for which the device number is 0).	1.0
<code>mac</code>	The instance's media access control (MAC) address. In cases where multiple network interfaces are present, this refers to the <code>eth0</code> device (the device for which the device number is 0).	2011-01-01
<code>network/interfaces/macs/mac/device-number</code>	The unique device number associated with that interface. The device number corresponds to the device name; for example, a <code>device-number</code> of 2 is for the <code>eth2</code> device. This category corresponds to the <code>DeviceIndex</code> and <code>device-index</code> fields that are used by the Amazon EC2 API and the EC2 commands for the AWS CLI.	2011-01-01
<code>network/interfaces/macs/mac/ipv4-associations/public-ip</code>	The private IPv4 addresses that are associated with each <code>public-ip</code> address and assigned to that interface.	2011-01-01
<code>network/interfaces/macs/mac/ipv6s</code>	The IPv6 addresses associated with the interface. Returned only for instances launched into a VPC.	2016-06-30
<code>network/interfaces/macs/mac/local-hostname</code>	The interface's local hostname.	2011-01-01
<code>network/interfaces/macs/mac/local-ipv4s</code>	The private IPv4 addresses associated with the interface.	2011-01-01

Data	Description	Version Introduced
<code>network/interfaces/macs/mac/mac</code>	The instance's MAC address.	2011-01-01
<code>network/interfaces/macs/mac/owner-id</code>	The ID of the owner of the network interface. In multiple-interface environments, an interface can be attached by a third party, such as Elastic Load Balancing. Traffic on an interface is always billed to the interface owner.	2011-01-01
<code>network/interfaces/macs/mac/public-hostname</code>	The interface's public DNS (IPv4). If the instance is in a VPC, this category is only returned if the <code>enableDnsHostnames</code> attribute is set to <code>true</code> . For more information, see Using DNS with Your VPC .	2011-01-01
<code>network/interfaces/macs/mac/public-ipv4s</code>	The Elastic IP addresses associated with the interface. There may be multiple IPv4 addresses on an instance.	2011-01-01
<code>network/interfaces/macs/mac/security-groups</code>	Security groups to which the network interface belongs. Returned only for instances launched into a VPC.	2011-01-01
<code>network/interfaces/macs/mac/security-group-ids</code>	The IDs of the security groups to which the network interface belongs. Returned only for instances launched into a VPC. For more information on security groups in the EC2-VPC platform, see Security Groups for Your VPC .	2011-01-01
<code>network/interfaces/macs/mac/subnet-id</code>	The ID of the subnet in which the interface resides. Returned only for instances launched into a VPC.	2011-01-01
<code>network/interfaces/macs/mac/subnet-ipv4-cidr-block</code>	The IPv4 CIDR block of the subnet in which the interface resides. Returned only for instances launched into a VPC.	2011-01-01
<code>network/interfaces/macs/mac/subnet-ipv6-cidr-blocks</code>	The IPv6 CIDR block of the subnet in which the interface resides. Returned only for instances launched into a VPC.	2016-06-30
<code>network/interfaces/macs/mac/vpc-id</code>	The ID of the VPC in which the interface resides. Returned only for instances launched into a VPC.	2011-01-01
<code>network/interfaces/macs/mac/vpc-ipv4-cidr-block</code>	The IPv4 CIDR block of the VPC in which the interface resides. Returned only for instances launched into a VPC.	2011-01-01

Amazon Elastic Compute Cloud
User Guide for Windows Instances
Instance Metadata and User Data

Data	Description	Version Introduced
<code>network/interfaces/mac/mac/vpc-ipv4-cidr-blocks</code>	The IPv4 CIDR block of the VPC in which the interface resides. Returned only for instances launched into a VPC.	2016-06-30
<code>network/interfaces/mac/mac/vpc-ipv6-cidr-blocks</code>	The IPv6 CIDR block of the VPC in which the interface resides. Returned only for instances launched into a VPC.	2016-06-30
<code>placement/availability-zone</code>	The Availability Zone in which the instance launched.	2008-02-01
<code>product-codes</code>	Product codes associated with the instance, if any.	2007-03-01
<code>public-hostname</code>	The instance's public DNS. If the instance is in a VPC, this category is only returned if the <code>enableDnsHostnames</code> attribute is set to <code>true</code> . For more information, see Using DNS with Your VPC .	2007-01-19
<code>public-ipv4</code>	The public IPv4 address. If an Elastic IP address is associated with the instance, the value returned is the Elastic IP address.	2007-01-19
<code>public-keys/0/openssh-key</code>	Public key. Only available if supplied at instance launch time.	1.0
<code>ramdisk-id</code>	The ID of the RAM disk specified at launch time, if applicable.	2007-10-10
<code>reservation-id</code>	The ID of the reservation.	1.0
<code>security-groups</code>	The names of the security groups applied to the instance. After launch, you can only change the security groups of instances running in a VPC. Such changes are reflected here and in <code>network/interfaces/mac/mac/security-groups</code> .	1.0
<code>services/domain</code>	The domain for AWS resources for the region; for example, <code>amazonaws.com</code> for <code>us-east-1</code> .	2014-02-25
<code>services/partition</code>	The partition that the resource is in. For standard AWS regions, the partition is <code>aws</code> . If you have resources in other partitions, the partition is <code>aws-partitionname</code> . For example, the partition for resources in the China (Beijing) region is <code>aws-cn</code> .	2015-10-20

Data	Description	Version Introduced
spot/termination-time	The approximate time, in UTC, that the operating system for your Spot instance will receive the shutdown signal. This item is present and contains a time value (for example, 2015-01-05T18:02:00Z) only if the Spot instance has been marked for termination by Amazon EC2. The termination-time item is not set to a time if you terminated the Spot instance yourself.	2014-11-05

Dynamic Data Categories

The following table lists the categories of dynamic data.

Data	Description	Version introduced
fws/instance-monitoring	Value showing whether the customer has enabled detailed one-minute monitoring in CloudWatch. Valid values: enabled disabled	2009-04-04
instance-identity/document	JSON containing instance attributes, such as instance-id, private IP address, etc. See Instance Identity Documents (p. 282) .	2009-04-04
instance-identity/pkcs7	Used to verify the document's authenticity and content against the signature. See Instance Identity Documents (p. 282) .	2009-04-04
instance-identity/signature	Data that can be used by other parties to verify its origin and authenticity. See Instance Identity Documents (p. 282) .	2009-04-04

Instance Identity Documents

An instance identity document is a JSON file that describes an instance. The instance identity document is accompanied by a signature and a PKCS7 signature which can be used to verify the accuracy, origin, and authenticity of the information provided in the document. For example, you may have downloaded free software with paid updates.

The instance identity document is generated when the instance is launched, and exposed to the instance through [instance metadata \(p. 271\)](#). It validates the attributes of the instances, such as the subscribed software, instance size, instance type, operating system, and AMI.

Important

Due to the dynamic nature of instance identity documents and signatures, we recommend retrieving the instance identity document and signature regularly.

Obtaining the Instance Identity Document and Signatures

To retrieve the instance identity document, use the following URL from your running instance:

```
http://169.254.169.254/latest/dynamic/instance-identity/document
```

```
{
  "devpayProductCodes" : null,
  "availabilityZone" : "us-east-1d",
  "privateIp" : "10.158.112.84",
  "version" : "2010-08-31",
  "region" : "us-east-1",
  "instanceId" : "i-1234567890abcdef0",
  "billingProducts" : null,
  "instanceType" : "t1.micro",
  "accountId" : "123456789012",
  "pendingTime" : "2015-11-19T16:32:11Z",
  "imageId" : "ami-5fb8c835",
  "kernelId" : "aki-919dcaf8",
  "ramdiskId" : null,
  "architecture" : "x86_64"
}
```

To retrieve the instance identity signature, use the following URL from your running instance:

```
http://169.254.169.254/latest/dynamic/instance-identity/signature
```

```
dExamplesjNqhhJan7pORLpLSr7lJEF4V2DhKGLyoYVBoUYrY9njyBCmhEayaGrhtS/AWY+LPx
lVSQURF5n0gwpNCuO6ICT0fNrm5IH7w9ydyexamplejJw8XvWPxbuRkcN0TAA1p4RtCAqm4ms
x2oALjWSCBExample=
```

To retrieve the PKCS7 signature, use the following URL from your running instance:

```
http://169.254.169.254/latest/dynamic/instance-identity/pkcs7
```

```
MIICiTCCAfICCCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMx CzAJBgNVBAGTAldBMRAdG9YDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24x FDASBgNVBAStC0lBTsBDb25zb2xlMRlW EAYDVQQDEw1UZXR0Q21sYWMxHZAAd
BgkqhkiG9w0BCQEWEG5vb25lQGFTYXpvi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMx CzAJBgNVBAGTAldBMRAdG9YD
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24x FDASBgNVBAStC0lBTsBDb25z
b2xlMRlW EAYDVQQDEw1UZXR0Q21sYWMxHZAAdBgkqhkiG9w0BCQEWEG5vb25lQGFT
YXpvi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySwTC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb30hjZnczvQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntned9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFbjvSfpJlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvJx79LjStb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE
```

Configuring a Windows Instance Using the EC2Config Service

Windows AMIs include an optional service called the EC2Config service (EC2Config.exe). EC2Config starts when the instance boots and performs tasks during startup and each time you stop or start the instance. EC2Config can also perform tasks on demand. Some of these tasks are automatically enabled, while others must be enabled manually. Although optional, this service provides access to advanced features that aren't otherwise available. This service runs in the LocalSystem account.

Note

Windows Server 2016 AMIs do not include the EC2Config service. For more information, see [Changes in Windows Server 2016 AMIs \(p. 108\)](#).

EC2Config uses settings files to control its operation. You can update these settings files using either a graphical tool or by directly editing XML files. The service binaries and additional files are contained in the `%ProgramFiles%\Amazon\EC2ConfigService` directory.

Contents

- [Overview of EC2Config Tasks \(p. 284\)](#)
- [Ec2 Service Properties \(p. 287\)](#)
- [EC2Config Settings Files \(p. 289\)](#)
- [Configure Proxy Settings for the EC2Config Service \(p. 293\)](#)
- [Managing the EC2Config Service \(p. 295\)](#)
- [Sending Performance Counters to CloudWatch and Logs to CloudWatch Logs Using EC2Config \(p. 307\)](#)
- [Troubleshooting Problems with the EC2Config Service \(p. 317\)](#)

Overview of EC2Config Tasks

EC2Config runs initial startup tasks when the instance is first started and then disables them. To run these tasks again, you must explicitly enable them prior to shutting down the instance, or by running Sysprep manually. These tasks are as follows:

- Set a random, encrypted password for the administrator account.
- Generate and install the host certificate used for Remote Desktop Connection.
- Dynamically extend the operating system partition to include any unpartitioned space.
- Execute the specified user data (and Cloud-Init, if it's installed).

EC2Config performs the following tasks every time the instance starts:

- Change the host name to match the private IP address in Hex notation (this task is disabled by default and must be enabled in order to run at instance start).
- Configure the key management server (KMS), check for Windows activation status, and activate Windows as necessary.
- Mount all Amazon EBS volumes and instance store volumes, and map volume names to drive letters.
- Write event log entries to the console to help with troubleshooting (this task is disabled by default and must be enabled in order to run at instance start).
- Write to the console that Windows is ready.
- Add a custom route to the primary network adapter to enable the following IP addresses when multiple NICs are attached: 169.254.169.250, 169.254.169.251, and 169.254.169.254. These addresses are used by Windows Activation and when you access instance metadata.

EC2Config performs the following task every time a user logs in:

- Display wallpaper information to the desktop background.

While the instance is running, you can request that EC2Config perform the following task on demand:

- Run Sysprep and shut down the instance so that you can create an AMI from it. For more information, see [Create a Standard Amazon Machine Image Using Sysprep \(p. 111\)](#).

EC2Config and Amazon EC2 Simple Systems Manager (SSM)

The EC2Config service processes SSM requests on instances created from Windows Server 2003-2012 R2 AMIs published before November 2016.

Beginning with the release of November 2016 AMIs, Windows Server 2003-2012 R2 AMIs include the EC2Config service and SSM Agent. EC2Config performs all of the tasks described earlier and SSM Agent processes requests for Run Command and SSM Config. The following table describes how this change affects different components and configurations.

Note

Windows Server 2016 AMIs don't include the EC2Config service. For more information, see [Changes in Windows Server 2016 AMIs \(p. 108\)](#).

In the following table, *legacy* refers to AMIs, instances, or the EC2Config service before November 2016. *New* refers to November 2016 and later.

Item	Details
AMIs and instances	<p>Legacy (pre-November 2016) AMIs use the legacy EC2Config service. The EC2Config service performs initialization tasks during instance launch and processes SSM requests on Windows instances.</p> <p>If you attempt to execute commands using SSM features released after November 2016, the commands fail because SSM features developed after November 2016 must be processed by SSM Agent, not the EC2Config service. To avoid errors, upgrade EC2Config using Run Command. For more information, see Updating the SSM Agent Using Amazon EC2 Run Command (p. 470).</p> <p>After upgrade, AMIs run the EC2Config service and SSM Agent. EC2Config service processes tasks described earlier in this topic and SSM Agent processes Run Command and SSM Config requests. Instances can process commands using new SSM features released after November 2016.</p> <p>Note Windows managed instances (on-premises servers or VMs configured for Run Command) use the SSM agent to process Run Command requests, regardless of when they were registered. For more information about managed instances, see Setting Up Systems Manager in Hybrid Environments (p. 397)</p>
Imported AMIs	Windows Server 2003-2012 R2 AMIs imported into EC2 after November 2016 include the new EC2Config service and SSM Agent.
EC2Config installer	If the latest EC2Config installer detects the legacy version of the EC2Config service,

Item	Details
	the installer installs the new version of the EC2Config service and SSM Agent.
SSM agent installer	<ul style="list-style-type: none"> If the latest version of SSM Agent installer detects the legacy version of the EC2Config service, the installation fails. You must run the latest version of the EC2Config installer to update the EC2Config service <i>and</i> install SSM Agent. If the latest version of SSM Agent installer detects the new version of the EC2Config service, the installer installs the latest version of SSM Agent.
Execute the Run Command AWS-UpdateEC2Config document	If you execute this command against any Windows Server 2003 - 2012 R2 instance after November 8, 2016, the command runs the latest EC2Config installer, which installs the new version of the EC2Config service and SSM Agent.
Run Command AWS-UpdateSSMAgent document	If you execute this command against any instance after November 8, 2016, the command runs the most recent SSM Agent installer. If Run Command detects the legacy EC2Config service, the command fails. If Run Command detects the new version of the EC2Config service, the command updates SSM Agent independently.
Amazon WorkSpaces environment	Amazon WorkSpaces AMIs are configured with the appropriate agent based on the date the AMI was created.

You can use Run Command to upgrade your existing instances to use to the latest version of the EC2Config service and SSM Agent. For more information, see [Updating the SSM Agent Using Amazon EC2 Run Command \(p. 470\)](#).

EC2Config and SysPrep

The EC2Config service runs Sysprep, a Microsoft tool that enables you to create a customized Windows AMI that can be reused. When EC2Config calls Sysprep, it uses the settings files in `%ProgramFiles%\Amazon\EC2ConfigService\Settings` to determine which operations to perform. You can edit these files indirectly using the **Ec2 Service Properties** dialog box, or directly using an XML editor or a text editor. However, there are some advanced settings that aren't available in the **Ec2 Service Properties** dialog box, so you must edit those entries directly.

Note

Windows Server 2016 AMIs do not include the EC2Config service. Sysprep is handled by the EC2Launch Windows PowerShell script. For more information, see [Changes in Windows Server 2016 AMIs \(p. 108\)](#).

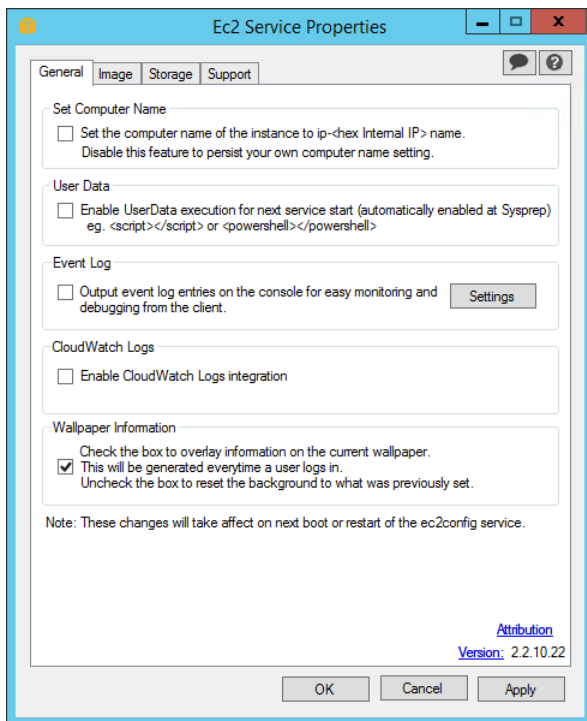
If you create an AMI from an instance after updating its settings, the new settings are applied to any instance that's launched from the new AMI. For information about creating an AMI, see [Creating an Amazon EBS-Backed Windows AMI \(p. 77\)](#).

Ec2 Service Properties

The following procedure describes how to use the **Ec2 Service Properties** dialog box to enable or disable settings.

To change settings using the Ec2 Service Properties dialog box

1. Launch and connect to your Windows instance.
2. From the **Start** menu, click **All Programs**, and then click **EC2ConfigService Settings**.



3. On the **General** tab of the **Ec2 Service Properties** dialog box, you can enable or disable the following settings.

Set Computer Name

If this setting is enabled (it is disabled by default), the host name is compared to the current internal IP address at each boot; if the host name and internal IP address do not match, the host name is reset to contain the internal IP address and then the system reboots to pick up the new host name. To set your own host name, or to prevent your existing host name from being modified, do not enable this setting.

User Data

User data execution enables you to inject scripts into the instance metadata during the first launch. From an instance, you can read user data at <http://169.254.169.254/latest/user-data/>. The scripts remain static for the life of the instance, persisting when the instance is stopped and started, until it is terminated.

If you use a large script, we recommend that you use user data to download the script, and then execute it.

For more information, see [Executing Scripts with User Data \(p. 274\)](#).

Event Log

Use this setting to display event log entries on the console during boot for easy monitoring and debugging.

Click **Settings** to specify filters for the log entries sent to the console. The default filter sends the three most recent error entries from the system event log to the console.

CloudWatch Logs

Starting with EC2Config version 2.2.5 (version 2.2.6 or later is recommended), you can export all Windows Server messages in the System log, Security log, Application log, and IIS log to CloudWatch Logs and monitor them using CloudWatch metrics. EC2Config version 2.2.10 or later adds the ability to export any event log data, Event Tracing (Windows) data, or text-based log files to CloudWatch Logs. In addition, you can also export performance counter data to CloudWatch. For more information, see [Monitoring System, Application, and Custom Log Files](#) in the Amazon CloudWatch User Guide.

1. Select **Enable CloudWatch integration**, and then click **OK**.
2. Edit the `Amazon\Ec2ConfigService\Settings\AWS.EC2.Windows.CloudWatch.json` file and configure the types of logs you want to send to CloudWatch Logs. For more information, see [Sending Performance Counters to CloudWatch and Logs to CloudWatch Logs Using EC2Config](#) (p. 307).

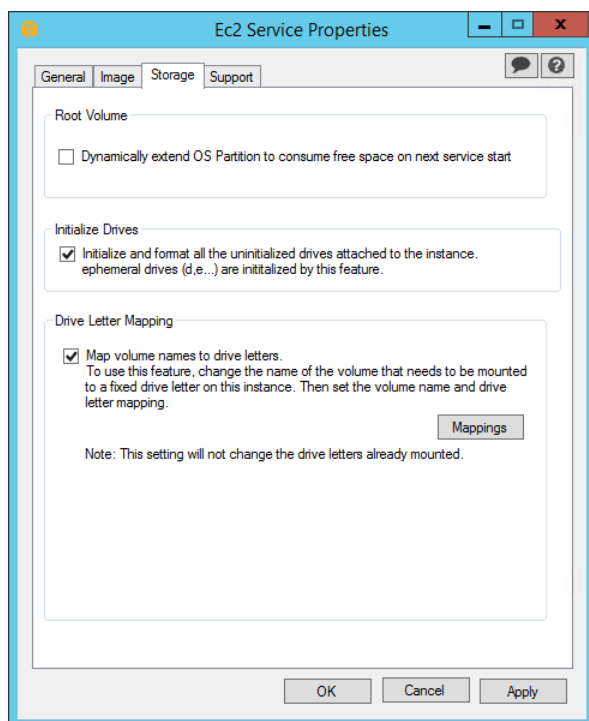
Wallpaper Information

Use this setting to display system information on the desktop background. The following is an example of the information displayed on the desktop background.

```
Hostname      : WIN-U0RFOJCTPUU
Instance ID   : i-d583f76a
Public IP Address : 54.208.43.227
Private IP Address : 172.31.42.195
Availability Zone : us-east-1b
Instance Size  : t2.micro
Architecture  : AMD64
```

The information displayed on the desktop background is controlled by the settings file `EC2ConfigService\Settings\WallpaperSettings.xml`.

4. Click the **Storage** tab. You can enable or disable the following settings.



Root Volume

This setting dynamically extends Disk 0/Volume 0 to include any unpartitioned space. This can be useful when the instance is booted from a root device volume that has a custom size.

Initialize Drives

This setting formats and mounts all volumes attached to the instance during start.

Drive Letter Mapping

The system maps the volumes attached to an instance to drive letters. For Amazon EBS volumes, the default is to assign drive letters going from D: to Z:. For instance store volumes, the default depends on the driver. Citrix PV drivers assign instance store volumes drive letters going from Z: to A:. Red Hat drivers assign instance store volumes drive letters going from D: to Z:.

To choose the drive letters for your volumes, click **Mappings**. In the **DriveLetterSetting** dialog box, specify the **Volume Name** and **Drive Letter** values for each volume, and then click **OK**. We recommend that you select drive letters that avoid conflicts with drive letters that are likely to be in use, such as drive letters in the middle of the alphabet.

After you specify a drive letter mapping and attach a volume with same label as one of the volume names that you specified, EC2Config automatically assigns your specified drive letter to that volume. However, the drive letter mapping fails if the drive letter is already in use. Note that EC2Config doesn't change the drive letters of volumes that were already mounted when you specified the drive letter mapping.

5. To save your settings and continue working on them later, click **OK** to close the **Ec2 Service Properties** dialog box. If you have finished customizing your instance and want to create an AMI from that instance, see [Create a Standard Amazon Machine Image Using Sysprep \(p. 111\)](#).

EC2Config Settings Files

The settings files control the operation of the EC2Config service. These files are located in the `C:\Program Files\Amazon\Ec2ConfigService\Settings` directory:

- `ActivationSettings.xml`—Controls product activation using a key management server (KMS).
- `AWS.EC2.Windows.CloudWatch.json`—Controls which performance counters to send to CloudWatch and which logs to send to CloudWatch Logs. For more information about how to change the settings in this file, see [Sending Performance Counters to CloudWatch and Logs to CloudWatch Logs Using EC2Config \(p. 307\)](#).
- `BundleConfig.xml`—Controls how EC2Config prepares an instance store-backed instance for AMI creation. Note that the only Windows instances that can be backed by instance store are those for Windows Server 2003.
- `Config.xml`—Controls the primary settings.
- `DriveLetterConfig.xml`—Controls drive letter mappings.
- `EventLogConfig.xml`—Controls the event log information that's displayed on the console while the instance is booting.
- `WallpaperSettings.xml`—Controls the information that's displayed on the desktop background.

ActivationSettings.xml

This file contains settings that control product activation. When Windows boots, the EC2Config service checks whether Windows is already activated. If Windows is not already activated, it attempts to activate Windows by searching for the specified KMS server.

- `SetAutodiscover`—Indicates whether to detect a KMS automatically.

- **TargetKMSServer**—Stores the private IP address of a KMS. The KMS must be in the same region as your instance.
- **DiscoverFromZone**—Discovers the KMS server from the specified DNS zone.
- **ReadFromUserData**—Gets the KMS server from UserData.
- **LegacySearchZones**—Discovers the KMS server from the specified DNS zone.
- **DoActivate**—Attempts activation using the specified settings in the section. This value can be true or false.
- **LogResultToConsole**—Displays the result to the console.

BundleConfig.xml

This file contains settings that control how EC2Config prepares an instance for AMI creation.

- **AutoSysprep**—Indicates whether to use Sysprep automatically. Change the value to `Yes` to use Sysprep.
- **SetRDPCertificate**—Sets a self-signed certificate to the Remote Desktop server running on a Windows 2003 instance. This enables you to securely RDP into the instances. Change the value to `Yes` if the new instances should have the certificate.

This setting is not used with Windows Server 2008 or Windows Server 2012 instances because they can generate their own certificates.

- **SetPasswordAfterSysprep**—Sets a random password on a newly launched instance, encrypts it with the user launch key, and outputs the encrypted password to the console. Change the value of this setting to `No` if the new instances should not be set to a random encrypted password.

Config.xml

Plug-ins

- **Ec2SetPassword**—Generates a random encrypted password each time you launch an instance. This feature is disabled by default after the first launch so that reboots of this instance don't change a password set by the user. Change this setting to `Enabled` to continue to generate passwords each time you launch an instance.

This setting is important if you are planning to create an AMI from your instance.

- **Ec2SetComputerName**—Sets the host name of the instance to a unique name based on the IP address of the instance and reboots the instance. To set your own host name, or prevent your existing host name from being modified, you must disable this setting.
- **Ec2InitializeDrives**—Initializes and formats all volumes during startup. This feature is enabled by default.
- **Ec2EventLog**—Displays event log entries in the console. By default, the three most recent error entries from the system event log are displayed. To specify the event log entries to display, edit the `EventLogConfig.xml` file located in the `EC2ConfigService\Settings` directory. For information about the settings in this file, see [Eventlog Key](#) in the MSDN Library.
- **Ec2ConfigureRDP**—Sets up a self-signed certificate on the instance, so users can securely access the instance using Remote Desktop. This feature is disabled on Windows Server 2008 and Windows Server 2012 instances because they can generate their own certificates.
- **Ec2OutputRDPcert**—Displays the Remote Desktop certificate information to the console so that the user can verify it against the thumbprint.
- **Ec2SetDriveLetter**—Sets the drive letters of the mounted volumes based on user-defined settings. By default, when an Amazon EBS volume is attached to an instance, it can be mounted using the drive letter on the instance. To specify your drive letter mappings, edit the `DriveLetterConfig.xml` file located in the `EC2ConfigService\Settings` directory.

- `Ec2WindowsActivate`—The plug-in handles Windows activation. It checks to see if Windows is activated. If not, it updates the KMS client settings, and then activates Windows.

To modify the KMS settings, edit the `ActivationSettings.xml` file located in the `EC2ConfigService\Settings` directory.

- `Ec2DynamicBootVolumeSize`—Extends Disk 0/Volume 0 to include any unpartitioned space.
- `Ec2HandleUserData`—Creates and executes scripts created by the user on the first launch of an instance after Sysprep is run. Commands wrapped in script tags are saved to a batch file, and commands wrapped in PowerShell tags are saved to a .ps1 file.

Global Settings

- `ManageShutdown`—Ensures that instances launched from instance store-backed AMIs do not terminate while running Sysprep.
- `SetDnsSuffixList`—Sets the DNS suffix of the network adapter for Amazon EC2. This allows DNS resolution of servers running in Amazon EC2 without providing the fully qualified domain name.
- `WaitForMetaDataAvailable`—Ensures that the EC2Config service will wait for metadata to be accessible and the network available before continuing with the boot. This check ensures that EC2Config can obtain information from metadata for activation and other plug-ins.
- `ShouldAddRoutes`—Adds a custom route to the primary network adapter to enable the following IP addresses when multiple NICs are attached: 169.254.169.250, 169.254.169.251, and 169.254.169.254. These addresses are used by Windows Activation and when you access instance metadata.
- `RemoveCredentialsfromSyspreponStartup`—Removes the administrator password from `Sysprep.xml` the next time the service starts. To ensure that this password persists, edit this setting.

DriveLetterConfig.xml

This file contains settings that control drive letter mappings. By default, a volume can be mapped to any available drive letter. You can mount a volume to a particular drive letter as follows.

```
<?xml version="1.0" standalone="yes"?>
<DriveLetterMapping>
  <Mapping>
    <VolumeName></VolumeName>
    <DriveLetter></DriveLetter>
  </Mapping>
  . . .
  <Mapping>
    <VolumeName></VolumeName>
    <DriveLetter></DriveLetter>
  </Mapping>
</DriveLetterMapping>
```

- `VolumeName`—The volume label. For example, `My Volume`. To specify a mapping for an instance storage volume, use the label `Temporary Storage X`, where `X` is a number from 0 to 25.
- `DriveLetter`—The drive letter. For example, `M:`. The mapping fails if the drive letter is already in use.

EventLogConfig.xml

This file contains settings that control the event log information that's displayed on the console while the instance is booting. By default, we display the three most recent error entries from the System event log.

- `Category`—The event log key to monitor.
- `ErrorType`—The event type (for example, `Error`, `Warning`, `Information`.)
- `NumEntries`—The number of events stored for this category.
- `LastMessageTime`—To prevent the same message from being pushed repeatedly, the service updates this value every time it pushes a message.
- `AppName`—The event source or application that logged the event.

WallpaperSettings.xml

This file contains settings that control the information that's displayed on the desktop background. The following information is displayed by default.

- `Hostname`—Displays the computer name.
- `Instance ID`—Displays the ID of the instance.
- `Public IP Address`—Displays the public IP address of the instance.
- `Private IP Address`—Displays the private IP address of the instance.
- `Availability Zone`—Displays the Availability Zone in which the instance is running.
- `Instance Size`—Displays the type of instance.
- `Architecture`—Displays the setting of the `PROCESSOR_ARCHITECTURE` environment variable.

You can remove any of the information that's displayed by default by deleting its entry. You can add additional instance metadata to display as follows.

```
<WallpaperInformation>
  <name>display_name</name>
  <source>metadata</source>
  <identifier>meta-data/path</identifier>
</WallpaperInformation>
```

You can add additional System environment variables to display as follows.

```
<WallpaperInformation>
  <name>display_name</name>
  <source>EnvironmentVariable</source>
  <identifier>variable-name</identifier>
</WallpaperInformation>
```

InitializeDrivesSettings.xml

This file contains settings that control how EC2Config initialize drives. By default, EC2Config initialize drives that were not brought online with the operating system. You can customize the plugin by using the following settings.

```
<InitializeDrivesSettings>
  <SettingsGroup>FormatWithTRIM</SettingsGroup>
</InitializeDrivesSettings>
```

Use a settings group to specify how you want to initialize drives.

Important

In EC2Config version 3.18 or later, the TRIM command is disabled for the duration of the disk format operation, by default. This change improves formatting times in Windows. To enable TRIM for the duration of the disk format operation, specify `FormatWithTRIM` for EC2Config version 3.18 or later. `FormatWithoutTRIM` is still available for earlier versions of EC2Config. Use `FormatWithoutTRIM` to disable TRIM.

- `FormatWithTRIM` (v3.18 and above): This setting enables the TRIM command when formatting drives. After a drive has been formatted and initialized, the system restores TRIM configuration.
- `FormatWithoutTRIM`: This setting disables the TRIM command when formatting drives and improves formatting times in Windows. After a drive has been formatted and initialized, the system restores TRIM configuration.
- `DisableInitializeDrives`: This setting disables formatting for new drives. Use this setting to initialize drives manually.

Configure Proxy Settings for the EC2Config Service

You can configure the EC2Config service to communicate through a proxy. This section describes three methods: using the AWS SDK for .NET, using the .Net system element, and using Microsoft Group Policy and Internet Explorer. Using the AWS SDK for .NET is the preferred method because you can specify a user name and password.

Configure Proxy Settings Using the AWS SDK for .NET (Preferred)

You can configure proxy settings for the EC2Config service by specifying the `proxy` element in the `Ec2Config.exe.config` file. For more information about the `proxy` element, see the [Configuration Files Reference for AWS SDK for .NET](#).

To specify the `proxy` element in the `Ec2Config.exe.config` file

1. Edit the `Ec2Config.exe.config` file on an instance where you want the EC2Config service to communicate through a proxy. By default, the file is located in the following directory:
%ProgramFiles%\Amazon\Ec2ConfigService.
2. Add the following `aws` element to the `<configSections>`. Do *not* add this to any existing `<sectionGroups>`.

For EC2Config versions 3.17 or earlier

```
<configSections>
  <section name="aws" type="Amazon.AWSSection, AWSSDK" />
</configSections>
```

For EC2Config versions 3.18 or later

```
<configSections>
  <section name="aws" type="Amazon.AWSSection, AWSSDK.Core" />
</configSections>
```

3. Add the following `<aws>` element to the `Ec2Config.exe.config` file.

```
<aws>
  <proxy
```

```
host="string value"  
port="string value"  
username="string value"  
password="string value" />  
</aws>
```

4. Save your changes.

Configure Proxy Settings Using the .Net System Element

You can specify proxy settings in a `<system.net>` element in the `Ec2Config.exe.config` file. For more information about the `<system.net>` element see [<defaultProxy> Element \(Network Settings\)](#) on MSDN.

To specify the `<system.net>` element in the `Ec2Config.exe.config` file

1. Edit the `Ec2Config.exe.config` file on an instance where you want the EC2Config service to communicate through a proxy. By default, the file is located in the following directory:
%ProgramFiles%\Amazon\Ec2ConfigService.
2. Add a `<defaultProxy>` entry to the `<system.net>` element. For more information about this element, see the [<defaultProxy> Element \(Network Settings\)](#) on MSDN.

For example, the following configuration routes all traffic to use the proxy that is currently configured for Internet Explorer, with the exception of the metadata and licensing traffic, which will bypass the proxy.

```
<defaultProxy>  
  <proxy usesystemdefault="true" />  
  <bypasslist>  
    <add address="169.254.169.250" />  
    <add address="169.254.169.251" />  
    <add address="169.254.169.254" />  
  </bypasslist>  
</defaultProxy>
```

3. Save your changes.

Configure Proxy Settings Using Microsoft Group Policy and Microsoft Internet Explorer

The EC2Config service runs under the Local System user account. You can specify instance-wide proxy settings for this account in Internet Explorer after you change Group Policy settings on the instance.

To configure proxy settings using Group Policy and Internet Explorer

1. On an instance where you want the EC2Config service to communicate through a proxy, open a Command prompt as an Administrator, type `gpedit.msc`, and press Enter.
2. In the Local Group Policy Editor, under **Local Computer Policy**, choose **Computer Configuration, Administrative Templates, Windows Components, Internet Explorer**.
3. In the right-pane, choose **Make proxy settings per-machine (rather than per-user)** and then choose **Edit policy setting**.
4. Choose **Enabled**, and then choose **Apply**.
5. Open Internet Explorer, and then choose the **Tools** button.
6. Choose **Internet Option**, and then choose the **Connections** tab.

7. Choose **LAN settings**.
8. Under **Proxy server**, choose the **Use a proxy server for your LAN** option.
9. Specify address and port information and then choose **OK**.

Managing the EC2Config Service

This section includes information to help you manage the EC2Config Service.

Contents

- [Installing the Latest Version of EC2Config \(p. 295\)](#)
- [Stopping, Restarting, Deleting, or Uninstalling EC2Config \(p. 305\)](#)
- [Subscribing to EC2Config Service Notifications \(p. 306\)](#)

Installing the Latest Version of EC2Config

By default, the EC2Config service is included in AWS Windows Server 2003-2012 R2 AMIs. (Windows Server 2016 AMIs use the EC2Launch PowerShell script. For more information about Windows Server 2016 AMIs, see [Changes in Windows Server 2016 AMIs \(p. 108\)](#).) When the EC2Config service is updated, all AWS Windows AMIs are updated with the latest version of the service. However, you need to update your own Windows AMIs and instances with the latest version.

For information about how to receive notifications for EC2Config updates, see [Subscribing to EC2Config Service Notifications \(p. 306\)](#). For information about the changes in each version, see the [Details about EC2Config Versions \(p. 296\)](#).

To verify the version of EC2Config included with your Windows AMI

1. Launch an instance from your AMI and connect to it.
2. In Control Panel, select **Programs and Features**.
3. In the list of installed programs, look for `Ec2ConfigService`. Its version number appears in the **Version** column.

To install the latest version of EC2Config on your instance

You can remotely install the latest version of the EC2Config service by using EC2 Run Command. For more information, see [Updating the SSM Agent Using Amazon EC2 Run Command \(p. 470\)](#). Or, use the following procedure to manually install the latest version.

1. If you changed EC2Config service settings, copy the `config.xml` file in the `%Program Files%\Amazon\Ec2ConfigService\Settings` directory. After you update the EC2Config service, you can paste this file into the `Settings` directory to retain your configuration changes.
2. [Download](#) and unzip the EC2Config installer.
3. Run `EC2Install.exe`. For a complete list of options, run `EC2Install` with the `/?` option. Note the following:
 - By default, the setup replaces your settings files with default settings files during installation and restarts the EC2Config service when the installation is completed. To keep the custom settings that you saved in step 1, run `EC2Install` with the `/norestart` option, restore your settings, and then restart the EC2Config service manually.
 - By default, the setup displays prompts. To run the command with no prompts, use the `/quiet` option.

Details about EC2Config Versions

Windows AMIs include an optional service called the EC2Config service (EC2Config.exe). EC2Config starts when the instance boots and performs tasks during startup and each time you stop or start the instance. EC2Config can also perform tasks on demand. Some of these tasks are automatically enabled, while others must be enabled manually. Although optional, this service provides access to advanced features that aren't otherwise available. This service runs in the LocalSystem account and performs tasks on the instance such as Windows Activation, setting the Administrator password, executing userdata, Cloud Formation Execution (requires [AWS CloudFormation executable](#)), writing to the AWS Console and one click sysprep from within the application. Its binaries and additional files are contained in the %ProgramFiles%\Amazon\EC2ConfigService directory. After you install the service, a log file is created in the %ProgramFiles%\Amazon\Ec2ConfigService\Logs\Ec2MsiInstall.txt directory.

Amazon Windows AMIs contain a service installed by Amazon Web Services; the EC2Config service. Although optional, this service provides access to advanced features that are not otherwise available.

Note

(Optional) If you have a version of EC2Config that is earlier than version 2.1.19 and you are trying to upgrade up to 2.2.12, you must first update to version 2.1.19, and then update to the current version. To update to version 2.1.19, download [EC2Install_2.1.19.zip](#), unzip the file, and then run EC2Install.exe. Please note this issue has been fixed in 2.3.313 version.

Requirements

.Net framework 3.5 SP1 or greater

You can receive notifications when new versions of the EC2Config service are released. For more information, see [Subscribing to EC2Config Service Notifications](#) (p. 306).

Version	Details
4.1.1378	New version of SSM Agent (2.0.558.0)
4.0.1343	<ul style="list-style-type: none"> • Run Command, SSM Config, the CloudWatch agent, and domain join support have been moved into another agent called SSM Agent. SSM Agent will be installed as part of the EC2Config upgrade. For more information, see EC2Config and Amazon EC2 Simple Systems Manager (SSM) (p. 285). • If you have a proxy set up in EC2Config, you will need to update your proxy settings for SSM Agent before upgrading. If you do not update the proxy settings, you will not be able to use Run Command to manage your instances. To avoid this, see the following information <i>before</i> updating to the newer version: Installing and Updating SSM Agent (p. 396). • If you previously enabled CloudWatch integration on your instances by using a local configuration file (AWS.EC2.Windows.CloudWatch.json), you will need to configure the file to work with SSM agent. For more information, see Using a Local Configuration File for CloudWatch Integration on Windows Server 2016 Instances (p. 109).
3.19.1153	<ul style="list-style-type: none"> • Re-enabled activation plugin for instances with old KMS configuration.

Amazon Elastic Compute Cloud
User Guide for Windows Instances
Using EC2Config

Version	Details
	<ul style="list-style-type: none"> Change default TRIM behavior to be disabled during disk format operation and added FormatWithTRIM for overriding InitializeDisks plugin with userdata.
3.18.1118	<ul style="list-style-type: none"> Fix to reliably add routes to the primary network adapter. Updates to improve support for AWS services.
3.17.1032	<ul style="list-style-type: none"> Fixes duplicate system logs appearing when filters set to same category. Fixes to prevent from hanging during disk initialization.
3.16.930	Added support to log "Window is Ready to use" event to Windows Event Log on start.
3.15.880	Fix to allow uploading run command output to S3 bucket names with '.' character.
3.14.786	<p>Added support to override InitializeDisks plugin settings. For example: To speed up SSD disk initialize, you can temporarily disable TRIM by specifying this in userdata:</p> <pre><InitializeDrivesSettings><SettingsGroup>FormatWithoutTRIM</SettingsGroup></InitializeDrivesSettings</pre>
3.13.727	SSM RunCommand - Fixes to process commands reliably after windows reboot.
3.12.649	<ul style="list-style-type: none"> Fix to gracefully handle reboot when running commands/scripts. Fix to reliably cancel running commands. Add support for (optionally) uploading MSI logs to S3 when installing applications via Run Command.
3.11.521	<ul style="list-style-type: none"> Fixes to enable RDP thumbprint generation for Windows 2003. Fixes to include timezone and UTC offset in the EC2Config log lines. SSM support to run commands in parallel. Roll back previous change to bring partitioned disks online.
3.10.442	<ul style="list-style-type: none"> Fix SSM (Simple Systems Manager) configuration failures when installing MSI applications. Fix to reliably bring storage disks online. Updates to improve support for AWS services.

Amazon Elastic Compute Cloud
User Guide for Windows Instances
Using EC2Config

Version	Details
3.9.359	<ul style="list-style-type: none"> • Fix in post Sysprep script to leave the configuration of windows update in a default state. • Fix the password generation plugin to improve the reliability in getting GPO password policy settings. • Restrict EC2Config/SSM log folder permissions to the local Administrators group. • Updates to improve support for AWS services.
3.8.294	<ul style="list-style-type: none"> • Fixed an issue with CloudWatch that prevented logs from getting uploaded when not on primary drive. • Improved the disk initialization process by adding retry logic. • Added improved error handling when the SetPassword plugin occasionally failed during AMI creation. • Updates to improve support for AWS services.
3.7.308	<ul style="list-style-type: none"> • Improvements to the ec2config-cli utility for config testing and troubleshooting within instance. • Avoid adding static routes for KMS and meta-data service on an OpenVPN adapter. • Fixed an issue where user-data execution was not honoring the "persist" tag. • Improved error handling when logging to the EC2 console is not available. • Updates to improve support for AWS services.
3.6.269	<ul style="list-style-type: none"> • Windows activation reliability fix to first use link local address 169.254.0.250/251 for activating windows via KMS • Improved proxy handling for SSM, Windows Activation and Domain Join scenarios • Fixed an issue where duplicate lines of user accounts were added to the Sysprep answer file
3.5.228	<ul style="list-style-type: none"> • Addressed a scenario where the CloudWatch plugin may consume excessive CPU and memory reading Windows Event Logs • Added a link to the CloudWatch configuration documentation in the EC2Config Settings UI
3.4.212	<ul style="list-style-type: none"> • Fixes to EC2Config when used in combination with VM-Import. • Fixed service naming issue in the WiX installer.

Version	Details
3.3.174	<ul style="list-style-type: none"> • Improved exception handling for ssm and domain join failures. • Change to support SSM schema versioning. • Fixed formatting ephemeral disks on Win2K3. • Change to support configuring disk size greater than 2TB. • Reduced virtual memory usage by setting GC mode to default. • Support for downloading artifacts from UNC path in aws:psModule and aws:application plugin. • Improved logging for Windows activation plugin.
3.2.97	<ul style="list-style-type: none"> • Performance improvements by delay loading SSM assemblies. • Improved exception handling for malformed sysprep2008.xml. • Command line support for SSM "Apply" configuration. • Change to support domain join when there is a pending computer rename. • Support for optional parameters in the aws:applications plugin. • Support for command array in aws:psModule plugin.
3.0.54	<ul style="list-style-type: none"> • Enable support for Amazon EC2 Simple Systems Manager (SSM). For more information see, Managing Windows Instance Configuration (p. 326) and Amazon EC2 Simple Systems Manager API Reference. • Automatically domain join EC2 Windows instances to an AWS directory via SSM. • Configure and upload CloudWatch logs/metrics via SSM. • Install PowerShell modules via SSM. • Install MSI applications via SSM.
2.4.233	<ul style="list-style-type: none"> • Added scheduled task to recover EC2Config from service startup failures. • Improvements to the Console log error messages. • Updates to improve support for AWS services.

Amazon Elastic Compute Cloud
User Guide for Windows Instances
Using EC2Config

Version	Details
2.3.313	<ul style="list-style-type: none"> • Fixed an issue with large memory consumption in some cases when the CloudWatch Logs feature is enabled. • Fixed an upgrade bug so that ec2config versions lower than 2.1.19 can now upgrade to latest. • Updated COM port opening exception to be more friendly and useful in logs. • Ec2configServiceSettings UI disabled resizing and fixed the attribution and version display placement in UI.
2.2.12	<ul style="list-style-type: none"> • Handled NullPointerException while querying a registry key for determining Windows Sysprep state which returned null occasionally. • Freed up unmanaged resources in finally block.
2.2.11	Fixed a issue in CloudWatch plugin for handling empty log lines.
2.2.10	<ul style="list-style-type: none"> • Removed configuring CloudWatch Logs settings through UI. • Enable users to define CloudWatch Logs settings in %ProgramFiles%\Amazon\Ec2ConfigService\Settings\AWS.EC2.Windows.CloudWatch.json file to allow future enhancements.
2.2.9	Fixed unhandled exception and added logging.
2.2.8	<ul style="list-style-type: none"> • Fixes Windows OS version check in EC2Config Installer to support Windows 2003 Sp1 and above. • Fixes null value handling when reading registry keys related to updating Sysprep config files.
2.2.7	<ul style="list-style-type: none"> • Added support for EC2Config to run during Sysprep execution for Windows 2008 and greater. • Improved exception handling and logging for better diagnostics
2.2.6	<ul style="list-style-type: none"> • Reduced the load on the instance and on CloudWatch Logs when uploading log events. • Addressed an upgrade issue where the CloudWatch Logs plug-in did not always stay enabled

Amazon Elastic Compute Cloud
User Guide for Windows Instances
Using EC2Config

Version	Details
2.2.5	<ul style="list-style-type: none"> • Added support to upload logs to CloudWatch Log Service. • Fixed a race condition issue in Ec2OutputRDPcert plug-in • Changed EC2Config Service recovery option to Restart from TakeNoAction • Added more exception information when EC2Config Crashes
2.2.4	<ul style="list-style-type: none"> • Fixed a typo in PostSysprep.cmd • Fixed the bug which EC2Config does not pin itself onto start menu for OS2012+
2.2.3	<ul style="list-style-type: none"> • Added option to install EC2Config without service starting immediately upon install. To use, run 'Ec2Install.exe start=false' from the command prompt • Added parameter in wallpaper plugin to control adding/removing wallpaper. To use, run 'Ec2WallpaperInfo.exe set' or 'Ec2WallpaperInfo.exe revert' from the command prompt • Added checking for RealTimeUniversal key, output incorrect settings of the RealTimeUniversal registry key to the Console • Removed EC2Config dependency on Windows temp folder • Removed UserData execution dependency on .Net 3.5
2.2.2	<ul style="list-style-type: none"> • Added check to service stop behavior to check that resources are being released • Fixed issue with long execution times when joined to domain
2.2.1	<ul style="list-style-type: none"> • Updated Installer to allow upgrades from older versions • Fixed Ec2WallpaperInfo bug in .Net4.5 only environment • Fixed intermittent driver detection bug • Added silent install option. Execute Ec2Install.exe with the '-q' option. eg: 'Ec2Install.exe -q'
2.2.0	<ul style="list-style-type: none"> • Added support for .Net4 and .Net4.5 only environments • Updated Installer

Version	Details
2.1.19	<ul style="list-style-type: none"> • Added ephemeral disk labeling support when using Intel network driver (eg. C3 instance Type). For more information, see Enhanced Networking on Windows (p. 737). • Added AMI Origin Version and AMI Origin Name support to the console output • Made changes to the Console Output for consistent formatting/parsing • Updated Help File
2.1.18	<ul style="list-style-type: none"> • Added EC2Config WMI Object for Completion notification (-Namespace root\Amazon -Class EC2_ConfigService) • Improved Performance of Startup WMI query with large Event Logs; could cause prolonged high CPU during initial execution
2.1.17	<ul style="list-style-type: none"> • Fixed UserData execution issue with Standard Output and Standard Error buffer filling • Fixed incorrect RDP thumbprint sometimes appearing in Console Output for >= w2k8 OS • Console Output now contains 'RDPCERTIFICATE-SubjectName:' for Windows 2008+, which contains the machine name value • Added D:\ to Drive Letter Mapping dropdown • Moved Help button to top right and changed look/feel • Added Feedback survey link to top right
2.1.16	<ul style="list-style-type: none"> • General Tab includes link to EC2Config download page for new Versions • Desktop Wallpaper overlay now stored in Users Local Appdata folder instead of My Documents to support MyDoc redirection • MSSQLServer name sync'd with system in Post-Sysprep script (2008+) • Reordered Application Folder (moved files to Plugin directory and removed duplicate files) • Changed System Log Output (Console): • *Moved to a date, name, value format for easier parsing (Please start migrating dependencies to new format) • *Added 'Ec2SetPassword' plugin status • *Added Sysprep Start and End times • Fixed issue of Ephemeral Disks not being labeled as 'Temporary Storage' for non-english Operating Systems • Fixed EC2Config Uninstall failure after running Sysprep

Version	Details
2.1.15	<ul style="list-style-type: none"> • Optimized requests to the Metadata service • Metadata now bypass Proxy Settings • Ephemeral Disks labeled as 'Temporary Storage' and Important.txt placed on volume when found (Citrix PV drivers only). For more information, see Upgrading PV Drivers on Your Windows AMI (p. 356). • Ephemeral Disks assigned drive letters from Z to A (Citrix PV drivers only) - assignment can be overwritten using Drive Letter Mapping plugin with Volume labels 'Temporary Storage X' where x is a number 0-25) • UserData now executes immediately following 'Windows is Ready'
2.1.14	Desktop wallpaper fixes
2.1.13	<ul style="list-style-type: none"> • Desktop wallpaper will display hostname by default • Removed dependency on Windows Time service • Route added in cases where multiple IPs are assigned to a single interface
2.1.11	<ul style="list-style-type: none"> • Changes made to Ec2Activation Plugin • -Verifies Activation status every 30 days • -If Grace Period has 90 days remaining (out of 180), reattempts activation
2.1.10	<ul style="list-style-type: none"> • Desktop wallpaper overlay no longer persists with Sysprep or Shutdown without Sysprep • Userdata option to execute on every service start with <persist>true</persist> • Changed location and name of / DisableWinUpdate.cmd to /Scripts/ PostSysprep.cmd • Administrator password set to not expire by default in /Scripts/PostSysprep.cmd • Uninstall will remove EC2Config PostSysprep script from c:\windows\setup\script \CommandComplete.cmd • Add Route supports custom interface metrics
2.1.9	UserData Execution no longer limited to 3851 Characters

Version	Details
2.1.7	<ul style="list-style-type: none"> • OS Version and language identifier written to console • EC2Config version written to console • PV driver version written to console • Detection of Bug Check and output to the console on next boot when found • Option added to config.xml to persist Sysprep credentials • Add Route Retry logic in cases of ENI being unavailable at start • User Data execution PID written to console • Minimum generated password length retrieved from GPO • Set service start to retry 3 attempts • Added S3_DownloadFile.ps1 and S3_Upload file.ps1 examples to /Scripts folder
2.1.6	<ul style="list-style-type: none"> • Version information added to General tab • Renamed the Bundle tab to Image • Simplified the process of specifying passwords and moved the password-related UI from the General tab to the Image tab • Renamed the Disk Settings tab to Storage • Added a Support tab with common tools for troubleshooting • Windows 2003 sysprep.ini set to extend OS partition by default • Added the private IP address to the wallpaper • Sysprep 2003 expand Root Volume • Private IP address displayed on wallpaper • Added retry logic for Console output • Fixed Com port exception for metadata accessibility -- caused EC2Config to terminate before console output is displayed • Checks for activation status on every boot -- activates as necessary • Fixed issue of relative paths -- caused when manually executing wallpaper shortcut from startup folder; pointing to Administrator/logs • Fixed default background color for Windows 2003 user (other than Administrator)

Version	Details
2.1.2	<ul style="list-style-type: none">• Console timestamps in UTC (Zulu)• Removed appearance of hyperlink on Sysprep tab• Addition of feature to dynamically expand Root Volume on first boot for Windows 2008+• When Set-Password is enabled, now automatically enables EC2Config to set the password• EC2Config checks activation status prior to running Sysprep (presents warning if not activated)• Windows 2003 Sysprep.xml now defaults to UTC timezone instead of Pacific• Randomized Activation Servers• Renamed Drive Mapping tab to Disk Settings• Moved Initialize Drives UI items from General to the Disk Settings tab• Help button now points to HTML help file• Updated HTML help file with changes• Updated 'Note' text for Drive Letter Mappings• Added InstallUpdates.ps1 to /Scripts folder for automating Patches and cleanup prior to Sysprep
2.1.0	<ul style="list-style-type: none">• Desktop wallpaper displays instance information by default upon first logon (not disconnect/reconnect)• PowerShell can be executed from the userdata by surrounding the code with <code><powershell></powershell></code>

Stopping, Restarting, Deleting, or Uninstalling EC2Config

You can manage the EC2Config service just as you would any other service.

To apply updated settings to your instance, you can stop and restart the service. If you're manually installing EC2Config, you must stop the service first.

To stop the EC2Config service

1. Launch and connect to your Windows instance.
2. On the **Start** menu, point to **Administrative Tools**, and then click **Services**.
3. In the list of services, right-click **EC2Config**, and select **Stop**.

To restart the EC2Config service

1. Launch and connect to your Windows instance.
2. On the **Start** menu, point to **Administrative Tools**, and then click **Services**.
3. In the list of services, right-click **EC2Config**, and select **Restart**.

If you don't need to update the configuration settings, create your own AMI, or use Amazon EC2 Simple Systems Manager (SSM), you can delete and uninstall the service. Deleting a service removes its registry subkey. Uninstalling a service removes the files, the registry subkey, and any shortcuts to the service.

To delete the EC2Config service

1. Start a command prompt window.
2. Run the following command:

```
C:\> sc delete ec2config
```

To uninstall EC2Config

1. Launch and connect to your Windows instance.
2. On the **Start** menu, click **Control Panel**.
3. Double-click **Programs and Features**.
4. On the list of programs, select **EC2ConfigService**, and click **Uninstall**.

Subscribing to EC2Config Service Notifications

Amazon SNS can notify you when new versions of the EC2Config service are released. Use the following procedure to subscribe to these notifications.

To subscribe to EC2Config notifications

1. Open the Amazon SNS console.
2. In the navigation bar, change the region to **US East (N. Virginia)**, if necessary. You must select this region because the SNS notifications that you are subscribing to were created in this region.
3. In the navigation pane, click **Subscriptions**.
4. Click **Create Subscription**.
5. In the **Create Subscription** dialog box, do the following:
 - a. In **TopicARN**, enter the following Amazon Resource Name (ARN):

```
arn:aws:sns:us-east-1:801119661308:ec2-windows-ec2config
```
 - b. In **Protocol**, select **Email**.
 - c. In **Endpoint**, enter an email address that you can use to receive the notifications.
 - d. Click **Subscribe**.
6. You'll receive a confirmation email with the subject line `EC2Config Interest`. Open the email and click **Confirm subscription** to complete your subscription.

Whenever a new version of the EC2Config service is released, we send notifications to subscribers. If you no longer want to receive these notifications, use the following procedure to unsubscribe.

To unsubscribe from EC2Config notifications

1. Open the Amazon SNS console.
2. In the navigation pane, click **Subscriptions**.
3. Select the subscription and then click **Delete Subscriptions**. When prompted for confirmation, click **Yes, Delete**.

Sending Performance Counters to CloudWatch and Logs to CloudWatch Logs Using EC2Config

Starting with EC2Config version 2.2.5 (version 2.2.6 or later is recommended), you can export all Windows Server messages in the system, security, application, and IIS logs to CloudWatch Logs and monitor them using CloudWatch metrics. EC2Config version 2.2.10 or later adds the ability to export any event log data, Event Tracing (Windows), or text-based log files to CloudWatch Logs. In addition, you can also export performance counter data to CloudWatch. Your Amazon EC2 instance must have outbound internet access in order to send log data to Amazon CloudWatch Logs. For more information about how to configure internet access, see [Internet Gateways](#) in the *Amazon VPC User Guide*. To manage the performance counters and logs for multiple instances, you can use Amazon EC2 Simple Systems Manager (SSM). For more information, see [Sending Performance Counters to CloudWatch and Logs to CloudWatch Logs Using Amazon EC2 Simple Systems Manager](#) (p. 340).

Note

Windows Server 2016 AMIs do not include the EC2Config service. You can send log data to CloudWatch by using the EC2Launch Windows PowerShell script. For more information, see [Changes in Windows Server 2016 AMIs](#) (p. 108).

To set up EC2Config to send data to CloudWatch Logs, complete the following steps:

Topics

- [Step 1: Configure IAM Permissions](#) (p. 307)
- [Step 2: Enable CloudWatch Logs Integration](#) (p. 308)
- [Step 3: Configure the Credentials for CloudWatch and CloudWatch Logs](#) (p. 309)
- [Step 4: Configure the Performance Counters and Logs to Send to CloudWatch and CloudWatch Logs](#) (p. 310)
- [Step 5: Configure the Flow Control](#) (p. 316)
- [Step 6: Restart EC2Config](#) (p. 316)
- [Troubleshooting CloudWatch Logs in EC2Config](#) (p. 316)

Step 1: Configure IAM Permissions

You can use the following IAM permissions in an instance profile attached to an Amazon EC2 instance when you launch the instance. EC2Config uses the instance profile when uploading CloudWatch metrics or logs to CloudWatch Logs. For more information about instance profiles, see [Instance Profiles](#) in the *IAM User Guide*. For more information about launching an instance with an IAM role, see [IAM Roles for Amazon EC2](#) (p. 658).

Note

These IAM permissions only work with the local JSON configuration file. If you want to upload logs through Amazon EC2 Simple Systems Manager (SSM), see [Sending Performance Counters to CloudWatch and Logs to CloudWatch Logs Using Amazon EC2 Simple Systems Manager](#) (p. 340).

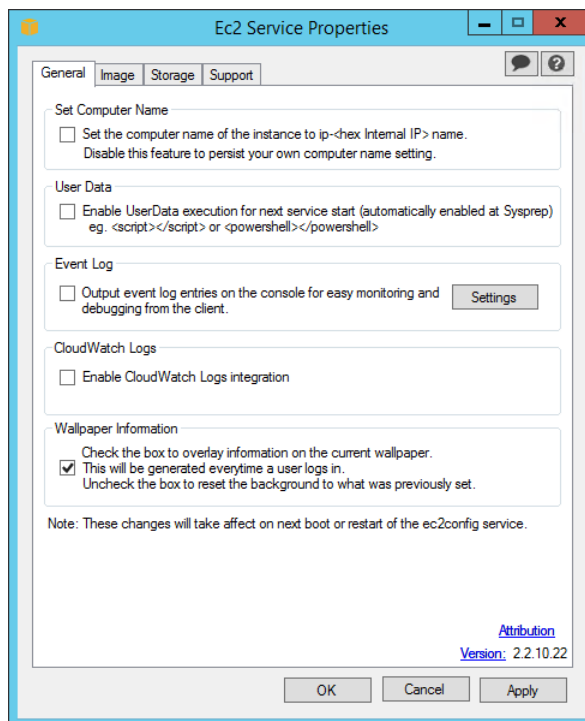
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowAccessToSSM",
      "Effect": "Allow",
      "Action": [
        "cloudwatch:PutMetricData",
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:DescribeLogGroups",

```

```
    "logs:DescribeLogStreams",  
    "logs:PutLogEvents"  
  ],  
  "Resource": [  
    "*" ]  
  }  
]
```

Step 2: Enable CloudWatch Logs Integration

1. Launch and connect to your Windows instance.
2. From the **Start** menu, click **All Programs**, and then click **EC2ConfigService Settings**.



3. On the **General** tab of the **Ec2 Service Properties** dialog box, under **CloudWatch Logs**, select **Enable CloudWatch Logs integration**, and then click **OK**.
4. Create a configuration file named **AWS.EC2.Windows.CloudWatch.json**.

To download a sample of the file, see [AWS.EC2.Windows.CloudWatch.json](#).

Note

You can also enable CloudWatch Logs by adding the following script to the user data field when you launch an instance. EC2Config will run this script every time your instance is restarted to make sure that CloudWatch Logs integration is enabled. To run this script only when an instance is first launched, remove `<persist>true</persist>` from the script.

```
<powershell>  
$EC2SettingsFile="C:\Program Files\Amazon\Ec2ConfigService\Settings  
\Config.xml"  
$xml = [xml](get-content $EC2SettingsFile)
```

```
$xmlElement = $xml.get_DocumentElement()
$xmlElementToModify = $xmlElement.Plugins

foreach ($element in $xmlElementToModify.Plugin)
{
    if ($element.name -eq "AWS.EC2.Windows.CloudWatch.PlugIn")
    {
        $element.State="Enabled"
    }
}
$xml.Save($EC2SettingsFile)
</powershell>
<persist>true</persist>
```

Step 3: Configure the Credentials for CloudWatch and CloudWatch Logs

To set the credentials, region, and metric namespace for CloudWatch

This section of the JSON file defines the credentials, region, and metric namespace that comprise the destination where your data is sent. You can add additional sections with unique IDs (for example, "CloudWatch2", "CloudWatch3", etc.) and specify a different region for each new ID to send the same data to different locations.

Note

You only need to set CloudWatch credentials if you are using EC2Config and plan to send performance counters to CloudWatch. If you're using Amazon EC2 Simple Systems Manager, your credentials are configured in the IAM role you used when you launched your Amazon EC2 instance.

1. In the JSON file, locate the **CloudWatch** section.

```
{
  "Id": "CloudWatch",
  "FullName":
  "AWS.EC2.Windows.CloudWatch.CloudWatch.CloudWatchOutputComponent,AWS.EC2.Windows.CloudW
  "Parameters": {
    "AccessKey": "",
    "SecretKey": "",
    "Region": "us-west-1",
    "NameSpace": "Windows/Default"
  }
},
```

2. In the **AccessKey** parameter, enter your access key ID. This is not supported if you launched your instance using an IAM role. For more information, see [IAM Roles for Amazon EC2 \(p. 658\)](#).
3. In the **SecretKey** parameter, enter your secret access key. This is not supported if you launched your instance using an IAM role. For more information, see [IAM Roles for Amazon EC2 \(p. 658\)](#).
4. In the **Region** parameter, enter the region where you want to send log data. You can specify us-east-1, us-west-1, us-west-2, eu-west-1, eu-central-1, ap-southeast-1, ap-southeast-2, or ap-northeast-1. Although you can send performance counters to a different region from where you send your log data, we recommend that you set this parameter to the same region where your instance is running.
5. In the **NameSpace** parameter, enter the metric namespace where you want performance counter data to be written in CloudWatch.

To set the credentials, region, log group, and log stream for CloudWatch Logs

This section of the JSON file defines the credentials, region, log group name and log stream namespace that comprise the destination where your data is sent. You can add additional sections with unique IDs (for example, "CloudWatchLogs2", "CloudWatchLogs3", etc.) and specify a different region for each new ID to send the same data to different locations.

1. In the JSON file, locate the **CloudWatchLogs** section.

```
{
  "Id": "CloudWatchLogs",
  "FullName":
  "AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
  "Parameters": {
    "AccessKey": "",
    "SecretKey": "",
    "Region": "us-east-1",
    "LogGroup": "Default-Log-Group",
    "LogStream": "{instance_id}"
  }
},
```

2. In the **AccessKey** parameter, enter your access key ID. This is not supported if you launched your instance using an IAM role. For more information, see [IAM Roles for Amazon EC2 \(p. 658\)](#).
3. In the **SecretKey** parameter, enter your secret access key. This is not supported if you launched your instance using an IAM role. For more information, see [IAM Roles for Amazon EC2 \(p. 658\)](#).
4. In the **Region** parameter, enter the region where you want EC2Config to send log data. You can specify us-east-1, us-west-1, us-west-2, eu-west-1, eu-central-1, ap-southeast-1, ap-southeast-2, or ap-northeast-1.
5. In the **LogGroup** parameter, enter the name for your log group. This is the same name that will be displayed on the **Log Groups** screen in the CloudWatch console.
6. In the **LogStream** parameter, enter the destination log stream. If you use **{instance_id}**, the default, EC2Config uses the instance ID of this instance as the log stream name.

If you enter a log stream name that doesn't already exist, CloudWatch Logs automatically creates it for you. You can use a literal string or predefined variables (**{instance_id}**, **{hostname}**, **{ip_address}**), or a combination of all three to define a log stream name.

The log stream name specified in this parameter appears on the **Log Groups > Streams for <YourLogStream>** screen in the CloudWatch console.

Step 4: Configure the Performance Counters and Logs to Send to CloudWatch and CloudWatch Logs

To configure the performance counters to send to CloudWatch

You can select any performance counters that are available in Performance Monitor. You can select different categories to upload to CloudWatch as metrics, such as .NET CLR Data, ASP.NET Applications, HTTP Service, Memory, or Process and Processors.

For each performance counter that you want to upload to CloudWatch, copy the **PerformanceCounter** section and change the **Id** parameter to make it unique (e.g., "PerformanceCounter2") and update the other parameters as necessary.

1. In the JSON file, locate the **PerformanceCounter** section.

```
{
```

```
"Id": "PerformanceCounter",  
  "FullName":  
"AWS.EC2.Windows.CloudWatch.PerformanceCounterComponent.PerformanceCounterInputComponent",  
  "Parameters": {  
    "CategoryName": "Memory",  
    "CounterName": "Available MBytes",  
    "InstanceName": "",  
    "MetricName": "AvailableMemory",  
    "Unit": "Megabytes",  
    "DimensionName": "",  
    "DimensionValue": ""  
  }  
},
```

2. In the **CategoryName** parameter, enter the performance counter category.
 - a. To find the available categories and counters, open Performance Monitor.
 - b. Click **Monitoring Tools**, and then click **Performance Monitor**.
 - c. In the results pane, click the green + (plus) button.

The categories and counters are listed in the **Add Counters** dialog box.

3. In the **CounterName** parameter, enter the name of the performance counter.
4. In the **InstanceName** parameter, enter valutes from the **Add Counters** dialog box in Performance Monitor, which can be one of the following:
 - Blank, if the selected object has no instances.
 - A single instance of the selected object.
 - **_Total** to use the aggregate of all instances.

Note

Do not use an asterisk (*) to indicate all instances because each performance counter component only supports one metric.

5. In the **MetricName** parameter, enter the CloudWatch metric that you want performance data to appear under.
6. In the **Unit** parameter, enter the appropriate unit of measure for the metric:

Seconds | Microseconds | Milliseconds | Bytes | Kilobytes | Megabytes | Gigabytes | Terabytes | Bits | Kilobits | Megabits | Gigabits | Terabits | Percent | Count | Bytes/Second | Kilobytes/Second | Megabytes/Second | Gigabytes/Second | Terabytes/Second | Bits/Second | Kilobits/Second | Megabits/Second | Gigabits/Second | Terabits/Second | Count/Second | None.

7. (optional) You can enter a dimension name and value in the **DimensionName** and **DimensionValue** parameters to specify a dimension for your metric. These parameters provide another view when listing metrics. You can also use the same dimension for multiple metrics so that you can view all metrics belonging to a specific dimension.

To send Windows application event log data to CloudWatch Logs

1. In the JSON file, locate the **ApplicationEventLog** section.

```
{  
  "Id": "ApplicationEventLog",  
  "FullName":  
"AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",  
  "Parameters": {  
    "LogName": "Application",
```

```
        "Levels": "1"  
    }  
},
```

2. In the **Levels** parameter, enter one of the following values:

- 1 - Only error messages uploaded.
- 2 - Only warning messages uploaded.
- 4 - Only information messages uploaded.

You can add values together to include more than one type of message. For example, **3** means that error messages (**1**) and warning messages (**2**) get uploaded. A value of **7** means that error messages (**1**), warning messages (**2**), and information messages (**4**) get uploaded.

To send security log data to CloudWatch Logs

1. In the JSON file, locate the **SecurityEventLog** section.

```
{  
  "Id": "SecurityEventLog",  
  "FullName":  
    "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",  
  "Parameters": {  
    "LogName": "Security",  
    "Levels": "7"  
  }  
},
```

2. In the **Levels** parameter, enter **7**, so that all messages are uploaded.

To send system event log data to CloudWatch Logs

1. In the JSON file, locate the **SystemEventLog** section.

```
{  
  "Id": "SystemEventLog",  
  "FullName":  
    "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",  
  "Parameters": {  
    "LogName": "System",  
    "Levels": "7"  
  }  
},
```

2. In the **Levels** parameter, enter one of the following values:

- 1 - Only error messages uploaded.
- 2 - Only warning messages uploaded.
- 4 - Only information messages uploaded.

You can add values together to include more than one type of message. For example, **3** means that error messages (**1**) and warning messages (**2**) get uploaded. A value of **7** means that error messages (**1**), warning messages (**2**), and information messages (**4**) get uploaded.

To send other types of event log data to CloudWatch Logs

In addition to the application, system, and security logs, you can upload other types of event logs.

1. In the JSON file, add a new section.

```
{
  "Id": "",
  "FullName":
  "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
  "Parameters": {
    "LogName": "",
    "Levels": "7"
  }
},
```

2. In the **Id** parameter, enter a name for the log you want to upload (e.g., WindowsBackup).
3. In the **LogName** parameter, enter the name of the log you want to upload.
 - a. To find the name of the log, in Event Viewer, in the navigation pane, click **Applications and Services Logs**.
 - b. In the list of logs, right-click the log you want to upload (e.g., Microsoft>Windows>Backup>Operational), and then click **Create Custom View**.
 - c. In the **Create Custom View** dialog box, click the **XML** tab. The **LogName** is in the <Select Path=> tag (e.g., Microsoft-Windows-Backup). Copy this text into the **LogName** parameter in the **AWS.EC2.Windows.CloudWatch.json** file.
4. In the **Levels** parameter, enter one of the following values:
 - 1 - Only error messages uploaded.
 - 2 - Only warning messages uploaded.
 - 4 - Only information messages uploaded.

You can add values together to include more than one type of message. For example, **3** means that error messages (**1**) and warning messages (**2**) get uploaded. A value of **7** means that error messages (**1**), warning messages (**2**), and information messages (**4**) get uploaded.

To send Event Tracing (Windows) data to CloudWatch Logs

ETW (Event Tracing for Windows) provides an efficient and detailed logging mechanism that applications can write logs to. Each ETW is controlled by a session manager that can start and stop the logging session. Each session has a provider and one or more consumers.

1. In the JSON file, locate the **ETW** section.

```
{
  "Id": "ETW",
  "FullName":
  "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
  "Parameters": {
    "LogName": "Microsoft-Windows-WinINet/Analytic",
    "Levels": "7"
  }
},
```

2. In the **LogName** parameter, enter the name of the log you want to upload.

- a. To find the name of the log, in Event Viewer, on the **View** menu, click **Show Analytic and Debug Logs**.
 - b. In the navigation pane, click **Applications and Services Logs**.
 - c. In the list of ETW logs, right-click the log you want to upload, and then click **Enable Log**.
 - d. Right-click the log again, and click **Create Custom View**.
 - e. In the **Create Custom View** dialog box, click the **XML** tab. The **LogName** is in the <Select Path=> tag (e.g., `Microsoft-Windows-WinINet/Analytic`). Copy this text into the **LogName** parameter in the **AWS.EC2.Windows.CloudWatch.json** file.
3. In the **Levels** parameter, enter one of the following values:
 - 1 - Only error messages uploaded.
 - 2 - Only warning messages uploaded.
 - 4 - Only information messages uploaded.

You can add values together to include more than one type of message. For example, **3** means that error messages (**1**) and warning messages (**2**) get uploaded. A value of **7** means that error messages (**1**), warning messages (**2**), and information messages (**4**) get uploaded.

To send custom logs (any text-based log file) to CloudWatch Logs

1. In the JSON file, locate the **CustomLogs** section.

```
{
  "Id": "CustomLogs",
  "FullName":
  "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch
  "Parameters": {
    "LogDirectoryPath": "C:\\CustomLogs\\",
    "TimestampFormat": "MM/dd/yyyy HH:mm:ss",
    "Encoding": "UTF-8",
    "Filter": "",
    "CultureName": "en-US",
    "TimeZoneKind": "Local",
    "LineCount": "5"
  }
},
```

2. In the **LogDirectoryPath** parameter, enter the path where logs are stored on your instance.
3. In the **TimestampFormat** parameter, enter the timestamp format you want to use. For a list of supported values, see the [Custom Date and Time Format Strings](#) topic on MSDN.

Important

Your source log file must have the timestamp at the beginning of each log line and there must be a space following the timestamp.

4. In the **Encoding** parameter, enter the file encoding to use (e.g., UTF-8). For a list of supported values, see the [Encoding Class](#) topic on MSDN.

Note

Use the encoding name, not the display name, as the value for this parameter.

5. (optional) In the **Filter** parameter, enter the prefix of log names. Leave this parameter blank to monitor all files. For a list of supported values, see the [FileSystemWatcherFilter Property](#) topic on MSDN.
6. (optional) In the **CultureName** parameter, enter the locale where the timestamp is logged. If **CultureName** is blank, it defaults to the same locale currently used by your Windows instance.

For a list of supported values, see the [National Language Support \(NLS\) API Reference](#) topic on MSDN.

Note

The **div**, **div-MV**, **hu**, and **hu-HU** values are not supported.

7. (optional) In the **TimeZoneKind** parameter, enter **Local** or **UTC**. You can set this to provide time zone information when no time zone information is included in your log's timestamp. If this parameter is left blank and if your timestamp doesn't include time zone information, CloudWatch Logs defaults to the local time zone. This parameter is ignored if your timestamp already contains time zone information.
8. (optional) In the **LineCount** parameter, enter the number of lines in the header to identify the log file. For example, IIS log files have virtually identical headers. You could enter **5**, which would read the first three lines of the log file's header to identify it. In IIS log files, the third line is the date and time stamp, but the time stamp is not always guaranteed to be different between log files. For this reason, we recommend including at least one line of actual log data for uniquely fingerprinting the log file.

To send IIS log data to CloudWatch Logs

1. In the JSON file, locate the **IISLog** section.

```
{
  "Id": "IISLogs",
  "FullName":
  "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWat
  "Parameters": {
    "LogDirectoryPath": "C:\\inetpub\\logs\\LogFiles\\W3SVC1",
    "TimestampFormat": "yyyy-MM-dd HH:mm:ss",
    "Encoding": "UTF-8",
    "Filter": "",
    "CultureName": "en-US",
    "TimeZoneKind": "UTC",
    "LineCount": "5"
  }
},
```

2. In the **LogDirectoryPath** parameter, enter the folder where IIS logs are stored for an individual site (e.g., **C:\\inetpub\\logs\\LogFiles\\W3SVC1**).

Note

Only W3C log format is supported. IIS, NCSA, and Custom formats are not supported.

3. In the **TimestampFormat** parameter, enter the timestamp format you want to use. For a list of supported values, see the [Custom Date and Time Format Strings](#) topic on MSDN.
4. In the **Encoding** parameter, enter the file encoding to use (e.g., UTF-8). For a list of supported values, see the [Encoding Class](#) topic on MSDN.

Note

Use the encoding name, not the display name, as the value for this parameter.

5. (optional) In the **Filter** parameter, enter the prefix of log names. Leave this parameter blank to monitor all files. For a list of supported values, see the [FileSystemWatcherFilter Property](#) topic on MSDN.
6. (optional) In the **CultureName** parameter, enter the locale where the timestamp is logged. If **CultureName** is blank, it defaults to the same locale currently used by your Windows instance. For a list of supported values, see the [National Language Support \(NLS\) API Reference](#) topic on MSDN.

Note

The **div**, **div-MV**, **hu**, and **hu-HU** values are not supported.

- (optional) In the **TimeZoneKind** parameter, enter **Local** or **UTC**. You can set this to provide time zone information when no time zone information is included in your log's timestamp. If this parameter is left blank and if your timestamp doesn't include time zone information, CloudWatch Logs defaults to the local time zone. This parameter is ignored if your timestamp already contains time zone information.
- (optional) In the **LineCount** parameter, enter the number of lines in the header to identify the log file. For example, IIS log files have virtually identical headers. You could enter **5**, which would read the first five lines of the log file's header to identify it. In IIS log files, the third line is the date and time stamp, but the time stamp is not always guaranteed to be different between log files. For this reason, we recommend including at least one line of actual log data for uniquely fingerprinting the log file.

Step 5: Configure the Flow Control

In order to send performance counter data to CloudWatch or to send log data to CloudWatch Logs, each data type must have a corresponding destination listed in the **Flows** section. For example, to send a performance counter defined in the **"Id": "PerformanceCounter"** section of the JSON file to the CloudWatch destination defined in the **"Id": "CloudWatch"** section of the JSON file, you would enter **"PerformanceCounter,CloudWatch"** in the **Flows** section. Similarly, to send the custom log, ETW log, and system log to CloudWatch Logs, you would enter **"(CustomLogs, ETW, SystemEventLog),CloudWatchLogs"**. In addition, you can send the same performance counter or log file to more than one destination. For example, to send the application log to two different destinations that you defined in the **"Id": "CloudWatchLogs"** section of the JSON file, you would enter **"ApplicationEventLog,(CloudWatchLogs, CloudWatchLogs2)"** in the **Flows** section.

- In the JSON file, locate the **Flows** section.

```
"Flows": {
  "Flows": [
    "PerformanceCounter,CloudWatch",
    "(PerformanceCounter,PerformanceCounter2), CloudWatch2",
    "(CustomLogs, ETW, SystemEventLog),CloudWatchLogs",
    "CustomLogs, CloudWatchLogs2",
    "ApplicationEventLog,(CloudWatchLogs, CloudWatchLogs2)"
  ]
}
```

- In the **Flows** parameter, enter each data type that you want to upload (e.g., ApplicationEventLog) and destination where you want to send it (e.g., CloudWatchLogs).

Step 6: Restart EC2Config

After you're finished updating the **C:\Program Files\Amazon\Ec2ConfigService\Settings\AWS.EC2.Windows.CloudWatch.json** file, you should restart EC2Config. For more information, see [Stopping, Restarting, Deleting, or Uninstalling EC2Config](#) (p. 305).

Troubleshooting CloudWatch Logs in EC2Config

If you're experiencing trouble with uploading performance counters or logs, the first place you should check is the **C:\Program Files\Amazon\Ec2ConfigService\Logs\Ec2ConfigLog.txt** file. Some of the most commonly encountered problems are listed below.

I cannot see logs in the CloudWatch console.

Please verify that you are using EC2Config version 2.2.6 or later. If you are still using EC2Config version 2.2.5, use the following steps to solve the issue:

1. In the Services Microsoft Management Console (MMC) snap-in, restart the EC2Config service. To open the **Services** snap-in, click the **Start** menu and then in the **Run** box, type **services.msc**.
2. Sign in to the AWS Management Console and open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
3. On the navigation bar, select the appropriate region.
4. In the navigation pane, click **Logs**.
5. In the contents pane, in the **Expire Events After** column, click the retention setting for the log group that you just created.
6. In the **Edit Retention** dialog box, in the **New Retention** list, select **10 years (3653 days)**, and then click **OK**.

Note

You can also set log retention (in days) using the following Windows PowerShell command:

```
Write-CWLRetentionPolicy-LogGroupName Default-Log-Group -  
RetentionInDays 3653
```

The Enable CloudWatch Logs integration check box won't stay selected after I click OK and then reopen EC2Config.

This issue might occur if you've performed an upgrade from an earlier version of EC2Config to version 2.2.5. To resolve this issue, install version 2.2.6 or later.

I see errors like *Log events cannot be more than 2 hours in the future or InvalidParameterException*.

This error might occur if you are using EC2Config version 2.2.5 and your instance's time zone falls between UTC-12:00 and UTC-02:00. To resolve this issue, install EC2Config version 2.2.6 or later.

I cannot see SQL Server logs in the CloudWatch console and see this error in *Ec2ConfigLog.txt* **[Error] Exception occurred: Index and length must refer to a location within the string. Parameter name: length.**

To resolve this issue, install EC2Config version 2.2.11 or later.

I'm running ten or fewer workflows and EC2Config is using over 500MB of memory.

To resolve this issue, install version 2.3.313 or later.

Only the first one or two IIS logs are uploaded and then no other IIS logs get uploaded.

Update the **IISlog** section of the **C:\Program Files\Amazon\Ec2ConfigService\Settings\AWS.EC2.Windows.CloudWatch.json** file and set the **LineCount** parameter to **3**, which would read the first three lines of the log file's header to identify it. In IIS log files, the third line is the date and time stamp, which is different between log files.

Troubleshooting Problems with the EC2Config Service

This section includes information to help you troubleshoot the EC2Config service.

Update EC2Config on an Unreachable Instance

Use the following procedure to update the EC2Config service on a Windows Server instance that is inaccessible using Remote Desktop.

To update EC2Config on an Amazon EBS-backed Windows instance that you can't connect to

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.

3. Locate the affected instance. Open the context (right-click) menu for the instance, choose **Instance State**, and then choose **Stop**.

Warning

When you stop an instance, the data on any instance store volumes is erased. Therefore, if you have any data on instance store volumes that you want to keep, be sure to back it up to persistent storage.

4. Choose **Launch Instance** and create a temporary `t2.micro` instance in the same Availability Zone as the affected instance. Use a Windows Server 2003 Amazon Machine Image (ami). If you use a later version of Windows Server, you won't be able to boot the original instance when you restore its root volume. To find an AMI for Windows Server 2003, search for public Windows AMIs with the name `Windows_Server-2003-R2_SP2`.

Important

If you do not create the instance in the same Availability Zone as the affected instance you will not be able to attach the root volume of the affected instance to the new instance.

5. In the EC2 console, choose **Volumes**.
6. Locate the root volume of the affected instance. **Detach** the volume and **attach** it to the temporary instance you created earlier. Attach it with the default device name (`xvdf`).
7. Use Remote Desktop to connect to the temporary instance, and then use the Disk Management utility to **make the volume available for use**.
8. **Download** the latest version of the EC2Config service. Extract the files from the `.zip` file to the `Temp` directory on the drive you attached.
9. On the temporary instance, open the Run dialog box, type `regedit`, and press Enter.
10. Choose `HKEY_LOCAL_MACHINE`. From the **File** menu, choose **Load Hive**. Choose the drive and then navigate to and open the following file: `Windows\System32\config\SOFTWARE`. When prompted, specify a key name.
11. Select the key you just loaded and navigate to `Microsoft\Windows\CurrentVersion`. Choose the `RunOnce` key. If this key doesn't exist, choose `CurrentVersion` from the context (right-click) menu, choose **New** and then choose **Key**. Name the key `RunOnce`.
12. From the context (right-click) menu choose the `RunOnce` key, choose **New** and then choose **String Value**. Enter `Ec2Install` as the name and `C:\Temp\Ec2Install.exe /quiet` as the data.
13. Choose the `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon` key. From the context (right-click) menu choose **New**, and then choose **String Value**. Enter `AutoAdminLogon` as the name and `1` as the value data.
14. Choose the `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon>` key. From the context (right-click) menu choose **New**, and then choose **String Value**. Enter `DefaultUserName` as the name and `Administrator` as the value data.
15. Choose the `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon` key. From the context (right-click) menu choose **New**, and then choose **String Value**. Enter `DefaultPassword` as the name and enter a password in the value data.
16. In the Registry Editor navigation pane, choose the temporary key that you created when you first opened the Registry Editor.
17. From the **File** menu, choose **Unload Hive**.
18. In the Disk Management Utility, choose the drive you attached earlier, open the context (right-click) menu, and choose **Offline**.
19. In the Amazon EC2 console, detach the affected volume from the temporary instance and reattach it to your instance with the device name `/dev/sda1`. You must specify this device name to designate the volume as a root volume.
20. **Start** the instance.
21. After the instance starts, check the system log and verify that you see the message `Windows is ready to use`.

22. Open Registry Editor and choose `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon`. Delete the String Value keys you created earlier: **AutoAdminLogon**, **DefaultUserName**, and **DefaultPassword**.
23. Delete or stop the temporary instance you created in this procedure.

Configuring a Windows Instance Using EC2Launch

To accommodate the change from .NET Framework to .NET Core, the EC2Config service has been deprecated on Windows Server 2016 AMIs and replaced by EC2Launch. EC2Launch is a bundle of Windows PowerShell scripts that perform many of the tasks performed by the EC2Config service.

For more information about Windows Server 2016, see [What's New with Windows Server 2016](#) and [Getting Started with Nano Server](#) on Microsoft.com.

Contents

- [Overview of EC2Launch \(p. 319\)](#)
- [Configuring EC2Launch \(p. 320\)](#)
- [Using Sysprep with EC2Launch \(p. 322\)](#)

Overview of EC2Launch

EC2Launch is a set of Windows PowerShell scripts that replaces the EC2Config service on Windows Server 2016 AMIs. EC2Launch performs the following tasks by default during the initial instance boot:

- Sets up new wallpaper that renders information about the instance. (Doesn't apply to Nano Server.)
- Sets the computer name.
- Sends instance information to the Amazon EC2 console.
- Sends the RDP certificate thumbprint to the EC2 console. (Doesn't apply to Nano Server.)
- Sets a random password for the administrator account.
- Adds DNS suffixes.
- Dynamically extends the operating system partition to include any unpartitioned space.
- Executes userdata (if specified).

The following tasks help to maintain backward compatibility with the EC2Config service. You can also configure EC2Launch to perform these tasks during startup:

- Initialize secondary EBS volumes.
- Send Windows Event logs to the EC2 console logs.
- Send the *Windows is ready to use* message to the EC2 console.

EC2Launch Directory Structure

EC2Launch is installed by default on Windows Server 2016 AMIs with the following root directory and sub-directories:

Note

By default, Windows hides files and folders under `C:\ProgramData`. To view EC2Launch directories and files, you must either type the path in Windows Explorer or change the folder properties to show hidden files and folders.

- **Root directory:** `C:\ProgramData\Amazon\EC2-Windows\Launch`
- **Scripts directory:** This directory includes the PowerShell scripts that make up EC2Launch.

- **Module directory:** This directory includes the Ec2Launch PowerShell module for building scripts related to Amazon EC2.
- **Config directory:** This directory includes the script configuration files that you can customize, as described later.
- **Sysprep directory:** This directory includes Sysprep resources.
- **Settings directory:** This directory includes an application for the Sysprep graphical user interface.
- **Logs directory:** This directory includes log files generated by scripts.

Configuring EC2Launch

After your instance has been initialized the first time, you can configure EC2Launch to run again and perform different startup tasks.

Configure Initialization Tasks

Enable or disable tasks in the LaunchConfig.json configuration file to change initialization tasks like the following:

- Set the computer name.
- Set up new wallpaper.
- Add DNS suffix list.
- Extend the boot volume size.
- Specify the administrator password.

Note

If you want to change the default setting for the administrator password, you must specify one of the following options.

- **Random:** EC2Launch generates a password, encrypts it with the user's key, and displays the encrypted password to the console.
- **Specify:** Specify a password that meets your system and organizational requirements. EC2Launch encrypts the password and sends it to the EC2 console so you can retrieve it later, if necessary.
- **Do Nothing:** Choose this option if you entered a password in an unattend.xml file for an unattended installation. If you are not using an unattend.xml file, choose one of the other options. If you choose this option and don't specify a password in an unattend.xml file, the system sets the password to match the password of the parent AMI.

To configure initialization settings

1. On the instance you want to configure, open the following file in a simple text editor.
C:\ProgramData\Amazon\EC2-Windows\Launch\Config\LaunchConfig.json
2. Type `true`, `false`, or a specific setting beside the tasks that you want to configure. For example:

```
{
  "setComputerName": false,
  "setWallpaper": true,
  "addDnsSuffixList": true,
  "extendBootVolumeSize": true,
  "adminPasswordType": "Random, Specify, Do Nothing",
  "adminPassword": "Password that adheres to your security policy."
}
```


Note

Enter a password only if you entered Specify for `adminPasswordType`.

3. Save your changes.
4. In Windows PowerShell, run the following command so that the system schedules the script to run as a Windows Scheduled Task.

```
C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeInstance.ps1 - Schedule
```

The script will execute only one time during the next boot and then disable these tasks from running again.

Initialize Drives and Drive Letter Mappings

Specify settings in the `DriveLetterMapping.json` file to initialize and format drives and map drive letters to EBS volumes on your EC2 instance. The script performs this operation if the drives have not already been initialized and partitioned.

To map drive letters to volumes

1. On the instance you want to configure, open the following file in a simple text editor.

```
C:\ProgramData\Amazon\EC2-Windows\Launch\Config\DriveLetterMapping.json
```

2. Specify the volume settings as in the following example:

```
{
  "driveLetterMapping": [
    {
      "volumeName": "Temporary Storage 0",
      "driveLetter": "H"
    }
  ]
}
```

3. Save your changes.
4. In Windows PowerShell, run the following command so that the system schedules the script to run as a Windows Scheduled Task.

```
C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeDisks.ps1
```

The script will execute once when the instance boots. If you need to initialize disks each time the instance starts (an option that is backwards compatible with EC2Config) run the following command:

```
C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeDisks.ps1 - Schedule
```

The script will execute *each* time the instance boots.

Note

You can also initialize attached disks at the instance launch by adding the following path to the PowerShell script in Amazon EC2 userdata.


```
<powershell>  
C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeDisks.ps1  
</powershell>
```

Send Windows Event Logs to the EC2 Console

Specify settings in the EventLogFilter.json configuration file to send Windows Event logs to EC2 console logs.

To configure settings to send Windows Event logs

1. On the instance you want to configure, open the following file in a simple text editor.

C:\ProgramData\Amazon\EC2-Windows\Launch\Config\EventLogFilter.json

2. Configure the log settings as in the following example:

```
{  
  "events": [  
    {  
      "logName": "System",  
      "source": "An event source (optional)",  
      "level": "Error",  
      "numEntries": 3  
    }  
  ]  
}
```

3. Save your changes.
4. In Windows PowerShell, run the following command so that the system schedules the script to run as a Windows Scheduled Task.

```
C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts  
\SendEventLogsToConsole.ps1 -Schedule
```

The logs can take three minutes or more to appear in the EC2 console logs. The script will execute *each* time the instance boots.

Send Windows Is Ready Message After A Successful Boot

The EC2Config service sent the *Windows is ready* message to the EC2 console after every boot. EC2Launch sends this message only after the initial boot. For backwards compatibility with the EC2Config service, you can schedule EC2Launch to send this message after every boot. On the instance you want to configure, open Windows PowerShell and run the following command. The system schedules the script to run as a Windows Scheduled Task.

```
C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\SendWindowsIsReady.ps1 -  
Schedule
```

The script will execute *each* time the instance boots.

Using Sysprep with EC2Launch

Sysprep simplifies the process of duplicating a customized installation of Windows Server 2016. EC2Launch offers a default answer file and batch files for Sysprep that automate and secure the

image-preparation process on your AMI. Modifying these files is optional. These files are located in the following directory, by default: C:\ProgramData\Amazon\EC2-Windows\Launch\Sysprep

Important

Sysprep is not supported on Windows Server 2016 Nano Server. Also, don't use Sysprep to create an instance backup. Sysprep removes system-specific information. If you remove this information there might be unintended consequences for an instance backup.

The EC2Launch answer file and batch files for Sysprep include the following:

- **Unattend.xml**: This is the default answer file. If you execute SysprepInstance.ps1 or choose **ShutdownWithSysprep** in the user interface, the system reads the setting from this file.
- **BeforeSysprep.cmd**: Customize this batch file to run commands before Ec2Launch executes Sysprep.
- **SysprepSpecialize.cmd**: Customize this batch file to run commands during the Sysprep specialize phase.

Running Sysprep with EC2Launch

On the full installation of Windows Server 2016 (with a desktop experience), you can run Sysprep with EC2Launch manually or by using the **EC2 Launch Settings** application.

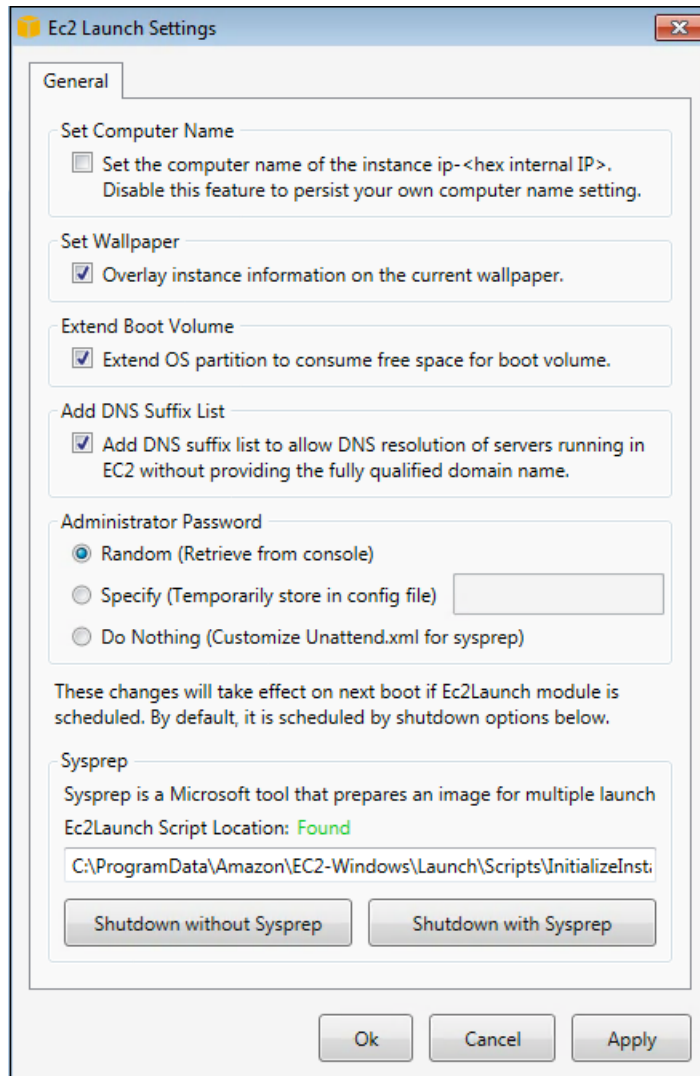
Note

Sysprep is not supported on the Nano Server installation.

Use one of the following procedures to create a standardized AMI using Sysprep and EC2Launch.

To run Sysprep using the EC2Launch Settings application

1. In the Amazon EC2 console locate or create a Windows Server 2016, Standard edition AMI that you want to duplicate.
2. Launch and connect to your Windows instance.
3. Customize it.
4. Search for and run the **EC2LaunchSettings** application. It is located in the following directory, by default: C:\ProgramData\Amazon\EC2-Windows\Launch\Settings.



5. Specify the desired options in the application. The options you specify configure the LaunchConfig.json file.
6. Select an option for the Administrator password.
 - **Random:** EC2Launch generates a password, encrypts it with the user's key, and displays the encrypted password to the console. The system disables this setting after the first launch so that this password persists if the instance is rebooted or stopped and started.
 - **Specify:** Specify a password that meets your system and organizational requirements. If you specify a password that doesn't meet the system requirements, the system will generate a random password. The password is stored in LaunchConfig.json file as clear text and is deleted once the password is set on the next boot. When Sysprep runs, it sets the administrator password. If you shut down now, the password is set immediately. When the service starts again, the administrator password is removed. You can retrieve the password from the EC2 console.
 - **Do Nothing:** Choose this option if you entered a password in an unattend.xml file for an unattended installation. If you are not using an unattend.xml file, choose one of the other options. If you choose this option and don't specify a password in an unattend.xml file, the system sets the password to match the password of the parent AMI.

For more information about administrator passwords and Sysprep unattend.xml files, see [AdministratorPassword](#).

Note

You can choose this option if you plan to choose **Shutdown without Sysprep** in the next step.

7. Choose **Shutdown with Sysprep** to begin creating a standardized AMI.

To manually run Sysprep using EC2Launch

1. In the Amazon EC2 console locate or create a Windows Server 2016, Standard edition AMI that you want to duplicate.
2. Launch and connect to your Windows instance.
3. Customize it.
4. Specify settings in the LaunchConfig.json file. The file is located in the following directory, by default: C:\ProgramData\Amazon\EC2-Windows\Launch\Config.

For **Administrator password**, choose one of the following:

- **Random:** EC2Launch generates a password, encrypts it with the user's key, and displays the encrypted password to the console. The system disables this setting after the first launch so that this password persists if the instance is rebooted or stopped and started.
- **Specify:** Specify a password that meets your system and organizational requirements. If you specify a password that doesn't meet the system requirements, the system will generate a random password. The password is stored in LaunchConfig.json file as clear text and is deleted once the password is set on the next boot. When Sysprep runs, it sets the administrator password. If you shut down now, the password is set immediately. When the service starts again, the administrator password is removed. You can retrieve the password from the EC2 console.
- **Do Nothing:** Choose this option if you entered a password in an unattend.xml file for an unattended installation. If you are not using an unattend.xml file, choose one of the other options. If you choose this option and don't specify a password in an unattend.xml file, the system sets the password to match the password of the parent AMI.

For more information about administrator passwords and Sysprep unattend.xml files, see [AdministratorPassword](#).

Note

You can choose this option if you plan to choose **Shutdown without Sysprep** in the next step.

5. Specify settings in the unattend.xml and other batch files, if you want. If plan to attended the installation, then you don't need to make changes in these files. The files are located in the following directory, by default: C:\ProgramData\Amazon\EC2-Windows\Launch\Sysprep.
6. In Windows PowerShell, run `./InitializeInstance.ps1 -Schedule`. The script is located in the following directory, by default: C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts. This script schedules the instance to initialize during the next boot. You must run this script before you execute the SysprepInstance.ps1 script in the next step.
7. In Windows PowerShell, run `./SysprepInstance.ps1`. The script is located in the following directory, by default: C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts.

You are logged off the instance, and the instance shuts down. If you check the **Instances** page in the Amazon EC2 console, the instance state changes from **running** to **stopping**, and then finally to **stopped**. At this point, it's safe to create an AMI from this instance.

Configuring a Windows Instance Using SSM Config

Amazon EC2 Simple Systems Manager (SSM) enables you to remotely manage the configuration of your Amazon EC2 instances, virtual machines (VMs), or servers in your on-premises environment or in an environment provided by other cloud providers using scripts, commands, or the Amazon EC2 console. SSM includes a lightweight instance configuration solution called *SSM Config* and an on-demand solution called *Amazon EC2 Run Command*. For more information about Run Command, see [Remote Management \(p. 437\)](#)

SSM Config helps you manage the configuration of your Windows instances while they're running. You create an *SSM document* that specifies the actions the system should perform on your instances, including which applications to install, which AWS Directory Service directory to join, which Microsoft PowerShell modules to install, etc. If an instance is missing one or more of these configurations, the system makes those changes. By default, the system checks every five minutes to see if there is a new configuration to apply. If so, the system updates the instances. In this way, you can remotely maintain a consistent configuration baseline on your instances. SSM Config is available using the AWS CLI or the AWS Tools for Windows PowerShell.

Contents

- [Managing Windows Instance Configuration \(p. 326\)](#)
- [Joining a Windows Instance to an AWS Directory Service Domain \(p. 331\)](#)
- [Sending Performance Counters to CloudWatch and Logs to CloudWatch Logs Using Amazon EC2 Simple Systems Manager \(p. 340\)](#)

Managing Windows Instance Configuration

The Amazon EC2 Simple Systems Manager (SSM) Config feature enables you to manage the configuration of your Windows instances while they are running. You create an *SSM document*, which describes configuration tasks (for example, installing software), and then associate the SSM document with one or more running Windows instances. The configuration agent on the instance processes the SSM document and configures the instance as specified.

If you disassociate an SSM document from an instance, this doesn't change the configuration of the instance. To change the configuration of an instance after you disassociate an SSM document, you must create a new SSM document that describes the configuration tasks (for example, uninstalling software), and then associate it with the instance.

To run scripts at instance launch only, consider using user data execution instead. For more information, see [Executing Scripts with User Data \(p. 274\)](#).

For more complex automation scenarios, consider using AWS CloudFormation or AWS OpsWorks instead. For more information, see the [AWS CloudFormation User Guide](#) or the [AWS OpsWorks User Guide](#).

Prerequisites

The EC2Config service processes SSM documents and configures the instance as specified. [Download](#) and install the latest version of the EC2Config service to each server you want to configure with SSM Config. For more information about how to install this service, see [Installing the Latest Version of EC2Config \(p. 295\)](#).

Limitations

- SSM Config is supported only for Windows instances.
- SSM Config is available in the following [regions](#).

To manage the configuration of your Windows instances using SSM Config, complete the following tasks.

Grant IAM Users Access to SSM Config

SSM documents run with administrative privilege on Windows instances because the EC2Config service runs in the Local System account. If a user has permission to execute any of the pre-defined SSM documents then that user also has administrator access to the instance. Delegate access to SSM Config and EC2 Run Command judiciously. This becomes extremely important if you create your own SSM documents. Amazon Web Services does not provide guidance about how to create secure SSM documents. You create SSM documents and delegate access to Run Command actions at your own risk. As a security best practice, we recommend that you create low-level SSM documents for low security tasks and delegate access to non-administrators.

Prepare the Instance

SSM Config and Run Command have the same limitations, prerequisites, and IAM permission requirements. Prepare your environment as described in [Systems Manager Prerequisites \(p. 394\)](#).

Create the JSON File

Open a text editor, add the JSON to describe the configuration, and then save the file with a `.json` file extension.

For more information about the structure of the JSON for an SSM document, see [SSM document](#) in the *Amazon EC2 Simple Systems Manager API Reference*.

Example: Install Applications

The following JSON describes applications to install on the instance. For each application, `source` is the URL of its `.msi` file.

```
{
  "schemaVersion": "1.0",
  "description": "Example instance configuration tasks",
  "runtimeConfig": {
    "aws:applications": {
      "properties": [
        {
          "action": "Install",
          "source": "http://dev.mysql.com/get/Downloads/MySQLInstaller/mysql-
installer-community-5.6.22.0.msi"
        },
        {
          "action": "Install",
          "source": "https://www.python.org/ftp/python/2.7.9/
python-2.7.9.msi"
        },
        {
          "action": "Install",
          "source": "http://download.winzip.com/winzip190-64.msi",
          "parameters": "INSTALLDIR=\"C:\\Program Files\\WinZipXX\""
        }
      ]
    }
  }
}
```

Example: Install PowerShell Modules and Run Commands

The following JSON describes PowerShell modules to install on your instance. For each module, source is the URL of the module and runCommand specifies the PowerShell command to run.

```
{
  "schemaVersion": "1.0",
  "description": "Example instance configuration tasks",
  "runtimeConfig": {
    "aws:psModule": {
      "properties": [
        {
          "description": "Example to install windows update PS module and
install all .NET 4 updates.",
          "source": "https://gallery.technet.microsoft.com/
scriptcenter/2d191bcd-3308-4edd-9de2-88dff796b0bc/file/41459/43/
PSWindowsUpdate.zip",
          "runCommand": "Get-WUInstall -ServiceID 9482f4b4-e343-43b6-
b170-9a65bc822c77 -Title \".NET Framework 4\" -AcceptAll"
        },
        {
          "description": "Example to install chocolatey package provider and
use it to install 7zip and GoogleChrome.",
          "runCommand": [
            "$url = 'https://chocolatey.org/install.ps1' " ,
            "iex ((new-object net.webclient).DownloadString($url))",
            "choco install -y 7zip",
            "choco install -y GoogleChrome"
          ]
        }
      ]
    }
  }
}
```

Example: Join an AWS Domain

For information about using SSM Config to join a Windows instance to a directory, see [Joining a Windows Instance to an AWS Directory Service Domain \(p. 331\)](#).

Example: Send Data to Amazon CloudWatch

For information about using SSM Config to send data to Amazon CloudWatch, see [Sending Performance Counters to CloudWatch and Logs to CloudWatch Logs Using Amazon EC2 Simple Systems Manager \(p. 340\)](#).

Create the SSM document

Use the AWS CLI or the Tools for Windows PowerShell to create a configuration document, specifying the JSON file that you created in the previous task.

AWS CLI

Use the following [create-document](#) command to name this configuration and make it available for use.

```
aws ssm create-document --content file://my-config.json --name "my-custom-
config"
```

Tools for Windows PowerShell

Use the following [New-SSMDocument](#) command to name this configuration and make it available for use.

```
$doc = Get-Content my-config.json | Out-String  
New-SSMDocument -Content $doc -Name "my-custom-config"
```

Associate the SSM document with the Instance

Use the AWS CLI or the Tools for Windows PowerShell to associate a configuration document with an instance. You'll specify the name of the configuration document that you created in the previous task. An instance can be associated with one configuration document at a time. If you associate a configuration document with an instance that already has an associated configuration document, the new configuration document replaces the existing configuration document.

AWS CLI

Use the following [create-association](#) command to associate your configuration document with your Windows instance.

```
aws ssm create-association --instance-id i-1a2b3c4d --name "my-custom-config"
```

Tools for Windows PowerShell

Use the following [New-SSMAssociation](#) command to associate your configuration document with your Windows instance.

```
New-SSMAssociation -InstanceId i-1a2b3c4d -Name "my-custom-config"
```

Manually Apply the Configuration

If you need to ensure that your instance is configured as specified in its current SSM document, you can run the `ec2config-cli` tool on your instance as follows:

```
ec2config-cli --apply-configuration
```

Alternatively, you can use Windows Task Scheduler to run `ec2config-cli` periodically to ensure that your instance maintains this configuration.

You can verify that `ec2config-cli` is installed by checking for it in the `C:\Program Files\Amazon\Ec2ConfigService` directory. If you do not have `ec2config-cli`, you can get it by installing the current version of the EC2Config service. For more information, see [Installing the Latest Version of EC2Config](#) (p. 295).

Disassociate the SSM document from the Instance

You can't update a configuration document after you create it. To associate a different configuration document with your instance, you can delete the existing association, and then associate a new configuration document with your instance. Note that terminating an instance does not automatically disassociate an associated configuration document.

AWS CLI

Use the following [delete-association](#) command to disassociate a configuration document from your Windows instance.


```
aws ssm delete-association --instance-id i-1a2b3c4d --name "my-custom-config"
```

Tools for Windows PowerShell

Use the following [Remove-SSMAssociation](#) command to disassociate a configuration document from your Windows instance.

```
Remove-SSMAssociation -InstanceId i-1a2b3c4d -Name "my-custom-config"
```

Delete the SSM document

When you are finished with a configuration document, you can delete it. You must disassociate the configuration document from any instances it is associated with before you can delete it.

AWS CLI

Use the following [delete-document](#) command to delete your configuration document.

```
aws ssm delete-document --name "my-custom-config"
```

Tools for Windows PowerShell

Use the following [Remove-SSMDocument](#) command to delete your configuration document.

```
Remove-SSMDocument -Name "my-custom-config"
```

Troubleshooting

This section includes information to help you troubleshoot problems with SSM Config.

Log4net Logging

The EC2Config service logs information in the following files using Apache log4net. The information in these files can help you troubleshoot problems.

- C:\Windows\System32\winevt\Logs\EC2ConfigService.evtx
- C:\Program Files\Amazon\Ec2ConfigService\Logs
- LocalSystem %LOCALAPPDATA%
 - **Windows Server 2008 or later**

```
C:\Windows\System32\config\systemprofile\AppData\Local\Amazon\Ec2Config\Logs  
  \Ec2ConfigPluginFramework.txt
```

- **Windows Server 2003**

```
C:\Documents and Settings\Default User\Local Settings\Amazon\Ec2Config\InstanceData\Logs  
  \Ec2ConfigPluginFramework.txt
```

You can enable extended logging by updating the log4net.config file. By default, the configuration file is located here:

```
C:\Program Files\Amazon\Ec2ConfigService\log4net.config
```

For more information about log4net configuration, see [Apache log4net Manual - Configuration](#). For examples of log4net configurations, see [Apache log4net Config Examples](#).

Windows Event Logs

The EC2Config service also logs information in a Windows Event log named *Ec2ConfigService*.

You can extract information from this event log to a log file by executing the following command from an elevated PowerShell command prompt:

```
Get-EventLog Ec2ConfigService | Sort-Object Index | Format-Table  
Message -AutoSize -Wrap | Out-File -Width 240 "C:\Program Files\Amazon  
\Ec2ConfigService\Logs\PluginFramework.txt"
```

If you want to log Windows Events to a log file with debugging enabled you must update the log4net.config file root element as follows: <root> <level value="DEBUG"/> <appender-ref ref="RollingFileAppender"/> <appender-ref ref="EventLogAppender"/> </root>

EC2 Console System Log

The following output in the EC2 console system log indicates that the EC2Config service was unable to connect to an SSM Config endpoint. These issues indicate problems with authorization and IAM role permissions, as noted in the following output messages:

```
Info: EC2Config configuration status:3;region:us-east-1;iam:0;authz:0 The  
output can  
  help you troubleshoot the cause of the failure: configuration status:3:  
The calls to (SSM)  
  failed. Ensure that you have granted the required IAM permissions to IAM  
users. (SSM) also  
  requires an Internet connection from your instance.
```

```
iam:0: The instance was not launched with an IAM role. You cannot download  
documents  
  if there is no IAM role/credentials associated with the instance.
```

```
authz:0: The instance is not authorized to access SSM. This happens if you  
launched  
  the instance without an IAM role, or if the role associated with your  
instance does not  
  have the necessary permissions to access the service.
```

You can troubleshoot specific reasons for an SSM document execution failure by checking the status of the association using the [describe-association](#) (AWS CLI) command or the [Get-SSMAssociation](#) (Tools for Windows PowerShell) command.

Joining a Windows Instance to an AWS Directory Service Domain

You can join an Amazon EC2 Windows instance to an active AWS Directory Service directory or AD Connector directory using Amazon EC2 Simple Systems Manager (SSM) Config. To perform this task with SSM Config, you use the AWS CLI or AWS Tools for Windows PowerShell to create an SSM document that specifies the domain join details, and then associate the SSM document with a running instance.

Alternatively, you can use the launch instance wizard in the Amazon EC2 console to launch an instance and specify the domain that you want to join. The wizard searches for any existing SSM documents for the domain in your account to associate with your instance; if it can't locate one, it creates an SSM document for you, and immediately associates it with your running instance.

Note

The Windows Server 2016 Nano server installation option (Nano Server) does not support online domain joining. You must perform an offline domain join instead. For more information, see [Offline Domain Join \(Djoin.exe\) Step-by-Step Guide](#) on Microsoft TechNet.

After you've associated the SSM document with your instance, you can connect to the instance using domain credentials you've defined in your AWS Directory Service directory.

There's no additional charge for using SSM Config or joining your instance to a domain. Standard charges for instance usage and AWS Directory Service usage apply.

For more information about SSM Config, see [Managing Windows Instance Configuration \(p. 326\)](#).

Contents

- [Limitations \(p. 332\)](#)
- [Prerequisites \(p. 332\)](#)
- [Joining a Domain Using the AWS CLI or AWS Tools for Windows PowerShell \(p. 333\)](#)
- [Joining a Domain Using the Amazon EC2 Launch Wizard \(p. 336\)](#)
- [Getting the Domain Join Status \(p. 337\)](#)
- [Connecting To Your Instance Using Domain Credentials \(p. 338\)](#)
- [Troubleshooting \(p. 338\)](#)
- [Viewing Information About Your Associations \(p. 339\)](#)
- [Changing an Association \(p. 340\)](#)
- [Deleting an SSM document \(p. 340\)](#)

Limitations

- SSM Config is supported only for Windows instances.
- SSM Config is available in the following [regions](#).

In other regions, you can manually join an instance to a domain. For more information, see [Joining an Instance to an AWS Directory Service Directory](#) in the *AWS Directory Service Administration Guide*.

Prerequisites

- To join a domain, ensure that you have the following resources available or configured in your AWS account:
 - An active AWS Directory Service directory. For more information about creating a directory, see [Getting Started with AWS Directory Service](#) in the *AWS Directory Service Administration Guide*.
 - To create a directory, you must have a VPC with two subnets. For more information about creating a VPC, see [What is Amazon VPC?](#) in the *Amazon VPC User Guide*. Instances that you join to the domain must be launched into the same VPC in which your domain is located.
 - A Windows instance that meets the requirements described in [Prepare the Instance \(p. 327\)](#).
 - An Internet connection for your instance, so that it can communicate with SSM Config. Ensure that you have a public subnet into which to launch your instance, and ensure that your instance has a public IP address. Alternatively, you can launch your instance into a private subnet without assigning it a public IP address, and use a NAT instance in a public subnet to initiate traffic to the Internet. For more information about NAT, see [NAT Instances](#) in the *Amazon VPC User Guide*.
- If you are using the AWS CLI or the AWS Tools for Windows PowerShell to create a configuration document, you need the following information:
 - The name and ID of the directory to join.

- The IP addresses of the DNS servers in the AWS Directory Service directory. For more information, see [Get the DNS Server Address](#) in the *AWS Directory Service Administration Guide*.

Configure Permissions for SSM

To join an instance to an AWS Directory Service Domain using SSM, you must configure permissions on the instance that will be joined to the domain and for any users who will use SSM. IAM managed policies for SSM can help you quickly configure access and permissions for users and instances. You can find these policies in the IAM console by searching for SSM, as shown in the following screen shot.



The managed policies perform the following functions:

- **AmazonEC2RoleForSSM (instance trust policy):** This policy enables the instance to communicate with the SSM Config API. You must assign this policy to the instance that you will join to the domain using SSM Config.

Note

You must assign the instance role in the launch wizard when you create the instance. You cannot assign the role after you create the instance. To assign the role to an existing instance:

1. Create an AMI from an existing instance.
 2. Launch a new instance from that AMI.
 3. Assign the instance role in the launch wizard when you create the new instance.
- **AmazonSSMFullAccess (user trust policy):** This policy gives a user access to the SSM Config API and SSM documents. To join an instance to a domain, your IAM account must be assigned either this policy or a comparable policy that you created. If delegating access to another user, assign this policy to administrators and trusted power users only.
 - **AmazonSSMReadOnlyAccess (user trust policy):** This policy gives a user access to read-only API actions such as Get and List. Users assigned this policy can't make changes on instances using SSM Config.

For information about how to configure these policies, see [Managed Policies and Inline Policies](#).

Joining a Domain Using the AWS CLI or AWS Tools for Windows PowerShell

To use the AWS CLI or the AWS Tools for Windows PowerShell to join a domain, you must create a configuration document, and then associate the SSM document with an already running instance.

To construct the SSM document, use a text editor of your choice, and save the file with the `*.json` extension. For more information about the structure of an SSM document, see [SSM documents](#) in the *Amazon EC2 Simple Systems Manager API Reference*.

SSM documents run with administrative privilege on Windows instances because the EC2Config service runs in the Local System account. If a user has permission to execute any of the pre-defined SSM documents then that user also has administrator access to the instance. Delegate access to SSM Config and Run Command judiciously. This becomes extremely important if you create your own SSM documents. Amazon Web Services does not provide guidance about how to create secure SSM documents. You create SSM documents and delegate access to Run Command actions at your own risk. As a security best practice, we recommend that you create low-level SSM documents for low security tasks and delegate access to non-administrators.

Use the following AWS CLI or AWS Tools for Windows PowerShell commands to create the SSM document, launch an instance, and then associate the file with your instance.

Action	AWS CLI	AWS Tools for Windows PowerShell
To create an SSM document in your account.	create-document	New-SSMDocument
To launch an instance. You can also join an existing instance to a domain, provided it meets the prerequisites. For more information, see Prerequisites (p. 332) .	run-instances	New-EC2Instance
To associate the SSM document with your instance.	create-association	New-SSMAssociation

To join a domain using the AWS CLI or AWS Tools for Windows PowerShell

1. Open a text editor on your computer, and write an SSM document. When you are done, save the file with a `.json` extension. The following is an example of an SSM document that allows instances to join domain `d-1234567890`:

```
{
  "schemaVersion": "1.0",
  "description": "Sample configuration to join an instance to a domain",
  "runtimeConfig": {
    "aws:domainJoin": {
      "properties": {
        "directoryId": "d-1234567890",
        "directoryName": "test.example.com",
        "dnsIpAddresses": [
          "198.51.100.1",
          "198.51.100.2"
        ]
      }
    }
  }
}
```

Note

If a valid organizational unit (OU) exists then you could also specify the following:

```
{
  "schemaVersion": "1.0",
  "description": "Sample configuration to join an instance to a domain",
  "runtimeConfig": {
    "aws:domainJoin": {
      "properties": {
        "directoryId": "d-1234567890",
        "directoryName": "test.example.com",
        "directoryOU":
          "OU=Computers,OU=example,DC=test,DC=example,DC=com",
      }
    }
  }
}
```

```
        "dnsIpAddresses": [
            "198.51.100.1",
            "198.51.100.2"
        ]
    }
}
```

2. Create the SSM document in your account, and give it a name. The name of the file must be between 1 and 64 characters in length.

AWS CLI

```
aws ssm create-document --content file://path/to/myconfigfile.json --name
  "My_Custom_Config_File"
```

Tools for Windows PowerShell

First create a variable that contains the file contents, and then create the document.

```
$doc = Get-Content C:\temp\myconfigfile.json | Out-String
New-SSMDocument -Content $doc -Name "My_Custom_Config_File"
```

3. Launch an EC2 instance into the same VPC in which your domain (d-1234567890) is located. You must assign an IAM role to your instance. You must also ensure that your instance has a public IP address, unless you're using a NAT instance for Internet communication. Take note of the instance ID in the output.

AWS CLI

```
aws ec2 run-instances --image-id ami-1a2b3c4d --subnet-id subnet-33cc44dd
  --key-name my-key-pair --instance-type m1.large --iam-instance-
  profile MyInstanceProfile --associate-public-ip-address

{
  "OwnerId": "123456789101",
  "ReservationId": "r-bbaa1122",
  "Groups": [
    {
      "GroupName": "default",
      "GroupId": "sg-5c5c5c5c"
    }
  ],
  "Instances": [
    ...
    "InstanceId": "i-1234567890abcdef0",
    ...
  ]
}
```

Tools for Windows PowerShell

```
New-EC2Instance -ImageId ami-1a2b3c4d -SubnetId subnet-33cc44dd
  -KeyName my-key-pair -InstanceType m1.large -
  InstanceProfile_Id MyInstanceProfile -associatePublicIp $true
```

4. Associate the SSM document with the running instance.

AWS CLI

```
aws ssm create-association --instance-id i-1234567890abcdef0 --name  
"My_Custom_Config_File"
```

Tools for Windows PowerShell

```
New-SSMAssociation -InstanceId i-1234567890abcdef0 -Name  
"My_Custom_Config_File"
```

5. Check the status of the domain join. For more information, see [Getting the Domain Join Status](#) (p. 337).

Joining a Domain Using the Amazon EC2 Launch Wizard

You can use the launch instance wizard in the Amazon EC2 console to join a new instance to a domain that you specify. If you don't already have one, the wizard creates an SSM document for you, and associates it with your new instance.

Note

You can't use the Amazon EC2 console to associate an SSM document with an existing instance.

To join a domain using the launch wizard

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the Amazon EC2 console, click **Launch Instance**.
3. On the first page of the wizard, select a Windows AMI. On the next page, select an instance type, and then click **Next: Configure Instance Details**.
4. On the **Step 3: Configure Instance Details** page, select a VPC from the **Network** list, and a subnet from the **Subnet** list. Ensure that you select the VPC in which your AWS Directory Service domain is located.
5. In the **Auto-assign Public IP** list, select **Enable** (if the subnet setting is not set to enable by default).

Note

If you're launching your instance into a private subnet and using a NAT instance in a public subnet for Internet communication, you do not have to assign your instance a public IP address.

6. Select your domain from the **Domain join directory** list, and select the IAM role to associate with the instance from the **IAM role** list.
7. Complete the rest of the configuration steps as required, and then click **Next** until you reach the **Step 6: Configure Security Group** page. Ensure that you select or create a security group with a rule that allows RDP access from your IP address, or from a range of IP addresses within your network. For more information about security group rules, see [Authorizing Inbound Traffic for Your Windows Instances](#) (p. 663).
8. Click **Review and Launch** to launch your instance.
9. Check the status of the domain join. For more information, see [Getting the Domain Join Status](#) (p. 337).

Getting the Domain Join Status

You can check the status of your domain join by viewing the system log for the instance, or by checking the status of the association.

Note

After a configuration file is associated with an instance, it may take several minutes before the instance is joined to the domain.

You can check your instance's system log by using the Amazon EC2 console, AWS CLI, or Tools for Windows PowerShell.

To get the system log using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, click **Instances**.
3. Select your instance, right-click, select **Instance Settings**, and then click **Get System Log**.

To get the system log using a command line tool

- Use the `get-console-output` (AWS CLI) command; for example:

```
aws ec2 get-console-output --instance-id i-1234567890abcdef0
```

- Use the `Get-EC2ConsoleOutput` (AWS Tools for Windows PowerShell) command; for example:

```
Get-EC2ConsoleOutput -instanceId i-1234567890abcdef0
```

In the system log, the following output indicates that the domain join was successful:

```
2015/02/02 10:59:36Z: Info: EC2Config configuration status:2;region:us-east-1;iam:1;authz:1
2015/02/02 10:59:42Z: Info: EC2Config: Downloading config
awsconfig_Domain_d-1234567890_corp.example.com
2015/02/02 10:59:45Z: Info: EC2Config: The instance is joining domain with
id:d-1234567890, name:corp.example.com ...
2015/02/02 10:59:48Z: Info: EC2Config: The instance successfully joined the
domain.
2015/02/02 10:59:48Z: Info: EC2Config: The instance will reboot shortly for
domain join to take effect.
```

Alternatively, you can check the status of the association between the configuration document and the instance by using the AWS CLI or the Tools for Windows PowerShell.

To check the status of the association

- Use the `describe-association` (AWS CLI) command; for example:

```
aws ssm describe-association --name "My_Custom_Config_File" --instance-id i-1234567890abcdef0
```

- Use the `Get-SSMAssociation` (Tools for Windows PowerShell) command; for example:

```
Get-SSMAssociation -Name "My_Custom_Config_File" -instanceId i-1234567890abcdef0
```


Connecting To Your Instance Using Domain Credentials

After you've joined your instance to a domain, you can connect to your instance using domain credentials that you've defined in AWS Directory Service.

To connect to an instance as an administrator using your directory credentials

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, click **Instances**, select your instance, and then click **Connect**.
3. In the dialog box, click **Download Remote Desktop File**, and open the file using an RDP client.
4. On the login screen, instead of using the local computer name and password generated from your key pair file, enter the details as follows:
 - **User name:** enter the fully-qualified name of your domain, followed by a backslash (\), and then the user name, in this case, `Admin`; for example: `corp.example.com\Admin`.
 - **Password:** enter the password that you specified when you created your domain.

For more information about connecting to an instance, see [Connecting to Your Windows Instance \(p. 254\)](#).

After you've verified that you can connect to your instance as an administrator, users in your domain can connect to the instance using the same procedure, replacing the `Admin` credentials with their own user name and password.

Troubleshooting

If you are having trouble joining your instance to a domain, or if you are having trouble connecting to your instance using domain credentials, first verify the status of the domain join by checking instance's system log, or by checking the status of the association: [Getting the Domain Join Status \(p. 337\)](#).

Cannot Connect to Instance

If the domain join was successful, but you are having trouble logging into to your instance, try the following:

- If you can connect to your instance, but you cannot log in, check that you are using the correct user name and password. The user name must include the fully qualified name of your domain (for example, `corp.example.com`), and the password must be the password configured in the domain, not the password generated by a key pair file.
- If you cannot connect to your instance, check your security group settings. You must have a rule that allows RDP access from your IP address or network.

The Domain Join was Unsuccessful

In the system log, the following output indicates the EC2Config service was unable to connect and download the associated SSM document, and therefore the domain join was unsuccessful:

```
Info: EC2Config configuration status:3;region:us-east-1;iam:0;authz:0
```

The output can help you troubleshoot the cause of the failure:

- `configuration status:3`: The calls to SSM Config failed. Ensure that you have granted the required IAM permissions to IAM users. SSM Config also requires an Internet connection from your instance - your instance must have a public IP address, and must be launched into a public subnet. For more information about public subnets, see [Your VPC With Subnets](#) in the Amazon VPC User Guide.

- `iam:0`: The instance was not launched with an IAM role. You cannot join your instance to a domain if there is no IAM role associated with the instance.
- `authz:0`: The instance is not authorized to access SSM Config. This happens if you launched the instance without an IAM role, or if the role associated with your instance does not have the necessary permissions to access the service.

You can also troubleshoot specific reasons for a domain join failure by checking the status of the association using the [describe-association](#) (AWS CLI) command or the [Get-SSMAssociation](#) (Tools for Windows PowerShell) command. For example, the following output indicates that the IAM role associated with the instance does not have permission to use the `ds:CreateComputer` action:

```
Name           : My_Config_Doc
InstanceId      : i-1234567890abcdef0
Date           : 2/10/2015 1:31:45 AM
Status.Name     : Failed
Status.Date    : 2/10/2015 1:38:38 AM
Status.Message  : RunId=631148a7-894f-4684-8718-ee4cexample,
                  status:Failed, code:0,
                  message:RuntimeStatusCounts=[Failed=1],
                  RuntimeStatus=[aws:domainJoin={Failed,User:
                  arn:aws:sts::123456789101:assumed-role/
NoDomainJoinPermission/i-1234567890abcdef0 is not authorized to
                  perform: ds:CreateComputer}]
Status.AdditionalInfo : {agent=EC2Config,ver=x.x.xx,osver=6.2.9200,os=Windows
                  Server 2012 Standard,lang=en-US}
```

Viewing Information About Your Associations

You can use the AWS CLI or the AWS Tools for Windows PowerShell to view information about your associations and your SSM documents.

Action	AWS CLI	AWS Tools for Windows PowerShell
To view information about an association for a specific instance and SSM document. You can also use this command to view the status of an association.	describe-association	Get-SSMAssociation
To view information about a specified SSM document. You can also use this command to view the status of an SSM document, for example, creating.	describe-document	Get-SSMDocumentDescription
To view the contents of a specified SSM document.	get-document	Get-SSMDocument
To view a list of associations for a specified SSM document or a specified instance.	list-associations	Get-SSMAssociationList
To view a list of your SSM documents.	list-documents	Get-SSMDocumentList

Changing an Association

You can't update an SSM document after you create it. If you want to join your instance to a new domain, you must first delete the association, and then create a new association using a new SSM document. It can take up to 15 minutes for the configuration changes to take effect.

For more information about deleting an association, see [Disassociate the SSM document from the Instance \(p. 329\)](#). For more information about associating a new document with an instance, see [Associate the SSM document with the Instance \(p. 329\)](#).

Deleting an association does not change the configuration on the instance. Your instance is still joined to a domain until you manually remove it from the domain by modifying the network connection configuration information and system properties of the instance.

Deleting an SSM document

If you no longer require an SSM document, you can delete it. You must first disassociate the file from any instances it is associated with before you delete it. For more information about deleting an SSM document, see [Delete the SSM document \(p. 330\)](#).

Sending Performance Counters to CloudWatch and Logs to CloudWatch Logs Using Amazon EC2 Simple Systems Manager

You can use Amazon EC2 Simple Systems Manager (SSM) Config to configure integration with Amazon CloudWatch and Amazon CloudWatch Logs on multiple instances to monitor their log files. You can send Windows Server messages in the application, system, security, and Event Tracing (Windows) logs to Amazon CloudWatch Logs. When you enable logging for the first time, SSM Config sends all logs generated within 1 minute from the time that you start uploading logs for the application, system, security, and ETW logs. Logs that occurred before this time are not included. If you disable logging and then later re-enable logging, SSM Config sends logs from where it left off. For any custom log files and Internet Information Services (IIS) logs, SSM Config reads the log files from the beginning. In addition, SSM Config can also send performance counter data to CloudWatch.

SSM Config enables you to manage the configuration of your Windows instances while they are running. You create a *configuration document*, which describes configuration tasks (for example, sending performance counters to CloudWatch and logs to CloudWatch Logs), and then associate the configuration document with one or more running Windows instances. The configuration agent on the instance processes the configuration document and configures the instance as specified.

Important

SSM documents run with administrative privilege on Windows instances because the EC2Config service runs in the Local System account. If a user has permission to execute any of the pre-defined SSM documents then that user also has administrator access to the instance. Delegate access to SSM Config and EC2 Run Command judiciously. This becomes extremely important if you create your own SSM documents. Amazon Web Services does not provide guidance about how to create secure SSM documents. You create SSM documents and delegate access to Run Command actions at your own risk. As a security best practice, we recommend that you create low-level SSM documents for low security tasks and delegate access to non-administrators.

If you previously enabled CloudWatch integration in EC2Config, the SSM Config settings override any settings stored locally on the instance in the **C:\Program Files\Amazon\Ec2ConfigService\Settings\AWS.EC2.Windows.CloudWatch.json** file. For more information about using EC2Config to manage performance counters and logs on single instance, see [Sending Performance Counters to CloudWatch and Logs to CloudWatch Logs Using EC2Config \(p. 307\)](#).

To manage the configuration of your Windows instances using SSM Config, complete the following tasks.

Tasks

- [Step 1: Prepare Your Environment \(p. 341\)](#)
- [Step 2: Create a JSON File \(p. 341\)](#)
- [Step 3: Configure the Region and Namespace for CloudWatch and CloudWatch Logs \(p. 343\)](#)
- [Step 4: Configure the Performance Counters and Logs to Send to CloudWatch and CloudWatch Logs \(p. 345\)](#)
- [Step 5: Configure the Flow Control \(p. 350\)](#)
- [Step 6: Create a Configuration Document \(p. 351\)](#)
- [Step 7: Associate the Configuration Document with the Instance \(p. 351\)](#)

Step 1: Prepare Your Environment

SSM Config and Run Command have the same limitations, prerequisites, and IAM permission requirements. Prepare your environment as described in [Systems Manager Prerequisites \(p. 394\)](#).

Step 2: Create a JSON File

If you don't already have a JSON file, you must create one. Open a text editor, add the JSON to describe the configuration, and then save the file with a `.json` file extension.

For more information about the structure of the JSON for a configuration document, see [Creating SSM Documents \(p. 478\)](#).

When using SSM Config you can only have one JSON file associated with your instance. Whether you create a new JSON file or you already have one associated with your instance, you'll need to add the following sections to it.

```
{
  "schemaVersion": "1.0",
  "description": "Example CloudWatch Logs tasks",
  "runtimeConfig": {
    "aws:cloudWatch": {
      "properties": {
        "EngineConfiguration": {
          "PollInterval": "00:00:15",
          "Components": [
            {
              "Id": "ApplicationEventLog",
              "FullName": "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent",
              "Parameters": {
                "LogName": "Application",
                "Levels": "value"
              }
            },
            {
              "Id": "SystemEventLog",
              "FullName": "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent",
              "Parameters": {
                "LogName": "System",
                "Levels": "value"
              }
            },
            {
              "Id": "SecurityEventLog",
```

```
"FullName": "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.Clo
  "Parameters": {
    "LogName": "Security",
    "Levels": "value"
  }
},
{
  "Id": "ETW",

"FullName": "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.Clo
  "Parameters": {
    "LogName": "Microsoft-Windows-WinINet/Analytic",
    "Levels": "value"
  }
},
{
  "Id": "IISLogs",

"FullName": "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.C
  "Parameters": {
    "LogDirectoryPath": "path",
    "TimestampFormat": "value",
    "Encoding": "value",
    "Filter": "value",
    "CultureName": "locale",
    "TimeZoneKind": "value",
    "LineCount": "value"
  }
},
{
  "Id": "CustomLogs",

"FullName": "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.C
  "Parameters": {
    "LogDirectoryPath": "path",
    "TimestampFormat": "value",
    "Encoding": "value",
    "Filter": "value",
    "CultureName": "locale",
    "TimeZoneKind": "value",
    "LineCount": "value"
  }
},
{
  "Id": "PerformanceCounter",

"FullName": "AWS.EC2.Windows.CloudWatch.PerformanceCounterComponent.PerformanceCounterInput
  "Parameters": {
    "CategoryName": "name",
    "CounterName": "name",
    "InstanceName": "name",
    "MetricName": "name",
    "Unit": "unit",
    "DimensionName": "name",
    "DimensionValue": "value"
  }
},
{
```

```
        "Id": "CloudWatchLogs",
    "FullName": "AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "AccessKey": "access-key-id",
        "SecretKey": "secret-access-key",
        "Region": "region",
        "LogGroup": "group",
        "LogStream": "stream"
    }
},
{
    "Id": "CloudWatch",
    "FullName": "AWS.EC2.Windows.CloudWatch.CloudWatch.CloudWatchOutputComponent,AWS.EC2.Window
    "Parameters": {
        "AccessKey": "access-key-id",
        "SecretKey": "secret-access-key",
        "Region": "region",
        "NameSpace": "namespace"
    }
},
],
"Flows": {
    "Flows": [
        "source,destination",
        "(source1, source2),destination",
        "source, (destination1,destination2)"
    ]
}
}
}
}
}
```

Step 3: Configure the Region and Namespace for CloudWatch and CloudWatch Logs

Next, you'll define the credentials, region, and metric namespace that comprise the destination where your data is sent.

To set the credentials, region, and metric namespace for CloudWatch

This section of the JSON file defines the credentials, region, and metric namespace that comprise the destination where your data is sent. You can add additional sections with unique IDs (for example, "CloudWatch2", "CloudWatch3", etc.) and specify a different region for each new ID to send the same data to different locations.

Note

You only need to set CloudWatch credentials if you are using EC2Config and plan to send performance counters to CloudWatch. If you're using Amazon EC2 Simple Systems Manager, your credentials are configured in the IAM role you used when you launched your Amazon EC2 instance.

1. In the JSON file, locate the **CloudWatch** section.

```
{
```

```
    "Id": "CloudWatch",  
    "FullName":  
    "AWS.EC2.Windows.CloudWatch.CloudWatch.CloudWatchOutputComponent,AWS.EC2.Windows.CloudW  
    "Parameters": {  
        "AccessKey": "",  
        "SecretKey": "",  
        "Region": "us-west-1",  
        "NameSpace": "Windows/Default"  
    }  
},
```

2. In the **AccessKey** parameter, enter your access key ID. This is not supported if you launched your instance using an IAM role. For more information, see [IAM Roles for Amazon EC2 \(p. 658\)](#).
3. In the **SecretKey** parameter, enter your secret access key. This is not supported if you launched your instance using an IAM role. For more information, see [IAM Roles for Amazon EC2 \(p. 658\)](#).
4. In the **Region** parameter, enter the region where you want to send log data. You can specify us-east-1, us-west-1, us-west-2, eu-west-1, eu-central-1, ap-southeast-1, ap-southeast-2, or ap-northeast-1. Although you can send performance counters to a different region from where you send your log data, we recommend that you set this parameter to the same region where your instance is running.
5. In the **NameSpace** parameter, enter the metric namespace where you want performance counter data to be written in CloudWatch.

To set the credentials, region, log group, and log stream for CloudWatch Logs

This section of the JSON file defines the credentials, region, log group name and log stream namespace that comprise the destination where your data is sent. You can add additional sections with unique IDs (for example, "CloudWatchLogs2", "CloudWatchLogs3", etc.) and specify a different region for each new ID to send the same data to different locations.

1. In the JSON file, locate the **CloudWatchLogs** section.

```
{  
    "Id": "CloudWatchLogs",  
    "FullName":  
    "AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",  
    "Parameters": {  
        "AccessKey": "",  
        "SecretKey": "",  
        "Region": "us-east-1",  
        "LogGroup": "Default-Log-Group",  
        "LogStream": "{instance_id}"  
    }  
},
```

2. In the **AccessKey** parameter, enter your access key ID. This is not supported if you launched your instance using an IAM role. For more information, see [IAM Roles for Amazon EC2 \(p. 658\)](#).
3. In the **SecretKey** parameter, enter your secret access key. This is not supported if you launched your instance using an IAM role. For more information, see [IAM Roles for Amazon EC2 \(p. 658\)](#).
4. In the **Region** parameter, enter the region where you want EC2Config to send log data. You can specify us-east-1, us-west-1, us-west-2, eu-west-1, eu-central-1, ap-southeast-1, ap-southeast-2, or ap-northeast-1.
5. In the **LogGroup** parameter, enter the name for your log group. This is the same name that will be displayed on the **Log Groups** screen in the CloudWatch console.
6. In the **LogStream** parameter, enter the destination log stream. If you use **{instance_id}**, the default, EC2Config uses the instance ID of this instance as the log stream name.

If you enter a log stream name that doesn't already exist, CloudWatch Logs automatically creates it for you. You can use a literal string or predefined variables (**{instance_id}**, **{hostname}**, **{ip_address}**), or a combination of all three to define a log stream name.

The log stream name specified in this parameter appears on the **Log Groups > Streams for <YourLogStream>** screen in the CloudWatch console.

Step 4: Configure the Performance Counters and Logs to Send to CloudWatch and CloudWatch Logs

Next, you'll configure the performance counters and logs that you want to send to CloudWatch and CloudWatch Logs.

To configure the performance counters to send to CloudWatch

You can select any performance counters that are available in Performance Monitor. You can select different categories to upload to CloudWatch as metrics, such as .NET CLR Data, ASP.NET Applications, HTTP Service, Memory, or Process and Processors.

For each performance counter that you want to upload to CloudWatch, copy the **PerformanceCounter** section and change the **Id** parameter to make it unique (e.g., "PerformanceCounter2") and update the other parameters as necessary.

1. In the JSON file, locate the **PerformanceCounter** section.

```
{
  "Id": "PerformanceCounter",
  "FullName":
  "AWS.EC2.Windows.CloudWatch.PerformanceCounterComponent.PerformanceCounterInputComponent",
  "Parameters": {
    "CategoryName": "Memory",
    "CounterName": "Available MBytes",
    "InstanceName": "",
    "MetricName": "AvailableMemory",
    "Unit": "Megabytes",
    "DimensionName": "",
    "DimensionValue": ""
  }
},
```

2. In the **CategoryName** parameter, enter the performance counter category.
 - a. To find the available categories and counters, open Performance Monitor.
 - b. Click **Monitoring Tools**, and then click **Performance Monitor**.
 - c. In the results pane, click the green **+** (plus) button.

The categories and counters are listed in the **Add Counters** dialog box.

3. In the **CounterName** parameter, enter the name of the performance counter.
4. In the **InstanceName** parameter, enter values from the **Add Counters** dialog box in Performance Monitor, which can be one of the following:
 - Blank, if the selected object has no instances.
 - A single instance of the selected object.
 - **_Total** to use the aggregate of all instances.

Note

Do not use an asterisk (*) to indicate all instances because each performance counter component only supports one metric.

5. In the **MetricName** parameter, enter the CloudWatch metric that you want performance data to appear under.
6. In the **Unit** parameter, enter the appropriate unit of measure for the metric:

Seconds | Microseconds | Milliseconds | Bytes | Kilobytes | Megabytes | Gigabytes | Terabytes | Bits | Kilobits | Megabits | Gigabits | Terabits | Percent | Count | Bytes/Second | Kilobytes/Second | Megabytes/Second | Gigabytes/Second | Terabytes/Second | Bits/Second | Kilobits/Second | Megabits/Second | Gigabits/Second | Terabits/Second | Count/Second | None.
7. (optional) You can enter a dimension name and value in the **DimensionName** and **DimensionValue** parameters to specify a dimension for your metric. These parameters provide another view when listing metrics. You can also use the same dimension for multiple metrics so that you can view all metrics belonging to a specific dimension.

To send Windows application event log data to CloudWatch Logs

1. In the JSON file, locate the **ApplicationEventLog** section.

```
{
  "Id": "ApplicationEventLog",
  "FullName":
  "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
  "Parameters": {
    "LogName": "Application",
    "Levels": "1"
  }
},
```

2. In the **Levels** parameter, enter one of the following values:

- 1 - Only error messages uploaded.
- 2 - Only warning messages uploaded.
- 4 - Only information messages uploaded.

You can add values together to include more than one type of message. For example, **3** means that error messages (**1**) and warning messages (**2**) get uploaded. A value of **7** means that error messages (**1**), warning messages (**2**), and information messages (**4**) get uploaded.

To send security log data to CloudWatch Logs

1. In the JSON file, locate the **SecurityEventLog** section.

```
{
  "Id": "SecurityEventLog",
  "FullName":
  "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
  "Parameters": {
    "LogName": "Security",
    "Levels": "7"
  }
}
```

```
},
```

2. In the **Levels** parameter, enter **7**, so that all messages are uploaded.

To send system event log data to CloudWatch Logs

1. In the JSON file, locate the **SystemEventLog** section.

```
{  
  "Id": "SystemEventLog",  
  "FullName":  
    "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",  
  "Parameters": {  
    "LogName": "System",  
    "Levels": "7"  
  }  
},
```

2. In the **Levels** parameter, enter one of the following values:

- 1 - Only error messages uploaded.
- 2 - Only warning messages uploaded.
- 4 - Only information messages uploaded.

You can add values together to include more than one type of message. For example, **3** means that error messages (**1**) and warning messages (**2**) get uploaded. A value of **7** means that error messages (**1**), warning messages (**2**), and information messages (**4**) get uploaded.

To send other types of event log data to CloudWatch Logs

In addition to the application, system, and security logs, you can upload other types of event logs.

1. In the JSON file, add a new section.

```
{  
  "Id": "",  
  "FullName":  
    "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",  
  "Parameters": {  
    "LogName": "",  
    "Levels": "7"  
  }  
},
```

2. In the **Id** parameter, enter a name for the log you want to upload (e.g., WindowsBackup).
3. In the **LogName** parameter, enter the name of the log you want to upload.
 - a. To find the name of the log, in Event Viewer, in the navigation pane, click **Applications and Services Logs**.
 - b. In the list of logs, right-click the log you want to upload (e.g., Microsoft>Windows>Backup>Operational), and then click **Create Custom View**.
 - c. In the **Create Custom View** dialog box, click the **XML** tab. The **LogName** is in the <Select Path=> tag (e.g., Microsoft-Windows-Backup). Copy this text into the **LogName** parameter in the **AWS.EC2.Windows.CloudWatch.json** file.
4. In the **Levels** parameter, enter one of the following values:

- 1 - Only error messages uploaded.
- 2 - Only warning messages uploaded.
- 4 - Only information messages uploaded.

You can add values together to include more than one type of message. For example, **3** means that error messages (**1**) and warning messages (**2**) get uploaded. A value of **7** means that error messages (**1**), warning messages (**2**), and information messages (**4**) get uploaded.

To send Event Tracing (Windows) data to CloudWatch Logs

ETW (Event Tracing for Windows) provides an efficient and detailed logging mechanism that applications can write logs to. Each ETW is controlled by a session manager that can start and stop the logging session. Each session has a provider and one or more consumers.

1. In the JSON file, locate the **ETW** section.

```
{
  "Id": "ETW",
  "FullName":
  "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
  "Parameters": {
    "LogName": "Microsoft-Windows-WinINet/Analytic",
    "Levels": "7"
  }
},
```

2. In the **LogName** parameter, enter the name of the log you want to upload.
 - a. To find the name of the log, in Event Viewer, on the **View** menu, click **Show Analytic and Debug Logs**.
 - b. In the navigation pane, click **Applications and Services Logs**.
 - c. In the list of ETW logs, right-click the log you want to upload, and then click **Enable Log**.
 - d. Right-click the log again, and click **Create Custom View**.
 - e. In the **Create Custom View** dialog box, click the **XML** tab. The **LogName** is in the <Select Path=> tag (e.g., `Microsoft-Windows-WinINet/Analytic`). Copy this text into the **LogName** parameter in the **AWS.EC2.Windows.CloudWatch.json** file.
3. In the **Levels** parameter, enter one of the following values:

- 1 - Only error messages uploaded.
- 2 - Only warning messages uploaded.
- 4 - Only information messages uploaded.

You can add values together to include more than one type of message. For example, **3** means that error messages (**1**) and warning messages (**2**) get uploaded. A value of **7** means that error messages (**1**), warning messages (**2**), and information messages (**4**) get uploaded.

To send custom logs (any text-based log file) to CloudWatch Logs

1. In the JSON file, locate the **CustomLogs** section.

```
{
  "Id": "CustomLogs",
```

```
"FullName" :  
"AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent ,AWS.EC2.Windows.CloudWat  
"Parameters" : {  
    "LogDirectoryPath" : "C:\\\\CustomLogs\\\\",  
    "TimestampFormat" : "MM/dd/yyyy HH:mm:ss",  
    "Encoding" : "UTF-8",  
    "Filter" : "",  
    "CultureName" : "en-US",  
    "TimeZoneKind" : "Local",  
    "LineCount" : "5"  
},
```

2. In the **LogDirectoryPath** parameter, enter the path where logs are stored on your instance.
3. In the **TimestampFormat** parameter, enter the timestamp format you want to use. For a list of supported values, see the [Custom Date and Time Format Strings](#) topic on MSDN.

Important

Your source log file must have the timestamp at the beginning of each log line and there must be a space following the timestamp.

4. In the **Encoding** parameter, enter the file encoding to use (e.g., UTF-8). For a list of supported values, see the [Encoding Class](#) topic on MSDN.

Note

Use the encoding name, not the display name, as the value for this parameter.

5. (optional) In the **Filter** parameter, enter the prefix of log names. Leave this parameter blank to monitor all files. For a list of supported values, see the [FileSystemWatcherFilter Property](#) topic on MSDN.
6. (optional) In the **CultureName** parameter, enter the locale where the timestamp is logged. If **CultureName** is blank, it defaults to the same locale currently used by your Windows instance. For a list of supported values, see the [National Language Support \(NLS\) API Reference](#) topic on MSDN.

Note

The **div**, **div-MV**, **hu**, and **hu-HU** values are not supported.

7. (optional) In the **TimeZoneKind** parameter, enter **Local** or **UTC**. You can set this to provide time zone information when no time zone information is included in your log's timestamp. If this parameter is left blank and if your timestamp doesn't include time zone information, CloudWatch Logs defaults to the local time zone. This parameter is ignored if your timestamp already contains time zone information.
8. (optional) In the **LineCount** parameter, enter the number of lines in the header to identify the log file. For example, IIS log files have virtually identical headers. You could enter **5**, which would read the first three lines of the log file's header to identify it. In IIS log files, the third line is the date and time stamp, but the time stamp is not always guaranteed to be different between log files. For this reason, we recommend including at least one line of actual log data for uniquely fingerprinting the log file.

To send IIS log data to CloudWatch Logs

1. In the JSON file, locate the **IISLog** section.

```
{  
    "Id" : "IISLogs",  
    "FullName" :  
"AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent ,AWS.EC2.Windows.CloudWat  
    "Parameters" : {  
        "LogDirectoryPath" : "C:\\\\inetpub\\\\logs\\\\LogFiles\\\\W3SVC1",
```

```
    "TimestampFormat": "yyyy-MM-dd HH:mm:ss",  
    "Encoding": "UTF-8",  
    "Filter": "",  
    "CultureName": "en-US",  
    "TimeZoneKind": "UTC",  
    "LineCount": "5"  
  }  
},
```

2. In the **LogDirectoryPath** parameter, enter the folder where IIS logs are stored for an individual site (e.g., **C:\inetpub\logs\LogFiles\W3SVCn**).

Note

Only W3C log format is supported. IIS, NCSA, and Custom formats are not supported.

3. In the **TimestampFormat** parameter, enter the timestamp format you want to use. For a list of supported values, see the [Custom Date and Time Format Strings](#) topic on MSDN.
4. In the **Encoding** parameter, enter the file encoding to use (e.g., UTF-8). For a list of supported values, see the [Encoding Class](#) topic on MSDN.

Note

Use the encoding name, not the display name, as the value for this parameter.

5. (optional) In the **Filter** parameter, enter the prefix of log names. Leave this parameter blank to monitor all files. For a list of supported values, see the [FileSystemWatcherFilter Property](#) topic on MSDN.
6. (optional) In the **CultureName** parameter, enter the locale where the timestamp is logged. If **CultureName** is blank, it defaults to the same locale currently used by your Windows instance. For a list of supported values, see the [National Language Support \(NLS\) API Reference](#) topic on MSDN.

Note

The **div**, **div-MV**, **hu**, and **hu-HU** values are not supported.

7. (optional) In the **TimeZoneKind** parameter, enter **Local** or **UTC**. You can set this to provide time zone information when no time zone information is included in your log's timestamp. If this parameter is left blank and if your timestamp doesn't include time zone information, CloudWatch Logs defaults to the local time zone. This parameter is ignored if your timestamp already contains time zone information.
8. (optional) In the **LineCount** parameter, enter the number of lines in the header to identify the log file. For example, IIS log files have virtually identical headers. You could enter **5**, which would read the first five lines of the log file's header to identify it. In IIS log files, the third line is the date and time stamp, but the time stamp is not always guaranteed to be different between log files. For this reason, we recommend including at least one line of actual log data for uniquely fingerprinting the log file.

Step 5: Configure the Flow Control

In order to send performance counter data to CloudWatch or to send log data to CloudWatch Logs, each data type must have a corresponding destination listed in the **Flows** section. For example, to send a performance counter defined in the **"Id": "PerformanceCounter"** section of the JSON file to the CloudWatch destination defined in the **"Id": "CloudWatch"** section of the JSON file, you would enter **"PerformanceCounter,CloudWatch"** in the **Flows** section. Similarly, to send the custom log, ETW log, and system log to CloudWatch Logs, you would enter **"(CustomLogs, ETW,SystemEventLog),CloudWatchLogs"**. In addition, you can send the same performance counter or log file to more than one destination. For example, to send the application log to two different destinations that you defined in the **"Id": "CloudWatchLogs"** section of the JSON file, you would enter **"ApplicationEventLog,(CloudWatchLogs, CloudWatchLogs2)"** in the **Flows** section.

1. In the JSON file, locate the **Flows** section.

```
"Flows": {
  "Flows": [
    "PerformanceCounter,CloudWatch",
    "(PerformanceCounter,PerformanceCounter2), CloudWatch2",
    "(CustomLogs, ETW, SystemEventLog),CloudWatchLogs",
    "CustomLogs, CloudWatchLogs2",
    "ApplicationEventLog,(CloudWatchLogs, CloudWatchLogs2)"
  ]
}
```

2. In the **Flows** parameter, enter each data type that you want to upload (e.g., ApplicationEventLog) and destination where you want to send it (e.g., CloudWatchLogs).

Step 6: Create a Configuration Document

Use the AWS CLI or the Tools for Windows PowerShell to create a configuration document, specifying the JSON file that you created in the previous task.

AWS CLI

Use the following [create-document](#) command to name this configuration and make it available for use.

```
aws ssm create-document --content file://my-config.json --name "my-custom-
config"
```

Tools for Windows PowerShell

Use the following [New-SSMDocument](#) command to name this configuration and make it available for use.

```
$doc = Get-Content my-config.json | Out-String
New-SSMDocument -Content $doc -Name "my-custom-config"
```

Step 7: Associate the Configuration Document with the Instance

Use the AWS CLI or the Tools for Windows PowerShell to associate a configuration document with an instance. You'll specify the name of the configuration document that you created in the previous task. An instance can be associated with one configuration document at a time. If you associate a configuration document with an instance that already has an associated configuration document, the new configuration document replaces the existing configuration document.

AWS CLI

Use the following [create-association](#) command to associate your configuration document with your Windows instance.

```
aws ssm create-association --instance-id i-1a2b3c4d --name "my-custom-config"
```

Tools for Windows PowerShell

Use the following [New-SSMAssociation](#) command to associate your configuration document with your Windows instance.

```
New-SSMAssociation -InstanceId i-1a2b3c4d -Name "my-custom-config"
```

To stop sending logs to CloudWatch Logs, you can disassociate the configuration document from the instance. For more information, see [Disassociate the SSM document from the Instance \(p. 329\)](#).

After you disassociate the configuration document from the instance, you can delete it. For more information, see [Delete the SSM document \(p. 330\)](#).

Paravirtual Drivers

Amazon Windows AMIs contain a set of drivers to permit access to virtualized hardware. These drivers are used by Amazon EC2 to map instance store and Amazon EBS volumes to their devices. The following table shows key differences between the different drivers.

	RedHat PV	Citrix PV	AWS PV
Instance type	Not supported for all instance types. If you specify an unsupported instance type, the instance is impaired.	Supported for all instance types.	Supported for all instance types.
Attached volumes	Supports up to 16 attached volumes.	Supports more than 16 attached volumes.	Supports more than 16 attached volumes.
Network	The driver has known issues where the network connection resets under high loads; for example, fast FTP file transfers.		The driver automatically configures jumbo frames on the network adapter when on a compatible instance type. When the instance is in a placement group (p. 731) , this offers better network performance between instances in the placement group.

Contents

- [AWS PV Drivers \(p. 353\)](#)
- [Citrix PV Drivers \(p. 355\)](#)
- [RedHat PV Drivers \(p. 355\)](#)

Drivers According to Windows Version

The following list shows which PV drivers you should run on each version of Windows Server in AWS.

- Windows Server 2003 and 2003 R2, Citrix PV 5.9
- Windows Server 2008, Citrix PV 5.9
- Windows Server 2008 R2, AWS PV
- Windows Server 2012 and 2012 R2, AWS PV
- Windows Server 2016, AWS PV

AWS PV Drivers

The AWS PV drivers are stored in the `%ProgramFiles%\Amazon\Xentools` directory. This directory also contains public symbols and a command line tool, `xenstore_client.exe`, that enables you to access entries in XenStore. For example, the following PowerShell command returns the current time from the Hypervisor:

```
[DateTime]::FromFileTimeUTC((gwmi -n root\wmi -cl  
AWSXenStoreBase).XenTime).ToString("hh:mm:ss")  
11:17:00
```

The AWS PV driver components are listed in the Windows registry under `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services`. These driver components are as follows: XENBUS, xeniface, xennet, xenvbd, and xenvif.

AWS PV also has a driver component named LiteAgent, which runs as a Windows service. It handles tasks such as shutdown and restart events from the API. You can access and manage services by running `Services.msc` from the command line.

Downloading the Latest AWS PV Drivers for EC2 Windows Instances

Amazon Windows AMIs contain a set of drivers to permit access to virtualized hardware. These drivers are used by Amazon EC2 to map instance store and Amazon EBS volumes to their devices. We recommend that you install the latest drivers to improve stability and performance of you EC2 Windows instances.

You can download the setup package [here](#). For step-by-step instructions that explain how to upgrade these drivers on a Windows Server instance, see [Upgrade Windows Server 2008 R2, 2012, and 2012 R2 Instances \(AWS PV Driver Upgrade\)](#) (p. 356).

EC2 Windows PV Driver Version History

The following table shows the changes to AWS PV drivers for each driver release.

Driver version	Details
7.4.3	Added support for Windows Server 2016. Stability fixes for all supported Windows OS versions.
7.4.2	Stability fixes for support of X1 instance type.
7.4.1	<ul style="list-style-type: none">• Performance improvement in AWS PV Storage driver.• Stability fixes in AWS PV Storage driver: Fixed an issue where the instances were hitting a system crash with bugcheck code 0x0000DEAD.• Stability fixes in AWS PV Network driver.• Added support for Windows Server 2008R2.

Driver version	Details
<p>7.3.2</p> <p>Choose the link to download this specific iteration of the driver.</p>	<ul style="list-style-type: none"> Improved logging and diagnostics. Stability fix in AWS PV Storage driver. In some cases disks may not surface in Windows after reattaching the disk to the instance. Added support for Windows Server 2012.
<p>7.3.1</p>	<p>TRIM update: Fix related to TRIM requests. This fix stabilizes instances and improves instance performance when managing large numbers of TRIM requests.</p>
<p>7.3.0</p>	<p>TRIM support: The AWS PV driver now sends TRIM requests to the hypervisor. Ephemeral disks will properly process TRIM requests given the underlying storage supports TRIM (SSD). Note that EBS-based storage does not support TRIM as of March 2015.</p>
<p>7.2.5</p> <p>Choose the link to download this specific iteration of the driver.</p>	<ul style="list-style-type: none"> Stability fix in AWS PV Storage drivers: In some cases the AWS PV driver could dereference invalid memory and cause a system failure. Stability fix while generating a crash dump: In some cases the AWS PV driver could get stuck in a race condition when writing a crash dump. Before this release, the issue could only be resolved by forcing the driver to stop and restart which lost the memory dump.
<p>7.2.4</p>	<p>Device ID persistence: This driver fix masks the platform PCI device ID and forces the system to always surface the same device ID, even if the instance is moved. More generally, the fix affects how the hypervisor surfaces virtual devices. The fix also includes modifications to the co-installer for the AWS PV drivers so the system persists mapped virtual devices.</p>
<p>7.2.2</p>	<ul style="list-style-type: none"> Load the AWS PV drivers in Directory Services Restore Mode (DSRM) mode: Directory Services Restore Mode is a safe mode boot option for Windows Server domain controllers. Persist device ID when virtual network adapter device is reattached: This fix forces the system to check the MAC address mapping and persist the device ID. This fix ensures that adapters retain their static settings if the adapters are reattached.

Driver version	Details
7.2.1	<ul style="list-style-type: none">• Run in safe mode: Fixed an issue where the driver would not load in safe mode. Previously the AWS PV Drivers would only instantiate in normal running systems.• Add disks to Microsoft Windows Storage Pools: Previously we synthesized page 83 queries. The fix disabled page 83 support. Note this does not affect storage pools that are used in a cluster environment because PV disks are not valid cluster disks.
7.2.0	Base: The AWS PV base version.

Citrix PV Drivers

The Citrix PV drivers are stored in the `%ProgramFiles%\Citrix\XenTools` (32-bit instances) or `%ProgramFiles(x86)%\Citrix\XenTools` (64-bit instances) directory.

The Citrix PV driver components are listed in the Windows registry under `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services`. These driver components are as follows: `xenevtchn`, `xeniface`, `xennet`, `Xennet6`, `xensvc`, `xenvbd`, and `xenvif`.

Citrix also has a driver component named `XenGuestAgent`, which runs as a Windows service. It handles tasks such as time synchronization at boot (Windows Server 2003 only), and shutdown and restart events from the API. You can access and manage services by running `Services.msc` from the command line.

If you are encountering networking errors while performing certain workloads, you may need to disable the TCP offloading feature for the Citrix PV driver. For more information, see [TCP Offloading \(p. 367\)](#).

RedHat PV Drivers

RedHat drivers are supported for legacy instances, but are not recommended on newer instances with more than 12GB of RAM due to driver limitations. Instances with more than 12GB of RAM running RedHat drivers can fail to boot and become inaccessible. We recommend upgrading RedHat drivers to Citrix PV drivers, and then upgrade Citrix PV drivers to AWS PV drivers.

The source files for the RedHat drivers are in the `%ProgramFiles%\RedHat` (32-bit instances) or `%ProgramFiles(x86)%\RedHat` (64-bit instances) directory. The two drivers are `rhelnet`, the RedHat Paravirtualized network driver, and `rhelscsi`, the RedHat SCSI miniport driver.

Subscribing to Amazon EC2 Windows Driver Notifications

Amazon SNS can notify you when new versions of EC2 Windows Drivers are released. Use the following procedure to subscribe to these notifications.

To subscribe to EC2 notifications

1. Open the Amazon SNS console.
2. In the navigation bar, change the region to **US East (N. Virginia)**, if necessary. You must select this region because the SNS notifications that you are subscribing to were created in this region.
3. In the navigation pane, click **Subscriptions**.
4. Click **Create Subscription**.

5. In the **Create Subscription** dialog box, do the following:
 - a. In **TopicARN**, enter the following Amazon Resource Name (ARN):
arn:aws:sns:us-east-1:801119661308:ec2-windows-drivers
 - b. In **Protocol**, select `Email`.
 - c. In **Endpoint**, enter an email address that you can use to receive the notifications.
 - d. Click **Subscribe**.
6. You'll receive a confirmation email with the subject line `EC2 window`. Open the email and click **Confirm subscription** to complete your subscription.

Whenever new EC2 Windows drivers are released, we send notifications to subscribers. If you no longer want to receive these notifications, use the following procedure to unsubscribe.

To unsubscribe from Amazon EC2 Windows driver notification

1. Open the Amazon SNS console.
2. In the navigation pane, click **Subscriptions**.
3. Select the subscription and then click **Delete Subscriptions**. When prompted for confirmation, click **Yes, Delete**.

Related Topics

Upgrade: For more information about upgrading PV drivers, see [Upgrading PV Drivers on Your Windows AMI \(p. 356\)](#).

Troubleshooting: For more information about troubleshooting EC2 drivers, see [Troubleshooting PV Drivers \(p. 363\)](#). For information about troubleshooting EC2 Windows instances, see [Troubleshooting Windows Instances \(p. 939\)](#).

Upgrading PV Drivers on Your Windows AMI

To verify which driver your Windows instance uses, open **Network Connections** in Control Panel and view the **Local Area Connection**. Check whether the driver is one of the following:

- AWS PV Network Device
- Citrix PV Ethernet Adapter
- RedHat PV NIC Driver

Alternatively, you can check the output from the `pnputil -e` command.

Contents

- [Upgrade Windows Server 2008 R2, 2012, and 2012 R2 Instances \(AWS PV Driver Upgrade\) \(p. 356\)](#)
- [Upgrade Windows Server 2008 and 2008 R2 Instances \(Redhat to Citrix PV Upgrade\) \(p. 359\)](#)
- [Upgrade Windows Server 2003 Instances \(Redhat to Citrix PV Drivers\) \(p. 360\)](#)
- [Upgrade Your Citrix Xen Guest Agent Service \(p. 362\)](#)

Upgrade Windows Server 2008 R2, 2012, and 2012 R2 Instances (AWS PV Driver Upgrade)

Use the following procedure to perform an in-place upgrade of AWS PV drivers, or to upgrade from Citrix PV drivers to AWS PV drivers on Windows Server 2012 R2, Windows Server 2012, or Windows

Server 2008 R2. This upgrade is not available for RedHat drivers, or for other versions of Windows Server.

Important

If your instance is a domain controller, see [Upgrade an Instance that is a Domain Controller \(AWS PV Driver Upgrade\) \(p. 357\)](#). The upgrade process for a domain controller requires additional steps compared to non-domain controller instances.

To upgrade AWS PV drivers

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Choose the instance that requires the driver upgrade, open the context (right-click) menu, choose **Instance State**, and then choose **Stop**.

Warning

When you stop an instance, the data on any instance store volumes is erased. Therefore, if you have any data on instance store volumes that you want to keep, be sure to back it up to persistent storage.

4. After the instance is stopped, create a backup. Open the context (right-click) menu for the instance, choose **Image**, and then choose **Create Image**.
5. From the context (right-click) menu for the instance, choose **Instance State**, and then choose **Start**.
6. Connect to the instance using Remote Desktop and prepare the instance for upgrade. We recommend that you take all non-system disks offline before you perform this upgrade. Note that this step is not required if you are performing an in-place update of AWS PV drivers. We also recommend setting non-essential services to **Manual** start-up in the Services console.
7. [Download](#) the latest driver package to the instance.
8. Extract the contents of the folder and then run `AWSPVDriverSetup.msi`.

After running the MSI, the instance automatically reboots and then upgrades the driver. The instance will not be available for up to 15 minutes. After the upgrade is complete and the instance passes both health checks in the Amazon EC2 console, connect to the instance using Remote Desktop and verify that the new driver was installed. In Device Manager, under **Storage Controllers**, locate **AWS PV Storage Host Adapter**. Verify that the driver version is the same as the latest version listed in the Driver Version History table. For more information, see [EC2 Windows PV Driver Version History \(p. 353\)](#).

Note

If you previously disabled [TCP Offloading \(p. 367\)](#) using Netsh for Citrix PV drivers we recommend that you re-enable this feature after upgrading to AWS PV Drivers. TCP Offloading issues with Citrix drivers are not present in the AWS PV drivers. As a result, TCP Offloading provides better performance with AWS PV drivers.

[Upgrade an Instance that is a Domain Controller \(AWS PV Driver Upgrade\)](#)

Use the following procedure on a domain controller to perform either an in-place upgrade of AWS PV drivers, or to upgrade from Citrix PV drivers to AWS PV drivers.

Warning

The following procedure describes how to edit the Windows Registry using Registry Editor. If you are not familiar with the Registry or how to safely make changes using Registry Editor, read about the Registry on [Microsoft TechNet](#). Use caution when making any registry changes.

To upgrade a domain controller

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

-
2. In the navigation pane, choose **Instances**.
3. Choose the instance that requires the driver upgrade, open the context (right-click) menu, choose **Instance State**, and then choose **Stop**.

Warning

When you stop an instance, the data on any instance store volumes is erased. Therefore, if you have any data on instance store volumes that you want to keep, be sure to back it up to persistent storage.

-
-
-
4. After the instance is stopped, create a backup. Open the context (right-click) menu for the instance, choose **Image**, and then choose **Create Image**.
5. From the context (right-click) menu for the instance, choose **Instance State**, and then choose **Start**.
6. Run the following command. This command configures Windows to boot into Directory Services Restore Mode (DSRM) after the system restarts. The system must boot into DSRM because the upgrade utility removes Citrix PV storage drivers so it can install AWS PV drivers. When Citrix PV storage drivers are not present, secondary drives will not be detected. Domain controllers that use an NTDS folder on secondary drives will not boot because the secondary disk will not be detected.

bcdedit /set {default} safeboot dsrepair

Warning

After you run this command do *not* manually reboot the system. The system will be unreachable because Citrix PV drivers do not support DSRM.

-
-
-
-
-
-
7. Open Registry Editor and create the following key: HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Amazon\AWSPVDriverSetup.
8. From the context (right-click) menu, choose **New** and then choose **String Value**. Specify **DisableDCCheck** as the name and the value as **true**.
9. [Download](#) the latest driver package to the instance.
10. Extract the contents of the folder and then run `AWSPVDriverSetup.msi`.

After running the MSI, the instance automatically reboots and then upgrades the driver. The instance will not be available for up to 15 minutes.

-
-
-
-
-
-
-
-
-
11. After the upgrade is complete and the instance passes both health checks in the Amazon EC2 console, connect to the instance using Remote Desktop.

Important

You must connect to the instance by specifying user name in the following format `hostname\administrator`. For example, `Win2k12TestBox\administrator`.

-
-
-
-
-
-
-
-
-
-
-
12. From a Command prompt, run the following command to remove the DSRM boot configuration:

bcdedit /deletevalue safeboot

-
-
-
-
-
-
-
-
-
-
-
-
13. Reboot the instance.
14. To complete the upgrade process, verify that the new driver was installed. In Device Manager, under **Storage Controllers**, locate **AWS PV Storage Host Adapter**. Verify that the driver version is the same as the latest version listed in the Driver Version History table. For more information, see [EC2 Windows PV Driver Version History \(p. 353\)](#).
15. Open Registry Editor and delete **DisableDCCheck** from the following key: HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Amazon\AWSPVDriverSetup.

Note

If you previously disabled [TCP Offloading \(p. 367\)](#) using Netsh for Citrix PV drivers we recommend that you re-enable this feature after upgrading to AWS PV Drivers. TCP Offloading issues with Citrix drivers are not present in the AWS PV drivers. As a result, TCP Offloading provides better performance with AWS PV drivers.

Upgrade Windows Server 2008 and 2008 R2 Instances (Redhat to Citrix PV Upgrade)

Before you start upgrading your RedHat drivers to Citrix PV drivers, make sure you do the following:

- Install the latest version of the EC2Config service. For more information, see [Installing the Latest Version of EC2Config \(p. 295\)](#).
- Verify that you have Windows PowerShell 2.0 installed. To verify the version that you have installed, run the following command in a PowerShell window:

```
PS C:> $PSVersionTable.PSVersion
```

If you need to install version 2.0, see [Windows Management Framework \(Windows PowerShell 2.0, WinRM 2.0, and BITS 4.0\)](#) from Microsoft Support.

- Back up your important information on the instance, or create an AMI from the instance. For more information about creating an AMI, see [Creating an Amazon EBS-Backed Windows AMI \(p. 77\)](#). If you create an AMI, make sure that you do the following:
 - Write down your password.
 - Do not run the Sysprep tool manually or using the EC2Config service.
 - Set your Ethernet adapter to obtain an IP address automatically using DHCP. For more information, see [Configure TCP/IP Settings](#) in the Microsoft TechNet Library.

To upgrade Redhat drivers

1. Connect to your instance and log in as the local administrator. For more information about connecting to your instance, see [Connecting to Your Windows Instance \(p. 254\)](#).
2. In your instance, [download](#) the Citrix PV upgrade package.
3. Extract the contents of the upgrade package to a location of your choice.
4. Double-click the **Upgrade.bat** file. If you get a security warning, click **Run**.
5. In the **Upgrade Drivers** dialog box, review the information and click **Yes** if you are ready to start the upgrade.
6. In the **Red Hat Paravirtualized Xen Drivers for Windows uninstaller** dialog box, click **Yes** to remove the RedHat software. Your instance will be rebooted.

Note

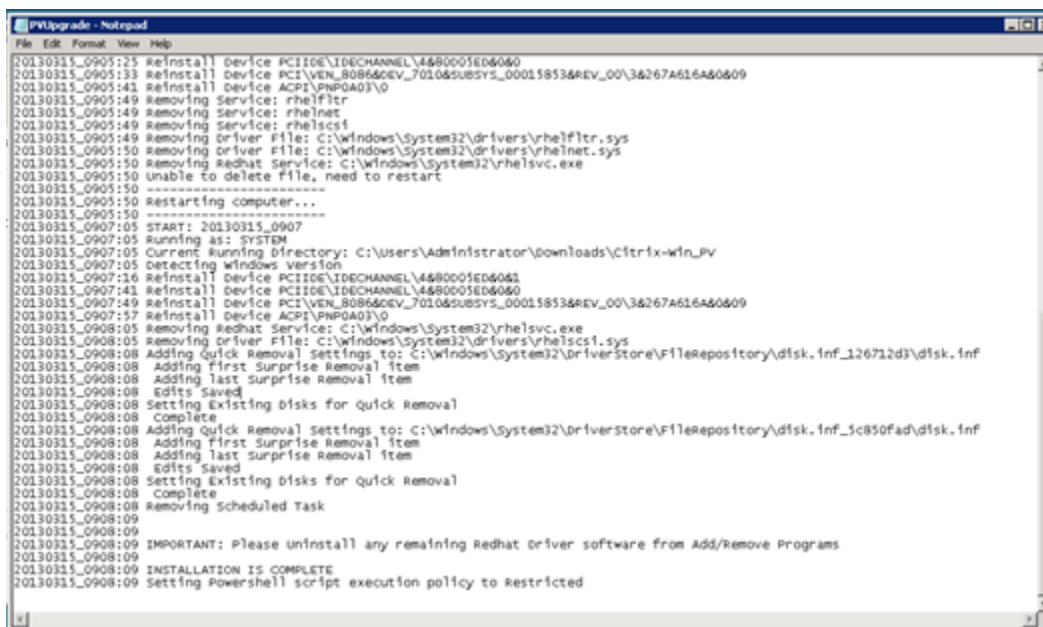
If you do not see the uninstaller dialog box, click **Red Hat Paravirtualize...** in the Windows taskbar.



7. Check that the instance has rebooted and is ready to be used.
 - a. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
 - b. On the **Instances** page, right-click your instance and select **Get System Log**.
 - c. The upgrade operations should have restarted the server 3 or 4 times. You can see this in the log file by the number of times `Windows is Ready to use` is displayed.

```
Microsoft Windows NT 6.0.6002 Service Pack 2 (en-US)
Ec2Config service v2.1.9.0
RedHat PV NIC Driver v1.3.10.0
2013/03/15 17:11:01Z: Waiting for meta-data accessibility...
2013/03/15 17:11:02Z: Meta-data is now available.
<RDPCERTIFICATE>
<THUMBPRINT>D68FD64F21359516C781CA7DF2821C5EFC35648A</THUMBPRINT>
</RDPCERTIFICATE>
<Username>Administrator</Username>
<Password>
L79ThJPF8LyIL38I2ht0FBrjet3vnT2csTiU/XGVMRCH7kQtBznznAnXrKdIsirXlXl9BwMad9b38jFJqv01IUpgNNJR2oCdc7IbUw
</Password>
2013/03/15 17:11:30Z: Product activation was successful.
2013/03/15 17:11:32Z: Message: Windows is Ready to use
Microsoft Windows NT 6.0.6002 Service Pack 2 (en-US)
Ec2Config service v2.1.9.0
2013/03/15 21:04:24Z: There was an exception writing driver information to console: System.Exception:
at Ec2Config.Service1.Go()
2013/03/15 21:04:35Z: Waiting for meta-data accessibility...
2013/03/15 21:04:40Z: Meta-data is now available.
<RDPCERTIFICATE>
<THUMBPRINT>D68FD64F21359516C781CA7DF2821C5EFC35648A</THUMBPRINT>
</RDPCERTIFICATE>
2013/03/15 21:05:08Z: Product activation was successful.
2013/03/15 21:05:09Z: Message: Windows is Ready to use
Microsoft Windows NT 6.0.6002 Service Pack 2 (en-US)
Ec2Config service v2.1.9.0
Citrix PV Ethernet Adapter v5.9.960.49119
2013/03/15 21:07:20Z: Waiting for meta-data accessibility...
2013/03/15 21:07:21Z: Meta-data is now available.
<RDPCERTIFICATE>
<THUMBPRINT>D68FD64F21359516C781CA7DF2821C5EFC35648A</THUMBPRINT>
</RDPCERTIFICATE>
2013/03/15 21:07:27Z: Message: Windows is Ready to use
```

8. Connect to your instance and log in as the local administrator.
9. Close the **Red Hat Paravirtualized Xen Drivers for Windows uninstaller** dialog box.
10. Confirm that the installation is complete. Navigate to the `Citrix-WIN_PV` folder that you extracted earlier, open the `PVUpgrade.log` file, and then check for the text `INSTALLATION IS COMPLETE`.



```
PVUpgrade - Notepad
File Edit Format View Help
20130315_0905:25 Reinstall Device PCI\IDE\IDECHANNEL\4480005ED6060
20130315_0905:33 Reinstall Device PCI\VEN_8086&DEV_7010&SUBSYS_00015853&REV_00\3&267A616A&0609
20130315_0905:41 Reinstall Device ACPI\PNP0A03\0
20130315_0905:49 Removing Service: rhelfiltr
20130315_0905:49 Removing Service: rhelnet
20130315_0905:49 Removing Driver File: C:\Windows\System32\drivers\rhelfiltr.sys
20130315_0905:50 Removing Driver File: C:\Windows\System32\drivers\rhelnet.sys
20130315_0905:50 Removing Redhat Service: C:\Windows\System32\rhelsvc.exe
20130315_0905:50 Unable to delete file, need to restart
20130315_0905:50
20130315_0905:50 Restarting computer...
20130315_0905:50
-----
20130315_0907:05 START: 20130315_0907
20130315_0907:05 Running as: SYSTEM
20130315_0907:05 Current running directory: C:\Users\Administrator\downloads\Citrix-win_PV
20130315_0907:05 Detecting Windows version
20130315_0907:16 Reinstall Device PCI\IDE\IDECHANNEL\4480005ED6060
20130315_0907:41 Reinstall Device PCI\IDE\IDECHANNEL\4480005ED6060
20130315_0907:49 Reinstall Device PCI\VEN_8086&DEV_7010&SUBSYS_00015853&REV_00\3&267A616A&0609
20130315_0908:05 Removing Redhat Service: C:\Windows\System32\rhelsvc.exe
20130315_0908:05 Removing Driver File: C:\Windows\System32\drivers\rhelfiltr.sys
20130315_0908:08 Adding quick Removal Settings to: C:\Windows\System32\DriverStore\FileRepository\disk_inf_126712d3\disk_inf
20130315_0908:08 Adding first Surprise Removal Item
20130315_0908:08 Adding last surprise Removal Item
20130315_0908:08 Edits Saved
20130315_0908:08 Setting Existing disks for quick Removal
20130315_0908:08 complete
20130315_0908:08 Adding Quick Removal Settings to: C:\Windows\System32\DriverStore\FileRepository\disk_inf_1c850fad\disk_inf
20130315_0908:08 Adding first Surprise Removal Item
20130315_0908:08 Adding last surprise Removal Item
20130315_0908:08 Edits Saved
20130315_0908:08 Setting Existing disks for quick Removal
20130315_0908:08 complete
20130315_0908:08 Removing Scheduled Task
20130315_0908:09
20130315_0908:09 IMPORTANT: Please uninstall any remaining Redhat driver software from Add/Remove Programs
20130315_0908:09
20130315_0908:09 INSTALLATION IS COMPLETE
20130315_0908:09 Setting Powershell script execution policy to Restricted
```

Upgrade Windows Server 2003 Instances (Redhat to Citrix PV Drivers)

Before you start upgrading your RedHat drivers to Citrix PV drivers, make sure you do the following:

- Verify that Windows PowerShell 2.0 is installed on your Windows Server 2003 instance. The upgrade process requires PowerShell 2.0. For information about how to install PowerShell 2.0, see [Windows Management Framework Core package \(Windows PowerShell 2.0 and WinRM 2.0\)](#).

- Back up your important information on the instance, or create an AMI from the instance. For more information about creating an AMI, see [Creating an Amazon EBS-Backed Windows AMI \(p. 77\)](#). If you create an AMI, make sure you do the following:
 - Do not enable the Sysprep tool in the EC2Config service.
 - Write down your password.
 - Set your Ethernet adapter to DHCP.
- Install the latest version of the EC2Config service. For more information, see [Installing the Latest Version of EC2Config \(p. 295\)](#).

To upgrade Redhat drivers

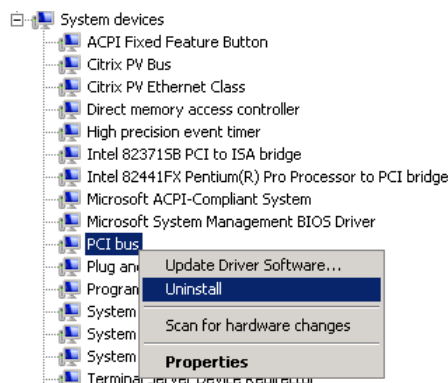
1. Connect to your instance and log in as the local administrator. For more information about connecting to your instance, see [Connecting to Your Windows Instance \(p. 254\)](#).
2. In your instance, [download](#) the Citrix PV upgrade package.
3. Extract the contents of the upgrade package to a location of your choice.
4. Double-click the **Upgrade.bat** file. If you get a security warning, click **Run**.
5. In the **Upgrade Drivers** dialog box, review the information and click **Yes** if you're ready to start the upgrade.
6. In the **Red Hat Paravirtualized Xen Drivers for Windows uninstaller** dialog box, click **Yes** to remove the RedHat software. Your instance will be rebooted.

Note

If you do not see the uninstaller dialog box, click **Red Hat Paravirtualize...** in the Windows taskbar.

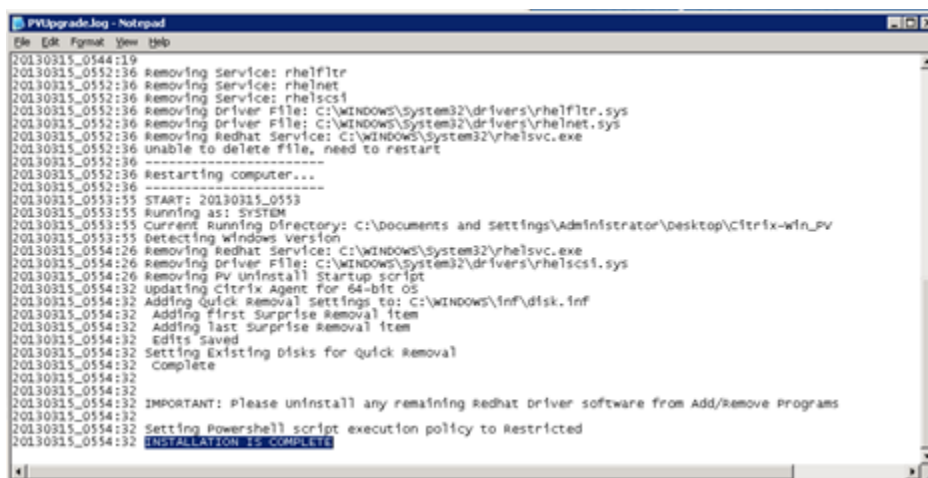


7. Check that the instance has been rebooted and is ready to be used.
 - a. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
 - b. On the **Instances** page, right-click your instance and select **Get System Log**.
 - c. Check the end of the log message. It should read `Windows is Ready to use`.
8. Connect to your instance and log in as the local administrator. The upgrade will continue by opening four applications: PowerShell, RedHat uninstaller, PVUpgrade.log and the Windows Device Manager.
9. Uninstall the PCI BUS.
 - a. In the **Device Manager** window, expand **System devices**, right-click **PCI bus** and click **Uninstall**.



- b. When prompted, click **OK**.

- c. In the **System Settings Change** dialog, click **No** as you do not want to restart your instance immediately.
 - d. Close **Device Manager**. The upgrade script reboots your instance.
10. Check that the instance is ready by repeating the procedure in step 7. After you've confirmed it is ready, log in as the administrator.
 11. Confirm that the installation is complete. Navigate to the `Citrix-WIN_PV` folder that you extracted earlier, open the `PVUpgrade.log` file, and then check for the text `INSTALLATION IS COMPLETE`.



```
PVUpgrade.log - Notepad
File Edit Format View Help
20130315_0544:19
20130315_0552:36 Removing Service: rhelftr
20130315_0552:36 Removing Service: rhelnet
20130315_0552:36 Removing Service: rhelnet.sys
20130315_0552:36 Removing driver File: C:\WINDOWS\system32\drivers\rhelftr.sys
20130315_0552:36 Removing driver File: C:\WINDOWS\system32\drivers\rhelnet.sys
20130315_0552:36 Removing Redhat Service: C:\WINDOWS\system32\rhelsvc.exe
20130315_0552:36 unable to delete file, need to restart
20130315_0552:36
20130315_0552:36 Restarting computer...
20130315_0552:36 -----
20130315_0553:55 START: 20130315_0553
20130315_0553:55 Running as: SYSTEM
20130315_0553:55 Current Running Directory: C:\documents and settings\Administrator\Desktop\Citrix-win_pv
20130315_0553:55 Detecting windows version
20130315_0554:26 Removing Redhat Service: C:\WINDOWS\system32\rhelsvc.exe
20130315_0554:26 Removing driver File: C:\WINDOWS\system32\drivers\rhelnet.sys
20130315_0554:26 Removing PV uninstall Startup script
20130315_0554:32 updating Citrix agent for 64-bit OS
20130315_0554:32 Adding quick removal settings to: C:\WINDOWS\inf\disk.inf
20130315_0554:32 Adding first Surprise Removal Item
20130315_0554:32 Adding last Surprise Removal Item
20130315_0554:32 Edits Saved
20130315_0554:32 Setting Existing disks for quick removal
20130315_0554:32 complete
20130315_0554:32
20130315_0554:32 IMPORTANT: Please uninstall any remaining Redhat Driver software from Add/Remove Programs
20130315_0554:32
20130315_0554:32 Setting Powershell script execution policy to Restricted
20130315_0554:32 INSTALLATION IS COMPLETE
20130315_0554:32
```

Upgrade Your Citrix Xen Guest Agent Service

If you are using Citrix PV drivers on your Windows server, you can upgrade the Citrix Xen guest agent service. This Windows service handles tasks such as time synchronization at boot, as well as shutdown and restart events from the API. You can run this upgrade package on any version of Windows Server, including Windows Server 2012, as long as the Windows Server 2012 instance is running Citrix PV drivers.

Important

Do not perform these steps on Windows Server 2012 or 2012 R2 instances that are running AWS PV drivers.

Before you start upgrading your drivers, make sure you back up your important information on the instance, or create an AMI from the instance. For more information about creating an AMI, see [Creating an Amazon EBS-Backed Windows AMI \(p. 77\)](#). If you create an AMI, make sure you do the following:

- Do not enable the Sysprep tool in the EC2Config service.
- Write down your password.
- Set your Ethernet adapter to DHCP.

To upgrade your Citrix Xen guest agent service

1. Connect to your instance and log in as the local administrator. For more information about connecting to your instance, see [Connecting to Your Windows Instance \(p. 254\)](#).
2. On your instance, [download](#) the Citrix upgrade package.
3. Extract the contents of the upgrade package to a location of your choice.
4. Double-click the **Upgrade.bat** file. If you get a security warning, click **Run**.

5. In the **Upgrade Drivers** dialog box, review the information and click **Yes** if you are ready to start the upgrade.
6. When the upgrade is complete, the `PVUpgrade.log` file will open and contain the text `UPGRADE IS COMPLETE`.
7. Reboot your instance.

Troubleshooting PV Drivers

This topic describes solutions to common issues that you might encounter with Amazon EC2 PV drivers.

Contents

- [Windows Server 2012 R2 loses network and storage connectivity after an instance reboot \(p. 363\)](#)
- [TCP Offloading \(p. 367\)](#)
- [Time Synchronization \(p. 370\)](#)

Windows Server 2012 R2 loses network and storage connectivity after an instance reboot

Windows Server 2012 R2 Amazon Machine Images (AMIs) made available *before* September 10, 2014 can lose network and storage connectivity after an instance reboot. The error in the AWS Management Console system log states: "Difficulty detecting PV driver details for Console Output." The connectivity loss is caused by the Windows Server 2012 R2 Plug and Play Cleanup feature. This feature scans for and disables inactive system devices every 30 days. The feature incorrectly identifies the EC2 network device as inactive and removes it from the system. When this happens, the instance loses network connectivity after a reboot.

For systems that you suspect could be affected by this issue, you can download and run an in-place driver upgrade. If you are unable to perform the in-place driver upgrade, you can run a helper script. The script determines if your instance is affected. If it is affected, and the Amazon EC2 network device has *not* been removed, the script disables the Plug and Play Cleanup scan. If the Amazon EC2 network device has been removed, the script repairs the device, disables the Plug and Play Cleanup scan, and allows your instance to reboot with network connectivity enabled.

In this section

- [Choose How You Want to Fix This Problem \(p. 363\)](#)
- [Method 1 - Enhanced Networking \(p. 364\)](#)
- [Method 2 - Registry configuration \(p. 365\)](#)
- [Run the Remediation Script \(p. 366\)](#)

Choose How You Want to Fix This Problem

There are two methods for restoring network and storage connectivity to an instance affected by this issue. Choose one of the following methods:

Method	Prerequisites	Procedure Overview
Method 1 - Enhanced networking	Enhanced networking is only available in a virtual private cloud (VPC) which requires a	You change the server instance type to a C3 instance. Enhanced networking then enables you to

Method	Prerequisites	Procedure Overview
	C3 instance type. If the server does not currently use the C3 instance type, then you must temporarily change it. Enhanced networking is not available for ec2-classic.	connect to the affected instance and fix the problem. After you fix the problem, you change the instance back to the original instance type. This method is typically faster than Method 2 and less likely to result in user error. You will incur additional charges as long as the C3 instance is running.
Method 2 - Registry configuration	Ability to create or access a second server. Ability to change Registry settings.	You detach the root volume from the affected instance, attach it to a different instance, connect, and make changes in the Registry. You will incur additional charges as long as the additional server is running. This method is slower than Method 1, but this method has worked in situations where Method 1 failed to resolve the problem.

Method 1 - Enhanced Networking

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Locate the affected instance. Open the context (right-click) menu for the instance, choose **Instance State**, and then choose **Stop**.

Warning

When you stop an instance, the data on any instance store volumes is erased. Therefore, if you have any data on instance store volumes that you want to keep, be sure to back it up to persistent storage.

4. After the instance is stopped create a backup. Open the context (right-click) menu for the instance, choose **Image**, and then choose **Create Image**.
5. [Change](#) the instance type to any C3 instance type.
6. [Start](#) the instance.
7. Connect to the instance using Remote Desktop and then [download](#) the AWS PV Drivers Upgrade package to the instance.
8. Extract the contents of the folder and run `AWSPVDriverSetup.msi`.

After running the MSI, the instance automatically reboots and then upgrades the drivers. The instance will not be available for up to 15 minutes.

9. After the upgrade is complete and the instance passes both health checks in the Amazon EC2 console, connect to the instance using Remote Desktop and verify that the new drivers were installed. In Device Manager, under **Storage Controllers**, locate **AWS PV Storage Host Adapter**. Verify that the driver version is the same as the latest version listed in the Driver Version History table. For more information, see [EC2 Windows PV Driver Version History \(p. 353\)](#).
10. Stop the instance and change the instance back to its original instance type.
11. Start the instance and resume normal use.

Method 2 - Registry configuration

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Locate the affected instance. Open the context (right-click) menu for the instance, choose **Instance State**, and then choose **Stop**.

Warning

When you stop an instance, the data on any instance store volumes is erased. Therefore, if you have any data on instance store volumes that you want to keep, be sure to back it up to persistent storage.

4. Choose **Launch Instance** and create a temporary Windows Server 2008 or Windows Server 2012 instance in the same Availability Zone as the affected instance. Do *not* create a Windows Server 2012 R2 instance.

Important

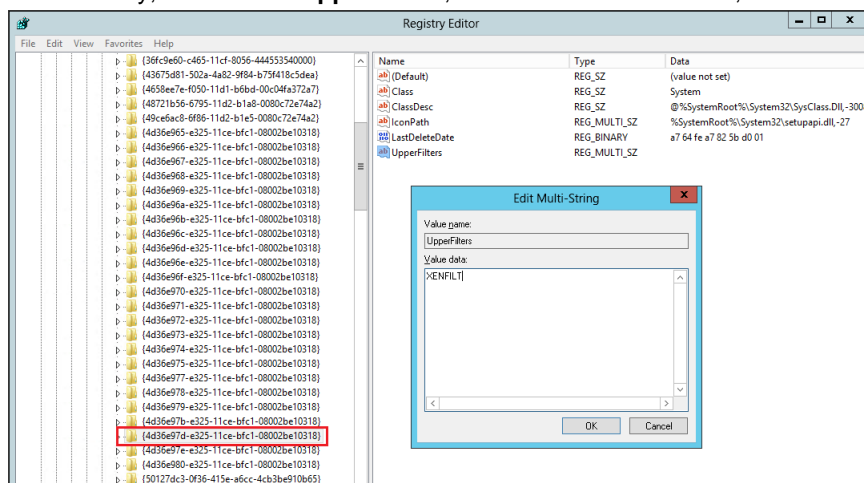
If you do not create the instance in the same Availability Zone as the affected instance you will not be able to attach the root volume of the affected instance to the new instance.

5. In the navigation pane, choose **Volumes**.
6. Locate the root volume of the affected instance. **Detach** the volume and **attach** it to the temporary instance you created earlier. Attach it with the default device name (xvdf).
7. Use Remote Desktop to connect to the temporary instance, and then use the Disk Management utility to **make the volume available for use**.
8. On the temporary instance, open the Run dialog box, type regedit, and press Enter.
9. In the Registry Editor navigation pane, choose **HKEY_LOCAL_MACHINE**, and then from the **File** menu choose **Load Hive**.
10. In the **Load Hive** dialog box, navigate to *Affected Volume*\Windows\System32\config\System and type a temporary name in the **Key Name** dialog box. For example, enter OldSys.
11. In the navigation pane of the Registry Editor, locate the following keys:

HKEY_LOCAL_MACHINE\your_temporary_key_name\ControlSet001\Control\Class\4d36e97d-e325-11ce-bfc1-08002be10318

HKEY_LOCAL_MACHINE\your_temporary_key_name\ControlSet001\Control\Class\4d36e96a-e325-11ce-bfc1-08002be10318

12. For each key, double-click **UpperFilters**, enter a value of XENFILT, and then click **OK**.



13. Locate the following key:

HKEY_LOCAL_MACHINE\your_temporary_key_name\ControlSet001\Services\XENBUS\Parameters

14. Create a new string (REG_SZ) with the name `ActiveDevice` and the following value:

PCI\VEN_5853&DEV_0001&SUBSYS_00015853&REV_01

15. Locate the following key:

HKEY_LOCAL_MACHINE\your_temporary_key_name\ControlSet001\Services\XENBUS

16. Change the **Count** from 0 to 1.

17. Locate and delete the following keys:

HKEY_LOCAL_MACHINE\your_temporary_key_name\ControlSet001\Services\xenvbd\StartOverride

HKEY_LOCAL_MACHINE\your_temporary_key_name\ControlSet001\Services\xenfilt\StartOverride

18. In the Registry Editor navigation pane, choose the temporary key that you created when you first opened the Registry Editor.
19. From the **File** menu, choose **Unload Hive**.
20. In the Disk Management Utility, choose the drive you attached earlier, open the context (right-click) menu, and choose **Offline**.
21. In the Amazon EC2 console, detach the affected volume from the temporary instance and reattach it to your Windows Server 2012 R2 instance with the device name `/dev/sda1`. You must specify this device name to designate the volume as a root volume.
22. [Start](#) the instance.
23. Connect to the instance using Remote Desktop and then [download](#) the AWS PV Drivers Upgrade package to the instance.
24. Extract the contents of the folder and run `AWSPVDriverSetup.msi`.

After running the MSI, the instance automatically reboots and then upgrades the drivers. The instance will not be available for up to 15 minutes.

25. After the upgrade is complete and the instance passes both health checks in the Amazon EC2 console, connect to the instance using Remote Desktop and verify that the new drivers were installed. In Device Manager, under **Storage Controllers**, locate **AWS PV Storage Host Adapter**. Verify that the driver version is the same as the latest version listed in the Driver Version History table. For more information, see [EC2 Windows PV Driver Version History \(p. 353\)](#).
26. Delete or stop the temporary instance you created in this procedure.

Run the Remediation Script

If you are unable to perform an in-place driver upgrade or migrate to a newer instance you can run the remediation script to fix the problems caused by the Plug and Play Cleanup task.

To run the remediation script

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Choose the instance for which you want to run the remediation script. Open the context (right-click) menu for the instance, choose **Instance State**, and then choose **Stop**.

Warning

When you stop an instance, the data on any instance store volumes is erased. Therefore, if you have any data on instance store volumes that you want to keep, be sure to back it up to persistent storage.

4. After the instance is stopped create a backup. Open the context (right-click) menu for the instance, choose **Image**, and then choose **Create Image**.
5. Open the context (right-click) menu for the instance, choose **Instance State**, and then choose **Start**.
6. Connect to the instance by using Remote Desktop and then [download](#) the RemediateDriverIssue.zip folder to the instance.
7. Extract the contents of the folder.
8. Run the remediation script according to the instructions in the Readme.txt file. The file is located in the folder where you extracted RemediateDriverIssue.zip.

TCP Offloading

By default, TCP offloading is enabled for the Citrix PV drivers in Windows AMIs. If you encounter transport-level errors or packet transmission errors (as visible on the Windows Performance Monitor)—for example, when you're running certain SQL workloads—you may need to disable this feature.

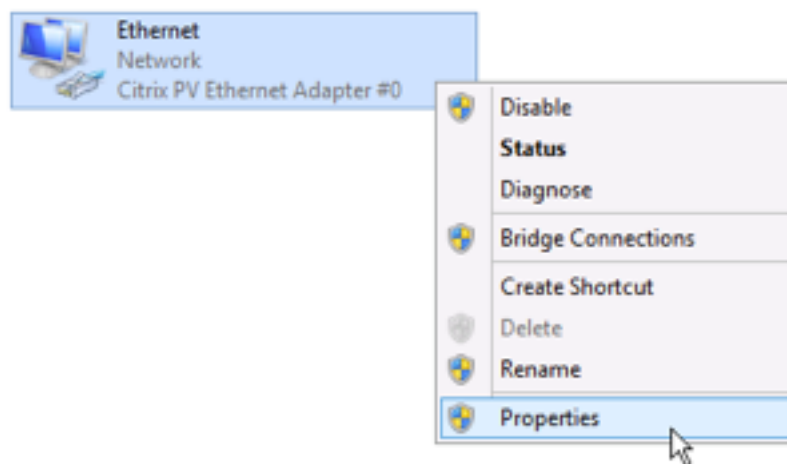
Important

Disabling TCP offloading may reduce the network performance of your instance.

You do not need to perform this procedure on instances running AWS PV or Intel network drivers.

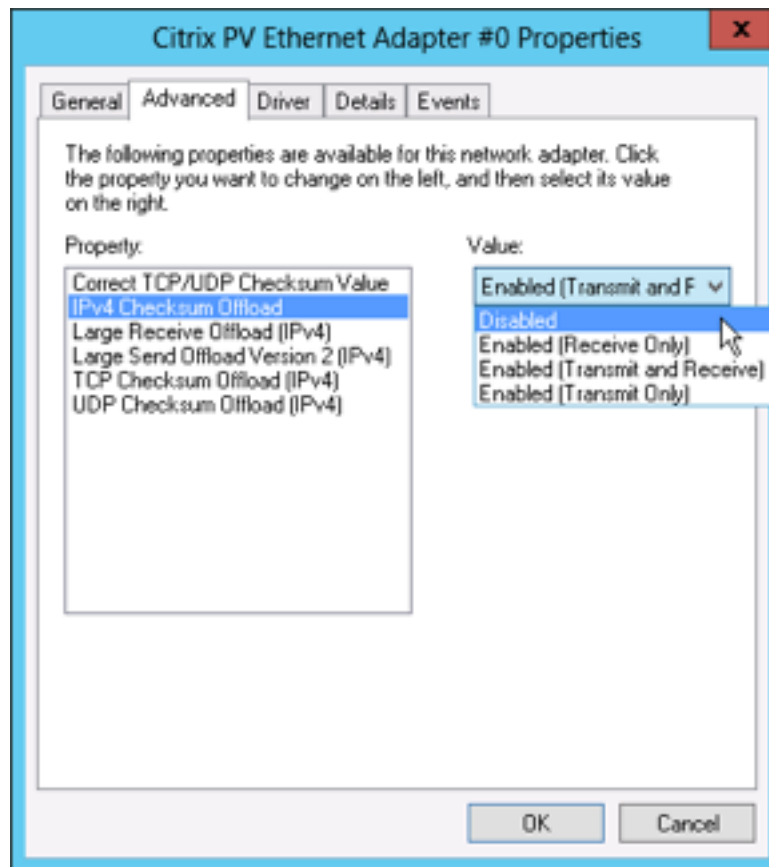
To disable TCP offloading for Windows Server 2012 and 2008

1. Connect to your instance and log in as the local administrator.
2. If you're using Windows Server 2012, press **Ctrl+Esc** to access the **Start** screen, and then click **Control Panel**. If you're using Windows Server 2008, click **Start** and select **Control Panel**.
3. Click **Network and Internet**, then **Network and Sharing Center**.
4. Click **Change adapter settings**.
5. Right-click **Citrix PV Ethernet Adapter #0** and select **Properties**.



6. In the **Local Area Connection Properties** dialog box, click **Configure** to open the **Citrix PV Ethernet Adapter #0 Properties** dialog box.
7. On the **Advanced** tab, disable each of the following properties by selecting them in the **Property** list, and selecting **Disabled** from the **Value** list:
 - **IPv4 Checksum Offload**
 - **Large Receive Offload (IPv4)**
 - **Large Send Offload Version 2 (IPv4)**

- **TCP Checksum Offload (IPv4)**
- **UDP Checksum Offload (IPv4)**



8. Click **OK**.
9. Run the following commands from a Command Prompt window.

```
C:\> netsh int ip set global taskoffload=disabled
C:\> netsh int tcp set global chimney=disabled
C:\> netsh int tcp set global rss=disabled
C:\> netsh int tcp set global netdma=disabled
```

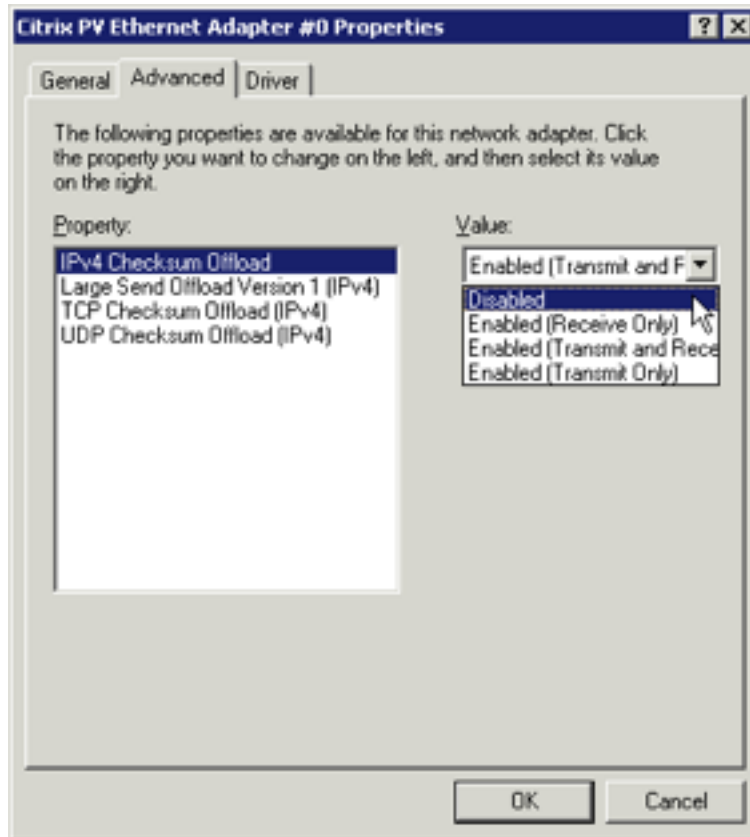
10. Reboot the instance.

To disable TCP offloading for Windows Server 2003

1. Connect to your instance and log in as the local administrator.
2. Click **Start**, and select **Control Panel**, then **Network Connections**, and then **Local Area Connection 3**.
3. Click **Properties**.
4. In the **Local Area Connection 3** dialog box, click **Configure...** to open the **Citrix PV Ethernet Adapter #0 Properties** dialog box.
5. On the **Advanced** tab, disable each of the following properties by selecting them in the **Property** list, and selecting **Disabled** from the **Value** list:

- **IPv4 Checksum Offload**

- **Large Send Offload Version 1 (IPv4)**
- **TCP Checksum Offload (IPv4)**
- **UDP Checksum Offload (IPv4)**



6. Click **OK**.
7. Run the following PowerShell script.

```
$n = Get-ItemProperty "HKLM:\SYSTEM\Select" | Select -expand Current
$root = "HKEY_LOCAL_MACHINE\SYSTEM\ControlSet00$n\Control\Class\{4D36E972-
E325-11CE-BFC1-08002BE10318}"
$items = Get-ChildItem -Path Registry::$Root -Name

Foreach ($item in $items) {
    if ($item -ne "Properties") {
        $path = $root + "\" + $item
        $DriverDesc = Get-ItemProperty -Path Registry::$path | Select-Object
        -expandproperty DriverDesc
        if ($DriverDesc -eq "Citrix PV Ethernet Adapter") {
            Set-ItemProperty -path Registry::$path -Name
            *IPChecksumOffloadIPv4 -Value 0
            Set-ItemProperty -path Registry::$path -Name
            *TCPChecksumOffloadIPv4 -Value 0
            Set-ItemProperty -path Registry::$path -Name
            *UDPChecksumOffloadIPv4 -Value 0
            Set-ItemProperty -path Registry::$path -Name *LSOv1IPv4 -Value 0
        }
    }
}
```



```
}  
}
```

8. Reboot the instance.

Time Synchronization

Prior to the release of the 2013.02.13 Windows AMI, the Citrix Xen guest agent could set the system time incorrectly. This can cause your DHCP lease to expire. If you have issues connecting to your instance, you might need to update the agent.

To determine whether you have the updated Citrix Xen guest agent, check whether the `C:\Program Files\Citrix\XenGuestAgent.exe` file is from March 2013. If the date on this file is earlier than that, update the Citrix Xen guest agent service. For more information, see [Upgrade Your Citrix Xen Guest Agent Service \(p. 362\)](#).

Related Topics

For information about troubleshooting EC2 Windows instances, see [Troubleshooting Windows Instances \(p. 939\)](#).

Setting Passwords for Windows Instances

When you connect to a Windows instance, you must specify a user account and password that has permission to access the instance. The first time that you connect to an instance, you are prompted to specify the Administrator account and the default password. The default password is automatically generated by the EC2Config service.

When you connect to an instance the first time, we recommend that you change the Administrator password from its default value. If you lose your password or it expires, you can manually configure EC2Config to generate a new password.

Contents

- [Changing the Administrator Password After Connecting \(p. 370\)](#)
- [Resetting an Administrator Password that's Lost or Expired \(p. 371\)](#)

Changing the Administrator Password After Connecting

Use the following procedure to change the Administrator password for a Windows instance.

Important

Store the new password in a safe place. You won't be able to retrieve the new password using the Amazon EC2 console. The console can only retrieve the default password. If you attempt to connect to the instance using the default password after changing it, you'll get a "Your credentials did not work" error.

To change the local Administrator password

1. Connect to the instance and open a command prompt.
2. Run the following command. If your new password includes special characters, ensure that you enclose the password in double quotes:

```
C:\> net user Administrator "new_password"
```

3. Store the new password in a safe place.

Resetting an Administrator Password that's Lost or Expired

If you've lost the Windows Administrator password for your Amazon EC2 instance, or if the password has expired, you can reset it using the EC2Config service, as described in this section.

Note

You can't reset the password if you've disabled the local Administrator account on the instance.

This section also describes how to connect to an instance if you've lost the key pair that was used to create the instance. EC2 uses a public key to encrypt a piece of data, such as a password, and a private key to decrypt the data. The public and private keys are known as a *key pair*. With Windows instances, you use a key pair to obtain the administrator password and then log in using RDP.

Before You Begin

Before you attempt to reset the administrator password, use the following procedure to verify that the EC2Config service is installed and running. You will use the EC2Config service to reset the Administrator password later in this section.

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances** and then choose the instance that needs a password reset. (This instance is referred to as the *original* instance in this procedure.)
3. Choose **Actions, Instance settings, Get system log**.
4. Locate the EC2 Agent entry. For example **EC2 Agent: Ec2Config service v3.18.1118**. If you see this entry, the EC2Config service is running.

If the system log output is empty, or if the EC2Config service is not running, troubleshoot the instance using the Instance Console Screenshot service. For more information, see [Troubleshoot an Unreachable Instance \(p. 939\)](#).

To reset an Administrator password for an EC2 instance, you modify a configuration file on the instance boot volume. However, you can't modify this file if it is attached to the instance as a root volume. You must detach the volume and attach it to a temporary instance. After you modify the configuration file on the temporary instance, you reattach it to your original instance as the root volume, as described in the following procedure.

Important

The instance gets a new public IP address after you stop and start it as described in the following procedure. After resetting the password, be sure to connect to the instance using its current public DNS name. If the instance is in EC2-Classic, any Elastic IP address is disassociated from the instance, so you must reassociate it. For more information, see [Instance Lifecycle \(p. 241\)](#).

To reset the Administrator password

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the EC2 console, choose **Instances**, and then choose the instance that needs a password reset.
3. Choose **Actions, Instance state, Stop**.

Warning

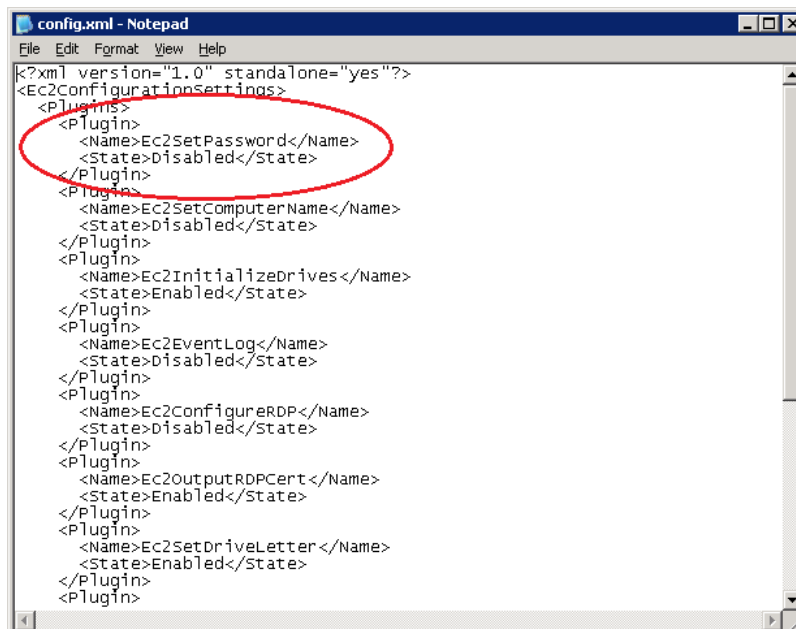
When you stop an instance, the data on any instance store volumes is erased. Therefore, if you have any data on instance store volumes that you want to keep, be sure to back it up to persistent storage.

4. In the **Stop instances** dialog box, choose **Yes, Stop**. After the instance has stopped, proceed with the next step.

Warning

If your temporary instance is based on the same AMI that the original instance is based on, and the operating system is later than Windows Server 2003, you must complete additional steps or you won't be able to boot the original instance after you restore its root volume because of a disk signature collision. Alternatively, select a different AMI for the temporary instance. For example, if the original instance uses the AWS Windows AMI for Windows Server 2008 R2, launch the temporary instance using the AWS Windows AMI for Windows Server 2012 or Windows Server 2003. (To find an AMI for Windows Server 2003, search for an AMI using the name `Windows_Server-2003-R2_SP2`.)

7. Detach the root volume from the original instance as follows:
 - a. On the **Description** pane of the original instance, note the EBS ID of the volume listed as the **Root device**.
 - b. In the navigation pane, choose **Volumes**.
 - c. In the list of volumes, select the volume, and then choose **Actions, Detach Volume**. After the volume's status changes to **available**, proceed with the next step.
8. Attach the volume to the temporary instance as a secondary volume as follows:
 - a. Choose **Actions, Attach Volume**.
 - b. In the **Attach Volume** dialog box, start typing the name or ID of your temporary instance in the **Instances** field, and then select it from the list of suggested options.
 - c. In the **Device** box, type `xvd\x` (if it isn't already there), and then choose **Attach**.
 - d. Connect to the temporary instance, open the **Disk Management** utility, and bring the drive online using these instructions: [Making the Volume Available on Windows \(p. 767\)](#).
9. On the secondary volume, modify the configuration file as follows:
 - a. From the temporary instance, navigate to the secondary volume, and open `\Program Files\Amazon\Ec2ConfigService\Settings\config.xml` using a text editor, such as Notepad.
 - b. At the top of the file, find the plugin with the name `Ec2SetPassword`, as shown here. Change the state from `Disabled` to `Enabled` and then save the file.



10. (Optional) If your temporary instance is based on the same AMI that the original instance is based on, and the operating system is later than Windows Server 2003, you must complete the following steps or you won't be able to boot the original instance after you restore its root volume because of a disk signature collision.

Warning

The following procedure describes how to edit the Windows Registry using Registry Editor. If you are not familiar with the Registry or how to safely make changes using Registry Editor, read about the Registry on [Microsoft TechNet](#).

- a. Open a command prompt, type **regedit.exe**, and press Enter.
- b. In the **Registry Editor**, choose **HKEY_LOCAL_MACHINE** from the context menu (right-click), and then choose **Find**.
- c. Type **Windows Boot Manager** and then choose **Find Next**.
- d. Choose the key named 11000001. This key is a sibling of the key you found in the previous step.
- e. In the right pane, choose **Element** and then choose **Modify** from the context menu (right-click).
- f. Locate the four-byte disk signature at offset 0x38 in the data. Reverse the bytes to create the disk signature, and write it down. For example, the disk signature represented by the following data is E9EB3AA5:

```
...
0030  00 00 00 00 01 00 00 00
0038  A5 3A EB E9 00 00 00 00
0040  00 00 00 00 00 00 00 00
...
```

- g. In a Command Prompt window, run the following command to start Microsoft DiskPart.

```
C:\> diskpart
```

- h. Run the following DiskPart command to select the volume. (You can verify that the disk number is 1 using the **Disk Management** utility.)

```
DISKPART> select disk 1

Disk 1 is now the selected disk.
```

- i. Run the following DiskPart command to get the disk signature.

```
DISKPART> uniqueid disk

Disk ID: 0C764FA8
```

- j. If the disk signature shown in the previous step doesn't match the disk signature from BCD that you wrote down earlier, use the following DiskPart command to change the disk signature so that it matches:

```
DISKPART> uniqueid disk id=E9EB3AA5
```

11. Detach the secondary volume from the temporary instance as follows:

- a. Using the **Disk Management** utility, bring the volume offline.

Note

The drive is automatically offline if the temporary instance is running the same operating system as the affected instance, so you won't need to bring it offline manually.

- b. From the Amazon EC2 console, in the navigation pane, choose **Volumes**.
 - c. Select the volume, and choose **Actions, Detach Volume**. After the volume's status changes to **available**, proceed with the next step.
12. Reattach the volume to the original instance as its root volume as follows:
- a. Select the volume, and choose **Actions, Attach Volume**.
 - b. In the **Attach Volume** dialog box, start typing the name or ID of the original instance in the **Instances** list, and then select the instance.
 - c. In the **Device** box, enter `/dev/sda1`.
 - d. Choose **Yes, Attach**.
13. Restart the original instance as follows:
- a. In the navigation pane, choose **Instances**.
 - b. Select the original instance and then choose **Actions, Instance State, Start**.
 - c. In the **Start Instances** dialog box, choose **Yes, Start**.
14. Retrieve the new default password as follows:
- a. In the navigation pane, choose **Instances**.
 - b. Select the original instance and then choose **Actions, Get Windows Password**.
 - c. In the **Retrieve Default Windows Administrator Password** dialog box, choose **Browse**, and then select the `.pem` file that corresponds to the key pair that you specified when you launched the instance.
 - d. Choose **Decrypt Password**. You'll use the decrypted password to connect to the original instance using the local Administrator account.

Note

(Optional) If you completed the optional steps in this procedure to resolve the issue of a missing key pair (Step 5), then note the following:

- If your instance used an elastic IP address, you must reassign that elastic IP address to the new instance that you just created. For more information, see [Associating an Elastic IP Address with a Running Instance \(p. 713\)](#).
- Ensure that any DNS entries that referenced the public and/or private DNS or IP address point to the appropriate value.

Setting the Time for a Windows Instance

A consistent and accurate time reference is crucial for many server tasks and processes. Most system logs include a time stamp that you can use to determine when problems occur and in what order the events take place. If you use the AWS CLI or an AWS SDK to make requests from your instance, these tools sign requests on your behalf. If your instance's date and time are not set correctly, the date in the signature may not match the date of the request, and AWS rejects the request. We recommend that you use Coordinated Universal Time (UTC) for your Windows instances. However, you can use a different time zone if you want.

Contents

- [Changing the Time Zone \(p. 376\)](#)
- [Configuring Network Time Protocol \(NTP\) \(p. 376\)](#)

- [Configuring Time Settings for Windows Server 2008 and later \(p. 377\)](#)
- [Configuring Time Settings for Windows Server 2003 \(p. 378\)](#)
- [Related Topics \(p. 378\)](#)

Changing the Time Zone

Windows instances are set to the UTC time zone by default. you can change the time to correspond to your local time zone or a time zone for another part of your network.

To change the time zone on an instance

1. From your instance, open a Command Prompt window.
2. Identify the time zone to use on the instance. To get a list of time zones, use the following command: `tzutil /l`. This command returns a list of all available time zones, using the following format:

```
display name  
time zone ID
```

3. Locate the time zone ID to assign to the instance.
4. Assign the time zone to the instance by using the following command:

```
C:\> tzutil /s "Pacific Standard Time"
```

The new time zone should take effect immediately.

Configuring Network Time Protocol (NTP)

Windows instances use the time.windows.com NTP server to configure the system time; however, you can change the instance to use a different set of NTP servers if you need to. For example, if you have Windows instances that do not have Internet access, you can configure them to use an NTP server located within your private network. Your instance's security group must allow outbound UDP traffic on port 123 (NTP). The procedures in this section show how you can verify and change the NTP configuration for an instance.

To verify the NTP configuration

1. From your instance, open a Command Prompt window.
2. Get the current NTP configuration by typing the following command:

```
C:\> w32tm /query /configuration
```

This command returns the current configuration settings for the Windows instance.

3. (Optional) Get the status of the current configuration by typing the following command:

```
C:\> w32tm /query /status
```

This command returns information such as the last time the instance synced with the NTP server and the poll interval.

To change the NTP configuration

1. From the Command Prompt window, run the following command:

```
C:\> w32tm /config /manualpeerlist:comma-delimited list of NTP servers /  
syncfromflags:manual /update
```

Where *comma-delimited list of NTP servers* is the list of NTP servers for the instance to use.

2. Verify your new settings by using the following command:

```
C:\> w32tm /query /configuration
```

Configuring Time Settings for Windows Server 2008 and later

When you change the time on a Windows instance, you must ensure that the time persists through system restarts. Otherwise, when the instance restarts, it reverts back to using UTC time. For Windows Server 2008 and later, you can persist your time setting by adding a **RealTimeIsUniversal** registry key.

To set the RealTimeIsUniversal registry key

1. From the instance, open a Command Prompt window.
2. Use the following command to add the registry key:

```
C:\> reg add "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control  
\TimeZoneInformation" /v RealTimeIsUniversal /d 1 /t REG_DWORD /f
```

3. If you are using a Windows Server 2008 AMI (*not* Windows Server 2008 R2) that was created before February 22, 2013, you should verify that the Microsoft hotfix [KB2800213](#) is installed. If this hotfix is not installed, install it. This hotfix resolves a known issue in which the **RealTimeIsUniversal** key causes the Windows CPU to run at 100% during Daylight savings events and the start of each calendar year (January 1).

If you are using an AMI running Windows Server 2008 R2 (*not* Windows Server 2008), you must verify that the Microsoft hotfix [KB2922223](#) is installed. If this hotfix is not installed, install it. This hotfix resolves a known issue in which the **RealTimeIsUniversal** key prevents the system from updating the CMOS clock.

4. (Optional) Verify that the instance saved the key successfully using the following command:

```
C:\> reg query "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control  
\TimeZoneInformation" /s
```

This command returns the subkeys for the **TimeZoneInformation** registry key. You should see the **RealTimeIsUniversal** key at the bottom of the list, similar to the following:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\TimeZoneInformation  
    Bias                                REG_DWORD        0x1e0  
    DaylightBias                        REG_DWORD        0xffffffffc4  
    DaylightName                        REG_SZ           @tzres.dll,-211  
    DaylightStart                       REG_BINARY  
    00000300020002000000000000000000  
    StandardBias                       REG_DWORD        0x0
```


StandardName	REG_SZ	@tzres.dll,-212
StandardStart	REG_BINARY	00000B000100020000000000000000
TimeZoneKeyName	REG_SZ	Pacific Standard Time
DynamicDaylightTimeDisabled	REG_DWORD	0x0
ActiveTimeBias	REG_DWORD	0x1a4
RealTimeIsUniversal	REG_DWORD	0x1

Configuring Time Settings for Windows Server 2003

When you change the time zone on an instance running Windows Server 2003, you must ensure that the time persists through system restarts. Otherwise, if you restart the instance, it reverts to using the UTC clock for your time zone, resulting in a time skew that correlates with your time offset. You can persist your time setting by updating your Citrix PV drivers. For more information, see [Upgrading PV Drivers on Your Windows AMI \(p. 356\)](#).

After you update the Citrix PV drivers, the Citrix Tools for Virtual Machines Service sets the time on the instance when the service is started.

Related Topics

For more information about how the Windows operating system coordinates and manages time, including the addition of a leap second, see the following topics:

- [How the Windows Time Service Works \(TechNet\)](#)
- [W32tm \(TechNet\)](#)
- [How the Windows Time service treats a leap second \(TechNet\)](#)
- [The story around Leap Seconds and Windows: It's likely not Y2K \(blog\)](#)

Configuring a Secondary Private IPv4 Address for Your Windows Instance in a VPC

On the EC2-VPC platform, you can specify multiple private IPv4 addresses for your instances. After you assign a secondary private IPv4 address to an instance in a VPC, you must configure the operating system on the instance to recognize the secondary private IPv4 address.

Configuring the operating system on a Windows instance to recognize a secondary private IPv4 address requires the following:

- [Step 1: Configure Static IP Addressing on Your Windows Instance \(p. 379\)](#)
- [Step 2: Configure a Secondary Private IP Address for Your Windows Instance \(p. 380\)](#)
- [Step 3: Configure Applications to Use the Secondary Private IP Address \(p. 381\)](#)

Note

These instructions are based on Windows Server 2008 R2. The implementation of these steps may vary based on the operating system of the Windows instance.

Prerequisites

Before you begin, make sure you meet the following requirements:

- As a best practice, launch your Windows instances using the latest AMIs. If you are using an older Windows AMI, ensure that it has the Microsoft hot fix referenced in <http://support.microsoft.com/kb/2582281>.
- After you launch your instance in your VPC, add a secondary private IP address. For more information, see [Assigning a Secondary Private IPv4 Address \(p. 703\)](#).
- To allow Internet requests to your website after you complete the tasks in these steps, you must configure an Elastic IP address and associate it with the secondary private IP address. For more information, see [Associating an Elastic IP Address with the Secondary Private IPv4 Address \(p. 705\)](#).

Step 1: Configure Static IP Addressing on Your Windows Instance

To enable your Windows instance to use multiple IP addresses, you must configure your instance to use static IP addressing rather than a DHCP server.

Important

When you configure static IP addressing on your instance, the IP address must match exactly what is shown in the console, CLI, or API. If you enter these IP addresses incorrectly, the instance could become unreachable.

To configure static IP addressing on a Windows instance

1. Connect to your instance.
2. Find the IP address, subnet mask, and default gateway addresses for the instance by performing the following steps:
 - a. Choose **Start**. For **Search**, type `cmd` to open a command prompt window, and then press **Enter**.
 - b. At the command prompt, run the following command: `ipconfig /all`. Review the following section in your output, and note the **IPv4 Address**, **Subnet Mask**, **Default Gateway**, and **DNS Servers** values for the network interface.

```
Ethernet adapter Local Area Connection:

Connection-specific DNS Suffix . . :
Description . . . . . :
Physical Address . . . . . :
DHCP Enabled. . . . . :
Autoconfiguration Enabled . . . . :
IPv4 Address. . . . . : 10.0.0.131
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 10.0.0.1
DNS Servers . . . . . : 10.1.1.10
                          10.1.1.20
```

3. Open the **Network and Sharing Center** by running the following command from the command prompt:

```
C:\> %SystemRoot%\system32\control.exe ncpa.cpl
```

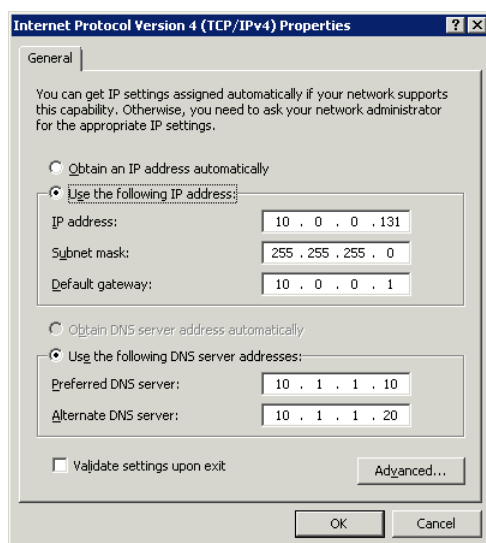
4. Open the context (right-click) menu for the network interface (Local Area Connection) and choose **Properties**.
5. Choose **Internet Protocol Version 4 (TCP/IPv4), Properties**.

6. In the **Internet Protocol Version 4 (TCP/IPv4) Properties** dialog box, choose **Use the following IP address**, enter the following values, and then choose **OK**.

Field	Value
IP address	The IPv4 address obtained in step 2 above.
Subnet mask	The subnet mask obtained in step 2 above.
Default gateway	The default gateway address obtained in step 2 above.
Preferred DNS server	The DNS server obtained in step 2 above.
Alternate DNS server	The alternate DNS server obtained in step 2 above. If an alternate DNS server was not listed, leave this field blank.

Important

If you set the IP address to any value other than the current IP address, you will lose connectivity to the instance.



You will lose RDP connectivity to the Windows instance for a few seconds while the instance converts from using DHCP to static addressing. The instance retains the same IP address information as before, but now this information is static and not managed by DHCP.

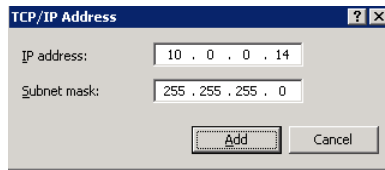
Step 2: Configure a Secondary Private IP Address for Your Windows Instance

After you have set up static IP addressing on your Windows instance, you are ready to prepare a second private IP address.

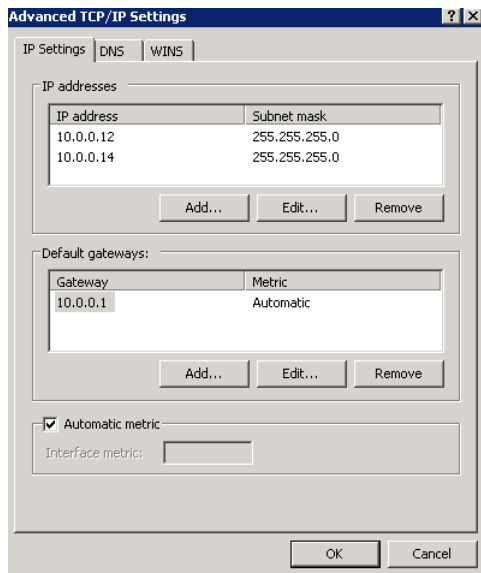
To configure a secondary IP address for a Windows instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances** and select your instance.

3. On the **Description** tab, note the secondary IP address.
4. Connect to your instance.
5. On your Windows instance, choose **Start, Control Panel**.
6. Choose **Network and Internet, Network and Sharing Center**.
7. Select the network interface (Local Area Connection) and choose **Properties**.
8. On the **Local Area Connection Properties** page, choose **Internet Protocol Version 4 (TCP/IPv4), Properties, Advanced**.
9. Choose **Add**.
10. In the **TCP/IP Address** dialog box, type the secondary private IP address for **IP address**. For **Subnet mask**, type the same subnet mask that you entered for the primary private IP address in [Step 1: Configure Static IP Addressing on Your Windows Instance](#) (p. 379), and then choose **Add**.



11. Verify the IP address settings and choose **OK**.



12. Choose **OK, Close**.
13. To confirm that the secondary IP address has been added to the operating system, at a command prompt, run the command **ipconfig /all**.

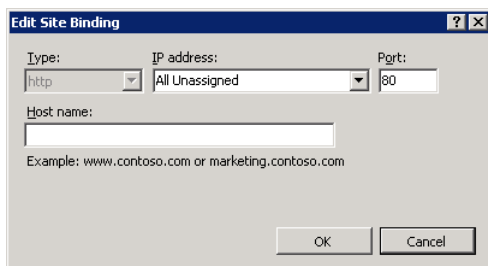
Step 3: Configure Applications to Use the Secondary Private IP Address

You can configure any applications to use the secondary private IP address. For example, if your instance is running a website on IIS, you can configure IIS to use the secondary private IP address.

To configure IIS to use the secondary private IP address

1. Connect to your instance.

2. Open Internet Information Services (IIS) Manager.
3. In the **Connections** pane, expand **Sites**.
4. Open the context (right-click) menu for your website and choose **Edit Bindings**.
5. In the **Site Bindings** dialog box, for **Type**, choose **http**, **Edit**.
6. In the **Edit Site Binding** dialog box, for **IP address**, select the secondary private IP address. (By default, each website accepts HTTP requests from all IP addresses.)



7. Choose **OK**, **Close**.

Configure a Secondary Elastic Network Interface

You can attach a second elastic network interface to the instance.

To configure a second network interface

1. Configure the static IP addressing for the primary elastic network interface as per the procedures above in [Step 1: Configure Static IP Addressing on Your Windows Instance](#) (p. 379).
2. Configure the static IP addressing for the secondary elastic network interface as per the same procedures.

Upgrading a Windows Server EC2 Instance to a Newer Version of Windows Server

This topic describes two methods for upgrading an older version of a Windows Server EC2 instance to a newer version: in-place upgrade and migration (also called side-by-side upgrade). Microsoft has traditionally recommended migrating to a newer version of Windows Server instead of upgrading. Migrating can result in fewer upgrade errors or issues, but can take longer than an in-place upgrade because of the need to provision a new instance, plan for and port applications, and adjust configurations settings on the new instance. An in-place upgrade can be faster, but software incompatibilities can produce errors.

Note

- This document does not describe how to upgrade earlier versions of Windows Server to Windows Server 2016. Upgrading to Windows Server 2016 is currently not supported.
- This topic includes information about a known issue during the upgrade process where Setup removes portions of the para-virtual (PV) drivers that enable a user to connect to the instance by using Remote Desktop.

Contents

- [In-Place Upgrade](#) (p. 383)
- [Migration](#) (p. 389)

- [Troubleshooting an Upgrade \(p. 390\)](#)

In-Place Upgrade

Before you perform an in-place upgrade of an EC2 Windows Server instance you must determine which PV drivers the instance is running. PV drivers enable you to access your instance by using Remote Desktop. Windows Server 2012 RTM and R2 instances use *AWS PV* drivers. All other Windows Server instances use *Citrix PV* drivers. For more information, see [Paravirtual Drivers \(p. 352\)](#).

Important

AWS provides upgrade support for issues or problems with the Upgrade Helper Service, an AWS utility that helps you perform in-place upgrades involving Citrix PV drivers. For all other issues or problems with an operating system upgrade or migration we recommend reviewing the TechNet articles listed in the *Before You Begin* section of this document.

This section includes the following information:

- [Before You Begin \(p. 383\)](#)
- [Upgrade instances running AWS PV drivers \(p. 384\)](#)
- [Upgrade instance running Citrix PV drivers \(p. 386\)](#)

Before You Begin

Complete the following tasks and note the following important details before you upgrade.

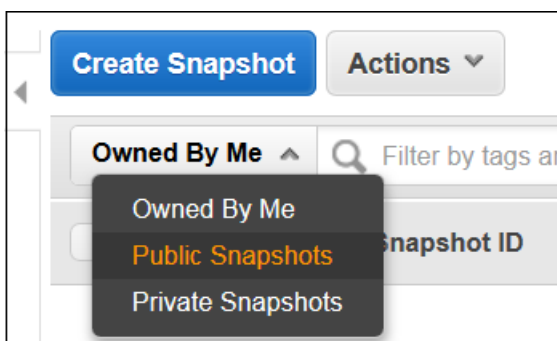
- Read the Microsoft documentation to understand the upgrade requirements, known issues, and restrictions. You should also review the official instructions for upgrading.
 - [Upgrading to Windows Server 2008](#)
 - [Upgrading to Windows Server 2008 R2](#)
 - [Upgrading to Windows Server 2012](#)
 - [Upgrading to Windows Server 2012 R2](#)
- We do not recommend performing an operating system upgrade on a T1 or T2 instance type. These types of instances might not have enough resources to manage the upgrade process. If you need to upgrade one of these instances, you must resize the instance to another instance type, perform the upgrade, and then resize it back to a T1 or T2 instance type. For more information, see [Resizing Your Instance \(p. 147\)](#).
- Create an AMI of the system you plan to upgrade for either backup or testing purposes. You can then perform the upgrade on the copy to simulate a test environment. If the upgrade completes, you can switch traffic to this instance with little downtime. If the upgrade fails, you can revert to the backup. For more information, see [Creating an Amazon EBS-Backed Windows AMI \(p. 77\)](#).
- Verify that the root volume on your Windows instance has enough free disk space. The Windows Setup process might not warn you of insufficient disk space. For information about how much disk space is required to upgrade a specific operating system, see the Microsoft documentation. If the volume does not have enough space, it can be expanded. For more information, see [Expanding the Storage Space of an EBS Volume on Windows](#).
- Determine your upgrade path. You must upgrade the operating system to the same architecture. For example, you must upgrade a 32-bit system to a 32-bit system. Windows Server 2008 R2 and later are 64-bit only.
- Disable anti-virus and anti-spyware software and firewalls. These types of software can conflict with the upgrade process. Re-enable anti-virus and anti-spyware software and firewalls after the upgrade completes.
- The Upgrade Helper Service only supports instances running Citrix PV drivers. If the instance is running Red Hat drivers, you must manually [upgrade](#) those drivers before you upgrade.

Upgrade instances running AWS PV drivers

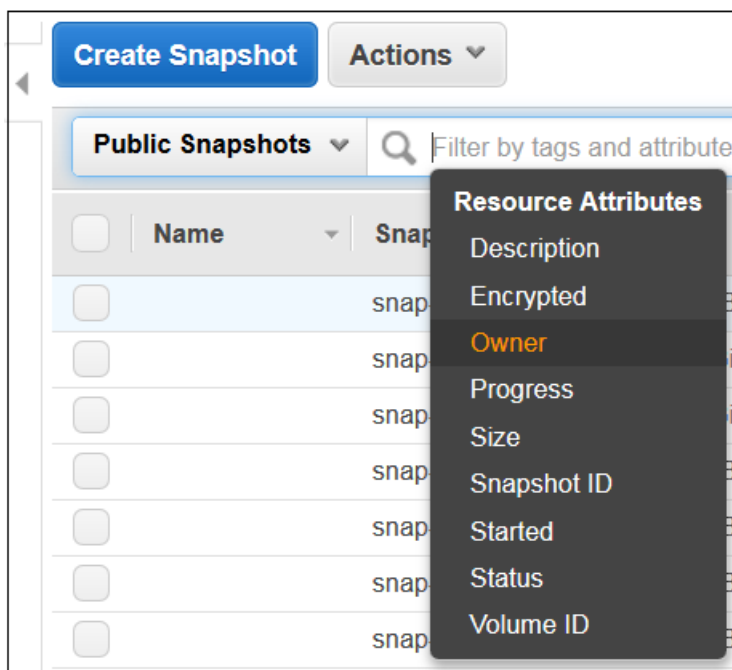
Use the following procedure to upgrade EC2 Windows Server 2012 RTM or R2 instances. These versions of EC2 Windows run AWS PV drivers and do *not* require that you run the AWS Upgrade Helper Service during your in-place upgrade.

To upgrade an EC2 Windows Server instance running AWS PV drivers

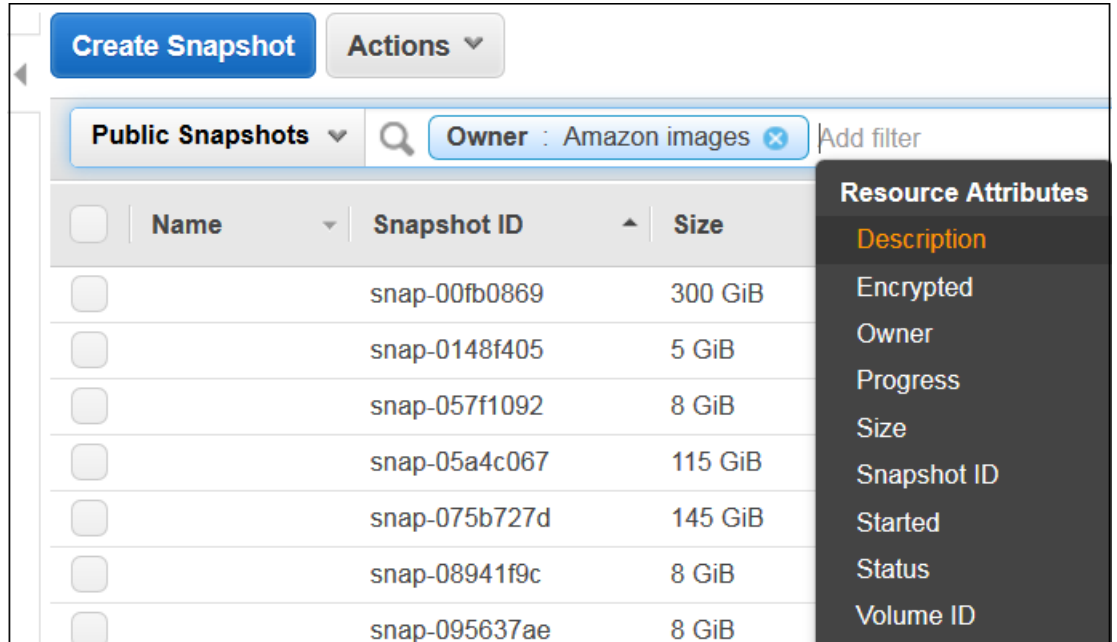
1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Instances** and then locate the Windows Server EC2 instance that you want to upgrade. Make a note of the **Instance ID** and the **Availability Zone**. You will specify this information when you create and attach the Windows installation media volume later in this procedure.
3. Create a new volume from a Windows Server installation media snapshot.
 - a. In the EC2 console, choose **Snapshots**.
 - b. Choose **Owned by me** and then choose **Public Snapshots**.



- c. Choose the **Search** field, add then choose **Owner** from the **Resource Attributes** list.



- d. Choose **Amazon images**.
- e. Choose the **Search** field, add then choose **Description** from the **Resource Attributes** list.



- f. Type **Windows** and press **Enter**.
 - g. Select the snapshot that matches your system architecture. For example, select **Windows 2012 R2 Installation Media** if your instances is currently run Windows Server 2012 RTM.
 - h. From the context menu (right-click) choose **Create Volume**.
 - i. In the **Create Volume** dialog box, choose the Availability Zone that matches your Windows instance, and then choose **Create**.
4. In the **Volume Successfully Created** message, choose the volume you just created.
 5. Choose the volume in the list, and then from the context menu (right-click) choose **Attach Volume**.
 6. In the **Attach Volume** dialog box, type the instance ID, and choose **Attach**.
 7. Begin the upgrade by using Windows Explorer to open the Installation Media volume you attached to the instance earlier in this procedure.
 8. In the **Sources** folder, run Setup.exe.
 9. On the **Select the operating system you want to install** page, select the *Full Installation* SKU that matches your Windows Server instance, and choose **Next**.
 10. On the **Which type of installation do you want?** page, choose **Upgrade**.
 11. Complete the Setup wizard.

Windows Server Setup will then copy and process files. After several minutes, your Remote Desktop session closes. The time it takes to upgrade will depend on the number of applications and server roles running on your Windows Server instance. The upgrade process could take as little as 40 minutes or as long as several hours. The instance will fail status check 1 of 2 in the EC2 console during the upgrade process. When the upgrade completes, both status checks pass. You can check the system log for console outputs or refer to Amazon CloudWatch monitors for disk and CPU activity to determine if the upgrade is not progressing.

If the instance has not passed both status checks after several hours see *Troubleshooting the Upgrade* in this topic.

Upgrade instance running Citrix PV drivers

This section describes how to upgrade EC2 Windows Server instances running Citrix PV drivers. Citrix PV drivers are used in all versions of Windows Server 2003 and 2008. There is a known issue during the upgrade process where Setup removes portions of the Citrix PV drivers that enable you to connect to the instance by using Remote Desktop. To avoid this problem, the following procedure describes how to use the Upgrade Helper Service during your in-place upgrade.

About the Upgrade Helper Service

You must run `UpgradeHelperService.exe` before you start the upgrade. After you run it, the utility creates a Windows service that executes during the post-upgrade steps to correct the driver state. The executable is written in C# and can run on .NET Framework versions 2.0 through 4.0.

When you run `UpgradeHelperService.exe` on the system *before* the upgrade it performs the following tasks:

- Creates a new Windows service called *UpgradeHelperService*.
- Verifies that Citrix PV drivers are installed.
- Checks for unsigned boot critical drivers and presents a warning if any are found. Unsigned boot critical drivers could cause system failure after the upgrade if the drivers are not compatible with the newer Windows Server version.

When you run `UpgradeHelperService.exe` on the system *after* the upgrade it performs the following tasks:

- Enables the `RealTimeUniversal` registry key for correct time synchronization in Amazon Elastic Compute Cloud (Amazon EC2).
- Restores the missing PV driver by executing the following command:

```
pnputil -i -a "C:\Program Files (x86)\Citrix\XenTools\*.inf"
```

- Installs the missing device by executing the following command:

```
C:\Temp\EC2DriverUtils.exe install "C:\Program Files (x86)\Citrix\XenTools\xevtchn.inf" ROOT\XENEVTCHN
```

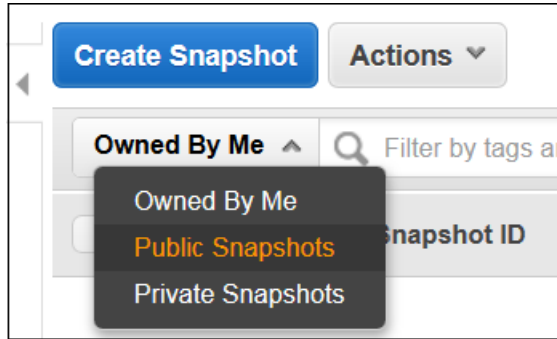
- Once complete, automatically removes the *UpgradeHelperService* Windows service.

Performing the Upgrade on Instances Running Citrix PV Drivers

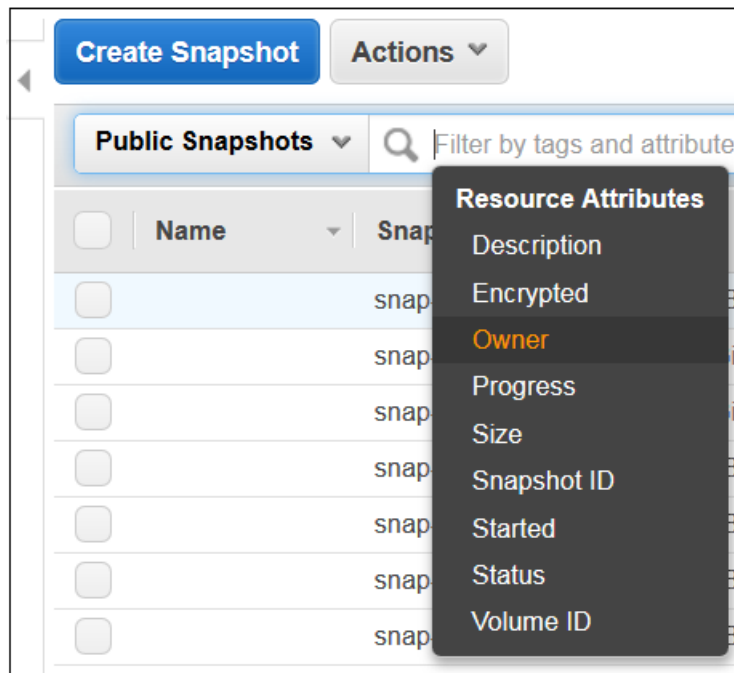
This procedure describes how to attach the installation media volume to your EC2 instance and how to upgrade the instance by using `UpgradeHelperService.exe`.

To upgrade an EC2 Windows Server instance running Citrix PV drivers

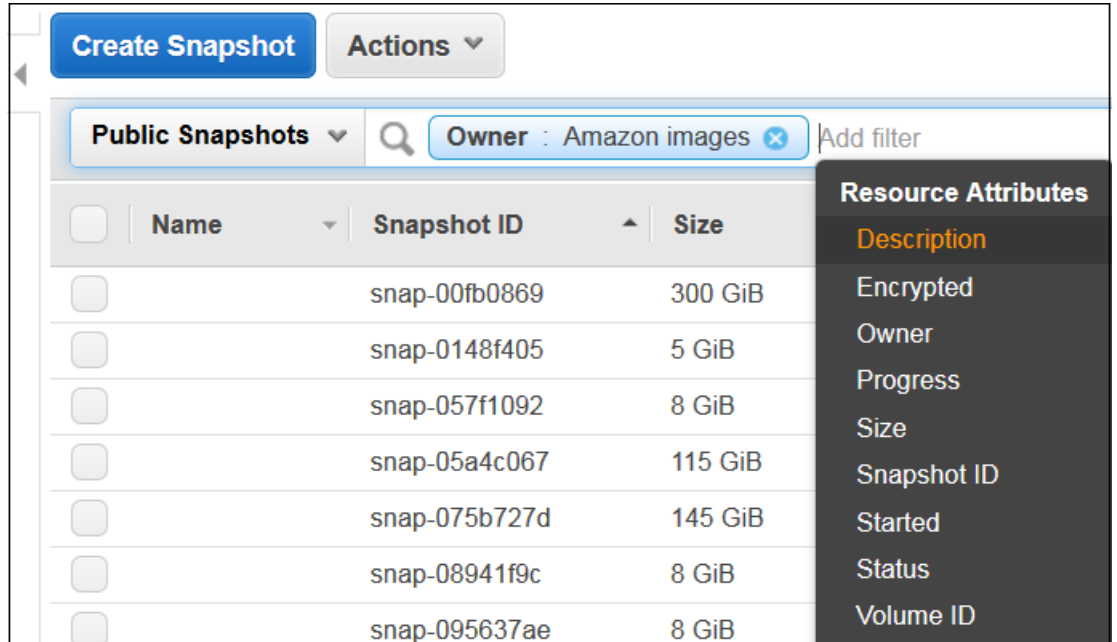
1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Instances** and then locate the Windows Server EC2 instance that you want to upgrade. Make a note of the **Instance ID** and the **Availability Zone**. You will specify this information when you create and attach the Windows installation media volume later in this procedure.
3. Create a new volume from a Windows Server installation media snapshot.
 - a. In the EC2 console, choose **Snapshots**.
 - b. Choose **Owned by me** and then choose **Public Snapshots**.



- c. Choose the **Search** field, add then choose **Owner** from the **Resource Attributes** list.



- d. Choose **Amazon images**.
- e. Choose the **Search** field, add then choose **Description** from the **Resource Attributes** list.



- f. Type **Windows** and press **Enter**.
 - g. Select the snapshot that matches your system architecture. For example, select **Windows 2008 64-bit Installation Media** if your Windows Server 2003 instance is 64-bit.
 - h. From the context menu (right-click) choose **Create Volume**.
 - i. In the **Create Volume** dialog box, choose the Availability Zone that matches your Windows instance, and then choose **Create**.
4. In the **Volume Successfully Created** message, choose the volume you just created.
 5. Choose the volume in the list, and then from the context menu (right-click) choose **Attach Volume**.
 6. In the **Attach Volume** dialog box, type the instance ID, and choose **Attach**.
 7. On your Windows instance, on the C:\ drive, create a new folder called temp. This folder must be available in the same location after the upgrade. Creating the temp folder in a Windows system folder or a user profile folder, such as the desktop, can cause the upgrade to fail.
 8. [Download OSUpgrade.zip](#) and extract the files into the C:\temp folder.
 9. Run UpgradeHelperService.exe from c:\temp and review the Log.txt file in c:\temp for any warnings.
 10. Use Microsoft [Knowledge Base article 950376](#) to uninstall PowerShell from a Windows 2003 instance, or perform the following unsupported steps to bypass the Windows Upgrade check:
 - a. In Windows Explorer, choose **WINDOWS**, and then choose **System32**.
 - b. Rename the WindowsPowerShell folder to *oldWindowsPowerShell*. For 64-bit instances, you must also rename the WindowsPowerShell folder in the **WINDOWS > SysWow64** folder.
 11. Begin the upgrade by using Windows Explorer to open the Installation Media volume you attached to the instance earlier in this procedure.
 12. In the **Sources** folder, run Setup.exe.
 13. On the **Select the operating system you want to install** page, select the *Full Installation* SKU that matches your Windows Server instance, and choose **Next**.
 14. On the **Which type of installation do you want?** page, choose **Upgrade**.
 15. Complete the Setup wizard.

Windows Server Setup will then copy and process files. After several minutes, your Remote Desktop session closes. The time it takes to upgrade will depend on the number of applications and server roles running on your Windows Server instance. The upgrade process could take as little as 40 minutes or as long as several hours. The instance will fail status check 1 of 2 in the EC2 console during the upgrade process. When the upgrade completes, both status checks pass. You can check the system log for console outputs or refer to Amazon CloudWatch monitors for disk and CPU activity to determine if the upgrade is not progressing.

If the instance has not passed both status checks after several hours see *Troubleshooting the Upgrade* in this topic.

Post Upgrade Tasks

1. Log into the instance to initiate an upgrade for the .NET Framework and reboot the system when prompted.

Note

After the upgrade, the instance might *temporarily* experience higher than average CPU utilization while the .NET Runtime Optimization service optimizes the .NET framework. This is expected behavior.

2. Install the latest version of the EC2Config service. For more information, see [Installing the Latest Version of EC2Config \(p. 295\)](#).
3. Install Microsoft hotfix [KB2800213](#).
4. Install Microsoft hotfix [KB2922223](#).
5. If you upgraded to Windows Server 2012 R2, we recommend that you upgrade the PV drivers to AWS PV drivers when they are available. For more information, see [Important information about Amazon EC2 instances running Windows Server 2012 R2](#).
6. Re-enable anti-virus and anti-spyware software and firewalls.

Migration

Migrating involves capturing settings, configurations, and data and porting these to a newer operating system on separate hardware. Once validated, the migrated system can be promoted to production. You can migrate EC2 instances by launching a new instance from an AMI of the new operating system. You can streamline the process further by using [AWS CloudFormation](#) and [Amazon EC2 Simple Systems Manager](#) to automatically apply settings and configurations to the new system with little manual work.

To migrate your server

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, click **AMIs**.
3. Choose **Owned by me**, and then choose **Public images**.
4. In the **Search** field, add the following filters and press Enter.
 - a. Owner : Amazon images
 - b. AMI Name : Windows_Server-2008

Note

The **Search** field is case sensitive.

5. Launch a new instance from an AMI.
6. Log onto the new instance and install all updates.
7. Perform application installation and configuration changes.
8. Test the server.

9. When validated, promote the server to production.

Troubleshooting an Upgrade

This section can help you locate and diagnose errors or failures.

- If the instance has not passed both status checks after several hours do the following.
 - If you upgraded to Windows Server 2008 and both status checks fail after several hours, the upgrade may have failed and be presenting a prompt to **Click OK** to confirm rolling back. Because the console is not accessible at this state, there is no way to click the button. To get around this, perform a reboot via the EC2 console or API. The reboot will take ten minutes or more to initiate. The instance might become available after 25 minutes.
 - Remove applications or server roles from the server and try again.
- If the instance does not pass both status checks after removing applications or server roles from the server, do the following.
 - Stop the instance and attach the root volume to another instance. For more information, see the description of how to stop and attach the root volume to another instance in [Waiting for the metadata service](#).
 - Analyze Windows Setup log files and event logs for failures.

Identify EC2 Instances in a Mixed Computing Environment

If you are running computer resources on another cloud infrastructure, such as Azure or Google Cloud Platform, or if you use on-premises virtualization from VMware, Xen, or KVM, you may benefit from a simple method to determine whether a virtual machine is an EC2 instance. The methods described in this topic determine optimistically whether a virtual machine is an EC2 instance by examining the Xen domain UUID. The UUID of a non-EC2 virtual machine is less likely to contain "ec2" as its first three characters.

Note

There is a small chance that a Xen instance not in EC2 could also begin with these characters.

You can discover the Xen UUID using the following approaches:

- On a Linux VM, run the following command:

```
$ cat /sys/hypervisor/uuid
```

This returns a UUID:

```
ec2e1916-9099-7caf-fd21-012345abcdef
```

In this example, the prepended "ec2" indicates that you are probably looking at an EC2 instance.

- Alternatively, on HVM instances only, the Desktop Management Interface (DMI) contains the same UUID as the System Serial Number and the System UUID (capitalized):

```
$ sudo dmidecode --string system-serial-number  
ec2e1916-9099-7caf-fd21-32803a1d3c6b
```

```
$ sudo dmidecode --string system-uuid  
EC2E1916-9099-7CAF-FD21-32803A1D3C6B
```

Note

Unlike the previous method, the DMI method requires superuser privileges. However, some older Linux kernels may not expose the UUID via `/sys/`.

You can also use this method on a Windows VM using the Windows Management Instrumentation command line (WMIC):

```
C:\>wmic path win32_computersystemproduct get uuid
```

Or, you can use PowerShell:

```
PS C:\>Get-WmiObject -query "select uuid from Win32_ComputerSystemProduct" |  
Select UUID
```

- For a cryptographically verified method, check the instance identity document, including its signature. For more information, see [Instance Identity Documents](#).

Amazon EC2 Systems Manager

Amazon EC2 Systems Manager is a collection of capabilities that helps you automate management tasks such as collecting system inventory, applying operating system (OS) patches, automating the creation of Amazon Machine Images (AMIs), and configuring operating systems (OSs) and applications at scale. Systems Manager works with managed instances: Amazon EC2 instances, or servers and virtual machines (VMs) in your on-premises environment that are configured for Systems Manager.

Note

Systems Manager features and shared components are offered at no additional cost. You pay only for the EC2 resources that you use.

Information for Linux Users

See [Amazon EC2 Systems Manager](#) in the *Amazon EC2 User Guide for Linux Instances*.

Systems Manager simplifies the following tasks.

Tasks	Details
Remote Administration (p. 437)	Run Command lets you remotely and securely manage the configuration of your managed instances at scale. Use Run Command to perform ad hoc changes like updating applications or running Windows PowerShell commands on a target set of dozens or hundreds of instances.
Inventory Management (p. 515)	Inventory Manager automates the process of collecting software inventory from managed instances. You can use Inventory Manager to gather metadata about OS and system configurations and application deployments.
State Management (p. 522)	State Manager automates the process of keeping your managed instances in a defined state. You can use State Manager to ensure that your instances are bootstrapped with specific software at startup, joined to a Windows domain, or patched with specific software updates.

Tasks	Details
Maintenance and Deployment Automation (p. 530)	Automation automates common maintenance and deployment tasks. You can use Automation to create and update Amazon Machine Images, apply driver and agent updates, and apply OS patches or application updates.

Systems Manager also includes the following shared components to help you efficiently administer managed instances while minimizing the impact to them.

Component	Details
Maintenance Windows (p. 410)	Maintenance Windows let you set up recurring schedules for managed instances to execute administrative tasks like installing patches and updates without interrupting business-critical operations.
Parameter Store (p. 429)	Parameter Store centralizes the management of configuration data. You can use Parameter Store to store passwords, license keys, or database connection strings that you commonly reference in scripts, commands, or other automation and configuration workflows.

Getting Started

Use the following task list to get started with Systems Manager.

1. Complete the Systems Manager walkthroughs in a test environment. These walkthroughs describe how to configure roles and permissions and use Systems Manager features on an EC2 instance.
 - [Maintenance Windows \(p. 413\)](#)
 - [Parameter Store \(p. 435\)](#)
 - [Run Command \(p. 41\)](#)
 - [Inventory Manager \(p. 518\)](#)
 - [Automation \(p. 532\)](#)
 - [State Manager \(p. 527\)](#)
2. Verify [prerequisites \(p. 394\)](#) for your EC2 instances and on-premises servers or VMs.
3. Create a managed instance [activation \(p. 397\)](#) (on-premises servers and VMs only).
4. Configure user and instance [roles and permissions \(p. 400\)](#). The roles and permissions described in the walkthroughs are not restrictive. Use the information in *Configuring Access to Systems Manager* to create more restrictive roles and permissions for your production machines.

Contents

- [Systems Manager Prerequisites \(p. 394\)](#)
- [Installing and Updating SSM Agent \(p. 396\)](#)
- [Setting Up Systems Manager in Hybrid Environments \(p. 397\)](#)
- [Configuring Access to Systems Manager \(p. 400\)](#)
- [Systems Manager Shared Components \(p. 407\)](#)

- [Remote Management \(p. 437\)](#)
- [Inventory Management \(p. 515\)](#)
- [State Management \(p. 522\)](#)
- [Maintenance and Deployment Automation \(p. 530\)](#)

Systems Manager Prerequisites

Amazon EC2 Systems Manager includes the following prerequisites.

Information for Linux Users

See [Amazon EC2 Systems Manager Prerequisites](#) in the *Amazon EC2 User Guide for Linux Instances*.

Limitations

Systems Manager is [only available in these regions](#).

Note

For servers and VMs in your hybrid environment, we recommend that you choose the region closest to your data center or computing environment.

Prerequisites

Requirement	Details	For More Information
Supported Operating System	Instances must be running a supported version of Windows Server. Supported versions include Windows Server 2003 - 2016, including all R2 versions.	Finding a Windows AMI (p. 67)
SSM Agent for servers and VMs in your hybrid environment	You must download and install SSM Agent on servers and VMs in your hybrid environment.	Install the SSM Agent on Servers and VMs in Your Hybrid Environment (p. 400)
SSM Agent	<p>For the following Amazon EC2 instances, SSM Agent processes Systems Manager requests and configures your machine as specified in the request.</p> <ul style="list-style-type: none">• Windows Server 2016 instances• Instances created from Windows Server 2003-2012 R2 Amazon Machine Images (AMIs) published in November 2016 or later <p>SSM Agent is installed by default on Windows Server 2016 instances and instances created from Windows Server 2003-2012 R2 AMIs published in November 2016 or later.</p>	Installing and Updating SSM Agent (p. 396)

Amazon Elastic Compute Cloud
User Guide for Windows Instances
Prerequisites

Requirement	Details	For More Information
EC2Config service	<p>On Windows Server 2003-2012 R2 instances published before November 2016, the EC2Config service processes Systems Manager requests and configures your machine as specified in the request. In order to use Systems Manager features published after November 2016, you must upgrade the EC2Config service to version 2.0.533.0 or later. Upgrading EC2Config also installs the latest version of SSM Agent side-by-side with EC2Config.</p>	<p>Updating the SSM Agent Using Amazon EC2 Run Command (p. 470)</p>
Access to Systems Manager	<p>Before you can execute commands using Systems Manager, you must configure an AWS Identity and Access Management (IAM) EC2 instance role for instances that will process commands. You must also configure a separate user role for users executing commands. Both roles require permission policies that enable them to communicate with the SSM API.</p> <p>Note For servers and VMs in your hybrid environment, you must also create an IAM service role that enables your on-premises server or VM or VM hosted by another cloud provider to communicate with the SSM service. For more information, see Create an IAM Service Role (p. 398).</p>	<p>Configuring Access to Systems Manager (p. 400)</p>
Internet Access	<p>Verify that your EC2 instances have outbound Internet access. Inbound Internet access is not required.</p>	<p>Internet Gateways</p>

Requirement	Details	For More Information
Amazon S3 Bucket (Optional)	You can store System Manager output in an Amazon Simple Storage Service (S3) bucket. Output in the Amazon EC2 console is truncated after 2500 characters. Additionally, you might want to create an Amazon S3 key prefix (a subfolder) to help you organize output.	Create a Bucket

Note

SSM communicates with the SSM agent or the EC2Config service on your instance by using the EC2 Messaging service. If you monitor traffic, you will see your instances communicating with `ec2messages.*` endpoints.

Installing and Updating SSM Agent

An SSM Agent running on your instances processes Systems Manager requests and configures your machines as specified in the request. This is true for Windows Server 2016 instances or Windows Server 2003-2012 R2 instances created from November 2016 or later Amazon Machine Images (AMIs). SSM Agent is installed, by default, on these AMIs.

For Windows Server 2003-2012 R2 instances created from AMIs published *before* November 2016, EC2Config processes SSM requests.

Information for Linux Users

See [Installing SSM Agent](#) in the *Amazon EC2 User Guide for Linux Instances*.

If EC2Config processes Systems Manager requests on your instances, then we recommend that you upgrade your existing instances to use the latest version of EC2Config. By upgrading to the latest version, you install SSM Agent side-by-side with EC2Config. This version of SSM Agent is compatible with your instances created from earlier Windows AMIs and enables you to use SSM features published after November 2016. You can update EC2Config and install SSM Agent by using Run Command. For more information, see [Updating the SSM Agent Using Amazon EC2 Run Command](#) (p. 470).

Configuring SSM Agent to Use a Proxy

For information about configuring EC2Config to use a proxy, see [Configure Proxy Settings for the EC2Config Service](#) (p. 293).

To configure SSM Agent to use a proxy

1. Using Remote Desktop or Windows PowerShell, connect to the instance that you would like to configure to use a proxy. For Windows Server 2016 instances that use the Nano installation option (Nano Server), you must connect using PowerShell. For more information, see [Connect to a Windows Server 2016 Nano Server Instance](#) (p. 257).
2. If you connected using Remote Desktop, then launch PowerShell as an administrator.
3. Run the following command block in PowerShell:

```
$serviceKey = "HKLM:\SYSTEM\CurrentControlSet\Services\AmazonSSMAgent "
```

```
$proxyVariables = @"http_proxy=hostname:port ",  
"no_proxy=169.254.169.254"  
New-ItemProperty -Path $serviceKey -Name Environment -Value  
$proxyVariables -PropertyType MultiString  
Restart-Service AmazonSSMAgent
```

To reset the SSM agent proxy configuration

1. Using Remote Desktop or Windows PowerShell, connect to the instance that you would like to configure.
2. If you connected using Remote Desktop, then launch PowerShell as an administrator.
3. Run the following command block in PowerShell:

```
Remove-ItemProperty -Path HKLM:\SYSTEM\CurrentControlSet\Services  
\AmazonSSMAgent -Name Environment  
Restart-Service AmazonSSMAgent
```

Setting Up Systems Manager in Hybrid Environments

Amazon EC2 Systems Manager lets you remotely and securely manage on-premises servers and virtual machines (VMs) and VMs from other cloud providers. By setting up Systems Manager in this way, you do the following.

- Create a consistent and secure way to remotely manage your on-premises and cloud workloads from one location using the same tools or scripts.
- Centralize access control for actions that can be performed on your servers and VMs by using AWS Identity and Access Management (IAM).
- Centralize auditing and your view into the actions performed on your servers and VMs because all actions are recorded in AWS CloudTrail.
- Centralize monitoring because you can configure CloudWatch Events and Amazon SNS to send notifications about service execution success.

After you set up your hybrid machines for Systems Manager, they are listed in the EC2 console and called *managed instances*, like other EC2 instances.

Information for Linux Users

See [Setting Up Systems Manager in Hybrid Environments](#) in the *Amazon EC2 User Guide for Linux Instances*.

Contents

- [Create an IAM Service Role \(p. 398\)](#)
- [Create a Managed-Instance Activation \(p. 399\)](#)
- [Install the SSM Agent on Servers and VMs in Your Hybrid Environment \(p. 400\)](#)

To get started using Systems Manager in hybrid environments

1. **Create IAM service and user roles:** The IAM *service* role enables your servers and VMs in your hybrid environment to communicate with the Systems Manager SSM service. The IAM *user* role enables users to communicate with the SSM API to execute commands from either the Amazon

EC2 console or by directly calling the API. Creating the service role is described later in this topic. That section includes a link to a topic with information about how to create a user role.

2. **Verify prerequisites:** Verify that your servers and VMs in your hybrid environment meet the minimum requirements for Systems Manager. For more information, see [Systems Manager Prerequisites](#) (p. 394).
3. **Create a managed-instance activation:** A managed-instance activation registers one or more servers and VMs in your hybrid environment with Systems Manager. Creating a managed-instance activation is described later in this topic.
4. **Deploy SSM Agent:** SSM Agent processes Systems Manager requests and configures your machine as specified in the request. You must download and install SSM Agent on servers and VMs in your hybrid environment, as described later in this topic.

Create an IAM Service Role

Servers and VMs in a hybrid environment require an IAM role to communicate with the Systems Manager SSM service. The role grants AssumeRole trust to the SSM service.

The following example shows how to create the IAM service role for Systems Manager using AWS Tools for Windows PowerShell. For examples of how to create a service role using the AWS Management Console or the AWS CLI, see [Creating a Role to Delegate Permissions to an AWS Service](#).

Note

You only need to create the service role once for each AWS account.

To create an IAM service role for servers and VMs in your hybrid environment using AWS Tools for Windows PowerShell

1. Create a text file (in this example it is named `SSMService-Trust.json`) with the following trust policy. Save the file with the `.json` file extension.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Principal": {"Service": "ssm.amazonaws.com"},
    "Action": "sts:AssumeRole"
  }
}
```

2. Use [New-IAMRole](#) as follows to create a service role. This example creates a role named `SSMServiceRole`.

```
New-IAMRole -RoleName SSMServiceRole -AssumeRolePolicyDocument (Get-Content -raw SSMService-Trust.json)
```

3. Use [Register-IAMRolePolicy](#) as follows to enable the `SSMServiceRole` to create a session token. The session token gives your managed instance permission to execute commands using Systems Manager.

```
Register-IAMRolePolicy -RoleName SSMServiceRole -PolicyArn
arn:aws:iam::ssm:policy/service-role/AmazonEC2RoleforSSM
```

You must now create IAM roles that enable users to communicate with the SSM API. For more information, see [Configuring Access to Systems Manager](#) (p. 400).

Create a Managed-Instance Activation

To set up servers and VMs in your hybrid environment as managed instances, you need to create a managed-instance activation. After you complete the activation, you receive an activation code and ID. This code/ID combination functions like an Amazon EC2 access ID and secret key to provide secure access to the Systems Manager service from your managed instances.

Important

Store the managed-instance activation code and ID in a safe place. You specify this code and ID when you install the SSM agent on servers and VMs in your hybrid environment. If you lose the code and ID, you must create a new activation.

The procedures in this section require that you specify a [region where SSM is available](#). We recommend that you specify the region closest to your data center or computing environment.

Note

When you create a managed-instance activation, you specify a date when the activation expires. If you want to register additional managed instances after the expiry date, you must create a new activation. The expiry date has no impact on registered and running instances.

To create a managed-instance activation using the console

1. Open the [Amazon EC2 console](#), expand **Commands** in the navigation pane, and choose **Activations**.
2. Choose **Create an Activation**.
3. Fill out the form and choose **Create Activation**.

To create a managed-instance activation using the AWS Tools for Windows PowerShell

1. On a machine where you have installed AWS Tools for Windows PowerShell, execute the following command in AWS Tools for Windows PowerShell.

```
New-SSMActivation -DefaultInstanceName name -IamRole IAM service role -  
RegistrationLimit number of managed instances -Region region
```

For example:

```
New-SSMActivation -DefaultInstanceName MyWebServers -IamRole  
RunCommandServiceRole -RegistrationLimit 10 -Region us-east-1
```

2. Press Enter. If the activation is successful, the system returns an activation code and an ID. Store the activation code and ID in a safe place.

To create a managed-instance activation using the AWS CLI

1. On a machine where you have installed the AWS Command Line Interface (AWS CLI), execute the following command in the CLI.

```
aws ssm create-activation --default-instance-name name --iam-role IAM  
service role --registration-limit number of managed instances --  
region region
```

For example:

```
aws ssm create-activation --default-instance-name MyWebServers --iam-role  
RunCommandServiceRole --registration-limit 10 --region us-east-1
```

2. Press Enter. If the activation is successful, the system returns an activation code and an ID. Store the activation code and ID in a safe place.

Install the SSM Agent on Servers and VMs in Your Hybrid Environment

Before you begin, locate the activation code and ID that was sent to you after you completed the managed-instance activation. You will specify the code and ID in the following procedure.

To install the SSM agent on servers and VMs in your hybrid environment

1. Log on to a server or VM in your hybrid environment.
2. Open Windows PowerShell.
3. Copy and paste the following command block into AWS Tools for Windows PowerShell. Specify your activation code, activation ID, and the region where you want to download the SSM agent from. For *region*, choose a [region where SSM is available](#). For example, us-west-2.

```
$dir = $env:TEMP + "\ssm"  
New-Item -ItemType directory -Path $dir  
cd $dir  
(New-Object System.Net.WebClient).DownloadFile("https://amazon-  
ssm-region.s3.amazonaws.com/latest/windows_amd64/AmazonSSMAgentSetup.exe",  
$dir + "\AmazonSSMAgentSetup.exe")  
Start-Process .\AmazonSSMAgentSetup.exe -ArgumentList @("/q", "/log",  
"install.log", "CODE=code", "ID=id", "REGION=region") -Wait  
Get-Content ($env:ProgramData + "\Amazon\SSM\InstanceData\registration")  
Get-Service -Name "AmazonSSMAgent"
```

4. Press Enter.

The command downloads and installs the SSM agent onto the server or VM. The command also registers the server or VM with the SSM service. The server or VM is now a managed instance. In the console, these instances are listed with the prefix "mi-". You can view all instances using a `List` command. For more information, see the [Amazon EC2 Simple Systems Manager API Reference](#).

Configuring Access to Systems Manager

Amazon EC2 Systems Manager requires an IAM role for EC2 instances that will process commands and a separate role for users executing commands. Both roles require permission policies that enable them to communicate with the [SSM API](#). You can choose to use SSM managed policies or you can create your own roles and specify permissions as described in this section.

If you are configuring on-premises servers or VMs or VMs hosted by other cloud providers that you want to configure using Systems Manager you must also configure an IAM service role. For more information, see [Create an IAM Service Role](#) (p. 398).

Information for Linux Users

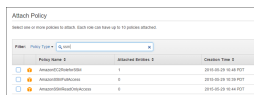
See [Configuring Access to Systems Manager](#) in the *Amazon EC2 User Guide for Linux Instances*.

Contents

- [Use SSM Managed Policies \(p. 401\)](#)
- [Configure Your Own Roles and Policies \(p. 401\)](#)
- [Create EC2 Instances that Use the EC2 Instance Role \(p. 407\)](#)

Use SSM Managed Policies

IAM managed policies for SSM can help you quickly configure access and permissions for Systems Manager users and instances. You can find these policies in the **Policies** page of the IAM console by searching for SSM, as shown in the following screen shot.



The managed policies perform the following functions:

- **AmazonEC2RoleforSSM (instance trust policy):** Enables an instance to communicate with the Run Command API.
- **AmazonSSMFullAccess (user trust policy):** Grants the user access to the SSM API and SSM JSON documents. Assign this policy to administrators and trusted power users.
- **AmazonSSMReadOnlyAccess (user trust policy):** Grants the user access to read-only API actions, such as Get and List.

For information about how to configure these policies, see [Managed Policies and Inline Policies](#).

Configure Your Own Roles and Policies

If you choose not to use SSM managed policies, then use the following procedures to create and configure an SSM EC2 instance role and an SSM user account.

Important

If you want to use an existing EC2 instance role and user account, you must attach the policies shown in this section to the role and the user account. You must also verify that `ec2.amazonaws.com` is listed in the trust policy for the EC2 instance role. For more information, see [Verify the Trust Policy \(p. 406\)](#).

Topics

- [Create the IAM Policy for EC2 Instances \(p. 401\)](#)
- [Create the IAM User Policy \(p. 403\)](#)
- [Create a Restrictive IAM User Policy \(p. 404\)](#)
- [Create the EC2 Instance Role \(p. 406\)](#)
- [Verify the Trust Policy \(p. 406\)](#)
- [Create the User Account \(p. 406\)](#)

Create the IAM Policy for EC2 Instances

The following IAM policy enables EC2 instances to communicate with the Run Command API. You will create the role and attach this policy to that role later in this topic.

To create an IAM policy for EC2 instances

1. Open the Identity and Access Management (IAM) console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Policies**. (If this is your first time using IAM, choose **Get Started**, and then choose **Create Policy**.)
3. Beside **Create Your Own Policy**, choose **Select**.
4. Type a policy name (for example, *SystemsManagerInstance*) and description, and then copy and paste the following policy into the **Policy Document** field:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssm:DescribeAssociation",
        "ssm:GetDocument",
        "ssm:ListAssociations",
        "ssm:UpdateAssociationStatus",
        "ssm:UpdateInstanceInformation"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2messages:AcknowledgeMessage",
        "ec2messages:DeleteMessage",
        "ec2messages:FailMessage",
        "ec2messages:GetEndpoint",
        "ec2messages:GetMessages",
        "ec2messages:SendReply"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:PutMetricData"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstanceStatus"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ds:CreateComputer",
        "ds:DescribeDirectories"
      ],
      "Resource": "*"
    }
  ],
}
```

```
{
  "Effect": "Allow",
  "Action": [
    "logs:CreateLogGroup",
    "logs:CreateLogStream",
    "logs:DescribeLogGroups",
    "logs:DescribeLogStreams",
    "logs:PutLogEvents"
  ],
  "Resource": "*"
},
{
  "Effect": "Allow",
  "Action": [
    "s3:PutObject",
    "s3:GetObject",
    "s3:AbortMultipartUpload",
    "s3:ListMultipartUploadParts",
    "s3:ListBucketMultipartUploads"
  ],
  "Resource": "*"
}
]
```

Important

In the last section of this IAM policy, you can restrict access to the Amazon S3 bucket by specifying an Amazon Resource Name (ARN). For example, you can change the last `"Resource": "*" item to "Resource": "arn:aws:s3:::AnS3Bucket/*"`

5. Choose **Validate Policy**. Verify that the policy is valid. If you receive an error, verify that you included the opening and closing brackets { }. After the policy is validated, choose **Create Policy**.

Create the IAM User Policy

The IAM user policy determines which SSM documents a user can see in the **Command document** list. Users can see this list in either the Amazon EC2 console or by calling `ListDocuments` using the AWS CLI or AWS Tools for Windows PowerShell. The policy also limits the actions the user can perform with an SSM JSON document.

Note

You will create a user account and attach this policy to that account later on.

The IAM policy in the following procedure enables the user to perform any SSM action on the instance. Assign this policy only to trusted administrators. For all other users, create a restrictive IAM policy, as described in this section.

To create the IAM user policy

1. Repeat the previous procedure to create a policy for a user.
2. Copy and paste the following policy into the **Policy Document** field and create the policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
```

```
        "ssm:*",
        "ec2:DescribeInstanceStatus"
    ],
    "Resource": "*"
}
]
```

Create a Restrictive IAM User Policy

Create restrictive IAM user policies to further delegate access to Run Command. The following example IAM policy allows a user to list SSM JSON documents and view details about those documents, send a command using the RestartService document, and cancel or view details about the command after it has been sent. The user has permission to execute the RestartService document on three instances, as determined by the "arn:aws:ec2:us-east-1:*:instance/i-xxxxxxxxxxxxxxxx" items in the second Resource section. If you want to give the user access to run the command on any instance for which the user currently has access (as determined by the AWS user account), you could specify "arn:aws:ec2:us-east-1:*:instance/*" in the Resource section and remove the other instance resources.

Note that the Resource section includes an S3 ARN entry:

```
arn:aws:s3:::bucket_name
```

You can also format this entry as follows:

```
arn:aws:s3:::bucket_name/*
-or-
arn:aws:s3:::bucket_name/key_prefix_name
```

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ssm:ListDocuments",
        "ssm:DescribeDocument",
        "ssm:GetDocument",
        "ssm:DescribeInstanceInformation"
      ],
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Action": "ssm:SendCommand",
      "Effect": "Allow",
      "Resource": [
        "arn:aws:ec2:us-east-1:*:instance/i-1234567890abcdef0",
        "arn:aws:ec2:us-east-1:*:instance/i-0598c7d356eba48d7",
        "arn:aws:ec2:us-east-1:*:instance/i-345678abcdef12345",
        "arn:aws:s3:::bucket_name",
        "arn:aws:ssm:us-east-1:*:document/RestartService"
      ]
    }
  ]
}
```

```
        "Action": [
            "ssm:CancelCommand",
            "ssm:ListCommands",
            "ssm:ListCommandInvocations"
        ],
        "Effect": "Allow",
        "Resource": "*"
    },
    {
        "Action": "ec2:DescribeInstanceStatus",
        "Effect": "Allow",
        "Resource": "*"
    }
]
}
```

The following IAM policy document enables the user to install, uninstall, or repair applications using the AWS-InstallApplication SSM document.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ssm:ListDocuments",
        "ssm:DescribeDocument",
        "ssm:GetDocument",
        "ssm:DescribeInstanceInformation"
      ],
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Action": "ssm:SendCommand",
      "Effect": "Allow",
      "Resource": [
        "arn:aws:ssm:us-east-1:*:document/AWS-InstallApplication",
        "arn:aws:ec2:us-east-1:*:instance/*"
      ]
    },
    {
      "Action": [
        "ssm:CancelCommand",
        "ssm:ListCommands",
        "ssm:ListCommandInvocations"
      ],
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Action": "ec2:DescribeInstanceStatus",
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

For more information about creating IAM user policies, see [Managed Policies and Inline Policies](#).

Create the EC2 Instance Role

The EC2 Instance role enables the instance to communicate with the SSM API. The role uses the EC2 instance policy you created earlier.

To create the EC2 instance role

1. In the navigation pane of the IAM console, choose **Roles**, and then choose **Create New Role**.
2. On the **Set Role Name** page, enter a name for the role that designates it as the instance role, for example, *SystemsManagerInstance*. Choose **Next Step**.
3. On the **Select Role Type** page, choose **Select** next to **Amazon EC2**.
4. On the **Attach Policy** page, select the *SystemsManagerInstance* policy you created earlier. Choose **Next Step**.
5. Review the role information and then choose **Create Role**.

Verify the Trust Policy

If you want to use an existing EC2 instance role, you must verify that `ec2.amazonaws.com` is listed in the trust policy for the role. If you created a new EC2 instance role, you must add `ec2.amazonaws.com` as a trusted entity.

To verify the trust policy

1. In the navigation pane of the IAM console, choose **Roles**, and then choose the server role you just created.
2. Choose **Trust Relationships**.
3. Under **Trusted Entities** verify that `ec2.amazonaws.com` is listed. If it's not listed, choose **Edit Trust Relationship**.
4. Copy and paste the following policy into the **Policy Document** field and create the policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Create the User Account

The user account enables a user to call the SSM API on an instance. This account uses the IAM user policy you created earlier.

To create the user account

1. From the **Users** page on the [IAM console](#), choose **Create New Users**.

2. Specify a user name (for example, *SystemsManagerUser*) and verify that the **Generate an access key for each user** option is selected.
3. Choose **Create**.
4. Choose **Download Credentials**. By default, the system prompts you to save the credentials as a .csv file.
Important
Make a note of the *SystemsManagerUser* access key and secret key from the .csv file you downloaded.
5. Choose **Close**.
6. In the **IAM Dashboard**, choose **Users**, and then locate the user you just created.
7. Choose the user name (do *not* select the option beside the name), and then choose **Attach Policy**.
8. Choose the user policy you created earlier, and then choose **Attach Policy**.

Create EC2 Instances that Use the EC2 Instance Role

This procedure describes how to create an EC2 instance that uses the role you created. You must assign a role to an EC2 instance when you launch it. You can't assign a role to an instance that is already running. Instead, you would create an image of the instance, and then launch an instance from that image, with the role assigned.

To create an instance that use the EC2 instance role

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Select a supported [region](#).
3. Choose **Launch Instance** and select a Windows Server instance.
4. Choose your instance type and then choose **Next: Configure Instance Details**.
5. In the **IAM role** drop-down list choose the EC2 instance role you created earlier.
6. Complete the wizard.

If you create other instances that you want to configure using Systems Manager, you must specify the EC2 instance role for each instance.

Systems Manager Shared Components

Amazon EC2 Systems Manager includes the following shared components. These components are designed to standardize, automate, and simplify the process of administering managed instances.

Note

Systems Manager features and shared components are offered at no additional cost. You pay only for the EC2 resources that you use.

Information for Linux Users

See [Systems Manager Shared Components](#) in the *Amazon EC2 User Guide for Linux Instances*.

Before you begin

Verify that your EC2 instances and on-premises servers or virtual machines meet Systems Manager prerequisites. For more information, see [Systems Manager Prerequisites](#) (p. 394).

Contents

- [Specifying a Cron Schedule for Your Systems Manager Shared Components \(p. 408\)](#)
- [Systems Manager Maintenance Windows \(p. 410\)](#)
- [Systems Manager Parameter Store \(p. 429\)](#)

Specifying a Cron Schedule for Your Systems Manager Shared Components

When you create a Systems Manager Maintenance Window or an association using Systems Manager State Manager, you specify a schedule for when the window/association should run. System Manager lets you specify a schedule in the form of either a time-based entry, called a *cron expression* or a frequency-based entry, called a *rate expression*.

Example: This cron expression runs the Maintenance Window or the association at 4 PM (16:00) every Tuesday: `cron(0 16 ? * TUE *)`

In the AWS CLI, specify this expression using the `--schedule` parameter as follows:

```
--schedule "cron(0 16 ? * TUE *)"
```

Example: This rate expression runs the Maintenance Window or the association every other day: `rate(2 days)`

In the AWS CLI, specify this expression using the `--schedule` parameter as follows:

```
--schedule "rate(2 days)"
```

Cron expressions have six required fields. Fields are separated by white space.

Minutes	Hours	Day of month	Month	Day of week	Year	Meaning
0	10	*	*	?	*	Run at 10:00 am (UTC) every day
15	12	*	*	?	*	Run at 12:15 PM (UTC) every day
0	18	?	*	MON-FRI	*	Run at 6:00 PM (UTC) every Monday through Friday
0	8	1	*	?	*	Run at 8:00 AM (UTC) every 1st day of the month
0/15	*	*	*	?	*	Run every 15 minutes

Minutes	Hours	Day of month	Month	Day of week	Year	Meaning
0/10	*	?	*	MON-FRI	*	Run every 10 minutes Monday through Friday
0/5	8-17	?	*	MON-FRI	*	Run every 5 minutes Monday through Friday between 8:00 AM and 5:55 PM (UTC)

The following table shows more examples of cron expressions:

Cron Expression Example	Runs At
0 0 2 ? 1/1 THU#3 *	02:00 AM the third Thursday of every month
0 15 10 ? * *	10:15 AM every day
0 0 0 21 1/1 ? *	midnight on the 21st of each month
0 15 10 ? * MON-FRI	10:15 AM every Monday, Tuesday, Wednesday, Thursday and Friday
0 0 2 L * ?	02:00 AM on the last day of every month
0 15 10 ? * 6L	10:15 AM on the last Friday of every month

The following table shows supported values for required cron entries:

Field	Values	Wildcards
Minutes	0-59	, - * /
Hours	0-23	, - * /
Day-of-month	1-31	, - * ? / L W
Month	1-12 or JAN-DEC	, - * /
Day-of-week	1-7 or SUN-SAT	, - * ? / L
Year	1970-2199	, - * /

Note

You cannot specify a value in the Day-of-month and in the Day-of-week fields in the same cron expression. If you specify a value in one of the fields, you must use a ? (question mark) in the other field.

Wildcards

Cron expressions support the following wildcards:

- The , (comma) wildcard includes additional values. In the Month field, JAN,FEB,MAR would include January, February, and March.
- The - (dash) wildcard specifies ranges. In the Day field, 1-15 would include days 1 through 15 of the specified month.
- The * (asterisk) wildcard includes all values in the field. In the Hours field, * would include every hour.
- The / (forward slash) wildcard specifies increments. In the Minutes field, you could enter 1/10 to specify every tenth minute, starting from the first minute of the hour (for example, the 11th, 21st, and 31st minute, and so on).
- The ? (question mark) wildcard specifies one or another. In the Day-of-month field you could enter 7 and if you didn't care what day of the week the 7th was, you could enter ? in the Day-of-week field.
- The L wildcard in the Day-of-month or Day-of-week fields specifies the last day of the month or week.
- The W wildcard in the Day-of-month field specifies a weekday. In the Day-of-month field, 3W specifies the day closest to the third weekday of the month.

Note

Cron expressions that lead to rates faster than 5 minute are not supported. Support for specifying both a day-of-week and a day-of-month value is not complete. You must currently use the '?' character in one of these fields.

For more information about cron expressions, see [CRON expression](#) at the *Wikipedia website*.

Rate Expressions

Rate expressions have the following two required fields. Fields are separated by white space.

Field	Values
Value	positive number
Unit	minute(s) OR hour(s) OR day(s)

Note

If the value is equal to 1, then the unit must be singular. Similarly, for values greater than 1, the unit must be plural. For example, rate(1 hours) and rate(5 hour) are not valid, but rate(1 hour) and rate(5 hours) are valid.

Systems Manager Maintenance Windows

Systems Manager Maintenance Windows let you define a schedule for when to perform potentially disruptive actions on your instances such as patching an operating system (OS), updating drivers, or installing software. Each Maintenance Window has a schedule, a duration, a set of registered targets, and a set of registered tasks. Each task is defined to run for a subset of the registered targets. Currently, you can perform tasks like installing applications, installing or updating SSM Agent, installing patches or OS updates, or running Windows PowerShell commands using Amazon EC2 Run Command.

Information for Linux Users

See [Amazon EC2 Maintenance Windows](#) in the *Amazon EC2 User Guide for Linux Instances*.

Service Limits

Maintenance Windows currently have the following service limits:

Resource	Limit
Maintenance Windows per account	50
Tasks per Maintenance Window	20
Targets per Maintenance Window	50
Instance ids per target	50
Targets per task	10
Concurrent Executions of a single Maintenance Window	1
Concurrent Executions of Maintenance Windows	5
Execution History Retention	30 days

Before you begin

Verify that your EC2 instances and on-premises servers or virtual machines meet Systems Manager prerequisites. For more information, see [Systems Manager Prerequisites \(p. 394\)](#).

Contents

- [Creating a Maintenance Window \(p. 411\)](#)
- [Configuring Access to Maintenance Windows \(p. 412\)](#)
- [Maintenance Window Walkthroughs \(p. 413\)](#)

Creating a Maintenance Window

Creating a Maintenance Window requires that you complete the following tasks:

- Create one or more SSM command documents that define the tasks to perform on your instances during the Maintenance Window. For information about how to create an SSM command document, see [Creating SSM Documents \(p. 478\)](#).
- Create the Maintenance Window and define its schedule.
- Register targets for the Maintenance Window. Targets can either be instance IDs or EC2 tags.
- Register one or more tasks (SSM command documents) with the Maintenance Window.

After you complete these tasks, the Maintenance Window runs according to the schedule you defined and executes the tasks in your SSM documents on the targets you specified. After a task completes, Systems Manager logs the details of the execution.

Before you create a Maintenance Window, you must configure a Maintenance Window role with an Amazon Resource Name (ARN). For more information, see [Configuring Access to Maintenance Windows \(p. 412\)](#).

You can create a Maintenance Window using the **Maintenance Window** page in the Amazon EC2 console, the AWS CLI, the SSM API, or the AWS SDKs. For examples of how to create a Maintenance Window, see the [Maintenance Window Walkthroughs \(p. 413\)](#).

Configuring Access to Maintenance Windows

Use the following procedures to configure security roles and permissions for EC2 Maintenance Windows. After you configure roles and permissions, you can perform a test run with Maintenance Windows as described in [Maintenance Window Walkthroughs](#) (p. 413).

Create an IAM Role for Systems Manager

Use the following procedure to create a role so that Systems Manager can act on your behalf when creating and processing Maintenance Windows.

To create an IAM role for Maintenance Windows

1. Open the Identity and Access Management (IAM) console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Roles**, and then choose **Create New Role**.
3. In **Step 1: Set Role Name**, enter a name that identifies this role as a Maintenance Windows role.
4. In **Step 2: Select Role Type**, choose **Amazon EC2**. The system skips **Step 3: Establish Trust** because this is a managed policy.
5. In **Step 4: Attach Policy**, choose **AmazonSSMMaintenanceWindowRole**.
6. In **Step 5: Review**, make a note of the **Role Name** and **Role ARN**. You will specify the role ARN when you attach the iam:PassRole policy to your IAM account in the next procedure. You will also specify the role name and the ARN when you create a Maintenance Window.
7. Choose **Create Role**. The system returns you to the **Roles** page.
8. Locate the role you just created and double-click it.
9. Choose the **Trust Relationships** tab, and then choose **Edit Trust Relationship**.
10. Add a comma after "ec2.amazonaws.com", and then add "Service": "ssm.amazonaws.com" to the existing policy as the following code snippet illustrates:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com",
        "Service": "ssm.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

11. Choose **Update Trust Policy**.
12. Copy or make a note of the **Role ARN**. You will specify this ARN when you create your Maintenance Window.

Configure Account Permissions

Systems Manager must assume your role so that it has permission to perform the actions you specify for your Maintenance Window. Use the following procedure to attach the iam:PassRole policy to your

existing IAM user account, or create a new IAM account and attach this policy to it. If you create a new account, you must also attach the **AmazonSSMFullAccess** policy so the account can communicate with the SSM API. If you need to create a new user account, see [Creating an IAM User in Your AWS Account](#) in the *IAM User Guide*.

To attach the iam:PassRole policy to your user account

1. In the IAM console navigation pane, choose **Users** and then double-click your user account.
2. In the **Managed Policies** section, verify that either the **AmazonSSMFullAccess** policy is listed or there is a comparable policy that gives you permission to the SSM API.
3. In the **Inline Policies** section, choose **Create User Policy**. If you don't see this button, choose the down arrow beside **Inline Policies**, and then choose **click here**.
4. On the **Set Permissions** page, choose **Policy Generator**, and then choose **Select**.
5. Verify that **Effect** is set to **Allow**.
6. From **AWS Services** choose **AWS Identity and Access Management**.
7. From **Actions** choose **PassRole**.
8. In the **Amazon Resource Name (ARN)** field, paste the role ARN you created in the previous procedure.
9. Choose **Add Statement**, and then choose **Next Step**.
10. On the **Review Policy** page, choose **Apply Policy**.

Maintenance Window Walkthroughs

Use the following walkthroughs to create and run a Maintenance Window in a test environment. Before you use these walkthroughs, you must configure Maintenance Window roles and permissions. For more information, see [Configuring Access to Maintenance Windows \(p. 412\)](#).

Contents

- [Launch a New Instance \(p. 413\)](#)
- [Maintenance Window Console Walkthrough \(p. 414\)](#)
- [Maintenance Window CLI Walkthrough \(p. 415\)](#)

Launch a New Instance

Use the following procedure to create a test instance with the required AWS Identity and Access Management (IAM) role. The role enables the instance to communicate with the Systems Manager (SSM) API. You must assign the IAM role when you create the new instance. You can't assign a role to an instance that is already running.

If you want to assign the role to one of your existing instances, you must create an image of the instance, launch an instance from that image, and assign the IAM role as you launch the instance. For more information, see [Creating an Amazon EBS-Backed Windows AMI \(p. 77\)](#).

To create an instance that uses a Systems Manager-supported role

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Select a supported [region](#).
3. Choose **Launch Instance** and select a Windows Server Amazon Machine Image (AMI).
4. Choose your instance type and then choose **Next: Configure Instance Details**.
5. In **Auto-assign Public IP**, choose **Enable**.

6. Beside **IAM role** choose **Create new IAM role**. The IAM console opens in a new tab.
 - a. Choose **Create New Role**.
 - b. In **Step 1: Set Role Name**, enter a name that identifies this role as a Systems Manager role.
 - c. In **Step 2: Select Role Type**, choose **Amazon EC2 Role for Simple Systems Manager**. The system skips **Step 3: Establish Trust** because this is a managed policy.
 - d. In **Step 4: Attach Policy**, choose **AmazonEC2RoleforSSM**.
 - e. Choose **Next Step**, and then choose **Create Role**.
 - f. Close the tab with the IAM console.
7. In the Amazon EC2 console, choose the **Refresh** button beside **Create New IAM role**.
8. From **IAM role**, choose the role you just created.
9. Complete the wizard to launch the new instance. Make a note of the instance ID. You will need to specify this ID later in this walkthrough.

Maintenance Window Console Walkthrough

The following walkthrough introduces you to Maintenance Windows concepts and walks you through the process of creating and configuring a maintenance window using the Amazon EC2 console. You'll configure the Maintenance Window to run on a test instance that is configured for Systems Manager. After you finish the walkthrough, you can delete the test instance.

To create a Maintenance Window

1. Open the [Amazon EC2 console](#), expand **Systems Manager Shared Resources** in the navigation pane, and then choose **Maintenance Windows**.
2. Choose **Create a Maintenance Window**.
3. For **Name**, type a descriptive name to help you identify this Maintenance Window as a test Maintenance Window.
4. **Allow unregistered targets**: This option is not selected by default, which means any managed instance can execute a Maintenance Window task as long as the instance is targeted using its instance ID. Targets defined by tags must be registered.
5. Specify a schedule for the Maintenance Window using either the schedule builder or by specifying a schedule in cron format. For more information about cron format, see [Specifying a Cron Schedule for Your Systems Manager Shared Components \(p. 408\)](#).
6. In the **Duration** field, type the number of hours the Maintenance Window should run.
7. In the **Stop initiating tasks** field, type the number of hours before the end of the Maintenance Window that the system should stop scheduling new tasks to run.
8. Choose **Create maintenance window**. The system returns you to the Maintenance Window page. The state of the Maintenance Window you just created is **Enabled**.

After you create a Maintenance Window, you assign targets where the tasks will run.

To assign targets to a Maintenance Window

1. In the Maintenance Window list, choose the Maintenance Window you just created.
2. From the **Actions** list, choose **Register targets**.
3. In the **Owner information** field, specify your name or work alias.
4. In the **Select targets by** section, choose **Specifying instances**.
5. Choose the instance you created at the start of this walkthrough.
6. Choose **Register targets**.

The tasks you specified run on the targets you selected according to the Maintenance Window you defined when you created the window.

After you assign targets, you assign tasks to perform during the window.

To assign tasks to a Maintenance Window

1. In the Maintenance Window list, choose the Maintenance Window you just created.
2. From the **Actions** list, choose **Register task**.
3. From the **Document** list, choose the SSM command document that defines the task(s) to run. For more information about creating SSM command documents, see [Creating SSM Documents \(p. 478\)](#).
4. In the **Task Priority** field, specify a priority for this task. 1 is the highest priority. Tasks in a Maintenance Window are scheduled in priority order with tasks that have the same priority scheduled in parallel.
5. In the **Target by** section, choose **Selecting unregistered targets**, and then choose the instance you created at the start of this walkthrough.
6. In the **Parameters** section, specify parameters for the SSM command document.
7. In the **Role** field, specify the Maintenance Windows ARN. For more information about creating a Maintenance Windows ARN, see [Configuring Access to Maintenance Windows \(p. 412\)](#).
8. The **Execute on** field lets you specify either a number of targets where the Maintenance Window tasks can run concurrently or a percentage of the total number of targets. This field is relevant when you target a large number of instances using tags. For the purposes of this walkthrough, specify 1.
9. In the **Stop after** field, specify the number of allowed errors before the system stops sending the task to new instances.
10. Choose **Register task**.

Maintenance Window CLI Walkthrough

The following walkthrough introduces you to Maintenance Windows concepts and walks you through the process of creating and configuring a Maintenance Window using the AWS CLI. You'll perform the walkthrough on a test instance that is configured for Systems Manager. After you finish the walkthrough, you can delete the test instance.

Creating and Configuring a Maintenance Window Using the CLI

To create and configure a Maintenance Window Using the AWS CLI

1. [Download](#) the AWS CLI to your local machine.
2. Open the AWS CLI and execute the following command to create a Maintenance Window that runs at 4 PM on every Tuesday for 4 hours, with a 1 hour cutoff, and that allows unassociated targets. For more information about creating cron expressions for the `schedule` parameter, see [Specifying a Cron Schedule for Your Systems Manager Shared Components \(p. 408\)](#).

```
aws ssm create-maintenance-window --name "My-First-Maintenance-Window"
--schedule "cron(0 16 ? * TUE *)" --duration 4 --cutoff 1 --allow-
unassociated-targets --region an SSM region
```

The system returns information like the following.

```
{
  "WindowId": "mw-ab12cd34ef56gh78"
```

```
}
```

3. Execute the following command to list all Maintenance Windows in your AWS account.

```
aws ssm describe-maintenance-windows --region an SSM region
```

The system returns information like the following.

```
{
  "WindowIdentities": [
    {
      "Duration": 4,
      "Cutoff": 1,
      "WindowId": "mw-ab12cd34ef56gh78",
      "Enabled": true,
      "Name": "My-First-Maintenance-Window"
    }
  ]
}
```

4. Execute the following command to register the instance you created earlier as a target for this Maintenance Windows. The system returns a Maintenance Window target ID. You will use this ID in a later step to register a task for this Maintenance Window.

```
aws ssm register-target-with-maintenance-window --region an SSM region --
window-id "mw-ab12cd34ef56gh78" --target "Key=InstanceIds,Values=ID" --
owner-information "Single instance" --Resource-type "INSTANCE"
```

The system returns information like the following.

```
{
  "WindowTargetId": "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
}
```

You could register multiple instances using the following command.

```
aws ssm register-target-with-maintenance-window --region an SSM region
--window-id "mw-ab12cd34ef56gh78" --targets "Key=InstanceIds,Values=ID
1, ID 2" --owner-information "Two instances in a list" --Resource-type
"INSTANCE"
```

You could also register instances using EC2 tags.

```
aws ssm register-target-with-maintenance-window --window-id "mw-
ab12cd34ef56gh78" --targets "Key=tag:Environment,Values=Prod"
"Key=Role,Values=Web" --owner-information "Production Web Servers" --
resource-type "INSTANCE"
```

5. Use the following command to display the targets for a Maintenance Window.

```
aws ssm describe-maintenance-window-targets --region an SSM region --
window-id "mw-ab12cd34ef56gh78"
```

The system returns information like the following.

```
{
  "Targets": [
    {
      "ResourceType": "INSTANCE",
      "OwnerInformation": "Single instance",
      "WindowId": "mw-ab12cd34ef56gh78",
      "Targets": [
        {
          "Values": [
            "i-11aa22bb33cc44dd5"
          ],
          "Key": "InstanceIds"
        }
      ],
      "WindowTargetId": "alb2c3d4-alb2-alb2-alb2-alb2c3d4"
    },
    {
      "ResourceType": "INSTANCE",
      "OwnerInformation": "Two instances in a list",
      "WindowId": "mw-ab12cd34ef56gh78",
      "Targets": [
        {
          "Values": [
            "i-1a2b3c4d5e6f7g8h9",
            "i-aa11bb22cc33dd44e "
          ],
          "Key": "InstanceIds"
        }
      ],
      "WindowTargetId": "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
    },
    {
      "ResourceType": "INSTANCE",
      "OwnerInformation": "Production Web Servers",
      "WindowId": "mw-ab12cd34ef56gh78",
      "Targets": [
        {
          "Values": [
            "Prod"
          ],
          "Key": "tag:Environment"
        },
        {
          "Values": [
            "Web"
          ],
          "Key": "tag:Role"
        }
      ],
      "WindowTargetId": "1111aaa-2222-3333-4444-1111aaa "
    }
  ]
}
```

6. Execute the following command to register a task on the instance you created earlier. This task uses Systems Manager Run Command to execute `ip config` using the AWS-RunPowerShellScript document. This command uses the following parameters:

- **targets:** Specify either `Key=WindowTargetIds,Values=Window Target ID` to specify a target registered with the Maintenance Window or `Key=InstanceIds,Values=Instance ID` to specify individual instances registered with the Maintenance Window.
- **task-arn:** Specify the name of an SSM Run Command document. For example: `AWS-RunShellScript`, `AWS-RunPowerShellScript`, or `arn:aws:ssm:us-east-1:123456789:document/Restart_Apache` (for a shared document).
- **window-id:** Specify the ID of the Maintenance Window.
- **task-type:** Specify `RUN_COMMAND`. Currently only Run Command tasks are supported.
- **task-parameters:** Specify required and optional parameters for the Run Command document.
- **max-concurrency:** (Optional) Specify the maximum number of instances that are allowed to execute the command at the same time. You can specify a number such as 10 or a percentage such as 10%.
- **max-errors:** (Optional) Specify the maximum number of errors allowed without the command failing. When the command fails one more time beyond the value of `MaxErrors`, the systems stops sending the command to additional targets. You can specify a number such as 10 or a percentage such as 10%.
- **priority:** Specify the priority of the task in the Maintenance Window. The lower the number the higher the priority (for example, 1 is highest priority). Tasks in a Maintenance Window are scheduled in priority order. Tasks that have the same priority are scheduled in parallel.

```
aws ssm register-task-with-maintenance-window --targets
"Key=InstanceIds,Values=Instance ID" --task-arn "AWS-RunPowerShellScript"
--service-role-arn "arn:aws:iam::1122334455:role/MW-Role" --window-
id "mw-ab12cd34ef56gh78" --task-type "RUN_COMMAND" --task-parameters
'{"commands":{"Values":["driverquery.exe"]}}' --max-concurrency 1
--max-errors 1 --priority 3
```

The system returns information like the following.

```
{
  "WindowTaskId": "44444444-5555-6666-7777-88888888"
}
```

You can also register a task using a Maintenance Window target ID. The Maintenance Window target ID was returned from an earlier command.

```
aws ssm register-task-with-maintenance-window --targets
"Key=WindowTargetIds,Values=Window Target ID" --task-arn "AWS-
RunPowerShellScript" --service-role-arn "arn:aws:iam::1122334455:role/
MW-Role" --window-id "mw-ab12cd34ef56gh78" --task-type "RUN_COMMAND" --
task-parameters '{"commands":{"Values":["ipconfig.exe"]}}' --max-
concurrency 1 --max-errors 1 --priority 1
```

7. Execute the following command to list all registered tasks for a Maintenance Window.

```
aws ssm describe-maintenance-window-tasks --window-id "mw-
ab12cd34ef56gh78"
```

The system returns information like the following.

```
{
```

```
"Tasks":[
  {
    "ServiceRoleArn":"arn:aws:iam::11111111:role/MW-Role",
    "MaxErrors":"1",
    "TaskArn":"AWS-RunPowerShellScript",
    "MaxConcurrency":"1",
    "WindowTaskId":"3333-3333-3333-333333",
    "TaskParameters":{
      "commands":{
        "Values":[
          "driverquery.exe"
        ]
      }
    },
    "Priority":3,
    "Type":"RUN_COMMAND",
    "Targets":[
      {
        "Values":[
          "i-1a2b3c4d5e6f7g8h9"
        ],
        "Key":"InstanceIds"
      }
    ]
  },
  {
    "ServiceRoleArn":"arn:aws:iam::2222222222:role/MW-Role",
    "MaxErrors":"1",
    "TaskArn":"AWS-RunPowerShellScript",
    "MaxConcurrency":"1",
    "WindowTaskId":"44444-44-44-444444",
    "TaskParameters":{
      "commands":{
        "Values":[
          "ipconfig.exe"
        ]
      }
    },
    "Priority":1,
    "Type":"RUN_COMMAND",
    "Targets":[
      {
        "Values":[
          "555555-55555-555-5555555"
        ],
        "Key":"WindowTargetIds"
      }
    ]
  }
]
```

8. Execute the following command to view a list of task executions for a specific Maintenance Window.

```
aws ssm describe-maintenance-window-executions --window-id "mw-  
ab12cd34ef56gh78"
```

The system returns information like the following.

```
{
  "WindowExecutions": [
    {
      "Status": "SUCCESS",
      "WindowExecutionId": "1111-1111-1111-1111",
      "StartTime": 1478230495.469
    },
    {
      "Status": "SUCCESS",
      "WindowExecutionId": "2222-2-2-22222222-22",
      "StartTime": 1478231395.677
    },
    # ... omitting a number of entries in the interest of space...
    {
      "Status": "SUCCESS",
      "WindowExecutionId": "33333-333-333-33333333",
      "StartTime": 1478272795.021
    },
    {
      "Status": "SUCCESS",
      "WindowExecutionId": "4444-44-44-44444444",
      "StartTime": 1478273694.932
    }
  ],
  "NextToken": "111111 ..."
}
```

9. Execute the following command to get information about a Maintenance Window task execution.

```
aws ssm get-maintenance-window-execution --window-execution-id
"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
```

The system returns information like the following.

```
{
  "Status": "SUCCESS",
  "TaskIds": [
    "333-33-3333-3333333"
  ],
  "StartTime": 1478230495.472,
  "EndTime": 1478230516.505,
  "WindowExecutionId": "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
}
```

10. Execute the following command to list the tasks executed as part of a Maintenance Window execution.

```
aws ssm describe-maintenance-window-execution-tasks --window-execution-id
"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
```

The system returns information like the following.

```
{
```

```
"WindowExecutionTaskIdentities": [
  {
    "Status": "SUCCESS",
    "EndTime": 1478230516.425,
    "StartTime": 1478230495.782,
    "TaskId": "33333-333-333-3333333"
  }
]
```

11. Execute the following command to get the details of a task execution.

```
aws ssm get-maintenance-window-execution-task --window-execution-id
"555555-555-55-555555" --task-id "4444-4444-4444-444444"
```

The system returns information like the following.

```
{
  "Status": "SUCCESS",
  "MaxErrors": "1",
  "TaskArn": "AWS-RunPowerShellScript",
  "MaxConcurrency": "1",
  "ServiceRole": "arn:aws:iam::333333333:role/MW-Role",
  "WindowExecutionId": "555555-555-55-555555",
  "Priority": 0,
  "StartTime": 1478230495.782,
  "EndTime": 1478230516.425,
  "Type": "RUN_COMMAND",
  "TaskParameters": [
  ],
  "TaskExecutionId": "4444-4444-4444-444444"
}
```

12. Execute the following command to get the specific task invocations performed for a task execution.

```
aws ssm describe-maintenance-window-execution-task-invocations --window-
execution-id "555555-555-55-555555" --task-id "4444-4444-4444-444444"
```

The system returns information like the following.

```
{
  "WindowExecutionTaskInvocationIdentities": [
    {
      "Status": "SUCCESS",
      "Parameters": "{\n  documentName\n  : \n  AWS-RunPowerShellScript\n  \n  , \n  instanceIds\n  \" : [ \n  i-1a2b3c4d5e6f7g8h9\n  \n  , \n  i-0a00def7faa94f1dc\n  \n  ], \n  parameters\n  \" : { \n  commands\n  \" : [ \n  ipconfig.exe\n  \n  ] }, \n  maxConcurrency\n  \" : \n  1\n  \n  , \n  maxErrors\n  \" : \n  1\n  \n  }",
      "ExecutionId": "555555-555-55-555555",
      "InvocationId": "3333-33333-3333-33333",
      "StartTime": 1478230495.842,
      "EndTime": 1478230516.291
    }
  ]
}
```

```
}
```

Additional Maintenance Window Configuration Commands

This section includes commands to help you update or get information about your Maintenance Windows, tasks, executions, and invocations.

List All Maintenance Windows in Your AWS Account

```
aws ssm describe-maintenance-windows --region an SSM region
```

The system returns information like the following.

```
{
  "WindowIdentities": [
    {
      "Duration": 2,
      "Cutoff": 0,
      "WindowId": "mw-ab12cd34ef56gh78",
      "Enabled": true,
      "Name": "IAD-Every-15-Minutes"
    },
    {
      "Duration": 4,
      "Cutoff": 1,
      "WindowId": "mw-1a2b3c4d5e6f7g8h9",
      "Enabled": true,
      "Name": "My-First-Maintenance-Window"
    },
    {
      "Duration": 8,
      "Cutoff": 2,
      "WindowId": "mw-123abc456def789",
      "Enabled": false,
      "Name": "Every-Day"
    }
  ]
}
```

List all enabled Maintenance Windows

```
aws ssm describe-maintenance-windows --region an SSM region --filters
  "Key=Enabled,Values=true"
```

The system returns information like the following.

```
{
  "WindowIdentities": [
    {
      "Duration": 2,
      "Cutoff": 0,
      "WindowId": "mw-ab12cd34ef56gh78",
      "Enabled": true,
      "Name": "IAD-Every-15-Minutes"
    }
  ]
}
```

```
    },  
    {  
      "Duration":4,  
      "Cutoff":1,  
      "WindowId":"mw-1a2b3c4d5e6f7g8h9",  
      "Enabled":true,  
      "Name":"My-First-Maintenance-Window"  
    }  
  ]  
}
```

List all Disabled Maintenance Windows

```
aws ssm describe-maintenance-windows --region an SSM region --filters  
  "Key=Enabled,Values=false"
```

The system returns information like the following.

```
{  
  "WindowIdentities":[  
    {  
      "Duration":8,  
      "Cutoff":2,  
      "WindowId":"mw-1a2b3c4d5e6f7g8h9",  
      "Enabled":false,  
      "Name":"Every-Day"  
    }  
  ]  
}
```

Filter by Name

In this example, the command returns all Maintenance Windows with a name starting with 'My'.

```
aws ssm describe-maintenance-windows --region an SSM region --filters  
  "Key=Name,Values=My"
```

The system returns information like the following.

```
{  
  "WindowIdentities":[  
    {  
      "Duration":4,  
      "Cutoff":1,  
      "WindowId":"mw-1a2b3c4d5e6f7g8h9",  
      "Enabled":true,  
      "Name":"My-First-Maintenance-Window"  
    }  
  ]  
}
```

Modify a Maintenance Window

You can modify the following parameters: Name, Schedule, Duration, Cutoff, AllowUnassociatedTargets, and Enabled. The following example modifies the `name` value.

```
aws ssm update-maintenance-window --region an SSM region --window-id  
"mw-1a2b3c4d5e6f7g8h9" --name "My-Renamed-MW"
```

The system returns information like the following.

```
{  
  "Cutoff": 1,  
  "Name": "My-Renamed-MW",  
  "Schedule": "cron(0 16 ? * TUE *)",  
  "Enabled": true,  
  "AllowUnassociatedTargets": true,  
  "WindowId": "mw-1a2b3c4d5e6f7g8h9",  
  "Duration": 4  
}
```

Modifying the unassociated targets parameter

```
aws ssm update-maintenance-window --region an SSM region --window-id  
"mw-1a2b3c4d5e6f7g8h9" --no-allow-unassociated-targets
```

The system returns information like the following.

```
{  
  "Cutoff": 2,  
  "Name": "Every-Tuesday-4pm",  
  "Schedule": "cron(0 16 ? * TUE *)",  
  "Enabled": true,  
  "AllowUnassociatedTargets": false,  
  "WindowId": "mw-1a2b3c4d5e6f7g8h9",  
  "Duration": 8  
}
```

```
aws ssm update-maintenance-window --region an SSM region --window-id  
"mw-1a2b3c4d5e6f7g8h9" --allow-unassociated-targets --no-enabled
```

The system returns information like the following.

```
{  
  "Cutoff": 2,  
  "Name": "Every-Tuesday-4pm",  
  "Schedule": "cron(0 16 ? * TUE *)",  
  "Enabled": false,  
  "AllowUnassociatedTargets": true,  
  "WindowId": "mw-1a2b3c4d5e6f7g8h9",  
  "Duration": 8  
}
```

Display the Targets for a Maintenance Window Matching a Specific Owner Information Value

```
aws ssm describe-maintenance-window-targets --region an SSM region --window-  
id "mw-ab12cd34ef56gh78" --filters "Key=OwnerInformation,Values=Single  
instance"
```

The system returns information like the following.

```
{
  "Targets": [
    {
      "TargetType": "INSTANCE",
      "TagFilters": [
      ],
      "TargetIds": [
        "i-1a2b3c4d5e6f7g8h9"
      ],
      "WindowTargetId": "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2",
      "OwnerInformation": "Single instance"
    }
  ]
}
```

Show All Registered Tasks that Invoke the AWS-RunPowerShellScript Run Command

```
aws ssm describe-maintenance-window-tasks --window-id "mw-ab12cd34ef56gh78"
--filters "Key=TaskArn,Values=AWS-RunPowerShellScript"
```

The system returns information like the following.

```
{
  "Tasks": [
    {
      "ServiceRoleArn": "arn:aws:iam::444444444444:role/MW-Role",
      "MaxErrors": "1",
      "TaskArn": "AWS-RunPowerShellScript",
      "MaxConcurrency": "1",
      "WindowTaskId": "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d5e6c",
      "TaskParameters": {
        "commands": {
          "Values": [
            "driverquery.exe"
          ]
        }
      },
      "Priority": 3,
      "Type": "RUN_COMMAND",
      "Targets": [
        {
          "TaskTargetId": "i-1a2b3c4d5e6f7g8h9",
          "TaskTargetType": "INSTANCE"
        }
      ]
    },
    {
      "ServiceRoleArn": "arn:aws:iam::333333333333:role/MW-Role",
      "MaxErrors": "1",
      "TaskArn": "AWS-RunPowerShellScript",
      "MaxConcurrency": "1",
      "WindowTaskId": "333333-333333-333-333333",
      "TaskParameters": {
        "commands": {
```



```
        "Values": [
            "ipconfig.exe"
        ]
    },
    "Priority": 1,
    "Type": "RUN_COMMAND",
    "Targets": [
        {
            "TaskTargetId": "44444-444-4444-4444444",
            "TaskTargetType": "WINDOW_TARGET"
        }
    ]
}
]
```

Show All Registered Tasks that Have a Priority of 3

```
aws ssm describe-maintenance-window-tasks --window-id "mw-ab12cd34ef56gh78"
--filters "Key=Priority,Values=3"
```

The system returns information like the following.

```
{
  "Tasks": [
    {
      "ServiceRoleArn": "arn:aws:iam::222222222:role/MW-Role",
      "MaxErrors": "1",
      "TaskArn": "AWS-RunPowerShellScript",
      "MaxConcurrency": "1",
      "WindowTaskId": "333333-333-33333-33333",
      "TaskParameters": {
        "commands": {
          "Values": [
            "driverquery.exe"
          ]
        }
      },
      "Priority": 3,
      "Type": "RUN_COMMAND",
      "Targets": [
        {
          "TaskTargetId": "i-1a2b3c4d5e6f7g8h9",
          "TaskTargetType": "INSTANCE"
        }
      ]
    }
  ]
}
```

Show All Registered Tasks that Have a Priority of 1 and Use Run Command

```
aws ssm describe-maintenance-window-tasks --window-id "mw-ab12cd34ef56gh78"
--filters "Key=Priority,Values=1" "Key=TaskType,Values=RUN_COMMAND"
```

The system returns information like the following.

```
{
  "Tasks": [
    {
      "ServiceRoleArn": "arn:aws:iam::3333333333:role/MW-Role",
      "MaxErrors": "1",
      "TaskArn": "AWS-RunPowerShellScript",
      "MaxConcurrency": "1",
      "WindowTaskId": "66666-555-66-555-6666",
      "TaskParameters": {
        "commands": {
          "Values": [
            "ipconfig.exe"
          ]
        }
      },
      "Priority": 1,
      "Type": "RUN_COMMAND",
      "Targets": [
        {
          "TaskTargetId": "777-77-777-7777777",
          "TaskTargetType": "WINDOW_TARGET"
        }
      ]
    }
  ]
}
```

List All Tasks Executed Before a Date

```
aws ssm describe-maintenance-window-executions --window-id "mw-  
ab12cd34ef56gh78" --filters "Key=ExecutedBefore,Values=2016-11-04T05:00:00Z"
```

The system returns information like the following.

```
{
  "WindowExecutions": [
    {
      "Status": "SUCCESS",
      "EndTime": 1478229594.666,
      "WindowExecutionId": "",
      "StartTime": 1478229594.666
    },
    {
      "Status": "SUCCESS",
      "WindowExecutionId": "06dc5f8a-9ef0-4ae9-a466-ada2d4ce2d22",
      "StartTime": 1478230495.469
    },
    {
      "Status": "SUCCESS",
      "WindowExecutionId": "57ad6419-023e-44b0-a831-6687334390b2",
      "StartTime": 1478231395.677
    },
    {
      "Status": "SUCCESS",

```

Amazon Elastic Compute Cloud
User Guide for Windows Instances
Maintenance Windows

```
    "WindowExecutionId": "ed1372b7-866b-4d64-bc2a-bbfd5195f4ae",  
    "StartTime": 1478232295.529  
  },  
  {  
    "Status": "SUCCESS",  
    "WindowExecutionId": "154eb2fa-6390-4cb7-8c9e-55686b88c7b3",  
    "StartTime": 1478233195.687  
  },  
  {  
    "Status": "SUCCESS",  
    "WindowExecutionId": "1c4de752-eff6-4778-b477-1681c6c03cf1",  
    "StartTime": 1478234095.553  
  },  
  {  
    "Status": "SUCCESS",  
    "WindowExecutionId": "56062f75-e4d8-483f-b5c2-906d613409a4",  
    "StartTime": 1478234995.12  
  }  
]  
}
```

List All Tasks Executed After a Date

```
aws ssm describe-maintenance-window-executions --window-id "mw-  
ab12cd34ef56gh78" --filters "Key=ExecutedAfter,Values=2016-11-04T17:00:00Z"
```

The system returns information like the following.

```
{  
  "WindowExecutions": [  
    {  
      "Status": "SUCCESS",  
      "WindowExecutionId": "33333-4444-444-5555555",  
      "StartTime": 1478279095.042  
    },  
    {  
      "Status": "SUCCESS",  
      "WindowExecutionId": "55555-6666-6666-777777",  
      "StartTime": 1478279994.958  
    },  
    {  
      "Status": "SUCCESS",  
      "WindowExecutionId": "8888-888-888-888888",  
      "StartTime": 1478280895.149  
    }  
  ]  
}
```

Remove a Target from a Maintenance Window

```
aws ssm deregister-target-from-maintenance-window --region an  
SSM region --window-id "mw-ab12cd34ef56gh78" --window-target-id  
"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
```

The system returns information like the following.

```
{
  "WindowId": "mw-ab12cd34ef56gh78",
  "WindowTargetId": "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
}
```

Remove a Task from a Maintenance Window

```
aws ssm deregister-task-from-maintenance-window --window-id "mw-
ab12cd34ef56gh78" --window-task-id "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d5e6c"
```

The system returns information like the following.

```
{
  "WindowTaskId": "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d5e6c",
  "WindowId": "mw-ab12cd34ef56gh78"
}
```

Delete a Maintenance Window

```
aws ssm delete-maintenance-window --window-id "mw-1a2b3c4d5e6f7g8h9"
```

The system returns information like the following:

```
{
  "WindowId": "mw-1a2b3c4d5e6f7g8h9"
}
```

Systems Manager Parameter Store

Storing and referencing configuration data such as passwords, license keys, key pairs, certificates, and lists of users can be a time-consuming and error-prone process, especially at scale. Storing and using password in a secure manner is equally challenging at scale. Parameter Store efficiently and securely centralizes the management of configuration data that you commonly reference in scripts, commands, or other automation and configuration workflows. Parameter Store lets you reference parameters (called Systems Manager parameters) across Systems Manager features, including Run Command, State Manager, and Automation. You can also reference Systems Manager parameters across AWS services, including AWS Lambda and AWS CloudFormation.

For parameters such as passwords or key pairs that should be encrypted, Parameter Store lets you encrypt data by using an AWS Key Management Service (AWS KMS) key. You can then delegate access to users who should be allowed to decrypt and view the sensitive data. You can also monitor and audit parameter usage in Amazon EC2 or AWS CloudTrail.

Information for Linux Users

See [Systems Manager Parameter Store](#) in the *Amazon EC2 User Guide for Linux Instances*.

Contents

- [Service Limits \(p. 430\)](#)
- [About Systems Manager parameters \(p. 430\)](#)

- [Using Systems Manager parameters \(p. 431\)](#)
- [About Secure String Parameters \(p. 432\)](#)
- [Configuring Access to Systems Manager parameters \(p. 433\)](#)
- [Systems Manager Parameter Store Walkthroughs \(p. 435\)](#)

Service Limits

Parameter Store currently has the following service limits:

Resource	Limit
Maximum number of parameters per account	100
Max size for parameter value	1024 characters
Max history for a parameter	100 past values

Also note the following limitations:

- You can't use parameters across regions. You must reference Systems Manager parameters in the region where they were created.
- The Amazon EC2 console currently doesn't support creating an encrypted parameter (Secure String) with a custom KMS key. The console creates a Secure String parameter that uses the default KMS key assigned to your AWS account. You can create a Secure String with a custom KMS key by using the AWS CLI or the AWS SDK.

Before you begin

Verify that your EC2 instances and on-premises servers or virtual machines meet Systems Manager prerequisites. For more information, see [Systems Manager Prerequisites \(p. 394\)](#).

About Systems Manager parameters

An Systems Manager parameter is a key-value pair that you create by specifying the following information.

- **Name:** (Required) Specify a name to identify your parameter. Be aware of the following requirements and restrictions for Systems Manager parameter names:
 - A parameter name must be unique within your AWS account.
 - Parameter names are case-sensitive.
 - A parameter name *can't* be prefixed with "aws" or "ssm" (case-insensitive). For example, awsTestParameter or SSM-testparameter will fail with an exception.
 - Parameter names can only include the following symbols and letters:
a-zA-Z0-9_.-
- **Data Type:** (Required) Specify a data type to define how the system uses a parameter. Parameter Store currently supports the following data types: String, String List, and Secure String.
- **Description** (Optional): Type a description to help you identify your parameters and their intended use.
- **Value:** (Required) Your parameter value.
- **Key ID** (for Secure String): Either the default AWS KMS key automatically assigned to your AWS account or a custom key.

Note

You can use "." Or "_" to group similar parameters. For example, you could group parameters as follows: prod.db.string and prod.domain.password.

Using Systems Manager parameters

After you create a parameter, you can specify it in your SSM documents, commands, or scripts using the following syntax:

```
{{ ssm:parameter_name }}
```

Note

The *name* of an Systems Manager parameter can't be prefixed with "ssm" or "aws", but when you specify the parameter in an SSM document or a command, the name must be prefixed with "ssm:". Valid: {{ssm:addUsers}}. Invalid: {{ssm:ssmAddUsers}}.

The following is an example of an AWS CLI Run Command command using an SSM Parameter.

```
aws ssm send-command --instance-ids i-1a2b3c4d5e6f7g8 --document-name AWS-RunPowerShellScript --parameter '{"commands":["echo {{ssm:addUsers}}"]}'
```

Note

The runtimeConfig section of SSM documents use similar syntax for *local parameters*. You can distinguish local parameters from Systems Manager parameters by the absence of the "ssm:" prefix.

```
"runtimeConfig":{
  "aws:runShellScript":{
    "properties":[
      {
        "id":"0.aws:runShellScript",
        "runCommand":"{{ commands }}",
        "workingDirectory":"{{ workingDirectory }}",
        "timeoutSeconds":"{{ executionTimeout }}"
      }
    ]
  }
}
```

You can reference Systems Manager parameters in the *Parameters* section of an SSM document, as show in the following example.

```
{
  "schemaVersion":"2.0",
  "$schema":"http://amazonaws.com/schemas/ec2/v3-0/runcommand#",
  "description":"Sample version 2.0 document v2",
  "parameters":{
    "commands" : {
      "type": "StringList",
      "default": ["{{ssm:commands}}"]
    }
  },
  "mainSteps":[
    {
      "action":"aws:runShellScript",
      "name":"runShellScript",
      "inputs":{
        "commands": "{{commands}}"
      }
    }
  ]
}
```

```
]
}
```

Predefined SSM documents (all documents that begin with "AWS-") currently don't support Secure Strings or references to Secure String type parameters. This means that to use Secure String parameters with Run Command, you have to retrieve the parameter value before passing it to Run Command, as shown in the following example:

```
$secure = (Get-SSMParameters -Names SecureParam -WithDecryption
$True).Parameters[0].Value | ConvertTo-SecureString -AsPlainText -Force
```

```
$scred = New-Object System.Management.Automation.PSCredential -
argumentlist username,$secure
```

About Secure String Parameters

A secure string is any sensitive data that needs to be stored and referenced in a secure manner. If you have data that you don't want users to alter or reference in clear text, such as domain join passwords or license keys, then specify those values using the Secure String data type. You should use secure strings when:

- You want to use data/parameters across AWS services without exposing the values as clear text in commands, functions, agent logs, or AWS CloudTrail logs.
- You want to control who has access to sensitive data.
- You want to be able to audit when sensitive data is accessed (AWS CloudTrail).
- You want AWS-level encryption for your sensitive data and you want to bring your own encryption keys to manage access.

If you choose the Secure String data type when you create your parameter, then Systems Manager encrypts the parameter value when it is passed into a command and decrypts it when processing it on the managed instance. Encryption ensures that you can pass sensitive data such as passwords or license keys into commands or scripts without exposing the values in the command, agent logs, or other services such as AWS CloudTrail.

Encryption is handled by AWS KMS. Each AWS account is assigned a default AWS KMS key. You can view your key by executing the following command from the AWS CLI:

```
aws kms describe-key --key-id alias/aws/ssm
```

If you create a Secure String parameter using your own KMS key, then you don't have to provide a value for the Key ID parameter. The following CLI example shows the command to create a new Secure String parameter in Parameter Store without the `--key-id` parameter:

```
aws ssm put-parameter --name secure_string1_default_key --value
"a_secure_string_value" --type SecureString
```

If you want to use a custom KMS key instead of the default key assigned to your account, then you must specify the ARN using the `--key-id` parameter, as shown in the following AWS CLI example:

```
aws ssm put-parameter --name secure_string1_custom_key --value
"a_secure_string_value" --type SecureString --key-id arn:aws:kms:us-
east-1:123456789012:key/1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d5e
```

You can create a custom AWS KMS key from the AWS CLI by using the following command:

```
aws kms create-key
```

You can type an encrypted value for your parameter. In this case, because the string is already encrypted, you don't have to choose the Secure String data type. If you do choose Secure String, your parameter will be doubly encrypted.

The Secure String data type is recognized by Systems Manager Run Command, which means you can define a parameter as a Secure String within the Parameters section of your Run Command SSM document. When you author your document, you set the type of the parameter to SecureString, as shown below. When returning the parameter values that were sent to a given command, the values for secure parameters will be an encrypted string. However, for existing AWS documents, not all parameters can be passed securely.

```
{
  "schemaVersion": "1.2",
  "description": "Run a PowerShell script or specify the paths to scripts to run.",
  "parameters": {
    "DNS": {
      "type": "SecureString",
      "description": "(Required) Specify the license key to be set on the instance."
    },
    "runtimeConfig": {
      "aws:runPowerShellScript": {
        "properties": [
          {
            "id": "0.aws:runPowerShellScript",
            "runCommand": "set-dns {{ license_key }}"
          }
        ]
      }
    }
  }
}
```

By default, all Secure String values are cipher text when displayed using the EC2 console and the AWS CLI. To decrypt a Secure String value, a user must have KMS decryption permissions, as described in the next section. For more information about AWS KMS, see [AWS Key Management Service Developer Guide](#).

Configuring Access to Systems Manager parameters

We recommend that you restrict user access to Systems Manager parameters by creating restrictive AWS Identity and Access Management (IAM) user policies. For example, the following policy gives the user read-only permission (GetParameters and DescribeParameters) to all production parameters (parameters that begin with prod.*).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
```



```
        "Action": [
            "ssm:DescribeParameters"
        ],
        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "ssm:GetParameters",
        ],
        "Resource": "arn:aws:ssm:us-east-1:348788061755:parameter/prod.*"
    }
]
```

If you want to provide a user with full access to all Systems Manager Parameter API operations, use a policy like the following example. This policy gives the user full access to all production parameters that begin with `prod.*`.

```
{
    "Version": "2012-10-17",
    "Effect": "Allow",
    "Action": [
        "ssm:DescribeParameter",
        "ssm:PutParameter",
        "ssm:GetParameter",
        "ssm>DeleteParameter"
    ],
    "Resource": [
        "arn:aws:ssm:region:account id:parameter/dbserver.prod.*"
    ]
}
```

You can also delegate access so that instances can only run specific parameters. For secure strings, you have to provide KMS decrypt permissions so that secure string parameters can be decrypted by the instance. The following example enable instances to get a parameter value only for parameters that begin with `prod.`. If the parameter is a secure string, then the instance decrypts the string using KMS.

```
{
    "Version": "2012-10-17",
    "Effect": "Allow",
    "Action": [
        "ssm:GetParameter",
    ],
    "Resource": [
        "arn:aws:ssm:region:account id:parameter/prod.*"
    ]
},
{
    "Version": "2012-10-17",
    "Effect": "Allow",
    "Action": [
        "kms:Decrypt",
    ],
    "Resource": [
        "arn:aws:kms: region:account id:key/CMK"
    ]
}
```

```
}  
  }  
]
```

Note

Instance policies, like in the previous example, are assigned to the instance role in IAM. For more information about configuring access to Systems Manager features, including how to assign policies to users and instances, see [Configuring Access to Systems Manager](#) (p. 400).

Systems Manager Parameter Store Walkthroughs

Use the following walkthroughs to create, store, and execute parameters with Parameter Store in a test environment.

Contents

- [Grant Your User Account Access to SSM](#) (p. 435)
- [Launch a New Instance](#) (p. 435)
- [Systems Manager Parameter Store Console Walkthrough](#) (p. 436)
- [Systems Manager Parameter Store CLI Walkthrough](#) (p. 436)

Grant Your User Account Access to SSM

Your user account must be configured to communicate with the SSM API. Use the following procedure to attach a managed IAM policy to your user account that grants you full access to SSM API actions.

To create the IAM policy for your user account

1. Open the Identity and Access Management (IAM) console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Policies**. (If this is your first time using IAM, choose **Get Started**, and then choose **Create Policy**.)
3. In the **Filter** field, type `AmazonSSMFullAccess` and press Enter.
4. Select the check box next to **AmazonSSMFullAccess** and then choose **Policy Actions, Attach**.
5. On the **Attach Policy** page, choose your user account and then choose **Attach Policy**.

Launch a New Instance

Use the following procedure to create a test instance with the required AWS Identity and Access Management (IAM) role. The role enables the instance to communicate with the Systems Manager (SSM) API. You must assign the IAM role when you create the new instance. You can't assign a role to an instance that is already running.

If you want to assign the role to one of your existing instances, you must create an image of the instance, launch an instance from that image, and assign the IAM role as you launch the instance. For more information, see [Creating an Amazon EBS-Backed Windows AMI](#) (p. 77).

To create an instance that uses a Systems Manager-supported role

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Select a supported [region](#).
3. Choose **Launch Instance** and select a Windows Server Amazon Machine Image (AMI).
4. Choose your instance type and then choose **Next: Configure Instance Details**.
5. In **Auto-assign Public IP**, choose **Enable**.

6. Beside **IAM role** choose **Create new IAM role**. The IAM console opens in a new tab.
 - a. Choose **Create New Role**.
 - b. In **Step 1: Set Role Name**, enter a name that identifies this role as a Systems Manager role.
 - c. In **Step 2: Select Role Type**, choose **Amazon EC2 Role for Simple Systems Manager**. The system skips **Step 3: Establish Trust** because this is a managed policy.
 - d. In **Step 4: Attach Policy**, choose **AmazonEC2RoleforSSM**.
 - e. Choose **Next Step**, and then choose **Create Role**.
 - f. Close the tab with the IAM console.
7. In the Amazon EC2 console, choose the **Refresh** button beside **Create New IAM role**.
8. From **IAM role**, choose the role you just created.
9. Complete the wizard to launch the new instance. Make a note of the instance ID. You will need to specify this ID later in this walkthrough.

Systems Manager Parameter Store Console Walkthrough

The following procedure walks you through the process of creating a parameter in Parameter Store and then executing a Run Command command that uses this parameter.

To create a parameter using Parameter Store

1. Open the [Amazon EC2 console](#), expand **Systems Manager Shared Resources** in the navigation pane, and then choose **Parameter Store**.
2. Choose **Create Parameter**.
3. For **Name**, type helloWorld.
4. In the **Description** field, type a description that identifies this parameter as a test parameter.
5. For **Type**, choose **String**.
6. In the **Value** field, enter a word.
7. Choose **Create Parameter** and then choose **OK** after the system creates the parameter.
8. In the EC2 console navigation pane, expand **Commands** and then choose **Run Command**.
9. Choose **Run a command**.
10. In the **Command Document** list, choose AWS-RunPowershellScript.
11. Under **Target instances**, choose the instance you created earlier.
12. In the **Commands** field, type `echo {{ssm:helloWorld}}` and then choose **Run**.
13. In the command history list, choose the command you just ran, choose the **Output** tab, and then choose **View Output**. The output is the name of the parameter you created earlier, for example, `{{ssm:helloWorld}}`.

Systems Manager Parameter Store CLI Walkthrough

The following procedure walks you through the process of creating and storing a parameter using the AWS CLI.

To create a parameter using Parameter Store

1. [Download](#) the AWS CLI to your local machine.
2. Execute the following command to create a parameter that uses the String data type.

```
aws ssm put-parameter --name commands --type String --value "helloWorld"
```

3. Execute the following command to view the parameter metadata.

```
aws ssm describe-parameters --filters Key=Name,Values=helloWorld
```

4. Execute the following command to change the parameter value.

```
aws ssm put-parameter --name helloWorld --type String --value "good day  
sunshine"
```

5. Execute the following command to view the latest parameter value.

```
aws ssm get-parameters --name helloWorld
```

6. Execute the following command to view the parameter value history.

```
aws ssm get-parameter-history --name helloWorld
```

7. Execute the following command to use this parameter in a Run Command command.

```
aws ssm send-command --name AWS-RunPowerShellScript --parameters  
commands=["echo {{ssm:helloWorld}}"] --targets Key=instanceids,Values=the  
ID of the instance you created earlier
```

Remote Management

Systems Manager Run Command lets you remotely and securely manage the configuration of your Amazon EC2 instances, virtual machines (VMs) and servers in hybrid environments, or VMs from other cloud providers. Run Command enables you to automate common administrative tasks and perform ad hoc configuration changes at scale. You can use Run Command from the EC2 console, the AWS Command Line Interface, Windows PowerShell, or the AWS SDKs. Run Command is offered at no additional cost.

Administrators use Run Command to perform the following types of tasks: monitor their systems, install applications on their machines, inventory machines to see which applications are missing or need patching, patch machines, build a deployment pipeline, bootstrap applications, and join instances to a domain, to name a few.

Information for Linux Users

For information about Run Command for Linux, see [Remote Management](#) in the *Amazon EC2 User Guide for Linux Instances*.

Run Command Features and Benefits

Features	Benefits
Fully-managed AWS service offered at no additional cost.	Is available on Linux and Windows and works with EC2, servers and VMs in your hybrid environment, or VMs from other cloud providers.
Automates administrative and configuration tasks at scale.	Can be configured to send notifications about command and configuration results using CloudWatch Events or Amazon SNS.
Provides a single view into configuration changes at scale.	Uses SSM documents, which enable you to quickly define and execute commands.

Features	Benefits
Improves administration security because there is not need to connect to your machines using Secure Shell (SSH) or Remote Desktop Protocol (RDP).	Includes pre-defined SSM documents and the ability to create your own, which you can share across accounts or publicly
Offers delegated access control.	Includes auditing and access control using AWS Identity and Access Management (IAM).

Getting Started on EC2 Instances

The following table includes information to help you get started with Run Command.

Topic	Details
Tutorial: Remotely Manage Your Amazon EC2 Instances (p. 41)	The tutorial shows you how to quickly send a command using Run Command with AWS Tools for Windows PowerShell.
Amazon EC2 Run Command Components and Concepts (p. 439)	Learn about Run Command features and concepts.
Systems Manager Prerequisites (p. 394)	Verify that your instances meet the minimum requirements for Run Command.
Executing a Command Using Amazon EC2 Run Command (p. 442)	Execute commands from the EC2 console and create a command that you can execute from the AWS Command Line Interface.
Amazon EC2 Run Command Walkthroughs (p. 489)	Learn from these detailed walkthroughs how to execute commands using Run Command from either the Amazon EC2 console or AWS Tools for Windows.

Getting Started in a Hybrid Environment

The following table includes information to help you get started with Run Command.

Topic	Details
Amazon EC2 Run Command Components and Concepts (p. 439)	Learn about Run Command features and concepts.
Setting Up Systems Manager in Hybrid Environments (p. 397)	Register on-premises servers and VMs or servers hosted by other cloud providers with AWS so that you can manage them using Run Command.
Executing a Command Using Amazon EC2 Run Command (p. 442)	Execute commands from the EC2 console and create a command that you can execute from the AWS Command Line Interface.

Related Content

- [Configuring Access to Systems Manager \(p. 400\)](#)
- [Creating SSM Documents \(p. 478\)](#)
- [Sharing SSM Documents \(p. 484\)](#)
- [Command Status and Monitoring \(p. 501\)](#)
- [Amazon EC2 Simple Systems Manager API Reference](#)
- [SSM AWS Tools for Windows PowerShell Reference](#)
- [SSM AWS CLI Reference](#)
- [AWS SDKs](#)

Amazon EC2 Run Command Components and Concepts

As you get started with Amazon EC2 Run Command, you'll benefit from understanding the components and concepts of this feature.

Component/Concept	Details
Amazon EC2 Simple Systems Manager (SSM)	Run Command is a component of SSM. Run Command uses the SSM API. For more information, see Amazon EC2 Simple Systems Manager API Reference .
Servers and VMs in Your Hybrid Environment	Amazon EC2 Run Command lets you remotely and securely manage on-premises servers and virtual machines (VMs) and VMs from other cloud providers. By setting up Run Command in this way, you create a consistent and secure way to remotely manage your on-premises and cloud workloads using the same tools or scripts. After you configure a server or VM in your hybrid environment for Run Command it is called a <i>managed instance</i> and is listed in the EC2 console like your other EC2 instances. For more information, see Setting Up Systems Manager in Hybrid Environments (p. 397) .
Commands	You can configure managed instances by sending commands from your local machine. You don't need to log on locally to configure your machines. You can send commands using one of the following: the Command History page of the Amazon EC2 console , AWS Tools for Windows PowerShell, the AWS Command Line Interface (AWS CLI), the SSM API, or Amazon SDKs. For more information, see SSM AWS Tools for Windows PowerShell Reference , SSM AWS CLI Reference , and the AWS SDKs .
SSM Documents	An SSM document defines the plugins to run and the parameters to use when a command executes on a machine. When you execute a command, you specify the SSM document that Run Command uses. Run Command includes

Component/Concept	Details
	pre-defined documents that enable you to quickly perform common tasks on a machine. You can also create your own SSM documents. The first time you execute a command from a new SSM document, the system stores the document with your AWS account. For more information, see Creating SSM Documents (p. 478) .
SSM Agent	The SSM agent is AWS software that you install on servers and VMs in your hybrid environment. The agent processes Run Command requests and configures your machine as specified in the request. For information, see Installing and Updating SSM Agent (p. 396) .
EC2Config service for EC2 Windows Instances	On Amazon EC2 Windows instances, the EC2Config service processes Run Command requests and configures your machine as specified in the request. By default, the EC2Config service is installed on all Windows Amazon Machines Images (AMIs), excluding Window Server 2016. If your instance was launched from a recent AMI, you don't need to download and install the EC2Config service. If you want, you can upgrade the EC2Config service on your EC2 instances. For information, see Installing the Latest Version of EC2Config (p. 295) .
IAM Roles and Polices	AWS user accounts and instances must be configured with AWS Identity and Access Management (IAM) roles and trust policies that enable them to communicate with the SSM API. For more information, see Configuring Access to Systems Manager (p. 400) .

How It Works

After you verify prerequisites for your instances, you send a command from your local machine. The SSM service verifies the integrity of the command and any parameters and then forwards the request to the Amazon EC2 messaging service. The SSM agent running each instance (or EC2Config service on EC2 Windows instances) communicates with the EC2 messaging service to retrieve commands. The agent processes the command, configures the instance as specified, and logs the output and results.

Note

The agent attempts to execute each command once. You can send multiple commands at the same time.

The system manages the queuing, execution, cancellation, and reporting of each command. However, the order of command execution is not guaranteed. By default, Run Command uses throttle limits to ensure that no more than 60 commands are issued per minute per instance. If an instance is not running or is unresponsive when you execute a command, the system queues the command and attempts to run it when the instance is responsive. By default, the system will queue a command and attempt to run it for up to 31 days after request. For more information about command status, see [Command Status and Monitoring \(p. 501\)](#).

Run Command reports the status and results of each command for each instance, server, or VM. Run Command stores the command history for 30 days. The information is also stored in AWS CloudTrail and remains available until you delete the data. For more information, see [Auditing API Calls](#) in the *Amazon EC2 Simple Systems Manager API Reference*.

More about SSM Documents

After you configure Run Command prerequisites, you determine what type of configuration change you want to make on your instance and which SSM document will enable you to make that change. Run Command includes pre-defined SSM documents that enable you to quickly execute commands on instances. The commands available to you depend on the permissions your administrator specified for you. Any command that begins with AWS-* uses a pre-defined SSM document provided by AWS. A developer or administrator can create additional documents and provision these for you based on your permissions. For more information, see [Creating SSM Documents](#) (p. 478).

Important

Only trusted administrators should be allowed to use AWS pre-configured documents. The commands or scripts specified in SSM documents run with administrative privilege on your instances because the EC2Config service runs in the Local System account. If a user has permission to execute any of the pre-defined SSM documents (any document that begins with AWS-*), then that user also has administrator access to the instance. For all other users, you should create restrictive documents and share them with specific users. For more information about restricting access to Run Command, see [Configuring Access to Systems Manager](#) (p. 400).

Run Command includes the following pre-configured SSM documents.

Amazon Pre-configured SSM documents for Windows

Name	Description
AWS-JoinDirectoryServiceDomain	Join an AWS Directory
AWS-RunPowerShellScript	Run PowerShell commands or scripts
AWS-UpdateEC2Config	Update the EC2Config service
AWS-ConfigureWindowsUpdate	Configure Windows Update settings
AWS-InstallApplication	Install, repair, or uninstall software using an MSI package
AWS-InstallPowerShellModule	Install PowerShell modules
AWS-ConfigureCloudWatch	Configure Amazon CloudWatch Logs to monitor applications and systems
AWS-ListWindowsInventory	Collect information about an EC2 instance running in Windows.
AWS-FindWindowsUpdates	Scan an instance and determines which updates are missing.
AWS-InstallMissingWindowsUpdates	Install missing updates on your EC2 instance.
AWS-InstallSpecificWindowsUpdates	Install one or more specific updates.

You can select a document from a list in the [Amazon EC2 console](#) or use a `list documents` command to view a list a commands available to you in either the AWS CLI or AWS Tools for Windows PowerShell.

Executing a Command Using Amazon EC2 Run Command

You can execute commands using the **Command History** page in the [Amazon EC2 console](#), [AWS Tools for Windows PowerShell](#), the [AWS Command Line Interface](#), or programmatically using the [Amazon EC2 Simple Systems Manager API Reference](#) and the [AWS SDKs](#).

The section describes how to execute commands using the Amazon EC2 console and how to send commands to tens, hundreds, or thousands of instances while controlling the execution of those commands. For examples of how to send commands using AWS Tools for Windows PowerShell, see [Amazon EC2 Run Command Walkthrough Using the AWS Tools for Windows PowerShell](#) (p. 494).

Topics

- [Executing Commands Using the Amazon EC2 Console](#) (p. 442)
- [Sending a Command to Multiple Instances](#) (p. 476)

Executing Commands Using the Amazon EC2 Console

The topics in this section walk you through the process of executing a command with Run Command using the Amazon EC2 console.

Topics

- [Running PowerShell Commands or Scripts with Amazon EC2 Run Command](#) (p. 442)
- [Installing Applications Using Amazon EC2 Run Command](#) (p. 445)
- [Installing PowerShell Modules with Amazon EC2 Run Command](#) (p. 448)
- [Joining EC2 Instances to a Domain Using Amazon EC2 Run Command](#) (p. 451)
- [Uploading Logs from EC2 Instances to Amazon CloudWatch Using Amazon EC2 Run Command](#) (p. 454)
- [Enabling or Disabling Windows Updates Using Amazon EC2 Run Command](#) (p. 467)
- [Updating the SSM Agent Using Amazon EC2 Run Command](#) (p. 470)
- [Inventory an Amazon EC2 Instance for Windows Using Amazon EC2 Run Command](#) (p. 473)
- [Managing Updates for an EC2 Windows Instance Using Amazon EC2 Run Command](#) (p. 476)

Running PowerShell Commands or Scripts with Amazon EC2 Run Command

Use the [AWS-RunPowerShellScript](#) document to send commands to your EC2 instances, or specify the path to a script to run on your instances. For example, you can send commands like **dir c:**, **ipconfig**, or **net stop *service_name***. You can also specify the location of a script to run. For example, **c:\script.ps1** or ***network_share*\script.ps1**.

Note

This procedure does not include information about how to configure Run Command for Amazon SNS notifications. To learn more about the Amazon SNS notification fields on the **Command History** page and how to execute commands that return notifications, see [Getting Amazon SNS Notifications When a Command Changes Status](#) (p. 505).

To run PowerShell commands or scripts using Run Command

1. Open the [Amazon EC2 console](#) and choose **Command History** in the navigation pane.
2. Choose **Run a Command**.
3. In the **Command document** list, choose **AWS-RunPowerShellScript**.

4. Choose **Select target instances** to select the instances where you want the command to run. If you do not see a complete list of instances, the missing instances might not be configured properly for Run Command. For more information, see [Systems Manager Prerequisites \(p. 394\)](#).
5. Type a command or the path to a script in the **Commands** field.
6. (Optional) In the **Working Directory** field, type the path to the folder on your EC2 instances where you want to run the command. For example, C:\temp.
7. (Optional) In the **Execution Timeout** field, type the number of seconds the EC2Config service will attempt to run the command before it times out and fails.
8. (Optional) In the **Comment** field, type information you want to provide about this command. Comments are stored in the log file and appear in the Command Invocation List in the console.

Tip

We recommend that you enter specific comments about each command you run. The list of commands that you send can grow quickly, and the **Comment** field can help you identify commands that you want to monitor.

9. In the **Timeout (seconds)** field, type the number of seconds Run Command should attempt to reach instances before an instance is considered unreachable and the command execution fails. The minimum is 30 seconds. The maximum is 30 days. The default value is 10 minutes.
10. In the **Amazon S3 bucket** field type the name of an Amazon S3 bucket where you want to store the output of the command.

Important

The Run Command **Output** page in the Amazon EC2 console truncates output after 2500 characters. Configure an Amazon S3 bucket before executing commands using Run Command. If your command output was longer than 2500 characters, you can view the full output in your Amazon S3 bucket. For more information, see [Create a Bucket](#).

11. In the **S3 key prefix**, type the name of a subfolder in the Amazon S3 bucket. This subfolder can help you organize Run Command output.

Note

The section called **AWS Command Line Interface command** displays a usable CLI script that is generated based on the parameters you entered.

After you execute a command using Run Command, the system returns you to the commands list.

Important

The Amazon EC2 console truncates all command output beyond 2500 characters. If your command output was longer than 2500 characters, you can view the full output in your S3 bucket.

[Canceling a Command](#)

You can attempt to cancel a command as long as the service shows that it is in either a Pending or Executing state. However, even if a command is still in one of these states, we cannot guarantee that the command will be terminated and the underlying process stopped.

To cancel a command using the console

1. Open the [Amazon EC2 console](#) and choose **Command History** in the navigation pane.
2. Select the command invocation that you want to cancel.
3. Choose **Actions** and then choose **Cancel command**.

To cancel a command using the AWS CLI

Use the following command.

```
aws ssm cancel-command --command-id "command ID" --instance-ids "instance ID"
```

For information about the status of a cancelled command, see [Command Status and Monitoring](#) (p. 501).

View Command Output

Use the following procedure to view the results of command execution in the EC2 console.

To view command output

1. In the Amazon EC2 console, select a command in the list.
2. Choose the **Output** tab.
3. Choose **View Output**.

The screenshot shows the Amazon EC2 console interface. At the top, there is a 'Run a command' button and an 'Actions' dropdown menu. Below this is a search bar labeled 'Filter by attributes'. A table lists several commands with columns for Command ID, Instance ID, Document name, Status, Requested date, and Comment. The first row is selected, and its 'Output' tab is active. Below the table, the command ID and instance ID are displayed. The 'Output' section shows a table with columns for Plugin name, Status, Response code, Start Time, Finish Time, and Output. The first row in this table shows 'aws.runPower...' with a 'Success' status and a 'View Output' link.

Command ID	Instance ID	Document name	Status	Requested date	Comment
ca4b10c6-cee1-437...	i-d583f76a	AWS-RunPowerSh...	Success	October 20, 2015 at...	getting list of proce
561e5f4a-27d2-419...	i-d583f76a	AWS-RunPowerSh...	Success	October 20, 2015 at...	ipconfig on the box
d5b589e6-ad94-4d...	i-d583f76a	AWS-RunPowerSh...	Success	October 20, 2015 at...	Getting services or

Command ID: ca4b10c6-cee1-437b-9f70-9746da1477e7 Instance ID: i-d583f76a

Plugin name	Status	Response code	Start Time	Finish Time	Output
aws.runPower...	Success	0	October 20, 2015 at 4:15:58 PM...	October 20, 2015 at 4:15:59 PM...	View Output

4. The command output page shows the results of your command execution.

Commands > Output

Output for aws:runPowerShellScript

Handles	NPM (K)	PM (K)	WS (K)	VM (M)	CPU (s)	Id	ProcessName
40	4	612	2752	27	0.00	2900	conhost
194	11	1712	3776	46	0.06	520	csrss
86	8	1236	3576	43	0.08	584	csrss
209	11	1700	21036	66	2.06	880	csrss
173	14	10764	1608	88	0.03	872	dwm
208	25	23216	45228	164	0.58	2560	dwm
836	63	47216	21880	643	24.92	2276	Ec2Config
1344	64	52420	108976	495	11.92	2776	explorer
0	0	0	4	0		0	Idle
105	8	1196	4844	33	0.02	1176	LiteAgent
268	21	12692	24108	138	0.08	2148	LogonUI
797	18	4336	10892	40	0.72	688	lsass
159	12	2108	3760	41	0.02	2888	msdtc
362	23	56848	53412	593	0.30	3688	powershell
219	11	1976	9600	87	0.28	732	rdpclip
650	48	108564	105064	773	4.58	3472	ServerManager
209	10	2420	7148	25	2.27	680	services
55	2	276	1048	4	0.03	436	smss
424	23	4536	12124	91	1.94	1148	spoolsv

-----Output truncated-----

For information about how to run commands using Windows PowerShell, see [Amazon EC2 Run Command Walkthrough Using the AWS Tools for Windows PowerShell \(p. 494\)](#) or the [AWS Tools for Windows PowerShell Reference](#). For information about how to run commands using the AWS CLI, see the [SSM CLI Reference](#).

Installing Applications Using Amazon EC2 Run Command

You can use the `AWS-InstallApplication` document to install, repair, or uninstall applications on EC2 instances. You must specify the URL or the path to an `.msi` file.

Note

This procedure does not include information about how to configure Run Command for Amazon SNS notifications. To learn more about the Amazon SNS notification fields on the **Command History** page and how to execute commands that return notifications, see [Getting Amazon SNS Notifications When a Command Changes Status \(p. 505\)](#).

To install, repair, or uninstall applications using Run Command

1. Open the [Amazon EC2 console](#) and choose **Command History** in the navigation pane.
2. Choose **Run a Command**.
3. In the **Command document** list, choose **AWS-InstallApplication**.
4. Choose **Select target instances** to select the instances where you want the command to run. If you do not see a complete list of instances, the missing instances might not be configured properly for Run Command. For more information, see [Systems Manager Prerequisites \(p. 394\)](#).
5. In the **Action** list, choose the task you want to perform.
6. (Optional) In the **Parameters** field, type parameters for the installer.
7. In the **Source** field, type either the URL or the path to an `.msi` file. For example:

URL: `http://sdk-for-net.amazonwebservices.com/latest/AWSToolsAndSDKForNet.msi`

File: `file://c:\temp\AWSToolsAndSDKForNet.msi`

8. (Optional) In the **Source Hash** field, type an SHA256 hash for the installer.

- (Optional) In the **Comment** field, type information you want to provide about this command. Comments are stored in the log file and appear in the Command Invocation List in the Amazon EC2 console.

Tip

We recommend that you enter specific comments about each command you run. The list of commands that you send can grow quickly, and the **Comment** field can help you identify commands that you want to monitor.

- In the **Timeout (seconds)** field, type the number of seconds Run Command should attempt to reach instances before an instance is considered unreachable and the command execution fails. The minimum is 30 seconds. The maximum is 30 days. The default value is 10 minutes.
- In the **S3 bucket** field, type the name of an Amazon S3 bucket where you want to store the output of the command.

Important

The Run Command **Output** page in the Amazon EC2 console truncates output after 2500 characters. Configure an Amazon S3 bucket before executing commands using Run Command. If your command output was longer than 2500 characters, you can view the full output in your Amazon S3 bucket. For more information, see [Create a Bucket](#).

- In the **S3 key prefix**, field enter the name of a subfolder in the Amazon S3 bucket. This subfolder can help you organize Run Command output.

Note

The section called **AWS Command Line Interface command** displays a usable CLI script that is generated based on the parameters you entered.

After you execute a command using Run Command, the system returns you to the commands list.

Important

The Amazon EC2 console truncates all command output beyond 2500 characters. If your command output was longer than 2500 characters, you can view the full output in your S3 bucket.

Canceling a Command

You can attempt to cancel a command as long as the service shows that it is in either a Pending or Executing state. However, even if a command is still in one of these states, we cannot guarantee that the command will be terminated and the underlying process stopped.

To cancel a command using the console

- Open the [Amazon EC2 console](#) and choose **Command History** in the navigation pane.
- Select the command invocation that you want to cancel.
- Choose **Actions** and then choose **Cancel command**.

To cancel a command using the AWS CLI

Use the following command.

```
aws ssm cancel-command --command-id "command ID" --instance-ids "instance ID"
```

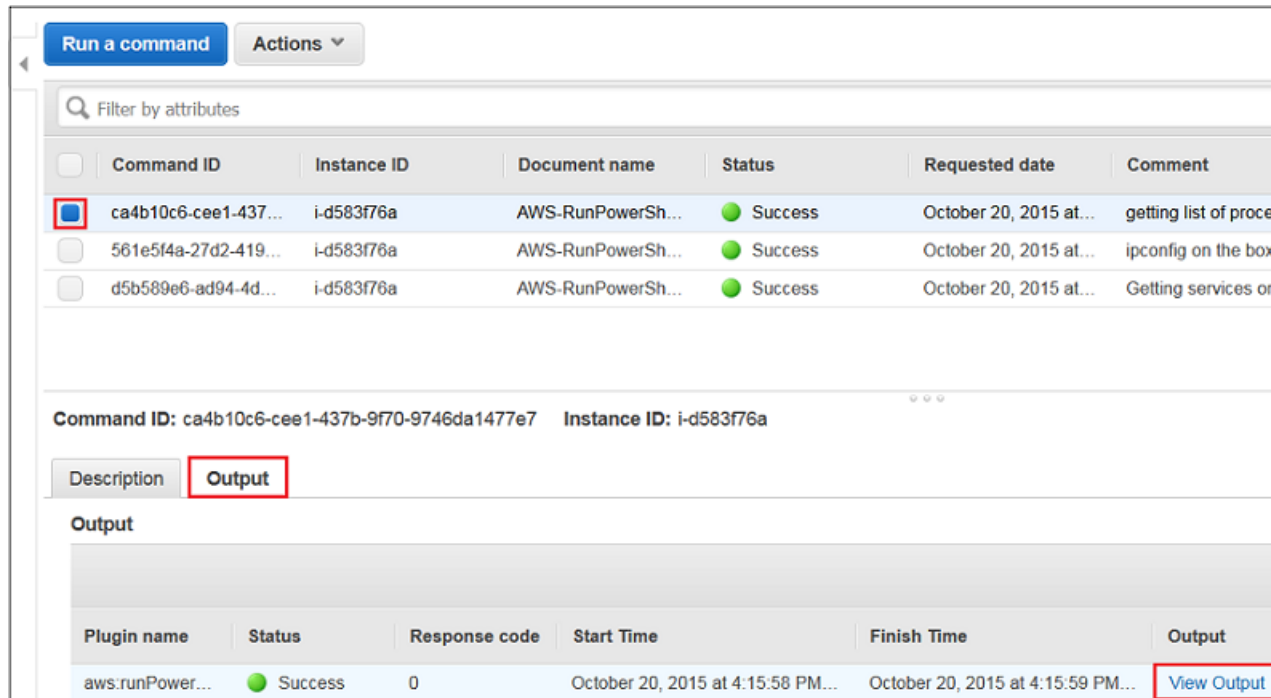
For information about the status of a cancelled command, see [Command Status and Monitoring \(p. 501\)](#).

View Command Output

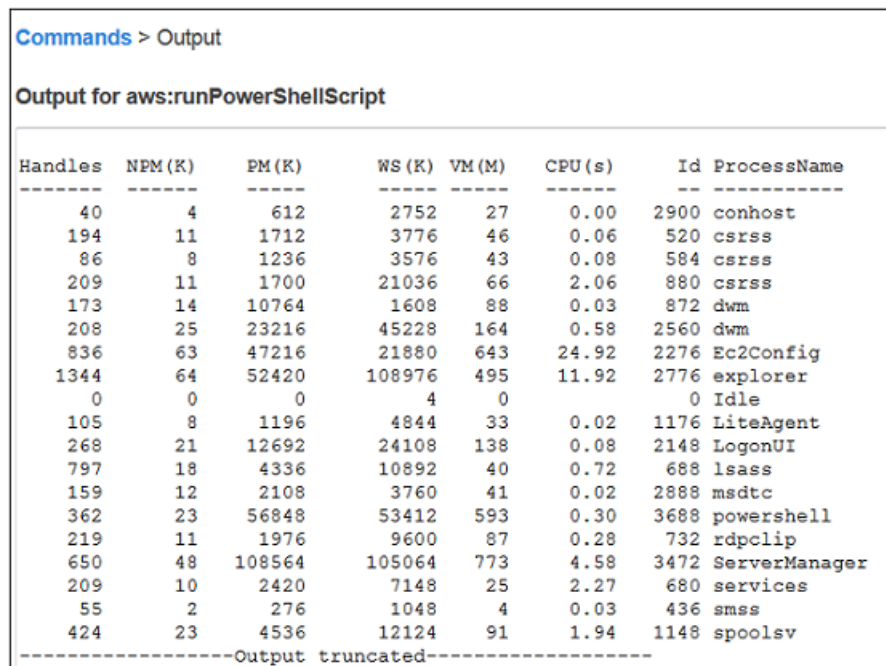
Use the following procedure to view the results of command execution in the EC2 console.

To view command output

1. In the Amazon EC2 console, select a command in the list.
2. Choose the **Output** tab.
3. Choose **View Output**.



4. The command output page shows the results of your command execution.



For information about how to run commands using Windows PowerShell, see [Amazon EC2 Run Command Walkthrough Using the AWS Tools for Windows PowerShell \(p. 494\)](#) or the [AWS Tools for Windows PowerShell Reference](#). For information about how to run commands using the AWS CLI, see the [SSM CLI Reference](#).

Installing PowerShell Modules with Amazon EC2 Run Command

You can use the `AWS-InstallPowerShellModule` document to install PowerShell modules on EC2 instances. You can also specify PowerShell commands to run after the module has been installed. For example, you could install the EZOut module for flexible PowerShell formatting and then run a command to install a Windows feature like XPS Viewer to view files you create with EZOut.

Note

This procedure does not include information about how to configure Run Command for Amazon SNS notifications. To learn more about the Amazon SNS notification fields on the **Command History** page and how to execute commands that return notifications, see [Getting Amazon SNS Notifications When a Command Changes Status \(p. 505\)](#).

To install PowerShell modules using Run Command

1. Open the [Amazon EC2 console](#) and choose **Command History** in the navigation pane.
2. Choose **Run a Command**.
3. In the **Command document** list, choose **AWS-InstallPowerShellModule**.
4. Choose **Select target instances** to select the instances where you want the command to run. If you do not see a complete list of instances, the missing instances might not be configured properly for Run Command. For more information, see [Systems Manager Prerequisites \(p. 394\)](#).
5. (Optional) In the **Working Directory** field, type the path to the folder on your EC2 instances where you want to run the command. For example, `C:\temp`.
6. In the **Source** field, type either the URL or the path to .zip file. For example:

URL: `http://www.microsoft.com/en-us/download/SomePSModule.msi`

File: `file://c:\temp\EZOut.zip`

7. (Optional) In the **Source Hash** field, type an SHA256 hash for the .zip file.
8. (Optional) type a command in the **Commands** field. Choose the plus sign to add additional commands.
9. (Optional) In the **Execution Timeout** field, type the number of seconds the EC2Config service will attempt to run the command before it times out and fails.
10. (Optional) In the **Comment** field, type information you want to provide about this command. Comments are stored in the log file and appear in the Command Invocation List in the Amazon EC2 console.

Tip

We recommend that you enter specific comments about each command you run. The list of commands that you send can grow quickly, and the **Comment** field can help you identify commands that you want to monitor.

11. In the **Timeout (seconds)** field, type the number of seconds Run Command should attempt to reach instances before an instance is considered unreachable and the command execution fails. The minimum is 30 seconds. The maximum is 30 days. The default value is 10 minutes.
12. In the **S3 bucket** field, type the name of an Amazon S3 bucket where you want to store the output of the command.

Important

The Run Command **Output** page in the Amazon EC2 console truncates output after 2500 characters. Configure an Amazon S3 bucket before executing commands using Run Command. If your command output was longer than 2500 characters, you can view the full output in your Amazon S3 bucket. For more information, see [Create a Bucket](#).

13. In the **S3 key prefix**, field, type the name of a subfolder in the Amazon S3 bucket. This subfolder can help you organize Run Command output.

Note

The section called **AWS Command Line Interface command** displays a usable CLI script that is generated based on the parameters you entered.

After you execute a command using Run Command, the system returns you to the commands list.

Important

The Amazon EC2 console truncates all command output beyond 2500 characters. If your command output was longer than 2500 characters, you can view the full output in your S3 bucket.

Canceling a Command

You can attempt to cancel a command as long as the service shows that it is in either a Pending or Executing state. However, even if a command is still in one of these states, we cannot guarantee that the command will be terminated and the underlying process stopped.

To cancel a command using the console

1. Open the [Amazon EC2 console](#) and choose **Command History** in the navigation pane.
2. Select the command invocation that you want to cancel.
3. Choose **Actions** and then choose **Cancel command**.

To cancel a command using the AWS CLI

Use the following command.

```
aws ssm cancel-command --command-id "command ID" --instance-ids "instance ID"
```

For information about the status of a cancelled command, see [Command Status and Monitoring \(p. 501\)](#).

View Command Output

Use the following procedure to view the results of command execution in the EC2 console.

To view command output

1. In the Amazon EC2 console, select a command in the list.
2. Choose the **Output** tab.
3. Choose **View Output**.

Run a command Actions

Filter by attributes

Command ID	Instance ID	Document name	Status	Requested date	Comment
ca4b10c6-cee1-437...	i-d583f76a	AWS-RunPowerSh...	Success	October 20, 2015 at...	getting list of proce
561e5f4a-27d2-419...	i-d583f76a	AWS-RunPowerSh...	Success	October 20, 2015 at...	ipconfig on the box
d5b589e6-ad94-4d...	i-d583f76a	AWS-RunPowerSh...	Success	October 20, 2015 at...	Getting services on

Command ID: ca4b10c6-cee1-437b-9f70-9746da1477e7 Instance ID: i-d583f76a

Description Output

Output

Plugin name	Status	Response code	Start Time	Finish Time	Output
aws:runPower...	Success	0	October 20, 2015 at 4:15:58 PM...	October 20, 2015 at 4:15:59 PM...	View Output

- The command output page shows the results of your command execution.

Commands > Output

Output for aws:runPowerShellScript

Handles	NPM (K)	PM (K)	WS (K)	VM (M)	CPU (s)	Id	ProcessName
40	4	612	2752	27	0.00	2900	conhost
194	11	1712	3776	46	0.06	520	csrss
86	8	1236	3576	43	0.08	584	csrss
209	11	1700	21036	66	2.06	880	csrss
173	14	10764	1608	88	0.03	872	dwm
208	25	23216	45228	164	0.58	2560	dwm
836	63	47216	21880	643	24.92	2276	Ec2Config
1344	64	52420	108976	495	11.92	2776	explorer
0	0	0	4	0		0	Idle
105	8	1196	4844	33	0.02	1176	LiteAgent
268	21	12692	24108	138	0.08	2148	LogonUI
797	18	4336	10892	40	0.72	688	lsass
159	12	2108	3760	41	0.02	2888	msdtc
362	23	56848	53412	593	0.30	3688	powershell
219	11	1976	9600	87	0.28	732	rdpclip
650	48	108564	105064	773	4.58	3472	ServerManager
209	10	2420	7148	25	2.27	680	services
55	2	276	1048	4	0.03	436	smss
424	23	4536	12124	91	1.94	1148	spoolsv

-----Output truncated-----

For information about how to run commands using Windows PowerShell, see [Amazon EC2 Run Command Walkthrough Using the AWS Tools for Windows PowerShell \(p. 494\)](#) or the [AWS Tools for Windows PowerShell Reference](#). For information about how to run commands using the AWS CLI, see the [SSM CLI Reference](#).

Joining EC2 Instances to a Domain Using Amazon EC2 Run Command

You can use the `AWS-JoinDirectoryServiceDomain` command to join an instance to an AWS Directory Service domain. Before executing this command you must [create a directory](#). We recommend that you learn more about the AWS Directory Service. For more information, see [What Is AWS Directory Service?](#).

Note

This procedure does not include information about how to configure Run Command for Amazon SNS notifications. To learn more about the Amazon SNS notification fields on the [Command History](#) page and how to execute commands that return notifications, see [Getting Amazon SNS Notifications When a Command Changes Status \(p. 505\)](#).

To join an instance to a domain using Run Command

1. Open the [Amazon EC2 console](#) and choose **Command History** in the navigation pane.
2. Choose **Run a Command**.
3. In the **Command document** list, choose **AWS-JoinDirectoryServiceDomain**.
4. Choose **Select target instances** to select the instances where you want the command to run. If you do not see a complete list of instances, the missing instances might not be configured properly for Run Command. For more information, see [Systems Manager Prerequisites \(p. 394\)](#).
5. In the **Directory ID** field, type the ID of an AWS directory. For example: d-1234567890.
6. In the **Directory Name** field, type the directory name. For example: example.com.
7. In the **Directory OU** field, type the organizational unit (OU) and directory components (DC) for the directory; for example, OU=Computers,OU=example,DC=test,DC=example,DC=com.
8. (Optional) In the **DNS IP Addresses** field, type an IP address. For example: 198.51.100.1. Choose the plus sign to add more IP addresses.
9. (Optional) In the **Comment** field, type information you want to provide about this command. Comments are stored in the log file and appear in the Command Invocation List in the Amazon EC2 console.

Tip

We recommend that you enter specific comments about each command you run. The list of commands that you send can grow quickly, and the **Comment** field can help you identify commands that you want to monitor.

10. In the **Timeout (seconds)** field, type the number of seconds Run Command should attempt to reach instances before an instance is considered unreachable and the command execution fails. The minimum is 30 seconds. The maximum is 30 days. The default value is 10 minutes.
11. In the **S3 bucket** field, type the name of an Amazon S3 bucket where you want to store the output of the command.

Important

The Run Command **Output** page in the Amazon EC2 console truncates output after 2500 characters. Configure an Amazon S3 bucket before executing commands using Run Command. If your command output was longer than 2500 characters, you can view the full output in your Amazon S3 bucket. For more information, see [Create a Bucket](#).

12. In the **S3 key prefix**, field, type the name of a subfolder in the Amazon S3 bucket. This subfolder can help you organize Run Command output.

Note

The section called **AWS Command Line Interface command** displays a usable CLI script that is generated based on the parameters you entered.

After you execute a command using Run Command, the system returns you to the commands list.

Important

The Amazon EC2 console truncates all command output beyond 2500 characters. If your command output was longer than 2500 characters, you can view the full output in your S3 bucket.

Canceling a Command

You can attempt to cancel a command as long as the service shows that it is in either a Pending or Executing state. However, even if a command is still in one of these states, we cannot guarantee that the command will be terminated and the underlying process stopped.

To cancel a command using the console

1. Open the [Amazon EC2 console](#) and choose **Command History** in the navigation pane.
2. Select the command invocation that you want to cancel.
3. Choose **Actions** and then choose **Cancel command**.

To cancel a command using the AWS CLI

Use the following command.

```
aws ssm cancel-command --command-id "command ID" --instance-ids "instance ID"
```

For information about the status of a cancelled command, see [Command Status and Monitoring \(p. 501\)](#).

View Command Output

Use the following procedure to view the results of command execution in the EC2 console.

To view command output

1. In the Amazon EC2 console, select a command in the list.
2. Choose the **Output** tab.
3. Choose **View Output**.

Run a command Actions

Filter by attributes

Command ID	Instance ID	Document name	Status	Requested date	Comment
ca4b10c6-cee1-437...	i-d583f76a	AWS-RunPowerSh...	Success	October 20, 2015 at...	getting list of proce
561e5f4a-27d2-419...	i-d583f76a	AWS-RunPowerSh...	Success	October 20, 2015 at...	ipconfig on the box
d5b589e6-ad94-4d...	i-d583f76a	AWS-RunPowerSh...	Success	October 20, 2015 at...	Getting services on

Command ID: ca4b10c6-cee1-437b-9f70-9746da1477e7 Instance ID: i-d583f76a

Description Output

Output

Plugin name	Status	Response code	Start Time	Finish Time	Output
aws:runPower...	Success	0	October 20, 2015 at 4:15:58 PM...	October 20, 2015 at 4:15:59 PM...	View Output

4. The command output page shows the results of your command execution.

Commands > Output

Output for aws:runPowerShellScript

Handles	NPM (K)	PM (K)	WS (K)	VM (M)	CPU (s)	Id	ProcessName
40	4	612	2752	27	0.00	2900	conhost
194	11	1712	3776	46	0.06	520	csrss
86	8	1236	3576	43	0.08	584	csrss
209	11	1700	21036	66	2.06	880	csrss
173	14	10764	1608	88	0.03	872	dwm
208	25	23216	45228	164	0.58	2560	dwm
836	63	47216	21880	643	24.92	2276	Ec2Config
1344	64	52420	108976	495	11.92	2776	explorer
0	0	0	4	0		0	Idle
105	8	1196	4844	33	0.02	1176	LiteAgent
268	21	12692	24108	138	0.08	2148	LogonUI
797	18	4336	10892	40	0.72	688	lsass
159	12	2108	3760	41	0.02	2888	msdtc
362	23	56848	53412	593	0.30	3688	powershell
219	11	1976	9600	87	0.28	732	rdpclip
650	48	108564	105064	773	4.58	3472	ServerManager
209	10	2420	7148	25	2.27	680	services
55	2	276	1048	4	0.03	436	smss
424	23	4536	12124	91	1.94	1148	spoolsv

-----Output truncated-----

For information about how to run commands using Windows PowerShell, see [Amazon EC2 Run Command Walkthrough Using the AWS Tools for Windows PowerShell \(p. 494\)](#) or the [AWS Tools for Windows PowerShell Reference](#). For information about how to run commands using the AWS CLI, see the [SSM CLI Reference](#).

Uploading Logs from EC2 Instances to Amazon CloudWatch Using Amazon EC2 Run Command

You can use Run Command to configure integration with Amazon CloudWatch and Amazon CloudWatch Logs on multiple instances to monitor their log files. You can send Windows Server messages in the application, system, security, and Event Tracing (Windows) logs to Amazon CloudWatch Logs. When you enable logging for the first time, Run Command sends all logs generated within one minute from the time that you start uploading logs for the application, system, security, and ETW logs. Logs that occurred before this time are not included. If you disable logging and then later re-enable logging, Run Command sends logs from the time logging was disabled. For any custom log files and Internet Information Services (IIS) logs, Run Command reads the log files from the beginning. In addition, Run Command can also send performance counter data to CloudWatch.

If you previously enabled CloudWatch integration in EC2Config, the Run Command settings override any settings stored locally on the instance in the **C:\Program Files\Amazon\Ec2ConfigService\Settings\AWS.EC2.Windows.CloudWatch.json** file.

When you upload logs to CloudWatch you have the option to specify properties in a JSON code sample and paste the sample into the **Properties** field. If you have an existing JSON sample for using Amazon Simple System Manager (SSM) to upload logs to CloudWatch, you can specify properties in the following sample and use it in **Properties** field. Or, you can specify properties as described in this section and copy/paste it. To learn about the properties and the values you can specify, see [aws:cloudWatch](#) in the API Reference.

Contents

- [Create a JSON File \(p. 454\)](#)
- [Configure the Region and Namespace for CloudWatch and CloudWatch Logs \(p. 456\)](#)
- [Configure the Performance Counters and Logs to Send to CloudWatch and CloudWatch Logs \(p. 458\)](#)
- [Configure the Flow Control \(p. 463\)](#)
- [Upload Logs to Amazon CloudWatch Using Run Command \(p. 464\)](#)

Create a JSON File

If you don't already have a JSON file, you must create one. Copy and paste the following sample into a text editor and save the file with a `.json` file extension.

For more information about the structure of the JSON for an SSM document, see [SSM Documents](#) in the *Amazon EC2 Simple Systems Manager API Reference*.

```
{
  "EngineConfiguration": {
    "PollInterval": "00:00:15",
    "Components": [
      {
        "Id": "ApplicationEventLog",
        "FullName":
"AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
        "Parameters": {
          "LogName": "Application",
          "Levels": "value"
        }
      },
      {
        "Id": "SystemEventLog",
        "FullName":
"AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
```

Amazon Elastic Compute Cloud
User Guide for Windows Instances
Executing Commands

```
    "Parameters": {
      "LogName": "System",
      "Levels": "value"
    }
  },
  {
    "Id": "SecurityEventLog",
    "FullName":
"AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
      "LogName": "Security",
      "Levels": "value"
    }
  },
  {
    "Id": "ETW",
    "FullName":
"AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
      "LogName": "Microsoft-Windows-WinINet/Analytic",
      "Levels": "value"
    }
  },
  {
    "Id": "IISLogs",
    "FullName":
"AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
      "LogDirectoryPath": "path",
      "TimestampFormat": "value",
      "Encoding": "value",
      "Filter": "value",
      "CultureName": "locale",
      "TimeZoneKind": "value",
      "LineCount": "value"
    }
  },
  {
    "Id": "CustomLogs",
    "FullName":
"AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
      "LogDirectoryPath": "path",
      "TimestampFormat": "value",
      "Encoding": "value",
      "Filter": "value",
      "CultureName": "locale",
      "TimeZoneKind": "value",
      "LineCount": "value"
    }
  },
  {
    "Id": "PerformanceCounter",
    "FullName":
"AWS.EC2.Windows.CloudWatch.PerformanceCounterComponent.PerformanceCounterInputComponent,A
    "Parameters": {
      "CategoryName": "name",
      "CounterName": "name",
      "InstanceName": "name",
```

```
        "MetricName": "name",
        "Unit": "unit",
        "DimensionName": "name",
        "DimensionValue": "value"
    },
    {
        "Id": "CloudWatchLogs",
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Parameters": {
            "AccessKey": "access-key-id",
            "SecretKey": "secret-access-key",
            "Region": "region",
            "LogGroup": "group",
            "LogStream": "stream"
        }
    },
    {
        "Id": "CloudWatch",
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatch.CloudWatchOutputComponent,AWS.EC2.Windows.CloudWatch",
        "Parameters": {
            "AccessKey": "access-key-id",
            "SecretKey": "secret-access-key",
            "Region": "region",
            "NameSpace": "namespace"
        }
    }
],
"Flows": {
    "Flows": [
        "source,destination",
        "(source1, source2),destination",
        "source, (destination1,destination2)"
    ]
}
}
```

Configure the Region and Namespace for CloudWatch and CloudWatch Logs

Next, you'll define the credentials, region, and metric namespace that comprise the destination where your data is sent.

To set the credentials, region, and metric namespace for CloudWatch

This section of the JSON file defines the credentials, region, and metric namespace that comprise the destination where your data is sent. You can add additional sections with unique IDs (for example, "CloudWatch2", "CloudWatch3", etc.) and specify a different region for each new ID to send the same data to different locations.

Note

You only need to set CloudWatch credentials if you are using EC2Config and plan to send performance counters to CloudWatch. If you're using Amazon EC2 Simple Systems Manager, your credentials are configured in the IAM role you used when you launched your Amazon EC2 instance.

1. In the JSON file, locate the **CloudWatch** section.

```
{
  "Id": "CloudWatch",
  "FullName":
  "AWS.EC2.Windows.CloudWatch.CloudWatch.CloudWatchOutputComponent,AWS.EC2.Windows.CloudW
  "Parameters": {
    "AccessKey": "",
    "SecretKey": "",
    "Region": "us-west-1",
    "NameSpace": "Windows/Default"
  }
},
```

2. In the **AccessKey** parameter, enter your access key ID. This is not supported if you launched your instance using an IAM role. For more information, see [IAM Roles for Amazon EC2 \(p. 658\)](#).
3. In the **SecretKey** parameter, enter your secret access key. This is not supported if you launched your instance using an IAM role. For more information, see [IAM Roles for Amazon EC2 \(p. 658\)](#).
4. In the **Region** parameter, enter the region where you want to send log data. You can specify us-east-1, us-west-1, us-west-2, eu-west-1, eu-central-1, ap-southeast-1, ap-southeast-2, or ap-northeast-1. Although you can send performance counters to a different region from where you send your log data, we recommend that you set this parameter to the same region where your instance is running.
5. In the **NameSpace** parameter, enter the metric namespace where you want performance counter data to be written in CloudWatch.

To set the credentials, region, log group, and log stream for CloudWatch Logs

This section of the JSON file defines the credentials, region, log group name and log stream namespace that comprise the destination where your data is sent. You can add additional sections with unique IDs (for example, "CloudWatchLogs2", "CloudWatchLogs3", etc.) and specify a different region for each new ID to send the same data to different locations.

1. In the JSON file, locate the **CloudWatchLogs** section.

```
{
  "Id": "CloudWatchLogs",
  "FullName":
  "AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
  "Parameters": {
    "AccessKey": "",
    "SecretKey": "",
    "Region": "us-east-1",
    "LogGroup": "Default-Log-Group",
    "LogStream": "{instance_id}"
  }
},
```

2. In the **AccessKey** parameter, enter your access key ID. This is not supported if you launched your instance using an IAM role. For more information, see [IAM Roles for Amazon EC2 \(p. 658\)](#).
3. In the **SecretKey** parameter, enter your secret access key. This is not supported if you launched your instance using an IAM role. For more information, see [IAM Roles for Amazon EC2 \(p. 658\)](#).
4. In the **Region** parameter, enter the region where you want EC2Config to send log data. You can specify us-east-1, us-west-1, us-west-2, eu-west-1, eu-central-1, ap-southeast-1, ap-southeast-2, or ap-northeast-1.
5. In the **LogGroup** parameter, enter the name for your log group. This is the same name that will be displayed on the **Log Groups** screen in the CloudWatch console.

6. In the **LogStream** parameter, enter the destination log stream. If you use **{instance_id}**, the default, EC2Config uses the instance ID of this instance as the log stream name.

If you enter a log stream name that doesn't already exist, CloudWatch Logs automatically creates it for you. You can use a literal string or predefined variables (**{instance_id}**, **{hostname}**, **{ip_address}**), or a combination of all three to define a log stream name.

The log stream name specified in this parameter appears on the **Log Groups > Streams for <YourLogStream>** screen in the CloudWatch console.

Configure the Performance Counters and Logs to Send to CloudWatch and CloudWatch Logs

Next, you'll configure the performance counters and logs that you want to send to CloudWatch and CloudWatch Logs.

To configure the performance counters to send to CloudWatch

You can select any performance counters that are available in Performance Monitor. You can select different categories to upload to CloudWatch as metrics, such as .NET CLR Data, ASP.NET Applications, HTTP Service, Memory, or Process and Processors.

For each performance counter that you want to upload to CloudWatch, copy the **PerformanceCounter** section and change the **Id** parameter to make it unique (e.g., "PerformanceCounter2") and update the other parameters as necessary.

1. In the JSON file, locate the **PerformanceCounter** section.

```
{
  "Id": "PerformanceCounter",
  "FullName":
  "AWS.EC2.Windows.CloudWatch.PerformanceCounterComponent.PerformanceCounterInputComponent",
  "Parameters": {
    "CategoryName": "Memory",
    "CounterName": "Available MBytes",
    "InstanceName": "",
    "MetricName": "AvailableMemory",
    "Unit": "Megabytes",
    "DimensionName": "",
    "DimensionValue": ""
  }
},
```

2. In the **CategoryName** parameter, enter the performance counter category.
 - a. To find the available categories and counters, open Performance Monitor.
 - b. Click **Monitoring Tools**, and then click **Performance Monitor**.
 - c. In the results pane, click the green + (plus) button.

The categories and counters are listed in the **Add Counters** dialog box.

3. In the **CounterName** parameter, enter the name of the performance counter.
4. In the **InstanceName** parameter, enter values from the **Add Counters** dialog box in Performance Monitor, which can be one of the following:
 - Blank, if the selected object has no instances.
 - A single instance of the selected object.
 - **_Total** to use the aggregate of all instances.

Note

Do not use an asterisk (*) to indicate all instances because each performance counter component only supports one metric.

5. In the **MetricName** parameter, enter the CloudWatch metric that you want performance data to appear under.
6. In the **Unit** parameter, enter the appropriate unit of measure for the metric:

Seconds | Microseconds | Milliseconds | Bytes | Kilobytes | Megabytes | Gigabytes | Terabytes | Bits | Kilobits | Megabits | Gigabits | Terabits | Percent | Count | Bytes/Second | Kilobytes/Second | Megabytes/Second | Gigabytes/Second | Terabytes/Second | Bits/Second | Kilobits/Second | Megabits/Second | Gigabits/Second | Terabits/Second | Count/Second | None.

7. (optional) You can enter a dimension name and value in the **DimensionName** and **DimensionValue** parameters to specify a dimension for your metric. These parameters provide another view when listing metrics. You can also use the same dimension for multiple metrics so that you can view all metrics belonging to a specific dimension.

To send Windows application event log data to CloudWatch Logs

1. In the JSON file, locate the **ApplicationEventLog** section.

```
{
  "Id": "ApplicationEventLog",
  "FullName":
  "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
  "Parameters": {
    "LogName": "Application",
    "Levels": "1"
  }
},
```

2. In the **Levels** parameter, enter one of the following values:

- 1 - Only error messages uploaded.
- 2 - Only warning messages uploaded.
- 4 - Only information messages uploaded.

You can add values together to include more than one type of message. For example, **3** means that error messages (**1**) and warning messages (**2**) get uploaded. A value of **7** means that error messages (**1**), warning messages (**2**), and information messages (**4**) get uploaded.

To send security log data to CloudWatch Logs

1. In the JSON file, locate the **SecurityEventLog** section.

```
{
  "Id": "SecurityEventLog",
  "FullName":
  "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
  "Parameters": {
    "LogName": "Security",
    "Levels": "7"
  }
}
```

```
},
```

2. In the **Levels** parameter, enter **7**, so that all messages are uploaded.

To send system event log data to CloudWatch Logs

1. In the JSON file, locate the **SystemEventLog** section.

```
{
  "Id": "SystemEventLog",
  "FullName":
  "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch"
  "Parameters": {
    "LogName": "System",
    "Levels": "7"
  }
},
```

2. In the **Levels** parameter, enter one of the following values:

- 1 - Only error messages uploaded.
- 2 - Only warning messages uploaded.
- 4 - Only information messages uploaded.

You can add values together to include more than one type of message. For example, **3** means that error messages (**1**) and warning messages (**2**) get uploaded. A value of **7** means that error messages (**1**), warning messages (**2**), and information messages (**4**) get uploaded.

To send other types of event log data to CloudWatch Logs

In addition to the application, system, and security logs, you can upload other types of event logs.

1. In the JSON file, add a new section.

```
{
  "Id": "",
  "FullName":
  "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch"
  "Parameters": {
    "LogName": "",
    "Levels": "7"
  }
},
```

2. In the **Id** parameter, enter a name for the log you want to upload (e.g., WindowsBackup).
3. In the **LogName** parameter, enter the name of the log you want to upload.
 - a. To find the name of the log, in Event Viewer, in the navigation pane, click **Applications and Services Logs**.
 - b. In the list of logs, right-click the log you want to upload (e.g., Microsoft>Windows>Backup>Operational), and then click **Create Custom View**.
 - c. In the **Create Custom View** dialog box, click the **XML** tab. The **LogName** is in the <Select Path=> tag (e.g., Microsoft-Windows-Backup). Copy this text into the **LogName** parameter in the **AWS.EC2.Windows.CloudWatch.json** file.
4. In the **Levels** parameter, enter one of the following values:

- 1 - Only error messages uploaded.
- 2 - Only warning messages uploaded.
- 4 - Only information messages uploaded.

You can add values together to include more than one type of message. For example, **3** means that error messages (**1**) and warning messages (**2**) get uploaded. A value of **7** means that error messages (**1**), warning messages (**2**), and information messages (**4**) get uploaded.

To send Event Tracing (Windows) data to CloudWatch Logs

ETW (Event Tracing for Windows) provides an efficient and detailed logging mechanism that applications can write logs to. Each ETW is controlled by a session manager that can start and stop the logging session. Each session has a provider and one or more consumers.

1. In the JSON file, locate the **ETW** section.

```
{
  "Id": "ETW",
  "FullName":
  "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
  "Parameters": {
    "LogName": "Microsoft-Windows-WinINet/Analytic",
    "Levels": "7"
  }
},
```

2. In the **LogName** parameter, enter the name of the log you want to upload.
 - a. To find the name of the log, in Event Viewer, on the **View** menu, click **Show Analytic and Debug Logs**.
 - b. In the navigation pane, click **Applications and Services Logs**.
 - c. In the list of ETW logs, right-click the log you want to upload, and then click **Enable Log**.
 - d. Right-click the log again, and click **Create Custom View**.
 - e. In the **Create Custom View** dialog box, click the **XML** tab. The **LogName** is in the <Select Path=> tag (e.g., `Microsoft-Windows-WinINet/Analytic`). Copy this text into the **LogName** parameter in the **AWS.EC2.Windows.CloudWatch.json** file.
3. In the **Levels** parameter, enter one of the following values:

- 1 - Only error messages uploaded.
- 2 - Only warning messages uploaded.
- 4 - Only information messages uploaded.

You can add values together to include more than one type of message. For example, **3** means that error messages (**1**) and warning messages (**2**) get uploaded. A value of **7** means that error messages (**1**), warning messages (**2**), and information messages (**4**) get uploaded.

To send custom logs (any text-based log file) to CloudWatch Logs

1. In the JSON file, locate the **CustomLogs** section.

```
{
  "Id": "CustomLogs",
```

```
"FullName" :  
"AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent ,AWS.EC2.Windows.CloudWat  
"Parameters" : {  
    "LogDirectoryPath" : "C:\\\\CustomLogs\\\\",  
    "TimestampFormat" : "MM/dd/yyyy HH:mm:ss",  
    "Encoding" : "UTF-8",  
    "Filter" : "",  
    "CultureName" : "en-US",  
    "TimeZoneKind" : "Local",  
    "LineCount" : "5"  
},
```

2. In the **LogDirectoryPath** parameter, enter the path where logs are stored on your instance.
3. In the **TimestampFormat** parameter, enter the timestamp format you want to use. For a list of supported values, see the [Custom Date and Time Format Strings](#) topic on MSDN.

Important

Your source log file must have the timestamp at the beginning of each log line and there must be a space following the timestamp.

4. In the **Encoding** parameter, enter the file encoding to use (e.g., UTF-8). For a list of supported values, see the [Encoding Class](#) topic on MSDN.

Note

Use the encoding name, not the display name, as the value for this parameter.

5. (optional) In the **Filter** parameter, enter the prefix of log names. Leave this parameter blank to monitor all files. For a list of supported values, see the [FileSystemWatcherFilter Property](#) topic on MSDN.
6. (optional) In the **CultureName** parameter, enter the locale where the timestamp is logged. If **CultureName** is blank, it defaults to the same locale currently used by your Windows instance. For a list of supported values, see the [National Language Support \(NLS\) API Reference](#) topic on MSDN.

Note

The **div**, **div-MV**, **hu**, and **hu-HU** values are not supported.

7. (optional) In the **TimeZoneKind** parameter, enter **Local** or **UTC**. You can set this to provide time zone information when no time zone information is included in your log's timestamp. If this parameter is left blank and if your timestamp doesn't include time zone information, CloudWatch Logs defaults to the local time zone. This parameter is ignored if your timestamp already contains time zone information.
8. (optional) In the **LineCount** parameter, enter the number of lines in the header to identify the log file. For example, IIS log files have virtually identical headers. You could enter **5**, which would read the first three lines of the log file's header to identify it. In IIS log files, the third line is the date and time stamp, but the time stamp is not always guaranteed to be different between log files. For this reason, we recommend including at least one line of actual log data for uniquely fingerprinting the log file.

To send IIS log data to CloudWatch Logs

1. In the JSON file, locate the **IISLog** section.

```
{  
    "Id" : "IISLogs",  
    "FullName" :  
"AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent ,AWS.EC2.Windows.CloudWat  
    "Parameters" : {  
        "LogDirectoryPath" : "C:\\\\inetpub\\\\logs\\\\LogFiles\\\\W3SVC1",
```

```
    "TimestampFormat": "yyyy-MM-dd HH:mm:ss",  
    "Encoding": "UTF-8",  
    "Filter": "",  
    "CultureName": "en-US",  
    "TimeZoneKind": "UTC",  
    "LineCount": "5"  
  },  
}
```

2. In the **LogDirectoryPath** parameter, enter the folder where IIS logs are stored for an individual site (e.g., **C:\inetpub\logs\LogFiles\W3SVC*n***).

Note

Only W3C log format is supported. IIS, NCSA, and Custom formats are not supported.

3. In the **TimestampFormat** parameter, enter the timestamp format you want to use. For a list of supported values, see the [Custom Date and Time Format Strings](#) topic on MSDN.
4. In the **Encoding** parameter, enter the file encoding to use (e.g., UTF-8). For a list of supported values, see the [Encoding Class](#) topic on MSDN.

Note

Use the encoding name, not the display name, as the value for this parameter.

5. (optional) In the **Filter** parameter, enter the prefix of log names. Leave this parameter blank to monitor all files. For a list of supported values, see the [FileSystemWatcherFilter Property](#) topic on MSDN.
6. (optional) In the **CultureName** parameter, enter the locale where the timestamp is logged. If **CultureName** is blank, it defaults to the same locale currently used by your Windows instance. For a list of supported values, see the [National Language Support \(NLS\) API Reference](#) topic on MSDN.

Note

The **div**, **div-MV**, **hu**, and **hu-HU** values are not supported.

7. (optional) In the **TimeZoneKind** parameter, enter **Local** or **UTC**. You can set this to provide time zone information when no time zone information is included in your log's timestamp. If this parameter is left blank and if your timestamp doesn't include time zone information, CloudWatch Logs defaults to the local time zone. This parameter is ignored if your timestamp already contains time zone information.
8. (optional) In the **LineCount** parameter, enter the number of lines in the header to identify the log file. For example, IIS log files have virtually identical headers. You could enter **5**, which would read the first five lines of the log file's header to identify it. In IIS log files, the third line is the date and time stamp, but the time stamp is not always guaranteed to be different between log files. For this reason, we recommend including at least one line of actual log data for uniquely fingerprinting the log file.

Configure the Flow Control

In order to send performance counter data to CloudWatch or to send log data to CloudWatch Logs, each data type must have a corresponding destination listed in the **Flows** section. For example, to send a performance counter defined in the **"Id": "PerformanceCounter"** section of the JSON file to the CloudWatch destination defined in the **"Id": "CloudWatch"** section of the JSON file, you would enter **"PerformanceCounter,CloudWatch"** in the **Flows** section. Similarly, to send the custom log, ETW log, and system log to CloudWatch Logs, you would enter **"(CustomLogs, ETW, SystemEventLog),CloudWatchLogs"**. In addition, you can send the same performance counter or log file to more than one destination. For example, to send the application log to two different destinations that you defined in the **"Id": "CloudWatchLogs"** section of the JSON file, you would enter **"ApplicationEventLog,(CloudWatchLogs, CloudWatchLogs2)"** in the **Flows** section.

1. In the JSON file, locate the **Flows** section.

```
"Flows": {
  "Flows": [
    "PerformanceCounter,CloudWatch",
    "(PerformanceCounter,PerformanceCounter2), CloudWatch2",
    "(CustomLogs, ETW, SystemEventLog),CloudWatchLogs",
    "CustomLogs, CloudWatchLogs2",
    "ApplicationEventLog,(CloudWatchLogs, CloudWatchLogs2)"
  ]
}
```

2. In the **Flows** parameter, enter each data type that you want to upload (e.g., ApplicationEventLog) and destination where you want to send it (e.g., CloudWatchLogs).

Upload Logs to Amazon CloudWatch Using Run Command

Note

This procedure does not include information about how to configure Run Command for Amazon SNS notifications. To learn more about the Amazon SNS notification fields on the **Command History** page and how to execute commands that return notifications, see [Getting Amazon SNS Notifications When a Command Changes Status \(p. 505\)](#).

To upload logs to Amazon CloudWatch using Run Command

1. Open the [Amazon EC2 console](#) and choose **Command History** in the navigation pane.
2. Choose **Run a Command**.
3. In the **Command document** list, choose **AWS-ConfigureCloudWatch**.
4. Choose **Select target instances** to select the instances where you want the command to run. If you do not see a complete list of instances, the missing instances might not be configured properly for Run Command. For more information, see [Systems Manager Prerequisites \(p. 394\)](#).
5. (Optional) In the **Status** list choose **Enable** to configure instances to upload logs to CloudWatch. Choose **Disabled** to configure instances to stop sending logs to CloudWatch.
6. Copy and paste your JSON example into the **Properties** field.
7. (Optional) In the **Comment** field, type information you want to provide about this command. Comments are stored in the log file and appear in the Command Invocation List in the Amazon EC2 Console.

Tip

We recommend that you enter specific comments about each command you run. The list of commands that you send can grow quickly, and the **Comment** field can help you identify commands that you want to monitor.

8. In the **Timeout (seconds)** field, type the number of seconds Run Command should attempt to reach instances before an instance is considered unreachable and the command execution fails. The minimum is 30 seconds. The maximum is 30 days. The default value is 10 minutes.
9. In the **S3 bucket** field, type the name of an AWS S3 bucket where you want to store the output of the command.

Important

The Run Command **Output** page in the Amazon EC2 console truncates output after 2500 characters. Configure an Amazon S3 bucket before executing commands using Run Command. If your command output was longer than 2500 characters, you can view the full output in your Amazon S3 bucket. For more information, see [Create a Bucket](#).

10. In the **S3 key prefix**, field, type the name of a subfolder in the Amazon S3 bucket. This subfolder can help you organize Run Command output.

Note

The section called **AWS Command Line Interface command** displays a usable CLI script that is generated based on the parameters you entered.

After you execute a command using Run Command, the system returns you to the commands list.

Important

The Amazon EC2 console truncates all command output beyond 2500 characters. If your command output was longer than 2500 characters, you can view the full output in your S3 bucket.

Canceling a Command

You can attempt to cancel a command as long as the service shows that it is in either a Pending or Executing state. However, even if a command is still in one of these states, we cannot guarantee that the command will be terminated and the underlying process stopped.

To cancel a command using the console

1. Open the [Amazon EC2 console](#) and choose **Command History** in the navigation pane.
2. Select the command invocation that you want to cancel.
3. Choose **Actions** and then choose **Cancel command**.

To cancel a command using the AWS CLI

Use the following command.

```
aws ssm cancel-command --command-id "command ID" --instance-ids "instance ID"
```

For information about the status of a cancelled command, see [Command Status and Monitoring \(p. 501\)](#).

View Command Output

Use the following procedure to view the results of command execution in the EC2 console.

To view command output

1. In the Amazon EC2 console, select a command in the list.
2. Choose the **Output** tab.
3. Choose **View Output**.

Run a command Actions

Filter by attributes

Command ID	Instance ID	Document name	Status	Requested date	Comment
ca4b10c6-cee1-437...	i-d583f76a	AWS-RunPowerSh...	Success	October 20, 2015 at...	getting list of proce
561e5f4a-27d2-419...	i-d583f76a	AWS-RunPowerSh...	Success	October 20, 2015 at...	ipconfig on the box
d5b589e6-ad94-4d...	i-d583f76a	AWS-RunPowerSh...	Success	October 20, 2015 at...	Getting services on

Command ID: ca4b10c6-cee1-437b-9f70-9746da1477e7 Instance ID: i-d583f76a

Description Output

Output

Plugin name	Status	Response code	Start Time	Finish Time	Output
aws:runPower...	Success	0	October 20, 2015 at 4:15:58 PM...	October 20, 2015 at 4:15:59 PM...	View Output

4. The command output page shows the results of your command execution.

Commands > Output

Output for aws:runPowerShellScript

Handles	NPM (K)	PM (K)	WS (K)	VM (M)	CPU (s)	Id	ProcessName
40	4	612	2752	27	0.00	2900	conhost
194	11	1712	3776	46	0.06	520	csrss
86	8	1236	3576	43	0.08	584	csrss
209	11	1700	21036	66	2.06	880	csrss
173	14	10764	1608	88	0.03	872	dwm
208	25	23216	45228	164	0.58	2560	dwm
836	63	47216	21880	643	24.92	2276	Ec2Config
1344	64	52420	108976	495	11.92	2776	explorer
0	0	0	4	0		0	Idle
105	8	1196	4844	33	0.02	1176	LiteAgent
268	21	12692	24108	138	0.08	2148	LogonUI
797	18	4336	10892	40	0.72	688	lsass
159	12	2108	3760	41	0.02	2888	msdtc
362	23	56848	53412	593	0.30	3688	powershell
219	11	1976	9600	87	0.28	732	rdpclip
650	48	108564	105064	773	4.58	3472	ServerManager
209	10	2420	7148	25	2.27	680	services
55	2	276	1048	4	0.03	436	smss
424	23	4536	12124	91	1.94	1148	spoolsv

-----Output truncated-----

For information about how to run commands using Windows PowerShell, see [Amazon EC2 Run Command Walkthrough Using the AWS Tools for Windows PowerShell \(p. 494\)](#) or the [AWS Tools for Windows PowerShell Reference](#). For information about how to run commands using the AWS CLI, see the [SSM CLI Reference](#).

To learn more about Amazon CloudWatch, see [What is Amazon CloudWatch?](#)

Enabling or Disabling Windows Updates Using Amazon EC2 Run Command

You can use the `AWS-ConfigureWindowsUpdate` document to enable or disable automatic Windows updates on your instances. This command configures the Windows update agent to download and install Windows updates on the day and hour that you specify. If an update requires a reboot, the computer reboots automatically 15 minutes after updates have been installed. With this command you can also configure Windows update to check for updates but not install them. The `AWS-ConfigureWindowsUpdate` document is compatible with Windows Server 2008, 2008 R2, 2012, and 2012 R2.

Note

This procedure does not include information about how to configure Run Command for Amazon SNS notifications. To learn more about the Amazon SNS notification fields on the **Command History** page and how to execute commands that return notifications, see [Getting Amazon SNS Notifications When a Command Changes Status \(p. 505\)](#).

To enable or disable Windows Updates using Run Command

1. Open the [Amazon EC2 console](#) and choose **Command History** in the navigation pane.
2. Choose **Run a Command**.
3. In the **Command document** list, choose **AWS-ConfigureWindowsUpdate**.
4. Choose **Select target instances** to select the instances where you want the command to run. If you do not see a complete list of instances, the missing instances might not be configured properly for Run Command. For more information, see [Systems Manager Prerequisites \(p. 394\)](#).
5. In the **Update Level** list, choose **InstallUpdatesAutomatically** to have Windows automatically download and install updates. If an update requires a reboot, the computer is automatically rebooted 15 minutes after updates have been installed. Or choose **NeverCheckForUpdates**. If you choose this option Windows never checks for or downloads updates.

Important

If you choose **NeverCheckForUpdates** be aware that your system could become vulnerable to malicious attacks if you do not manually install important updates, such as security updates.

6. In the **Scheduled Install Day** field, choose the day of the week when you want Windows to download and install updates. This applies only if you selected the **InstallUpdatesAutomatically** option.
7. In the **Scheduled Install Time** field, choose the time of day when you want Windows to download and install updates. This applies only if you selected the **InstallUpdatesAutomatically** option.

Note

Scheduled Install Time is the time where the instance is located. For example, if the instance is located in the N. Virginia region, the **Scheduled Install Time** would be Eastern time.

8. (Optional) In the **Comment** field, type information you want to provide about this command. Comments are stored in the log file and appear in the Command Invocation List in the Amazon EC2 console.

Tip

We recommend that you enter specific comments about each command you run. The list of commands that you send can grow quickly, and the **Comment** field can help you identify commands that you want to monitor.

9. (Optional) In the **Execution Timeout** field, type the number of seconds the EC2Config service will attempt to run the command before it times out and fails.
10. (Optional) In the **Day** list, choose the day of the week when you want to have the system download and install updates.
11. In the **Timeout (seconds)** field, type the number of seconds Run Command should attempt to reach instances before an instance is considered unreachable and the command execution fails. The minimum is 30 seconds. The maximum is 30 days. The default value is 10 minutes.

12. In the **S3 bucket** field, type the name of an Amazon S3 bucket where you want to store the output of the command.

Important

The Run Command **Output** page in the Amazon EC2 console truncates output after 2500 characters. Configure an Amazon S3 bucket before executing commands using Run Command. If your command output was longer than 2500 characters, you can view the full output in your Amazon S3 bucket. For more information, see [Create a Bucket](#).

13. In the **S3 key prefix**, field, type the name of a subfolder in the Amazon S3 bucket. This subfolder can help you organize Run Command output.

Note

The section called **AWS Command Line Interface command** displays a usable CLI script that is generated based on the parameters you entered.

After you execute a command using Run Command, the system returns you to the commands list.

Important

The Amazon EC2 console truncates all command output beyond 2500 characters. If your command output was longer than 2500 characters, you can view the full output in your S3 bucket.

Canceling a Command

You can attempt to cancel a command as long as the service shows that it is in either a Pending or Executing state. However, even if a command is still in one of these states, we cannot guarantee that the command will be terminated and the underlying process stopped.

To cancel a command using the console

1. Open the [Amazon EC2 console](#) and choose **Command History** in the navigation pane.
2. Select the command invocation that you want to cancel.
3. Choose **Actions** and then choose **Cancel command**.

To cancel a command using the AWS CLI

Use the following command.

```
aws ssm cancel-command --command-id "command ID" --instance-ids "instance ID"
```

For information about the status of a cancelled command, see [Command Status and Monitoring \(p. 501\)](#).

View Command Output

Use the following procedure to view the results of command execution in the EC2 console.

To view command output

1. In the Amazon EC2 console, select a command in the list.
2. Choose the **Output** tab.
3. Choose **View Output**.

Run a command Actions

Filter by attributes

Command ID	Instance ID	Document name	Status	Requested date	Comment
ca4b10c6-cee1-437...	i-d583f76a	AWS-RunPowerSh...	Success	October 20, 2015 at...	getting list of proce
561e5f4a-27d2-419...	i-d583f76a	AWS-RunPowerSh...	Success	October 20, 2015 at...	ipconfig on the box
d5b589e6-ad94-4d...	i-d583f76a	AWS-RunPowerSh...	Success	October 20, 2015 at...	Getting services on

Command ID: ca4b10c6-cee1-437b-9f70-9746da1477e7 Instance ID: i-d583f76a

Description Output

Output

Plugin name	Status	Response code	Start Time	Finish Time	Output
aws:runPower...	Success	0	October 20, 2015 at 4:15:58 PM...	October 20, 2015 at 4:15:59 PM...	View Output

- The command output page shows the results of your command execution.

Commands > Output

Output for aws:runPowerShellScript

Handles	NPM (K)	PM (K)	WS (K)	VM (M)	CPU (s)	Id	ProcessName
40	4	612	2752	27	0.00	2900	conhost
194	11	1712	3776	46	0.06	520	csrss
86	8	1236	3576	43	0.08	584	csrss
209	11	1700	21036	66	2.06	880	csrss
173	14	10764	1608	88	0.03	872	dwm
208	25	23216	45228	164	0.58	2560	dwm
836	63	47216	21880	643	24.92	2276	Ec2Config
1344	64	52420	108976	495	11.92	2776	explorer
0	0	0	4	0		0	Idle
105	8	1196	4844	33	0.02	1176	LiteAgent
268	21	12692	24108	138	0.08	2148	LogonUI
797	18	4336	10892	40	0.72	688	lsass
159	12	2108	3760	41	0.02	2888	msdtc
362	23	56848	53412	593	0.30	3688	powershell
219	11	1976	9600	87	0.28	732	rdpclip
650	48	108564	105064	773	4.58	3472	ServerManager
209	10	2420	7148	25	2.27	680	services
55	2	276	1048	4	0.03	436	smss
424	23	4536	12124	91	1.94	1148	spoolsv

-----Output truncated-----

For information about how to run commands using Windows PowerShell, see [Amazon EC2 Run Command Walkthrough Using the AWS Tools for Windows PowerShell \(p. 494\)](#) or the [AWS Tools for Windows PowerShell Reference](#). For information about how to run commands using the AWS CLI, see the [SSM CLI Reference](#).

Updating the SSM Agent Using Amazon EC2 Run Command

You can use the [AWS-UpdateEC2Config](#) document to update the SSM Agent running on instances. Depending on the operating system used by your instance and when it was created, the SSM Agent is either the EC2Config service or new service called SSM Agent. Running the process described here updates both services. You can update to either the latest version or downgrade to an older version. When you execute the command, the system downloads the version from AWS, installs it, and then uninstalls the version that existed before the command was run. If an error occurs during this process, the system rolls back to the version on the server before the command was run and the command status shows that the command failed.

Note

Updating the EC2Config service using Run Command is only supported if the instances is running EC2Config service version 3.10.442 or higher.

This procedure does not include information about how to configure Run Command for Amazon SNS notifications. To learn more about the Amazon SNS notification fields on the **Command History** page and how to execute commands that return notifications, see [Getting Amazon SNS Notifications When a Command Changes Status](#) (p. 505).

To update the SSM Agent using Run Command

1. Open the [Amazon EC2 console](#) and choose **Command History** in the navigation pane.
2. Choose **Run a Command**.
3. In the **Command document** list, choose **AWS-UpdateEC2Config**.
4. Choose **Select target instances** to select the instances where you want the command to run. If you do not see a complete list of instances, the missing instances might not be configured properly for Run Command. For more information, see [Systems Manager Prerequisites](#) (p. 394).
5. (Optional) In the **Version** field, enter a specific version of the EC2Config service to install. You can install older versions of the service. If you do not specify a version, the service will be updated to the latest version.
6. (Optional) In the **Allow downgrade** list, choose **True** if you want to install an earlier version of the EC2Config service. If you choose this option, you must specify the earlier version number. Choose **False** if you want the system to install only the newest version of the service.
7. (Optional) In the **Comment** field, enter information you want to provide about this command. Comments are stored in the log file and appear in the Command Invocation List in the Amazon EC2 console.

Tip

We recommend that you enter specific comments about each command you run. The list of commands that you send can grow quickly, and the **Comment** field can help you identify commands that you want to monitor.

8. In the **Timeout (seconds)** field, enter the number of seconds Run Command should attempt to reach instances before an instance is considered unreachable and the command execution fails. The minimum is 30 seconds. The maximum is 30 days. The default value is 10 minutes.
9. In the **S3 bucket** field enter the name of an Amazon S3 bucket where you want to store the output of the command.

Important

The Run Command **Output** page in the Amazon EC2 console truncates output after 2500 characters. Configure an Amazon S3 bucket before executing commands using Run Command. If your command output was longer than 2500 characters, you can view the full output in your Amazon S3 bucket. For more information, see [Create a Bucket](#).

10. In the **S3 key prefix**, field enter the name of a subfolder in the Amazon S3 bucket. This subfolder can help you organize Run Command output.

Note

The section called **AWS Command Line Interface command** displays a usable CLI script that is generated based on the parameters you entered.

After you execute a command using Run Command, the system returns you to the commands list.

Important

The Amazon EC2 console truncates all command output beyond 2500 characters. If your command output was longer than 2500 characters, you can view the full output in your S3 bucket.

Canceling a Command

You can attempt to cancel a command as long as the service shows that it is in either a Pending or Executing state. However, even if a command is still in one of these states, we cannot guarantee that the command will be terminated and the underlying process stopped.

To cancel a command using the console

1. Open the [Amazon EC2 console](#) and choose **Command History** in the navigation pane.
2. Select the command invocation that you want to cancel.
3. Choose **Actions** and then choose **Cancel command**.

To cancel a command using the AWS CLI

Use the following command.

```
aws ssm cancel-command --command-id "command ID" --instance-ids "instance ID"
```

For information about the status of a cancelled command, see [Command Status and Monitoring \(p. 501\)](#).

View Command Output

Use the following procedure to view the results of command execution in the EC2 console.

To view command output

1. In the Amazon EC2 console, select a command in the list.
2. Choose the **Output** tab.
3. Choose **View Output**.

Run a command Actions

Filter by attributes

Command ID	Instance ID	Document name	Status	Requested date	Comment
ca4b10c6-cee1-437...	i-d583f76a	AWS-RunPowerSh...	Success	October 20, 2015 at...	getting list of proce
561e5f4a-27d2-419...	i-d583f76a	AWS-RunPowerSh...	Success	October 20, 2015 at...	ipconfig on the box
d5b589e6-ad94-4d...	i-d583f76a	AWS-RunPowerSh...	Success	October 20, 2015 at...	Getting services on

Command ID: ca4b10c6-cee1-437b-9f70-9746da1477e7 Instance ID: i-d583f76a

Description Output

Output

Plugin name	Status	Response code	Start Time	Finish Time	Output
aws:runPower...	Success	0	October 20, 2015 at 4:15:58 PM...	October 20, 2015 at 4:15:59 PM...	View Output

- The command output page shows the results of your command execution.

Commands > Output

Output for aws:runPowerShellScript

Handles	NPM (K)	PM (K)	WS (K)	VM (M)	CPU (s)	Id	ProcessName
40	4	612	2752	27	0.00	2900	conhost
194	11	1712	3776	46	0.06	520	csrss
86	8	1236	3576	43	0.08	584	csrss
209	11	1700	21036	66	2.06	880	csrss
173	14	10764	1608	88	0.03	872	dwm
208	25	23216	45228	164	0.58	2560	dwm
836	63	47216	21880	643	24.92	2276	Ec2Config
1344	64	52420	108976	495	11.92	2776	explorer
0	0	0	4	0		0	Idle
105	8	1196	4844	33	0.02	1176	LiteAgent
268	21	12692	24108	138	0.08	2148	LogonUI
797	18	4336	10892	40	0.72	688	lsass
159	12	2108	3760	41	0.02	2888	msdtc
362	23	56848	53412	593	0.30	3688	powershell
219	11	1976	9600	87	0.28	732	rdpclip
650	48	108564	105064	773	4.58	3472	ServerManager
209	10	2420	7148	25	2.27	680	services
55	2	276	1048	4	0.03	436	smss
424	23	4536	12124	91	1.94	1148	spoolsv

-----Output truncated-----

For information about how to run commands using Windows PowerShell, see [Amazon EC2 Run Command Walkthrough Using the AWS Tools for Windows PowerShell \(p. 494\)](#) or the [AWS Tools for Windows PowerShell Reference](#). For information about how to run commands using the AWS CLI, see the [SSM CLI Reference](#).

Inventory an Amazon EC2 Instance for Windows Using Amazon EC2 Run Command

You can use the `AWS-ListWindowsInventory` document to collect information about an Amazon EC2 instance running in Windows. The command returns the following information:

- Operating system version, language, and details
- Installed applications
- Installed system updates

Note

This procedure does not include information about how to configure Run Command for Amazon SNS notifications. To learn more about the Amazon SNS notification fields on the **Command History** page and how to execute commands that return notifications, see [Getting Amazon SNS Notifications When a Command Changes Status \(p. 505\)](#).

To inventory an EC2 instance using Run Command

1. Open the [Amazon EC2 console](#) and choose **Command History** in the navigation pane.
2. Choose **Run a Command**.
3. In the **Command document** list, choose **AWS-ListWindowsInventory**.
4. Choose **Select target instances** to select the instances where you want the command to run. If you do not see a complete list of instances, the missing instances might not be configured properly for Run Command. For more information, see [Systems Manager Prerequisites \(p. 394\)](#).
5. Choose the list options that you want to execute in your command. For more information about these options, view the tooltip help.
6. (Optional) In the **Comment** field, type information you want to provide about this command. Comments are stored in the log file and appear in the Command Invocation List in the Amazon EC2 console.

Tip

We recommend that you enter specific comments about each command you run. The list of commands that you send can grow quickly, and the **Comment** field can help you identify commands that you want to monitor.

7. In the **Timeout (seconds)** field, type the number of seconds Run Command should attempt to reach instances before an instance is considered unreachable and the command execution fails. The minimum is 30 seconds. The maximum is 30 days. The default value is 10 minutes.
8. In the **S3 bucket** field, type the name of an Amazon S3 bucket where you want to store the output of the command.

Important

The Run Command **Output** page in the Amazon EC2 console truncates output after 2500 characters. Configure an Amazon S3 bucket before executing commands using Run Command. If your command output was longer than 2500 characters, you can view the full output in your Amazon S3 bucket. For more information, see [Create a Bucket](#).

9. In the **S3 key prefix**, field enter the name of a subfolder in the Amazon S3 bucket. This subfolder can help you organize Run Command output.

Note

The section called **AWS Command Line Interface command** displays a usable CLI script that is generated based on the parameters you entered.

After you execute a command using Run Command, the system returns you to the commands list.

Important

The Amazon EC2 console truncates all command output beyond 2500 characters. If your command output was longer than 2500 characters, you can view the full output in your S3 bucket.

Canceling a Command

You can attempt to cancel a command as long as the service shows that it is in either a Pending or Executing state. However, even if a command is still in one of these states, we cannot guarantee that the command will be terminated and the underlying process stopped.

To cancel a command using the console

1. Open the [Amazon EC2 console](#) and choose **Command History** in the navigation pane.
2. Select the command invocation that you want to cancel.
3. Choose **Actions** and then choose **Cancel command**.

To cancel a command using the AWS CLI

Use the following command.

```
aws ssm cancel-command --command-id "command ID" --instance-ids "instance ID"
```

For information about the status of a cancelled command, see [Command Status and Monitoring \(p. 501\)](#).

View Command Output

Use the following procedure to view the results of command execution in the EC2 console.

To view command output

1. In the Amazon EC2 console, select a command in the list.
2. Choose the **Output** tab.
3. Choose **View Output**.

Run a command Actions

Filter by attributes

Command ID	Instance ID	Document name	Status	Requested date	Comment
ca4b10c6-cee1-437...	i-d583f76a	AWS-RunPowerSh...	Success	October 20, 2015 at...	getting list of proce
561e5f4a-27d2-419...	i-d583f76a	AWS-RunPowerSh...	Success	October 20, 2015 at...	ipconfig on the box
d5b589e6-ad94-4d...	i-d583f76a	AWS-RunPowerSh...	Success	October 20, 2015 at...	Getting services on

Command ID: ca4b10c6-cee1-437b-9f70-9746da1477e7 Instance ID: i-d583f76a

Description Output

Output

Plugin name	Status	Response code	Start Time	Finish Time	Output
aws:runPower...	Success	0	October 20, 2015 at 4:15:58 PM...	October 20, 2015 at 4:15:59 PM...	View Output

- The command output page shows the results of your command execution.

Commands > Output

Output for aws:runPowerShellScript

Handles	NPM (K)	PM (K)	WS (K)	VM (M)	CPU (s)	Id	ProcessName
40	4	612	2752	27	0.00	2900	conhost
194	11	1712	3776	46	0.06	520	csrss
86	8	1236	3576	43	0.08	584	csrss
209	11	1700	21036	66	2.06	880	csrss
173	14	10764	1608	88	0.03	872	dwm
208	25	23216	45228	164	0.58	2560	dwm
836	63	47216	21880	643	24.92	2276	Ec2Config
1344	64	52420	108976	495	11.92	2776	explorer
0	0	0	4	0		0	Idle
105	8	1196	4844	33	0.02	1176	LiteAgent
268	21	12692	24108	138	0.08	2148	LogonUI
797	18	4336	10892	40	0.72	688	lsass
159	12	2108	3760	41	0.02	2888	msdtc
362	23	56848	53412	593	0.30	3688	powershell
219	11	1976	9600	87	0.28	732	rdpclip
650	48	108564	105064	773	4.58	3472	ServerManager
209	10	2420	7148	25	2.27	680	services
55	2	276	1048	4	0.03	436	smss
424	23	4536	12124	91	1.94	1148	spoolsv

-----Output truncated-----

For information about how to run commands using Windows PowerShell, see [Amazon EC2 Run Command Walkthrough Using the AWS Tools for Windows PowerShell \(p. 494\)](#) or the [AWS Tools for Windows PowerShell Reference](#). For information about how to run commands using the AWS CLI, see the [SSM CLI Reference](#).

Managing Updates for an EC2 Windows Instance Using Amazon EC2 Run Command

Run Command includes three documents to help you manage updates for EC2 Windows instances.

AWS-FindWindowsUpdates

Scans an instance and determines which updates are missing.

AWS-InstallMissingWindowsUpdates

Installs missing updates on your EC2 instance.

AWS-InstallSpecificWindowsUpdates

Installs one or more specific updates.

Note

This procedure does not include information about how to configure Run Command for Amazon SNS notifications. To learn more about the Amazon SNS notification fields on the **Command History** page and how to execute commands that return notifications, see [Getting Amazon SNS Notifications When a Command Changes Status \(p. 505\)](#).

To manage updates for an EC2 instance using Run Command

1. Open the [Amazon EC2 console](#) and choose **Command History** in the navigation pane.
2. Choose **Run a Command**.
3. In the **Command document** list, choose the document you want to use.
4. Choose **Select target instances** to select the instances where you want the command to run. If you do not see a complete list of instances, the missing instances might not be configured properly for Run Command. For more information, see [Systems Manager Prerequisites \(p. 394\)](#).
5. Choose the update and KB options that you want to execute in your command. For more information about these options, view the tooltip help.
6. (Optional) In the **Comment** field, type information you want to provide about this command. Comments are stored in the log file and appear in the Command Invocation List in the Amazon EC2 console.

Tip

We recommend that you enter specific comments about each command you run. The list of commands that you send can grow quickly, and the **Comment** field can help you identify commands that you want to monitor.

For information about how to run commands using the AWS CLI, see the [SSM CLI Reference](#).

Sending a Command to Multiple Instances

You can send commands to tens, hundreds, or thousands of instances by using the `targets` parameter, which is currently supported when executing commands from the AWS CLI. The `targets` parameter accepts a `Key:Value` combination based on Amazon EC2 tags that you specified for your instances. When you execute the command, the system locates and attempts to run the command on all instances that match the specified criteria. For more information about Amazon EC2 tags, see [Tagging Your Amazon EC2 Resources \(p. 859\)](#).

To control command execution across hundreds or thousands of instances, Run Command also includes parameters for restricting how many instances can simultaneously process a request and how many errors can be thrown by a command before the command is terminated.

Contents

- [Targeting Multiple Instances \(p. 477\)](#)
- [Using Concurrency Controls \(p. 477\)](#)

- [Using Error Controls](#) (p. 477)

Targeting Multiple Instances

The `targets` parameter uses the following syntax:

```
aws ssm send-command --document-name name --targets "Key=tag:tag  
name[:Values=tag values] [...]"
```

Note

Example commands in this section are truncated using [...].

The following examples show you how to specify values for the `targets` parameter:

Example: Targeting tagged instances using `Key;Value` criteria

If you tagged instances for different environments using a `Key` named `Environments` and `Values` of `Development`, `Test`, `Pre-production` and `Production`, then you could send a command to all of the instances in one of these environments by using the `targets` parameter with the following syntax:

```
aws ssm send-command --document-name name --targets  
"Key=tag:Environment;Values=Development" [...]"
```

Example: Targeting instances tagged with a specific `Key` criteria

If you specified `Key` tags for different types of servers, for example `Database`, `WebServer`, and `FileServer`, you could target one type of server using only the `Key` portion of the `Key;Value` criteria.

```
aws ssm send-command --document-name name --targets "Key=tag:Database" [...]"
```

Using Concurrency Controls

You can control how many servers execute the command at the same time by using the `max-concurrency` parameter. You can specify either an absolute number of instances, for example 10, or a percentage of the target set, for example 10%. The queuing system delivers the command to a single instance and waits until the initial invocation completes before sending the command to two more instances. The system exponentially sends commands to more instances until the value of `max-concurrency` is met. The default for value `max-concurrency` is 50. The following examples show you how to specify values for the `max-concurrency` parameter:

```
aws ssm send-command --document-name name --max-concurrency 10 --targets  
"Key=tag:Environment;Values=Development" [...]"
```

```
aws ssm send-command --document-name name --max-concurrency 10%  
--targets "Key=tag:Department;Values=Finance,Marketing"  
"Key=tag:ServerRole;Values=WebServer,Database" [...]"
```

Using Error Controls

You can also control the execution of a command to hundreds or thousands of instances by setting an error limit using the `max-errors` parameters. The parameter specifies how many errors are allowed before the system stops sending the command to additional instances. You can specify either an

absolute number of errors, for example 10, or a percentage of the target set, for example 10%. If you specify 1, then the system stops sending the command to additional instances after the first error result is returned. If you send a command to 50 instances and set `max-errors` to 10%, then the system stops sending the command to additional instances after the fifth error.

Invocations that are already running a command when `max-errors` is reached are allowed to complete, but some of these invocations may fail as well. If you need to ensure that there won't be more than `max-errors` failed invocations, set `max-concurrency` to 1 so the invocations proceed one at a time. The default for `max-concurrency` is 50. The following examples show you how to specify values for the `max-errors` parameter:

```
aws ssm send-command --document-name name --max-errors 10 --targets  
"Key=tag:Database" [...]
```

```
--document-name name --max-errors 10% --targets  
"Key=tag:Environment;Values=Development" [...]
```

```
aws ssm send-command --document-name name --max-concurrency 1 --max-errors 1  
--targets "Key=tag:Environment;Values=Production" [...]
```

Creating SSM Documents

When you execute a command using Amazon EC2 Run Command, the system reads the actions to be performed from a document that defines the plugins to run and parameters to use. This document is called an SSM document. The first time you execute a command from a new SSM document, the system stores the document with your AWS account.

Limitations

As you begin working with SSM documents, be aware of the following limitations.

- You can create a maximum of 200 SSM documents per AWS account.
- SSM documents that you create are only available in the region where you created them. To add a document in another region, copy the content and recreate it in the new region.

Note

If you need to create more than the maximum number of SSM documents, contact AWS Support.

When giving a user access to Run Command the best practice is to start with a policy of least privilege. Create different SSM documents that allow the user to do a minimum number of tasks. For example, you can create SSM documents that enable the user to perform the following types of actions: install a specific application, reset Internet Information Services (IIS), or view a list of running services or processes. For more information, see [Sample SSM Documents \(p. 479\)](#).

Create an SSM Document Using the Amazon EC2 Console

1. Open the [Amazon EC2 console](#) and choose **Command History** in the navigation pane.
2. Choose **Documents** and then choose **Create Document**.
3. Enter a descriptive name for the document and then specify plugins and parameters in the **Content** field in JSON format. For more information, see [SSM Plugins](#) in the *Amazon EC2 Simple Systems Manager API Reference*.
4. Choose **Create Document** to save it with your AWS user account.

Create an SSM Document Using Windows PowerShell

1. Specify plugins and parameters in a file. Save the document with a descriptive name and a `.json` file extension. For more information, see [SSM Plugins](#) in the *Amazon EC2 Simple Systems Manager API Reference*.
2. Create the document and save it with your AWS user account using AWS Tools for Windows PowerShell.

```
$json = Get-Content C:\your file | Out-String  
New-SSMDocument -Name document name -Content $json
```

Create an SSM Document Using the AWS CLI

1. Specify plugins and parameters in a file. Save the document with a descriptive name and a `.json` file extension. For more information, see [SSM Plugins](#) in the *Amazon EC2 Simple Systems Manager API Reference*.
2. Create the document and save it with your AWS user account using the AWS CLI.

```
aws ssm create-document --content file://c:\temp\your file --name  
"document name"
```

Sample SSM Documents

SSM documents are currently supported in JavaScript Object Notation (JSON) and use the following:

- schemaVersion 1.2
- A runtimeConfig that uses one or more plugins to execute tasks. Plugins are platform specific, meaning they run on either a supported version of Windows or Linux. For more information about plugins, see [SSM Plugins](#) in the *Amazon EC2 Simple Systems Manager API Reference*.

Use the following examples as a foundation to create your own documents.

Restrictive SSM Document for Windows

The following example shows a highly-restrictive SSM document that uses the `AWS-RunPowerShellScript` document on Windows. The user can only run the `ipconfig` command to check the IP configuration of the instance:

```
{  
  "schemaVersion": "1.2",  
  "description": "Run ipconfig on the instance.",  
  "parameters": {  
  
  },  
  "runtimeConfig": {  
    "aws:runPowerShellScript": {  
      "properties": [  
        {  
          "id": "0.aws:runPowerShellScript",  
          "runCommand": ["ipconfig"],  
          "workingDirectory": "",  
          "timeoutSeconds": ""  
        }  
      ]  
    }  
  }  
}
```

```
}  
  }  
} }  
}
```

You can use the following JSON templates to create your own SSM documents. These templates are based on the AWS public SSM documents.

AWS-RunPowerShellScript

```
{  
  "schemaVersion": "1.2",  
  "description": "Run a PowerShell script or specify the paths to scripts to  
run.",  
  "parameters": {  
    "commands": {  
      "type": "StringList",  
      "description": "(Required) Specify the commands to run or the  
paths to existing scripts on the instance.",  
      "minItems": 1,  
      "displayType": "textarea"  
    },  
    "workingDirectory": {  
      "type": "String",  
      "default": "",  
      "description": "(Optional) The path to the working directory on  
your instance.",  
      "maxChars": 4096  
    },  
    "executionTimeout": {  
      "type": "String",  
      "default": "3600",  
      "description": "(Optional) The time in seconds for a command to be  
completed before it is considered to have failed. Default is 3600 (1 hour).  
Maximum is 28800 (8 hours).",  
      "allowedPattern": "([1-9][0-9]{0,3})|(1[0-9]{1,4})|(2[0-7][0-9]  
{1,3})|(28[0-7][0-9]{1,2})|(28800)"  
    }  
  },  
  "runtimeConfig": {  
    "aws:runPowerShellScript": {  
      "properties": [  
        {  
          "id": "0.aws:runPowerShellScript",  
          "runCommand": "{{ commands }}",  
          "workingDirectory": "{{ workingDirectory }}",  
          "timeoutSeconds": "{{ executionTimeout }}"  
        }  
      ]  
    }  
  }  
}
```

AWS-ConfigureCloudWatch

```
{
```

```
"schemaVersion": "1.2",
"description": "Export metrics and log files from your instances to Amazon
CloudWatch.",
"parameters": {
  "status": {
    "type": "String",
    "default": "Enabled",
    "description": "(Optional) Enable or disable CloudWatch. Valid
values: Enabled | Disabled",
    "allowedValues": [
      "Enabled",
      "Disabled"
    ]
  },
  "properties": {
    "type": "String",
    "default": "",
    "description": "(Optional) The configuration for CloudWatch
in JSON format. Learn more at http://docs.aws.amazon.com/ssm/latest/
APIReference/aws-cloudWatch.html",
    "displayType": "textarea"
  }
},
"runtimeConfig": {
  "aws:cloudWatch": {
    "settings": {
      "startType": "{{ status }}"
    },
    "properties": "{{ properties }}"
  }
}
}
```

AWS-JoinDirectoryServiceDomain

```
{
  "schemaVersion": "1.2",
  "description": "Join your instances to an AWS Directory Service domain.",
  "parameters": {
    "directoryId": {
      "type": "String",
      "description": "(Required) The ID of the AWS Directory Service
directory."
    },
    "directoryName": {
      "type": "String",
      "description": "(Required) The name of the directory; for example,
test.example.com"
    },
    "directoryOU": {
      "type": "String",
      "default": "",
      "description": "(Optional) The Organizational Unit (OU)
and Directory Components (DC) for the directory; for example,
OU=test,DC=example,DC=com"
    },
    "dnsIpAddresses": {
      "type": "StringList",

```



```
        "default": [
            ],
            "description": "(Optional) The IP addresses of the DNS servers
in the directory. Required when DHCP is not configured. Learn more
at http://docs.aws.amazon.com/directoryservice/latest/admin-guide/
dns\_with\_simple\_ad.html",
            "allowedPattern": "((25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\\.){3}
(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)"
        }
    },
    "runtimeConfig": {
        "aws:domainJoin": {
            "properties": {
                "directoryId": "{{ directoryId }}",
                "directoryName": "{{ directoryName }}",
                "directoryOU": "{{ directoryOU }}",
                "dnsIpAddresses": "{{ dnsIpAddresses }}"
            }
        }
    }
}
```

AWS-InstallPowerShellModule

```
{
    "schemaVersion": "1.2",
    "description": "Deploy and install PowerShell modules.",
    "parameters": {
        "workingDirectory": {
            "type": "String",
            "default": "",
            "description": "(Optional) The path to the working directory on
your instance.",
            "maxChars": 4096
        },
        "source": {
            "type": "String",
            "description": "(Optional) The URL or local path on the instance
to the application .zip file."
        },
        "sourceHash": {
            "type": "String",
            "default": "",
            "description": "(Optional) The SHA256 hash of the zip file."
        },
        "commands": {
            "type": "StringList",
            "default": [
            ],
            "description": "(Optional) Specify PowerShell commands to run on
your instance.",
            "displayType": "textarea"
        },
        "executionTimeout": {
            "type": "String",
            "default": "3600",

```

```
        "description": "(Optional) The time in seconds for a command to be
        completed before it is considered to have failed. Default is 3600 (1 hour).
        Maximum is 28800 (8 hours).",
        "allowedPattern": "([1-9][0-9]{0,3})|(1[0-9]{1,4})|(2[0-7][0-9]
        {1,3})|(28[0-7][0-9]{1,2})|(28800)"
    },
    "runtimeConfig": {
        "aws:psModule": {
            "properties": [
                {
                    "id": "0.aws:psModule",
                    "runCommand": "{{ commands }}",
                    "source": "{{ source }}",
                    "sourceHash": "{{ sourceHash }}",
                    "workingDirectory": "{{ workingDirectory }}",
                    "timeoutSeconds": "{{ executionTimeout }}"
                }
            ]
        }
    }
}
```

AWS-InstallApplication

```
{
  "schemaVersion": "1.2",
  "description": "Install, repair, or uninstall an application using an .msi
  file.",
  "parameters": {
    "action": {
      "type": "String",
      "default": "Install",
      "description": "(Optional) The type of action to perform. Valid
      values: Install | Repair | Uninstall",
      "allowedValues": [
        "Install",
        "Repair",
        "Uninstall"
      ]
    },
    "parameters": {
      "type": "String",
      "default": "",
      "description": "(Optional) The parameters for the installer."
    },
    "source": {
      "type": "String",
      "description": "(Required) The URL or local path on the instance
      to the application .msi file."
    },
    "sourceHash": {
      "type": "String",
      "default": "",
      "description": "(Optional) The SHA256 hash of the .msi file."
    }
  },
  "runtimeConfig": {
```

```
"aws:applications":{
  "properties":[
    {
      "id":"0.aws:applications",
      "action":"{{ action }}",
      "parameters":"{{ parameters }}",
      "source":"{{ source }}",
      "sourceHash":"{{ sourceHash }}"
    }
  ]
}
```

Sharing SSM Documents

You can share Amazon EC2 Simple Systems Manager (SSM) documents privately or publicly. To privately share an SSM document, you modify the document permissions and allow specific individuals to access it according to their Amazon Web Services (AWS) ID. To publicly share a SSM document, you modify the document permissions and specify `All`.

Warning

Use shared SSM documents only from trusted sources. When using any shared document, carefully review the contents of the document before using it so that you understand how it will change the configuration of your instance. For more information about shared document best practices, see [Guidelines for Sharing and Using Shared SSM Documents \(p. 484\)](#).

Limitations

As you begin working with SSM documents, be aware of the following limitations.

- Only the owner can share a document.
- You must stop sharing a document before you can delete it. For more information, see [How to Modify Permissions for a Shared Document \(p. 487\)](#).
- You can share a document with a maximum of 20 AWS accounts.
- You can publicly share a maximum of five SSM documents.

Note

If you need to share more than the maximum number of AWS accounts or SSM documents, contact AWS Support.

This topic includes the following sections.

- [Guidelines for Sharing and Using Shared SSM Documents \(p. 484\)](#)
- [How to Share an SSM Document \(p. 485\)](#)
- [How to Modify Permissions for a Shared Document \(p. 487\)](#)
- [How to Use a Shared SSM Document \(p. 488\)](#)

Guidelines for Sharing and Using Shared SSM Documents

Review the following guidelines before you share or use a shared document.

Remove Sensitive Information

Review your SSM document carefully and remove any sensitive information. For example, verify that the document does not include your AWS credentials. If you share a document with specific

individuals, those users can view the information in the document. If you share a document publicly, anyone can view the information in the document.

Limit Run Command Actions Using an IAM User Trust Policy

Create a restrictive AWS Identity and Access Management (IAM) user policy for users who will have access to the document. The IAM policy determines which SSM documents a user can see in either the Amazon EC2 console or by calling `ListDocuments` using the AWS CLI or AWS Tools for Windows PowerShell. The policy also limits the actions the user can perform with an SSM document. You can create a restrictive policy so that a user can only use specific documents. For more information, see [Configuring Access to Systems Manager \(p. 400\)](#).

Review the Contents of a Shared Document Before Using It

Review the contents of every document that is shared with you, especially public documents, to understand the commands that will be executed on your instances. A document could intentionally or unintentionally have negative repercussions after it is run. If the document references an external network, review the external source before you use the document.

Send Commands Using the Document Hash

When you share a document, the system creates a Sha-256 hash and assigns it to the document. The system also saves a snapshot of the document content. When you send a command using a shared document, you can specify the hash in your command to ensure that the following conditions are true:

- You are executing a command from the correct SSM document
- The content of the document has not changed since it was shared with you.

If the hash does not match the specified document or if the content of the shared document has changed, the command returns an `InvalidDocument` exception. Note: The hash cannot verify document content from external locations.

How to Share an SSM Document

You can share an SSM document by using the Amazon EC2 console or by programmatically calling the `ModifyDocumentPermission` API operation using the AWS CLI, AWS Tools for Windows PowerShell, or the AWS SDK. Before you share a document, get the AWS account IDs of the people with whom you want to share. You will specify these account IDs when you share the document.

Share a Document Using the Amazon EC2 Console

1. Open the [Amazon EC2 console](#) and choose **Command History** in the navigation pane.
2. Choose **Documents**.
3. In the documents list, choose the document you want to share. Choose the **Permissions** tab and verify that you are the document owner. Only a document owner can share a document.
4. Choose **Edit**.
5. To share the command publicly, choose **Public** and then choose **Save**. To share the command privately, choose **Private**, enter the AWS account ID, choose **Add Permission**, and then choose **Save**.

Share a Document Using the AWS CLI

The following procedure requires that you specify a region for your CLI session. Run Command is currently available in the following SSM [regions](#).

1. Open the AWS CLI on your local computer and execute the following command to specify your credentials.

```
aws config
```

```
AWS Access Key ID: [your key]  
AWS Secret Access Key: [your key]  
Default region name: [us-east-1]  
Default output format [None]:
```

2. Use the following command to list all of the SSM documents that are available for you. The list includes documents that you created and documents that were shared with you.

```
aws ssm list-documents --document-filter-list key=Owner,value=all
```

3. Use the following command to get a specific document.

```
aws ssm get-document --name document name
```

4. Use the following command to get a description of the document.

```
aws ssm describe-document --name document name
```

5. Use the following command to view the permissions for the document.

```
aws ssm describe-document-permission --name document name --permission-type Share
```

6. Use the following command to modify the permissions for the document and share it. You must be the owner of the document to edit the permissions. This command privately shares the document with a specific individual, based on that person's AWS account ID.

```
aws ssm modify-document-permission --name document name --permission-type Share --account-ids-to-add AWS account ID
```

Use the following command to share a document publicly.

```
aws ssm modify-document-permission --name document name --permission-type Share --account-ids-to-add 'all'
```

Share a Document Using AWS Tools for Windows PowerShell

The following procedure requires that you specify a region for your PowerShell session. Run Command is currently available in the following SSM [regions](#).

1. Open **AWS Tools for Windows PowerShell** on your local computer and execute the following command to specify your credentials.

```
Set-AWSCredentials -AccessKey your key -SecretKey your key
```

2. Use the following command to set the region for your PowerShell session. The example uses the us-west-2 region.

```
Set-DefaultAWSRegion -Region us-west-2
```

3. Use the following command to list all of the SSM documents available for you. The list includes documents that you created and documents that were shared with you.

```
Get-SSMDocumentList -DocumentFilterList (@{"key"="Owner";"value"="All"})
```

4. Use the following command to get a specific document.

```
Get-SSMDocument -Name document name
```

5. Use the following command to get a description of the document.

```
Get-SSMDocumentDescription -Name document name
```

6. Use the following command to view the permissions of the document.

```
Get-SSMDocumentPermission -Name document name -PermissionType Share
```

7. Use the following command to modify the permissions for the document and share it. You must be the owner of the document to edit the permissions. This command privately shares the document with a specific individual, based on that person's AWS account ID.

```
Edit-SSMDocumentPermission -Name document name -PermissionType Share -  
AccountIdsToAdd AWS account ID
```

Use the following command to share a document publicly.

```
Edit-SSMDocumentPermission -Name document name -AccountIdsToAdd ('all') -  
PermissionType Share
```

How to Modify Permissions for a Shared Document

If you share a command, users can view and use that command until you either remove access to the SSM document or delete the SSM document. However, you cannot delete a document as long as it is shared. You must stop sharing it first and then delete it.

Stop Sharing a Document Using the Amazon EC2 Console

1. Open the [Amazon EC2 console](#) and choose **Command History** in the navigation pane.
2. Choose **Documents**.
3. In the documents list, choose the document you want to stop sharing. Choose the **Permissions** tab and verify that you are the document owner. Only a document owner can stop sharing a document.
4. Choose **Edit**.
5. Delete the AWS account ID that should no longer have access to the command, and then choose **Save**.

Stop Sharing a Document Using the AWS CLI

Open the AWS CLI on your local computer and execute the following command to stop sharing a command.

```
aws ssm modify-document-permission --name document name --permission-type  
Share --account-ids-to-remove 'AWS account ID'
```

Stop Sharing a Document Using AWS Tools for Windows PowerShell

Open **AWS Tools for Windows PowerShell** on your local computer and execute the following command to stop sharing a command.

```
Edit-SSMDocumentPermission -Name document name -AccountIdsToRemove AWS account ID -PermissionType Share
```

How to Use a Shared SSM Document

When you share an SSM document, the system generates an Amazon Resource Name (ARN) and assigns it to the command. If you select and execute a shared document from the Amazon EC2 console, you do not see the ARN. However, if you want to execute a shared SSM document from a command line application, you must specify a full ARN. You are shown the full ARN for an SSM document when you execute the command to list documents.

Note

You are not required to specify ARNs for AWS public documents (documents that begin with AWS-*) or commands that you own.

This section includes examples of how to view and execute shared SSM documents from the AWS CLI and AWS Tools for Windows PowerShell.

Using a Shared SSM Document from the AWS CLI

To list all public SSM documents

```
aws ssm list-documents --document-filter-list key=Owner,value=Public
```

To list private SSM documents that have been shared with you

```
aws ssm list-documents --document-filter-list key=Owner,value=Private
```

To list all SSM documents available to you

```
aws ssm list-documents --document-filter-list key=Owner,value=All
```

Execute a command from a shared SSM document using a full ARN

```
aws ssm send-command --document-name FullARN/name
```

For example:

```
aws ssm send-command --document-name arn:aws:ssm:us-east-1:12345678912:document/highAvailabilityServerSetup --instance-ids i-12121212
```

Using a Shared SSM Document from the AWS Tools for Windows PowerShell

To list all public SSM documents

```
Get-SSMDocumentList -DocumentFilterList @(New-Object  
Amazon.SimpleSystemsManagement.Model.DocumentFilter("Owner", "Public"))
```

To list private SSM documents that have been shared with you

```
Get-SSMDocumentList -DocumentFilterList @(New-Object  
Amazon.SimpleSystemsManagement.Model.DocumentFilter("Owner", "Shared"))
```

To get information about an SSM document that has been shared with you

```
Get-SSMDocument -Name FullARN/name
```

For example:

```
Get-SSMDocument -Name arn:aws:ssm:us-east-1:12345678912:document/  
highAvailabilityServerSetup
```

To get a description of an SSM document that has been shared with you

```
Get-SSMDocumentDescription -Name FullARN/name
```

For example:

```
Get-SSMDocumentDescription -Name arn:aws:ssm:us-east-1:12345678912:document/  
highAvailabilityServerSetup
```

To execute a command from a shared SSM document using a full ARN

```
Send-SSMCommand -DocumentName FullARN/name -InstanceId IDs
```

For example:

```
Send-SSMCommand -DocumentName arn:aws:ssm:us-east-1:555450671542:document/  
highAvailabilityServerSetup -InstanceId @"{i-273d4e9e}"
```

Amazon EC2 Run Command Walkthroughs

The following examples or walkthroughs to help you understand how to execute commands using Run Command from either the Amazon EC2 console or AWS Tools for Windows PowerShell.

Caution

If this is your first time using Run Command, we recommend executing commands against a test instance or an instance that is not being used in a production environment.

Contents

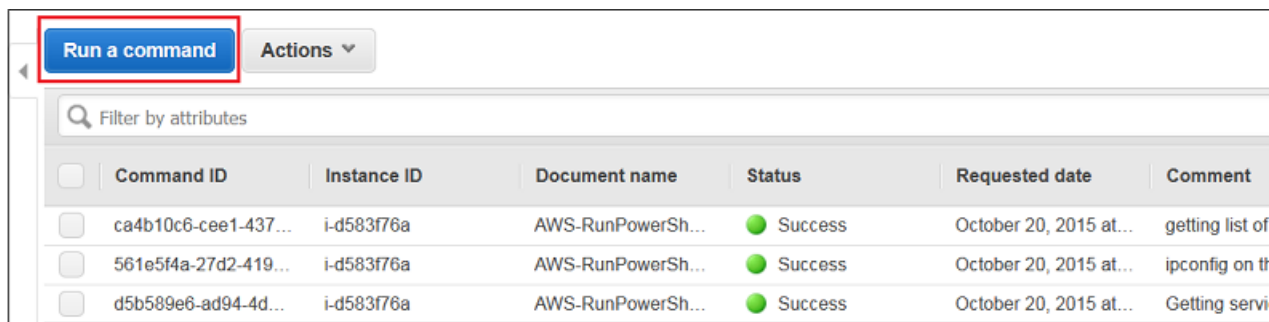
- [Amazon EC2 Run Command Walkthrough Using the Console \(p. 490\)](#)
- [Amazon EC2 Run Command Walkthrough Using the AWS Tools for Windows PowerShell \(p. 494\)](#)

Amazon EC2 Run Command Walkthrough Using the Console

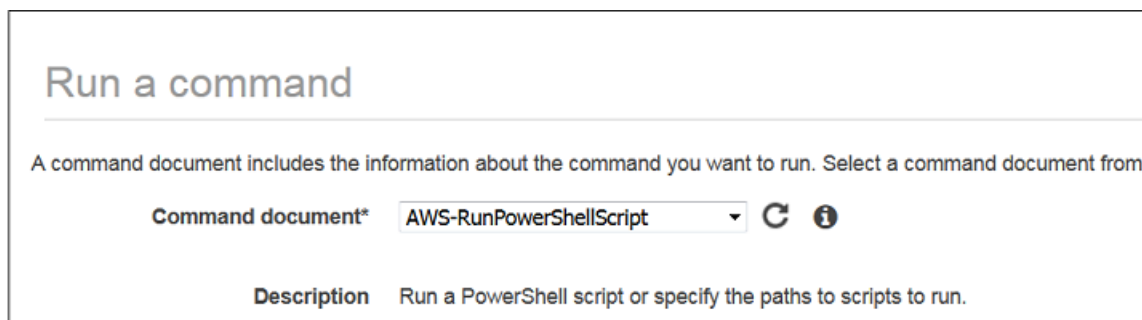
The following sample walkthrough shows you how to execute commands using Run Command from the **Command History** page in the Amazon EC2 console. This example shows how to execute a command with the AWS-RunPowerShellScript SSM JSON document. For PowerShell examples, see [Amazon EC2 Run Command Walkthrough Using the AWS Tools for Windows PowerShell](#) (p. 494).

To execute a command using Run Command from the console

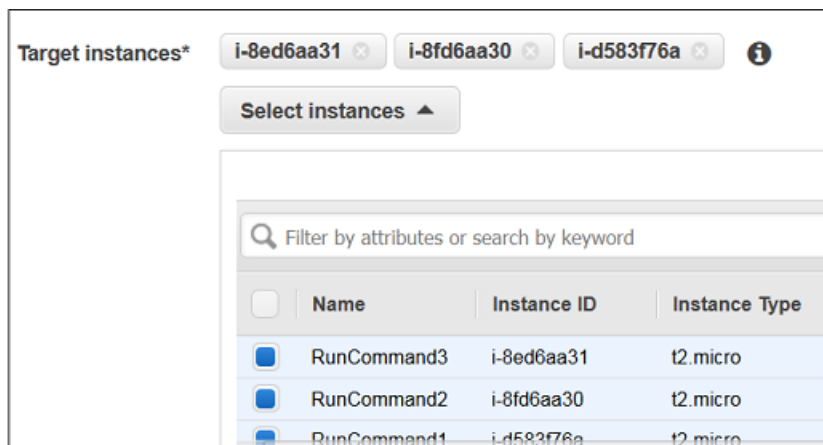
1. In the [Amazon EC2 console](#) choose **Command History** in the navigation pane, and then choose **Run a Command**.



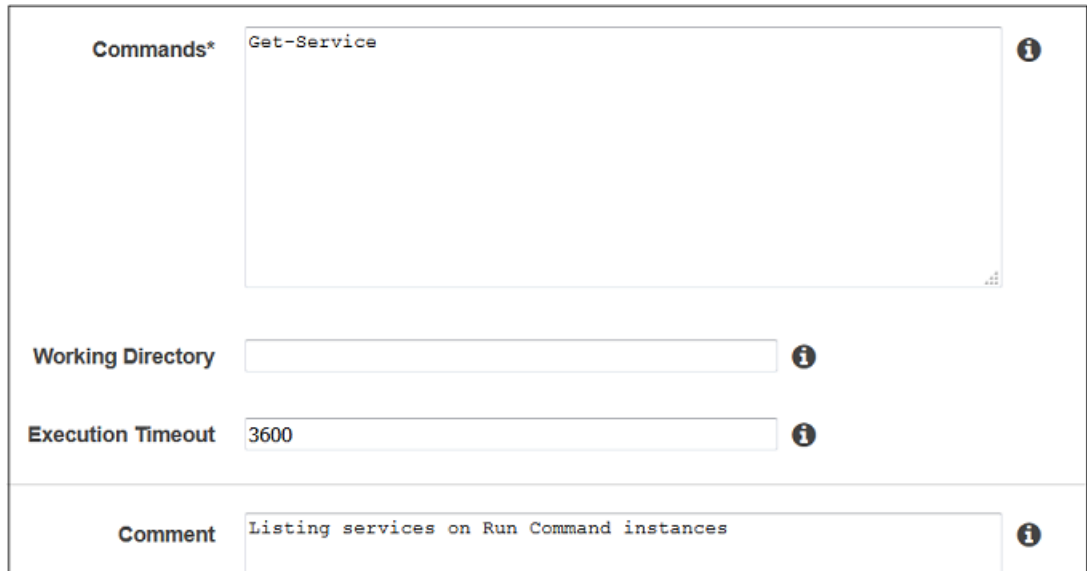
2. In the **Command document** list, choose **AWS-RunPowerShellScript**.



3. Choose **Select instances**, and then choose the instances where you want to execute the command. If you do not see a complete list of instances, the missing instances might not be configured properly for Run Command. For more information, see [Systems Manager Prerequisites](#) (p. 394).



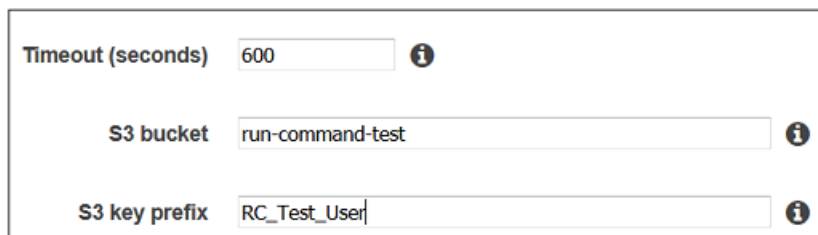
4. Type a valid PowerShell command or the path to a PowerShell script file in the **Commands** field. You can specify a **Working Directory** and **Execution Timeout**, if you want. The **Execution Timeout** is the number of seconds the EC2Config service or the SSM agent will attempt to run the command before it is considered to have failed. We recommend entering a comment in the **Comments** field. A comment will help you identify the command in the list of pending commands and make it easier to view the output.



The screenshot shows a configuration form for a Run Command operation. It includes the following fields:

- Commands***: A text area containing the PowerShell command `Get-Service`.
- Working Directory**: An empty text input field.
- Execution Timeout**: A text input field containing the value `3600`.
- Comment**: A text input field containing the text `Listing services on Run Command instances`.

5. In the **Timeout (seconds)** field, type the number of seconds Run Command should attempt to reach instances before an instance is considered unreachable and the command execution fails.
6. In the **S3 bucket** field, type the name of an Amazon S3 bucket where you want to store command output. Enter an Amazon S3 subfolder in the **S3 key prefix** field. A subfolder can help you organize output if you are executing multiple commands against multiple instances.



The screenshot shows a configuration form for a Run Command operation. It includes the following fields:

- Timeout (seconds)**: A text input field containing the value `600`.
- S3 bucket**: A text input field containing the value `run-command-test`.
- S3 key prefix**: A text input field containing the value `RC_Test_User`.

7. Choose **Run** to execute the command simultaneously on the selected instances. Run Command displays a status screen.
8. Choose **View results**.

Run a command



Success

We are running your command against the instances listed below.

Instance IDs i-8ed6aa31, i-8fd6aa30, i-d583f76a

Command ID 65555b90-ee60-4520-9dc3-e42e94445469

The command list shows three invocations for the command because it was sent to three instances. Each invocation has its own **Command ID** and status. To view status, choose an invocation, choose the **Output** tab for the invocation, and then choose **View Output**.

Run a command Actions

Filter by attributes

Command ID	Instance ID	Document name	Status	Requested date	Command
65555b90-ee60-45...	i-8fd6aa30	AWS-RunPowerSh...	Success	October 21, 2015 at...	Listing
65555b90-ee60-45...	i-d583f76a	AWS-RunPowerSh...	Success	October 21, 2015 at...	Listing
65555b90-ee60-45...	i-8ed6aa31	AWS-RunPowerSh...	Success	October 21, 2015 at...	Listing
ca4b10c6-cee1-437...	i-d583f76a	AWS-RunPowerSh...	Success	October 20, 2015 at...	getting
561e5f4a-27d2-419...	i-d583f76a	AWS-RunPowerSh...	Success	October 20, 2015 at...	ipconfi

Command ID: 65555b90-ee60-4520-9dc3-e42e94445469 Instance ID: i-8fd6aa30

Description Output

Command ID 65555b90-ee60-4520-9dc3-e42e94445469 **Instance**

Document name AWS-RunPowerShellScript **Sta**

Date requested October 21, 2015 at 3:56:59 PM UTC-7 **Comm**

Output S3 bucket run-command-test **Document paramet**

The system displays the output in your browser. If the output is longer than 2500 characters, only the first 2500 characters are shown and the rest is truncated.

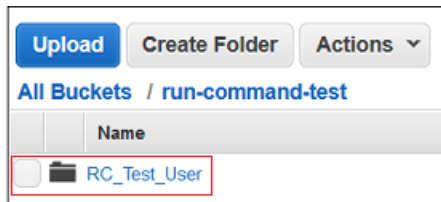
```

Commands > Output

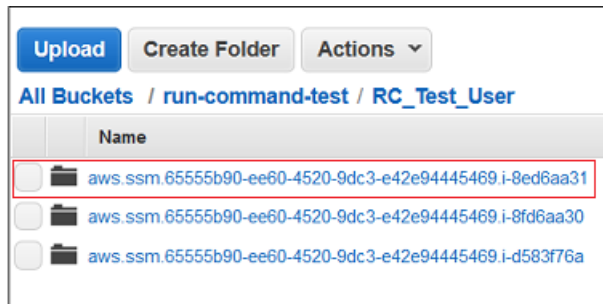
Output for aws:runPowerShellScript

Status  Name                DisplayName
-----  ----                -
Stopped AeLookupSvc         Application Experience
Stopped ALG            Application Layer Gateway Service
Stopped AppIDSvc       Application Identity
Running Appinfo         Application Information
Stopped AppMgmt        Application Management
Stopped AppReadiness   App Readiness
Stopped AppXSvc        AppX Deployment Service (AppXSVC)
Stopped AudioEndpointBu... Windows Audio Endpoint Builder
Stopped Audiosrv       Windows Audio
Running AWSLiteAgent   AWS Lite Guest Agent
Running BFE            Base Filtering Engine
Running BITS           Background Intelligent Transfer Ser...
Running BrokerInfrastru... Background Tasks Infrastructure Ser...
Stopped Browser        Computer Browser
Running CertPropSvc    Certificate Propagation
Stopped cfn-hup         CloudFormation cfn-hup
Stopped COMSysApp      COM+ System Application
Running CryptSvc       Cryptographic Services
Running DcomLaunch     DCOM Server Process Launcher
Stopped defragsvc      Optimize drives
Stopped DeviceAssociati... Device Association Service
Stopped DeviceInstall  Device Install Service
Running Dhcp           DHCP Client
-----Output truncated-----
  
```

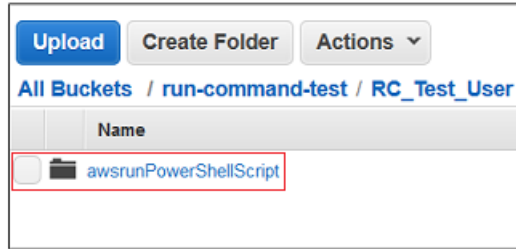
- To view the full command output in Amazon S3. Open the [Amazon S3 console](#) and choose your Amazon S3 bucket.



- Choose the *Command-ID.Instance-ID* for which you want to view command output.



- Choose the **awsrunShellScript** sub-folder.



12. Choose the **stdout.txt** file. S3 displays the full command output.



Amazon EC2 Run Command Walkthrough Using the AWS Tools for Windows PowerShell

The following examples show how to use the Tools for Windows PowerShell to view information about commands and command parameters, how to execute commands, and how to view the status of those commands. This walkthrough includes an example for each of the pre-defined SSM documents.

Tip

The **Command History** page in the console includes a section called **AWS Command Line Interface command**. This section displays a usable CLI script that's generated based on the parameters you entered.

Configure AWS Tools for Windows PowerShell Session Settings

Open **AWS Tools for Windows PowerShell** on your local computer and execute the following command to specify your credentials. You must either have administrator privileges on the instances you want to configure or you must have been granted the appropriate permission in IAM. For more information, see [Systems Manager Prerequisites \(p. 394\)](#).

```
Set-AWSCredentials -AccessKey key_name -SecretKey key_name
```

Execute the following command to set the region for your PowerShell session. The example uses the us-east-1 region. Run Command is currently available in the following SSM [regions](#).

```
Set-DefaultAWSRegion -Region us-east-1
```

List all Available Documents

This command lists all of the documents available for your account:

```
Get-SSMDocumentList
```

Run PowerShell Commands or Scripts

Using Run Command and the AWS-RunPowerShell document, you can execute any command or script on an EC2 instance as if you were logged onto the instance using Remote Desktop. You can issue commands or type in a path to a local script to execute the command.

View the description and available parameters

```
Get-SSMDocumentDescription -Name "AWS-RunPowerShellScript"
```

View more information about parameters

```
Get-SSMDocumentDescription -Name "AWS-RunPowerShellScript" | select -  
ExpandProperty Parameters
```

Send a command using the AWS-RunPowerShellScript document

The following command shows the contents of the C:\Users directory and the contents of the C:\ directory on two instances.

```
$runPSCCommand=Send-SSMCommand -InstanceId @( 'Instance-ID', 'Instance-ID' ) -  
DocumentName AWS-RunPowerShellScript -Comment 'Demo AWS-RunPowerShellScript  
with two instances' -Parameter @{ 'commands'=@('dir C:\Users', 'dir C:\')}
```

Get command request details

The following command uses the Command ID to get the status of the command execution on both instances. This example uses the Command ID that was returned in the previous command.

```
Get-SSMCommand -CommandId $runPSCCommand.CommandId
```

The status of the command in this example can be Success, Pending, or InProgress.

Get command information per instance

The following command uses the command ID from the previous command to get the status of the command execution on a per instance basis.

```
Get-SSMCommandInvocation -CommandId $runPSCCommand.CommandId
```

Get command information with response data for a specific instance

The following command returns the output of the original Send-SSMCommand for a specific instance.

```
Get-SSMCommandInvocation -CommandId $runPSCCommand.CommandId -Details $true -  
InstanceId Instance-ID | select -ExpandProperty CommandPlugins
```

Cancel a command

The following command cancels the Send-SSMCommand for the AWS-RunPowerShellScript document.

```
$cancelCommandResponse=Send-SSMCommand -InstanceId @( 'Instance-  
ID', 'Instance-ID' ) -DocumentName AWS-RunPowerShellScript -Comment  
'Demo AWS-RunPowerShellScript with two instances' -Parameter  
@{ 'commands'='Start-Sleep -Seconds 120; dir C:\'} Stop-SSMCommand -
```

```
CommandId $cancelCommandResponse.CommandId Get-SSMCommand -CommandId  
$cancelCommandResponse.CommandId
```

Check the command status

The following command checks the status of the Cancel command

```
Get-SSMCommand -CommandId $cancelCommandResponse.CommandId
```

Install an Application Using the AWS-InstallApplication Document

Using Run Command and the AWS-InstallApplication document, you can install, repair, or uninstall applications on instances. The command requires the path or address to an MSI.

View the description and available parameters

```
Get-SSMDocumentDescription -Name "AWS-InstallApplication"
```

View more information about parameters

```
Get-SSMDocumentDescription -Name "AWS-InstallApplication" | select -  
ExpandProperty Parameters
```

Send a command using the AWS-InstallApplication document

The following command installs a version of Python on your instance in unattended mode, and logs the output to a local text file on your C: drive.

```
$installAppCommand=Send-SSMCommand -InstanceId Instance-ID -DocumentName  
AWS-InstallApplication -Parameter @{ 'source'='https://www.python.org/ftp/  
python/2.7.9/python-2.7.9.msi'; 'parameters'='/norestart /quiet /log c:  
\pythoninstall.txt' }
```

Get command information per instance

The following command uses the Command ID to get the status of the command execution

```
Get-SSMCommandInvocation -CommandId $installAppCommand.CommandId -Details  
$true
```

Get command information with response data for a specific instance

The following command returns the results of the Python installation.

```
Get-SSMCommandInvocation -CommandId $installAppCommand.CommandId -Details  
$true -InstanceId Instance-ID | select -ExpandProperty CommandPlugins
```

Install a PowerShell Module Using the AWS-InstallPowerShellModule JSON Document

You can use Run Command to install PowerShell modules on an EC2 instance. For more information about PowerShell modules, see [Windows PowerShell Modules](#).

View the description and available parameters

```
Get-SSMDocumentDescription -Name "AWS-InstallPowerShellModule"
```

View more information about parameters

```
Get-SSMDocumentDescription -Name "AWS-InstallPowerShellModule" | select -  
ExpandProperty Parameters
```

Install a PowerShell module

The following command downloads the EZOut.zip file, installs it, and then runs an additional command to install XPS viewer. Lastly, the output of this command is uploaded to an Amazon S3 bucket named demo-ssm-output-bucket.

```
$installPSCCommand=Send-SSMCommand -InstanceId Instance-ID -DocumentName  
AWS-InstallPowerShellModule -Parameter @{'source'='https://  
gallery.technet.microsoft.com/EZOut-33ae0fb7/file/110351/1/  
EZOut.zip';'commands'=@('Add-WindowsFeature -name XPS-Viewer -restart')}} -  
OutputS3BucketName demo-ssm-output-bucket
```

Get command information per instance

The following command uses the Command ID to get the status of the command execution.

```
Get-SSMCommandInvocation -CommandId $installPSCCommand.CommandId -Details  
$true
```

Get command information with response data for the instance

The following command returns the output of the original Send-SSMCommand for the specific command ID.

```
Get-SSMCommandInvocation -CommandId $installPSCCommand.CommandId -Details  
$true | select -ExpandProperty CommandPlugins
```

Join an Instance to a Domain Using the AWS-JoinDirectoryServiceDomain JSON Document

Using Run Command, you can quickly join an instance to an AWS Directory Service domain. Before executing this command you must [create a directory](#). We also recommend that you learn more about the AWS Directory Service. For more information, see [What Is AWS Directory Service?](#)

Currently you can only join an instance to a domain. You cannot remove an instance from a domain.

View the description and available parameters

```
Get-SSMDocumentDescription -Name "AWS-JoinDirectoryServiceDomain"
```

View more information about parameters

```
Get-SSMDocumentDescription -Name "AWS-JoinDirectoryServiceDomain" | select -  
ExpandProperty Parameters
```

Join an instance to a domain

The following command joins the instance to the given AWS Directory Service domain and uploads any generated output to the Amazon S3 bucket.


```
$domainJoinCommand=Send-SSMCommand -InstanceId Instance-ID -DocumentName  
AWS-JoinDirectoryServiceDomain -Parameter @{ 'directoryId'='d-9067386b64';  
'directoryName'='ssm.test.amazon.com'; 'dnsIpAddresses'=@('172.31.38.48',  
'172.31.55.243')} -OutputS3BucketName demo-ssm-output-bucket
```

Get command information per instance

The following command uses the Command ID to get the status of the command execution.

```
Get-SSMCommandInvocation -CommandId $domainJoinCommand.CommandId -Details  
$true
```

Get command information with response data for the instance

This command returns the output of the original Send-SSMCommand for the specific command ID.

```
Get-SSMCommandInvocation -CommandId $domainJoinCommand.CommandId -Details  
$true | select -ExpandProperty CommandPlugins
```

Send Windows Metrics to Amazon CloudWatch using the AWS-ConfigureCloudWatch document

You can send Windows Server messages in the application, system, security, and Event Tracing for Windows (ETW) logs to Amazon CloudWatch Logs. When you enable logging for the first time, SSM sends all logs generated within 1 minute from the time that you start uploading logs for the application, system, security, and ETW logs. Logs that occurred before this time are not included. If you disable logging and then later re-enable logging, SSM sends logs from the time it left off. For any custom log files and Internet Information Services (IIS) logs, SSM reads the log files from the beginning. In addition, SSM can also send performance counter data to Amazon CloudWatch.

If you previously enabled CloudWatch integration in EC2Config, the SSM settings override any settings stored locally on the instance in the C:\Program Files\Amazon\EC2ConfigService\Settings\AWS.EC2.Windows.CloudWatch.json file. For more information about using EC2Config to manage performance counters and logs on single instance, see [Sending Performance Counters to CloudWatch and Logs to CloudWatch Logs](#).

View the description and available parameters

```
Get-SSMDocumentDescription -Name "AWS-ConfigureCloudWatch"
```

View more information about parameters

```
Get-SSMDocumentDescription -Name "AWS-ConfigureCloudWatch" | select -  
ExpandProperty Parameters
```

Send Application Logs to CloudWatch

The following command configures the instance and moves Windows Applications logs to CloudWatch.

```
$cloudWatchCommand=Send-SSMCommand -InstanceID Instance-ID -DocumentName  
'AWS-ConfigureCloudWatch' -Parameter @{ 'properties'='{ "engineConfiguration":  
{ "PollInterval": "00:00:15", "Components": [{ "Id": "ApplicationEventLog",  
"FullName": "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent, AWS.EC2.Windows.Clo  
"Parameters": { "LogName": "Application", "Levels": "7" } }, { "Id": "CloudWatch",  
"FullName": "AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput, AWS.EC2.Windows.CloudWatch",  
"Parameters": { "Region": "us-east-1", "LogGroup": "My-Log-Group",
```

```
"LogStream": "i-1234567890abcdef0"}]}, "Flows": {"Flows":  
[ "ApplicationEventLog, CloudWatch" ]}}}' }
```

Get command information per instance

The following command uses the Command ID to get the status of the command execution.

```
Get-SSMCommandInvocation -CommandId $cloudWatchCommand.CommandId -Details  
$true
```

Get command information with response data for a specific instance

The following command returns the results of the Amazon CloudWatch configuration.

```
Get-SSMCommandInvocation -CommandId $cloudWatchCommand.CommandId -Details  
$true -InstanceId Instance-ID | select -ExpandProperty CommandPlugins
```

Send Performance Counters to CloudWatch Using the [AWS-ConfigureCloudWatch](#) document

The following demonstration command uploads performance counters to CloudWatch. For more information, see the [Amazon CloudWatch Documentation](#).

```
$cloudWatchMetricsCommand=Send-SSMCommand -InstanceId Instance-  
ID -DocumentName 'AWS-ConfigureCloudWatch' -  
Parameter @{ 'properties'='{ "engineConfiguration":  
  { "PollInterval": "00:00:15", "Components": [{ "Id": "PerformanceCounter",  
    "FullName": "AWS.EC2.Windows.CloudWatch.PerformanceCounterComponent.PerformanceCounterInput",  
    "Parameters": { "CategoryName": "Memory", "CounterName": "Available  
MBytes", "InstanceName": "", "MetricName": "AvailableMemory",  
    "Unit": "Megabytes", "DimensionName": "", "DimensionValue": "" } }],  
  { "Id": "CloudWatch",  
    "FullName": "AWS.EC2.Windows.CloudWatch.CloudWatch.CloudWatchOutputComponent, AWS.EC2.Window  
    "Parameters": { "AccessKey": "", "SecretKey": "", "Region": "us-  
east-1", "NameSpace": "Windows-Default" } } ] } } } } }'  
[ "PerformanceCounter, CloudWatch" ]}}}' }
```

Enable/Disable Windows Automatic Update Using the [AWS-ConfigureWindowsUpdate](#) document

Using Run Command and the [AWS-ConfigureWindowsUpdate](#) document, you can enable or disable automatic Windows updates on your Windows instances. This command configures the Windows update agent to download and install Windows updates on the day and hour that you specify. If an update requires a reboot, the computer reboots automatically 15 minutes after updates have been installed. With this command you can also configure Windows update to check for updates but not install them. The [AWS-ConfigureWindowsUpdate](#) document is compatible with Windows Server 2008, 2008 R2, 2012, 2012 R2.

View the description and available parameters

```
Get-SSMDocumentDescription -Name "AWS-ConfigureWindowsUpdate"
```

View more information about parameters

```
Get-SSMDocumentDescription -Name "AWS-ConfigureWindowsUpdate" | select -  
ExpandProperty Parameters
```

Enable Windows automatic update

The following command configures Windows Update to automatically download and install updates daily at 10:00 pm.

```
$configureWindowsUpdateCommand = Send-SSMCommand -  
InstanceId Instance-ID -DocumentName 'AWS-ConfigureWindowsUpdate'  
-Parameters @{ 'updateLevel'='InstallUpdatesAutomatically';  
'scheduledInstallDay'='Daily'; 'scheduledInstallTime'=' 22:00' }
```

View command status for enabling Windows automatic update

The following command uses the Command ID to get the status of the command execution for enabling Windows Automatic Update.

```
Get-SSMCommandInvocation -Details $true -CommandId  
$configureWindowsUpdateCommand.CommandId | select -ExpandProperty  
CommandPlugins
```

Disable Windows automatic update

The following command lowers the Windows Update notification level so the system checks for updates but does not automatically update the instance.

```
$configureWindowsUpdateCommand = Send-SSMCommand -InstanceId Instance-  
ID -DocumentName 'AWS-ConfigureWindowsUpdate' -Parameters  
@{ 'updateLevel'='NeverCheckForUpdates' }
```

View command status for disabling Windows automatic update

The following command uses the Command ID to get the status of the command execution for disabling Windows automatic update.

```
Get-SSMCommandInvocation -Details $true -CommandId  
$configureWindowsUpdateCommand.CommandId | select -ExpandProperty  
CommandPlugins
```

Update EC2Config Using the AWS-UpdateEC2Config Document

Using Run Command and the AWS-EC2ConfigUpdate document, you can update the EC2Config service running on your Windows instances. This command can update the EC2Config service to the latest version or a version you specify.

View the description and available parameters

```
Get-SSMDocumentDescription -Name "AWS-UpdateEC2Config"
```

View more information about parameters

```
Get-SSMDocumentDescription -Name "AWS-UpdateEC2Config" | select -  
ExpandProperty Parameters
```

Update EC2Config to the latest version

```
Send-SSMCommand -InstanceId Instance-ID -DocumentName "AWS-UpdateEC2Config"
```

Get command information with response data for the instance

This command returns the output of the specified command from the previous Send-SSMCommand:

```
Get-SSMCommandInvocation -CommandId ID -Details $true -InstanceId Instance-ID  
| select -ExpandProperty CommandPlugins
```

Update EC2Config to a specific version

The following command will downgrade EC2Config to an older version:

```
Send-SSMCommand -InstanceId Instance-ID -DocumentName "AWS-UpdateEC2Config" -  
Parameter @{'version'='3.8.354'; 'allowDowngrade'='true' }
```

Manage Windows Updates Using Run Command

Run Command includes three documents to help you manage updates for Amazon EC2 Windows instances.

- **AWS-FindWindowsUpdates** — Scans an instance and determines which updates are missing.
- **AWS-InstallMissingWindowsUpdates** — Installs missing updates on your EC2 instance.
- **AWS-InstallSpecificUpdates** — Installs a specific update.

The following examples demonstrate how to perform the specified Windows Update management tasks.

Search for all missing Windows updates

```
Send-SSMCommand -InstanceId Instance-ID -DocumentName 'AWS-  
FindWindowsUpdates' -Parameters @{'UpdateLevel'='All' }
```

Install specific Windows updates

```
Send-SSMCommand -InstanceId Instance-ID -DocumentName  
'AWS-InstallSpecificWindowsUpdates' -Parameters  
@{'KbArticleIds'='123456,KB567890,987654' }
```

Install important missing Windows updates

```
Send-SSMCommand -InstanceId Instance-ID -DocumentName 'AWS-  
InstallMissingWindowsUpdates' -Parameters @{'UpdateLevel'='Important' }
```

Install missing Windows updates with specific exclusions

```
Send-SSMCommand -InstanceId Instance-ID -DocumentName  
'AWS-InstallMissingWindowsUpdates' -Parameters  
@{'UpdateLevel'='All'; 'ExcludeKbArticleIds'='KB567890,987654' }
```

Command Status and Monitoring

Amazon EC2 Run Command reports detailed status information about the different states a command experiences during processing and for each instance that processed the command. Run Command

includes options to monitor command status manually or automatically. Monitoring command status can help you troubleshoot problems if a command fails.

Contents

- [About Command Statuses \(p. 502\)](#)
- [About Monitoring Commands \(p. 504\)](#)
- [Getting Amazon SNS Notifications When a Command Changes Status \(p. 505\)](#)
- [Log Command Execution Status Changes for Run Command \(p. 510\)](#)

About Command Statuses

Run Command reports status details for three areas: plugins, invocations, and an overall command status. A *plugin* is a code-execution block that is defined in your command's SSM document. For example, the AWS-RunPowerShellScript document includes the `aws:runPowerShellScript` plugin. The AWS-* documents include only one plugin, but you can create your own documents that use multiple plugins. For more information about plugins, see [SSM Plugins](#) in the *Amazon EC2 Simple Systems Manager API Reference*.

When you send a command to multiple instances at the same time, each copy of the command targeting each instance is a *command invocation*. For example, if you use the AWS-RunPowerShellScript document and send an **ipconfig** command to 20 instances, that command has 20 invocations. Each command invocation individually reports status. The plugins for a given command invocation individually report status as well.

Lastly, Run Command includes an aggregated command status for all plugins and invocations. The aggregated command status can be different than the status reported by plugins or invocations, as noted in the following tables.

Note

If you execute commands to large numbers of instances using the `max-concurrency` or `max-errors` parameters, command status reflects the limits imposed by those parameters, as described in the following tables. For more information about these parameters, see [Sending a Command to Multiple Instances \(p. 476\)](#).

Detailed Status for Command Plugins and Invocations

Status	Details
Pending	The command was not yet received by the agent on the instance. If the command is not received by the agent before the value specified by the Timeout (seconds) parameter is reached, then the status changes to <code>Delivery Timed Out</code> .
In Progress	The command was received by the agent, or the command started executing on the instance. Depending on the result of all command plugins, the status will change to <code>Success</code> , <code>Failed</code> , or <code>Execution Timed Out</code> . If the agent is not available on the instance, the command status will show <code>In Progress</code> until the agent is available again. The status will then change to a terminal state.
Delayed	The system attempted to send the command to the instance but was not successful. The system will retry again.

Status	Details
Success	The command or plugin execution was successfully completed. This is a terminal state.
Delivery Timed Out	The command was not delivered to the instance before the delivery timeout expired. Delivery timeouts do not count against the parent command's <code>max-errors</code> limit, but they do contribute to whether the parent command status is <code>Success</code> or <code>Incomplete</code> . This is a terminal state.
Execution Timed Out	Command execution started on the instance, but the execution was not complete before the execution timeout expired. Execution timeouts count against the <code>max-errors</code> limit of the parent command. This is a terminal state.
Failed	The command was not successful on the instance. For a plugin, this indicates that the result code was not zero. For a command invocation, this indicates that the result code for one or more plugins was not zero. Invocation failures count against the <code>max-errors</code> limit of the parent command. This is a terminal state.
Canceled	The command was terminated before it was completed. This is a terminal state.
Undeliverable	The command can't be delivered to the instance. The instance might not exist or it might not be responding. Undeliverable invocations don't count against the parent command's <code>max-errors</code> limit, and they don't contribute to whether the parent command status is <code>Success</code> or <code>Incomplete</code> . This is a terminal state.
Terminated	The parent command exceeded its <code>max-errors</code> limit and subsequent command invocations were canceled by the system. This is a terminal state.

Detailed Status for a Command

Status	Details
Pending	The command was not yet received by an agent on any instances.
In Progress	The command has been sent to at least one instance but has not reached a final state on all instances.
Delayed	The system attempted to send the command to the instance but was not successful. The system will retry again.
Success	The command attempted to execute on all specified or targeted instances, all command

Status	Details
	invocations have reached a terminal state, and the value of <code>max-errors</code> was not reached. This is a terminal state.
Delivery Timed Out	The command was not delivered to the instance before the delivery timeout expired. The value of <code>max-errors</code> or more command invocations shows a status of <code>Delivery Timed Out</code> . This is a terminal state.
Execution Timed Out	Command execution started on the instance, but the execution was not complete before the execution timeout expired. The value of <code>max-errors</code> or more command invocations shows a status of <code>Execution Timed Out</code> . This is a terminal state.
Failed	The command was not successful on the instance. The value of <code>max-errors</code> or more command invocations shows a status of <code>Failed</code> . This is a terminal state.
Incomplete	The command was attempted on all instances and one or more of the invocations does not have a value of <code>Success</code> . However, not enough invocations failed for the status to be <code>Failed</code> . This is a terminal state.
Canceled	The command was terminated before it was completed. This is a terminal state.
Rate Exceeded	The number of instances targeted by the command exceeded the account limit for pending invocations. The system has canceled the command before executing it on any instance. This is a terminal state.

About Monitoring Commands

You can monitor command status manually or automatically. The method you choose will depend on the number of commands you send and the number of instances processing those commands. For example, if you're sending commands to hundreds of instances, then it's not practical to monitor command status by clicking the Refresh icon in the **Command History** page in the Amazon EC2 console. In this case, you might want to configure Amazon SNS notifications or CloudWatch Events.

Ways to Monitor Command Status

- Click the Refresh icon on the **Command History** page in the Amazon EC2 console.
- Call [list-commands](#) or [list-command-invocations](#) using the AWS CLI.
- Configure Amazon SNS to send notifications for all status changes or specific statuses like `Failed` or `TimedOut`. For more information, see [Getting Amazon SNS Notifications When a Command Changes Status \(p. 505\)](#).
- Configure CloudWatch Events to log status changes. For more information, see [Log Command Execution Status Changes for Run Command \(p. 510\)](#).

Getting Amazon SNS Notifications When a Command Changes Status

You can configure Amazon Simple Notification Service (Amazon SNS) to send notifications about the status of commands you send using Amazon EC2 Run Command. Amazon SNS coordinates and manages the delivery or sending of notifications to subscribing clients or endpoints. You can receive a notification whenever a command changes to a new state or changes to a specific state, such as failed or timed out. In cases where you send a command to multiple instances, you can receive a notification for each copy of the command sent to a specific instance. Each copy is called an *invocation*.

Amazon SNS can deliver notifications as HTTP or HTTPS POST, email (SMTP, either plain-text or in JSON format), or as a message posted to an Amazon Simple Queue Service (Amazon SQS) queue. For more information, see [What Is Amazon SNS](#) in the *Amazon Simple Notification Service Developer Guide*.

For example, if you configure Amazon SNS to send a notification when a command status changes to failed, SNS sends an email notification with the details of the command execution.

Note

If you prefer, you can use Amazon CloudWatch Events to configure a target to invoke an AWS Lambda function when a command changes status. For more information, see [Log Command Execution Status Changes for Run Command \(p. 510\)](#).

To set up Amazon SNS notifications when a command changes status, you must complete the following tasks.

1. [Configure Account Permissions \(p. 507\)](#)
2. [Create an IAM Role for Notifications \(p. 507\)](#)
3. [Configure Amazon SNS \(p. 508\)](#)
4. [Send a Command that Returns Status Notifications \(p. 509\)](#)

Configure Amazon SNS Notifications for SSM

Run Command supports sending Amazon SNS notifications for commands that enter the following statuses. For information about the conditions that cause a command to enter one of these statuses, see [Command Status and Monitoring \(p. 501\)](#).

- In Progress
- Success
- Failed
- Timed Out
- Canceled

Note

Commands sent using Run Command also report Cancelling and Pending status. These statuses are not captured by SNS notifications.

If you configure Run Command for SNS notifications, SNS sends summary messages that include the following information:

Field	Type	Description
EventTime	String	The time the event was triggered. The time stamp is important because SNS does not guarantee message

Field	Type	Description
		delivery order. Example: 2016-04-26T13:15:30Z
DocumentName	String	The name of the SSM document used to execute this command.
CommandId	String	The ID generated by Run Command after the command was sent.
ExpiresAfter	Date	If this time is reached and the command has not already started executing, it will not execute.
OutputS3BucketName	String	The Amazon Simple Storage Service (Amazon S3) bucket where the responses to the command execution should be stored.
OutputS3KeyPrefix	String	The Amazon S3 directory path inside the bucket where the responses to the command execution should be stored.
RequestedDateTime	String	The time and date the request was sent to this specific instance.
InstanceId	String	The instance targeted by the command.
Status	String	Command status for the command.

If you send a command to multiple instances, Amazon SNS can send messages about each copy or invocation of the command that include the following information:

Field	Type	Description
EventTime	String	The time the event was triggered. The time stamp is important because SNS does not guarantee message delivery order. Example: 2016-04-26T13:15:30Z
DocumentName	String	The name of the SSM document used to execute this command.
RequestedDateTime	String	The time and date the request was sent to this specific instance.

Field	Type	Description
CommandId	String	The ID generated by Run Command after the command was sent.
InstanceId	String	The instance targeted by the command.
Status	String	Command status for this invocation.

Configure Account Permissions

When you send a command that is configured for notifications, you specify a service role Amazon Resource Name (ARN). For example: `--service-role-arn=arn:aws:iam::123456789012:myrole`. This service role is used by SSM to trigger SNS notifications.

To receive notifications from the Amazon SNS service, you must either attach the `iam:PassRole` policy to your existing AWS Identity and Access Management (IAM) user account, or create a new IAM account and attach this policy to it. If you create a new account, you must also attach the `AmazonSSMFullAccess` policy so the account can communicate with the SSM API.

Use the following procedure to attach an IAM policy to your user account. If you need to create a new user account, see [Creating an IAM User in Your AWS Account](#) in the *IAM User Guide*.

To attach the `iam:PassRole` policy to your user account

1. Open the Identity and Access Management (IAM) console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Users** and select the user (under **User name**).
3. At the top of the page, copy your **User ARN** to the clipboard.
4. Under **Permissions**, verify that either the `AmazonSSMFullAccess` policy is listed or there is a comparable policy that gives you permission to the SSM API.
5. Choose **Add inline policy**.
6. On the **Set Permissions** page, choose **Policy Generator**, and then choose **Select**.
7. Verify that **Effect** is set to **Allow**.
8. From **AWS Services** choose **AWS Identity and Access Management**.
9. From **Actions** choose **PassRole**.
10. In the **Amazon Resource Name (ARN)** field, paste your ARN.
11. Choose **Add Statement**, and then choose **Next**.
12. On the **Review Policy** page, choose **Apply Policy**.

Create an IAM Role for Notifications

In the previous procedure, you added an IAM policy to your user account so that you could send commands that return notifications. In the following procedure, you will create a role so that the SSM service can act on your behalf when sending notifications.

1. Open the Identity and Access Management (IAM) console at <https://console.aws.amazon.com/iam/>.

2. In the navigation pane, choose **Roles**, and then choose **Create New Role**.
3. In **Step 1: Set Role Name** enter a name that identifies this role as a Run Command role for notifications.
4. In **Step 2: Select Role Type** choose **Amazon EC2**. The system skips **Step 3: Establish Trust** because this is a managed policy.
5. In **Step 4: Attach Policy** choose **AmazonSNSFullAccess**.
6. Choose **Next Step** and then choose **Create Role**. The system returns you to the **Roles** page.
7. Locate the role you just created and double-click it.
8. Choose the **Trust Relationships** tab, and then choose **Edit Trust Relationship**.
9. Add "ssm.amazonaws.com" to the existing policy as the following code snippet illustrates:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com",
        "Service": "ssm.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Note

You must add a comma after the existing entry. "Service": "sns.amazonaws.com", or the JSON will not validate.

10. Choose **Update Trust Policy**.
11. Copy or make a note of the **Role ARN**. You will specify this ARN when you send a command that is configured to return notifications.

Configure Amazon SNS

To use Amazon SNS to send email notifications, you must first create a *topic* and then subscribe your email addresses to the topic.

Create an Amazon SNS Topic

An Amazon SNS topic is a logical access point, a communication channel that Run Command uses to send the notifications. You create a topic by specifying a name for your topic.

For more information, see [Create a Topic](#) in the *Amazon Simple Notification Service Developer Guide*.

Note

After you create the topic, copy or make a note of the **Topic ARN**. You will specify this ARN when you send a command that is configured to return status notifications.

Subscribe to the Amazon SNS Topic

To receive the notifications that Run Command sends to the topic, you must subscribe an endpoint to the topic. In this procedure, for **Endpoint**, specify the email address where you want to receive the notifications from Run Command.

For more information, see [Subscribe to a Topic](#) in the *Amazon Simple Notification Service Developer Guide*.

Confirm Your Amazon SNS Subscription

Amazon SNS sends a confirmation email to the email address that you specified in the previous step.

Make sure you open the email from AWS Notifications and choose the link to confirm the subscription before you continue with the next step.

You will receive an acknowledgement message from AWS. Amazon SNS is now configured to receive notifications and send the notification as an email to the email address that you specified.

Send a Command that Returns Status Notifications

This section shows you how to send a command that is configured to return status notifications using either the Amazon EC2 console or the AWS Command Line Interface (AWS CLI).

To send a command from the Amazon EC2 console that returns notifications

1. Open the [Amazon EC2 console](#) and choose **Command History** in the navigation pane.
2. Choose **Run a Command**.
3. In the **Command document** list, choose an SSM document.
4. Choose **Select target instances** to select the instances where you want the command to run. If you do not see a complete list of instances, the missing instances might not be configured properly for Run Command. For more information, see [Systems Manager Prerequisites \(p. 394\)](#).
5. Enter information in the fields required by the SSM document. In the **SNS Notifications** section, choose **Enable SNS notifications**.
6. In the **Role** field, type or paste the IAM role ARN you created earlier.
7. In the **SNS Topic** field, type or paste the Amazon SNS ARN you created earlier.
8. In the **Notify me on** field, choose the events for which you want to receive notifications.
9. In the **Notify me for** field, choose to receive notifications for each copy of a command sent to multiple instances (invocations) or the command summary.
10. Choose **Run**.
11. Check your email for a message from Amazon SNS and open the email. Amazon SNS can take a few minutes to send the mail.

To send a command that is configured for notifications from the AWS CLI

1. Open the AWS CLI.
2. Specify parameters in the following command.

```
aws ssm send-command --instance-ids "ID-1, ID-2" --document-name "name"  
  --parameters commands=date --service-role ServiceRole ARN --notification-  
  config NotificationArn=SNS ARN
```

For example

```
aws ssm send-command --instance-ids "i-12345678, i-34567890" --  
  document-name "AWS-RunPowerShellScript" --parameters commands=date --  
  service-role arn:aws-cn:iam::123456789012:myrole --notification-config  
  NotificationArn=arn:aws-cn:sns:cn-north-1:123456789012:test
```

3. Press Enter.
4. Check your email for a message from Amazon SNS and open the email. Amazon SNS can take a few minutes to send the mail.

For more information about configuring Run Command from the command line, see [Amazon EC2 Simple Systems Manager API Reference](#) and the [SSM AWS CLI Reference](#).

Log Command Execution Status Changes for Run Command

You can use Amazon CloudWatch Events and a simple AWS Lambda function to log command execution status changes. You can create a rule that runs whenever there is a state transition, or when there is a transition to one or more states that are of interest.

Amazon EC2 Simple Systems Manager Event Types

SSM sends the following data to CloudWatch Events.

Example 1—EC2 Command Status-change Notification: This example includes information about execution status changes for a command that was sent to multiple instances.

```
{
  "version": "0",
  "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
  "detail-type": "EC2 Run Command - Command Status change",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2016-03-14T18:43:48Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2:us-east-1:123456789012:instance/i-12345678",
    "arn:aws:ec2:us-east-1:123456789012:instance/i-12345670",
    "arn:aws:ec2:us-east-1:123456789012:instance/i-12345679"
  ],
  "detail": {
    "command-id": "aws.ssm.12345678-1234-1234-1234-12345678",
    "requested-date-time": "2016-03-14T18:43:48Z",
    "expire-after": "2016-03-14T18:43:48Z",
    "output-s3bucket-name": "mybucket",
    "output-s3key-prefix": "test",
    "parameters": "parameter",
    "status": "Success"
  }
}
```

Example 2—EC2 Command Invocation Status-change Notification: This example includes information about a command that was sent to multiple instances, but the event shows details for only one instance, or *invocation* of that command.

```
{
  "version": "0",
  "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
  "detail-type": "EC2 Run Command - Command Invocation Status change",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2016-03-14T18:43:48Z",
```

```
"region": "us-east-1",
"resources": [
  "arn:aws:ec2:us-east-1:123456789012:instance/i-12345678"
],
"detail": {
  "command-id": "aws.ssm.12345678-1234-1234-1234-12345678",
  "instance-id": "i-12345678",
  "requested-date-time": "2016-03-14T18:43:48Z",
  "status": "Success"
}
}
```

Log SSM Command Execution Status Changes

In the following example scenario, you will create a simple AWS Lambda function, route events from SSM to it, and then test your scenario to ensure that it's set up correctly.

To log command execution status changes for Run Command, you must do the following.

1. [Step 1: Create an AWS Lambda Function \(p. 511\)](#)
2. [Step 2: Route Events to Your AWS Lambda Function \(p. 511\)](#)
3. [Step 3: Test Your Amazon CloudWatch Events Rule \(p. 512\)](#)

Step 1: Create an AWS Lambda Function

To create an AWS Lambda function

1. Open the AWS Lambda console at <https://console.aws.amazon.com/lambda/>.
2. Choose **Create a Lambda function**, and then on the **Select blueprint** screen, choose **hello-world**.
3. On the **Configure function** screen, in the **Name** field, type a name for the event. This example uses **SomethingHappened**.
4. In the **Lambda function code** section, edit the sample code to match the following example:

```
console.log('Loading function');

exports.handler = function(event, context, callback) {
  console.log('SomethingHappened()');
  console.log('Here is the event:', JSON.stringify(event, null, 2));
  callback(null, "Ready");
};
```

5. Under **Lambda function handler and role**, in the **Role** field, if you have a **lambda_basic_execution_rule**, select it. Otherwise, create a new basic execution role.
6. Choose **Next**, and then on the **Review** screen, choose **Edit** to make any changes. If you're satisfied with the function, choose **Create function**.

Step 2: Route Events to Your AWS Lambda Function

To create a CloudWatch Events rule

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Events**.

3. Choose **Create rule**, and then under **Event selector**, choose **EC2 instance state-change notification**.
4. Choose **Specific state(s)**, and then **Running** from the list.
5. Do one of the following:
 - To make the rule respond to any of your instances in the region, choose **Any instance**.
 - To make the rule respond to a specific instance, choose **Specific instance(s)** and then in the text box, enter the instance ID.
6. Under **Targets**, choose **Add target**. In the **Select target type** list, choose **AWS Lambda function**.
7. In the **Function** list, select the function that you created in "**Step 1: Create an AWS Lambda Function**."
8. Choose **Configure input**, and then choose one of the following options:
 - **Matched event**
 - Sends all of the data fields in the event to CloudWatch Logs.
 - **Part of the matched event**
 - Sends only the specified data field of the event to CloudWatch Logs. You specify the part of the event using a string formatted `$.first_parameter.second_parameter`

For example, to send just the Amazon EC2 instance ID, type `$.detail.state` in the field.
 - **Constant**
 - Sends a JSON-formatted text string that you specify to CloudWatch Logs. For example, to send a text string for the event, type `{"Name":"MyInstance"}`. The constant must be valid JSON.
9. Choose **Configure details**. On the **Configure rule details** screen, in the **Name** field, type a name for the rule.
10. In the **Description** field, type a brief description for your rule, for example, **Log command execution status changes**.
11. If you're satisfied with the rule, choose **Create rule**.

Step 3: Test Your Amazon CloudWatch Events Rule

You can test your rule by executing a command with Run Command. After waiting a few minutes for the command to process, check your AWS Lambda metrics in the Amazon CloudWatch Events console to verify that your function was invoked.

To test your CloudWatch Events rule using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane choose **Command History**, and then execute a command to one or more instances. For more information about executing a command, see [Executing a Command Using Amazon EC2 Run Command \(p. 442\)](#).
3. To view your AWS Lambda metrics, open the CloudWatch console <https://console.aws.amazon.com/cloudwatch/>.
4. In the navigation pane, under **Metrics**, choose **Lambda** to view the metrics generated by your Lambda function.
5. To view the output from your function, in the navigation pane, choose **Logs**, and then in the **Log Groups** list, select the `/aws/lambda` log group that contains the data.
6. Under **Log Streams**, select a log stream to view the data about command execution status changes.

Troubleshooting Amazon EC2 Run Command

Use the following information to help troubleshoot problems with Run Command. For information about troubleshooting Run Command for Linux, see [Troubleshooting Run Command](#) in the User Guide for Linux.

Where Are My Instances?

If you do not see the expected list of instances when you choose **Select Target instances** then verify that your instance is configured with an AWS Identity and Access Management (IAM) role that enables the instance to communicate with the SSM API. Also verify that your user account has an IAM user trust policy that enables your account to communicate with the SSM API. The following procedures describe how to configure the instance role and the user trust policy.

Note

You must assign the IAM instance role when you *create* a new instance. You can't assign a role to an instance that is already running. To configure an existing instance to use an SSM-supported role, you must create an image of the instance, launch an instance from that image, and assign the IAM role as you launch the instance. For more information, see [Creating an Amazon EBS-Backed Windows AMI \(p. 77\)](#).

To create an instance that uses an SSM-supported role

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Select a supported [region](#).
3. Choose **Launch Instance** and select a Windows Server instance.
4. Choose your instance type and then choose **Next: Configure Instance Details**.
5. Beside **IAM role** choose **Create new IAM role**. The IAM console opens in a new tab.
 - a. Choose **Create New Role**.
 - b. In **Step 1: Set Role Name**, enter a name that identifies this role as a Run Command role.
 - c. In **Step 2: Select Role Type**, choose **Amazon EC2 Role for Simple Systems Manager**. The system skips **Step 3: Establish Trust** because this is a managed policy.
 - d. In **Step 4: Attach Policy**, choose **AmazonEC2RoleforSSM**.
 - e. Choose **Next Step**, and then choose **Create Role**.
 - f. Close the tab with the IAM console.
6. In the EC2 Management Console, choose the **Refresh** button beside **Create New IAM role**.
7. In the **IAM role** drop-down list, choose the role you just created.
8. Complete the wizard to create and launch the new instance.

Grant Your User Account Access to SSM

Use the following procedure to attach an the **AmazonSSMFullAccess** IAM policy to your user account. This policy grants you full access to SSM API actions.

To create an IAM policy for EC2 instances

1. Open the Identity and Access Management (IAM) console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Policies**. (If this is your first time using IAM, choose **Get Started**, and then choose **Create Policy**.)
3. In the **Filter** field, type AmazonSSMFullAccess and press Enter.
4. Select the checkbox next to **AmazonSSMFullAccess** and then choose **Policy Actions, Attach**.



5. On the **Attach Policy** page, choose your user account and then choose **Attach Policy**.

After you attach the policy, see if your instances are visible in the **Select Target instances** section of the EC2 console. If they are not visible, then one or more of the prerequisites have not been met. For more information, see [Systems Manager Prerequisites \(p. 394\)](#).

Check Instance Status Using the Health API

You can use the Amazon EC2 Health API to quickly determine the following information about Amazon EC2 instances:

- The version of the EC2Config service
- The status of one or more instances
- The operating system
- The status of the EC2Config service
- The last time the instance sent a heartbeat value

Use the following command to get status details about one or more instances:

```
Get-SSMInstanceInformation -InstanceInformationFilterList  
@{Key="InstanceIds";ValueSet="instance-ID", "instance-ID" }
```

Use the following command with no filters to see all instances registered to your account that are currently reporting an online status. Substitute the ValueSet="Online" with "ConnectionLost" or "Inactive" to view those statuses:

```
Get-SSMInstanceInformation -InstanceInformationFilterList  
@{Key="PingStatus";ValueSet="Online" }
```

Use the following command to see which instances are running the latest version of the EC2Config service. Substitute ValueSet="LATEST" with a specific version (for example, 3.0.54 or 3.10) to view those details:

```
Get-SSMInstanceInformation -InstanceInformationFilterList  
@{Key="AgentVersion";ValueSet="LATEST" }
```

Troubleshooting the EC2Config Service

If you experience problems executing commands on an instance there could be a problem with the EC2Config service. This service is responsible for processing the commands on the instance. If you changed the network configuration and your instances do not show up in the list, then you need to restart the EC2Config Windows service. For information about troubleshooting the EC2Config service, see [Troubleshooting \(p. 330\)](#).

Inventory Management

You can use Systems Manager Inventory to collect operating system (OS) and application metadata from your Amazon EC2 instances and your on-premises servers or virtual machines (VMs). You can query the metadata to quickly understand which instances are running the software and configurations required by your software policy, and which instances need to be updated.

Information for Linux Users

See [Inventory Management](#) in the *Amazon EC2 User Guide for Linux Instances*.

Getting Started with Inventory

To get started with Inventory, complete the following tasks.

Task	For More Information
Update the SSM Agent on your EC2 instances to the latest version.	Updating the SSM Agent Using Amazon EC2 Run Command (p. 470)
Configure your on-premises servers and VMs for Systems Manager. After you configure them, they are described as <i>managed instances</i> .	Setting Up Systems Manager in Hybrid Environments (p. 397)
Verify Systems Manager prerequisites.	Systems Manager Prerequisites (p. 394)

Service Limits

Inventory currently has the following service limits.

Resource	Limit
Inventory data collected per instance per call	1 MB When this limit is reached, no new inventory data will be collected for the instance. Inventory data previously collected is stored until the expiration.
Inventory data collected per instance per day	5 MB
Inventory data expiration	30 days If you terminate an instance, inventory data for that instance is deleted immediately. For running instances, inventory data older than 30 days is deleted. If you need to store inventory data longer than 30 days, you can use AWS Config to record history or periodically query and upload the data to an Amazon S3 bucket. For more information, see, Recording Amazon EC2 managed instance inventory in the <i>AWS Config Developer Guide</i> .

Contents

- [About Systems Manager Inventory \(p. 516\)](#)

- [Configuring Inventory Collection \(p. 517\)](#)
- [Querying Inventory Collection \(p. 518\)](#)
- [Systems Manager Inventory Manager Walkthrough \(p. 518\)](#)

About Systems Manager Inventory

When you configure Systems Manager Inventory, you specify the type of metadata to collect, the instances from where the metadata should be collected, and when Inventory should run. These configurations are saved with your AWS account as a State Manager association.

Note

Inventory only collects metadata. It does not collect any personal or proprietary data.

The following table describes the different parts of inventory collection in more detail.

Part	Details
Type of information to collect	<ul style="list-style-type: none">• Instance details, including system name, OS name, OS version, last boot, DNS, domain, workgroup, OS architecture, etc.• Network configuration details, including IP address, MAC address, DNS, gateway, and subnet mask.• Application details, including application names, publishers, and versions• AWS component details, including EC2 driver, agents, and versions.• Windows Server Update history.• Custom inventory details. Custom inventory is described in more detail later in this section.
Instances to collect information from	You can individually select instances or target groups of instances using EC2 tag.
When to collect information	You can specify a collection interval in terms of minutes, hours, days, and weeks. The shortest collection interval is every 30 minutes.

Depending on the amount of data collected, the system can take several minutes to report the data to the output you specified. After the information is collected, the metadata is sent over a secure HTTPS channel to a plain-text AWS store that is accessible only from your AWS account. You can view the data in the Amazon S3 bucket you specified, or in the Amazon EC2 console on the **Inventory** tab for your managed instance. The **Inventory** tab includes several predefined filters to help you sort the data.

To start collecting inventory on your managed instance, see [Configuring Inventory Collection \(p. 517\)](#). To view samples of how to set up inventory collection using the Amazon EC2 console and the AWS CLI, see [Systems Manager Inventory Manager Walkthrough \(p. 518\)](#).

Custom Inventory

You can use the Systems Manager PutInventory API to attach metadata to your instances. For example, if you manage a large number of on-premises instances and you store information about rack location in a spreadsheet, then you could use the PutInventory API to write rack location metadata to each instance. The next time Inventory runs, if you select the option to collect **Custom Inventory**,

the system would collect the rack location for each instance and store this information with the other inventory data. For more information about the PutInventory API, see the [Amazon EC2 Simple Systems Manager API Reference](#).

Related AWS Services

Systems Manager Inventory provides a snapshot of your current inventory to help you manage software policy and improve the security posture of your entire fleet. You can extend your inventory management and migration capabilities using the following AWS services.

- AWS Config provides a historical record of changes to your inventory, along with the ability to create rules to generate notifications when a configuration item is changed. For more information, see, [Recording Amazon EC2 managed instance inventory](#) in the *AWS Config Developer Guide*.
- AWS Application Discovery Service is designed to collect inventory on OS type, application inventory, processes, connections, and server performance metrics from your on-premises VMs to support a successful migration to AWS. For more information, see the [Application Discovery Service User Guide](#).

Note

Each of these services uses its own agent.

Configuring Inventory Collection

Use the following procedure to configure inventory collection on a managed instance using the Amazon EC2 console. For an example of how to configure inventory collection using the AWS CLI, see [Systems Manager Inventory Manager Walkthrough \(p. 518\)](#).

Before you begin

Before you configure inventory collection, complete the following tasks.

- Verify that your instances meet Systems Manager prerequisites. For more information, see [Systems Manager Prerequisites \(p. 394\)](#).
- Update the SSM Agent if you plan to collect inventory from an existing instance. For more information, see [Updating the SSM Agent Using Amazon EC2 Run Command \(p. 470\)](#).

To configure inventory collection on a managed instance

1. Open the [Amazon EC2 console](#), expand **Systems Manager Shared Resources** in the navigation pane, and then choose **Managed Instances**.
2. Choose **Setup Inventory**.
3. In the **Targets** section, choose **Specify a Tag** if you want to configure inventory on multiple instances using EC2 tags. Choose **Manually Select Instances** if you want to individually choose which instances are configured for inventory.
4. In the **Schedule** section, choose how often you want the system to collect inventory metadata from your instances.
5. In the **Specify Parameters** section, use the lists to enable or disable different types of inventory collection.
6. In the **Specify Output Location** section, choose **Write to S3** if you want to store collected data in an Amazon S3 bucket.
7. Choose **Setup Inventory** and then choose **OK**.
8. In the **Managed Instances** page, choose an instance that you just configured for inventory and choose the **Description** tab. The **Association Status** shows **Pending** until the inventory

collection is processed. If the status showed **Failed**, verify that you have the latest version of the SSM Agent installed on your instances.

9. After the collection timeframe has passed, choose a managed instance, and then choose the **Inventory** tab.
10. Use the **Inventory Type** list to filter on different types of inventory data.

Querying Inventory Collection

After you collect inventory data, you can use the filter capability on the **Inventory** tab to filter on or filter out the instances you want.

To filter managed instance metadata

1. Open the [Amazon EC2 console](#), expand **Systems Manager Shared Resources** in the navigation pane, and then choose **Managed Instances**.
2. Choose the **Inventory** tab.
3. In the **Inventory Type** list, choose an attribute to filter on. For example: **AWS:Application**.
4. Choose the filter bar below the **Inventory Type** list to view a list of attributes on which to filter.
5. Choose a delimiter from the list. For example, choose **begins-with**.
6. Type a value. For example, type "ssm" and then choose the search icon at the left of the filter bar. The system returns all relevant managed instances.

Note

You can combine multiple filters to refine your search.

Systems Manager Inventory Manager Walkthrough

Use the following walkthrough to collect and manage inventory in a test environment.

Contents

- [Launch a New Instance](#) (p. 518)
- [Grant Your User Account Access to SSM](#) (p. 519)
- [Inventory Manager CLI Walkthrough](#) (p. 519)

Launch a New Instance

Instances require an AWS Identity and Access Management (IAM) role that enables the instance to communicate with Amazon EC2 Simple Systems Manager (SSM). You must assign the IAM role when you create the new instance. You can't assign a role to an instance that is already running. For existing instances, you must create an image of the instance, launch an instance from that image, and assign the IAM role as you launch the instance. For more information, see [Creating an Amazon EBS-Backed Windows AMI](#) (p. 77).

To create an instance that uses an SSM-supported role

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Select a supported [region](#).
3. Choose **Launch Instance** and select a Windows Server Amazon Machine Image (AMI).
4. Choose your instance type and then choose **Next: Configure Instance Details**.
5. In **Auto-assign Public IP**, choose **Enable**.
6. Beside **IAM role** choose **Create new IAM role**. The IAM console opens in a new tab.

- a. Choose **Create New Role**.
 - b. In **Step 1: Set Role Name**, enter a name that identifies this role as a Run Command role.
 - c. In **Step 2: Select Role Type**, choose **Amazon EC2 Role for Simple Systems Manager**. The system skips **Step 3: Establish Trust** because this is a managed policy.
 - d. In **Step 4: Attach Policy**, choose **AmazonEC2RoleforSSM**.
 - e. Choose **Next Step**, and then choose **Create Role**.
 - f. Close the tab with the IAM console.
7. In the Amazon EC2 console, choose the **Refresh** button beside **Create New IAM role**.
 8. From **IAM role**, choose the role you just created.
 9. Complete the wizard to launch the new instance. Make a note of the instance ID. You will need to specify this ID later in this tutorial.

Grant Your User Account Access to SSM

Your user account must be configured to communicate with the SSM API. Use the following procedure to attach a managed IAM policy to your user account that grants you full access to SSM API actions.

To create the IAM policy for your user account

1. Open the Identity and Access Management (IAM) console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Policies**. (If this is your first time using IAM, choose **Get Started**, and then choose **Create Policy**.)
3. In the **Filter** field, type **AmazonSSMFullAccess** and press Enter.
4. Select the check box next to **AmazonSSMFullAccess** and then choose **Policy Actions, Attach**.
5. On the **Attach Policy** page, choose your user account and then choose **Attach Policy**.

Inventory Manager CLI Walkthrough

The following procedure walks you through the process of using Inventory to collect metadata from the test instance you created earlier.

To gather inventory from an instance

1. Execute the following command to create a State Manager association that runs Inventory on the instance you created earlier. This command configures the service to run every six hours and to collect network configuration, Windows Update, and application metadata on the test instance you created earlier.

```
aws ssm create-association --name a name --targets
  Key=InstanceIds,Values=ID of the instance you created earlier --schedule-
expression "cron(0 0 0/6 1/1 * ? *)" --output-location "{ \"S3Location
\": { \"OutputS3Region\": \"us-east-1\", \"OutputS3BucketName\":
  \"Test bucket\", \"OutputS3KeyPrefix\": \"Test\" } }" --parameters
  networkConfig=Enabled,windowsUpdates=Enabled,applications=Enabled
```

The system responds with information like the following.

```
{
  "AssociationDescription": {
    "ScheduleExpression": "cron(0 0/30 * 1/1 * ? *)",
```

```
"OutputLocation": {
  "S3Location": {
    "OutputS3KeyPrefix": "Test",
    "OutputS3BucketName": "Test bucket",
    "OutputS3Region": "us-east-1"
  }
},
"Name": "The name you specified",
"Parameters": {
  "applications": [
    "Enabled"
  ],
  "networkConfig": [
    "Enabled"
  ],
  "windowsUpdates": [
    "Enabled"
  ]
},
"Overview": {
  "Status": "Pending",
  "DetailedStatus": "Creating"
},
"AssociationId":
"1a2b3c4d5e6f7g-1a2b3c-1a2b3c-1a2b3c-1a2b3c4d5e6f7g",
"DocumentVersion": "$DEFAULT",
"LastUpdateAssociationDate": 1480544990.06,
"Date": 1480544990.06,
"Targets": [
  {
    "Values": [
      "i-1a2b3c4d5e6f7g"
    ],
    "Key": "InstanceIds"
  }
]
}
```

You can target large groups of instances by using the `Targets` parameter with EC2 tags.

```
aws ssm create-association --name a name --targets
Key=tag:Environment,Values=Production --schedule-expression
"cron(0 0/30 * 1/1 * ? *)" --output-location "{ \"S3Location\":
{ \"OutputS3Region\": \"us-east-1\", \"OutputS3BucketName\":
\"Test bucket\", \"OutputS3KeyPrefix\": \"Test\" } }" --parameters
networkConfig=Enabled,windowsUpdates=Enabled,applications=Enabled
```

2. Execute the following command to view the association status.

```
aws ssm describe-instance-associations-status --instance-id ID of the
instance you created earlier
```

The system responds with information like the following.

```
{
  "InstanceAssociationStatusInfos": [
```

```
{
  "Status": "Pending",
  "DetailedStatus": "Associated",
  "Name": "reInvent2016PolicyDocumentTest",
  "InstanceId": "i-1a2b3c4d5e6f7g",
  "AssociationId":
"1a2b3c4d5e6f7g-1a2b3c-1a2b3c-1a2b3c-1a2b3c4d5e6f7g",
  "DocumentVersion": "1"
}
```

The following procedure walks you through the process of using the PutInventory API to assign custom metadata to the test instance you created earlier. This example assigns rack location information to a managed instance.

To assign custom metadata to an instance for Inventory

1. Execute the following command to assign rack location information to the test instance you created earlier.

```
aws ssm put-inventory --instance-id ID --items '[{"CaptureTime":
"2016-08-22T10:01:01Z", "TypeName": "Custom:RackInfo", "Content":
[{"RackLocation": "Bay B/Row C/Rack D/Shelf E"}], "SchemaVersion":
"1.0"}]'
```

2. Execute the following command to view custom inventory entries for this instance.

```
aws ssm list-inventory-entries --instance-id ID --type-name
Custom:RackInfo
```

The system responds with information like the following.

```
{
  "InstanceId": "ID",
  "TypeName": "Custom:RackInfo",
  "Entries": [
    {
      "RackLocation": "Bay B/Row C/Rack D/Shelf E"
    }
  ],
  "SchemaVersion": "1.0",
  "CaptureTime": "2016-08-22T10:01:01Z"
}
```

3. Execute the following command to view the custom metadata.

```
aws ssm get-inventory
```

The system responds with information like the following.

```
{
  "Entities": [
    {
      "Data": {
```



```
    "AWS:InstanceInformation": {  
      "Content": [  
        {  
          "ComputerName": "WIN-9JHCEPEGORG.WORKGROUP",  
          "InstanceId": "ID",  
          "ResourceType": "EC2Instance",  
          "AgentVersion": "3.19.1153",  
          "PlatformVersion": "6.3.9600",  
          "PlatformName": "Windows Server 2012 R2  
Standard",  
          "PlatformType": "Windows"  
        },  
        ],  
      "TypeName": "AWS:InstanceInformation",  
      "SchemaVersion": "1.0"  
    },  
    "Id": "ID"  
  ],  
}
```

State Management

Systems Manager State Manager is a secure and scalable service that automates the process of keeping your Amazon EC2 and hybrid infrastructure in a state that you define. You can use State Manager to ensure that your instances are bootstrapped with specific software at startup, configured according to your security policy, joined to a Windows domain, or patched with specific software updates throughout their lifecycle. You can also use State Manager to execute Linux shell scripts or Windows PowerShell scripts at different times during the lifecycle of an instance.

State Manager integrates with AWS CloudTrail to keep an audit trail of all association executions.

Information for Linux Users

See [State Management](#) in the *Amazon EC2 User Guide for Linux Instances*.

How It Works

You start by specifying the state you want to apply to your managed instances (for example, applications to bootstrap or network settings to configure) in a Systems Manager command or policy document. These documents are written in JSON and are called simply *documents*. Next, you bind the document to targets by using the AWS CLI or the Amazon EC2 console. You can target instance IDs or EC2 tags. The binding of the document to a target is called an association. After you associate your instance with a specific policy document, the instance remains in the state that you want because State Manager reapplies the state defined in the associated document according to the schedule that you define.

Getting Started with State Manager

To get started with State Manager, complete the following tasks.

Task	For More Information
Update the SSM Agent on your EC2 instances to the latest version.	Updating the SSM Agent Using Amazon EC2 Run Command (p. 470)

Task	For More Information
Configure your on-premises servers and VMs for Systems Manager. After you configure them, they are described as <i>managed instances</i> .	Setting Up Systems Manager in Hybrid Environments (p. 397)
Verify Systems Manager prerequisites.	Systems Manager Prerequisites (p. 394)
Create a policy document that defines the actions to perform on your instances.	Creating a Document for State Manager (p. 523)
Create and apply the association to your instances.	About State Manager Associations (p. 526)

Contents

- [Creating a Document for State Manager \(p. 523\)](#)
- [About State Manager Associations \(p. 526\)](#)
- [Systems Manager State Manager Walkthroughs \(p. 527\)](#)

Creating a Document for State Manager

When you create a State Manager association, the system reads the actions to be performed from either a policy or a command document. These documents are written in JSON, and they define the steps to run and parameters to use for your associations. The first time you create an association from a new policy document, the system stores the document with your AWS account.

The following is a sample of a basic policy document that defines the schema to use. It also defines a main step that uses the `aws:runPowerShellScript` plugin to get information about a process. A policy document can have multiple steps.

```
{
  "schemaVersion": "2.0",
  "$schema": "http://amazonaws.com/schemas/ec2/v3-0/runcommand#",
  "description": "Sample version 2.0 document v2",
  "parameters": {
  },
  "mainSteps": [
    {
      "action": "aws:runPowerShellScript",
      "name": "runPowerShellScript",
      "inputs": {
        "commands": [
          "Get-Process"
        ]
      }
    }
  ]
}
```

This sample includes two actions.

```
{
  "schemaVersion": "2.0",
  "$schema": "http://amazonaws.com/schemas/ec2/v3-0/runcommand#",
```

```
"description": "Sample version 2.0 document v2",
"parameters": {
},
"mainSteps": [
  {
    "action": "aws:runPowerShellScript",
    "name": "runShellScript",
    "inputs": {
      "commands": [
        "ipconfig"
      ]
    }
  },
  {
    "action": "aws:applications",
    "name": "installapp",
    "inputs": {
      "action": "Install",
      "source": "http://dev.mysql.com/get/Downloads/MySQLInstaller/
mysql-installer-community-5.6.22.0.msi"
    }
  }
]
}
```

Limitations

As you begin working with SSM documents for Systems Manager, be aware of the following limitations.

- You can create a maximum of 200 SSM documents per AWS account.
- SSM documents that you create are only available in the region where you created them. To add a document in another region, copy the content and recreate it in the new region.

Note

You currently can't use the same plugin twice in a policy document.

To create a policy document using the Amazon EC2 console

1. Open the [Amazon EC2 console](#) and choose **Systems Manager Shared Resources** in the navigation pane.
2. Choose **Documents** and then choose **Create Document**.
3. Enter a descriptive name for the document, choose **Policy** from the **Document Type** list, and then, in the **Content** field, specify plugins in JSON format. For more information, see [SSM Plugins](#) in the *Amazon EC2 Simple Systems Manager API Reference*.
4. Choose **Create Document** to save it with your AWS user account.

After you create your document, associate it with your instances. For more information, see [About State Manager Associations \(p. 526\)](#).

To create a policy document using Windows PowerShell

1. Copy a sample policy document and paste it into a simple text editor like Notepad.
2. Specify plugins and parameters in the file. Save the document with a descriptive name and a `.json` file extension. For more information, see [SSM Plugins](#) in the *Amazon EC2 Simple Systems Manager API Reference*.

3. Execute the following command to create the document and save it with your AWS user account using AWS Tools for Windows PowerShell.

```
$json = Get-Content C:\your file | Out-String  
New-SSMDocument -Name document name -Content $json
```

After you create your document, associate it with your instances. For more information, see [About State Manager Associations \(p. 526\)](#).

To create a policy document using the AWS CLI

1. Copy a sample policy document and paste it into a simple text editor like Notepad.
2. Specify plugins and parameters in a file. Save the document with a descriptive name and a `.json` file extension. For more information, see [SSM Plugins](#) in the *Amazon EC2 Simple Systems Manager API Reference*.
3. Execute the following command to create the document and save it with your AWS user account using the AWS CLI.

```
aws ssm create-document --content file://c:\temp\your file --name  
"document name"
```

After you create your document, associate it with your instances. For more information, see [About State Manager Associations \(p. 526\)](#).

About Document Versions and Execution

You can create and save different versions of policy documents. You can then specify default version and change the default version as you create newer documents. You can also revert to an earlier version. If you change the default version of a document, any association that uses the document will start using the new default version the next time Systems Manager applies the association to the instance.

When you change the JSON content of a document, State Manager automatically increments the version of the document. You can retrieve and view previous versions of the document. Documents can be associated with either instances or tagged groups.

Also note the following details about policy documents.

- You can assign multiple policy documents to a target by creating different associations that use different policy documents.
- If you associate multiple documents to a target, you can use the AWS CLI or SDK to view a consolidated list of plugins that will be executed across all associated documents.
- The order in which steps are specified in a document is the order in which they will be executed.
- You can use a *shared* policy document with State Manager, as long as you have permission, but you can't associate a shared document to an instance. If you want to use or share a document that is associated with one or more targets, you must create a copy of the document and then use or share it.
- If you create a policy document with conflicting plugins (e.g., domain join and remove from domain), the last plugin executed will be the final state. State Manager does not validate the logical sequence or rationality of the commands or plugins in your policy document.
- When processing policy documents, instance associations are applied first, and next tagged group associations are applied. If an instance is part of multiple tagged groups, then the documents that are part of the tagged group will not be executed in any particular order. If an instance is directly targeted through multiple documents by its instance ID, there is no particular order of execution.

About State Manager Associations

After you define the actions to perform on your instances in a policy document, you create an association. An association binds a policy document and one or more targets. Any actions defined in the document will be applied to instances when the association runs. You can create an association using the Amazon EC2 console, the AWS CLI, AWS Tools for Windows PowerShell, or the AWS SDKs. For examples of how to create and use associations using the Amazon EC2 console and the AWS CLI, see [Systems Manager State Manager Walkthroughs \(p. 527\)](#).

When you create an association, specify the following items.

- A policy document to use.
- The instances that should be associated with the policy document. You choose instances by manually selecting them, or by using the Targets option, which locates instances using EC2 tags.
- A schedule, which specifies how often the association should run.
- Parameters to execute when applying the association.
- An Amazon S3 bucket where the output should be written.

Scheduling and Running Associations

You can run the tasks of an association on demand or set a schedule when the Association should be reapplied. If you set a schedule, you can still run the association on demand.

Note

If a new association is scheduled to run while an earlier association is still running, the earlier association will be timed out and the new association will execute.

Your instances are accessible while associations are running.

Creating Associations Using the Targets Parameter

You can create associations on tens, hundreds, or thousands of instances by using the `targets` parameter. The `targets` parameter accepts a `Key:Value` combination based on Amazon EC2 tags that you specified for your instances. When you execute the request to create the association, the system locates and attempts to create the association on all instances that match the specified criteria. For more information about the `targets` parameter, see [Sending a Command to Multiple Instances \(p. 476\)](#). For more information about Amazon EC2 tags, see [Tagging Your Amazon EC2 Resources \(p. 859\)](#).

The following AWS CLI examples show you how to use the `targets` parameter when creating associations. The example commands have been truncated using [...].

Create an association for all the database servers (hosts with a tag named "Database" regardless of tag value).

```
aws ssm create-association --document-name value --targets  
"Key=tag:Database" [...]
```

Create an association for a managed instance named "ws-0123456789012345"

```
aws ssm create-association --document-name value --targets "Key=Instance  
Ids:Values=ws-0123456789" } [...]
```

Note

If you remove an instance from a tagged group that's associated with a document, then the instance will be dissociated from the document.

Systems Manager State Manager Walkthroughs

Use the following walkthroughs to manage the state of an EC2 instance in a test environment.

Contents

- [Launch a New Instance](#) (p. 527)
- [Grant Your User Account Access to SSM](#) (p. 527)
- [Systems Manager State Manager Console Walkthrough](#) (p. 528)
- [Systems Manager State Manager CLI Walkthrough](#) (p. 529)

Launch a New Instance

Instances require an AWS Identity and Access Management (IAM) role that enables the instance to communicate with State Manager (SSM). You must assign the IAM role when you create the new instance. You can't assign a role to an instance that is already running. For existing instances, you must create an image of the instance, launch an instance from that image, and assign the IAM role as you launch the instance. For more information, see [Creating an Amazon EBS-Backed Windows AMI](#) (p. 77).

To create an instance that uses an SSM-supported role

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Select a supported [region](#).
3. Choose **Launch Instance** and select a Windows Server Amazon Machine Image (AMI).
4. Choose your instance type and then choose **Next: Configure Instance Details**.
5. In **Auto-assign Public IP**, choose **Enable**.
6. Beside **IAM role** choose **Create new IAM role**. The IAM console opens in a new tab.
 - a. Choose **Create New Role**.
 - b. In **Step 1: Set Role Name**, enter a name that identifies this role as a Systems Manager role.
 - c. In **Step 2: Select Role Type**, choose **Amazon EC2 Role for Simple Systems Manager**. The system skips **Step 3: Establish Trust** because this is a managed policy.
 - d. In **Step 4: Attach Policy**, choose **AmazonEC2RoleforSSM**.
 - e. Choose **Next Step**, and then choose **Create Role**.
 - f. Close the tab with the IAM console.
7. In the Amazon EC2 console, choose the **Refresh** button beside **Create New IAM role**.
8. From **IAM role**, choose the role you just created.
9. Complete the wizard to launch the new instance. Make a note of the instance ID. You will need to specify this ID later in this tutorial.

Grant Your User Account Access to SSM

Your user account must be configured to communicate with the SSM API. Use the following procedure to attach a managed IAM policy to your user account that grants you full access to SSM API actions.

To create the IAM policy for your user account

1. Open the Identity and Access Management (IAM) console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Policies**. (If this is your first time using IAM, choose **Get Started**, and then choose **Create Policy**.)

3. In the **Filter** field, type `AmazonSSMFullAccess` and press Enter.
4. Select the check box next to **AmazonSSMFullAccess** and then choose **Policy Actions, Attach**.
5. On the **Attach Policy** page, choose your user account and then choose **Attach Policy**.

Systems Manager State Manager Console Walkthrough

The following procedure walks you through the process of creating an association using the EC2 console.

To create an association using State Manager

1. Open the [Amazon EC2 console](#) and choose **Systems Manager Shared Resources** in the navigation pane.
2. Choose **Documents** and then choose **Create Document**.
3. For **Name**, type a descriptive name that identifies this document as a test policy document.
4. In the **Document type** list, choose **Command**.
5. Delete the pre-populated brackets `{}` in the **Content field** and then copy and paste the following sample document in the **Content** field.

```
{
  "schemaVersion": "2.0",
  "$schema": "http://amazonaws.com/schemas/ec2/v3-0/runcommand#",
  "description": "Sample version 2.0 document v2",
  "parameters": {
  },
  "mainSteps": [
    {
      "action": "aws:runPowerShellScript",
      "name": "runShellScript",
      "inputs": {
        "commands": [
          "ipconfig"
        ]
      }
    },
    {
      "action": "aws:applications",
      "name": "installapp",
      "inputs": {
        "action": "Install",
        "source": "http://dev.mysql.com/get/Downloads/MySQLInstaller/mysql-installer-community-5.6.22.0.msi"
      }
    }
  ]
}
```

6. Choose **Create document**, and then choose **OK** after the system creates the policy document.
7. In the EC2 console navigation pane, expand **Systems Manager Services**, and then choose **State Manager**.
8. Choose **Create Association**.
9. In the **Document name** list, choose the document you just created.
10. In the **Select Targets by** section, choose **Manually Selecting Instances**, and then choose the instance you created at the beginning of this walkthrough.

11. In the **Schedule** section, choose an option.
12. Disregard the **Specify Parameters** section, as the test policy document does not take parameters.
13. Choose **Create Association**.

Systems Manager State Manager CLI Walkthrough

The following procedure walks you through the process of creating an association using the AWS Command Line Interface (AWS CLI).

1. Copy the following sample policy document and paste it into a simple text editor like Notepad.

```
{
  "schemaVersion": "2.0",
  "$schema": "http://amazonaws.com/schemas/ec2/v3-0/runcommand#",
  "description": "Sample version 2.0 document v2",
  "parameters": {
  },
  "mainSteps": [
    {
      "action": "aws:runPowerShellScript",
      "name": "runShellScript",
      "inputs": {
        "commands": [
          "ipconfig"
        ]
      }
    },
    {
      "action": "aws:applications",
      "name": "installapp",
      "inputs": {
        "action": "Install",
        "source": "http://dev.mysql.com/get/Downloads/MySQLInstaller/mysql-installer-community-5.6.22.0.msi"
      }
    }
  ]
}
```

2. Save the document with a descriptive name and a `.json` file extension.
3. Execute the following command to create the document and save it with your AWS user account using the AWS CLI.

```
aws ssm create-document --content file://c:\temp\your file --name "document name"
```

4. Execute the following command to create an association with the instance you created at the start of this walkthrough. The `Schedule` parameter sets a schedule to run the association every 30 minutes.

```
aws ssm create-association --targets Key=instanceids,Values=Instance ID --document your document name --schedule "cron(0 0/30 * 1/1 * ? *)"
```

5. Execute the following command to view the associations for the instance. Copy the association ID returned by the command. You'll specify this ID in the next step.


```
aws ssm list-instance-associations --instance-id=Instance ID
```

Maintenance and Deployment Automation

Systems Manager Automation is an AWS-hosted automation service that simplifies common maintenance and deployment tasks. You can use EC2 Automation to execute custom scripts, perform patch updates, and update Amazon Machine Images (AMIs), drivers, agents, and applications.

Information for Linux Users

See [Maintenance and Deployment Automation](#) in the *Amazon EC2 User Guide for Linux Instances*.

To get started with Automation, first configure access, and then try the service in a test environment using the Amazon EC2 console walkthrough or the AWS CLI walkthrough.

Contents

- [Configuring Access to Automation](#) (p. 530)
- [Systems Manager Automation Walkthroughs](#) (p. 532)
- [Actions Reference for Automation Documents](#) (p. 540)
- [Automation System Variables](#) (p. 552)

Configuring Access to Automation

Use the following procedures to configure security roles and permissions for EC2 Automation. After you configure roles and permissions, you can perform a test run with Automation as described in [Systems Manager Automation Walkthroughs](#) (p. 532).

Create an IAM Role for Automation

Systems Manager Automation needs to have permission to perform the actions that you specify for the service. It obtains these permissions by assuming your IAM role. Use the following procedures to:

- Create a role so that Automation can act on your behalf when processing workflows.
- Assign permissions to the role so that you can reference IAM roles within an Automation document.

To create an IAM role and allow Automation to assume it

1. Open the Identity and Access Management (IAM) console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Roles**, and then choose **Create New Role**.
3. In **Step 1: Set Role Name**, enter a name that identifies this role as an Automation role.
4. In **Step 2: Select Role Type**, choose **Amazon EC2**. The system skips **Step 3: Establish Trust** because this is a managed policy.
5. In **Step 4: Attach Policy**, choose the **AmazonSSMAutomationRole** managed policy. They provide the same access permissions.
6. In **Step 5: Review**, make a note of the **Role Name** and **Role ARN**. You will specify the role ARN when you attach the iam:PassRole policy to your IAM account in the next procedure. You will also specify the role name and the ARN in EC2 Automation documents.
7. Choose **Create Role**. The system returns you to the **Roles** page.

8. Locate the role you just created and double-click it.
9. Choose the **Trust Relationships** tab, and then choose **Edit Trust Relationship**.
10. Using the following code snippet as an example, add a comma after "ec2.amazonaws.com", and then add "Service": "ssm.amazonaws.com" to the existing policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com",
        "Service": "ssm.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

11. Choose **Update Trust Policy**.
12. Copy or make a note of the **Role ARN**. You will specify this ARN in your automation document.

Note

The AmazonSSMAutomationRole policy assigns the Automation role permission to a subset of AWS Lambda functions within your account (functions beginning with Automation). If you plan to use Automation with Lambda functions, the Lambda ARN must use the following format:

```
"arn:aws:lambda:*:*:function:Automation*"
```

If the ARN doesn't use this format, then you must attach an additional Lambda policy to the AmazonSSMAutomationRole role or an additional Lambda role, such as AWSLambdaRole, to the AWS account. The additional policy or role must provide broader access to Lambda functions within the AWS account.

Attach the iam:PassRole Policy to Your Role

Use the following procedure to attach the iam:PassRole policy to the role you just created.

To attach the iam:PassRole policy to your role

1. Locate the role you just created and double-click it.
2. In the **Inline Policies** section, choose **Create User Policy**. If you don't see this button, choose the down arrow beside **Inline Policies**, and then choose **click here**.
3. On the **Set Permissions** page, choose **Policy Generator**, and then choose **Select**.
4. Verify that **Effect** is set to **Allow**.
5. From **AWS Services**, choose **AWS Identity and Access Management**.
6. From **Actions**, choose **PassRole**.
7. In the **Amazon Resource Name (ARN)** field, paste the role ARN you created in the previous procedure.
8. Choose **Add Statement**, and then choose **Next Step**.
9. On the **Review Policy** page, choose **Apply Policy**.

Important

Repeat this procedure for any role referenced in your Automation documents. For example, repeat this procedure if you want to specify an InstanceProfile role in the `aws:launchInstance` step of your Automation document. Alternatively, you can repeat this procedure and specify an asterisk (*) for the ARN. By specifying the asterisk, you allow Systems Manager to reference all roles within your AWS account.

Configure Account Permissions

Use the following procedure to attach the `iam:PassRole` policy to your user account. Or, you can create a new IAM account and attach the `iam:PassRole` policy to it. If you create a new account, you must also attach the **AmazonSSMFullAccess** policy so the account can communicate with the SSM API. If you need to create a new user account, see [Creating an IAM User in Your AWS Account](#) in the *IAM User Guide*.

To attach the `iam:PassRole` policy to your user account

1. In the IAM navigation pane, choose **Users**, and then double-click your user account.
2. In the **Managed Policies** section, verify that either the **AmazonSSMFullAccess** policy is listed or there is a comparable policy that gives you permissions for the SSM API.
3. In the **Inline Policies** section, choose **Create User Policy**. If you don't see this button, choose the down arrow beside **Inline Policies**, and then choose **click here**.
4. On the **Set Permissions** page, choose **Policy Generator**, and then choose **Select**.
5. Verify that **Effect** is set to **Allow**.
6. From **AWS Services**, choose **AWS Identity and Access Management**.
7. From **Actions**, choose **PassRole**.
8. In the **Amazon Resource Name (ARN)** field, paste the role ARN you created in the previous procedure.
9. Choose **Add Statement**, and then choose **Next Step**.
10. On the **Review Policy** page, choose **Apply Policy**.

Systems Manager Automation Walkthroughs

The following walkthroughs show you how to execute an Automation workflow either from the EC2 console or the AWS CLI. Before you use these walkthrough, you must configure Automation roles and permissions. For more information, see [Configuring Access to Automation](#) (p. 530).

Contents

- [Systems Manager Automation Console Walkthrough](#) (p. 532)
- [Systems Manager Automation CLI Walkthrough](#) (p. 535)

Systems Manager Automation Console Walkthrough

The following walkthrough uses a sample automation document to perform the following tasks using Automation:

- Launch an Windows instance from a specified AMI.
- Execute a command using Run Command that applies Windows updates to the instance.
- Stop the instance.
- Create a new Windows AMI.
- Terminate the original instance.

In this walkthrough, you will configure and execute the Automation workflow by using the EC2 console.

Automation Sample Document

Automation executes automation documents written in JavaScript Object Notation (JSON). Automation documents include the actions to be performed during workflow execution. The following list shows the supported actions:

- **aws:runInstance**: Launches one or more instances for a given AMI ID.
- **aws:runCommand**: Remote command execution. Executes an SSM Run Command document.
- **aws:invokeLambdaFunction**: Enables you to run external worker functions in your automation workflow.
- **aws:changeInstanceState**: Changes an instance state to `stopped`, `terminated`, or `running`.
- **aws:createImage**: Creates an AMI from a running instance.
- **DeleteImage**: Deletes an AMI.

You can view these actions in the sample Automation document in the following procedure.

To create a patched AMI using Automation

1. Open the [Amazon EC2 console](#), expand **Commands** in the navigation pane, and then choose **Documents**.
2. Choose **Create document**.
3. For **Name**, type `patchWindowsAmi`.
4. In the **Document type** list, choose **Automation**.
5. Delete the pre-populated brackets `{}` in the **Content field** and then copy and paste the following sample document the **Content field**. Change the value of `assumeRole` to the role ARN you created earlier and change the value of `IamInstanceProfileName` to the name of the role you created earlier when you created an IAM role for Automation.

```
{
  "description": "Systems Manager Automation Demo - Patch and Create a New AMI",
  "schemaVersion": "0.3",
  "assumeRole": "the role ARN you created",
  "parameters": {
    "sourceAMIId": {
      "type": "String",
      "description": "AMI to patch",
      "default": "{{ssm:sourceAMI}}"
    },
    "targetAMIName": {
      "type": "String",
      "description": "Name of new AMI",
      "default": "patchedAMI-{{global:DATE_TIME}}"
    }
  },
  "mainSteps": [
    {
      "name": "startInstances",
      "action": "aws:runInstances",
      "timeoutSeconds": 1200,
      "maxAttempts": 1,
      "onFailure": "Abort",
      "inputs": {
        "ImageId": "{{ sourceAMIId }}"
      }
    }
  ]
}
```

```
        "InstanceType": "m3.large",
        "MinInstanceCount": 1,
        "MaxInstanceCount": 1,
        "IamInstanceProfileName": "the name of the IAM role you
created"
    },
    {
        "name": "installMissingWindowsUpdates",
        "action": "aws:runCommand",
        "maxAttempts": 1,
        "onFailure": "Continue",
        "inputs": {
            "DocumentName": "AWS-InstallMissingWindowsUpdates",
            "InstanceIds": [
                "{{ startInstances.InstanceIds }}"
            ],
            "Parameters": {
                "UpdateLevel": "Important"
            }
        }
    },
    {
        "name": "stopInstance",
        "action": "aws:changeInstanceState",
        "maxAttempts": 1,
        "onFailure": "Continue",
        "inputs": {
            "InstanceIds": [
                "{{ startInstances.InstanceIds }}"
            ],
            "DesiredState": "stopped"
        }
    },
    {
        "name": "createImage",
        "action": "aws:createImage",
        "maxAttempts": 1,
        "onFailure": "Continue",
        "inputs": {
            "InstanceId": "{{ startInstances.InstanceIds }}",
            "ImageName": "{{ targetAMIname }}",
            "NoReboot": true,
            "ImageDescription": "AMI created by EC2 Automation"
        }
    },
    {
        "name": "terminateInstance",
        "action": "aws:changeInstanceState",
        "maxAttempts": 1,
        "onFailure": "Continue",
        "inputs": {
            "InstanceIds": [
                "{{ startInstances.InstanceIds }}"
            ],
            "DesiredState": "terminated"
        }
    }
],
```

```
"outputs": [
  "createImage.ImageId"
]
}
```

About

6. Choose **Create document** and then choose **OK** after the system creates the automation document.
7. In the EC2 console navigation pane, expand **Systems Manager Services** and then choose **Automation executions**.
8. Choose **Run automation document**.
9. In the **Document name** list, choose the document you just created.
10. In the **Value** field for **sourceAMId**, type in the ID of the AMI you'd like to patch, or use the default ID provided.
11. In the **Value** field for **targetAMIname**, type in the name of the your AMI, or use the default value which will append a timestamp.
12. Choose **Run automation**. The system displays an automation execution ID. Choose **OK**.
13. In the execution list, choose the execution you just ran and then choose the **Steps** tab. This tab shows you the status of the workflow actions.

Note

Depending on the number of patches applied, the Windows patching process executed in this sample workflow can take 30 minutes or more to complete.

Systems Manager Automation CLI Walkthrough

The following walkthrough uses a sample automation document to perform the following tasks using Automation:

- Launch an Windows instance from a specified AMI.
- Execute a command using Run Command that applies Windows updates to the instance.
- Stop the instance.
- Create a new Windows AMI.
- Terminate the original instance.

In this walkthrough, you will configure and execute the Automation workflow by using the AWS CLI.

Automation Sample Document

Automation executes automation documents written in JavaScript Object Notation (JSON). Automation documents include the actions to be performed during workflow execution. The following list shows the supported actions:

- **aws:runInstance**: Launches one or more instances for a given AMI ID.
- **aws:runCommand**: Remote command execution. Executes an SSM Run Command document.
- **aws:invokeLambdaFunction**: Enables you to run external worker functions in your automation workflow.
- **aws:changeInstanceState**: Changes an instance state to `stopped`, `terminated` or `running`.
- **aws:createImage**: Creates an AMI from a running instance.
- **DeleteImage**: Deletes an AMI.

You can view these actions in the sample Automation document in the following procedure.

To create a patched AMI using Automation

1. Copy the following example document into a text editor such as Notepad. Change the value of `assumeRole` to the role ARN you created earlier when you created an IAM role for Automation and change the value of `IamInstanceProfileName` to the name of the role you created earlier. Save the document on a local drive as `patchWindowsAmi.json`.

```
{
  "description": "Automation Demo - Patch and Create a New AMI",
  "schemaVersion": "0.3",
  "assumeRole": "the role ARN you created",
  "parameters": {
    "sourceAMIId": {
      "type": "String",
      "description": "AMI to patch",
      "default": "ami-3f0c4628"
    },
    "targetAMIname": {
      "type": "String",
      "description": "Name of new AMI",
      "default": "patchedAMI-{{global:DATE_TIME}}"
    }
  },
  "mainSteps": [
    {
      "name": "startInstances",
      "action": "aws:runInstances",
      "timeoutSeconds": 1200,
      "maxAttempts": 1,
      "onFailure": "Abort",
      "inputs": {
        "ImageId": "{{ sourceAMIId }}",
        "InstanceType": "m3.large",
        "MinInstanceCount": 1,
        "MaxInstanceCount": 1,
        "IamInstanceProfileName": "the name of the IAM role you created"
      }
    },
    {
      "name": "installMissingWindowsUpdates",
      "action": "aws:runCommand",
      "maxAttempts": 1,
      "onFailure": "Abort",
      "inputs": {
        "DocumentName": "AWS-InstallMissingWindowsUpdates",
        "InstanceIds": [
          "{{ startInstances.InstanceIds }}"
        ],
        "Parameters": {
          "UpdateLevel": "Important"
        }
      }
    },
    {
      "name": "stopInstance",
      "action": "aws:changeInstanceState",
```

```

    "maxAttempts":1,
    "onFailure":"Abort",
    "inputs":{
      "InstanceIds":[
        "{{ startInstances.InstanceIds }}"
      ],
      "DesiredState":"stopped"
    }
  },
  {
    "name":"createImage",
    "action":"aws:createImage",
    "maxAttempts":1,
    "onFailure":"Abort",
    "inputs":{
      "InstanceId":"{{ startInstances.InstanceIds }}",
      "ImageName":"{{ targetAMIname }}",
      "NoReboot":true,
      "ImageDescription":"AMI created by Automation"
    }
  },
  {
    "name":"terminateInstance",
    "action":"aws:changeInstanceState",
    "maxAttempts":1,
    "onFailure":"Abort",
    "inputs":{
      "InstanceIds":[
        "{{ startInstances.InstanceIds }}"
      ],
      "DesiredState":"terminated"
    }
  }
],
"outputs":[
  "createImage.ImageId"
]
}

```

2. [Download](#) the AWS CLI to your local machine.
3. Edit the following command, and specify the path to the patchWindowsAmi.json file on your local machine. Execute the command to create the required Automation document.

```
aws ssm create-document --name "patchWindowsAmi" --content file:///Users/test-user/Documents/patchWindowsAmi.json --document-type Automation
```

The system returns information about the command progress.

```

{
  "DocumentDescription": {
    "Status": "Creating",
    "Hash":
    "fdeacee0a97ea710f6efc1dc43a9025ce3a34c5743a11e30a4ac6f42127f3a8e",
    "Name": "patchWindowsAmi",
    "DocumentType": "Ec2Automation",
    "PlatformTypes": [],
    "DocumentVersion": "1",

```



```
    "HashType": "Sha256",  
    "CreateDate": 1475776111.117,  
    "Owner": "860547654709"  
  }  
}
```

4. Execute the following command to view a list of documents that you can access.

```
aws ssm list-documents --document-filter-list key=Owner,value=Self
```

The system returns information like the following:

```
{  
  "DocumentIdentifiers": [  
    {  
      "Name": " patchWindowsAmi",  
      "PlatformTypes": [  
  
      ],  
      "DocumentVersion": "5",  
      "DocumentType": "Automation",  
      "Owner": "12345678901",  
      "SchemaVersion": "0.3"  
    }  
  ]  
}
```

5. Execute the following command to view details about the patchWindowsAmi document.

```
aws ssm describe-document --name patchWindowsAmi
```

The system returns information like the following:

```
{  
  "Document": {  
    "Status": "Active",  
  
    "Hash": "99d5b2e33571a6bb52c629283bca0a164026cd201876adf0a76de16766fb98ac",  
    "Name": "patchWindowsAmi",  
    "Parameters": [  
      {  
        "DefaultValue": "ami-3f0c4628",  
        "Type": "String",  
        "Name": "sourceAMIid",  
        "Description": "AMI to patch"  
      },  
      {  
        "DefaultValue": "patchedAMI-{{global:DATE_TIME}}",  
        "Type": "String",  
        "Name": "targetAMIname",  
        "Description": "Name of new AMI"  
      }  
    ],  
    "DocumentType": "Automation",  
    "PlatformTypes": [  
  
    ],  
  }  
}
```

```
"DocumentVersion": "5",  
"HashType": "Sha256",  
"CreateDate": 1478904417.477,  
"Owner": "12345678901",  
"SchemaVersion": "0.3",  
"DefaultVersion": "5",  
"LatestVersion": "5",  
"Description": "Automation Demo - Patch and Create a New AMI"  
}  
}
```

6. Execute the following command to run the patchWindowsAmi document and run the Automation workflow. This command takes two input parameters: the ID of the AMI to be patched, and the name of the new AMI. The example command below uses a recent EC2 AMI to minimize the number of patches that need to be applied. If you run this command more than once, you must specify a unique value for `targetAMIname`. AMI names must be unique.

```
aws ssm start-automation-execution --document-name="patchWindowsAmi" --  
parameters sourceAMIid="ami-bd3ba0aa"
```

The command returns an execution ID. Copy this ID to the clipboard. You will use this ID to view the status of the workflow.

```
{  
  "AutomationExecutionId": "ID"  
}
```

You can monitor the status of the workflow in the EC2 console. Check the console to verify that a new instance is launching. After the instance launch is complete, you can confirm that the Run Command action was executed by viewing the **Commands History** page. After Run Command execution is complete, you should see a new AMI in your list of AMIimages.

7. To view the workflow execution using the CLI, execute the following command:

```
aws ssm describe-automation-executions
```

8. To view details about the execution progress, execute the following command.

```
aws ssm get-automation-execution --automation-execution-id ID
```

Note

Depending on the number of patches applied, the Windows patching process executed in this sample workflow can take 30 minutes or more to complete.

Additional Automation Tasks

You can manage other aspects of Automation execution using the following tasks.

Stop an Execution

Execute the following to stop a workflow. The command doesn't terminate associated instances.

```
aws ssm stop-automation-execution --automation-execution-id ID
```

Create Versions of Automation Documents

You can't change an existing automation document, but you can create a new version using the following command:

```
aws ssm update-document --name "patchWindowsAmi" --content file:///Users/test-user/Documents/patchWindowsAmi.json --document-version "\$LATEST"
```

Execute the following command to view details about the existing document versions:

```
aws ssm list-document-versions --name "patchWindowsAmi"
```

The command returns information like the following:

```
{
  "DocumentVersions": [
    {
      "IsDefaultVersion": false,
      "Name": "patchWindowsAmi",
      "DocumentVersion": "2",
      "CreateDate": 1475799950.484
    },
    {
      "IsDefaultVersion": false,
      "Name": "patchWindowsAmi",
      "DocumentVersion": "1",
      "CreateDate": 1475799931.064
    }
  ]
}
```

Execute the following command to update the default version for execution. The default execution version only changes when you explicitly set it to a new version. Creating a new document version does not change the default version.

```
aws ssm update-document-default-version --name patchWindowsAmi --document-version 2
```

Delete a Document

Execute the following command to delete an automation document:

```
aws ssm delete-document --name patchWindowsAMI
```

Actions Reference for Automation Documents

Systems Manager Automation performs tasks defined in Automation documents. To define a task, you specify one or more of the following actions in any order in the MainSteps section of your Automation document.

- **aws:runInstance:** Launches one or more instances for a given AMI ID.
- **aws:runCommand:** Remote command execution. Executes an SSM Run Command document.
- **aws:invokeLambdaFunction:** Enables you to run external worker functions in your automation workflow.

- **aws:changeInstanceState**: Changes an instance state to stopped, terminated, or running.
- **aws:createImage**: Creates an AMI from a running instance.
- **aws:deleteImage**: Deletes an AMI.

All actions use the syntax shown later in this section. The properties specified outside the inputs section remain the same across all actions. You can use different values for any property in any action.

Common Properties In All Actions

```
name
```

A string specifying a unique identifier. The value of this property is expected to be unique across all step names in the document.

Required: Yes

```
action
```

A string specifying the name of the action a particular step intends to execute.

Required: Yes

```
maxAttempts
```

An integer value for number of times the step should be retried in case of failures. If the value specified is greater than one, the step is not declared failed until all attempts have failed. One is assumed as the default value.

Required: No

```
timeoutSeconds
```

An integer value for the step to timeout the execution.

Required: No

```
onFailure
```

A string indicating if the workflow should Abort or Continue on failures. By default the value Abort is assumed.

Required: No

```
inputs
```

Map of properties specific to the action

Required: Yes

Important

The outputs of an action are not supposed to be specified in the document. They are available to the user for linking the steps or adding into the output section of the doc. E.g. if you want

to make output of `aws:runInstances` step, i.e. the `instanceId` an input for the next action which is let's say `aws:runCommand`, you have it available to you. See the example below to understand more.

Syntax

```
"mainSteps": [
  {
    "name": "launchInstance",
    "action": "aws:runInstances",
    "maxAttempts": 3,
    "timeoutSeconds": 1200,
    "onFailure": "Abort",
    "inputs": {
      "ImageId": "ami-123456",
      "InstanceType": "t2.micro"
    }
  },
  {
    "name": "updateInstance",
    "action": "aws:runCommand",
    "timeoutSeconds": 1200,
    "onFailure": "Continue",
    "inputs": {
      "DocumentName": "AWS-RunShellScript",
      "InstanceIds": [
        "{{launchInstance.InstanceIds}}"
      ],
      "Parameters": {
        "commands": [
          "ls -l"
        ]
      }
    }
  }
]
```

Action `aws:runInstance`

You can use this action to launch anew instance. The action supports most run-instance API arguments.

JSON Sample

```
{
  "name": "launchInstance",
  "action": "aws:runInstances",
  "maxAttempts": 3,
  "timeoutSeconds": 1200,
  "onFailure": "Abort",
  "inputs": {
    "ImageId": "ami-123456",
    "InstanceType": "t2.micro",
    "MinInstanceCount": 1,
    "MaxInstanceCount": 1,
    "IamInstanceProfileName": "MyRunCmdRole"
  }
}
```

```
}
```

Inputs

The action supports most run-instances API's parameters. For specific permitted value, please refer to the run-instances api documentation.

```
ImageId
```

A string literal containing the id of the image to launch the instance.

Required: Yes

```
AdditionalInfo
```

A string containing additional info to launch the instance.

Required: No

```
BlockDeviceMappings
```

A map list containing the mappings for the instance.

Required: No

```
ClientToken
```

A string literal.

Required: No

```
DisableApiTermination
```

A boolean value.

Required: No

```
EbsOptimized
```

A boolean value.

Required: No

```
IamInstanceProfileArn
```

A string literal containing the ARN of.

Required: No

```
IamInstanceProfileName
```

A string literal containing name of the IAM profile to associate with the instance.

Required: No

```
InstanceInitiatedShutdownBehavior
```

A string value.

Required: No

```
InstanceType
```

A string literal containing the instance type.

Required: No

```
KernelId
```

A string value containing kernel id.

Required: No

```
KeyName
```

A string literal containing the name of the security key.

Required: No

```
MaxInstanceCount
```

An integer value for defining the maximum number of instances to be launched.

Required: No

```
MinInstanceCount
```

An integer value for defining the minimum number of instances to be launched.

Required: No

```
Monitoring
```

A boolean value to indicate enabling cloud watch monitoring.

Required: No

```
NetworkInterfaces
```

A list of maps containing all the network interfaces.

Required: No

```
Placement
```

A map of string literals.

Required: No

```
PrivateIpAddress
```

A string value containing the IP address.

Required: No

```
RamdiskId
```

A string literal containing ram disk id.

Required: No

```
SecurityGroupIds
```

A list of string literals containing the Ids of the security groups.

Required: No

```
SecurityGroups
```

A list of string literals containing the names of the security groups.

Required: No

```
SubnetId
```

A string value containing the subnet id.

Required: No

```
UserData
```

An execution script provided as a string literal value.

Required: No

Outputs

```
InstanceIds
```

List of string literals containing instance ids.

Action `aws:runCommand`

You can use this action to run any commands using send-command API. This action supports most send-command API arguments. The example below shows using only one to many public run command documents.

JSON Sample

```
{
  "name": "installPowerShellModule",
  "action": "aws:runCommand",
  "inputs": {
    "DocumentName": "AWS-InstallPowerShellModule",
    "InstanceIds": ["i-123456789"],
    "Parameters": {
      "source": "https://my-s3-url.com/MyModule.zip ",
      "sourceHash": "ASDFWER12321WRW"
    }
  }
}
```

Inputs

All the inputs listed here are simply the enumeration of all send-command parameters across all public documents. For specific public command document specific value, please refer to the send-command api documentation.

DocumentName

A string literal containing the name of the run command document.

Required: Yes

InstanceIds

A list of string literals containing the ids of the instances.

Required: Yes

Parameters

This is not a property. This is an additional section inside inputs section. It is a map of properties below.

Required: No

Comment

A string literal.

Required: No

DocumentHash

A string containing hash for the PowerShell module to be installed.

Required: No

DocumentHashType

A string containing type of the hash. Permitted values are Sha256 and Sha1.

Required: No

```
NotificationConfig
```

A map of string literals.

Required: No

```
OutputS3BucketName
```

A string literal.

Required: No

```
OutputS3KeyPrefix
```

A string literal.

Required: No

```
ServiceRoleArn
```

A string literal containing the ARN.

Required: No

```
TimeoutSeconds
```

An integer value to specify the run-command timeout seconds.

Required: No

Outputs

```
CommandId
```

String literal containing command id.

```
Output
```

String literal containing the truncated output of the command.

```
ResponseCode
```

String literals containing command status code.

```
Status
```

String literal indicating the status of the command.

Action `aws:invokeLambdaFunction`

You can use this action to invoke an existing Lambda function. Note that this action does not create the function if it does not exist.

JSON Sample

```
{
  "name": "invokeMyLambdaFunction",
  "action": "aws:invokeLambdaFunction",
  "maxAttempts": 3,
  "timeoutSeconds": 120,
  "onFailure": "Abort",
  "inputs": {
    "FunctionName": "MyLambdaFunction"
  }
}
```

Inputs

The action supports most invoke API's parameters for Lambda service. For specific permitted value, please refer to the invoke api documentation.

FunctionName

A string literal containing the id of the image to launch the instance.

Required: Yes

ClientContext

A string containing client context info.

Required: No

InvocationType

A string literal with permitted values RequestResponse or Event or DryRun.

Required: No

LogType

A string literal with permitted values None or Tail.

Required: No

Payload

A string literal.

Required: No

Qualifier

A string literal.

Required: No

Outputs

StatusCode

A string literals containing the function execution status code.

Action `aws:changeInstanceState`

You can use this action to either change or assert the state of the instance.

Important

This action can be used in assert mode i.e. not execute the start-instance/stop-instances/terminate-instances API to achieve the desired state, instead just validate that instance is desired state. The assert mode is activated by supplying parameter `CheckStateOnly` as *true*. This mode is very useful for customers executing Sysprep command in Windows AMIs. Sysprep is an asynchronous command. It runs in the background and can run for long time. So while its execution is incomplete if `aws:changeInstanceState` is executed without this flag `CheckStateOnly` set to true, when instance is stopped it will be in undesired state. The AMIs created from such instances may be defective.

JSON Sample

```
{
  "name": "stopMyInstance",
  "action": "aws:changeInstanceState",
  "maxAttempts": 3,
  "timeoutSeconds": 3600,
  "onFailure": "Abort",
  "inputs": {
    "InstanceIds": ["i-123456789"],
    "CheckStateOnly": true,
    "DesiredState": "stopped"
  }
}
```

Inputs

DesiredState

A string literal with permitted values running or stopped or terminated.

Required: Yes

InstanceIds

A list of string literals containing the ids of the instances.

Required: Yes

```
AdditionalInfo
```

A string literal.

Required: No

```
CheckStateOnly
```

This is a boolean literal. If value is false, it'll execute EC2 API like start-instance/stop-instances/terminate-instances to cause the desired state transition. If true, it only will assert desired state by polling for it.

Required: No

```
Force
```

A Boolean value. If set Forces the instances to stop. The instances do not have an opportunity to flush file system caches or file system metadata. If you use this option, you must perform file system check and repair procedures. This option is not recommended for Windows instances.

Required: No

Outputs

None.

Action `aws:createImage`

You can use this action to create a new image from a stopped instance.

Important

This action does not stop the instance implicitly. The user needs to use `aws:changeInstanceState` action to stop the instance. If this action is used on a running instance, the resultant AMI may be defective.

JSON Sample

```
{
  "name": "createMyImage",
  "action": "aws:createImage",
  "maxAttempts": 3,
  "onFailure": "Abort",
  "inputs": {
    "InstanceId": "i-123456789",
    "ImageName": "AMI Created on{{global:DATE_TIME}}",
    "NoReboot": true,
    "ImageDescription": "My newly created AMI"
  }
}
```

Inputs

The action supports most create-image API's parameters. For specific permitted value, please refer to the create-image api documentation.

```
ImageName
```

A string literal containing the name of the image.

Required: Yes

```
InstanceId
```

A string literal containing the id of the instance.

Required: Yes

```
BlockDeviceMappings
```

A map list containing the mappings for the instance.

Required: No

```
ImageDescription
```

A string literal.

Required: No

```
NoReboot
```

A boolean literal.

Required: No

Outputs

```
ImageId
```

A string literals containing the id of the newly created image.

```
ImageState
```

A string literals containing the state of the newly created image.

Action `aws:deleteImage`

You can use this action to delete an existing image.

JSON Sample

```
{
  "name": "deleteMyImage",
  "action": "aws:deleteImage",
  "maxAttempts": 3,
  "timeoutSeconds": 180,
  "onFailure": "Abort",
  "inputs": {
    "ImageId": "ami-1234567890"
  }
}
```

Inputs

The action supports most of delete-image API's parameters for Lambda service. For specific permitted value, please refer to the delete-image api documentation.

ImageId

A string literal containing the id of the image to be deleted.

Required: Yes

Outputs

None.

Automation System Variables

This section describes variable and parameter uses in Systems Manager Automation documents.

System Variables

Automation documents currently support the following system variables.

Variable	Details
global:DATE	The date (at execution time) in the format yyyy-MM-dd.
global:DATE_TIME	The date and time (at execution time) in the format yyyy-MM-dd_HH.mm.ss.
global:REGION	The region which the document is executed in. For example, us-east-1.

Automation Variables

Automation documents currently support the following automation variables.

Variable	Details
automation:EXECUTION_ID	The unique identifier assigned to the current automation execution. For example 1a2b3c-1a2b3c-1a2b3c-1a2b3c1a2b3c1a2b3c.

Terminology

This section uses the following terms to describe how variables and parameters are resolved.

Term	Definition	Example
Constant ARN	A valid ARN without variables	arn:aws:iam::123456789012:role/ roleName
Document Parameter	A parameter defined at the document level for an Automation document (e.g. instanceId in this example). The parameter is used in a basic string replace. Its value is supplied at Start Execution time.	<pre>{ "description": "Create Image Demo", "version": "0.3", "assumeRole": "Your_Automation_Assume_R "parameters": { "instanceId": { "type": "STRING", "description": "Instance to create image from" } } }</pre>
System variable	A general variable substituted into the document when any part of the document is evaluated.	<pre>"activities": [{ "id": "copyImage", "activityType": "AWS- CopyImage", "maxAttempts": 1, "onFailure": "Continue", "inputs": { "ImageName": "{{imageName}}", "SourceImageId": "{{sourceImageId}}", "SourceRegion": "{{sourceRegion}}", "Encrypted": true, "ImageDescription": "Test CopyImage Description created on {{global:DATE}}" } }]</pre>
Automation variable	A variable relating to the automation execution substituted into the document when any part of the document is evaluated.	<pre>{ "name": "runFixedCmds",</pre>

Term	Definition	Example
		<pre> "action": "aws:runCommand", "maxAttempts": 1, "onFailure": "Continue", "inputs": { "DocumentName": "AWS- RunPowerShellScript", "InstanceIds": ["{{LaunchInstance.InstanceIds}}"], "Parameters": { "commands": ["dir", "date", "echo {Hello {{ssm:administratorName}}}", "{{outputFormat}}" -f "left", "right", "{{global:DATE}}", "{{au] } } } </pre>

Term	Definition	Example
SSM parameter	A variable defined within the Parameter Service. It is not declared as a Document Parameter. It may require permissions to access.	<pre> { "description": "Run Command Demo", "schemaVersion": "0.3", "assumeRole": "arn:aws:iam::12345678901:roleName", "parameters": { "commands": { "type": "STRING_LIST", "description": "list of commands to execute as part of first step" }, "instanceIds": { "type": "STRING_LIST", "description": "list of instances to execute commands on" } }, "mainSteps": [{ "name": "runFixedCmds", "action": "aws:runCommand", "maxAttempts": 1, "onFailure": "Continue", "inputs": { "DocumentName": "AWS-RunPowerShellScript", "InstanceIds": ["{{LaunchInstance.InstanceIds}}"], "Parameters": { "commands": ["dir", "date", "echo {Hello {{ssm:administratorName}}}"] } } }] } </pre>

Term	Definition	Example
		<pre> "{{outputFormat}}" -f "left","right","{{global:DATE}}", "{{au] } } } </pre>

Supported Scenarios

Scenario	Comments	Example
Constant ARN assumeRole at create	An authorization check will be performed to check the calling user is permitted to pass the given assume role.	<pre> { "description": "Test all Automation resolvable parameters", "schemaVersion": "0.3", "assumeRole": "arn:aws:iam::123456789012:role/ roleName", "parameters": { ... </pre>
Document Parameter supplied for assumeRole at create	Must be defined in the Parameter list of the document.	<pre> { "description": "Test all Automation resolvable parameters", "schemaVersion": "0.3", "assumeRole": "{{dynamicARN}}", "parameters": { ... </pre>
Value supplied for Document Parameter at start.	Customer supplies the value to use for a parameter. Any execution inputs supplied at start time need to be defined in the parameter list of the document.	<pre> ... "parameters": { "amiId": { "type": "STRING", "default": "ami-7f2e6015", "description": "list of commands to execute as part of first step" }, ... </pre> <p>Inputs to Start Automation Execution include : {"amiId" : ["ami-12345678"] }</p>

Scenario	Comments	Example
<p>SSM parameter referenced within step definition</p>	<p>The variable exists within the customers account and the assumeRole for the document has access to the variable. A check will be performed at create time to confirm the assumeRole has access. SSM parameters do not need to be set in the parameter list of the document.</p>	<pre> ... "mainSteps": [{ "name": "RunSomeCommands", "action": "aws:runCommand", "aws:runCommand": { "maxAttempts": 1, "onFailure": "Continue", "inputs": { "DocumentName": "AWS:RunPowerShell", "InstanceIds": [{{LaunchInstance.InstanceIds}}], "Parameters": { "commands" : ["echo {Hello {{ssm:administratorName}}}"] } }, ... </pre>

Scenario	Comments	Example
<p>System variable referenced within step definition</p>	<p>A system variable is substituted into the document at execution time. The value injected into the document is relative to when the substitution occurs. e.g. The value of a time variable injected at step 1 will be different to the value injected at step 3 due to the time taken to execute the steps between. System variables do not need to be set in the parameter list of the document.</p>	<pre> ... "mainSteps": [{ "name": "RunSomeCommands", "action": "aws:runCommand", "maxAttempts": 1, "onFailure": "Continue", "inputs": { "DocumentName": "AWS:RunPowerShell", "InstanceIds": [[{LaunchInstance.InstanceIds}]], "Parameters": { "commands" : ["echo {The time is now {{global:TIME}}}"] } } }, ... </pre>
<p>Automation variable referenced within step definition.</p>	<p>Automation variables do not need to be set in the parameter list of the document. The only supported Automation variable is automation:EXECUTION_ID.</p>	<pre> ... "mainSteps": [{ "name": "invokeLambdaFunction", "action": "aws:invokeLambdaFunction", "maxAttempts": 1, "onFailure": "Continue", "inputs": { "FunctionName": "Hello-World- LambdaFunction", "Payload" : "{ "executionId" : "{{automation:EXECUTION_ID}}"}" }" } } }, ... </pre>

Scenario	Comments	Example
<p>Refer to output from previous step within next step definition.</p>	<p>This is parameter redirection. The output of a previous step is referenced using the syntax <code>{{stepName.OutputName}}</code>. This syntax cannot be used by the customer for Document Parameters. This is resolved at the time of execution for the referring step. The parameter is not listed in the parameters of the document.</p>	<pre> ... "mainSteps": [{ "name": "LaunchInstance", "action": "aws:runInstances", "aws:runInstances": { "maxAttempts": 1, "onFailure": "Continue", "inputs": { "ImageId": "{{amiId}}", "MinInstanceCount": 1, "MaxInstanceCount": 2 } }, { "name": "changeState", "action": "aws:changeInstanceState", "aws:changeInstanceState": { "maxAttempts": 1, "onFailure": "Continue", "inputs": { "InstanceIds": ["{{LaunchInstance.InstanceIds}} "], "DesiredState": "terminated" } } } }] ... </pre>

Unsupported Scenarios

Scenario	Comment	Example
SSM Parameter supplied for assumeRole at create	Not supported.	<pre> ... { "description": "Test all Automation resolvable parameters", "schemaVersion": "0.3", "assumeRole": "{{ssm:administratorRoleARN}}", "parameters": { ... </pre>
SSM Parameter supplied for Document Parameter at start	The user supplies an input parameter at start time which is an SSM parameter	<pre> ... "parameters": { "amiId": { "type": "STRING", "default": "ami-7f2e6015", "description": "list of commands to execute as part of first step" }, ... User supplies input : { "amiId" : "{{ssm:goldenAMIId}}" } </pre>

Scenario	Comment	Example
Variable step definition	The definition of a step in the document is constructed by variables.	<pre> ... "mainSteps": [{ "name": "LaunchInstance", "action": "aws:runInstances", "{{attemptModel}}": 1, "onFailure": "Continue", "inputs": { "ImageId": "ami-123456", "MinInstanceCount": 1, "MaxInstanceCount": 2 } }] ... User supplies input : { "attemptModel" : "minAttempts" }</pre>
Cross referencing Document Parameters	The user supplies an input parameter at start time which is a reference to another parameter in the document.	<pre> ... "parameters": { "amiId": { "type": "STRING", "default": "ami-7f2e6015", "description": "list of commands to execute as part of first step" }, "otherAmiId": { "type": "STRING", "description": "The other amiId to try if this one fails". } } "default" : "{{amiId}}" } ... </pre>

Scenario	Comment	Example
Multi-level expansion	The document defines a variable which evaluates to the name of a variable. This sits within the variable delimiters (that is <code>{{}}</code>) and is expanded to the value of that variable/parameter.	<pre> ... "parameters": { "param1": { "type": "STRING", "default": "param2", "description": "The parameter to reference" }, "param2": { "type": "STRING", "default" : "echo {Hello world}", "description": "What to execute" } }, "mainSteps": [{ "name": "runFixedCmds", "action": "aws:runCommand", "maxAttempts": 1, "onFailure": "Continue", "inputs": { "DocumentName": "AWS- RunPowerShellScript", "InstanceIds" : "{{LaunchInstance.InstanceIds}}", "Parameters": { "commands": ["{{ {{param1}} }"}"] } } ... Note: The customer intention here would be to execute a runCommand of "echo {Hello world}" </pre>

Related Content

- [Amazon EC2 Simple Systems Manager API Reference](#)
- [SSM AWS Tools for Windows PowerShell Reference](#)
- [SSM AWS CLI Reference](#)
- [AWS SDKs](#)

Monitoring Amazon EC2

Monitoring is an important part of maintaining the reliability, availability, and performance of your Amazon Elastic Compute Cloud (Amazon EC2) instances and your AWS solutions. You should collect monitoring data from all of the parts in your AWS solutions so that you can more easily debug a multi-point failure if one occurs. Before you start monitoring Amazon EC2, however, you should create a monitoring plan that should include:

- What are your goals for monitoring?
- What resources you will monitor?
- How often you will monitor these resources?
- What monitoring tools will you use?
- Who will perform the monitoring tasks?
- Who should be notified when something goes wrong?

After you have defined your monitoring goals and have created your monitoring plan, the next step is to establish a baseline for normal Amazon EC2 performance in your environment. You should measure Amazon EC2 performance at various times and under different load conditions. As you monitor Amazon EC2, you should store a history of monitoring data that you've collected. You can compare current Amazon EC2 performance to this historical data to help you to identify normal performance patterns and performance anomalies, and devise methods to address them. For example, you can monitor CPU utilization, disk I/O, and network utilization for your Amazon EC2 instances. When performance falls outside your established baseline, you might need to reconfigure or optimize the instance to reduce CPU utilization, improve disk I/O, or reduce network traffic.

To establish a baseline you should, at a minimum, monitor the following items:

Item to Monitor	Amazon EC2 Metric	Monitoring Script/CloudWatch Logs
CPU utilization	CPUUtilization (p. 577)	
Memory utilization		(Linux instances) Monitoring Memory and Disk Metrics for Amazon EC2 Linux Instances (Windows instances) Sending Performance Counters to CW; and Logs to CloudWatch Logs

Item to Monitor	Amazon EC2 Metric	Monitoring Script/CloudWatch Logs
Memory used		(Linux instances) Monitoring Memory and Disk Metrics for Amazon EC2 Linux Instances (Windows instances) Sending Performance Counters to CW; and Logs to CloudWatch Logs
Memory available		(Linux instances) Monitoring Memory and Disk Metrics for Amazon EC2 Linux Instances (Windows instances) Sending Performance Counters to CW; and Logs to CloudWatch Logs
Network utilization	NetworkIn (p. 577) NetworkOut (p. 577)	
Disk performance	DiskReadOps (p. 577) DiskWriteOps (p. 577)	
Disk Swap utilization (Linux instances only) Swap used (Linux instances only)		Monitoring Memory and Disk Metrics for Amazon EC2 Linux Instances
Page File utilization (Windows instances only) Page File used (Windows instances only) Page File available (Windows instances only)		Sending Performance Counters to CW; and Logs to CloudWatch Logs
Disk Reads/Writes	DiskReadBytes (p. 577) DiskWriteBytes (p. 577)	
Disk Space utilization (Linux instances only)		Monitoring Memory and Disk Metrics for Amazon EC2 Linux Instances
Disk Space used (Linux instances only)		Monitoring Memory and Disk Metrics for Amazon EC2 Linux Instances
Disk Space available (Linux instances only)		Monitoring Memory and Disk Metrics for Amazon EC2 Linux Instances

Automated and Manual Monitoring

AWS provides various tools that you can use to monitor Amazon EC2. You can configure some of these tools to do the monitoring for you, while some of the tools require manual intervention.

Topics

- [Automated Monitoring Tools \(p. 565\)](#)
- [Manual Monitoring Tools \(p. 566\)](#)

Automated Monitoring Tools

You can use the following automated monitoring tools to watch Amazon EC2 and report back to you when something is wrong:

- **System Status Checks** - monitor the AWS systems required to use your instance to ensure they are working properly. These checks detect problems with your instance that require AWS involvement to repair. When a system status check fails, you can choose to wait for AWS to fix the issue or you can resolve it yourself (for example, by stopping and restarting or terminating and replacing an instance). Examples of problems that cause system status checks to fail include:
 - Loss of network connectivity
 - Loss of system power
 - Software issues on the physical host
 - Hardware issues on the physical host

For more information, see [Status Checks for Your Instances \(p. 567\)](#).

- **Instance Status Checks** - monitor the software and network configuration of your individual instance. These checks detect problems that require your involvement to repair. When an instance status check fails, typically you will need to address the problem yourself (for example by rebooting the instance or by making modifications in your operating system). Examples of problems that may cause instance status checks to fail include:
 - Failed system status checks
 - Misconfigured networking or startup configuration
 - Exhausted memory
 - Corrupted file system
 - Incompatible kernel

For more information, see [Status Checks for Your Instances \(p. 567\)](#).

- **Amazon CloudWatch Alarms** - watch a single metric over a time period you specify, and perform one or more actions based on the value of the metric relative to a given threshold over a number of time periods. The action is a notification sent to an Amazon Simple Notification Service (Amazon SNS) topic or Auto Scaling policy. Alarms invoke actions for sustained state changes only. CloudWatch alarms will not invoke actions simply because they are in a particular state, the state must have changed and been maintained for a specified number of periods. For more information, see [Monitoring Your Instances Using CloudWatch \(p. 575\)](#).
- **Amazon CloudWatch Logs** - monitor, store, and access your log files from Amazon EC2 instances, AWS CloudTrail, or other sources. For more information, see [Monitoring Log Files](#).
- **Amazon EC2 Monitoring Scripts** - Perl scripts that can monitor memory, disk, and swap file usage in your instances. For more information, see [Monitoring Memory and Disk Metrics for Amazon EC2 Linux Instances](#).
- **AWS Management Pack for Microsoft System Center Operations Manager** - links Amazon EC2 instances and the Windows or Linux operating systems running inside them. The AWS

Management Pack is an extension to Microsoft System Center Operations Manager. It uses a designated computer in your datacenter (called a watcher node) and the Amazon Web Services APIs to remotely discover and collect information about your AWS resources. For more information, see [AWS Management Pack for Microsoft System Center \(p. 896\)](#).

Manual Monitoring Tools

Another important part of monitoring Amazon EC2 involves manually monitoring those items that the monitoring scripts, status checks, and CloudWatch alarms don't cover. The Amazon EC2 and CloudWatch console dashboards provide an at-a-glance view of the state of your Amazon EC2 environment.

- Amazon EC2 Dashboard shows:
 - Service Health and Scheduled Events by region
 - Instance state
 - Status checks
 - Alarm status
 - Instance metric details (In the navigation pane click **Instances**, select an instance, and then click the **Monitoring** tab)
 - Volume metric details (In the navigation pane click **Volumes**, select a volume, and then click the **Monitoring** tab)
- Amazon CloudWatch Dashboard shows:
 - Current alarms and status
 - Graphs of alarms and resources
 - Service health status

In addition, you can use CloudWatch to do the following:

- Graph Amazon EC2 monitoring data to troubleshoot issues and discover trends
- Search and browse all your AWS resource metrics
- Create and edit alarms to be notified of problems
- See at-a-glance overviews of your alarms and AWS resources

Best Practices for Monitoring

Use the following best practices for monitoring to help you with your Amazon EC2 monitoring tasks.

- Make monitoring a priority to head off small problems before they become big ones.
 - Create and implement a monitoring plan that collects monitoring data from all of the parts in your AWS solution so that you can more easily debug a multi-point failure if one occurs. Your monitoring plan should address, at a minimum, the following questions:
 - What are your goals for monitoring?
 - What resources you will monitor?
 - How often you will monitor these resources?
 - What monitoring tools will you use?
 - Who will perform the monitoring tasks?
 - Who should be notified when something goes wrong?
 - Automate monitoring tasks as much as possible.
 - Check the log files on your EC2 instances⁵⁶⁶
-

Monitoring the Status of Your Instances

You can monitor the status of your instances by viewing status checks and scheduled events for your instances. A status check gives you the information that results from automated checks performed by Amazon EC2. These automated checks detect whether specific issues are affecting your instances. The status check information, together with the data provided by Amazon CloudWatch, gives you detailed operational visibility into each of your instances.

You can also see status on specific events scheduled for your instances. Events provide information about upcoming activities such as rebooting or retirement that are planned for your instances, along with the scheduled start and end time of each event.

Contents

- [Status Checks for Your Instances \(p. 567\)](#)
- [Scheduled Events for Your Instances \(p. 571\)](#)

Status Checks for Your Instances

With instance status monitoring, you can quickly determine whether Amazon EC2 has detected any problems that might prevent your instances from running applications. Amazon EC2 performs automated checks on every running EC2 instance to identify hardware and software issues. You can view the results of these status checks to identify specific and detectable problems. This data augments the information that Amazon EC2 already provides about the intended state of each instance (such as `pending`, `running`, `stopping`) as well as the utilization metrics that Amazon CloudWatch monitors (CPU utilization, network traffic, and disk activity).

Status checks are performed every minute and each returns a pass or a fail status. If all checks pass, the overall status of the instance is **OK**. If one or more checks fail, the overall status is **impaired**. Status checks are built into Amazon EC2, so they cannot be disabled or deleted. You can, however create or delete alarms that are triggered based on the result of the status checks. For example, you can create an alarm to warn you if status checks fail on a specific instance. For more information, see [Creating and Editing Status Check Alarms \(p. 570\)](#).

Contents

- [Types of Status Checks \(p. 567\)](#)
- [Viewing Status Checks \(p. 568\)](#)
- [Reporting Instance Status \(p. 569\)](#)
- [Creating and Editing Status Check Alarms \(p. 570\)](#)

Types of Status Checks

There are two types of status checks: system status checks and instance status checks.

System Status Checks

Monitor the AWS systems required to use your instance to ensure they are working properly. These checks detect problems with your instance that require AWS involvement to repair. When a system status check fails, you can choose to wait for AWS to fix the issue, or you can resolve it yourself (for example, by stopping and starting an instance, or by terminating and replacing an instance).

The following are examples of problems that can cause system status checks to fail:

- Loss of network connectivity
- Loss of system power

- Software issues on the physical host
- Hardware issues on the physical host

Instance Status Checks

Monitor the software and network configuration of your individual instance. These checks detect problems that require your involvement to repair. When an instance status check fails, typically you will need to address the problem yourself (for example, by rebooting the instance or by making instance configuration changes).

The following are examples of problems that can cause instance status checks to fail:

- Failed system status checks
- Incorrect networking or startup configuration
- Exhausted memory
- Corrupted file system
- Status checks that occur during instance reboot or while a Windows instance store-backed instance is being bundled report an instance status check failure until the instance becomes available again.

Viewing Status Checks

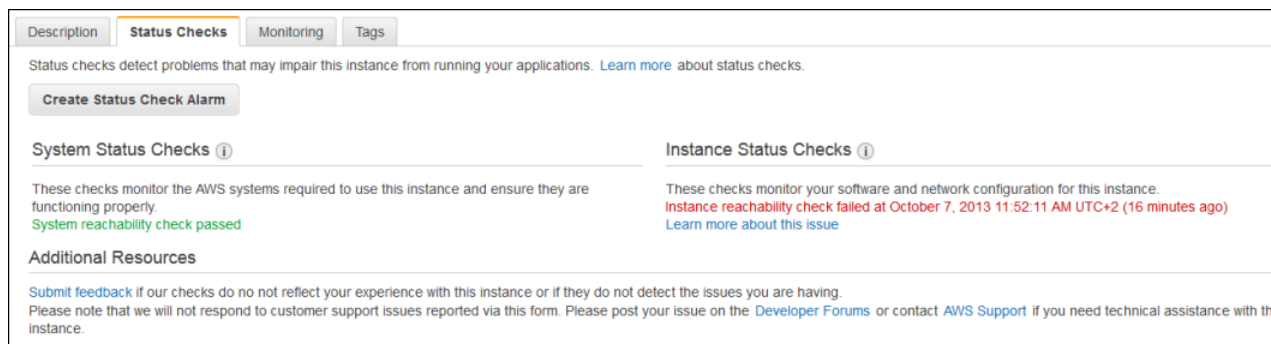
Amazon EC2 provides you with several ways to view and work with status checks.

Viewing Status Using the Console

You can view status checks using the AWS Management Console.

To view status checks using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. On the **Instances** page, the **Status Checks** column lists the operational status of each instance.
4. To view the status of a specific instance, select the instance, and then choose the **Status Checks** tab.



5. If you have an instance with a failed status check and the instance has been unreachable for over 20 minutes, choose **AWS Support** to submit a request for assistance.

Viewing Status Using the Command Line or API

You can view status checks for running instances using the `describe-instance-status` (AWS CLI) command.

To view the status of all instances, use the following command:

```
aws ec2 describe-instance-status
```

To get the status of all instances with a instance status of `impaired`:

```
aws ec2 describe-instance-status --filters Name=instance-  
status,status,Values=impaired
```

To get the status of a single instance, use the following command:

```
aws ec2 describe-instance-status --instance-ids i-1234567890abcdef0
```

Alternatively, use the following commands:

- [Get-EC2InstanceStatus](#) (AWS Tools for Windows PowerShell)
- [DescribeInstanceStatus](#) (Amazon EC2 Query API)

Reporting Instance Status

You can provide feedback if you are having problems with an instance whose status is not shown as `impaired`, or want to send AWS additional details about the problems you are experiencing with an `impaired` instance.

We use reported feedback to identify issues impacting multiple customers, but do not respond to individual account issues. Providing feedback does not change the status check results that you currently see for the instance.

Reporting Status Feedback Using the Console

To report instance status using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance.
4. Select the **Status Checks** tab, and then choose **Submit feedback**.
5. Complete the **Report Instance Status** form, and then choose **Submit**.

Reporting Status Feedback Using the Command Line or API

Use the following [report-instance-status](#) (AWS CLI) command to send feedback about the status of an `impaired` instance:

```
aws ec2 report-instance-status --instances i-1234567890abcdef0 --status  
impaired --reason-codes code
```

Alternatively, use the following commands:

- [Send-EC2InstanceStatus](#) (AWS Tools for Windows PowerShell)
- [ReportInstanceStatus](#) (Amazon EC2 Query API)

Creating and Editing Status Check Alarms

You can create instance status and system status alarms to notify you when an instance has a failed status check.

Creating a Status Check Alarm Using the Console

You can create status check alarms for an existing instance to monitor instance status or system status. You can configure the alarm to send you a notification by email or stop, terminate, or recover an instance when it fails an instance status check or system status check.

To create a status check alarm

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance.
4. Select the **Status Checks** tab, and then choose **Create Status Check Alarm**.
5. Select **Send a notification to**. Choose an existing SNS topic, or click **create topic** to create a new one. If creating a new topic, in **With these recipients**, enter your email address and the addresses of any additional recipients, separated by commas.
6. (Optional) Choose **Take the action**, and then select the action that you'd like to take.
7. In **Whenever**, select the status check that you want to be notified about.

Note

If you selected **Recover this instance** in the previous step, select **Status Check Failed (System)**.

8. In **For at least**, set the number of periods you want to evaluate and in **consecutive periods**, select the evaluation period duration before triggering the alarm and sending an email.
9. (Optional) In **Name of alarm**, replace the default name with another name for the alarm.
10. Choose **Create Alarm**.

Important

If you added an email address to the list of recipients or created a new topic, Amazon SNS sends a subscription confirmation email message to each new address. Each recipient must confirm the subscription by clicking the link contained in that message. Alert notifications are sent only to confirmed addresses.

If you need to make changes to an instance status alarm, you can edit it.

To edit a status check alarm

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance, choose **Actions**, select **CloudWatch Monitoring**, and then choose **Add/Edit Alarms**.
4. In the **Alarm Details** dialog box, choose the name of the alarm.
5. In the **Edit Alarm** dialog box, make the desired changes, and then choose **Save**.

Creating a Status Check Alarm Using the AWS CLI

In the following example, the alarm publishes a notification to an SNS topic, `arn:aws:sns:us-west-2:111122223333:my-sns-topic`, when the instance fails either the instance check or system status check for at least two consecutive periods. The metric is `StatusCheckFailed`.

To create a status check alarm using the CLI

1. Select an existing SNS topic or create a new one. For more information, see [Using the AWS CLI with Amazon SNS](#) in the *AWS Command Line Interface User Guide*.
2. Use the following `list-metrics` command to view the available Amazon CloudWatch metrics for Amazon EC2:

```
aws cloudwatch list-metrics --namespace AWS/EC2
```

3. Use the following `put-metric-alarm` command to create the alarm:

```
aws cloudwatch put-metric-alarm --alarm-name StatusCheckFailed-Alarm-for-i-1234567890abcdef0 --metric-name StatusCheckFailed --namespace AWS/EC2 --statistic Maximum --dimensions Name=InstanceId,Value=i-1234567890abcdef0 --unit Count --period 300 --evaluation-periods 2 --threshold 1 --comparison-operator GreaterThanOrEqualToThreshold --alarm-actions arn:aws:sns:us-west-2:111122223333:my-sns-topic
```

Note

- `--period` is the time frame, in seconds, in which Amazon CloudWatch metrics are collected. This example uses 300, which is 60 seconds multiplied by 5 minutes.
- `--evaluation-periods` is the number of consecutive periods for which the value of the metric must be compared to the threshold. This example uses 2.
- `--alarm-actions` is the list of actions to perform when this alarm is triggered. Each action is specified as an Amazon Resource Name (ARN). This example configures the alarm to send an email using Amazon SNS.

Scheduled Events for Your Instances

AWS can schedule events for your instances, such as a reboot, stop/start, or retirement. These events do not occur frequently. If one of your instances will be affected by a scheduled event, AWS sends an email to the email address that's associated with your AWS account prior to the scheduled event, with details about the event, including the start and end date. Depending on the event, you might be able to take action to control the timing of the event.

To update the contact information for your account so that you can be sure to be notified about scheduled events, go to the [Account Settings](#) page.

Contents

- [Types of Scheduled Events](#) (p. 571)
- [Viewing Scheduled Events](#) (p. 572)
- [Working with Instances Scheduled to Stop or Retire](#) (p. 573)
- [Working with Instances Scheduled for Reboot](#) (p. 574)
- [Working with Instances Scheduled for Maintenance](#) (p. 574)

Types of Scheduled Events

Amazon EC2 supports the following types of scheduled events for your instances:

- **Instance stop:** The instance will be stopped. When you start it again, it's migrated to a new host computer. Applies only to instances backed by Amazon EBS.
- **Instance retirement:** The instance will be stopped or terminated.

- **Reboot:** Either the instance will be rebooted (instance reboot) or the host computer for the instance will be rebooted (system reboot).
- **System maintenance:** The instance might be temporarily affected by network maintenance or power maintenance.

Viewing Scheduled Events

In addition to receiving notification of scheduled events in email, you can check for scheduled events.

To view scheduled events for your instances using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, click **Events**. Any resources with an associated event are displayed. You can filter by resource type, or by specific event types. You can select the resource to view details.

Filter:	All resource types	All event types	Ongoing and scheduled	
	Resource Name	Resource Type	Resource Id	Event Type
	my-instance	instance	i-c3870335	instance-stop

Event: i-c3870335

Availability Zone	us-west-2a
Event type	instance-stop
Event status	Scheduled
Description	The instance is running on degraded hardware
Start time	May 22, 2015 at 5:00:00 PM UTC-7
End time	

3. Alternatively, in the navigation pane, choose **EC2 Dashboard**. Any resources with an associated event are displayed under **Scheduled Events**.

Scheduled Events

US West (Oregon):
1 instances have scheduled events

4. Note that events are also shown for affected resource. For example, in the navigation pane, choose **Instances**, and then select an instance. If the instance has an associated event, it is displayed in the lower pane.

Retiring: This instance is scheduled for retirement after May 22, 2015 at 5:00:00 PM UTC-7.

To view scheduled events for your instances using the command line or API

Use the following AWS CLI command:

```
aws ec2 describe-instance-status --instance-id i-1234567890abcdef0
```

The following is example output showing an instance retirement event:

```
{
```

```
"InstanceStatuses": [
  {
    "InstanceStatus": {
      "Status": "ok",
      "Details": [
        {
          "Status": "passed",
          "Name": "reachability"
        }
      ]
    },
    "AvailabilityZone": "us-west-2a",
    "InstanceId": "i-1234567890abcdef0",
    "InstanceState": {
      "Code": 16,
      "Name": "running"
    },
    "SystemStatus": {
      "Status": "ok",
      "Details": [
        {
          "Status": "passed",
          "Name": "reachability"
        }
      ]
    },
    "Events": [
      {
        "Code": "instance-stop",
        "Description": "The instance is running on degraded hardware",
        "NotBefore": "2015-05-23T00:00:00.000Z"
      }
    ]
  }
]
```

Alternatively, use the following commands:

- [Get-EC2InstanceStatus](#) (AWS Tools for Windows PowerShell)
- [DescribeInstanceStatus](#) (Amazon EC2 Query API)

Working with Instances Scheduled to Stop or Retire

When AWS detects irreparable failure of the underlying host computer for your instance, it schedules the instance to stop or terminate, depending on the type of root device for the instance. If the root device is an EBS volume, the instance is scheduled to stop. If the root device is an instance store volume, the instance is scheduled to terminate. For more information, see [Instance Retirement](#) (p. 262).

Important

Any data stored on instance store volumes is lost when an instance is stopped or terminated. This includes instance store volumes that are attached to an instance that has an EBS volume as the root device. Be sure to save data from your instance store volumes that you will need later before the instance is stopped or terminated.

Actions for Instances Backed by Amazon EBS

You can wait for the instance to stop as scheduled. Alternatively, you can stop and start the instance yourself, which migrates it to a new host computer. For more information about stopping your instance, as well as information about the changes to your instance configuration when it's stopped, see [Stop and Start Your Instance \(p. 259\)](#).

Actions for Instances Backed by Instance Store

We recommend that you launch a replacement instance from your most recent AMI and migrate all necessary data to the replacement instance before the instance is scheduled to terminate. Then, you can terminate the original instance, or wait for it to terminate as scheduled.

Working with Instances Scheduled for Reboot

When AWS needs to perform tasks such as installing updates or maintaining the underlying host computer, it can schedule an instance or the underlying host computer for the instance for a reboot. Regardless of any existing instances that are scheduled for reboot, a new instance launch does not require a reboot, as the updates are already applied on the underlying host.

You can determine whether the reboot event is an instance reboot or a system reboot.

To view the type of scheduled reboot event using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Events**.
3. Select **Instance resources** from the filter list, and then select your instance.
4. In the bottom pane, locate **Event type**. The value is either `system-reboot` or `instance-reboot`.

To view the type of scheduled reboot event using the AWS CLI

Use the following [describe-instance-status](#) command:

```
aws ec2 describe-instance-status --instance-ids i-1234567890abcdef0
```

Actions for Instance Reboot

You can wait for the reboot to occur within its scheduled maintenance window. Alternatively, you can reboot your instance yourself at a time that is convenient for you. For more information, see [Reboot Your Instance \(p. 262\)](#).

After you reboot your instance, the scheduled event for the instance reboot is canceled immediately and the event's description is updated. The pending maintenance to the underlying host computer is completed, and you can begin using your instance again after it has fully booted.

Actions for System Reboot

No action is required on your part; the system reboot occurs during its scheduled maintenance window. A system reboot typically completes in a matter of minutes. To verify that the reboot has occurred, check that there is no longer a scheduled event for the instance. We recommend that you check whether the software on your instance is operating as you expect.

Working with Instances Scheduled for Maintenance

When AWS needs to maintain the underlying host computer for an instance, it schedules the instance for maintenance. There are two types of maintenance events: network maintenance and power maintenance.

During network maintenance, scheduled instances lose network connectivity for a brief period of time. Normal network connectivity to your instance will be restored after maintenance is complete.

During power maintenance, scheduled instances are taken offline for a brief period, and then rebooted. When a reboot is performed, all of your instance's configuration settings are retained.

After your instance has rebooted (this normally takes a few minutes), verify that your application is working as expected. At this point, your instance should no longer have a scheduled event associated with it, or the description of the scheduled event begins with **[Completed]**. It sometimes takes up to 1 hour for this instance status to refresh. Completed maintenance events are displayed on the Amazon EC2 console dashboard for up to a week.

Actions for Instances Backed by Amazon EBS

You can wait for the maintenance to occur as scheduled. Alternatively, you can stop and start the instance, which migrates it to a new host computer. For more information about stopping your instance, as well as information about the changes to your instance configuration when it's stopped, see [Stop and Start Your Instance \(p. 259\)](#).

Actions for Instances Backed by Instance Store

You can wait for the maintenance to occur as scheduled. Alternatively, if you want to maintain normal operation during a scheduled maintenance window, you can launch a replacement instance from your most recent AMI, migrate all necessary data to the replacement instance before the scheduled maintenance window, and then terminate the original instance.

Monitoring Your Instances Using CloudWatch

You can monitor your instances using Amazon CloudWatch, which collects and processes raw data from Amazon EC2 into readable, near real-time metrics. These statistics are recorded for a period of two weeks, so that you can access historical information and gain a better perspective on how your web application or service is performing.

By default, Amazon EC2 sends metric data to CloudWatch in 5-minute periods. To send metric data for your instance to CloudWatch in 1-minute periods, you can enable detailed monitoring on the instance. For more information, see [Enable or Disable Detailed Monitoring for Your Instances \(p. 575\)](#).

The Amazon EC2 console displays a series of graphs based on the raw data from Amazon CloudWatch. Depending on your needs, you might prefer to get data for your instances from Amazon CloudWatch instead of the graphs in the console.

For more information about Amazon CloudWatch, see the [Amazon CloudWatch User Guide](#).

Contents

- [Enable or Disable Detailed Monitoring for Your Instances \(p. 575\)](#)
- [List the Available CloudWatch Metrics for Your Instances \(p. 577\)](#)
- [Get Statistics for Metrics for Your Instances \(p. 583\)](#)
- [Graph Metrics for Your Instances \(p. 590\)](#)
- [Create a CloudWatch Alarm for an Instance \(p. 591\)](#)
- [Create Alarms That Stop, Terminate, Reboot, or Recover an Instance \(p. 592\)](#)

Enable or Disable Detailed Monitoring for Your Instances

By default, your instance is enabled for basic monitoring. You can optionally enable detailed monitoring. After you enable detailed monitoring, the Amazon EC2 console displays monitoring graphs with a 1-minute period for the instance. The following table describes basic and detailed monitoring for instances.

Type	Description
Basic	Data is available automatically in 5-minute periods at no charge.
Detailed	Data is available in 1-minute periods for an additional cost. To get this level of data, you must specifically enable it for the instance. For the instances where you've enabled detailed monitoring, you can also get aggregated data across groups of similar instances. For information about pricing, see the Amazon CloudWatch product page .

Enabling Detailed Monitoring

You can enable detailed monitoring on an instance as you launch it or after the instance is running or stopped.

To enable detailed monitoring for an existing instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance, choose **Actions, CloudWatch Monitoring, Enable Detailed Monitoring**.
4. In the **Enable Detailed Monitoring** dialog box, choose **Yes, Enable**.
5. Choose **Close**.

To enable detailed monitoring when launching an instance using the console

When launching an instance using the AWS Management Console, select the **Monitoring** check box on the **Configure Instance Details** page.

To enable detailed monitoring for an existing instance using the AWS CLI

Use the following `monitor-instances` command to enable detailed monitoring for the specified instances.

```
aws ec2 monitor-instances --instance-ids i-1234567890abcdef0
```

To enable detailed monitoring when launching an instance using the AWS CLI

Use the `run-instances` command with the `--monitoring` flag to enable detailed monitoring.

```
aws ec2 run-instances --image-id ami-09092360 --monitoring Enabled=true...
```

Disabling Detailed Monitoring

You can disable detailed monitoring on an instance as you launch it or after the instance is running or stopped.

To disable detailed monitoring using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

2. In the navigation pane, choose **Instances**.
3. Select the instance, choose **Actions, CloudWatch Monitoring, Disable Detailed Monitoring**.
4. In the **Disable Detailed Monitoring** dialog box, choose **Yes, Disable**.
5. Choose **Close**.

To disable detailed monitoring using the AWS CLI

Use the following `unmonitor-instances` command to disable detailed monitoring for the specified instances.

```
aws ec2 unmonitor-instances --instance-ids i-1234567890abcdef0
```

List the Available CloudWatch Metrics for Your Instances

Amazon EC2 sends metrics to Amazon CloudWatch. You can use the AWS Management Console, the AWS CLI, or an API to list the metrics that Amazon EC2 sends to CloudWatch. By default, each data point covers the previous 5 minutes of activity for the instance. If you've enabled detailed monitoring, each data point covers the previous 1 minute of activity.

For information about getting the statistics for these metrics, see [Get Statistics for Metrics for Your Instances \(p. 583\)](#).

Instance Metrics

The `AWS/EC2` namespace includes the following CPU credit metrics for your T2 instances. CPU credit metrics are available at a 5 minute frequency.

Metric	Description
<code>CPUCreditUsage</code>	<p>[T2 instances] The number of CPU credits consumed during the specified period.</p> <p>This metric identifies the amount of time during which physical CPUs were used for processing instructions by virtual CPUs allocated to the instance.</p> <p>CPU Credit metrics are available at a 5 minute frequency.</p> <p>Units: Count</p>
<code>CPUCreditBalance</code>	<p>[T2 instances] The number of CPU credits that an instance has accumulated.</p> <p>This metric determines how long an instance can burst beyond its baseline performance level at a given rate.</p> <p>CPU Credit metrics are available at a 5 minute frequency.</p> <p>Units: Count</p>

The `AWS/EC2` namespace includes the following instance metrics.

Amazon Elastic Compute Cloud
User Guide for Windows Instances
List Available Metrics

Metric	Description
CPUUtilization	<p>The percentage of allocated EC2 compute units that are currently in use on the instance. This metric identifies the processing power required to run an application upon a selected instance.</p> <p>To use the percentiles statistic, you must enable detailed monitoring.</p> <p>Depending on the instance type, tools in your operating system can show a lower percentage than CloudWatch when the instance is not allocated a full processor core.</p> <p>Units: Percent</p>
DiskReadOps	<p>Completed read operations from all instance store volumes available to the instance in a specified period of time.</p> <p>To calculate the average I/O operations per second (IOPS) for the period, divide the total operations in the period by the number of seconds in that period.</p> <p>Units: Count</p>
DiskWriteOps	<p>Completed write operations to all instance store volumes available to the instance in a specified period of time.</p> <p>To calculate the average I/O operations per second (IOPS) for the period, divide the total operations in the period by the number of seconds in that period.</p> <p>Units: Count</p>
DiskReadBytes	<p>Bytes read from all instance store volumes available to the instance.</p> <p>This metric is used to determine the volume of the data the application reads from the hard disk of the instance. This can be used to determine the speed of the application.</p> <p>Units: Bytes</p>
DiskWriteBytes	<p>Bytes written to all instance store volumes available to the instance.</p> <p>This metric is used to determine the volume of the data the application writes onto the hard disk of the instance. This can be used to determine the speed of the application.</p> <p>Units: Bytes</p>
NetworkIn	<p>The number of bytes received on all network interfaces by the instance. This metric identifies the volume of incoming network traffic to an application on a single instance.</p> <p>Units: Bytes</p>

Amazon Elastic Compute Cloud
User Guide for Windows Instances
List Available Metrics

Metric	Description
NetworkOut	The number of bytes sent out on all network interfaces by the instance. This metric identifies the volume of outgoing network traffic to an application on a single instance. Units: Bytes
NetworkPacketsIn	The number of packets received on all network interfaces by the instance. This metric identifies the volume of incoming traffic in terms of the number of packets on a single instance. This metric is available for basic monitoring only. Units: Count Statistics: Minimum, Maximum, Average
NetworkPacketsOut	The number of packets sent out on all network interfaces by the instance. This metric identifies the volume of outgoing traffic in terms of the number of packets on a single instance. This metric is available for basic monitoring only. Units: Count Statistics: Minimum, Maximum, Average

The `AWS/EC2` namespace includes the following status checks metrics. Status check metrics are available at a 1 minute frequency. For a newly-launched instance, status check metric data is only available after the instance has completed the initialization state (within a few minutes of the instance entering the running state).

Metric	Description
StatusCheckFailed	Reports whether the instance has passed both the instance status check and the system status check in the last minute. This metric can be either 0 (passed) or 1 (failed). Units: Count
StatusCheckFailed_Instance	Reports whether the instance has passed the instance status check in the last minute. This metric can be either 0 (passed) or 1 (failed). Units: Count
StatusCheckFailed_System	Reports whether the instance has passed the system status check in the last minute. This metric can be either 0 (passed) or 1 (failed). Units: Count

For information about the metrics provided for your EBS volumes, see [Amazon EBS Metrics \(p. 770\)](#). For information about the metrics provided for your Spot fleets, see [CloudWatch Metrics for Spot Fleet \(p. 213\)](#).

Amazon EC2 Dimensions

You can use the following dimensions to refine the metrics returned for your instances.

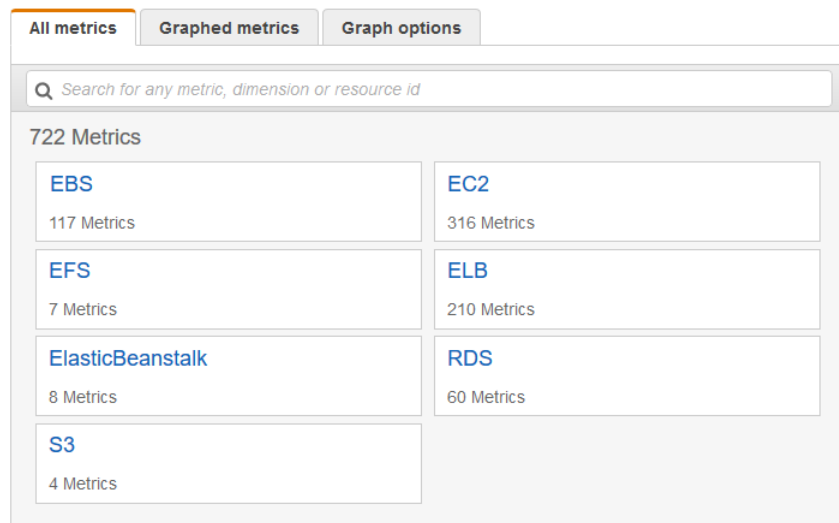
Dimension	Description
AutoScalingGroupName	This dimension filters the data you request for all instances in a specified capacity group. An <i>Auto Scaling group</i> is a collection of instances you define if you're using Auto Scaling. This dimension is available only for Amazon EC2 metrics when the instances are in such an Auto Scaling group. Available for instances with Detailed or Basic Monitoring enabled.
ImageId	This dimension filters the data you request for all instances running this Amazon EC2 Amazon Machine Image (AMI). Available for instances with Detailed Monitoring enabled.
InstanceId	This dimension filters the data you request for the identified instance only. This helps you pinpoint an exact instance from which to monitor data.
InstanceType	This dimension filters the data you request for all instances running with this specified instance type. This helps you categorize your data by the type of instance running. For example, you might compare data from an m1.small instance and an m1.large instance to determine which has the better business value for your application. Available for instances with Detailed Monitoring enabled.

Listing Metrics Using the Console

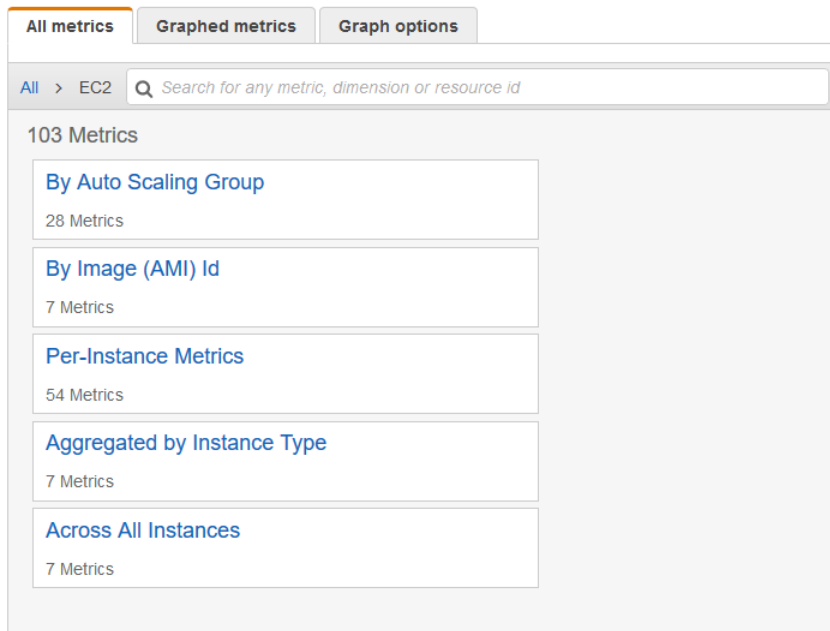
Metrics are grouped first by namespace, and then by the various dimension combinations within each namespace. For example, you can view all metrics provided by Amazon EC2, or metrics grouped by instance ID, instance type, image (AMI) ID, or Auto Scaling group.

To view available metrics by category

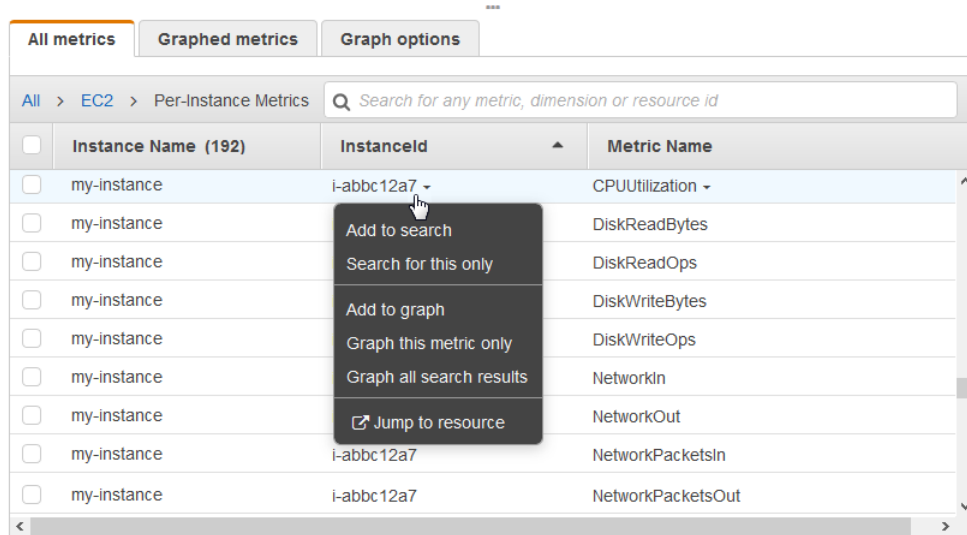
1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Metrics**.
3. Select the EC2 metric namespace.



4. Select a metric dimension (for example, Per-Instance Metrics).



5. To sort the metrics, use the column heading. To graph a metric, select the check box next to the metric. To filter by resource, choose the resource ID and then choose **Add to search**. To filter by metric, choose the metric name and then choose **Add to search**.



Listing Metrics Using the AWS CLI

Use the `list-metrics` command to list the CloudWatch metrics for your instances.

To list all the available metrics for Amazon EC2

The following example specifies the `AWS/EC2` namespace to view all the metrics for Amazon EC2.

```
aws cloudwatch list-metrics --namespace AWS/EC2
```

The following is example output:

```
{
  "Metrics": [
    {
      "Namespace": "AWS/EC2",
      "Dimensions": [
        {
          "Name": "InstanceId",
          "Value": "i-1234567890abcdef0"
        }
      ],
      "MetricName": "NetworkOut"
    },
    {
      "Namespace": "AWS/EC2",
      "Dimensions": [
        {
          "Name": "InstanceId",
          "Value": "i-1234567890abcdef0"
        }
      ],
      "MetricName": "CPUUtilization"
    },
    {
      "Namespace": "AWS/EC2",
      "Dimensions": [
```

```
        {  
          "Name": "InstanceId",  
          "Value": "i-1234567890abcdef0"  
        },  
        {  
          "MetricName": "NetworkIn"  
        },  
        ...  
      ]  
    }  
  }
```

To list all the available metrics for an instance

The following example specifies the `AWS/EC2` namespace and the `InstanceId` dimension to view the results for the specified instance only.

```
aws cloudwatch list-metrics --namespace AWS/EC2 --dimensions  
Name=InstanceId,Value=i-1234567890abcdef0
```

To list a metric across all instances

The following example specifies the `AWS/EC2` namespace and a metric name to view the results for the specified metric only.

```
aws cloudwatch list-metrics --namespace AWS/EC2 --metric-name CPUUtilization
```

Get Statistics for Metrics for Your Instances

You can get statistics for the CloudWatch metrics for your instances.

Contents

- [Statistics Overview \(p. 583\)](#)
- [Get Statistics for a Specific Instance \(p. 584\)](#)
- [Aggregate Statistics Across Instances \(p. 587\)](#)
- [Aggregate Statistics by Auto Scaling Group \(p. 588\)](#)
- [Aggregate Statistics by AMI \(p. 589\)](#)

Statistics Overview

Statistics are metric data aggregations over specified periods of time. CloudWatch provides statistics based on the metric data points provided by your custom data or provided by other services in AWS to CloudWatch. Aggregations are made using the namespace, metric name, dimensions, and the data point unit of measure, within the time period you specify. The following table describes the available statistics.

Statistic	Description
Minimum	The lowest value observed during the specified period. You can use this value to determine low volumes of activity for your application.
Maximum	The highest value observed during the specified period. You can use this value to determine high volumes of activity for your application.

Statistic	Description
Sum	All values submitted for the matching metric added together. This statistic can be useful for determining the total volume of a metric.
Average	The value of <code>Sum / SampleCount</code> during the specified period. By comparing this statistic with the <code>Minimum</code> and <code>Maximum</code> , you can determine the full scope of a metric and how close the average use is to the <code>Minimum</code> and <code>Maximum</code> . This comparison helps you to know when to increase or decrease your resources as needed.
SampleCount	The count (number) of data points used for the statistical calculation.
pNN.NN	The value of the specified percentile. You can specify any percentile, using up to two decimal places (for example, p95.45).

Get Statistics for a Specific Instance

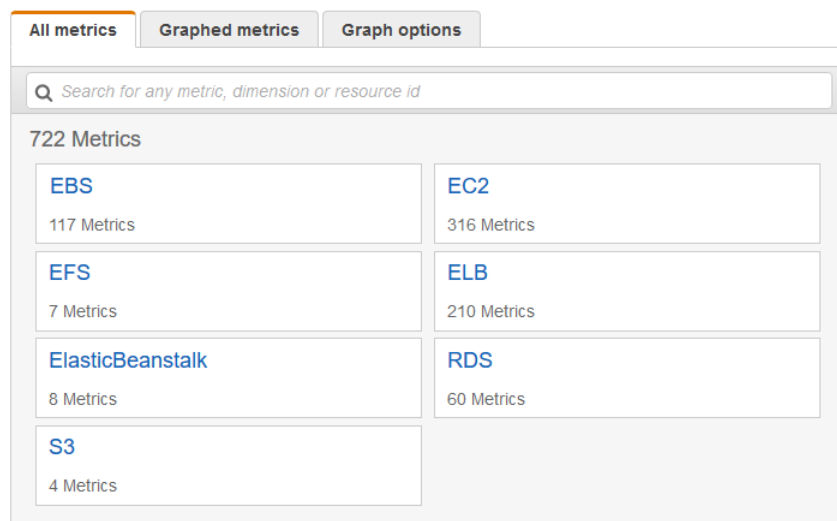
The following examples show you how to use the AWS Management Console or the AWS CLI to determine the maximum CPU utilization of a specific EC2 instance.

Requirements

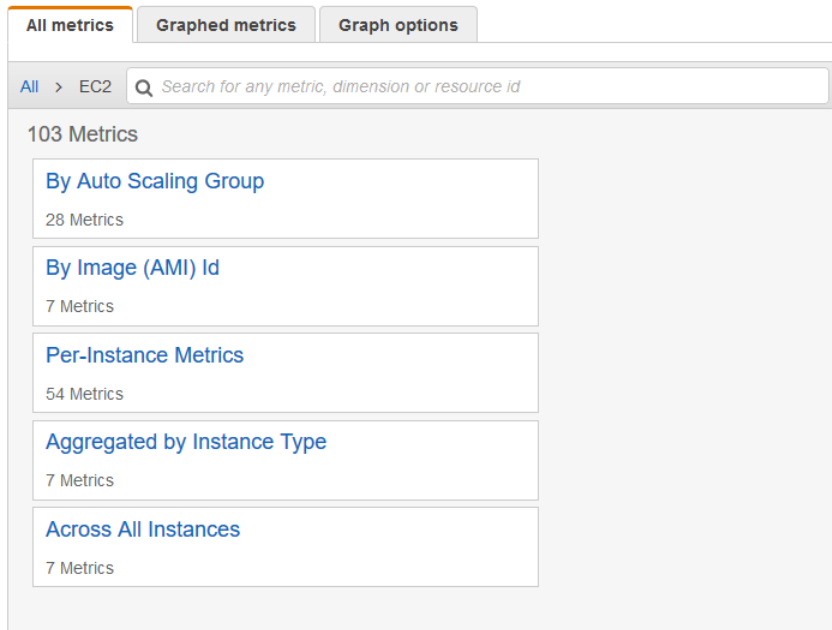
- You must have the ID of the instance. You can get the instance ID using the AWS Management Console or the `describe-instances` command.
- By default, basic monitoring is enabled, but you can enable detailed monitoring. For more information, see [Enable or Disable Detailed Monitoring for Your Instances \(p. 575\)](#).

To display the CPU utilization for a specific instance using the console

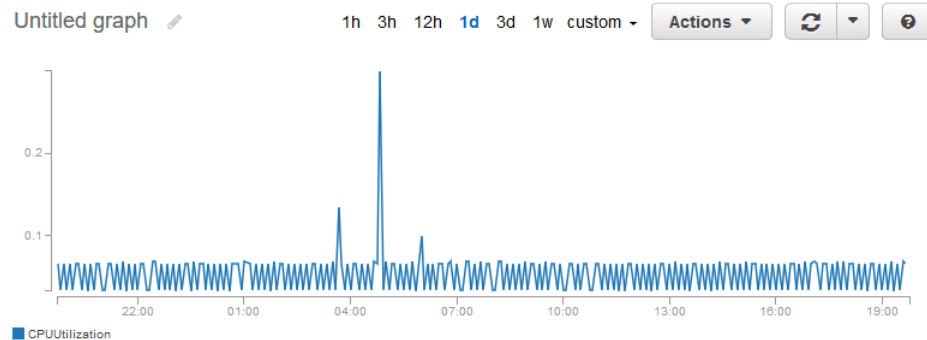
- Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
- In the navigation pane, choose **Metrics**.
- Select the EC2 metric namespace.



- Select the Per-Instance Metrics dimension.



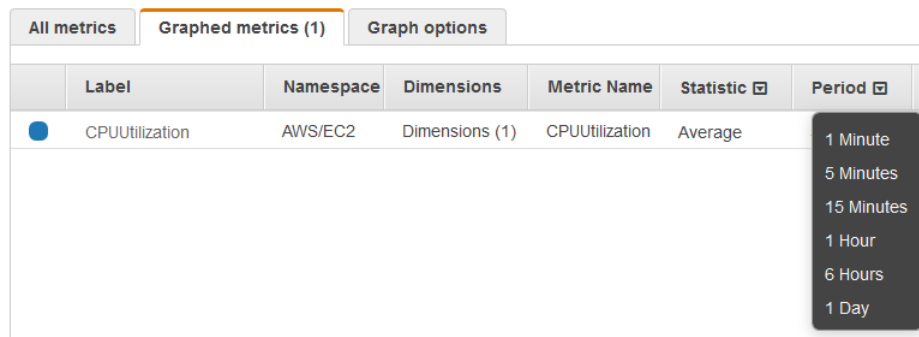
- In the search field, type **CPUUtilization** and press Enter. Select the row for the specific instance, which displays a graph for the **CPUUtilization** metric for the instance. To name the graph, choose the pencil icon. To change the time range, select one of the predefined values or choose **custom**.



Instance Name (4)	InstanceId	Metric Name
<input checked="" type="checkbox"/> my-instance	i-0dcbe8b2653841bd2	CPUUtilization
<input type="checkbox"/>	i-0b6eec80c79f745ad	CPUUtilization

- To change the statistic or the period for the metric, choose the **Graphed metrics** tab. Choose the column heading or an individual value, and then choose a different value.

Amazon Elastic Compute Cloud
User Guide for Windows Instances
Get Statistics for Metrics



To get the CPU utilization for a specific instance using the AWS CLI

Use the following [get-metric-statistics](#) command to get the **CPUUtilization** metric for the specified instance:

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name  
CPUUtilization --period 3600 \  
--statistics Maximum --dimensions Name=InstanceId,Value=i-1234567890abcdef0 \  
--start-time 2016-10-18T23:18:00 --end-time 2016-10-19T23:18:00
```

The returned statistics are six-minute values for the requested two-day time interval. Each value represents the maximum CPU utilization percentage for a single EC2 instance. The following is example output:

```
{  
  "Datapoints": [  
    {  
      "Timestamp": "2016-10-19T00:18:00Z",  
      "Maximum": 0.33000000000000002,  
      "Unit": "Percent"  
    },  
    {  
      "Timestamp": "2016-10-19T03:18:00Z",  
      "Maximum": 99.670000000000002,  
      "Unit": "Percent"  
    },  
    {  
      "Timestamp": "2016-10-19T07:18:00Z",  
      "Maximum": 0.34000000000000002,  
      "Unit": "Percent"  
    },  
    {  
      "Timestamp": "2016-10-19T12:18:00Z",  
      "Maximum": 0.34000000000000002,  
      "Unit": "Percent"  
    },  
    ...  
  ],  
  "Label": "CPUUtilization"  
}
```

Aggregate Statistics Across Instances

Aggregate statistics are available for the instances that have detailed monitoring enabled. Instances that use basic monitoring are not included in the aggregates. In addition, Amazon CloudWatch does not aggregate data across regions. Therefore, metrics are completely separate between regions. Before you can get statistics aggregated across instances, you must enable detailed monitoring (at an additional charge), which provides data in 1-minute periods.

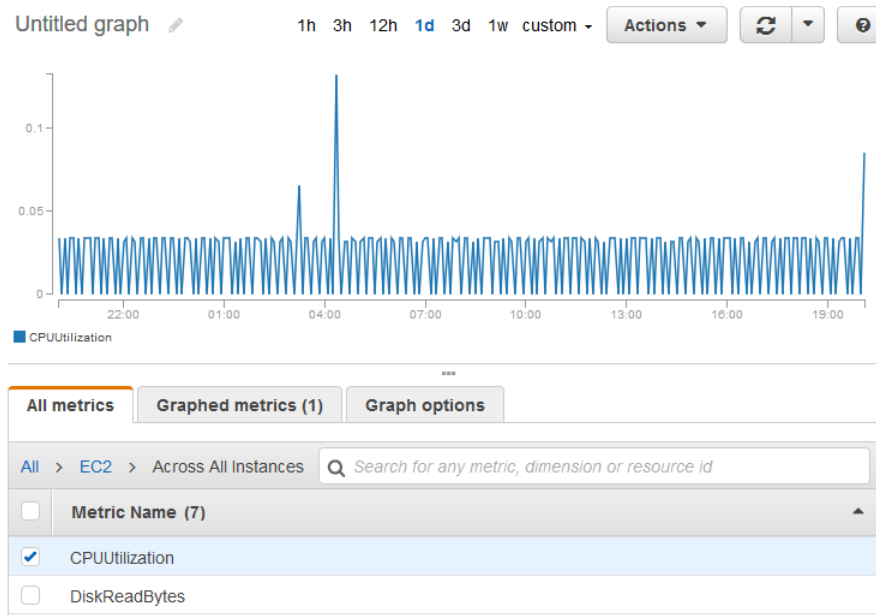
This example shows you how to use detailed monitoring to get the average CPU usage for your EC2 instances. Because no dimension is specified, CloudWatch returns statistics for all dimensions in the `AWS/EC2` namespace.

Important

This technique for retrieving all dimensions across an AWS namespace does not work for custom namespaces that you publish to Amazon CloudWatch. With custom namespaces, you must specify the complete set of dimensions that are associated with any given data point to retrieve statistics that include the data point.

To display average CPU utilization across your instances

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Metrics**.
3. Select the **EC2** namespace and then select **Across All Instances**.
4. Select the row that contains **CPUUtilization**, which displays a graph for the metric for all your EC2 instances. To name the graph, choose the pencil icon. To change the time range, select one of the predefined values or choose **custom**.



5. To change the statistic or the period for the metric, choose the **Graphed metrics** tab. Choose the column heading or an individual value, and then choose a different value.

To get average CPU utilization across your instances

Use the [get-metric-statistics](#) command as follows to get the average of the **CPUUtilization** metric across your instances.

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name  
CPUUtilization \  
--period 3600 --statistics "Average" "SampleCount" \  
--start-time 2016-10-11T23:18:00 --end-time 2016-10-12T23:18:00
```

The following is example output:

```
{  
  "Datapoints": [  
    {  
      "SampleCount": 238.0,  
      "Timestamp": "2016-10-12T07:18:00Z",  
      "Average": 0.038235294117647062,  
      "Unit": "Percent"  
    },  
    {  
      "SampleCount": 240.0,  
      "Timestamp": "2016-10-12T09:18:00Z",  
      "Average": 0.16670833333333332,  
      "Unit": "Percent"  
    },  
    {  
      "SampleCount": 238.0,  
      "Timestamp": "2016-10-11T23:18:00Z",  
      "Average": 0.041596638655462197,  
      "Unit": "Percent"  
    },  
    ...  
  ],  
  "Label": "CPUUtilization"  
}
```

Aggregate Statistics by Auto Scaling Group

You can aggregate statistics for the EC2 instances in an Auto Scaling group. Note that Amazon CloudWatch cannot aggregate data across regions. Metrics are completely separate between regions.

This example shows you how to retrieve the total bytes written to disk for one Auto Scaling group. The total is computed for one-minute periods for a 24-hour interval across all EC2 instances in the specified Auto Scaling group.

To display DiskWriteBytes for the instances in an Auto Scaling group using the console

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Metrics**.
3. Select the **EC2** namespace and then select **By Auto Scaling Group**.
4. Select the row for the **DiskWriteBytes** metric and the specific Auto Scaling group, which displays a graph for the metric for the instances in the Auto Scaling group. To name the graph, choose the pencil icon. To change the time range, select one of the predefined values or choose **custom**.
5. To change the statistic or the period for the metric, choose the **Graphed metrics** tab. Choose the column heading or an individual value, and then choose a different value.

To display DiskWriteBytes for the instances in an Auto Scaling group using the AWS CLI

Use the `get-metric-statistics` command as follows.

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name  
DiskWriteBytes --period 360 \  
--statistics "Sum" "SampleCount" --dimensions  
Name=AutoScalingGroupName,Value=my-asg --start-time 2016-10-16T23:18:00 --  
end-time 2016-10-18T23:18:00
```

The following is example output:

```
{  
  "Datapoints": [  
    {  
      "SampleCount": 18.0,  
      "Timestamp": "2016-10-19T21:36:00Z",  
      "Sum": 0.0,  
      "Unit": "Bytes"  
    },  
    {  
      "SampleCount": 5.0,  
      "Timestamp": "2016-10-19T21:42:00Z",  
      "Sum": 0.0,  
      "Unit": "Bytes"  
    }  
  ],  
  "Label": "DiskWriteBytes"  
}
```

Aggregate Statistics by AMI

You can aggregate statistics for your instances that have detailed monitoring enabled. Instances that use basic monitoring are not included. Note that Amazon CloudWatch cannot aggregate data across regions. Metrics are completely separate between regions.

Before you can get statistics aggregated across instances, you must enable detailed monitoring (at an additional charge), which provides data in 1-minute periods. For more information, see [Enable or Disable Detailed Monitoring for Your Instances \(p. 575\)](#).

This example shows you how to determine average CPU utilization for all instances that use a specific Amazon Machine Image (AMI). The average is over 60-second time intervals for a one-day period.

To display the average CPU utilization by AMI using the console

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Metrics**.
3. Select the **EC2** namespace and then select **By Image (AMI) Id**.
4. Select the row for the **CPUtilization** metric and the specific AMI, which displays a graph for the metric for the specified AMI. To name the graph, choose the pencil icon. To change the time range, select one of the predefined values or choose **custom**.
5. To change the statistic or the period for the metric, choose the **Graphed metrics** tab. Choose the column heading or an individual value, and then choose a different value.

To get the average CPU utilization for an image ID

Use the `get-metric-statistics` command as follows.

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name  
CPUtilization --period 3600 \  

```

```
--statistics Average --dimensions Name=ImageId,Value=ami-3c47a355 --start-time 2016-10-10T00:00:00 --end-time 2016-10-11T00:00:00
```

The following is example output. The operation returns statistics that are one-minute values for the one-day interval. Each value represents an average CPU utilization percentage for the EC2 instances running the specified AMI.

```
{
  "Datapoints": [
    {
      "Timestamp": "2016-10-10T07:00:00Z",
      "Average": 0.041000000000000009,
      "Unit": "Percent"
    },
    {
      "Timestamp": "2016-10-10T14:00:00Z",
      "Average": 0.079579831932773085,
      "Unit": "Percent"
    },
    {
      "Timestamp": "2016-10-10T06:00:00Z",
      "Average": 0.0360000000000000011,
      "Unit": "Percent"
    },
    ...
  ],
  "Label": "CPUUtilization"
}
```

Graph Metrics for Your Instances

After you launch an instance, you can open the Amazon EC2 console and view the monitoring graphs for an instance on the **Monitoring** tab. Each graph is based on one of the available Amazon EC2 metrics.

The following graphs are available:

- Average CPU Utilization (Percent)
- Average Disk Reads (Bytes)
- Average Disk Writes (Bytes)
- Maximum Network In (Bytes)
- Maximum Network Out (Bytes)
- Summary Disk Read Operations (Count)
- Summary Disk Write Operations (Count)
- Summary Status (Any)
- Summary Status Instance (Count)
- Summary Status System (Count)

For more information about the metrics and the data they provide to the graphs, see [List the Available CloudWatch Metrics for Your Instances \(p. 577\)](#).

Graph Metrics Using the CloudWatch Console

You can also use the CloudWatch console to graph metric data generated by Amazon EC2 and other AWS services. For more information, see [Graph Metrics](#) in the *Amazon CloudWatch User Guide*.

Create a CloudWatch Alarm for an Instance

You can create a CloudWatch alarm that monitors CloudWatch metrics for one of your instances. CloudWatch will automatically send you a notification when the metric reaches a threshold you specify. You can create a CloudWatch alarm using the Amazon EC2 console, or using the more advanced options provided by the CloudWatch console.

To create an alarm using the CloudWatch console

For examples, see [Creating Amazon CloudWatch Alarms](#) in the *Amazon CloudWatch User Guide*.

To create an alarm using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance.
4. On the **Monitoring** tab, choose **Create Alarm**.
5. On the **Create Alarm** page, do the following:
 - a. Choose **create topic**. For **Send a notification to**, type a name for the SNS topic. For **With these recipients**, type one or more email addresses to receive notification.
 - b. Specify the metric and the criteria for the policy. For example, you can leave the default settings for **Whenever** (Average of CPU Utilization). For **Is**, choose \geq and type 80 percent. For **For at least**, type 1 consecutive period of 5 Minutes.
 - c. Choose **Create Alarm**.

Create Alarm [X]

You can use CloudWatch alarms to be notified automatically whenever metric data reaches a level you define.
To edit an alarm, first choose whom to notify and then define when the notification should be sent.

Send a notification to: [cancel](#)

With these recipients:

Take the action:

- Recover this instance ⓘ
- Stop this instance ⓘ
- Terminate this instance ⓘ
- Reboot this instance ⓘ

Whenever: of

Is: Percent

For at least: consecutive period(s) of

Name of alarm:

[Cancel](#) [Create Alarm](#)

Create Alarms That Stop, Terminate, Reboot, or Recover an Instance

Using Amazon CloudWatch alarm actions, you can create alarms that automatically stop, terminate, reboot, or recover your instances. You can use the stop or terminate actions to help you save money when you no longer need an instance to be running. You can use the reboot and recover actions to automatically reboot those instances or recover them onto new hardware if a system impairment occurs.

Every alarm action you create uses alarm action ARNs. One set of ARNs is more secure because it requires you to have the `EC2ActionsAccess` IAM role in your account. This IAM role enables you to perform stop, terminate, or reboot actions—previously you could not execute an action if you were using an IAM role. Existing alarms that use the previous alarm action ARNs do not require this IAM role, however it is recommended that you change the ARN and add the role when you edit an existing alarm that uses these ARNs.

The `EC2ActionsAccess` role enables AWS to perform alarm actions on your behalf. When you create an alarm action for the first time using the Amazon EC2 or Amazon CloudWatch consoles, AWS automatically creates this role for you.

There are a number of scenarios in which you might want to automatically stop or terminate your instance. For example, you might have instances dedicated to batch payroll processing jobs or scientific computing tasks that run for a period of time and then complete their work. Rather than letting those instances sit idle (and accrue charges), you can stop or terminate them which can help you to save money. The main difference between using the stop and the terminate alarm actions is that you can easily restart a stopped instance if you need to run it again later, and you can keep the same instance ID and root volume. However, you cannot restart a terminated instance. Instead, you must launch a new instance.

You can add the stop, terminate, reboot, or recover actions to any alarm that is set on an Amazon EC2 per-instance metric, including basic and detailed monitoring metrics provided by Amazon CloudWatch (in the `AWS/EC2` namespace), as well as any custom metrics that include the `InstanceId=` dimension, as long as the `InstanceId` value refers to a valid running Amazon EC2 instance.

Console Support

You can create alarms using the Amazon EC2 console or the CloudWatch console. The procedures in this documentation use the Amazon EC2 console. For procedures that use the CloudWatch console, see [Create Alarms That Stop, Terminate, Reboot, or Recover an Instance](#) in the *Amazon CloudWatch User Guide*.

Permissions

If you are an AWS Identity and Access Management (IAM) user, you must have the following permissions to create or modify an alarm:

- `ec2:DescribeInstanceStatus` and `ec2:DescribeInstances` — For all alarms on Amazon EC2 instance status metrics
- `ec2:StopInstances` — For alarms with stop actions
- `ec2:TerminateInstances` — For alarms with terminate actions
- `ec2:DescribeInstanceRecoveryAttribute`, and `ec2:RecoverInstances` — For alarms with recover actions

If you have read/write permissions for Amazon CloudWatch but not for Amazon EC2, you can still create an alarm but the stop or terminate actions won't be performed on the Amazon EC2 instance.

However, if you are later granted permission to use the associated Amazon EC2 APIs, the alarm actions you created earlier will be performed. For more information about IAM permissions, see [Permissions and Policies](#) in the *IAM User Guide*.

If you want to use an IAM role to stop, terminate, or reboot an instance using an alarm action, you can only use the `EC2ActionsAccess` role. Other IAM roles are not supported. If you are using another IAM role, you cannot stop, terminate, or reboot the instance. However, you can still see the alarm state and perform any other actions such as Amazon SNS notifications or Auto Scaling policies.

If you are using temporary security credentials granted using the AWS Security Token Service (AWS STS), you cannot recover an Amazon EC2 instance using alarm actions.

Contents

- [Adding Stop Actions to Amazon CloudWatch Alarms](#) (p. 593)
- [Adding Terminate Actions to Amazon CloudWatch Alarms](#) (p. 594)
- [Adding Reboot Actions to Amazon CloudWatch Alarms](#) (p. 595)
- [Adding Recover Actions to Amazon CloudWatch Alarms](#) (p. 595)
- [Using the Amazon CloudWatch Console to View the History of Triggered Alarms and Actions](#) (p. 597)
- [Amazon CloudWatch Alarm Action Scenarios](#) (p. 597)

Adding Stop Actions to Amazon CloudWatch Alarms

You can create an alarm that stops an Amazon EC2 instance when a certain threshold has been met. For example, you may run development or test instances and occasionally forget to shut them off. You can create an alarm that is triggered when the average CPU utilization percentage has been lower than 10 percent for 24 hours, signaling that it is idle and no longer in use. You can adjust the threshold, duration, and period to suit your needs, plus you can add an Amazon Simple Notification Service (Amazon SNS) notification, so that you will receive an email when the alarm is triggered.

Instances that use an Amazon EBS volume as the root device can be stopped or terminated, whereas instances that use the instance store as the root device can only be terminated.

To create an alarm to stop an idle instance using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **INSTANCES**, choose **Instances**.
3. Select the instance. On the **Monitoring** tab, choose **Create Alarm**.
4. In the **Alarm Details for** dialog box, choose **Create Alarm**.
5. If you want to receive an email when the alarm is triggered, in the **Create Alarm for** dialog box, for **Send a notification to**, choose an existing Amazon SNS topic, or choose **Create Topic** to create a new one.

To create a new topic, for **Send a notification to**, type a name for the topic, and then for **With these recipients**, type the email addresses of the recipients (separated by commas). After you create the alarm, you will receive a subscription confirmation email that you must accept before you can get notifications for this topic.

6. Choose **Take the action**, and then choose the **Stop this instance** radio button.
7. If prompted, select **Create IAM role: EC2ActionsAccess** to automatically create an IAM role so that AWS can automatically stop the instance on your behalf when the alarm is triggered.
8. For **Whenever**, choose the statistic you want to use and then choose the metric. In this example, choose **Average** and **CPU Utilization**.

9. For **Is**, define the metric threshold. In this example, type **10** percent.
10. For **For at least**, choose the sampling period for the alarm. In this example, type **24** consecutive periods of one hour.
11. To change the name of the alarm, for **Name this alarm**, type a new name.

If you don't type a name for the alarm, Amazon CloudWatch will automatically create one for you.

Note

You can adjust the alarm configuration based on your own requirements before creating the alarm, or you can edit them later. This includes the metric, threshold, duration, action, and notification settings. However, after you create an alarm, you cannot edit its name later.

12. Choose **Create Alarm**.

Adding Terminate Actions to Amazon CloudWatch Alarms

You can create an alarm that terminates an EC2 instance automatically when a certain threshold has been met (as long as termination protection is not enabled for the instance). For example, you might want to terminate an instance when it has completed its work, and you don't need the instance again. If you might want to use the instance later, you should stop the instance instead of terminating it. For information on enabling and disabling termination protection for an instance, see [Enabling Termination Protection for an Instance](#) in the *Amazon EC2 User Guide for Linux Instances*.

To create an alarm to terminate an idle instance using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **INSTANCES**, choose **Instances**.
3. Select the instance. On the **Monitoring** tab, choose **Create Alarm**.
4. In the **Alarm Details for** dialog box, choose **Create Alarm**.
5. If you want to receive an email when the alarm is triggered, in the **Create Alarm for** dialog box, for **Send a notification to**, choose an existing Amazon SNS topic, or choose **Create Topic** to create a new one.

To create a new topic, for **Send a notification to**, type a name for the topic, and then for **With these recipients**, type the email addresses of the recipients (separated by commas). After you create the alarm, you will receive a subscription confirmation email that you must accept before you can get notifications for this topic.

6. Select **Take the action**, and then choose **Terminate this instance**.
7. If prompted, select **Create IAM role: EC2ActionsAccess** to automatically create an IAM role so that AWS can automatically stop the instance on your behalf when the alarm is triggered.
8. For **Whenever**, choose a statistic and then choose the metric. In this example, choose **Average** and **CPU Utilization**.
9. For **Is**, define the metric threshold. In this example, type **10** percent.
10. For **For at least**, choose the sampling period for the alarm. In this example, type **24** consecutive periods of one hour.
11. To change the name of the alarm, for **Name this alarm**, type a new name.

If you don't type a name for the alarm, Amazon CloudWatch will automatically create one for you.

Note

You can adjust the alarm configuration based on your own requirements before creating the alarm, or you can edit them later. This includes the metric, threshold, duration, action, and notification settings. However, after you create an alarm, you cannot edit its name later.

12. Choose **Create Alarm**.

Adding Reboot Actions to Amazon CloudWatch Alarms

You can create an Amazon CloudWatch alarm that monitors an Amazon EC2 instance and automatically reboots the instance. The reboot alarm action is recommended for Instance Health Check failures (as opposed to the recover alarm action, which is suited for System Health Check failures). An instance reboot is equivalent to an operating system reboot. In most cases, it takes only a few minutes to reboot your instance. When you reboot an instance, it remains on the same physical host, so your instance keeps its public DNS name, private IP address, and any data on its instance store volumes.

Rebooting an instance doesn't start a new instance billing hour, unlike stopping and restarting your instance. For more information, see [Reboot Your Instance](#) in the *Amazon EC2 User Guide for Linux Instances*.

Important

To avoid a race condition between the reboot and recover actions, we recommend that you set the alarm threshold to **3** for **1** minute when creating alarms that reboot an Amazon EC2 instance.

To create an alarm to reboot an instance using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **INSTANCES**, choose **Instances**.
3. Select the instance. On the **Monitoring** tab, choose **Create Alarm**.
4. In the **Alarm Details for** dialog box, choose **Create Alarm**.
5. If you want to receive an email when the alarm is triggered, in the **Create Alarm for** dialog box, for **Send a notification to**, choose an existing Amazon SNS topic, or choose **Create Topic** to create a new one.

To create a new topic, for **Send a notification to**, type a name for the topic, and for **With these recipients**, type the email addresses of the recipients (separated by commas). After you create the alarm, you will receive a subscription confirmation email that you must accept before you can get notifications for this topic.

6. Select **Take the action**, and then choose **Reboot this instance**.
7. If prompted, select **Create IAM role: EC2ActionsAccess** to automatically create an IAM role so that AWS can automatically stop the instance on your behalf when the alarm is triggered.
8. For **Whenever**, choose Status Check Failed (Instance).
9. For **For at least**, type **2**.
10. For **consecutive period(s) of**, choose **1 minute**.
11. To change the name of the alarm, for **Name of alarm**, type a new name.

If you don't type a name for the alarm, Amazon CloudWatch will automatically create one for you.

12. Choose **Create Alarm**.

Adding Recover Actions to Amazon CloudWatch Alarms

You can create an Amazon CloudWatch alarm that monitors an Amazon EC2 instance and automatically recovers the instance if it becomes impaired due to an underlying hardware failure or a problem that requires AWS involvement to repair. Terminated instances cannot be recovered. A recovered instance is identical to the original instance, including the instance ID, private IP addresses, Elastic IP addresses, and all instance metadata.

When the `StatusCheckFailed_System` alarm is triggered, and the recover action is initiated, you will be notified by the Amazon SNS topic that you chose when you created the alarm and associated the recover action. During instance recovery, the instance is migrated during an instance reboot, and any data that is in-memory is lost. When the process is complete, information is published to the SNS topic you've configured for the alarm. Anyone who is subscribed to this SNS topic will receive an email notification that includes the status of the recovery attempt and any further instructions. You will notice an instance reboot on the recovered instance.

Examples of problems that cause system status checks to fail include:

- Loss of network connectivity
- Loss of system power
- Software issues on the physical host
- Hardware issues on the physical host

The recover action is supported only on instances with the following characteristics:

- Use a C3, C4, M3, M4, R3, R4, T2, or X1 instance type
- Run in a VPC (not EC2-Classic)
- Use shared tenancy (the `tenancy` attribute is set to `default`)
- Use EBS volumes, including encrypted EBS volumes (not instance store volumes)

If your instance has a public IP address, it retains the public IP address after recovery.

Important

To avoid a race condition between the reboot and recover actions, we recommend that you set the alarm threshold to **2** for **1** minute when creating alarms that recover an Amazon EC2 instance.

To create an alarm to recover an instance using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **INSTANCES**, choose **Instances**.
3. Select the instance. On the **Monitoring** tab, choose **Create Alarm**.
4. In the **Alarm Details for** dialog box, choose **Create Alarm**.
5. To receive an email when the alarm is triggered, in the **Create Alarm for** dialog box, for **Send a notification to**, choose an existing Amazon SNS topic, or choose **Create Topic** to create a new one.

To create a new topic, for **Send a notification to**, type a name for the topic, and for **With these recipients**, type the email addresses of the recipients (separated by commas). After you create the alarm, you will receive a subscription confirmation email that you must accept before you can get email for this topic.

6. Select **Take the action**, and then choose **Recover this instance**.
7. If prompted, select **Create IAM role: EC2ActionsAccess** to automatically create an IAM role so that AWS can automatically stop the instance on your behalf when the alarm is triggered.
8. For **Whenever**, choose **Status Check Failed (System)**.
9. For **For at least**, type **2**.
10. For **consecutive period(s) of**, choose **1 minute**.
11. To change the name of the alarm, for **Name of alarm**, type a new name.

If you don't type a name for the alarm, Amazon CloudWatch will automatically create one for you.

12. Choose **Create Alarm**.

Using the Amazon CloudWatch Console to View the History of Triggered Alarms and Actions

You can view alarm and action history in the Amazon CloudWatch console. Amazon CloudWatch keeps the last two weeks' worth of alarm and action history.

To view the history of triggered alarms and actions

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Alarms**.
3. Select an alarm.
4. The **Details** tab shows the most recent state transition along with the time and metric values.
5. Choose the **History** tab to view the most recent history entries.

Amazon CloudWatch Alarm Action Scenarios

You can use the Amazon EC2 console to create alarm actions that stop or terminate an Amazon EC2 instance when certain conditions are met. In the following screen capture of the console page where you set the alarm actions, we've numbered the settings. We've also numbered the settings in the scenarios that follow, to help you create the appropriate actions.

Create Alarm

You can use CloudWatch alarms to be notified automatically whenever metric data reaches a level you define.
To edit an alarm, first choose whom to notify and then define when the notification should be sent.

Send a notification to: [create topic](#)

Take the action:

- Recover this instance [i](#)
- Stop this instance [i](#)
- Terminate this instance [i](#)
- Reboot this instance [i](#)

AWS will create the following IAM role in your account so that AWS can perform this action. [Learn more.](#)

Create IAM role: **EC2ActionsAccess** (show IAM policy document)

Whenever: **2** of **3**

is: **4** **5** Percent

For at least: **6** consecutive period(s) of **7**

Name of alarm:

[Cancel](#) [Create Alarm](#)

CPU Utilization Percent

75
50
25
0

7/21 22:00 7/22 00:00 7/22 02:00

i-0d23c3839e4e6d8e

Scenario 1: Stop Idle Development and Test Instances

Create an alarm that stops an instance used for software development or testing when it has been idle for at least an hour.

Setting	Value
	Stop
	Maximum
	CPUUtilization
	<=
	10%
	60 minutes
	1

Scenario 2: Stop Idle Instances

Create an alarm that stops an instance and sends an email when the instance has been idle for 24 hours.

Setting	Value
	Stop and email
	Average
	CPUUtilization
	<=
	5%
	60 minutes
	24

Scenario 3: Send Email About Web Servers with Unusually High Traffic

Create an alarm that sends email when an instance exceeds 10 GB of outbound network traffic per day.

Setting	Value
	Email
	Sum
	NetworkOut
	>
	10 GB
	1 day

Setting	Value
	1

Scenario 4: Stop Web Servers with Unusually High Traffic

Create an alarm that stops an instance and send a text message (SMS) if outbound traffic exceeds 1 GB per hour.

Setting	Value
	Stop and send SMS
	Sum
	NetworkOut
	>
	1 GB
	1 hour
	1

Scenario 5: Stop an Instance Experiencing a Memory Leak

Create an alarm that stops an instance when memory utilization reaches or exceeds 90%, so that application logs can be retrieved for troubleshooting.

Note

The MemoryUtilization metric is a custom metric. In order to use the MemoryUtilization metric, you must install the Perl scripts for Linux instances. For more information, see [Monitoring Memory and Disk Metrics for Amazon EC2 Linux Instances](#).

Setting	Value
	Stop
	Maximum
	MemoryUtilization
	>=
	90%
	1 minute
	1

Scenario 6: Stop an Impaired Instance

Create an alarm that stops an instance that fails three consecutive status checks (performed at 5-minute intervals).

Setting	Value
	Stop

Amazon Elastic Compute Cloud
User Guide for Windows Instances
Create Alarms That Stop, Terminate,
Reboot, or Recover an Instance

Setting	Value
	Average
	StatusCheckFailed_System
	>=
	1
	15 minutes
	1

Scenario 7: Terminate Instances When Batch Processing Jobs Are Complete

Create an alarm that terminates an instance that runs batch jobs when it is no longer sending results data.

Setting	Value
	Terminate
	Maximum
	NetworkOut
	<=
	100,000 bytes
	5 minutes
	1

Network and Security

Amazon EC2 provides the following network and security features.

Features

- [Amazon EC2 Key Pairs and Windows Instances \(p. 602\)](#)
- [Amazon EC2 Security Groups for Windows Instances \(p. 606\)](#)
- [Controlling Access to Amazon EC2 Resources \(p. 619\)](#)
- [Amazon EC2 and Amazon Virtual Private Cloud \(p. 665\)](#)
- [Amazon EC2 Instance IP Addressing \(p. 693\)](#)
- [Elastic IP Addresses \(p. 709\)](#)
- [Elastic Network Interfaces \(p. 716\)](#)
- [Placement Groups \(p. 731\)](#)
- [Network Maximum Transmission Unit \(MTU\) for Your EC2 Instance \(p. 734\)](#)
- [Enhanced Networking on Windows \(p. 737\)](#)

If you access Amazon EC2 using the command line tools or an API, you'll need your access key ID and secret access key. For more information, see [How Do I Get Security Credentials?](#) in the *Amazon Web Services General Reference*.

You can launch an instance into one of two platforms: EC2-Classic or EC2-VPC. An instance that's launched into EC2-Classic or a default VPC is automatically assigned a public IP address. An instance that's launched into a nondefault VPC can be assigned a public IP address on launch. For more information about EC2-Classic and EC2-VPC, see [Supported Platforms \(p. 672\)](#).

Instances can fail or terminate for reasons outside of your control. If an instance fails and you launch a replacement instance, the replacement has a different public IP address than the original. However, if your application needs a static IP address, you can use an *Elastic IP address*.

You can use *security groups* to control who can access your instances. These are analogous to an inbound network firewall that enables you to specify the protocols, ports, and source IP ranges that are allowed to reach your instances. You can create multiple security groups and assign different rules to each group. You can then assign each instance to one or more security groups, and we use the rules to determine which traffic is allowed to reach the instance. You can configure a security group so that only specific IP addresses or specific security groups have access to the instance.

Amazon EC2 Key Pairs and Windows Instances

Amazon EC2 uses public–key cryptography to encrypt and decrypt login information. Public–key cryptography uses a public key to encrypt a piece of data, such as a password, then the recipient uses the private key to decrypt the data. The public and private keys are known as a *key pair*.

To log in to your instance, you must create a key pair, specify the name of the key pair when you launch the instance, and provide the private key when you connect to the instance. With Windows instances, you use a key pair to obtain the administrator password and then log in using RDP. For more information about key pairs and Linux instances, see [Amazon EC2 Key Pairs](#) in the *Amazon EC2 User Guide for Linux Instances*.

Creating a Key Pair

You can use Amazon EC2 to create your key pair. For more information, see [Creating Your Key Pair Using Amazon EC2](#) (p. 602).

Alternatively, you could use a third-party tool and then import the public key to Amazon EC2. For more information, see [Importing Your Own Key Pair to Amazon EC2](#) (p. 603).

Each key pair requires a name. Be sure to choose a name that is easy to remember. Amazon EC2 associates the public key with the name that you specify as the key name.

Amazon EC2 stores the public key only, and you store the private key. Anyone who possesses your private key can decrypt your login information, so it's important that you store your private keys in a secure place.

The keys that Amazon EC2 uses are 2048-bit SSH-2 RSA keys. You can have up to five thousand key pairs per region.

Launching and Connecting to Your Instance

When you launch an instance, you should specify the name of the key pair you plan to use to connect to the instance. If you don't specify the name of an existing key pair when you launch an instance, you won't be able to connect to the instance. When you connect to the instance, you must specify the private key that corresponds to the key pair you specified when you launched the instance.

Note

Contents

- [Creating Your Key Pair Using Amazon EC2](#) (p. 602)
- [Importing Your Own Key Pair to Amazon EC2](#) (p. 603)
- [Retrieving the Public Key for Your Key Pair on Windows](#) (p. 604)
- [Verifying Your Key Pair's Fingerprint](#) (p. 605)
- [Deleting Your Key Pair](#) (p. 605)

Creating Your Key Pair Using Amazon EC2

You can create a key pair using the Amazon EC2 console or the command line. After you create a key pair, you can specify it when you launch your instance.

To create your key pair using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **NETWORK & SECURITY**, choose **Key Pairs**.

Tip

The navigation pane is on the left side of the Amazon EC2 console. If you do not see the pane, it might be minimized; choose the arrow to expand the pane.

3. Choose **Create Key Pair**.
4. Enter a name for the new key pair in the **Key pair name** field of the **Create Key Pair** dialog box, and then choose **Create**.
5. The private key file is automatically downloaded by your browser. The base file name is the name you specified as the name of your key pair, and the file name extension is `.pem`. Save the private key file in a safe place.

Important

This is the only chance for you to save the private key file. You'll need to provide the name of your key pair when you launch an instance and the corresponding private key each time you connect to the instance.

To create your key pair using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [create-key-pair](#) (AWS CLI)
- [New-EC2KeyPair](#) (AWS Tools for Windows PowerShell)

Importing Your Own Key Pair to Amazon EC2

If you used Amazon EC2 to create your key pair, as described in the previous section, you are ready to launch an instance. Otherwise, instead of using Amazon EC2 to create your key pair, you can create an RSA key pair using a third-party tool and then import the public key to Amazon EC2. For example, you can use **ssh-keygen** (a tool provided with the standard OpenSSH installation) to create a key pair. Alternatively, Java, Ruby, Python, and many other programming languages provide standard libraries that you can use to create an RSA key pair.

Amazon EC2 accepts the following formats:

- OpenSSH public key format
- Base64 encoded DER format
- SSH public key file format as specified in [RFC4716](#)

Amazon EC2 does not accept DSA keys. Make sure your key generator is set up to create RSA keys.

Supported lengths: 1024, 2048, and 4096.

To create a key pair using a third-party tool

1. Generate a key pair with a third-party tool of your choice.
2. Save the public key to a local file. For example, `C:\keys\my-key-pair.pub`. The file name extension for this file is not important.
3. Save the private key to a different local file that has the `.pem` extension. For example, `C:\keys\my-key-pair.pem`. Save the private key file in a safe place. You'll need to provide the name of your key pair when you launch an instance and the corresponding private key each time you connect to the instance.

Use the following steps to import your key pair using the Amazon EC2 console.

To import the public key

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **NETWORK & SECURITY**, choose **Key Pairs**.
3. Choose **Import Key Pair**.
4. In the **Import Key Pair** dialog box, choose **Browse**, and select the public key file that you saved previously. Enter a name for the key pair in the **Key pair name** field, and choose **Import**.

To import your key pair using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [import-key-pair](#) (AWS CLI)
- [Import-EC2KeyPair](#) (AWS Tools for Windows PowerShell)

After the public key file is imported, you can verify that the key pair was imported successfully using the Amazon EC2 console as follows.

To verify that your key pair was imported

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select the region in which you created the key pair.
3. In the navigation pane, under **NETWORK & SECURITY**, choose **Key Pairs**.
4. Verify that the key pair that you imported is in the displayed list of key pairs.

To view your key pair using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [describe-key-pairs](#) (AWS CLI)
- [Get-EC2KeyPair](#) (AWS Tools for Windows PowerShell)

Retrieving the Public Key for Your Key Pair on Windows

On Windows, you can use PuTTYgen to get the public key for your key pair. Start PuTTYgen, click **Load**, and select the `.ppk` or `.pem` file. PuTTYgen displays the public key.

The public key that you specified when you launched an instance is also available to you through its instance metadata. To view the public key that you specified when launching the instance, use the following command from your instance:

```
C:\> GET http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS7O6V
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/
d6RJhJJOI0iBXr
lsLnBITntckiJ7FbtxJMXLvwwJryDUiLBMTjYtwB+QhYXUMozce5Pjz5/i8SeJtjnV3iAoG/cQk
+0FzZ
qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi
+z7wB3Rb
```

```
BQoQzd8v7yeb7Oz1PnW0yN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE my-key-pair
```

For more information, see [Retrieving Instance Metadata \(p. 271\)](#).

Verifying Your Key Pair's Fingerprint

On the **Key Pairs** page in the Amazon EC2 console, the **Fingerprint** column displays the fingerprints generated from your key pairs. AWS calculates the fingerprint differently depending on whether the key pair was generated by AWS or a third-party tool. If you created the key pair using AWS, the fingerprint is calculated using an SHA-1 hash function. If you created the key pair with a third-party tool and uploaded the public key to AWS, or if you generated a new public key from an existing AWS-created private key and uploaded it to AWS, the fingerprint is calculated using an MD5 hash function.

You can use the fingerprint that's displayed on the **Key Pairs** page to verify that the private key you have on your local machine matches the public key that's stored in AWS.

If you created your key pair using AWS, you can use the OpenSSL tools to generate a fingerprint from the private key file:

```
C:\> openssl pkcs8 -in path_to_private_key -inform PEM -outform DER -topk8 -nocrypt | openssl sha1 -c
```

If you created your key pair using a third-party tool and uploaded the public key to AWS, you can use the OpenSSL tools to generate a fingerprint from the private key file on your local machine:

```
C:\> openssl rsa -in path_to_private_key -pubout -outform DER | openssl md5 -c
```

The output should match the fingerprint that's displayed in the console.

Deleting Your Key Pair

When you delete a key pair, you are only deleting Amazon EC2's copy of the public key. Deleting a key pair doesn't affect the private key on your computer or the public key on any instances already launched using that key pair. You can't launch a new instance using a deleted key pair, but you can continue to connect to any instances that you launched using a deleted key pair, as long as you still have the private key (.pem) file.

Note

If you're using an Auto Scaling group (for example, in an Elastic Beanstalk environment), ensure that the key pair you're deleting is not specified in your launch configuration. Auto Scaling launches a replacement instance if it detects an unhealthy instance; however, the instance launch fails if the key pair cannot be found.

You can delete a key pair using the Amazon EC2 console or the command line.

To delete your key pair using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **NETWORK & SECURITY**, choose **Key Pairs**.
3. Select the key pair and choose **Delete**.
4. When prompted, choose **Yes**.

To delete your key pair using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [delete-key-pair](#) (AWS CLI)
- [Remove-EC2KeyPair](#) (AWS Tools for Windows PowerShell)

Amazon EC2 Security Groups for Windows Instances

A *security group* acts as a virtual firewall that controls the traffic for one or more instances. When you launch an instance, you associate one or more security groups with the instance. You add rules to each security group that allow traffic to or from its associated instances. You can modify the rules for a security group at any time; the new rules are automatically applied to all instances that are associated with the security group. When we decide whether to allow traffic to reach an instance, we evaluate all the rules from all the security groups that are associated with the instance.

If you need to allow traffic to a Linux instance, see [Amazon EC2 Security Groups for Linux Instances](#) in the *Amazon EC2 User Guide for Linux Instances*.

Topics

- [Security Groups for EC2-Classic](#) (p. 606)
- [Security Groups for EC2-VPC](#) (p. 606)
- [Security Group Rules](#) (p. 607)
- [Default Security Groups](#) (p. 609)
- [Custom Security Groups](#) (p. 609)
- [Working with Security Groups](#) (p. 609)
- [Security Group Rules Reference](#) (p. 613)

If you have requirements that aren't met by security groups, you can maintain your own firewall on any of your instances in addition to using security groups.

Your account may support EC2-Classic in some regions, depending on when you created it. For more information, see [Supported Platforms](#) (p. 672). Security groups for EC2-Classic are separate to security groups for EC2-VPC.

Security Groups for EC2-Classic

If you're using EC2-Classic, you must use security groups created specifically for EC2-Classic. When you launch an instance in EC2-Classic, you must specify a security group in the same region as the instance. You can't specify a security group that you created for a VPC when you launch an instance in EC2-Classic.

After you launch an instance in EC2-Classic, you can't change its security groups. However, you can add rules to or remove rules from a security group, and those changes are automatically applied to all instances that are associated with the security group.

In EC2-Classic, you can have up to 500 security groups in each region for each account. You can associate an instance with up to 500 security groups and add up to 100 rules to a security group.

Security Groups for EC2-VPC

If you're using EC2-VPC, you must use security groups created specifically for your VPC. When you launch an instance in a VPC, you must specify a security group for that VPC. You can't specify a security group that you created for EC2-Classic when you launch an instance in a VPC. Security groups for EC2-VPC have additional capabilities that aren't supported by security groups for EC2-

Classic. For more information, see [Differences Between Security Groups for EC2-Classic and EC2-VPC](#) in the *Amazon VPC User Guide*.

After you launch an instance in a VPC, you can change its security groups. Security groups are associated with network interfaces. Changing an instance's security groups changes the security groups associated with the primary network interface (eth0). For more information, see [Changing an Instance's Security Groups](#) in the *Amazon VPC User Guide*. You can also change the security groups associated with any other network interface. For more information, see [Changing the Security Group](#) (p. 726).

Security groups for EC2-VPC have separate limits. For more information, see [Amazon VPC Limits](#) in the *Amazon VPC User Guide*. The security groups for EC2-Classic do not count against the security group limit for EC2-VPC.

Your VPC can be enabled for IPv6. For more information, see [IP addressing in Your VPC](#) in the *Amazon VPC User Guide*. You can add rules to your VPC security groups to enable inbound and outbound IPv6 traffic.

Security Group Rules

The rules of a security group control the inbound traffic that's allowed to reach the instances that are associated with the security group and the outbound traffic that's allowed to leave them.

The following are the characteristics of security group rules:

- By default, security groups allow all outbound traffic.
- You can't change the outbound rules for an EC2-Classic security group.
- Security group rules are always permissive; you can't create rules that deny access.
- Security groups are stateful — if you send a request from your instance, the response traffic for that request is allowed to flow in regardless of inbound security group rules. For VPC security groups, this also means that responses to allowed inbound traffic are allowed to flow out, regardless of outbound rules. For more information, see [Connection Tracking](#) (p. 608).
- You can add and remove rules at any time. Your changes are automatically applied to the instances associated with the security group after a short period.

Note

The effect of some rule changes may depend on how the traffic is tracked. For more information, see [Connection Tracking](#) (p. 608).

- When you associate multiple security groups with an instance, the rules from each security group are effectively aggregated to create one set of rules. We use this set of rules to determine whether to allow access.

Note

You can assign multiple security groups to an instance, therefore an instance can have hundreds of rules that apply. This might cause problems when you access the instance. We recommend that you condense your rules as much as possible.

For each rule, you specify the following:

- **Protocol:** The protocol to allow. The most common protocols are 6 (TCP) 17 (UDP), and 1 (ICMP).
- **Port range :** For TCP, UDP, or a custom protocol, the range of ports to allow.
- **ICMP type and code:** For ICMP, the ICMP type and code.
- **Source or destination:** The source (inbound rules) or destination (outbound rules) for the traffic. Specify one of these options:
 - An individual IPv4 address. You must use the /32 prefix after the IPv4 address; for example, 203.0.113.1/32.

- (VPC only) An individual IPv6 address. You must use the /128 prefix length; for example `2001:db8:1234:1a00::123/128`.
- A range of IPv4 addresses, in CIDR block notation, for example, `203.0.113.0/24`.
- (VPC only) A range of IPv6 addresses, in CIDR block notation, for example, `2001:db8:1234:1a00::/64`.
- Another security group. This allows instances associated with the specified security group to access instances associated with this security group. This does not add rules from the source security group to this security group. You can specify one of the following security groups:
 - The current security group.
 - EC2-Classic: A different security group for EC2-Classic in the same region.
 - EC2-Classic: A security group for another AWS account in the same region (add the AWS account ID as a prefix; for example, `111122223333/sg-edcd9784`).
 - EC2-VPC: A different security group for the same VPC or a peer VPC in a VPC peering connection.

When you specify a security group as the source or destination for a rule, the rule affects all instances associated with the security group. Incoming traffic is allowed based on the private IP addresses of the instances that are associated with the source security group (and not the public IP or Elastic IP addresses). For more information about IP addresses, see [Amazon EC2 Instance IP Addressing \(p. 693\)](#). If your security group rule references a security group in a peer VPC, and the referenced security group or VPC peering connection is deleted, the rule is marked as stale. For more information, see [Working with Stale Security Group Rules](#) in the *Amazon VPC Peering Guide*.

If there is more than one rule for a specific port, we apply the most permissive rule. For example, if you have a rule that allows access to TCP port 3389 (RDP) from IP address `203.0.113.1` and another rule that allows access to TCP port 3389 from everyone, everyone has access to TCP port 3389.

Connection Tracking

Your security groups use connection tracking to track information about traffic to and from the instance. Rules are applied based on the connection state of the traffic to determine if the traffic is allowed or denied. This allows security groups to be stateful — responses to inbound traffic are allowed to flow out of the instance regardless of outbound security group rules, and vice versa. For example, if you initiate an ICMP `ping` command to your instance from your home computer, and your inbound security group rules allow ICMP traffic, information about the connection (including the port information) is tracked. Response traffic from the instance for the `ping` command is not tracked as a new request, but rather as an established connection and is allowed to flow out of the instance, even if your outbound security group rules restrict outbound ICMP traffic.

Not all flows of traffic are tracked. If a security group rule permits TCP or UDP flows for all traffic and there is a corresponding rule in the other direction that permits the response traffic, then that flow of traffic is not tracked. The response traffic is therefore allowed to flow based on the inbound or outbound rule that permits the response traffic, and not on tracking information.

An existing flow of traffic that is tracked may not be interrupted when you remove the security group rule that enables that flow. Instead, the flow is interrupted when it's stopped by you or the other host for at least a few minutes (or up to 5 days for established TCP connections). For UDP, this may require terminating actions on the remote side of the flow. An untracked flow of traffic is immediately interrupted if the rule that enables the flow is removed or modified. For example, if you remove a rule that allows all inbound SSH traffic to the instance, then your existing SSH connections to the instance are immediately dropped.

For protocols other than TCP, UDP, or ICMP, only the IP address and protocol number is tracked. If your instance sends traffic to another host (host B), and host B initiates the same type of traffic to your instance in a separate request within 600 seconds of the original request or response, your instance accepts it regardless of inbound security group rules, because it's regarded as response traffic.

For VPC security groups, to ensure that traffic is immediately interrupted when you remove a security group rule, or to ensure that all inbound traffic is subject to firewall rules, you can use a network ACL for your subnet — network ACLs are stateless and therefore do not automatically allow response traffic. For more information, see [Network ACLs](#) in the *Amazon VPC User Guide*.

Default Security Groups

Your AWS account automatically has a *default security group* per VPC and per region for EC2-Classic. If you don't specify a security group when you launch an instance, the instance is automatically associated with the default security group.

A default security group is named `default`, and it has an ID assigned by AWS. The following are the default rules for each default security group:

- Allows all inbound traffic from other instances associated with the default security group (the security group specifies itself as a source security group in its inbound rules)
- Allows all outbound traffic from the instance.

You can add or remove the inbound rules for any default security group. You can add or remove outbound rules for any VPC default security group.

You can't delete a default security group. If you try to delete the EC2-Classic default security group, you'll get the following error: `Client.InvalidGroup.Reserved: The security group 'default' is reserved.` If you try to delete a VPC default security group, you'll get the following error: `Client.CannotDelete: the specified group: "sg-51530134" name: "default" cannot be deleted by a user.`

Custom Security Groups

If you don't want your instances to use the default security group, you can create your own security groups and specify them when you launch your instances. You can create multiple security groups to reflect the different roles that your instances play; for example, a web server or a database server.

When you create a security group, you must provide it with a name and a description. Security group names and descriptions can be up to 255 characters in length, and are limited to the following characters:

- EC2-Classic: ASCII characters
- EC2-VPC: a-z, A-Z, 0-9, spaces, and `._-:/()#,@[]+=&:{}!$*`

The following are the default rules for a security group that you create:

- Allows no inbound traffic
- Allows all outbound traffic

After you've created a security group, you can change its inbound rules to reflect the type of inbound traffic that you want to reach the associated instances. In EC2-VPC, you can also change its outbound rules.

For more information about the types of rules you can add to security groups, see [Security Group Rules Reference](#) (p. 613).

Working with Security Groups

You can create, view, update, and delete security groups and security group rules using the Amazon EC2 console.

Contents

- [Creating a Security Group \(p. 610\)](#)
- [Describing Your Security Groups \(p. 610\)](#)
- [Adding Rules to a Security Group \(p. 611\)](#)
- [Deleting Rules from a Security Group \(p. 612\)](#)
- [Deleting a Security Group \(p. 612\)](#)
- [API and Command Overview \(p. 612\)](#)

Creating a Security Group

You can create a custom security group using the Amazon EC2 console. For EC2-VPC, you must specify the VPC for which you're creating the security group.

To create a new security group

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups**.
3. Choose **Create Security Group**.
4. Specify a name and description for the security group.
5. (EC2-Classic only) To create a security group for use in EC2-Classic, choose **No VPC**.
(EC2-VPC) For **VPC**, choose a VPC ID to create a security group for that VPC.
6. You can start adding rules, or you can choose **Create** to create the security group now (you can always add rules later). For more information about adding rules, see [Adding Rules to a Security Group \(p. 611\)](#).

The Amazon EC2 console enables you to copy the rules from an existing security group to a new security group.

To copy a security group

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups**.
3. Select the security group you want to copy, choose **Actions, Copy to new**.
4. The **Create Security Group** dialog opens, and is populated with the rules from the existing security group. Specify a name and description for your new security group. In the **VPC** list, choose **No VPC** to create a security group for EC2-Classic, or choose a VPC ID to create a security group for that VPC. When you are done, choose **Create**.

You can assign a security group to an instance when you launch the instance. When you add or remove rules, those changes are automatically applied to all instances to which you've assigned the security group.

After you launch an instance in EC2-Classic, you can't change its security groups. After you launch an instance in a VPC, you can change its security groups. For more information, see [Changing an Instance's Security Groups](#) in the *Amazon VPC User Guide*.

Describing Your Security Groups

To describe your security groups for EC2-Classic

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

2. In the navigation pane, choose **Security Groups**.
3. Select **Network Platforms** from the filter list, then choose **EC2-Classic**.
4. Select a security group. The **Description** tab displays general information. The **Inbound** tab displays the inbound rules.

To describe your security groups for EC2-VPC

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups**.
3. Select **Network Platforms** from the filter list, then choose **EC2-VPC**.
4. Select a security group. We display general information in the **Description** tab, inbound rules on the **Inbound** tab, and outbound rules on the **Outbound** tab.

Adding Rules to a Security Group

When you add a rule to a security group, the new rule is automatically applied to any instances associated with the security group.

For more information about choosing security group rules for specific types of access, see [Security Group Rules Reference](#) (p. 613).

To add rules to a security group

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
 2. In the navigation pane, choose **Security Groups** and select the security group.
 3. On the **Inbound** tab, choose **Edit**.
 4. In the dialog, choose **Add Rule** and do the following:
 - For **Type**, select the protocol.
 - If you select a custom TCP or UDP protocol, specify the port range in **Port Range**.
 - If you select a custom ICMP protocol, choose the ICMP type name from **Protocol**, and, if applicable, the code name from **Port Range**.
 - For **Source**, choose one of the following:
 - **Custom**: in the provided field, you must specify an IP address in CIDR notation, a CIDR block, or another security group.
 - **Anywhere**: automatically adds the 0.0.0.0/0 IPv4 CIDR block. This option enables all traffic of the specified type to reach your instance. This is acceptable for a short time in a test environment, but it's unsafe for production environments. In production, authorize only a specific IP address or range of addresses to access your instance.
- Note**
If your security group is in a VPC that's enabled for IPv6, the **Anywhere** option creates two rules—one for IPv4 traffic (0.0.0.0/0) and one for IPv6 traffic (:::/0).
- **My IP**: automatically adds the public IPv4 address of your local computer.

For more information about the types of rules that you can add, see [Security Group Rules Reference](#) (p. 613).

5. Choose **Save**.
6. For a VPC security group, you can also specify outbound rules. On the **Outbound** tab, choose **Edit**, **Add Rule**, and do the following:
 - For **Type**, select the protocol.

- If you select a custom TCP or UDP protocol, specify the port range in **Port Range**.
- If you select a custom ICMP protocol, choose the ICMP type name from **Protocol**, and, if applicable, the code name from **Port Range**.
- For **Destination**, choose one of the following:
 - **Custom**: in the provided field, you must specify an IP address in CIDR notation, a CIDR block, or another security group.
 - **Anywhere**: automatically adds the 0.0.0.0/0 IPv4 CIDR block. This option enables outbound traffic to all IP addresses.

Note

If your security group is in a VPC that's enabled for IPv6, the **Anywhere** option creates two rules—one for IPv4 traffic (0.0.0.0/0) and one for IPv6 traffic (:::/0).

- **My IP**: automatically adds the IP address of your local computer.
7. Choose **Save**.

Deleting Rules from a Security Group

When you delete a rule from a security group, the change is automatically applied to any instances associated with the security group.

To delete a security group rule

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups**.
3. Select a security group.
4. On the **Inbound** tab (for inbound rules) or **Outbound** tab (for outbound rules), choose **Edit**. Choose **Delete** (a cross icon) next to each rule to delete.
5. Choose **Save**.

Deleting a Security Group

You can't delete a security group that is associated with an instance. You can't delete the default security group. You can't delete a security group that is referenced by a rule in another security group in the same VPC. If your security group is referenced by one of its own rules, you must delete the rule before you can delete the security group.

To delete a security group

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups**.
3. Select a security group and choose **Actions, Delete Security Group**.
4. Choose **Yes, Delete**.

API and Command Overview

You can perform the tasks described on this page using the command line or an API. For more information about the command line interfaces and a list of available APIs, see [Accessing Amazon EC2 \(p. 3\)](#).

When you specify a security group for a nondefault VPC when using a command line tool, you must use the security group ID and not the security group name to identify the security group.

Create a security group

- [create-security-group](#) (AWS CLI)
- [New-EC2SecurityGroup](#) (AWS Tools for Windows PowerShell)

Add one or more ingress rules to a security group

- [authorize-security-group-ingress](#) (AWS CLI)
- [Grant-EC2SecurityGroupIngress](#) (AWS Tools for Windows PowerShell)

[EC2-VPC] Add one or more egress rules to a security group

- [authorize-security-group-egress](#) (AWS CLI)
- [Grant-EC2SecurityGroupIngress](#) (AWS Tools for Windows PowerShell)

Describe one or more security groups

- [describe-security-groups](#) (AWS CLI)
- [Get-EC2SecurityGroup](#) (AWS Tools for Windows PowerShell)

[EC2-VPC] Modify the security groups for an instance

- [modify-instance-attribute](#) (AWS CLI)
- [Edit-EC2InstanceAttribute](#) (AWS Tools for Windows PowerShell)

Remove one or more ingress rules from a security group

- [revoke-security-group-ingress](#) (AWS CLI)
- [Revoke-EC2SecurityGroupIngress](#) (AWS Tools for Windows PowerShell)

[EC2-VPC] Remove one or more egress rules from a security group

- [revoke-security-group-egress](#) (AWS CLI)
- [Revoke-EC2SecurityGroupEgress](#) (AWS Tools for Windows PowerShell)

Delete a security group

- [delete-security-group](#) (AWS CLI)
- [Remove-EC2SecurityGroup](#) (AWS Tools for Windows PowerShell)

Security Group Rules Reference

You can create a security group and add rules that reflect the role of the instance that's associated with the security group. For example, an instance that's configured as a web server needs security group rules that allow inbound HTTP and HTTPS access, and a database instance needs rules that allow access for the type of database, such as access over port 3306 for MySQL.

The following are examples of the kinds of rules that you can add to security groups for specific kinds of access.

Topics

- [Web server \(p. 614\)](#)
- [Database server \(p. 614\)](#)
- [Access from another instance in the same group \(p. 616\)](#)
- [Access from local computer \(p. 616\)](#)
- [Path MTU Discovery \(p. 617\)](#)
- [Ping your instance \(p. 617\)](#)
- [DNS server \(p. 617\)](#)
- [Amazon EFS file system \(p. 618\)](#)
- [Elastic Load Balancing \(p. 618\)](#)

Web server

The following inbound rules allow HTTP and HTTPS access from any IP address. If your VPC is enabled for IPv6, you can add rules to control inbound HTTP and HTTPS traffic from IPv6 addresses.

Protocol type	Protocol number	Port	Source IP	Notes
TCP	6	80 (HTTP)	0.0.0.0/0	Allows inbound HTTP access from any IPv4 address
TCP	6	443 (HTTPS)	0.0.0.0/0	Allows inbound HTTPS access from any IPv4 address
TCP	6	80 (HTTP)	:::0	(VPC only) Allows inbound HTTP access from any IPv6 address
TCP	6	443 (HTTPS)	:::0	(VPC only) Allows inbound HTTPS access from any IPv6 address

Database server

The following inbound rules are examples of rules you might add for database access, depending on what type of database you're running on your instance. For more information about Amazon RDS instances, see the [Amazon Relational Database Service User Guide](#).

For the source IP, specify one of the following:

- A specific IP address or range of IP addresses in your local network
- A security group ID for a group of instances that access the database

Protocol type	Protocol number	Port	Notes
TCP	6	1433 (MS SQL)	The default port to access a Microsoft SQL Server database,

Protocol type	Protocol number	Port	Notes
			for example, on an Amazon RDS instance
TCP	6	3306 (MYSQL/Aurora)	The default port to access a MySQL or Aurora database, for example, on an Amazon RDS instance
TCP	6	5439 (Redshift)	The default port to access an Amazon Redshift cluster database.
TCP	6	5432 (PostgreSQL)	The default port to access a PostgreSQL database, for example, on an Amazon RDS instance
TCP	6	1521 (Oracle)	The default port to access an Oracle database, for example, on an Amazon RDS instance

(VPC only) You can optionally restrict outbound traffic from your database servers, for example, if you want allow access to the Internet for software updates, but restrict all other kinds of traffic. You must first remove the default outbound rule that allows all outbound traffic.

Protocol type	Protocol number	Port	Destination IP	Notes
TCP	6	80 (HTTP)	0.0.0.0/0	Allows outbound HTTP access to any IPv4 address
TCP	6	443 (HTTPS)	0.0.0.0/0	Allows outbound HTTPS access to any IPv4 address
TCP	6	80 (HTTP)	:::0	(IPv6-enabled VPC only) Allows outbound HTTP access to any IPv6 address
TCP	6	443 (HTTPS)	:::0	(IPv6-enabled VPC only) Allows outbound HTTPS access to any IPv6 address

Access from another instance in the same group

To allow instances that are associated with the same security group to communicate with each other, you must explicitly add rules for this.

The following table describes the inbound rule for a VPC security group that enables associated instances to communicate with each other. The rule allows all types of traffic.

Protocol type	Protocol number	Ports	Source IP
-1 (All)	-1 (All)	-1 (All)	The ID of the security group

The following table describes inbound rules for an EC2-Classic security group that enable associated instances to communicate with each other. The rules allow all types of traffic.

Protocol type	Protocol number	Ports	Source IP
ICMP	1	-1 (All)	The ID of the security group
TCP	6	0 - 65535 (All)	The ID of the security group
UDP	17	0 - 65535 (All)	The ID of the security group

Access from local computer

To connect to your instance, your security group must have inbound rules that allow SSH access (for Linux instances) or RDP access (for Windows instances).

Protocol type	Protocol number	Port	Source IP
TCP	6	22 (SSH)	The public IPv4 address of your computer, or a range of IP addresses in your local network. If your VPC is enabled for IPv6 and your instance has an IPv6 address, you can enter an IPv6 address or range.
TCP	6	3389 (RDP)	The public IPv4 address of your computer, or a range of IP addresses in your local network. If your VPC is enabled for IPv6 and your instance has an IPv6 address, you can enter an IPv6 address or range.

Path MTU Discovery

The path MTU is the maximum packet size that's supported on the path between the originating host and the receiving host. If a host sends a packet that's larger than the MTU of the receiving host or that's larger than the MTU of a device along the path, the receiving host returns the following ICMP message:

```
Destination Unreachable: Fragmentation Needed and Don't Fragment was Set
```

To ensure that your instance can receive this message and the packet does not get dropped, you must add an ICMP rule to your inbound security group rules.

Protocol type	Protocol number	ICMP type	ICMP code	Source IP
ICMP	1	3 (Destination Unreachable)	4 (Fragmentation Needed and Don't Fragment was Set)	The IP addresses of the hosts that communicate with your instance

Ping your instance

The `ping` command is a type of ICMP traffic. To ping your instance, you must add the following inbound ICMP rule.

Protocol type	Protocol number	ICMP type	ICMP code	Source IP
ICMP	1	8 (Echo)	N/A	The public IPv4 address of your computer, or a range of IPv4 addresses in your local network

To use the `ping6` command to ping the IPv6 address for your instance, you must add the following inbound ICMPv6 rule.

Protocol type	Protocol number	ICMP type	ICMP code	Source IP
ICMPv6	58	128 (Echo)	0	The IPv6 address of your computer, or a range of IPv6 addresses in your local network

DNS server

If you've set up your EC2 instance as a DNS server, you must ensure that TCP and UDP traffic can reach your DNS server over port 53.

For the source IP, specify one of the following:

- A specific IP address or range of IP addresses in a network
- A security group ID for a group of instances in your network that require access to the DNS server

Protocol type	Protocol number	Port
TCP	6	53
UDP	17	53

Amazon EFS file system

If you're mounting and accessing an Amazon EFS file system from your Amazon EC2 instances, your security group rules must allow access over the NFS protocol.

Protocol type	Protocol number	Ports	Source IP	Notes
TCP	6	2049 (NFS)	The ID of the security group.	Allows inbound NFS access from resources (including the mount target) associated with this security group.
TCP	6	22 (SSH)	The IP address range of your local computer, or the range of IP addresses for your network.	Allows inbound SSH access from your local computer.

Elastic Load Balancing

If you're using a load balancer, the security group associated with your load balancer must have rules that allow communication with your instances or targets.

Inbound				
Protocol type	Protocol number	Port	Source IP	Notes
TCP	6	The listener port	For an Internet-facing load-balancer: 0.0.0.0/0 (all IPv4 addresses) For an internal load-balancer: the IPv4 CIDR block of the VPC	Allow inbound traffic on the load balancer listener port.
Outbound				
Protocol type	Protocol number	Port	Destination IP	Notes
TCP	6	The instance listener port	The ID of the instance security group	Allow outbound traffic to instances

				on the instance listener port.
TCP	6	The health check port	The ID of the instance security group	Allow outbound traffic to instances on the health check port.

The security group rules for your instances must allow the load balancer to communicate with your instances on both the listener port and the health check port.

Inbound				
Protocol type	Protocol number	Port	Source IP	Notes
TCP	6	The instance listener port	The ID of the load balancer security group	Allow traffic from the load balancer on the instance listener port.
TCP	6	The health check port	The ID of the load balancer security group	Allow traffic from the load balancer on the health check port.

For more information, see [Configure Security Groups for Your Classic Load Balancer](#) in the *Classic Load Balancer Guide*, and [Security Groups for Your Application Load Balancer](#) in the *Application Load Balancer Guide*.

Controlling Access to Amazon EC2 Resources

Your security credentials identify you to services in AWS and grant you unlimited use of your AWS resources, such as your Amazon EC2 resources. You can use features of Amazon EC2 and AWS Identity and Access Management (IAM) to allow other users, services, and applications to use your Amazon EC2 resources without sharing your security credentials. You can use IAM to control how other users use resources in your AWS account, and you can use security groups to control access to your Amazon EC2 instances. You can choose to allow full use or limited use of your Amazon EC2 resources.

Contents

- [Network Access to Your Instance](#) (p. 619)
- [Amazon EC2 Permission Attributes](#) (p. 620)
- [IAM and Amazon EC2](#) (p. 620)
- [IAM Policies for Amazon EC2](#) (p. 621)
- [IAM Roles for Amazon EC2](#) (p. 658)
- [Authorizing Inbound Traffic for Your Windows Instances](#) (p. 663)

Network Access to Your Instance

A security group acts as a firewall that controls the traffic allowed to reach one or more instances. When you launch an instance, you assign it one or more security groups. You add rules to each

security group that control traffic for the instance. You can modify the rules for a security group at any time; the new rules are automatically applied to all instances to which the security group is assigned.

For more information, see [Authorizing Inbound Traffic for Your Windows Instances \(p. 663\)](#).

Amazon EC2 Permission Attributes

Your organization might have multiple AWS accounts. Amazon EC2 enables you to specify additional AWS accounts that can use your Amazon Machine Images (AMIs) and Amazon EBS snapshots. These permissions work at the AWS account level only; you can't restrict permissions for specific users within the specified AWS account. All users in the AWS account that you've specified can use the AMI or snapshot.

Each AMI has a `LaunchPermission` attribute that controls which AWS accounts can access the AMI. For more information, see [Making an AMI Public \(p. 70\)](#).

Each Amazon EBS snapshot has a `createVolumePermission` attribute that controls which AWS accounts can use the snapshot. For more information, see [Sharing an Amazon EBS Snapshot \(p. 794\)](#).

IAM and Amazon EC2

IAM enables you to do the following:

- Create users and groups under your AWS account
- Assign unique security credentials to each user under your AWS account
- Control each user's permissions to perform tasks using AWS resources
- Allow the users in another AWS account to share your AWS resources
- Create roles for your AWS account and define the users or services that can assume them
- Use existing identities for your enterprise to grant permissions to perform tasks using AWS resources

By using IAM with Amazon EC2, you can control whether users in your organization can perform a task using specific Amazon EC2 API actions and whether they can use specific AWS resources.

This topic helps you answer the following questions:

- How do I create groups and users in IAM?
- How do I create a policy?
- What IAM policies do I need to carry out tasks in Amazon EC2?
- How do I grant permissions to perform actions in Amazon EC2?
- How do I grant permissions to perform actions on specific resources in Amazon EC2?

Creating an IAM Group and Users

To create an IAM group

1. Sign in to the Identity and Access Management (IAM) console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Groups** and then choose **Create New Group**.
3. In the **Group Name** box, enter a name for your group, and then choose **Next Step**.
4. On the **Attach Policy** page, select an AWS managed policy. For example, for Amazon EC2, one of the following AWS managed policies might meet your needs:

- PowerUserAccess
 - ReadOnlyAccess
 - AmazonEC2FullAccess
 - AmazonEC2ReadOnlyAccess
5. Choose **Next Step** and then choose **Create Group**.

Your new group is listed under **Group Name**.

To create an IAM user, add the user to your group, and create a password for the user

1. In the navigation pane, choose **Users** and then choose **Add user**.
2. Enter a user name.
3. Select the type of access this set of users will have. Select both **Programmatic access** and **AWS Management Console access**.
4. For **Console password type**, choose one of the following:
 - **Autogenerated password**. Each user gets a randomly generated password that meets the current password policy in effect (if any). You can view or download the passwords when you get to the **Final** page.
 - **Custom password**. Each user is assigned the password that you type in the box.
5. Choose **Next: Permissions**.
6. On the **Set permissions** page, choose **Add user to group**. Select the group you created earlier.
7. Choose **Next: Review**, then **Create user**.
8. To view the users' access keys (access key IDs and secret access keys), choose **Show** next to each password and secret access key that you want to see. To save the access keys, choose **Download .csv** and then save the file to a safe location.

Note

You cannot retrieve the secret access key after you complete this step; if you misplace it you must create a new one.

9. Choose **Close**.
10. Give each user his or her credentials (access keys and password); this enables them to use services based on the permissions you specified for the IAM group.

Related Topics

For more information about IAM, see the following:

- [IAM Policies for Amazon EC2 \(p. 621\)](#)
- [IAM Roles for Amazon EC2 \(p. 658\)](#)
- [Identity and Access Management \(IAM\)](#)
- [IAM User Guide](#)

IAM Policies for Amazon EC2

By default, IAM users don't have permission to create or modify Amazon EC2 resources, or perform tasks using the Amazon EC2 API. (This means that they also can't do so using the Amazon EC2 console or CLI.) To allow IAM users to create or modify resources and perform tasks, you must create IAM policies that grant IAM users permission to use the specific resources and API actions they'll need, and then attach those policies to the IAM users or groups that require those permissions.

When you attach a policy to a user or group of users, it allows or denies the users permission to perform the specified tasks on the specified resources. For more general information about IAM policies, see [Permissions and Policies](#) in the *IAM User Guide*. For more information about managing and creating custom IAM policies, see [Managing IAM Policies](#).

Getting Started

An IAM policy must grant or deny permission to use one or more Amazon EC2 actions. It must also specify the resources that can be used with the action, which can be all resources, or in some cases, specific resources. The policy can also include conditions that you apply to the resource.

Amazon EC2 partially supports resource-level permissions. This means that for some EC2 API actions, you cannot specify which resource a user is allowed to work with for that action; instead, you have to allow users to work with all resources for that action.

Task	Topic
Understand the basic structure of a policy	Policy Syntax (p. 622)
Define actions in your policy	Actions for Amazon EC2 (p. 623)
Define specific resources in your policy	Amazon Resource Names for Amazon EC2 (p. 623)
Apply conditions to the use of the resources	Condition Keys for Amazon EC2 (p. 626)
Work with the available resource-level permissions for Amazon EC2	Supported Resource-Level Permissions for Amazon EC2 API Actions (p. 629)
Test your policy	Checking that Users Have the Required Permissions (p. 628)
Example policies for a CLI or SDK	Example Policies for Working With the AWS CLI or an AWS SDK (p. 642)
Example policies for the Amazon EC2 console	Example Policies for Working in the Amazon EC2 Console (p. 652)

Policy Structure

The following topics explain the structure of an IAM policy.

Topics

- [Policy Syntax \(p. 622\)](#)
- [Actions for Amazon EC2 \(p. 623\)](#)
- [Amazon Resource Names for Amazon EC2 \(p. 623\)](#)
- [Condition Keys for Amazon EC2 \(p. 626\)](#)
- [Checking that Users Have the Required Permissions \(p. 628\)](#)

Policy Syntax

An IAM policy is a JSON document that consists of one or more statements. Each statement is structured as follows:

```
{
  "Statement": [ {
    "Effect": "effect",
```

```
"Action": "action",  
"Resource": "arn",  
"Condition": {  
  "condition": {  
    "key": "value"  
  }  
}
```

There are various elements that make up a statement:

- **Effect:** The *effect* can be `Allow` or `Deny`. By default, IAM users don't have permission to use resources and API actions, so all requests are denied. An explicit allow overrides the default. An explicit deny overrides any allows.
- **Action:** The *action* is the specific API action for which you are granting or denying permission. To learn about specifying *action*, see [Actions for Amazon EC2 \(p. 623\)](#).
- **Resource:** The resource that's affected by the action. Some Amazon EC2 API actions allow you to include specific resources in your policy that can be created or modified by the action. To specify a resource in the statement, you need to use its Amazon Resource Name (ARN). For more information about specifying the *arn* value, see [Amazon Resource Names for Amazon EC2 \(p. 623\)](#). For more information about which API actions support which ARNs, see [Supported Resource-Level Permissions for Amazon EC2 API Actions \(p. 629\)](#). If the API action does not support ARNs, use the * wildcard to specify that all resources can be affected by the action.
- **Condition:** Conditions are optional. They can be used to control when your policy will be in effect. For more information about specifying conditions for Amazon EC2, see [Condition Keys for Amazon EC2 \(p. 626\)](#).

For more information about example IAM policy statements for Amazon EC2, see [Example Policies for Working With the AWS CLI or an AWS SDK \(p. 642\)](#).

Actions for Amazon EC2

In an IAM policy statement, you can specify any API action from any service that supports IAM. For Amazon EC2, use the following prefix with the name of the API action: `ec2:`. For example: `ec2:RunInstances` and `ec2:CreateImage`.

To specify multiple actions in a single statement, separate them with commas as follows:

```
"Action": ["ec2:action1", "ec2:action2"]
```

You can also specify multiple actions using wildcards. For example, you can specify all actions whose name begins with the word "Describe" as follows:

```
"Action": "ec2:Describe*"
```

To specify all Amazon EC2 API actions, use the * wildcard as follows:

```
"Action": "ec2:*"
```

For a list of Amazon EC2 actions, see [Actions](#) in the *Amazon EC2 API Reference*.

Amazon Resource Names for Amazon EC2

Each IAM policy statement applies to the resources that you specify using their ARNs.

Important

Currently, not all API actions support individual ARNs; we'll add support for additional API actions and ARNs for additional Amazon EC2 resources later. For information about which ARNs you can use with which Amazon EC2 API actions, as well as supported condition keys for each ARN, see [Supported Resource-Level Permissions for Amazon EC2 API Actions](#) (p. 629).

An ARN has the following general syntax:

```
arn:aws:[service]:[region]:[account]:resourceType/resourcePath
```

service

The service (for example, *ec2*).

region

The region for the resource (for example, *us-east-1*).

account

The AWS account ID, with no hyphens (for example, *123456789012*).

resourceType

The type of resource (for example, *instance*).

resourcePath

A path that identifies the resource. You can use the *** wildcard in your paths.

For example, you can indicate a specific instance (*i-1234567890abcdef0*) in your statement using its ARN as follows:

```
"Resource": "arn:aws:ec2:us-east-1:123456789012:instance/i-1234567890abcdef0"
```

You can also specify all instances that belong to a specific account by using the *** wildcard as follows:

```
"Resource": "arn:aws:ec2:us-east-1:123456789012:instance/*"
```

To specify all resources, or if a specific API action does not support ARNs, use the *** wildcard in the *Resource* element as follows:

```
"Resource": "*"
```

The following table describes the ARNs for each type of resource used by the Amazon EC2 API actions.

Resource Type	ARN
All Amazon EC2 resources	<code>arn:aws:ec2:*</code>
All Amazon EC2 resources owned by the specified account in the specified region	<code>arn:aws:ec2:region:account:*</code>
Customer gateway	<code>arn:aws:ec2:region:account:customer-gateway/cgw-id</code> Where <i>cgw-id</i> is <i>cgw-xxxxxxx</i>
DHCP options set	<code>arn:aws:ec2:region:account:dhcp-options/dhcp-options-id</code> Where <i>dhcp-options-id</i> is <i>dopt-xxxxxxx</i>

Resource Type	ARN
Image	arn:aws:ec2:region::image/image-id Where <i>image-id</i> is the ID of the AMI, AKI, or ARI, and <i>account</i> isn't used
Instance	arn:aws:ec2:region:account:instance/instance-id Where <i>instance-id</i> is i-xxxxxxx or i-xxxxxxxxxxxxxxxxxxx
Instance profile	arn:aws:iam::account:instance-profile/instance-profile-name Where <i>instance-profile-name</i> is the name of the instance profile, and <i>region</i> isn't used
Internet gateway	arn:aws:ec2:region:account:internet-gateway/igw-id Where <i>igw-id</i> is igw-xxxxxxx
Key pair	arn:aws:ec2:region:account:key-pair/key-pair-name Where <i>key-pair-name</i> is the key pair name (for example, gsg-keypair)
Network ACL	arn:aws:ec2:region:account:network-acl/nacl-id Where <i>nacl-id</i> is acl-xxxxxxx
Network interface	arn:aws:ec2:region:account:network-interface/eni-id Where <i>eni-id</i> is eni-xxxxxxx
Placement group	arn:aws:ec2:region:account:placement-group/placement-group-name Where <i>placement-group-name</i> is the placement group name (for example, my-cluster)
Route table	arn:aws:ec2:region:account:route-table/route-table-id Where <i>route-table-id</i> is rtb-xxxxxxx
Security group	arn:aws:ec2:region:account:security-group/security-group-id Where <i>security-group-id</i> is sg-xxxxxxx
Snapshot	arn:aws:ec2:region::snapshot/snapshot-id Where <i>snapshot-id</i> is snap-xxxxxxx or snap-xxxxxxxxxxxxxxxxxxx, and <i>account</i> isn't used
Subnet	arn:aws:ec2:region:account:subnet/subnet-id Where <i>subnet-id</i> is subnet-xxxxxxx
Volume	arn:aws:ec2:region:account:volume/volume-id Where <i>volume-id</i> is vol-xxxxxxx or vol-xxxxxxxxxxxxxxxxxxx

Resource Type	ARN
VPC	arn:aws:ec2:region:account:vpc/vpc-id Where <i>vpc-id</i> is vpc-xxxxxxx
VPC peering connection	arn:aws:ec2:region:account:vpc-peering-connection/vpc-peering-connection-id Where <i>vpc-peering connection-id</i> is pcx-xxxxxxx

Many Amazon EC2 API actions involve multiple resources. For example, `AttachVolume` attaches an Amazon EBS volume to an instance, so an IAM user must have permission to use the volume and the instance. To specify multiple resources in a single statement, separate their ARNs with commas, as follows:

```
"Resource": [ "arn1", "arn2" ]
```

For more general information about ARNs, see [Amazon Resource Names \(ARN\) and AWS Service Namespaces](#) in the *Amazon Web Services General Reference*. For more information about the resources that are created or modified by the Amazon EC2 actions, and the ARNs that you can use in your IAM policy statements, see [Granting IAM Users Required Permissions for Amazon EC2 Resources](#) in the *Amazon EC2 API Reference*.

Condition Keys for Amazon EC2

In a policy statement, you can optionally specify conditions that control when it is in effect. Each condition contains one or more key-value pairs. Condition keys are not case sensitive. We've defined AWS-wide condition keys, plus additional service-specific condition keys.

If you specify multiple conditions, or multiple keys in a single condition, we evaluate them using a logical AND operation. If you specify a single condition with multiple values for one key, we evaluate the condition using a logical OR operation. For permission to be granted, all conditions must be met.

You can also use placeholders when you specify conditions. For example, you can grant an IAM user permission to use resources with a tag that specifies his or her IAM user name. For more information, see [Policy Variables](#) in the *IAM User Guide*.

Amazon EC2 implements the AWS-wide condition keys (see [Available Keys](#)), plus the following service-specific condition keys. (We'll add support for additional service-specific condition keys for Amazon EC2 later.)

Important

Many condition keys are specific to a resource, and some API actions use multiple resources. If you write a policy with a condition key, use the `Resource` element of the statement to specify the resource to which the condition key applies. If not, the policy may prevent users from performing the action at all, because the condition check fails for the resources to which the condition key does not apply. If you do not want to specify a resource, or if you've written the `Action` element of your policy to include multiple API actions, then you must use the `...IfExists` condition type to ensure that the condition key is ignored for resources that do not use it. For more information, see [...IfExists Conditions](#) in the *IAM User Guide*.

Condition Key	Key/Value Pair	Evaluation Types
ec2:AccepterVpc	"ec2:AccepterVpc": "vpc-arn" Where <i>vpc-arn</i> is the VPC ARN for the peer VPC	ARN, Null

Condition Key	Key/Value Pair	Evaluation Types
ec2:AvailabilityZone	"ec2:AvailabilityZone": " <i>az-api-name</i> " Where <i>az-api-name</i> is the name of the Availability Zone (for example, <i>us-west-2a</i>) To list your Availability Zones, use describe-availability-zones	String, Null
ec2:EbsOptimized	"ec2:EbsOptimized": " <i>optimized-flag</i> " Where <i>optimized-flag</i> is <code>true</code> <code>false</code>	Boolean, Null
ec2:ImageType	"ec2:ImageType": " <i>image-type-api-name</i> " Where <i>image-type-api-name</i> is <code>ami</code> <code>aki</code> <code>ari</code>	String, Null
ec2:InstanceProfile	"ec2:InstanceProfile": " <i>instance-profile-arn</i> " Where <i>instance-profile-arn</i> is the instance profile ARN	ARN, Null
ec2:InstanceType	"ec2:InstanceType": " <i>instance-type-api-name</i> " Where <i>instance-type-api-name</i> is the name of the instance type.	String, Null
ec2:Owner	"ec2:Owner": " <i>account-id</i> " Where <i>account-id</i> is <code>amazon</code> <code>aws-marketplace</code> <code>aws-account-id</code>	String, Null
ec2:ParentSnapshot	"ec2:ParentSnapshot": " <i>snapshot-arn</i> " Where <i>snapshot-arn</i> is the snapshot ARN	ARN, Null
ec2:ParentVolume	"ec2:ParentVolume": " <i>volume-arn</i> " Where <i>volume-arn</i> is the volume ARN	ARN, Null
ec2:PlacementGroup	"ec2:PlacementGroup": " <i>placement-group-arn</i> " Where <i>placement-group-arn</i> is the placement group ARN	ARN, Null
ec2:PlacementGroupStrategy	"ec2:PlacementGroupStrategy": " <i>placement-group-strategy</i> " Where <i>placement-group-strategy</i> is <code>cluster</code>	String, Null
ec2:ProductCode	"ec2:ProductCode": " <i>product-code</i> " Where <i>product-code</i> is the product code	String, Null
ec2:Public	"ec2:Public": " <i>public-flag</i> " Where <i>public-flag</i> for an AMI is <code>true</code> <code>false</code>	Boolean, Null
ec2:Region	"ec2:Region": " <i>region-name</i> " Where <i>region-name</i> is the name of the region (for example, <i>us-west-2</i>). To list your regions, use describe-regions .	String, Null

Condition Key	Key/Value Pair	Evaluation Types
ec2:RequesterVpc	"ec2:RequesterVpc": " <i>vpc-arn</i> " Where <i>vpc-arn</i> is the VPC ARN for the requester's VPC	ARN, Null
ec2:ResourceTag/ <i>tag-key</i>	"ec2:ResourceTag/ <i>tag-key</i> ": " <i>tag-value</i> " Where <i>tag-key</i> and <i>tag-value</i> are the tag-key pair	String, Null
ec2:RootDeviceType	"ec2:RootDeviceType": " <i>root-device-type-name</i> " Where <i>root-device-type-name</i> is <i>ebs</i> <i>instance-store</i>	String, Null
ec2:Subnet	"ec2:Subnet": " <i>subnet-arn</i> " Where <i>subnet-arn</i> is the subnet ARN	ARN, Null
ec2:Tenancy	"ec2:Tenancy": " <i>tenancy-attribute</i> " Where <i>tenancy-attribute</i> is <i>default</i> <i>dedicated</i> <i>host</i>	String, Null
ec2:Volumelops	"ec2:Volumelops": " <i>volume-iops</i> " Where <i>volume-iops</i> is the input/output operations per second (IOPS); the range is 100 to 20,000	Numeric, Null
ec2:VolumeSize	"ec2:VolumeSize": " <i>volume-size</i> " Where <i>volume-size</i> is the size of the volume, in GiB	Numeric, Null
ec2:VolumeType	"ec2:VolumeType": " <i>volume-type-name</i> " Where <i>volume-type-name</i> is <i>gp2</i> for General Purpose SSD volumes, <i>io1</i> for Provisioned IOPS SSD volumes, <i>st1</i> for Throughput Optimized HDD volumes, <i>sc1</i> for Cold HDD volumes, or <i>standard</i> for Magnetic volumes.	String, Null
ec2:Vpc	"ec2:Vpc": " <i>vpc-arn</i> " Where <i>vpc-arn</i> is the VPC ARN	ARN, Null

For information about which condition keys you can use with which Amazon EC2 resources, on an action-by-action basis, see [Supported Resource-Level Permissions for Amazon EC2 API Actions](#) (p. 629). For example policy statements for Amazon EC2, see [Example Policies for Working With the AWS CLI or an AWS SDK](#) (p. 642).

Checking that Users Have the Required Permissions

After you've created an IAM policy, we recommend that you check whether it grants users the permissions to use the particular API actions and resources they need before you put the policy into production.

First, create an IAM user for testing purposes, and then attach the IAM policy that you created to the test user. Then, make a request as the test user.

If the action that you are testing creates or modifies a resource, you should make the request using the `DryRun` parameter (or run the CLI command with the `--auth-dry-run` option). In this case, the call completes the authorization check, but does not complete the operation. For example, you can check whether the user can terminate a particular instance without actually terminating it. If the

test user has the required permissions, the request returns `DryRunOperation`; otherwise, it returns `UnauthorizedOperation`.

If the policy doesn't grant the user the permissions that you expected, or is overly permissive, you can adjust the policy as needed and retest until you get the desired results.

Important

It can take several minutes for policy changes to propagate before they take effect. Therefore, we recommend that you allow five minutes to pass before you test your policy updates.

If an authorization check fails, the request returns an encoded message with diagnostic information. You can decode the message using the `DecodeAuthorizationMessage` action. For more information, see [DecodeAuthorizationMessage](#) in the *AWS Security Token Service API Reference*, and [decode-authorization-message](#) in the *AWS Command Line Interface Reference*.

Supported Resource-Level Permissions for Amazon EC2 API Actions

Resource-level permissions refers to the ability to specify which resources users are allowed to perform actions on. Amazon EC2 has partial support for resource-level permissions. This means that for certain Amazon EC2 actions, you can control when users are allowed to use those actions based on conditions that have to be fulfilled, or specific resources that users are allowed to use. For example, you can grant users permission to launch instances, but only of a specific type, and only using a specific AMI.

The following table describes the Amazon EC2 API actions that currently support resource-level permissions, as well as the supported resources (and their ARNs) and condition keys for each action. When specifying an ARN, you can use the `*` wildcard in your paths; for example, when you cannot or do not want to specify exact resource IDs. For examples of using wildcards, see [Example Policies for Working With the AWS CLI or an AWS SDK](#) (p. 642).

Important

If an Amazon EC2 API action is not listed in this table, then it does not support resource-level permissions. If an Amazon EC2 API action does not support resource-level permissions, you can grant users permission to use the action, but you have to specify a `*` for the resource element of your policy statement. For an example of how to do this, see [1: Read-only access](#) (p. 642). We'll add support for additional actions, ARNs, and condition keys later. For a list of Amazon EC2 API actions that currently do not support resource-level permissions, see [Unsupported Resource-Level Permissions](#) in the *Amazon EC2 API Reference*.

API Action	Resources	Condition Keys
AcceptVpcPeeringConnections	VPC peering connection	ec2:AcceptorVpc
	arn:aws:ec2:region:account:vpc-peering-connection/*	ec2:Region
	arn:aws:ec2:region:account:vpc-peering-connection/vpc-peering-connection-id	ec2:ResourceTag/tag-key
	VPC	ec2:RequesterVpc
	arn:aws:ec2:region:account:vpc/*	ec2:ResourceTag/tag-key
	arn:aws:ec2:region:account:vpc/vpc-id	ec2:Region
	Where <i>vpc-id</i> is a VPC owned by the acceptor.	ec2:Tenancy

Amazon Elastic Compute Cloud
User Guide for Windows Instances
IAM Policies

API Action	Resources	Condition Keys
AttachClassicLinkVpc	Instance arn:aws:ec2:region:account:instance/* arn:aws:ec2:region:account:instance/ instance-id	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/tag-key ec2:RootDeviceType ec2:Tenancy
	Security group arn:aws:ec2:region:account:security- group/* arn:aws:ec2:region:account:security- group/security-group-id Where the security group is the security group for the VPC.	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
	VPC arn:aws:ec2:region:account:vpc/* arn:aws:ec2:region:account:vpc/vpc-id	ec2:Region ec2:ResourceTag/tag-key ec2:Tenancy
AttachVolume	Instance arn:aws:ec2:region:account:instance/* arn:aws:ec2:region:account:instance/ instance-id	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/tag-key ec2:RootDeviceType ec2:Tenancy

API Action	Resources	Condition Keys
	Volume arn:aws:ec2:region:account:volume/* arn:aws:ec2:region:account:volume/ volume-id	ec2:AvailabilityZone ec2:ParentSnapshot ec2:Region ec2:ResourceTag/tag-key ec2:Volumelops ec2:VolumeSize ec2:VolumeType
AuthorizeSecurityGroupEgress	Security group arn:aws:ec2:region:account:security- group/* arn:aws:ec2:region:account:security- group/security-group-id	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
AuthorizeSecurityGroupIngress	Security group arn:aws:ec2:region:account:security- group/* arn:aws:ec2:region:account:security- group/security-group-id	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
CreateVpcPeeringConnection	VPC arn:aws:ec2:region:account:vpc/* arn:aws:ec2:region:account:vpc/vpc-id Where <i>vpc-id</i> is a requester VPC.	ec2:ResourceTag/tag-key ec2:Region ec2:Tenancy
	VPC peering connection arn:aws:ec2:region:account:vpc- peering-connection/*	ec2:AccepterVpc ec2:Region ec2:RequesterVpc
DeleteCustomerGateway	Customer gateway arn:aws:ec2:region:account:customer- gateway/* arn:aws:ec2:region:account:customer- gateway/cgw-id	ec2:Region ec2:ResourceTag/tag-key
DeleteDhcpOptions	DHCP options set arn:aws:ec2:region:account:dhcp- options/* arn:aws:ec2:region:account:dhcp- options/dhcp-options-id	ec2:Region ec2:ResourceTag/tag-key

Amazon Elastic Compute Cloud
User Guide for Windows Instances
IAM Policies

API Action	Resources	Condition Keys
DeleteInternetGateway	Internet gateway arn:aws:ec2:region:account:internet-gateway/* arn:aws:ec2:region:account:internet-gateway/igw-id	ec2:Region ec2:ResourceTag/tag-key
DeleteNetworkAcl	Network ACL arn:aws:ec2:region:account:network-acl/* arn:aws:ec2:region:account:network-acl/nacl-id	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
DeleteNetworkAclEntry	Network ACL arn:aws:ec2:region:account:network-acl/* arn:aws:ec2:region:account:network-acl/nacl-id	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
DeleteRoute	Route table arn:aws:ec2:region:account:route-table/* arn:aws:ec2:region:account:route-table/route-table-id	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
DeleteRouteTable	Route table arn:aws:ec2:region:account:route-table/* arn:aws:ec2:region:account:route-table/route-table-id	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
DeleteSecurityGroup	Security group arn:aws:ec2:region:account:security-group/security-group-id	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
DeleteVolume	Volume arn:aws:ec2:region:account:volume/* arn:aws:ec2:region:account:volume/volume-id	ec2:AvailabilityZone ec2:ParentSnapshot ec2:Region ec2:ResourceTag/tag-key ec2:Volumelops ec2:VolumeSize ec2:VolumeType

Amazon Elastic Compute Cloud
User Guide for Windows Instances
IAM Policies

API Action	Resources	Condition Keys
DeleteVpcPeeringConnection	<p>VPC peering connection</p> <p>arn:aws:ec2:region:account:vpc-peering-connection/*</p> <p>arn:aws:ec2:region:account:vpc-peering-connection/vpc-peering-connection-id</p>	<p>ec2:AcceptorVpc</p> <p>ec2:Region</p> <p>ec2:ResourceTag/tag-key</p> <p>ec2:RequesterVpc</p>
DetachClassicLinkVpc	<p>Instance</p> <p>arn:aws:ec2:region:account:instance/*</p> <p>arn:aws:ec2:region:account:instance/instance-id</p>	<p>ec2:AvailabilityZone</p> <p>ec2:EbsOptimized</p> <p>ec2:InstanceProfile</p> <p>ec2:InstanceType</p> <p>ec2:PlacementGroup</p> <p>ec2:Region</p> <p>ec2:ResourceTag/tag-key</p> <p>ec2:RootDeviceType</p> <p>ec2:Tenancy</p>
	<p>VPC</p> <p>arn:aws:ec2:region:account:vpc/*</p> <p>arn:aws:ec2:region:account:vpc/vpc-id</p>	<p>ec2:Region</p> <p>ec2:ResourceTag/tag-key</p> <p>ec2:Tenancy</p>
DetachVolume	<p>Instance</p> <p>arn:aws:ec2:region:account:instance/*</p> <p>arn:aws:ec2:region:account:instance/instance-id</p>	<p>ec2:AvailabilityZone</p> <p>ec2:EbsOptimized</p> <p>ec2:InstanceProfile</p> <p>ec2:InstanceType</p> <p>ec2:PlacementGroup</p> <p>ec2:Region</p> <p>ec2:ResourceTag/tag-key</p> <p>ec2:RootDeviceType</p> <p>ec2:Tenancy</p>

Amazon Elastic Compute Cloud
User Guide for Windows Instances
IAM Policies

API Action	Resources	Condition Keys
	Volume arn:aws:ec2:region:account:volume/* arn:aws:ec2:region:account:volume/ volume-id	ec2:AvailabilityZone ec2:ParentSnapshot ec2:Region ec2:ResourceTag/tag-key ec2:Volumelops ec2:VolumeSize ec2:VolumeType
DisableVpcClassicLink	VPC arn:aws:ec2:region:account:vpc/* arn:aws:ec2:region:account:vpc/vpc-id	ec2:Region ec2:ResourceTag/tag-key ec2:Tenancy
EnableVpcClassicLink	VPC arn:aws:ec2:region:account:vpc/* arn:aws:ec2:region:account:vpc/vpc-id	ec2:Region ec2:ResourceTag/tag-key ec2:Tenancy
GetConsoleScreenshot	Instance arn:aws:ec2:region:account:instance/* arn:aws:ec2:region:account:instance/ instance-id	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/tag-key ec2:RootDeviceType ec2:Tenancy

Amazon Elastic Compute Cloud
User Guide for Windows Instances
IAM Policies

API Action	Resources	Condition Keys
RebootInstances	Instance arn:aws:ec2:region:account:instance/* arn:aws:ec2:region:account:instance/ instance-id	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/tag-key ec2:RootDeviceType ec2:Tenancy
RejectVpcPeeringConnections	VPC peering connection arn:aws:ec2:region:account:vpc- peering-connection/* arn:aws:ec2:region:account:vpc- peering-connection/vpc-peering- connection-id	ec2:AccepterVpc ec2:Region ec2:ResourceTag/tag-key ec2:RequesterVpc
RevokeSecurityGroupEgress	Security group arn:aws:ec2:region:account:security- group/* arn:aws:ec2:region:account:security- group/security-group-id	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
RevokeSecurityGroupIngress	Security group arn:aws:ec2:region:account:security- group/* arn:aws:ec2:region:account:security- group/security-group-id	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
RunInstances	Image arn:aws:ec2:region::image/* arn:aws:ec2:region::image/image-id	ec2:ImageType ec2:Owner ec2:Public ec2:Region ec2:RootDeviceType ec2:ResourceTag/tag-key

API Action	Resources	Condition Keys
	Instance <i>arn:aws:ec2:region:account:instance/*</i>	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:RootDeviceType ec2:Tenancy
	Key pair <i>arn:aws:ec2:region:account:key-pair/*</i> <i>arn:aws:ec2:region:account:key-pair/key-pair-name</i>	ec2:Region
	Network interface <i>arn:aws:ec2:region:account:network-interface/*</i> <i>arn:aws:ec2:region:account:network-interface/eni-id</i>	ec2:AvailabilityZone ec2:Region ec2:Subnet ec2:ResourceTag/tag-key ec2:Vpc
	Placement group <i>arn:aws:ec2:region:account:placement-group/*</i> <i>arn:aws:ec2:region:account:placement-group/placement-group-name</i>	ec2:Region ec2:PlacementGroupStrategy
	Security group <i>arn:aws:ec2:region:account:security-group/*</i> <i>arn:aws:ec2:region:account:security-group/security-group-id</i>	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
	Snapshot <i>arn:aws:ec2:region::snapshot/*</i> <i>arn:aws:ec2:region::snapshot/snapshot-id</i>	ec2:Owner ec2:ParentVolume ec2:Region ec2:SnapshotTime ec2:ResourceTag/tag-key ec2:VolumeSize

Amazon Elastic Compute Cloud
User Guide for Windows Instances
IAM Policies

API Action	Resources	Condition Keys
	Subnet arn:aws:ec2:region:account:subnet/* arn:aws:ec2:region:account:subnet/ subnet-id	ec2:AvailabilityZone ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
	Volume arn:aws:ec2:region:account:volume/*	ec2:AvailabilityZone ec2:ParentSnapshot ec2:Region ec2:Volumelops ec2:VolumeSize ec2:VolumeType
StartInstances	Instance arn:aws:ec2:region:account:instance/* arn:aws:ec2:region:account:instance/ instance-id	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/tag-key ec2:RootDeviceType ec2:Tenancy
StopInstances	Instance arn:aws:ec2:region:account:instance/* arn:aws:ec2:region:account:instance/ instance-id	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/tag-key ec2:RootDeviceType ec2:Tenancy

API Action	Resources	Condition Keys
TerminateInstances	Instance arn:aws:ec2:region:account:instance/* arn:aws:ec2:region:account:instance/ instance-id	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/tag-key ec2:RootDeviceType ec2:Tenancy

Resource-Level Permissions for RunInstances

The [RunInstances](#) API action launches one or more instances, and creates and uses a number of Amazon EC2 resources. The action requires an AMI and creates an instance; and the instance must be associated with a security group. Launching into a VPC requires a subnet, and creates a network interface. Launching from an Amazon EBS-backed AMI creates a volume. The user must have permission to use these resources, so they must be specified in the `Resource` element of any policy that uses resource-level permissions for the `ec2:RunInstances` action. If you don't intend to use resource-level permissions with the `ec2:RunInstances` action, you can specify the `*` wildcard in the `Resource` element of your statement instead of individual ARNs.

If you are using resource-level permissions, the following table describes the minimum resources required to use the `ec2:RunInstances` action.

Type of launch	Resources required	Condition keys
Launching into EC2-Classic using an instance store-backed AMI	arn:aws:ec2:region:account:instance/*	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:RootDeviceType ec2:Tenancy
	arn:aws:ec2:region::image/* (or a specific AMI ID)	ec2:ImageType ec2:Owner ec2:Public ec2:Region ec2:RootDeviceType

Amazon Elastic Compute Cloud
User Guide for Windows Instances
IAM Policies

Type of launch	Resources required	Condition keys
		ec2:ResourceTag/tag-key
	arn:aws:ec2:region:account:securitygroup/* (or a specific security group ID)	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
Launching into EC2-Classic using an Amazon EBS-backed AMI	arn:aws:ec2:region:account:instance*	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:RootDeviceType ec2:Tenancy
	arn:aws:ec2:region:image/* (or a specific AMI ID)	ec2:ImageType ec2:Owner ec2:Public ec2:Region ec2:RootDeviceType ec2:ResourceTag/tag-key
	arn:aws:ec2:region:account:securitygroup/* (or a specific security group ID)	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
	arn:aws:ec2:region:account:volume*	ec2:AvailabilityZone ec2:ParentSnapshot ec2:Region ec2:Volumeops ec2:VolumeSize ec2:VolumeType

Amazon Elastic Compute Cloud
User Guide for Windows Instances
IAM Policies

Type of launch	Resources required	Condition keys
Launching into a VPC using an instance store-backed AMI	arn:aws:ec2:region:account:instance/*	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:RootDeviceType ec2:Tenancy
	arn:aws:ec2:region::image/* (or a specific AMI ID)	ec2:ImageType ec2:Owner ec2:Public ec2:Region ec2:RootDeviceType ec2:ResourceTag/tag-key
	arn:aws:ec2:region:account:securitygroup/* (or a specific security group ID)	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
	arn:aws:ec2:region:account:networkinterface/* (or a specific network interface ID)	ec2:AvailabilityZone ec2:Region ec2:Subnet ec2:ResourceTag/tag-key ec2:Vpc
	arn:aws:ec2:region:account:subnet/* (or a specific subnet ID)	ec2:AvailabilityZone ec2:Region ec2:ResourceTag/tag-key ec2:Vpc

Amazon Elastic Compute Cloud
User Guide for Windows Instances
IAM Policies

Type of launch	Resources required	Condition keys
Launching into a VPC using an Amazon EBS-backed AMI	arn:aws:ec2:region:account:instance*	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:RootDeviceType ec2:Tenancy
	arn:aws:ec2:region::image/* (or a specific AMI ID)	ec2:ImageType ec2:Owner ec2:Public ec2:Region ec2:RootDeviceType ec2:ResourceTag/tag-key
	arn:aws:ec2:region:account:securitygroup/* (or a specific security group ID)	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
	arn:aws:ec2:region:account:networkinterface/* (or a specific network interface ID)	ec2:AvailabilityZone ec2:Region ec2:Subnet ec2:ResourceTag/tag-key ec2:Vpc
	arn:aws:ec2:region:account:volume*	ec2:AvailabilityZone ec2:ParentSnapshot ec2:Region ec2:Volumeops ec2:VolumeSize ec2:VolumeType

Type of launch	Resources required	Condition keys
	arn:aws:ec2:region:account:subnet* (or a specific subnet ID)	ec2:AvailabilityZone ec2:Region ec2:ResourceTag/tag-key ec2:Vpc

We recommend that you also specify the key pair resource in your policy — even though it's not required to launch an instance, you cannot connect to your instance without a key pair. For examples of using resource-level permissions with the `ec2:RunInstances` action, see [4: Launching instances \(RunInstances\)](#) (p. 645).

For additional information about resource-level permissions in Amazon EC2, see the following AWS Security Blog post: [Demystifying EC2 Resource-Level Permissions](#).

Example Policies for Working With the AWS CLI or an AWS SDK

The following examples show policy statements that you could use to control the permissions that IAM users have to Amazon EC2. These policies are designed for requests that are made with the AWS CLI or an AWS SDK. For example policies for working in the Amazon EC2 console, see [Example Policies for Working in the Amazon EC2 Console](#) (p. 652). For examples of IAM policies specific to Amazon VPC, see [Controlling Access to Amazon VPC Resources](#)

- [1: Read-only access](#) (p. 642)
- [2: Working with instances](#) (p. 643)
- [3. Working with volumes](#) (p. 644)
- [4: Launching instances \(RunInstances\)](#) (p. 645)
- [5. Working with ClassicLink](#) (p. 649)
- [6. Working with Reserved Instances](#) (p. 651)

Example 1: Read-only access

The following policy grants users permission to use all Amazon EC2 API actions whose names begin with `Describe`. The `Resource` element uses a wildcard to indicate that users can specify all resources with these API actions. The `*` wildcard is also necessary in cases where the API action does not support resource-level permissions. For more information about which ARNs you can use with which Amazon EC2 API actions, see [Supported Resource-Level Permissions for Amazon EC2 API Actions](#) (p. 629).

Users don't have permission to perform any actions on the resources (unless another statement grants them permission to do so) because they're denied permission to use API actions by default.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "ec2:Describe*",
    "Resource": "*"
  }]
}
```

```
}
```

Example 2: Working with instances

a. Describe, launch, stop, start, and terminate all instances

The following policy grants users permission to use the API actions specified in the `Action` element. The `Resource` element uses a `*` wildcard to indicate that users can specify all resources with these API actions. The `*` wildcard is also necessary in cases where the API action does not support resource-level permissions. For more information about which ARNs you can use with which Amazon EC2 API actions, see [Supported Resource-Level Permissions for Amazon EC2 API Actions \(p. 629\)](#).

The users don't have permission to use any other API actions (unless another statement grants them permission to do so) because users are denied permission to use API actions by default.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "ec2:DescribeInstances", "ec2:DescribeImages",
      "ec2:DescribeKeyPairs", "ec2:DescribeSecurityGroups",
      "ec2:DescribeAvailabilityZones",
      "ec2:RunInstances", "ec2:TerminateInstances",
      "ec2:StopInstances", "ec2:StartInstances"
    ],
    "Resource": "*"
  }
]
```

b. Describe all instances, and stop, start, and terminate only particular instances

The following policy allows users to describe all instances, to start and stop only instances `i-1234567890abcdef0` and `i-0598c7d356eba48d7`, and to terminate only instances in the US East (N. Virginia) Region (`us-east-1`) with the resource tag `purpose=test`.

The first statement uses a `*` wildcard for the `Resource` element to indicate that users can specify all resources with the action; in this case, they can list all instances. The `*` wildcard is also necessary in cases where the API action does not support resource-level permissions (in this case, `ec2:DescribeInstances`). For more information about which ARNs you can use with which Amazon EC2 API actions, see [Supported Resource-Level Permissions for Amazon EC2 API Actions \(p. 629\)](#).

The second statement uses resource-level permissions for the `StopInstances` and `StartInstances` actions. The specific instances are indicated by their ARNs in the `Resource` element.

The third statement allows users to terminate all instances in the US East (N. Virginia) Region (`us-east-1`) that belong to the specified AWS account, but only where the instance has the tag `purpose=test`. The `Condition` element qualifies when the policy statement is in effect.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:DescribeInstances",
```

```
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "ec2:StopInstances",
      "ec2:StartInstances"
    ],
    "Resource": [
      "arn:aws:ec2:us-east-1:123456789012:instance/i-1234567890abcdef0",
      "arn:aws:ec2:us-east-1:123456789012:instance/i-0598c7d356eba48d7"
    ]
  },
  {
    "Effect": "Allow",
    "Action": "ec2:TerminateInstances",
    "Resource": "arn:aws:ec2:us-east-1:123456789012:instance/*",
    "Condition": {
      "StringEquals": {
        "ec2:ResourceTag/purpose": "test"
      }
    }
  }
]
}
```

Example 3. Working with volumes

When an API action requires a caller to specify multiple resources, you must create a policy statement that allows users to access all required resources. If you need to use a `Condition` element with one or more of these resources, you must create multiple statements as shown in this example.

The following policy allows users to attach volumes with the tag `volume_user=iam-user-name` to instances with the tag `department=dev`, and to detach those volumes from those instances. If you attach this policy to an IAM group, the `aws:username` policy variable gives each IAM user in the group permission to attach or detach volumes from the instances with a tag named `volume_user` that has his or her IAM user name as a value.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:AttachVolume",
        "ec2:DetachVolume"
      ],
      "Resource": "arn:aws:ec2:us-east-1:123456789012:instance/*",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/department": "dev"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:AttachVolume",

```

```
        "ec2:DetachVolume"
    ],
    "Resource": "arn:aws:ec2:us-east-1:123456789012:volume/*",
    "Condition": {
        "StringEquals": {
            "ec2:ResourceTag/volume_user": "${aws:username}"
        }
    }
}
]
```

Example 4: Launching instances (RunInstances)

The [RunInstances](#) API action launches one or more instances. `RunInstances` requires an AMI and creates an instance; and users can specify a key pair and security group in the request. Launching into EC2-VPC requires a subnet, and creates a network interface. Launching from an Amazon EBS-backed AMI creates a volume. Therefore, the user must have permission to use these Amazon EC2 resources. The caller can also configure the instance using optional parameters to `RunInstances`, such as the instance type and a subnet. You can create a policy statement that requires users to specify an optional parameter, or restricts users to particular values for a parameter. The examples in this section demonstrate some of the many possible ways that you can control the configuration of an instance that a user can launch.

Note that by default, users don't have permission to describe, start, stop, or terminate the resulting instances. One way to grant the users permission to manage the resulting instances is to create a specific tag for each instance, and then create a statement that enables them to manage instances with that tag. For more information, see [2: Working with instances \(p. 643\)](#).

a. AMI

The following policy allows users to launch instances using only the AMIs that have the specified tag, "department=dev", associated with them. The users can't launch instances using other AMIs because the `Condition` element of the first statement requires that users specify an AMI that has this tag. The users also can't launch into a subnet, as the policy does not grant permissions for the subnet and network interface resources. They can, however, launch into EC2-Classical. The second statement uses a wildcard to enable users to create instance resources, and requires users to specify the key pair `project_keypair` and the security group `sg-1a2b3c4d`. Users are still able to launch instances without a key pair.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
        "arn:aws:ec2:region::image/ami-*"
      ],
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/department": "dev"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
```

```
        "arn:aws:ec2:region:account:instance/*",
        "arn:aws:ec2:region:account:volume/*",
        "arn:aws:ec2:region:account:key-pair/project_keypair",
        "arn:aws:ec2:region:account:security-group/sg-1a2b3c4d"
    ]
}
]
```

Alternatively, the following policy allows users to launch instances using only the specified AMIs, `ami-9e1670f7` and `ami-45cf5c3c`. The users can't launch an instance using other AMIs (unless another statement grants the users permission to do so), and the users can't launch an instance into a subnet.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": [
      "arn:aws:ec2:region::image/ami-9e1670f7",
      "arn:aws:ec2:region::image/ami-45cf5c3c",
      "arn:aws:ec2:region:account:instance/*",
      "arn:aws:ec2:region:account:volume/*",
      "arn:aws:ec2:region:account:key-pair/*",
      "arn:aws:ec2:region:account:security-group/*"
    ]
  }
]
```

Alternatively, the following policy allows users to launch instances from all AMIs owned by Amazon. The `Condition` element of the first statement tests whether `ec2:Owner` is `amazon`. The users can't launch an instance using other AMIs (unless another statement grants the users permission to do so). The users are able to launch an instance into a subnet.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": [
      "arn:aws:ec2:region::image/ami-*"
    ],
    "Condition": {
      "StringEquals": {
        "ec2:Owner": "amazon"
      }
    }
  }],
  {
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": [
      "arn:aws:ec2:region:account:instance/*",
      "arn:aws:ec2:region:account:subnet/*",
      "arn:aws:ec2:region:account:volume/*",

```

```
        "arn:aws:ec2:region:account:network-interface/*",  
        "arn:aws:ec2:region:account:key-pair/*",  
        "arn:aws:ec2:region:account:security-group/*"  
    ]  
  }  
]  
}
```

b. Instance type

The following policy allows users to launch instances using only the `t2.micro` or `t2.small` instance type, which you might do to control costs. The users can't launch larger instances because the `Condition` element of the first statement tests whether `ec2:InstanceType` is either `t2.micro` or `t2.small`.

```
{  
  "Version": "2012-10-17",  
  "Statement": [{  
    "Effect": "Allow",  
    "Action": "ec2:RunInstances",  
    "Resource": [  
      "arn:aws:ec2:region:account:instance/*"  
    ],  
    "Condition": {  
      "StringEquals": {  
        "ec2:InstanceType": ["t2.micro", "t2.small"]  
      }  
    }  
  },  
  {  
    "Effect": "Allow",  
    "Action": "ec2:RunInstances",  
    "Resource": [  
      "arn:aws:ec2:region::image/ami-*",  
      "arn:aws:ec2:region:account:subnet/*",  
      "arn:aws:ec2:region:account:network-interface/*",  
      "arn:aws:ec2:region:account:volume/*",  
      "arn:aws:ec2:region:account:key-pair/*",  
      "arn:aws:ec2:region:account:security-group/*"  
    ]  
  }  
]  
}
```

Alternatively, you can create a policy that denies users permission to launch any instances except `t2.micro` and `t2.small` instance types.

```
{  
  "Version": "2012-10-17",  
  "Statement": [{  
    "Effect": "Deny",  
    "Action": "ec2:RunInstances",  
    "Resource": [  
      "arn:aws:ec2:region:account:instance/*"  
    ],  
    "Condition": {  
      "StringNotEquals": {
```

```
        "ec2:InstanceType": ["t2.micro", "t2.small"]
      }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
        "arn:aws:ec2:region::image/ami-*",
        "arn:aws:ec2:region:account:network-interface/*",
        "arn:aws:ec2:region:account:instance/*",
        "arn:aws:ec2:region:account:subnet/*",
        "arn:aws:ec2:region:account:volume/*",
        "arn:aws:ec2:region:account:key-pair/*",
        "arn:aws:ec2:region:account:security-group*"
      ]
    }
  ]
}
```

c. Subnet

The following policy allows users to launch instances using only the specified subnet, subnet-12345678. The group can't launch instances into any another subnet (unless another statement grants the users permission to do so). Users are still able to launch instances into EC2-Classic.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": [
      "arn:aws:ec2:region:account:subnet/subnet-12345678",
      "arn:aws:ec2:region:account:network-interface/*",
      "arn:aws:ec2:region:account:instance/*",
      "arn:aws:ec2:region:account:volume/*",
      "arn:aws:ec2:region::image/ami-*",
      "arn:aws:ec2:region:account:key-pair/*",
      "arn:aws:ec2:region:account:security-group*"
    ]
  }]
}
```

Alternatively, you could create a policy that denies users permission to launch an instance into any other subnet. The statement does this by denying permission to create a network interface, except where subnet subnet-12345678 is specified. This denial overrides any other policies that are created to allow launching instances into other subnets. Users are still able to launch instances into EC2-Classic.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Deny",
    "Action": "ec2:RunInstances",
    "Resource": [
```

```
    "arn:aws:ec2:region:account:network-interface/*"
  ],
  "Condition": {
    "ArnNotEquals": {
      "ec2:Subnet": "arn:aws:ec2:region:account:subnet/subnet-12345678"
    }
  }
},
{
  "Effect": "Allow",
  "Action": "ec2:RunInstances",
  "Resource": [
    "arn:aws:ec2:region::image/ami-*",
    "arn:aws:ec2:region:account:network-interface/*",
    "arn:aws:ec2:region:account:instance/*",
    "arn:aws:ec2:region:account:subnet/*",
    "arn:aws:ec2:region:account:volume/*",
    "arn:aws:ec2:region:account:key-pair/*",
    "arn:aws:ec2:region:account:security-group/*"
  ]
}
]
```

Example 5. Working with ClassicLink

You can enable a VPC for ClassicLink and then link an EC2-Classic instance to the VPC. You can also view your ClassicLink-enabled VPCs, and all of your EC2-Classic instances that are linked to a VPC. You can create policies with resource-level permission for the `ec2:EnableVpcClassicLink`, `ec2:DisableVpcClassicLink`, `ec2:AttachClassicLinkVpc`, and `ec2:DetachClassicLinkVpc` actions to control how users are able to use those actions. Resource-level permissions are not supported for `ec2:Describe*` actions.

a. Full permission to work with ClassicLink

The following policy grants users permission to view ClassicLink-enabled VPCs and linked EC2-Classic instances, to enable and disable a VPC for ClassicLink, and to link and unlink instances from a ClassicLink-enabled VPC.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "ec2:DescribeClassicLinkInstances", "ec2:DescribeVpcClassicLink",
      "ec2:EnableVpcClassicLink", "ec2:DisableVpcClassicLink",
      "ec2:AttachClassicLinkVpc", "ec2:DetachClassicLinkVpc"
    ],
    "Resource": "*"
  }
]
```

b. Enable and disable a VPC for ClassicLink

The following policy allows user to enable and disable VPCs for ClassicLink that have the specific tag `'purpose=classiclink'`. Users cannot enable or disable any other VPCs for ClassicLink.


```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:*VpcClassicLink",
      "Resource": "arn:aws:ec2:region:account:vpc/*",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/purpose": "classiclink"
        }
      }
    }
  ]
}
```

c. Link instances

The following policy grants users permission to link instances to a VPC only if the instance is an `m3.large` instance type. The second statement allows users to use the VPC and security group resources, which are required to link an instance to a VPC.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:AttachClassicLinkVpc",
      "Resource": "arn:aws:ec2:region:account:instance/*",
      "Condition": {
        "StringEquals": {
          "ec2:InstanceType": "m3.large"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:AttachClassicLinkVpc",
      "Resource": [
        "arn:aws:ec2:region:account:vpc/*",
        "arn:aws:ec2:region:account:security-group/*"
      ]
    }
  ]
}
```

The following policy grants users permission to link instances to a specific VPC (`vpc-1a2b3c4d`) only, and to associate only specific security groups from the VPC to the instance (`sg-1122aabb` and `sg-aabb2233`). Users cannot link an instance to any other VPC, and they cannot specify any other of the VPC security groups to associate with the instance in the request.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:AttachClassicLinkVpc",
```

```
"Resource": [
  "arn:aws:ec2:region:account:vpc/vpc-1a2b3c4d",
  "arn:aws:ec2:region:account:instance/*",
  "arn:aws:ec2:region:account:security-group/sg-1122aabb",
  "arn:aws:ec2:region:account:security-group/sg-aabb2233"
]
}
```

d. Unlink instances

The following grants users permission to unlink any linked EC2-Classic instance from a VPC, but only if the instance has the tag "unlink=true". The second statement grants users permission to use the VPC resource, which is required to unlink an instance from a VPC.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "ec2:DetachClassicLinkVpc",
    "Resource": [
      "arn:aws:ec2:region:account:instance/*"
    ],
    "Condition": {
      "StringEquals": {
        "ec2:ResourceTag/unlink": "true"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": "ec2:DetachClassicLinkVpc",
    "Resource": [
      "arn:aws:ec2:region:account:vpc/*"
    ]
  }
]
```

Example 6. Working with Reserved Instances

The following policy gives users permission to view, modify, and purchase Reserved Instances in your account.

It is not possible to set resource-level permissions for individual Reserved Instances. This policy means that users have access to all the Reserved Instances in the account.

The `Resource` element uses a `*` wildcard to indicate that users can specify all resources with the action; in this case, they can list and modify all Reserved Instances in the account. They can also purchase Reserved Instances using the account credentials. The `*` wildcard is also necessary in cases where the API action does not support resource-level permissions.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
```

```
"Action": [
  "ec2:DescribeReservedInstances", "ec2:ModifyReservedInstances",
  "ec2:PurchaseReservedInstancesOffering",
"ec2:DescribeAvailabilityZones",
  "ec2:DescribeReservedInstancesOfferings"
],
"Resource": "*"
}
]
```

To allow users to view and modify the Reserved Instances in your account, but not purchase new Reserved Instances.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "ec2:DescribeReservedInstances", "ec2:ModifyReservedInstances",
      "ec2:DescribeAvailabilityZones"
    ],
    "Resource": "*"
  }
]
```

Example Policies for Working in the Amazon EC2 Console

You can use IAM policies to grant users permissions to view and work with specific resources in the Amazon EC2 console. You can use the example policies in the previous section; however, they are designed for requests that are made with the AWS CLI or an AWS SDK. The console uses additional API actions for its features, so these policies may not work as expected. For example, a user that has permission to use only the `DescribeVolumes` API action will encounter errors when trying to view volumes in the console. This section demonstrates policies that enable users to work with specific parts of the console.

- [1: Read-only access \(p. 653\)](#)
- [2: Using the EC2 launch wizard \(p. 654\)](#)
- [3: Working with volumes \(p. 655\)](#)
- [4: Working with security groups \(p. 656\)](#)
- [5: Working with Elastic IP addresses \(p. 657\)](#)
- [6: Working with Reserved Instances \(p. 658\)](#)

Note

To help you work out which API actions are required to perform tasks in the console, you can use a service such as AWS CloudTrail. For more information, see the [AWS CloudTrail User Guide](#). If your policy does not grant permission to create or modify a specific resource, the console displays an encoded message with diagnostic information. You can decode the message using the `DecodeAuthorizationMessage` API action for AWS STS, or the `decode-authorization-message` command in the AWS CLI.

For additional information about creating policies for the Amazon EC2 console, see the following AWS Security Blog post: [Granting Users Permission to Work in the Amazon EC2 Console](#).

Example 1: Read-only access

To allow users to view all resources in the Amazon EC2 console, you can use the same policy as the following example: [1: Read-only access \(p. 642\)](#). Users cannot perform any actions on those resources or create new resources, unless another statement grants them permission to do so.

a. View instances, AMIs, and snapshots

Alternatively, you can provide read-only access to a subset of resources. To do this, replace the * wildcard in the `ec2:Describe` API action with specific `ec2:Describe` actions for each resource. The following policy allows users to view all instances, AMIs, and snapshots in the Amazon EC2 console. The `ec2:DescribeTags` action allows users to view public AMIs. The console requires the tagging information to display public AMIs; however, you can remove this action if you want users to view only private AMIs.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "ec2:DescribeInstances", "ec2:DescribeImages",
      "ec2:DescribeTags", "ec2:DescribeSnapshots"
    ],
    "Resource": "*"
  }
]
```

Note

Currently, the Amazon EC2 `ec2:Describe*` API actions do not support resource-level permissions, so you cannot control which individual resources users can view in the console. Therefore, the * wildcard is necessary in the `Resource` element of the above statement. For more information about which ARNs you can use with which Amazon EC2 API actions, see [Supported Resource-Level Permissions for Amazon EC2 API Actions \(p. 629\)](#).

b. View instances and CloudWatch metrics

The following policy allows users to view instances in the Amazon EC2 console, as well as CloudWatch alarms and metrics in the **Monitoring** tab of the **Instances** page. The Amazon EC2 console uses the CloudWatch API to display the alarms and metrics, so you must grant users permission to use the `cloudwatch:DescribeAlarms` and `cloudwatch:GetMetricStatistics` actions.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "ec2:DescribeInstances",
      "cloudwatch:DescribeAlarms",
      "cloudwatch:GetMetricStatistics"
    ],
    "Resource": "*"
  }
]
```

instance. Your policy must include permission to use the API actions that allow users to work with the wizard's options. If your policy does not include permission to use those actions, some items in the wizard cannot load properly, and users cannot complete a launch.

```

Statement: [
  {
    Action: [
      "iam:ListPolicies",
      "iam:ListPolicyVersions",
      "iam:PutPolicy"
    ],
    Effect: "Allow",
    Resource: "*"
  },
  {
    Action: [
      "ec2:DescribeInstances",
      "ec2:DescribeImages",
      "ec2:DescribeKeyPairs",
      "ec2:DescribeVpcs",
      "ec2:DescribeSubnets",
      "ec2:DescribeSecurityGroups"
    ],
    Effect: "Allow",
    Resource: "*"
  }
]

```

Example 2: Using the EC2 Launch wizard

The first statement grants users permission to provide the options in the launch wizard as demonstrated in the example above. The second statement grants users permission to use the network interface, volume, elastic block store, and subnet resources for EC2 Classic to view and select a specific resource to attach to an instance. For more information about using the `ec2:RunInstances` action, see [Using Amazon EC2 Classic to launch instances in a VPC](#). The third and fourth statements grant users permission to view and select existing network interfaces for the selected subnet. The fifth statement grants users permission to view and select existing AMI resources and only grants a specific permission for the `ami:RunInstances` action. Users can only launch in the sa-east-1 region. If users select a different region, or select a different instance type, AMI, or subnet in the launch wizard, the wizard's suggested `launch-wizard-x` security group. However, this action alone only creates

the security group; it does not add or modify any rules. To add inbound rules, users must be granted permission to use the `ec2:AuthorizeSecurityGroupIngress` API action. To add inbound rules to VPC security groups, users must be granted permission to use the `ec2:AuthorizeSecurityGroupIngress` API action. To modify or delete existing rules, users must be granted permission to use the `ec2:RevokeSecurityGroupIngress` API action.

b. Restricting access to specific instance type, subnet, and region

- `ec2:CreateTags`: To add a tag to the instance. By default, the launch wizard attempts to add a tag with a key of `aws:ec2:instance-type`. Users that do not have permission to use this action will encounter a warning that this tag could not be applied to an instance; however, this does not affect the success of the launch, so you should only grant users permission to use this action if it is absolutely necessary.

```

{
  "Effect": "Allow",
  "Action": "ec2:RunInstances",
  "Resource": [
    "arn:aws:ec2:sa-east-1:111122223333:instance/*"
  ],
  "Condition": {
    "StringEquals": {
      "ec2:InstanceType": "m1.small"
    }
  }
},
{
  "Effect": "Allow",
  "Action": "ec2:RunInstances",
  "Resource": [
    "arn:aws:ec2:sa-east-1::image/ami-*"
  ],
  "Condition": {
    "StringEquals": {
      "ec2:Owner": "amazon"
    }
  }
}
]
}

```

Example 3: Working with volumes

The following policy grants users permission to view and create volumes, and attach and detach volumes to specific instances.

Users can attach any volume to instances that have the tag "purpose=test", and also detach volumes from those instances. To attach a volume using the Amazon EC2 console, it is helpful for users to have permission to use the `ec2:DescribeInstances` action, as this allows them to select an instance from a pre-populated list in the **Attach Volume** dialog box. However, this also allows users to view all instances on the **Instances** page in the console, so you can omit this action.

In the first statement, the `ec2:DescribeVolumeStatus` and `ec2:DescribeAvailabilityZones` actions are necessary to ensure that volumes display correctly in the console.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "ec2:DescribeVolumes", "ec2:DescribeVolumeStatus",
      "ec2:DescribeAvailabilityZones", "ec2:CreateVolume",
      "ec2:DescribeInstances"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "ec2:AttachVolume",
      "ec2:DetachVolume"
    ],
    "Resource": "arn:aws:ec2:region:111122223333:instance/*",
    "Condition": {
      "StringEquals": {
        "ec2:ResourceTag/purpose": "test"
      }
    }
  }
],
  {
    "Effect": "Allow",
    "Action": [
      "ec2:AttachVolume",
      "ec2:DetachVolume"
    ],
    "Resource": "arn:aws:ec2:region:111122223333:volume/*"
  }
]
```

a. View security groups and add and remove rules

```

    "Action": [
      "ec2:AuthorizeSecurityGroupIngress",
      "ec2:RevokeSecurityGroupIngress",
      "ec2:AuthorizeSecurityGroupEgress", "ec2:RevokeSecurityGroupEgress"
    ],
    "Resource": [
      "arn:aws:ec2:region:111122223333:security-group/*"
    ]
  }
}

```

Example 4: Working with security groups

You can create a policy that allows users to work with the **Create Security Group** dialog box in the Amazon EC2 console. To use this dialog box, users must be granted permission to use at the least the following API actions:

- `ec2:CreateSecurityGroup`: To create a new security group.

b. Working with the **Create Security Group** dialog box

With these permissions, users can create a new security group successfully, but they cannot add any rules to it. To work with rules in the **Create Security Group** dialog box, you can add the following API actions to your policy:

- `ec2:AuthorizeSecurityGroupIngress`: To add inbound rules.
- `ec2:AuthorizeSecurityGroupEgress`: To add outbound rules to VPC security groups.
- `ec2:RevokeSecurityGroupIngress`: To modify or delete existing inbound rules. This is useful if you want to allow users to use the **Copy to new** feature in the console. This feature opens the **Create Security Group** dialog box and populates it with the same rules as the security group that was selected.
- `ec2:RevokeSecurityGroupEgress`: To modify or delete outbound rules for VPC security groups. This is useful to allow users to modify or delete the default outbound rule that allows all outbound traffic.
- `ec2>DeleteSecurityGroup`: To cater for scenarios where invalid rules cannot be saved. If a user creates a security group with an invalid rule, the console first creates the security group, then attempts to add the rules to it. After that fails, the security group is deleted. The user remains in the **Create Security Group** dialog box, where an error is displayed. The rules remain listed, so the user can correct the invalid rule and try to create the security group again. This API action is not required, but if a user is not granted permission to use it and attempts to create a security group with invalid rules, the security group is created without any rules, and the user is returned to the console.

The following policy grants users permission to use the **Create Security Group** dialog box and to create inbound and outbound rules for security groups that are associated with a specific VPC (`vpc-1a2b3c4d`). Users can create security groups for EC2-Classical or another VPC, but they cannot currently add rules to them. Similarly, users cannot add rules to any other resource-level permissions based on VPC IDs. This policy also grants users permission to delete security groups and the console. This makes it easier for users to actions to the security groups that they can add inbound rules. This policy also grants users permission to delete security groups that are associated with VPC `vpc-1a2b3c4d`.

```

    "Action": [
      "ec2:DescribeSecurityGroups", "ec2:CreateSecurityGroup",
      "ec2:DescribeVpcs"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "ec2>DeleteSecurityGroup", "ec2:AuthorizeSecurityGroupIngress",
      "ec2:AuthorizeSecurityGroupEgress"
    ],
    "Resource": "arn:aws:ec2:region:111122223333:security-group/*",
    "Condition": {
      "ArnEquals": {
        "ec2:Vpc": "arn:aws:ec2:region:111122223333:vpc/vpc-1a2b3c4d"
      }
    }
  }
}

```

Example 5: Working with Elastic IP addresses

The following policy grants users permission to view Elastic IP addresses in the Amazon EC2 console. The console uses the `ec2:DescribeInstances` action to display information about instances with which the Elastic IP addresses are associated. If users are not granted permission to use this action, the Elastic IP addresses page cannot load properly.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "ec2:DescribeAddresses", "ec2:DescribeInstances"
    ],
    "Resource": "*"
  }]
}
```

To allow users to work with Elastic IP addresses, you can add the following actions to your policy

- `ec2:AllocateAddress`: To allocate an address for use in VPC or EC2-Classic.
- `ec2:ReleaseAddress`: To release an Elastic IP address.
- `ec2:DescribeNetworkInterfaces`: To work with the **Associate Address** dialog box. The dialog box displays the available network interfaces to which you can associate an Elastic IP address, and will not open if users are not granted permission to use this action. However, this only applies to EC2-VPC; this action is not required for associating an Elastic IP address to an instance in EC2-Classic.
- `ec2:AssociateAddress`: To associate an Elastic IP address with an instance or a network interface.
- `ec2:DisassociateAddress`: To disassociate an Elastic IP address from an instance or a network interface,

Example 6: Working with Reserved Instances

The following policy can be attached to an IAM user. It gives the user access to view and modify Reserved Instances in your account, as well as purchase new Reserved Instances in the AWS Management Console.

This policy allows users to view all the Reserved Instances, as well as On-Demand Instances, in the account. It's not possible to set resource-level permissions for individual Reserved Instances.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "ec2:DescribeReservedInstances", "ec2:ModifyReservedInstances",
      "ec2:PurchaseReservedInstancesOffering", "ec2:DescribeInstances",
      "ec2:DescribeAvailabilityZones",
      "ec2:DescribeReservedInstancesOfferings"
    ],
    "Resource": "*"
  }
]
```

The `ec2:DescribeAvailabilityZones` action is necessary to ensure that the Amazon EC2 console can display information about the Availability Zones in which you can purchase Reserved Instances. The `ec2:DescribeInstances` action is not required, but ensures that the user can view the instances in the account and purchase reservations to match the correct specifications.

You can adjust the API actions to limit user access, for example removing `ec2:DescribeInstances` and `ec2:DescribeAvailabilityZones` means the user has read-only access.

IAM Roles for Amazon EC2

Applications must sign their API requests with AWS credentials. Therefore, if you are an application developer, you need a strategy for managing credentials for your applications that run on EC2 instances. For example, you can securely distribute your AWS credentials to the instances, enabling the applications on those instances to use your credentials to sign requests, while protecting them from other users. However, it's challenging to securely distribute credentials to each instance, especially those that AWS creates on your behalf, such as Spot instances or instances in Auto Scaling groups. You must also be able to update the credentials on each instance when you rotate your AWS credentials.

We designed IAM roles so that your applications can securely make API requests from your instances, without requiring you to manage the security credentials that the applications use. Instead of creating and distributing your AWS credentials, you can delegate permission to make API requests using IAM roles as follows:

1. Create an IAM role.
2. Define which accounts or AWS services can assume the role.
3. Define which API actions and resources the application can use after assuming the role.
4. Specify the role when you launch your instances.
5. Have the application retrieve a set of temporary credentials and use them.

For example, you can use IAM roles to grant permissions to applications running on your instances that needs to use a bucket in Amazon S3.

Note

Amazon EC2 uses an *instance profile* as a container for an IAM role. When you create an IAM role using the console, the console creates an instance profile automatically and gives it the same name as the role it corresponds to. If you use the AWS CLI, API, or an AWS SDK to create a role, you create the role and instance profile as separate actions, and you might give them different names. To launch an instance with an IAM role, you specify the name of its instance profile. When you launch an instance using the Amazon EC2 console, you can select a role to associate with the instance; however, the list that's displayed is actually a list of instance profile names. For more information, see [Instance Profiles](#) in the *IAM User Guide*.

You can specify permissions for IAM roles by creating a policy in JSON format. These are similar to the policies that you create for IAM users. If you make a change to a role, the change is propagated to all instances, simplifying credential management.

Note

You can't assign a role to an existing instance; you can only specify a role when you launch a new instance.

For more information about creating and using IAM roles, see [Roles](#) in the *IAM User Guide*.

Topics

- [Retrieving Security Credentials from Instance Metadata](#) (p. 659)
- [Granting an IAM User Permission to Launch an Instance with an IAM Role](#) (p. 660)
- [Creating an IAM Role Using the Console](#) (p. 660)
- [Launching an Instance with an IAM Role Using the Console](#) (p. 661)
- [Creating an IAM Role Using the AWS CLI](#) (p. 661)
- [Launching an Instance with an IAM Role Using the AWS CLI](#) (p. 663)

Retrieving Security Credentials from Instance Metadata

An application on the instance retrieves the security credentials provided by the role from the instance metadata item `iam/security-credentials/role-name`. The application is granted the permissions for the actions and resources that you've defined for the role through the security credentials associated with the role. These security credentials are temporary and we rotate them automatically. We make new credentials available at least five minutes prior to the expiration of the old credentials.

Warning

If you use services that use instance metadata with IAM roles, ensure that you don't expose your credentials when the services make HTTP calls on your behalf. The types of services that could expose your credentials include HTTP proxies, HTML/CSS validator services, and XML processors that support XML inclusion.

The following command retrieves the security credentials for an IAM role named `s3access`.

```
C:\> curl http://169.254.169.254/latest/meta-data/iam/security-credentials/s3access
```

The following is example output.

```
{
  "Code" : "Success",
  "LastUpdated" : "2012-04-26T16:39:16Z",
  "Type" : "AWS-HMAC",
  "AccessKeyId" : "AKIAIOSFODNN7EXAMPLE",
  "SecretAccessKey" : "wJalrXUtnFEMI/K7MDENG/bPxrFicYEXAMPLEKEY",
  "Token" : "token",
  "Expiration" : "2012-04-27T22:39:16Z"
```

```
}
```

For applications, AWS CLI, and Tools for Windows PowerShell commands that run on the instance, you do not have to explicitly get the temporary security credentials — the AWS SDKs, AWS CLI, and Tools for Windows PowerShell automatically get the credentials from the EC2 instance metadata service and use them. To make a call outside of the instance using temporary security credentials (for example, to test IAM policies), you must provide the access key, secret key, and the session token. For more information, see [Using Temporary Security Credentials to Request Access to AWS Resources](#) in the *IAM User Guide*.

For more information about instance metadata, see [Instance Metadata and User Data](#) (p. 271).

Granting an IAM User Permission to Launch an Instance with an IAM Role

To enable an IAM user to launch an instance with an IAM role, you must grant the user permission to pass the role to the instance.

For example, the following IAM policy grants users permission to launch an instance with the IAM role named `s3access`.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "iam:PassRole",
    "Resource": "arn:aws:iam::123456789012:role/s3access"
  }]
}
```

Alternatively, you could grant IAM users access to all your roles by specifying the resource as `">*` in this policy. However, consider whether users who launch instances with your roles (ones that exist or that you'll create later on) might be granted permissions that they don't need or shouldn't have.

For more information, see [Permissions Required for Using Roles with Amazon EC2](#) in the *IAM User Guide*.

Creating an IAM Role Using the Console

You must create an IAM role before you can launch an instance with that role.

To create an IAM role using the IAM console

1. Sign in to the Identity and Access Management (IAM) console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Roles**, and then choose **Create New Role**.
3. On the **Set Role Name** page, enter a name for the role and choose **Next Step**.
4. On the **Select Role Type** page, choose **Select** next to **Amazon EC2**.
5. On the **Attach Policy** page, select an AWS managed policy. For example, for Amazon EC2, one of the following AWS managed policies might meet your needs:
 - PowerUserAccess
 - ReadOnlyAccess
 - AmazonEC2FullAccess
 - AmazonEC2ReadOnlyAccess
6. Review the role information, edit the role as needed, and then choose **Create Role**.

Launching an Instance with an IAM Role Using the Console

After you've created an IAM role, you can launch an instance, and associate that role with the instance during launch.

Important

After you create an IAM role, it may take several seconds for the permissions to propagate. If your first attempt to launch an instance with a role fails, wait a few seconds before trying again. For more information, see [Troubleshooting Working with Roles](#) in the *IAM User Guide*.

To launch an instance with an IAM role

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the dashboard, choose **Launch Instance**.
3. Select an AMI, then select an instance type and choose **Next: Configure Instance Details**.
4. On the **Configure Instance Details** page, select the IAM role you created from the **IAM role** list.

Note

The **IAM role** list displays the name of the instance profile that you created when you created your IAM role. If you created your IAM role using the console, the instance profile was created for you and given the same name as the role. If you created your IAM role using the AWS CLI, API, or an AWS SDK, you may have named your instance profile differently.

5. Configure any other details, then follow the instructions through the rest of the wizard, or choose **Review and Launch** to accept default settings and go directly to the **Review Instance Launch** page.
6. Review your settings, then choose **Launch** to choose a key pair and launch your instance.
7. If you are using the Amazon EC2 API actions in your application, retrieve the AWS security credentials made available on the instance and use them to sign the requests. Note that the AWS SDK does this for you.

```
C:\> curl http://169.254.169.254/latest/meta-data/iam/security-credentials/role_name
```

Creating an IAM Role Using the AWS CLI

You must create an IAM role before you can launch an instance with that role.

To create an IAM role using the AWS CLI

- Create an IAM role with a policy that allows the role to use an Amazon S3 bucket.
 - a. Create the following trust policy and save it in a text file named `ec2-role-trust-policy.json`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": { "Service": "ec2.amazonaws.com" },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

- b. Create the `s3access` role and specify the trust policy that you created.

```
C:\> aws iam create-role --role-name s3access --assume-role-policy-  
document file://ec2-role-trust-policy.json  
{  
  "Role": {  
    "AssumeRolePolicyDocument": {  
      "Version": "2012-10-17",  
      "Statement": [  
        {  
          "Action": "sts:AssumeRole",  
          "Effect": "Allow",  
          "Principal": {  
            "Service": "ec2.amazonaws.com"  
          }  
        }  
      ]  
    },  
    "RoleId": "AROAIIZKPBKS2LEXAMPLE",  
    "CreateDate": "2013-12-12T23:46:37.247Z",  
    "RoleName": "s3access",  
    "Path": "/",  
    "Arn": "arn:aws:iam::123456789012:role/s3access"  
  }  
}
```

- c. Create an access policy and save it in a text file named `ec2-role-access-policy.json`. For example, this policy grants administrative permissions for Amazon S3 to applications running on the instance.

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": ["s3:*"],  
      "Resource": ["*"]  
    }  
  ]  
}
```

- d. Attach the access policy to the role.

```
C:\> aws iam put-role-policy --role-name s3access --policy-name S3-  
Permissions --policy-document file://ec2-role-access-policy.json
```

- e. Create an instance profile named `s3access-profile`.

```
C:\> aws iam create-instance-profile --instance-profile-name s3access-  
profile  
{  
  "InstanceProfile": {  
    "InstanceProfileId": "AIPAJTLPJLEGREXAMPLE",  
    "Roles": [],  
    "CreateDate": "2013-12-12T23:53:34.093Z",  
    "InstanceProfileName": "s3access-profile",  
    "Path": "/",  
  }  
}
```

```
"Arn": "arn:aws:iam::123456789012:instance-profile/s3access-  
profile"  
  }  
}
```

- f. Add the `s3access` role to the `s3access-profile` instance profile.

```
C:\> aws iam add-role-to-instance-profile --instance-profile-name  
s3access-profile --role-name s3access
```

For more information about these commands, see [create-role](#), [put-role-policy](#), and [create-instance-profile](#) in the *AWS Command Line Interface Reference*.

Launching an Instance with an IAM Role Using the AWS CLI

After you've created an IAM role, you can launch an instance, and associate that role with the instance during launch.

Important

After you create an IAM role, it may take several seconds for the permissions to propagate. If your first attempt to launch an instance with a role fails, wait a few seconds before trying again. For more information, see [Troubleshooting Working with Roles](#) in the *IAM User Guide*.

To launch an instance with an IAM role using the AWS CLI

1. Launch an instance using the instance profile. The following example shows how to launch an instance with the instance profile.

```
C:\> aws ec2 run-instances --image-id ami-11aa22bb --iam-instance-profile  
Name="s3access-profile" --key-name my-key-pair --security-groups my-  
security-group --subnet-id subnet-1a2b3c4d
```

For more information, see [run-instances](#) in the *AWS Command Line Interface Reference*.

2. If you are using the Amazon EC2 API actions in your application, retrieve the AWS security credentials made available on the instance and use them to sign the requests. Note that the AWS SDK does this for you.

```
C:\> curl http://169.254.169.254/latest/meta-data/iam/security-  
credentials/role_name
```

Authorizing Inbound Traffic for Your Windows Instances

Security groups enable you to control traffic to your instance, including the kind of traffic that can reach your instance. For example, you can allow computers from only your home network to access your instance using RDP. If your instance is a web server, you can allow all IP addresses to access your instance via HTTP, so that external users can browse the content on your web server.

To enable network access to your instance, you must allow inbound traffic to your instance. To open a port for inbound traffic, add a rule to a security group that you associated with your instance when you launched it.

To connect to your instance, you must set up a rule to authorize RDP traffic from your computer's public IPv4 address. To allow RDP traffic from additional IP address ranges, add another rule for each range you need to authorize.

If you've enabled your VPC for IPv6 and launched your instance with an IPv6 address, you can connect to your instance using its IPv6 address instead of a public IPv4 address. Your local computer must have an IPv6 address and must be configured to use IPv6.

If you need to enable network access to a Linux instance, see [Authorizing Inbound Traffic for Your Linux Instances](#) in the *Amazon EC2 User Guide for Linux Instances*.

Before You Start

Decide who requires access to your instance; for example, a single host or a specific network that you trust; for example, your local computer's public IPv4 address. The security group editor in the Amazon EC2 console can automatically detect the public IPv4 address of your local computer for you. Alternatively, you can use the search phrase "what is my IP address" in an Internet browser, or use the following service: <http://checkip.amazonaws.com/>. If you are connecting through an ISP or from behind your firewall without a static IP address, you need to find out the range of IP addresses used by client computers.

Caution

If you use `0.0.0.0/0`, you enable all IPv4 addresses to access your instance using RDP. If you use `::/0`, you enable all IPv6 address to access your instance. This is acceptable for a short time in a test environment, but it's unsafe for production environments. In production, you'll authorize only a specific IP address or range of addresses to access your instance.

For more information about security groups, see [Amazon EC2 Security Groups for Windows Instances](#) (p. 606).

Adding a Rule for Inbound RDP Traffic to a Windows Instance

Security groups act as a firewall for associated instances, controlling both inbound and outbound traffic at the instance level. You must add rules to a security group that enable you to connect to your Windows instance from your IP address using RDP.

To add a rule to a security group for inbound RDP traffic over IPv4 using the console

1. In the navigation pane of the Amazon EC2 console, choose **Instances**. Select your instance and look at the **Description** tab; **Security groups** lists the security groups that are associated with the instance. Choose **view rules** to display a list of the rules that are in effect for the instance.
2. In the navigation pane, choose **Security Groups**. Select one of the security groups associated with your instance.
3. In the details pane, on the **Inbound** tab, choose **Edit**. In the dialog, choose **Add Rule**, and then choose **RDP** from the **Type** list.
4. In the **Source** field, choose **My IP** to automatically populate the field with the public IPv4 address of your local computer. Alternatively, choose **Custom** and specify the public IPv4 address of your computer or network in CIDR notation. For example, if your IPv4 address is `203.0.113.25`, specify `203.0.113.25/32` to list this single IPv4 address in CIDR notation. If your company allocates addresses from a range, specify the entire range, such as `203.0.113.0/24`.

For information about finding your IP address, see [Before You Start](#) (p. 664).

5. Choose **Save**.

(VPC only) If you launched an instance with an IPv6 address and want to connect to your instance using its IPv6 address, you must add rules that allow inbound IPv6 traffic over RDP.

To add a rule to a security group for inbound RDP traffic over IPv6 using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups**. Select the security group for your instance.
3. Choose **Inbound, Edit, Add Rule**.
4. For **Type**, choose **RDP**.
5. In the **Source** field, specify the IPv6 address of your computer in CIDR notation. For example, if your IPv6 address is 2001:db8:1234:1a00:9691:9503:25ad:1761, specify 2001:db8:1234:1a00:9691:9503:25ad:1761/128 to list the single IP address in CIDR notation. If your company allocates addresses from a range, specify the entire range, such as 2001:db8:1234:1a00::/64.
6. Choose **Save**.

To add a rule to a security group using the command line

You can use one of the following commands. Be sure to run this command on your local system, not on the instance itself. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [authorize-security-group-ingress](#) (AWS CLI)
- [Grant-EC2SecurityGroupIngress](#) (AWS Tools for Windows PowerShell)

Assigning a Security Group to an Instance

You can assign a security group to an instance when you launch the instance. When you add or remove rules, those changes are automatically applied to all instances to which you've assigned the security group.

After you launch an instance in EC2-Classic, you can't change its security groups. After you launch an instance in a VPC, you can change its security groups. For more information, see [Changing an Instance's Security Groups](#) in the *Amazon VPC User Guide*.

Amazon EC2 and Amazon Virtual Private Cloud

Amazon Virtual Private Cloud (Amazon VPC) enables you to define a virtual network in your own logically isolated area within the AWS cloud, known as a *virtual private cloud (VPC)*. You can launch your AWS resources, such as instances, into your VPC. Your VPC closely resembles a traditional network that you might operate in your own data center, with the benefits of using AWS's scalable infrastructure. You can configure your VPC; you can select its IP address range, create subnets, and configure route tables, network gateways, and security settings. You can connect instances in your VPC to the Internet. You can connect your VPC to your own corporate data center, making the AWS cloud an extension of your data center. To protect the resources in each subnet, you can use multiple layers of security, including security groups and network access control lists. For more information, see the [Amazon VPC User Guide](#).

Your account may support both the EC2-VPC and EC2-Classic platforms, on a region-by-region basis. If you created your account after 2013-12-04, it supports EC2-VPC only. To find out which platforms your account supports, see [Supported Platforms \(p. 672\)](#). If your account supports EC2-VPC only, we create a *default VPC* for you. A default VPC is a VPC that is already configured and ready for you to use. You can launch instances into your default VPC immediately. For more information, see [Your Default VPC and Subnets](#) in the *Amazon VPC User Guide*. If your account supports EC2-Classic and EC2-VPC, you can launch instances into either platform. Regardless of which platforms your account supports, you can create your own *nondefault VPC*, and configure it as you need.

Contents

- [Benefits of Using a VPC \(p. 666\)](#)
- [Differences Between EC2-Classic and EC2-VPC \(p. 666\)](#)
- [Sharing and Accessing Resources Between EC2-Classic and EC2-VPC \(p. 669\)](#)
- [Instance Types Available Only in a VPC \(p. 671\)](#)
- [Amazon VPC Documentation \(p. 671\)](#)
- [Supported Platforms \(p. 672\)](#)
- [ClassicLink \(p. 673\)](#)
- [Migrating from a Windows Instance in EC2-Classic to a Windows Instance in a VPC \(p. 683\)](#)

Benefits of Using a VPC

By launching your instances into a VPC instead of EC2-Classic, you gain the ability to:

- Assign static private IPv4 addresses to your instances that persist across starts and stops
- Assign multiple IPv4 addresses to your instances
- Define network interfaces, and attach one or more network interfaces to your instances
- Change security group membership for your instances while they're running
- Control the outbound traffic from your instances (egress filtering) in addition to controlling the inbound traffic to them (ingress filtering)
- Add an additional layer of access control to your instances in the form of network access control lists (ACL)
- Run your instances on single-tenant hardware
- Assign IPv6 addresses to your instances

Differences Between EC2-Classic and EC2-VPC

The following table summarizes the differences between instances launched in EC2-Classic, instances launched in a default VPC, and instances launched in a nondefault VPC.

Characteristic	EC2-Classic	Default VPC	Nondefault VPC
Public IPv4 address (from Amazon's public IP address pool)	Your instance receives a public IPv4 address.	Your instance launched in a default subnet receives a public IPv4 address by default, unless you specify otherwise during launch, or you modify the subnet's public IPv4 address attribute.	Your instance doesn't receive a public IPv4 address by default, unless you specify otherwise during launch, or you modify the subnet's public IPv4 address attribute.
Private IPv4 address	Your instance receives a private IPv4 address from the EC2-Classic range each time it's started.	Your instance receives a static private IPv4 address from the address range of your default VPC.	Your instance receives a static private IPv4 address from the address range of your VPC.
Multiple private IPv4 addresses	We select a single private IP address for your instance; multiple IP addresses are not supported.	You can assign multiple private IPv4 addresses to your instance.	You can assign multiple private IPv4 addresses to your instance.

Amazon Elastic Compute Cloud
User Guide for Windows Instances
Differences Between EC2-Classic and EC2-VPC

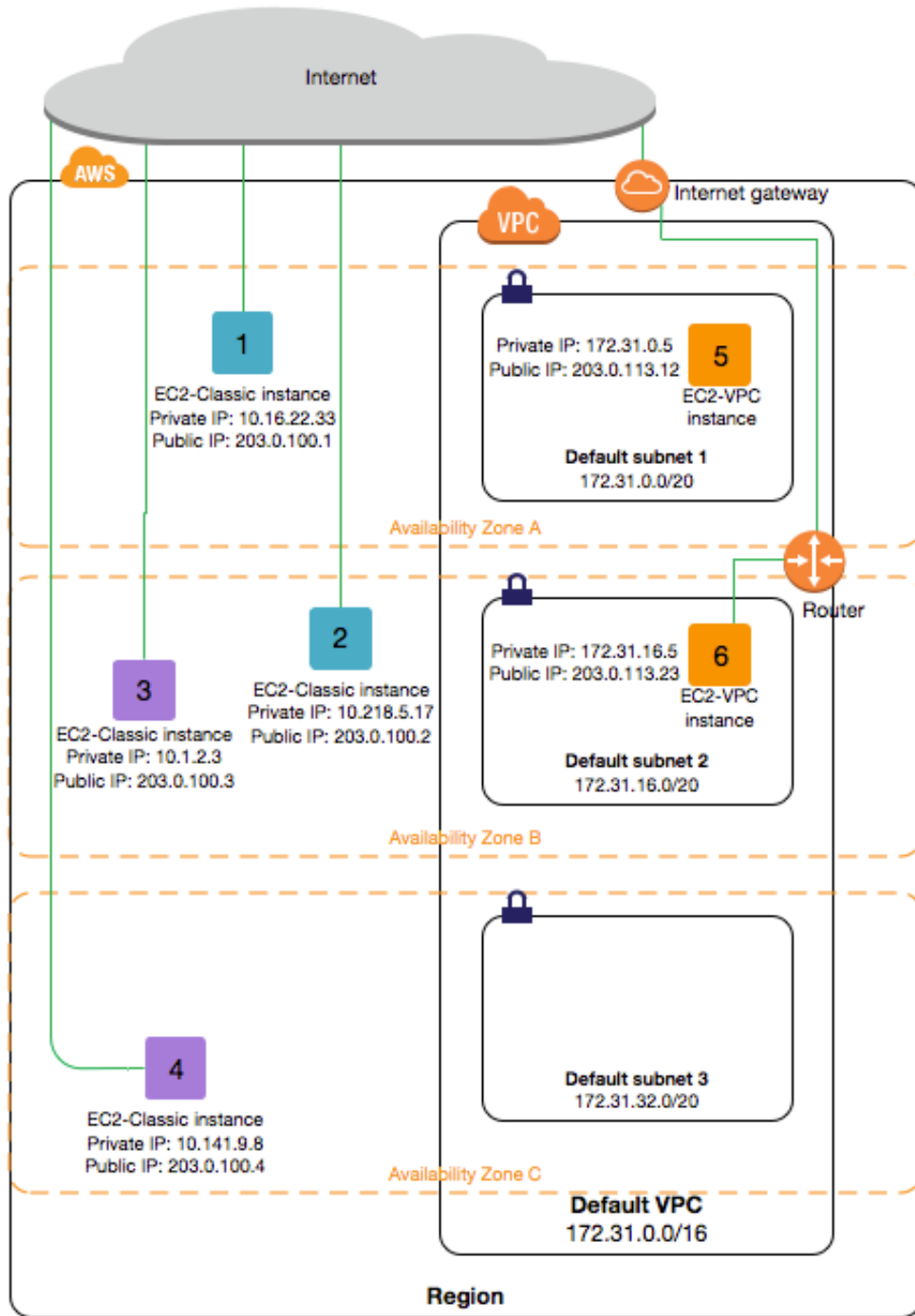
Characteristic	EC2-Classic	Default VPC	Nondefault VPC
Elastic IP address (IPv4)	An Elastic IP is disassociated from your instance when you stop it.	An Elastic IP remains associated with your instance when you stop it.	An Elastic IP remains associated with your instance when you stop it.
DNS hostnames	DNS hostnames are enabled by default.	DNS hostnames are enabled by default.	DNS hostnames are disabled by default.
Security group	<p>A security group can reference security groups that belong to other AWS accounts.</p> <p>You can create up to 500 security groups in each region.</p>	<p>A security group can reference security groups for your VPC only.</p> <p>You can create up to 100 security groups per VPC.</p>	<p>A security group can reference security groups for your VPC only.</p> <p>You can create up to 100 security groups per VPC.</p>
Security group association	<p>You can assign an unlimited number of security groups to an instance when you launch it.</p> <p>You can't change the security groups of your running instance. You can either modify the rules of the assigned security groups, or replace the instance with a new one (create an AMI from the instance, launch a new instance from this AMI with the security groups that you need, disassociate any Elastic IP address from the original instance and associate it with the new instance, and then terminate the original instance).</p>	<p>You can assign up to 5 security groups to an instance.</p> <p>You can assign security groups to your instance when you launch it and while it's running.</p>	<p>You can assign up to 5 security groups to an instance.</p> <p>You can assign security groups to your instance when you launch it and while it's running.</p>
Security group rules	<p>You can add rules for inbound traffic only.</p> <p>You can add up to 100 rules to a security group.</p>	<p>You can add rules for inbound and outbound traffic.</p> <p>You can add up to 50 rules to a security group.</p>	<p>You can add rules for inbound and outbound traffic.</p> <p>You can add up to 50 rules to a security group.</p>
Tenancy	Your instance runs on shared hardware.	You can run your instance on shared hardware or single-tenant hardware.	You can run your instance on shared hardware or single-tenant hardware.

Amazon Elastic Compute Cloud
User Guide for Windows Instances
Differences Between EC2-Classic and EC2-VPC

Characteristic	EC2-Classic	Default VPC	Nondefault VPC
Accessing the Internet	Your instance can access the Internet. Your instance automatically receives a public IP address, and can access the Internet directly through the AWS network edge.	By default, your instance can access the Internet. Your instance receives a public IP address by default. An Internet gateway is attached to your default VPC, and your default subnet has a route to the Internet gateway.	By default, your instance cannot access the Internet. Your instance doesn't receive a public IP address by default. Your VPC may have an Internet gateway, depending on how it was created.
IPv6 addressing	IPv6 addressing is not supported. You cannot assign IPv6 addresses to your instances.	You can optionally associate an IPv6 CIDR block with your VPC, and assign IPv6 addresses to instances in your VPC.	You can optionally associate an IPv6 CIDR block with your VPC, and assign IPv6 addresses to instances in your VPC.

The following diagram shows instances in each platform. Note the following:

- Instances 1, 2, 3, and 4 are in the EC2-Classic platform. 1 and 2 were launched by one account, and 3 and 4 were launched by a different account. These instances can communicate with each other, can access the Internet directly.
- Instances 5 and 6 are in different subnets in the same VPC in the EC2-VPC platform. They were launched by the account that owns the VPC; no other account can launch instances in this VPC. These instances can communicate with each other and can access instances in EC2-Classic and the Internet through the Internet gateway.



Sharing and Accessing Resources Between EC2-Classic and EC2-VPC

Some resources and features in your AWS account can be shared or accessed between the EC2-Classic and EC2-VPC platforms, for example, through ClassicLink. For more information about ClassicLink, see [ClassicLink](#) (p. 673).

Amazon Elastic Compute Cloud
User Guide for Windows Instances
Sharing and Accessing Resources
Between EC2-Classic and EC2-VPC

If your account supports EC2-Classic, you might have set up resources for use in EC2-Classic. If you want to migrate from EC2-Classic to a VPC, you must recreate those resources in your VPC. For more information about migrating from EC2-Classic to a VPC, see [Migrating from a Windows Instance in EC2-Classic to a Windows Instance in a VPC \(p. 683\)](#).

The following resources can be shared or accessed between EC2-Classic and a VPC.

Resource	Notes
AMI	
Bundle task	
EBS volume	
Elastic IP address (IPv4)	You can migrate an Elastic IP address from EC2-Classic to EC2-VPC. You can't migrate an Elastic IP address that was originally allocated for use in a VPC to EC2-Classic. For more information, see Migrating an Elastic IP Address from EC2-Classic to EC2-VPC (p. 712) .
Instance	<p>An EC2-Classic instance can communicate with instances in a VPC using public IPv4 addresses, or you can use ClassicLink to enable communication over private IPv4 addresses.</p> <p>You can't migrate an instance from EC2-Classic to a VPC. However, you can migrate your application from an instance in EC2-Classic to an instance in a VPC. For more information, see Migrating from a Windows Instance in EC2-Classic to a Windows Instance in a VPC (p. 683).</p>
Key pair	
Load balancer	<p>If you're using ClassicLink, you can register a linked EC2-Classic instance with a load balancer in a VPC, provided that the VPC has a subnet in the same Availability Zone as the instance.</p> <p>You can't migrate a load balancer from EC2-Classic to a VPC. You can't register an instance in a VPC with a load balancer in EC2-Classic.</p>
Placement group	
Reserved Instance	You can change the network platform for your Reserved Instances from EC2-Classic to EC2-VPC.
Security group	<p>A linked EC2-Classic instance can use a VPC security groups through ClassicLink to control traffic to and from the VPC. VPC instances can't use EC2-Classic security groups.</p> <p>You can't migrate a security group from EC2-Classic to a VPC. You can copy rules from a security group in EC2-Classic to a security group</p>

Resource	Notes
	in a VPC. For more information, see Creating a Security Group (p. 610) .
Snapshot	

The following resources can't be shared or moved between EC2-Classic and a VPC:

- Spot instances

Instance Types Available Only in a VPC

Instances of the following instance types are not supported in EC2-Classic and must be launched in a VPC:

- C4
- M4
- P2
- R4
- T2
- X1

If your account supports EC2-Classic but you have not created a nondefault VPC, you can do one of the following to launch a VPC-only instance:

- Create a nondefault VPC and launch your VPC-only instance into it by specifying a subnet ID or a network interface ID in the request. Note that you must create a nondefault VPC if you do not have a default VPC and you are using the AWS CLI, Amazon EC2 API, or AWS SDK to launch a VPC-only instance. For more information, see [Create a Virtual Private Cloud \(VPC\) \(p. 17\)](#).
- Launch your VPC-only instance using the Amazon EC2 console. The Amazon EC2 console creates a nondefault VPC in your account and launches the instance into the subnet in the first Availability Zone. The console creates the VPC with the following attributes:
 - One subnet in each Availability Zone, with the public IPv4 addressing attribute set to `true` so that instances receive a public IPv4 address. For more information, see [IP Addressing in Your VPC](#) in the *Amazon VPC User Guide*.
 - An Internet gateway, and a main route table that routes traffic in the VPC to the Internet gateway. This enables the instances you launch in the VPC to communicate over the Internet. For more information, see [Internet Gateways](#) in the *Amazon VPC User Guide*.
 - A default security group for the VPC and a default network ACL that is associated with each subnet. For more information, see [Security in Your VPC](#) in the *Amazon VPC User Guide*.

If you have other resources in EC2-Classic, you can take steps to migrate them to EC2-VPC. For more information, see [Migrating from a Windows Instance in EC2-Classic to a Windows Instance in a VPC \(p. 683\)](#).

Amazon VPC Documentation

For more information about Amazon VPC, see the following documentation.

Guide	Description
Amazon VPC Getting Started Guide	Provides a hands-on introduction to Amazon VPC.
Amazon VPC User Guide	Provides detailed information about how to use Amazon VPC.
Amazon VPC Network Administrator Guide	Helps network administrators configure your customer gateway.

Supported Platforms

Amazon EC2 supports the following platforms. Your AWS account is capable of launching instances either into both platforms or only into EC2-VPC, on a region by region basis.

Platform	Introduced In	Description
EC2-Classic	The original release of Amazon EC2	Your instances run in a single, flat network that you share with other customers.
EC2-VPC	The original release of Amazon VPC	Your instances run in a virtual private cloud (VPC) that's logically isolated to your AWS account.

For more information about the availability of either platform in your account, see [Availability](#) in the *Amazon VPC User Guide*. For more information about the differences between EC2-Classic and EC2-VPC, see [Differences Between EC2-Classic and EC2-VPC](#) (p. 666).

Supported Platforms in the Amazon EC2 Console

The Amazon EC2 console indicates which platforms you can launch instances into for the selected region, and whether you have a default VPC in that region.

Verify that the region you'll use is selected in the navigation bar. On the Amazon EC2 console dashboard, look for **Supported Platforms** under **Account Attributes**. If there are two values, `EC2` and `VPC`, you can launch instances into either platform. If there is one value, `VPC`, you can launch instances only into EC2-VPC.

If you can launch instances only into EC2-VPC, we create a default VPC for you. Then, when you launch an instance, we launch it into your default VPC, unless you create a nondefault VPC and specify it when you launch the instance.

EC2-VPC

The dashboard displays the following under **Account Attributes** to indicate that the account supports only the EC2-VPC platform, and has a default VPC with the identifier `vpc-1a2b3c4d`.




Supported Platforms

VPC

Default VPC

`vpc-1a2b3c4d`

If your account supports only EC2-VPC, you can select a VPC from the **Network** list, and a subnet from the **Subnet** list when you launch an instance using the launch wizard.

Network		vpc-1a2b3c4d (172.31.0.0/16) (default)		Create new VPC
Subnet		No preference (default subnet in any Availability Zor		Create new subnet

EC2-Classic, EC2-VPC

The dashboard displays the following under **Account Attributes** to indicate that the account supports both the EC2-Classic and EC2-VPC platforms.

Supported Platforms

EC2
VPC

If your account supports EC2-Classic and EC2-VPC, you can launch into EC2-Classic using the launch wizard by selecting **Launch into EC2-Classic** from the **Network** list. To launch into a VPC, you can select a VPC from the **Network** list, and a subnet from the **Subnet** list.

Related Topic

For more information about how you can tell which platforms you can launch instances into, see [Detecting Your Supported Platforms](#) in the *Amazon VPC User Guide*.

ClassicLink

ClassicLink allows you to link your EC2-Classic instance to a VPC in your account, within the same region. This allows you to associate the VPC security groups with the EC2-Classic instance, enabling communication between your EC2-Classic instance and instances in your VPC using private IPv4 addresses. ClassicLink removes the need to make use of public IPv4 addresses or Elastic IP addresses to enable communication between instances in these platforms. For more information about private and public IPv4 addresses, see [IP Addressing in Your VPC](#).

ClassicLink is available to all users with accounts that support the EC2-Classic platform, and can be used with any EC2-Classic instance. To find out which platform your account supports, see [Supported Platforms](#) (p. 672). For more information about the benefits of using a VPC, see [Amazon EC2 and Amazon Virtual Private Cloud](#) (p. 665). For more information about migrating your resources to a VPC, see [Migrating from a Windows Instance in EC2-Classic to a Windows Instance in a VPC](#) (p. 683).

There is no additional charge for using ClassicLink. Standard charges for data transfer and instance hour usage apply.

Note

EC2-Classic instances cannot be enabled for IPv6 communication. You can associate an IPv6 CIDR block with your VPC and assign IPv6 address to resources in your VPC, however, communication between a ClassicLinked instance and resources in the VPC is over IPv4 only.

Topics

- [ClassicLink Basics](#) (p. 673)
- [ClassicLink Limitations](#) (p. 676)
- [Working with ClassicLink](#) (p. 677)
- [API and CLI Overview](#) (p. 680)
- [Example: ClassicLink Security Group Configuration for a Three-Tier Web Application](#) (p. 681)

ClassicLink Basics

There are two steps to linking an EC2-Classic instance to a VPC using ClassicLink. First, you must enable the VPC for ClassicLink. By default, all VPCs in your account are not enabled for ClassicLink,

to maintain their isolation. After you've enabled the VPC for ClassicLink, you can then link any running EC2-Classic instance in the same region in your account to that VPC. Linking your instance includes selecting security groups from the VPC to associate with your EC2-Classic instance. After you've linked the instance, it can communicate with instances in your VPC using their private IP addresses, provided the VPC security groups allow it. Your EC2-Classic instance does not lose its private IP address when linked to the VPC.

Note

Linking your instance to a VPC is sometimes referred to as *attaching* your instance.

A linked EC2-Classic instance can communicate with instances in a VPC, but it does not form part of the VPC. If you list your instances and filter by VPC, for example, through the `DescribeInstances` API request, or by using the **Instances** screen in the Amazon EC2 console, the results do not return any EC2-Classic instances that are linked to the VPC. For more information about viewing your linked EC2-Classic instances, see [Viewing Your ClassicLink-Enabled VPCs and Linked EC2-Classic Instances \(p. 678\)](#).

By default, if you use a public DNS hostname to address an instance in a VPC from a linked EC2-Classic instance, the hostname resolves to the instance's public IP address. The same occurs if you use a public DNS hostname to address a linked EC2-Classic instance from an instance in the VPC. If you want the public DNS hostname to resolve to the private IP address, you can enable ClassicLink DNS support for the VPC. For more information, see [Enabling ClassicLink DNS Support \(p. 679\)](#).

If you no longer require a ClassicLink connection between your instance and the VPC, you can unlink the EC2-Classic instance from the VPC. This disassociates the VPC security groups from the EC2-Classic instance. A linked EC2-Classic instance is automatically unlinked from a VPC when it's stopped. After you've unlinked all linked EC2-Classic instances from the VPC, you can disable ClassicLink for the VPC.

Using Other AWS Services in Your VPC With ClassicLink

Linked EC2-Classic instances can access the following AWS services in the VPC: Amazon Redshift, Amazon ElastiCache, Elastic Load Balancing, and Amazon RDS. However, instances in the VPC cannot access the AWS services provisioned by the EC2-Classic platform using ClassicLink.

If you use Elastic Load Balancing in your VPC, you can register your linked EC2-Classic instance with the load balancer, provided that the instance is in an Availability Zone in which your VPC has a subnet. If you terminate the linked EC2-Classic instance, the load balancer deregisters the instance. For more information about working with load balancers in a VPC, see [Elastic Load Balancing in Amazon VPC](#) in the *Elastic Load Balancing User Guide*.

If you use Auto Scaling, you can create an Auto Scaling group with instances that are automatically linked to a specified ClassicLink-enabled VPC at launch. For more information, see [Linking EC2-Classic Instances to a VPC](#) in the *Auto Scaling User Guide*.

If you use Amazon RDS instances or Amazon Redshift clusters in your VPC, and they are publicly accessible (accessible from the Internet), the endpoint you use to address those resources from a linked EC2-Classic instance by default resolves to a public IP address. If those resources are not publicly accessible, the endpoint resolves to a private IP address. To address a publicly accessible RDS instance or Redshift cluster over private IP using ClassicLink, you must use their private IP address or private DNS hostname, or you must enable ClassicLink DNS support for the VPC.

If you use a private DNS hostname or a private IP address to address an RDS instance, the linked EC2-Classic instance cannot use the failover support available for Multi-AZ deployments.

You can use the Amazon EC2 console to find the private IP addresses of your Amazon Redshift, Amazon ElastiCache, or Amazon RDS resources.

To locate the private IP addresses of AWS resources in your VPC

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

2. In the navigation pane, choose **Network Interfaces**.
3. Check the descriptions of the network interfaces in the **Description** column. A network interface that's used by Amazon Redshift, Amazon ElastiCache, or Amazon RDS will have the name of the service in the description. For example, a network interface that's attached to an Amazon RDS instance will have the following description: `RDSNetworkInterface`.
4. Select the required network interface.
5. In the details pane, get the private IP address from the **Primary private IPv4 IP** field.

Controlling the Use of ClassicLink

By default, IAM users do not have permission to work with ClassicLink. You can create an IAM policy that grants users permissions to enable or disable a VPC for ClassicLink, link or unlink an instance to a ClassicLink-enabled VPC, and to view ClassicLink-enabled VPCs and linked EC2-Classic instances. For more information about IAM policies for Amazon EC2, see [IAM Policies for Amazon EC2](#) (p. 621).

For more information about policies for working with ClassicLink, see the following example: [5. Working with ClassicLink](#) (p. 649).

Security Groups in ClassicLink

Linking your EC2-Classic instance to a VPC does not affect your EC2-Classic security groups. They continue to control all traffic to and from the instance. This excludes traffic to and from instances in the VPC, which is controlled by the VPC security groups that you associated with the EC2-Classic instance. EC2-Classic instances that are linked to the same VPC cannot communicate with each other through the VPC; regardless of whether they are associated with the same VPC security group. Communication between EC2-Classic instances is controlled by the EC2-Classic security groups associated with those instances. For an example of a security group configuration, see [Example: ClassicLink Security Group Configuration for a Three-Tier Web Application](#) (p. 681).

After you've linked your instance to a VPC, you cannot change which VPC security groups are associated with the instance. To associate different security groups with your instance, you must first unlink the instance, and then link it to the VPC again, choosing the required security groups.

Routing for ClassicLink

When you enable a VPC for ClassicLink, a static route is added to all of the VPC route tables with a destination of `10.0.0.0/8` and a target of `local`. This allows communication between instances in the VPC and any EC2-Classic instances that are then linked to the VPC. If you add a custom route table to a ClassicLink-enabled VPC, a static route is automatically added with a destination of `10.0.0.0/8` and a target of `local`. When you disable ClassicLink for a VPC, this route is automatically deleted in all of the VPC route tables.

VPCs that are in the `10.0.0.0/16` and `10.1.0.0/16` IP address ranges can be enabled for ClassicLink only if they do not have any existing static routes in route tables in the `10.0.0.0/8` IP address range, excluding the local routes that were automatically added when the VPC was created. Similarly, if you've enabled a VPC for ClassicLink, you may not be able to add any more specific routes to your route tables within the `10.0.0.0/8` IP address range.

Important

If your VPC CIDR block is a publicly routable IP address range, consider the security implications before you link an EC2-Classic instance to your VPC. For example, if your linked EC2-Classic instance receives an incoming Denial of Service (DoS) request flood attack from a source IP address that falls within the VPC's IP address range, the response traffic is sent into your VPC. We strongly recommend that you create your VPC using a private IP address range as specified in [RFC 1918](#).

For more information about route tables and routing in your VPC, see [Route Tables](#) in the *Amazon VPC User Guide*.

Enabling a VPC Peering Connection for ClassicLink

If you have a VPC peering connection between two VPCs, and there are one or more EC2-Classic instances that are linked to one or both of the VPCs via ClassicLink, you can extend the VPC peering connection to enable communication between the EC2-Classic instances and the instances in the VPC on the other side of the VPC peering connection. This enables the EC2-Classic instances and the instances in the VPC to communicate using private IP addresses. To do this, you can enable a local VPC to communicate with a linked EC2-Classic instance in a peer VPC, or you can enable a local linked EC2-Classic instance to communicate with instances in a peer VPC.

If you enable a local VPC to communicate with a linked EC2-Classic instance in a peer VPC, a static route is automatically added to your route tables with a destination of `10.0.0.0/8` and a target of `local`.

For more information and examples, see [Configurations With ClassicLink](#) in the *Amazon VPC Peering Guide*.

ClassicLink Limitations

To use the ClassicLink feature, you need to be aware of the following limitations:

- You can link an EC2-Classic instance to only one VPC at a time.
- If you stop your linked EC2-Classic instance, it's automatically unlinked from the VPC, and the VPC security groups are no longer associated with the instance. You can link your instance to the VPC again after you've restarted it.
- You cannot link an EC2-Classic instance to a VPC that's in a different region, or a different AWS account.
- VPCs configured for dedicated hardware tenancy cannot be enabled for ClassicLink. Contact AWS support to request that your dedicated tenancy VPC be allowed to be enabled for ClassicLink.

Important

EC2-Classic instances are run on shared hardware. If you've set the tenancy of your VPC to `dedicated` because of regulatory or security requirements, then linking an EC2-Classic instance to your VPC may not conform to those requirements, as you will be allowing a shared tenancy resource to address your isolated resources directly using private IP addresses. If you want to enable your dedicated VPC for ClassicLink, provide a detailed motivation in your request to AWS support.

- VPCs with routes that conflict with the EC2-Classic private IP address range of `10/8` cannot be enabled for ClassicLink. This does not include VPCs with `10.0.0.0/16` and `10.1.0.0/16` IP address ranges that already have local routes in their route tables. For more information, see [Routing for ClassicLink \(p. 675\)](#).
- You cannot associate a VPC Elastic IP address with a linked EC2-Classic instance.
- You can link a running Spot instance to a VPC. To indicate in a Spot instance request that the instance should be linked to a VPC when the request is fulfilled, you must use the launch wizard in the Amazon EC2 console.
- ClassicLink does not support transitive relationships out of the VPC. Your linked EC2-Classic instance will not have access to any VPN connection, VPC endpoint, or Internet gateway associated with the VPC. Similarly, resources on the other side of a VPN connection, or an Internet gateway will not have access to a linked EC2-Classic instance.
- You cannot use ClassicLink to link a VPC instance to a different VPC, or to a EC2-Classic resource. To establish a private connection between VPCs, you can use a VPC peering connection. For more information, see [VPC Peering](#) in the *Amazon VPC User Guide*.

- If you link your EC2-Classic instance to a VPC in the 172.16.0.0/16 range, and you have a DNS server running on the 172.16.0.23/32 IP address within the VPC, then your linked EC2-Classic instance will not be able to access the VPC DNS server. To work around this issue, run your DNS server on a different IP address within the VPC.

Working with ClassicLink

You can use the Amazon EC2 and Amazon VPC consoles to work with the ClassicLink feature. You can enable or disable a VPC for ClassicLink, and link and unlink EC2-Classic instances to a VPC.

Note

The ClassicLink features are only visible in the consoles for accounts and regions that support EC2-Classic.

Topics

- [Enabling a VPC for ClassicLink \(p. 677\)](#)
- [Linking an Instance to a VPC \(p. 677\)](#)
- [Creating a VPC with ClassicLink Enabled \(p. 678\)](#)
- [Linking an EC2-Classic Instance to a VPC at Launch \(p. 678\)](#)
- [Viewing Your ClassicLink-Enabled VPCs and Linked EC2-Classic Instances \(p. 678\)](#)
- [Enabling ClassicLink DNS Support \(p. 679\)](#)
- [Disabling ClassicLink DNS Support \(p. 679\)](#)
- [Unlinking an EC2-Classic Instance from a VPC \(p. 679\)](#)
- [Disabling ClassicLink for a VPC \(p. 680\)](#)

Enabling a VPC for ClassicLink

To link an EC2-Classic instance to a VPC, you must first enable the VPC for ClassicLink. You cannot enable a VPC for ClassicLink if the VPC has routing that conflicts with the EC2-Classic private IP address range. For more information, see [Routing for ClassicLink \(p. 675\)](#).

To enable a VPC for ClassicLink

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Your VPCs**.
3. Choose a VPC, and then choose **Actions, Enable ClassicLink**.
4. In the confirmation dialog box, choose **Yes, Enable**.

Linking an Instance to a VPC

After you've enabled a VPC for ClassicLink, you can link an EC2-Classic instance to it.

Note

You can only link a running EC2-Classic instance to a VPC. You cannot link an instance that's in the `stopped` state.

To link an instance to a VPC

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the running EC2-Classic instance, choose **Actions, ClassicLink, Link to VPC**. You can select more than one instance to link to the same VPC.

4. In the dialog box that displays, select a VPC from the list. Only VPCs that have been enabled for ClassicLink are displayed.
5. Select one or more of the VPC security groups to associate with your instance. When you are done, choose **Link to VPC**.

Creating a VPC with ClassicLink Enabled

You can create a new VPC and immediately enable it for ClassicLink by using the VPC wizard in the Amazon VPC console.

To create a VPC with ClassicLink enabled

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. From the Amazon VPC dashboard, choose **Start VPC Wizard**.
3. Select one of the VPC configuration options and choose **Select**.
4. On the next page of the wizard, choose **Yes** for **Enable ClassicLink**. Complete the rest of the steps in the wizard to create your VPC. For more information about using the VPC wizard, see [Scenarios for Amazon VPC](#) in the *Amazon VPC User Guide*.

Linking an EC2-Classic Instance to a VPC at Launch

You can use the launch wizard in the Amazon EC2 console to launch an EC2-Classic instance and immediately link it to a ClassicLink-enabled VPC.

To link an instance to a VPC at launch

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the Amazon EC2 dashboard, choose **Launch Instance**.
3. Select an AMI, and then choose an instance type. On the **Configure Instance Details** page, ensure that you select **Launch into EC2-Classic** from the **Network** list.

Note

Some instance types, such as T2 instance types, can only be launched into a VPC. Ensure that you select an instance type that can be launched into EC2-Classic.

4. In the **Link to VPC (ClassicLink)** section, select a VPC from **Link to VPC**. Only ClassicLink-enabled VPCs are displayed. Select the security groups from the VPC to associate with the instance. Complete the other configuration options on the page, and then complete the rest of the steps in the wizard to launch your instance. For more information about using the launch wizard, see [Launching Your Instance from an AMI](#) (p. 245).

Viewing Your ClassicLink-Enabled VPCs and Linked EC2-Classic Instances

You can view all of your ClassicLink-enabled VPCs in the Amazon VPC console, and your linked EC2-Classic instances in the Amazon EC2 console.

To view your ClassicLink-enabled VPCs

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Your VPCs**.
3. Select a VPC, and in the **Summary** tab, look for the **ClassicLink** field. A value of **Enabled** indicates that the VPC is enabled for ClassicLink.
4. Alternatively, look for the **ClassicLink** column, and view the value that's displayed for each VPC (**Enabled** or **Disabled**). If the column is not visible, choose **Edit Table Columns** (the gear-shaped icon), select the **ClassicLink** attribute, and then choose **Close**.

To view your linked EC2-Classic instances

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select an EC2-Classic instance, and in the **Description** tab, look for the **ClassicLink** field. If the instance is linked to a VPC, the field displays the ID of the VPC to which the instance is linked. If the instance is not linked to any VPC, the field displays **Unlinked**.
4. Alternatively, you can filter your instances to display only linked EC2-Classic instances for a specific VPC or security group. In the search bar, start typing `ClassicLink`, select the relevant ClassicLink resource attribute, and then select the security group ID or the VPC ID.

Enabling ClassicLink DNS Support

You can enable ClassicLink DNS support for your VPC so that DNS hostnames that are addressed between linked EC2-Classic instances and instances in the VPC resolve to private IP addresses and not public IP addresses. For this feature to work, your VPC must be enabled for DNS hostnames and DNS resolution.

Note

If you enable ClassicLink DNS support for your VPC, your linked EC2-Classic instance can access any private hosted zone associated with the VPC. For more information, see [Working with Private Hosted Zones](#) in the *Amazon Route 53 Developer Guide*.

To enable ClassicLink DNS support

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Your VPCs**.
3. Select your VPC, and choose **Actions, Edit ClassicLink DNS Support**.
4. Choose **Yes** to enable ClassicLink DNS support, and choose **Save**.

Disabling ClassicLink DNS Support

You can disable ClassicLink DNS support for your VPC so that DNS hostnames that are addressed between linked EC2-Classic instances and instances in the VPC resolve to public IP addresses and not private IP addresses.

To disable ClassicLink DNS support

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Your VPCs**.
3. Select your VPC, and choose **Actions, Edit ClassicLink DNS Support**.
4. Choose **No** to disable ClassicLink DNS support, and choose **Save**.

Unlinking a EC2-Classic Instance from a VPC

If you no longer require a ClassicLink connection between your EC2-Classic instance and your VPC, you can unlink the instance from the VPC. Unlinking the instance disassociates the VPC security groups from the instance.

Note

A stopped instance is automatically unlinked from a VPC.

To unlink an instance from a VPC

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

2. In the navigation pane, choose **Instances**, and select your instance.
3. In the **Actions** list, select **ClassicLink, Unlink Instance**. You can select more than one instance to unlink from the same VPC.
4. Choose **Yes** in the confirmation dialog box.

Disabling ClassicLink for a VPC

If you no longer require a connection between EC2-Classic instances and your VPC, you can disable ClassicLink on the VPC. You must first unlink all linked EC2-Classic instances that are linked to the VPC.

To disable ClassicLink for a VPC

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Your VPCs**.
3. Select your VPC, then choose **Actions, Disable ClassicLink**.
4. In the confirmation dialog box, choose **Yes, Disable**.

API and CLI Overview

You can perform the tasks described on this page using the command line or the Query API. For more information about the command line interfaces and a list of available API actions, see [Accessing Amazon EC2 \(p. 3\)](#).

Enable a VPC for ClassicLink

- [enable-vpc-classic-link](#) (AWS CLI)
- [Enable-EC2VpcClassicLink](#) (AWS Tools for Windows PowerShell)
- [EnableVpcClassicLink](#) (Amazon EC2 Query API)

Link (attach) an EC2-Classic instance to a VPC

- [attach-classic-link-vpc](#) (AWS CLI)
- [Add-EC2ClassicLinkVpc](#) (AWS Tools for Windows PowerShell)
- [AttachClassicLinkVpc](#) (Amazon EC2 Query API)

Unlink (detach) an EC2-Classic instance from a VPC

- [detach-classic-link-vpc](#) (AWS CLI)
- [Dismount-EC2ClassicLinkVpc](#) (AWS Tools for Windows PowerShell)
- [DetachClassicLinkVpc](#) (Amazon EC2 Query API)

Disable ClassicLink for a VPC

- [disable-vpc-classic-link](#) (AWS CLI)
- [Disable-EC2VpcClassicLink](#) (AWS Tools for Windows PowerShell)
- [DisableVpcClassicLink](#) (Amazon EC2 Query API)

Describe the ClassicLink status of VPCs

- [describe-vpc-classic-link](#) (AWS CLI)

- [Get-EC2VpcClassicLink](#) (AWS Tools for Windows PowerShell)
- [DescribeVpcClassicLink](#) (Amazon EC2 Query API)

Describe linked EC2-Classic instances

- [describe-classic-link-instances](#) (AWS CLI)
- [Get-EC2ClassicLinkInstance](#) (AWS Tools for Windows PowerShell)
- [DescribeClassicLinkInstances](#) (Amazon EC2 Query API)

Enable a VPC peering connection for ClassicLink

- [modify-vpc-peering-connection-options](#) (AWS CLI)
- [Edit-EC2VpcPeeringConnectionOption](#) (AWS Tools for Windows PowerShell)
- [ModifyVpcPeeringConnectionOptions](#)(Amazon EC2 Query API)

Enable a VPC for ClassicLink DNS support

- [enable-vpc-classic-link-dns-support](#) (AWS CLI)
- [Enable-EC2VpcClassicLinkDnsSupport](#) (AWS Tools for Windows PowerShell)
- [EnableVpcClassicLinkDnsSupport](#) (Amazon EC2 Query API)

Disable a VPC for ClassicLink DNS support

- [disable-vpc-classic-link-dns-support](#) (AWS CLI)
- [Disable-EC2VpcClassicLinkDnsSupport](#) (AWS Tools for Windows PowerShell)
- [DisableVpcClassicLinkDnsSupport](#) (Amazon EC2 Query API)

Describe ClassicLink DNS support for VPCs

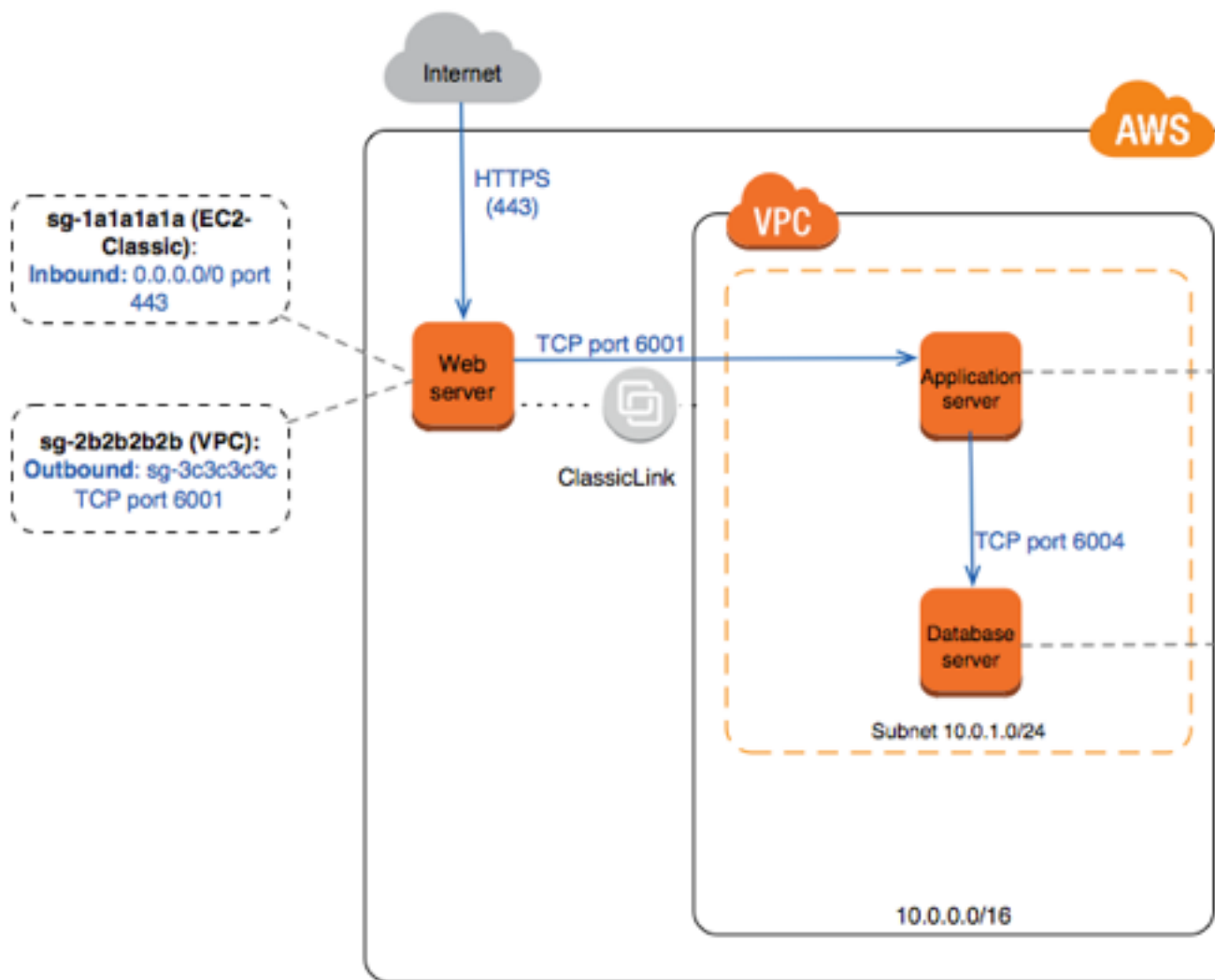
- [describe-vpc-classic-link-dns-support](#) (AWS CLI)
- [Get-EC2VpcClassicLinkDnsSupport](#) (AWS Tools for Windows PowerShell)
- [DescribeVpcClassicLinkDnsSupport](#) (Amazon EC2 Query API)

Example: ClassicLink Security Group Configuration for a Three-Tier Web Application

In this example, you have an application with three instances: a public-facing web server, an application server, and a database server. Your web server accepts HTTPS traffic from the Internet, and then communicates with your application server over TCP port 6001. Your application server then communicates with your database server over TCP port 6004. You're in the process of migrating your entire application to a VPC in your account. You've already migrated your application server and your database server to your VPC. Your web server is still in EC2-Classic and linked to your VPC via ClassicLink.

You want a security group configuration that allows traffic to flow only between these instances. You have four security groups: two for your web server (`sg-1a1a1a1a` and `sg-2b2b2b2b`), one for your application server (`sg-3c3c3c3c`), and one for your database server (`sg-4d4d4d4d`).

The following diagram displays the architecture of your instances, and their security group configuration.



Security Groups for Your Web Server (sg-1a1a1a1a and sg-2b2b2b2b)

You have one security group in EC2-Classic, and the other in your VPC. You associated the VPC security group with your web server instance when you linked the instance to your VPC via ClassicLink. The VPC security group enables you to control the outbound traffic from your web server to your application server.

The following are the security group rules for the EC2-Classic security group (sg-1a1a1a1a).

Inbound			
Source	Type	Port Range	Comments
0.0.0.0/0	HTTPS	443	Allows Internet traffic to reach your web server.

The following are the security group rules for the VPC security group (sg-2b2b2b2b).

Outbound			
----------	--	--	--

Destination	Type	Port Range	Comments
sg-3c3c3c3c	TCP	6001	Allows outbound traffic from your web server to your application server in your VPC (or to any other instance associated with sg-3c3c3c3c).

Security Group for Your Application Server (sg-3c3c3c3c)

The following are the security group rules for the VPC security group that's associated with your application server.

Inbound			
Source	Type	Port Range	Comments
sg-2b2b2b2b	TCP	6001	Allows the specified type of traffic from your web server (or any other instance associated with sg-2b2b2b2b) to reach your application server.
Outbound			
Destination	Type	Port Range	Comments
sg-4d4d4d4d	TCP	6004	Allows outbound traffic from the application server to the database server (or to any other instance associated with sg-4d4d4d4d).

Security Group for Your Database Server (sg-4d4d4d4d)

The following are the security group rules for the VPC security group that's associated with your database server.

Inbound			
Source	Type	Port Range	Comments
sg-3c3c3c3c	TCP	6004	Allows the specified type of traffic from your application server (or any other instance associated with sg-3c3c3c3c) to reach your database server.

Migrating from a Windows Instance in EC2-Classic to a Windows Instance in a VPC

Your AWS account might support both EC2-Classic and EC2-VPC, depending on when you created your account and which regions you've used. For more information, and to find out which platform your account supports, see [Supported Platforms \(p. 672\)](#). For more information about the benefits of using a VPC, and the differences between EC2-Classic and EC2-VPC, see [Amazon EC2 and Amazon Virtual Private Cloud \(p. 665\)](#).

You create and use resources in your AWS account. Some resources and features, such as enhanced networking and certain instance types, can be used only in a VPC. Some resources can be shared

between EC2-Classic and a VPC, while some can't. For more information, see [Sharing and Accessing Resources Between EC2-Classic and EC2-VPC](#) (p. 669).

If your account supports EC2-Classic, you might have set up resources for use in EC2-Classic. If you want to migrate from EC2-Classic to a VPC, you must recreate those resources in your VPC.

There are two ways of migrating to a VPC. You can do a full migration, or you can do an incremental migration over time. The method you choose depends on the size and complexity of your application in EC2-Classic. For example, if your application consists of one or two instances running a static website, and you can afford a short period of downtime, you can do a full migration. If you have a multi-tier application with processes that cannot be interrupted, you can do an incremental migration using ClassicLink. This allows you to transfer functionality one component at a time until your application is running fully in your VPC.

If you need to migrate a Linux instance, see [Migrating a Linux Instance from EC2-Classic to a VPC](#) in the *Amazon EC2 User Guide for Linux Instances*.

Contents

- [Full Migration to a VPC](#) (p. 684)
- [Incremental Migration to a VPC Using ClassicLink](#) (p. 690)

Full Migration to a VPC

Complete the following tasks to fully migrate your application from EC2-Classic to a VPC.

Tasks

- [Step 1: Create a VPC](#) (p. 684)
- [Step 2: Configure Your Security Group](#) (p. 684)
- [Step 3: Create an AMI from Your EC2-Classic Instance](#) (p. 685)
- [Step 4: Launch an Instance Into Your VPC](#) (p. 686)
- [Example: Migrating a Simple Web Application](#) (p. 687)

Step 1: Create a VPC

To start using a VPC, ensure that you have one in your account. You can create one using one of these methods:

- Use a new, EC2-VPC-only AWS account. Your EC2-VPC-only account comes with a default VPC in each region, which is ready for you to use. Instances that you launch are by default launched into this VPC, unless you specify otherwise. For more information about your default VPC, see [Your Default VPC and Subnets](#). Use this option if you'd prefer not to set up a VPC yourself, or if you do not need specific requirements for your VPC configuration.
- In your existing AWS account, open the Amazon VPC console and use the VPC wizard to create a new VPC. For more information, see [Scenarios for Amazon VPC](#). Use this option if you want to set up a VPC quickly in your existing EC2-Classic account, using one of the available configuration sets in the wizard. You'll specify this VPC each time you launch an instance.
- In your existing AWS account, open the Amazon VPC console and set up the components of a VPC according to your requirements. For more information, see [Your VPC and Subnets](#). Use this option if you have specific requirements for your VPC, such as a particular number of subnets. You'll specify this VPC each time you launch an instance.

Step 2: Configure Your Security Group

You cannot use the same security groups between EC2-Classic and a VPC. However, if you want your instances in your VPC to have the same security group rules as your EC2-Classic instances, you can

use the Amazon EC2 console to copy your existing EC2-Classic security group rules to a new VPC security group.

Important

You can only copy security group rules to a new security group in the same AWS account in the same region. If you've created a new AWS account, you cannot use this method to copy your existing security group rules to your new account. You'll have to create a new security group, and add the rules yourself. For more information about creating a new security group, see [Amazon EC2 Security Groups for Windows Instances \(p. 606\)](#).

To copy your security group rules to a new security group

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups**.
3. Select the security group that's associated with your EC2-Classic instance, then choose **Actions** and select **Copy to new**.
4. In the **Create Security Group** dialog box, specify a name and description for your new security group. Select your VPC from the **VPC** list.
5. The **Inbound** tab is populated with the rules from your EC2-Classic security group. You can modify the rules as required. In the **Outbound** tab, a rule that allows all outbound traffic has automatically been created for you. For more information about modifying security group rules, see [Amazon EC2 Security Groups for Windows Instances \(p. 606\)](#).

Note

If you've defined a rule in your EC2-Classic security group that references another security group, you will not be able to use the same rule in your VPC security group. Modify the rule to reference a security group in the same VPC.

6. Choose **Create**.

Step 3: Create an AMI from Your EC2-Classic Instance

An AMI is a template for launching your instance. You can create your own AMI based on an existing EC2-Classic instance, then use that AMI to launch instances into your VPC.

The method you use to create your AMI depends on the root device type of your instance, and the operating system platform on which your instance runs. To find out the root device type of your instance, go to the **Instances** page, select your instance, and look at the information in the **Root device type** field in the **Description** tab. If the value is `ebs`, then your instance is EBS-backed. If the value is `instance-store`, then your instance is instance store-backed. You can also use the [describe-instances](#) AWS CLI command to find out the root device type.

The following table provides options for you to create your AMI based on the root device type of your instance, and the software platform.

Instance Root Device Type	Action
EBS	Create an EBS-backed AMI from your instance. For more information, see Creating an Amazon EBS-Backed Windows AMI (p. 77) .
Instance store	Bundle your instance, and then create an instance store-backed AMI from the manifest that's created during bundling. For more information, see Creating an Instance Store-Backed Windows AMI (p. 80) .

(Optional) Store Your Data on Amazon EBS Volumes

You can create an Amazon EBS volume and use it to back up and store the data on your instance—like you would use a physical hard drive. Amazon EBS volumes can be attached and detached

from any instance in the same Availability Zone. You can detach a volume from your instance in EC2-Classic, and attach it to a new instance that you launch into your VPC in the same Availability Zone.

For more information about Amazon EBS volumes, see the following topics:

- [Amazon EBS Volumes \(p. 747\)](#)
- [Creating an Amazon EBS Volume \(p. 761\)](#)
- [Attaching an Amazon EBS Volume to an Instance \(p. 766\)](#)

To back up the data on your Amazon EBS volume, you can take periodic snapshots of your volume. If you need to, you can restore an Amazon EBS volume from your snapshot. For more information about Amazon EBS snapshots, see the following topics:

- [Amazon EBS Snapshots \(p. 788\)](#)
- [Creating an Amazon EBS Snapshot \(p. 789\)](#)
- [Restoring an Amazon EBS Volume from a Snapshot \(p. 763\)](#)

Step 4: Launch an Instance Into Your VPC

After you've created an AMI, you can launch an instance into your VPC. The instance will have the same data and configurations as your existing EC2-Classic instance.

You can either launch your instance into a VPC that you've created in your existing account, or into a new, VPC-only AWS account.

Using Your Existing EC2-Classic Account

You can use the Amazon EC2 launch wizard to launch an instance into your VPC.

To launch an instance into your VPC

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the dashboard, choose **Launch Instance**.
3. On the **Choose an Amazon Machine Image** page, select the **My AMIs** category, and select the AMI you created.
4. On the **Choose an Instance Type** page, select the type of instance, and choose **Next: Configure Instance Details**.
5. On the **Configure Instance Details** page, select your VPC from the **Network** list. Select the required subnet from the **Subnet** list. Configure any other details you require, then go through the next pages of the wizard until you reach the **Configure Security Group** page.
6. Select **Select an existing group**, and select the security group you created earlier. Choose **Review and Launch**.
7. Review your instance details, then choose **Launch** to specify a key pair and launch your instance.

For more information about the parameters you can configure in each step of the wizard, see [Launching an Instance \(p. 244\)](#).

Using Your New, VPC-Only Account

To launch an instance in your new AWS account, you'll first have to share the AMI you created with your new account. You can then use the Amazon EC2 launch wizard to launch an instance into your default VPC.

To share an AMI with your new AWS account

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Switch to the account in which you created your AMI.
3. In the navigation pane, choose **AMIs**.
4. In the **Filter** list, ensure **Owned by me** is selected, then select your AMI.
5. In the **Permissions** tab, choose **Edit**. Enter the account number of your new AWS account, choose **Add Permission**, and then choose **Save**.

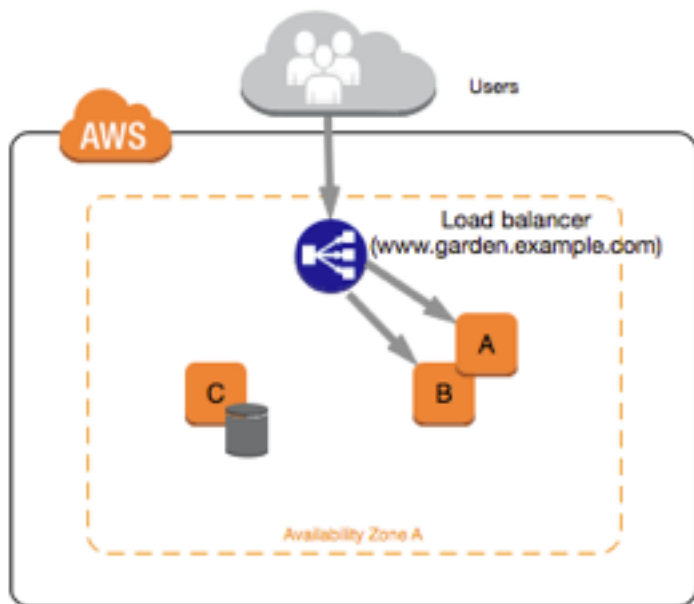
To launch an instance into your default VPC

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Switch to your new AWS account.
3. In the navigation pane, choose **AMIs**.
4. In the **Filter** list, select **Private images**. Select the AMI that you shared from your EC2-Classic account, then choose **Launch**.
5. On the **Choose an Instance Type** page, select the type of instance, and choose **Next: Configure Instance Details**.
6. On the **Configure Instance Details** page, your default VPC should be selected in the **Network** list. Configure any other details you require, then go through the next pages of the wizard until you reach the **Configure Security Group** page.
7. Select **Select an existing group**, and select the security group you created earlier. Choose **Review and Launch**.
8. Review your instance details, then choose **Launch** to specify a key pair and launch your instance.

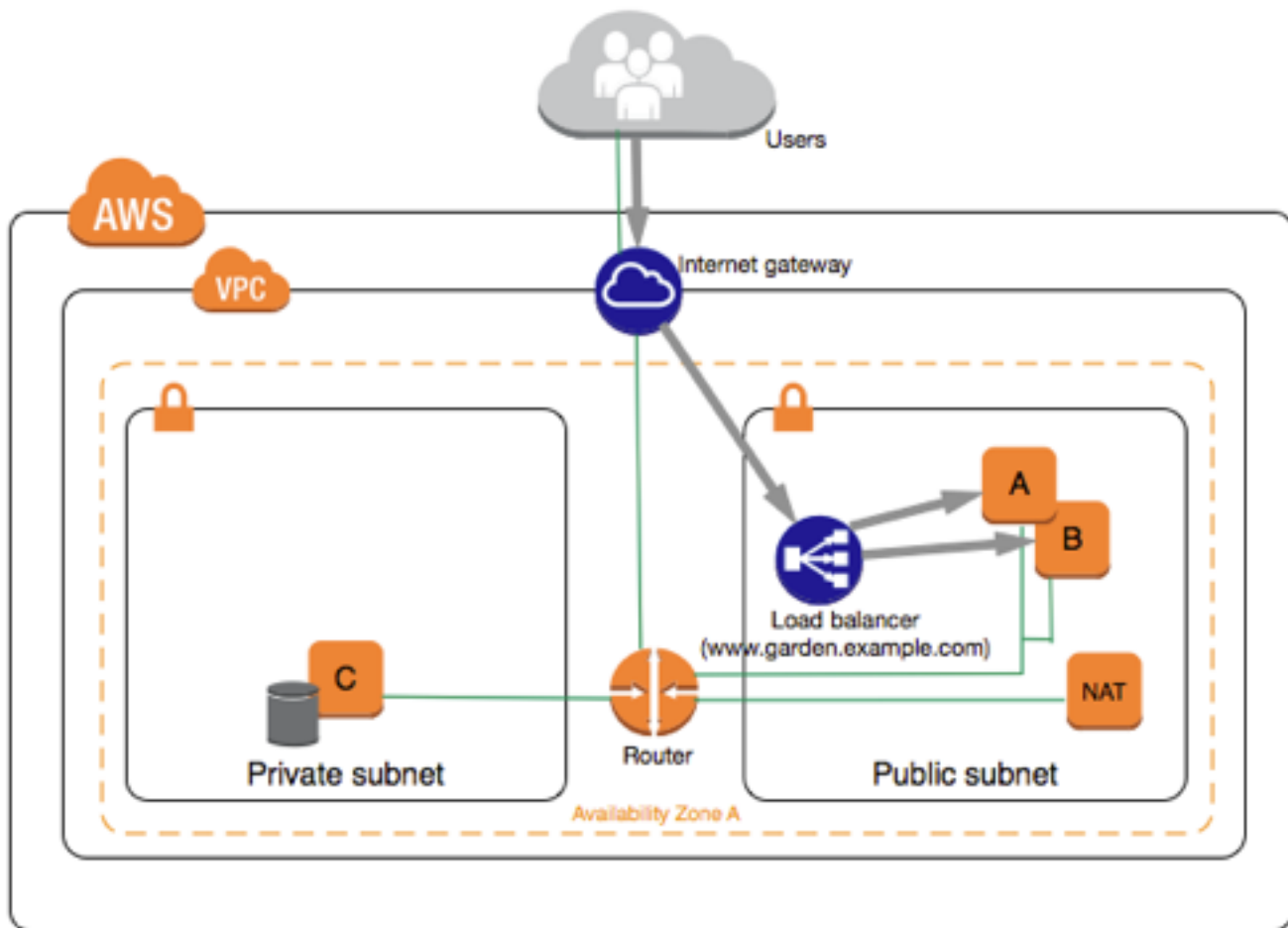
For more information about the parameters you can configure in each step of the wizard, see [Launching an Instance \(p. 244\)](#).

Example: Migrating a Simple Web Application

In this example, you use AWS to host your gardening website. To manage your website, you have three running instances in EC2-Classic. Instances A and B host your public-facing web application, and you use an Elastic Load Balancer to load balance the traffic between these instances. You've assigned Elastic IP addresses to instances A and B so that you have static IP addresses for configuration and administration tasks on those instances. Instance C holds your MySQL database for your website. You've registered the domain name `www.garden.example.com`, and you've used Amazon Route 53 to create a hosted zone with an alias record set that's associated with the DNS name of your load balancer.



The first part of migrating to a VPC is deciding what kind of VPC architecture will suit your needs. In this case, you've decided on the following: one public subnet for your web servers, and one private subnet for your database server. As your website grows, you can add more web servers and database servers to your subnets. By default, instances in the private subnet cannot access the Internet; however, you can enable Internet access through a Network Address Translation (NAT) device in the public subnet. You may want to set up a NAT device to support periodic updates and patches from the Internet for your database server. You'll migrate your Elastic IP addresses to EC2-VPC, and create an Elastic Load Balancer in your public subnet to load balance the traffic between your web servers.



To migrate your web application to a VPC, you can follow these steps:

- **Create a VPC:** In this case, you can use the VPC wizard in the Amazon VPC console to create your VPC and subnets. The second wizard configuration creates a VPC with one private and one public subnet, and launches and configures a NAT device in your public subnet for you. For more information, see [Scenario 2: VPC with Public and Private Subnets](#) in the *Amazon VPC User Guide*.
- **Create AMIs from your instances:** Create an AMI from one of your web servers, and a second AMI from your database server. For more information, see [Step 3: Create an AMI from Your EC2-Classical Instance](#) (p. 685).
- **Configure your security groups:** In your EC2-Classical environment, you have one security group for your web servers, and another security group for your database server. You can use the Amazon EC2 console to copy the rules from each security group into new security groups for your VPC. For more information, see [Step 2: Configure Your Security Group](#) (p. 684).

Tip

Create the security groups that are referenced by other security groups first.

- **Launch an instance into your new VPC:** Launch replacement web servers into your public subnet, and launch your replacement database server into your private subnet. For more information, see [Step 4: Launch an Instance Into Your VPC](#) (p. 686).
- **Configure your NAT device:** If you are using a NAT instance, you must create security group for it that allows HTTP and HTTPS traffic from your private subnet. For more information, see [NAT Instances](#). If you are using a NAT gateway, traffic from your private subnet is automatically allowed.

- **Configure your database:** When you created an AMI from your database server in EC2-Classic, all the configuration information that was stored in that instance was copied to the AMI. You may have to connect to your new database server and update the configuration details; for example, if you configured your database to grant full read, write, and modification permissions to your web servers in EC2-Classic, you'll have to update the configuration files to grant the same permissions to your new VPC web servers instead.
- **Configure your web servers:** Your web servers will have the same configuration settings as your instances in EC2-Classic. For example, if you configured your web servers to use the database in EC2-Classic, update your web servers' configuration settings to point to your new database instance.

Note

By default, instances launched into a nondefault subnet are not assigned a public IP address, unless you specify otherwise at launch. Your new database server may not have a public IP address. In this case, you can update your web servers' configuration file to use your new database server's private DNS name. Instances in the same VPC can communicate with each other via private IP address.

- **Migrate your Elastic IP addresses:** Disassociate your Elastic IP addresses from your web servers in EC2-Classic, and then migrate them to EC2-VPC. After you've migrated them, you can associate them with your new web servers in your VPC. For more information, see [Migrating an Elastic IP Address from EC2-Classic to EC2-VPC \(p. 712\)](#).
- **Create a new load balancer:** To continue using Elastic Load Balancing to load balance the traffic to your instances, make sure you understand the various ways you can configure your load balancer in VPC. For more information, see [Elastic Load Balancing in Amazon VPC](#).
- **Update your DNS records:** After you've set up your load balancer in your public subnet, ensure that your `www.garden.example.com` domain points to your new load balancer. To do this, you'll need to update your DNS records and update your alias record set in Amazon Route 53. For more information about using Amazon Route 53, see [Getting Started with Amazon Route 53](#).
- **Shut down your EC2-Classic resources:** After you've verified that your web application is working from within the VPC architecture, you can shut down your EC2-Classic resources to stop incurring charges for them. Terminate your EC2-Classic instances, and release your EC2-Classic Elastic IP addresses.

Incremental Migration to a VPC Using ClassicLink

The ClassicLink feature makes it easier to manage an incremental migration to a VPC. ClassicLink allows you to link an EC2-Classic instance to a VPC in your account in the same region, allowing your new VPC resources to communicate with the EC2-Classic instance using private IPv4 addresses. You can then migrate functionality to the VPC one step at a time. This topic provides some basic steps for managing an incremental migration from EC2-Classic to a VPC.

For more information about ClassicLink, see [ClassicLink \(p. 673\)](#).

Topics

- [Step 1: Prepare Your Migration Sequence \(p. 691\)](#)
- [Step 2: Create a VPC \(p. 691\)](#)
- [Step 3: Enable Your VPC for ClassicLink \(p. 691\)](#)
- [Step 4: Create an AMI from Your EC2-Classic Instance \(p. 691\)](#)
- [Step 5: Launch an Instance Into Your VPC \(p. 692\)](#)
- [Step 6: Link Your EC2-Classic Instances to Your VPC \(p. 693\)](#)
- [Step 7: Complete the VPC Migration \(p. 693\)](#)

Step 1: Prepare Your Migration Sequence

To use ClassicLink effectively, you must first identify the components of your application that must be migrated to the VPC, and then confirm the order in which to migrate that functionality.

For example, you have an application that relies on a presentation web server, a backend database server, and authentication logic for transactions. You may decide to start the migration process with the authentication logic, then the database server, and finally, the web server.

Step 2: Create a VPC

To start using a VPC, ensure that you have one in your account. You can create one using one of these methods:

- In your existing AWS account, open the Amazon VPC console and use the VPC wizard to create a new VPC. For more information, see [Scenarios for Amazon VPC](#). Use this option if you want to set up a VPC quickly in your existing EC2-Classic account, using one of the available configuration sets in the wizard. You'll specify this VPC each time you launch an instance.
- In your existing AWS account, open the Amazon VPC console and set up the components of a VPC according to your requirements. For more information, see [Your VPC and Subnets](#). Use this option if you have specific requirements for your VPC, such as a particular number of subnets. You'll specify this VPC each time you launch an instance.

Step 3: Enable Your VPC for ClassicLink

After you've created a VPC, you can enable it for ClassicLink. For more information about ClassicLink, see [ClassicLink \(p. 673\)](#).

To enable a VPC for ClassicLink

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Your VPCs**.
3. Select your VPC, and then select **Enable ClassicLink** from the **Actions** list.
4. In the confirmation dialog box, choose **Yes, Enable**.

Step 4: Create an AMI from Your EC2-Classic Instance

An AMI is a template for launching your instance. You can create your own AMI based on an existing EC2-Classic instance, then use that AMI to launch instances into your VPC.

The method you use to create your AMI depends on the root device type of your instance, and the operating system platform on which your instance runs. To find out the root device type of your instance, go to the **Instances** page, select your instance, and look at the information in the **Root device type** field in the **Description** tab. If the value is `ebs`, then your instance is EBS-backed. If the value is `instance-store`, then your instance is instance store-backed. You can also use the [describe-instances](#) AWS CLI command to find out the root device type.

The following table provides options for you to create your AMI based on the root device type of your instance, and the software platform.

Instance Root Device Type	Action
EBS	Create an EBS-backed AMI from your instance. For more information, see Creating an Amazon EBS-Backed Windows AMI (p. 77) .

Instance Root Device Type	Action
Instance store	Bundle your instance, and then create an instance store-backed AMI from the manifest that's created during bundling. For more information, see Creating an Instance Store-Backed Windows AMI (p. 80) .

(Optional) Store Your Data on Amazon EBS Volumes

You can create an Amazon EBS volume and use it to back up and store the data on your instance—like you would use a physical hard drive. Amazon EBS volumes can be attached and detached from any instance in the same Availability Zone. You can detach a volume from your instance in EC2-Classical, and attach it to a new instance that you launch into your VPC in the same Availability Zone.

For more information about Amazon EBS volumes, see the following topics:

- [Amazon EBS Volumes \(p. 747\)](#)
- [Creating an Amazon EBS Volume \(p. 761\)](#)
- [Attaching an Amazon EBS Volume to an Instance \(p. 766\)](#)

To back up the data on your Amazon EBS volume, you can take periodic snapshots of your volume. If you need to, you can restore an Amazon EBS volume from your snapshot. For more information about Amazon EBS snapshots, see the following topics:

- [Amazon EBS Snapshots \(p. 788\)](#)
- [Creating an Amazon EBS Snapshot \(p. 789\)](#)
- [Restoring an Amazon EBS Volume from a Snapshot \(p. 763\)](#)

Step 5: Launch an Instance Into Your VPC

The next step in the migration process is to launch instances into your VPC so that you can start transferring functionality to them. You can use the AMIs that you created in the previous step to launch instances into your VPC. The instances will have the same data and configurations as your existing EC2-Classical instances.

To launch an instance into your VPC using your custom AMI

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the dashboard, choose **Launch Instance**.
3. On the **Choose an Amazon Machine Image** page, select the **My AMIs** category, and select the AMI you created.
4. On the **Choose an Instance Type** page, select the type of instance, and choose **Next: Configure Instance Details**.
5. On the **Configure Instance Details** page, select your VPC from the **Network** list. Select the required subnet from the **Subnet** list. Configure any other details you require, then go through the next pages of the wizard until you reach the **Configure Security Group** page.
6. Select **Select an existing group**, and select the security group you created earlier. Choose **Review and Launch**.
7. Review your instance details, then choose **Launch** to specify a key pair and launch your instance.

For more information about the parameters you can configure in each step of the wizard, see [Launching an Instance \(p. 244\)](#).

After you've launched your instance and it's in the `running` state, you can connect to it and configure it as required.

Step 6: Link Your EC2-Classic Instances to Your VPC

After you've configured your instances and made the functionality of your application available in the VPC, you can use ClassicLink to enable private IP communication between your new VPC instances and your EC2-Classic instances.

To link an instance to a VPC

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select your EC2-Classic instance, then choose **Actions**, **ClassicLink**, and **Link to VPC**.

Note

Ensure that your instance is in the `running` state.

4. In the dialog box, select your ClassicLink-enabled VPC (only VPCs that are enabled for ClassicLink are displayed).
5. Select one or more of the VPC security groups to associate with your instance. When you are done, choose **Link to VPC**.

Step 7: Complete the VPC Migration

Depending on the size of your application and the functionality that must be migrated, repeat steps 4 to 6 until you've moved all the components of your application from EC2-Classic into your VPC.

After you've enabled internal communication between the EC2-Classic and VPC instances, you must update your application to point to your migrated service in your VPC, instead of your service in the EC2-Classic platform. The exact steps for this depend on your application's design. Generally, this includes updating your destination IP addresses to point to the IP addresses of your VPC instances instead of your EC2-Classic instances. You can migrate your Elastic IP addresses that you are currently using in the EC2-Classic platform to the EC2-VPC platform. For more information, see [Migrating an Elastic IP Address from EC2-Classic to EC2-VPC \(p. 712\)](#).

After you've completed this step and you've tested that the application is functioning from your VPC, you can terminate your EC2-Classic instances, and disable ClassicLink for your VPC. You can also clean up any EC2-Classic resources that you may no longer need to avoid incurring charges for them; for example, you can release Elastic IP addresses, and delete the volumes that were associated with your EC2-Classic instances.

Amazon EC2 Instance IP Addressing

We provide your instances with IP addresses and IPv4 DNS hostnames. These can vary depending on whether you launched the instance in the EC2-Classic platform or in a virtual private cloud (VPC). For information about the EC2-Classic and EC2-VPC platforms, see [Supported Platforms \(p. 672\)](#).

Amazon EC2 and Amazon VPC support both the IPv4 and IPv6 addressing protocols. By default, Amazon EC2 and Amazon VPC use the IPv4 addressing protocol; you can't disable this behavior. When you create a VPC, you must specify an IPv4 CIDR block (a range of private IPv4 addresses). You can optionally assign an IPv6 CIDR block to your VPC and subnets, and assign IPv6 addresses from that block to instances in your subnet. IPv6 addresses are reachable over the Internet. For more information about IPv6, see [IP Addressing in Your VPC](#) in the *Amazon VPC User Guide*.

IPv6 is not supported for the EC2-Classic platform. IPv6 is currently supported in the US East (Ohio) region only.

Contents

- [Private IPv4 Addresses and Internal DNS Hostnames \(p. 694\)](#)
- [Public IPv4 Addresses and External DNS Hostnames \(p. 695\)](#)
- [Elastic IP Addresses \(IPv4\) \(p. 696\)](#)
- [Amazon DNS Server \(p. 696\)](#)
- [IPv6 Addresses \(p. 696\)](#)
- [IP Address Differences Between EC2-Classic and EC2-VPC \(p. 697\)](#)
- [Working with IP Addresses for Your Instance \(p. 698\)](#)
- [Multiple IP Addresses \(p. 702\)](#)

Private IPv4 Addresses and Internal DNS Hostnames

A private IPv4 address is an IP address that's not reachable over the Internet. You can use private IPv4 addresses for communication between instances in the same network (EC2-Classic or a VPC). For more information about the standards and specifications of private IPv4 addresses, see [RFC 1918](#).

Note

You can create a VPC with a publicly routable CIDR block that falls outside of the private IPv4 address ranges specified in RFC 1918. However, for the purposes of this documentation, we refer to private IPv4 addresses (or 'private IP addresses') as the IP addresses that are within the IPv4 CIDR range of your VPC.

When you launch an instance, we allocate a private IPv4 address for the instance using DHCP. Each instance is also given an internal DNS hostname that resolves to the private IPv4 address of the instance; for example, `ip-10-251-50-12.ec2.internal`. You can use the internal DNS hostname for communication between instances in the same network, but we can't resolve the DNS hostname outside the network that the instance is in.

An instance launched in a VPC is given a primary private IP address in the IPv4 address range of the subnet. For more information, see [Subnet Sizing](#) in the *Amazon VPC User Guide*. If you don't specify a primary private IP address when you launch the instance, we select an available IP address in the subnet's IPv4 range for you. Each instance in a VPC has a default network interface (eth0) that is assigned the primary private IPv4 address. You can also specify additional private IPv4 addresses, known as *secondary private IPv4 addresses*. Unlike primary private IP addresses, secondary private IP addresses can be reassigned from one instance to another. For more information, see [Multiple IP Addresses \(p. 702\)](#).

For instances launched in EC2-Classic, we release the private IPv4 address when the instance is stopped or terminated. If you restart your stopped instance, it receives a new private IPv4 address.

For instances launched in a VPC, a private IPv4 address remains associated with the network interface when the instance is stopped and restarted, and is released when the instance is terminated.

If you create a custom firewall configuration in EC2-Classic, you must create a rule in your firewall that allows inbound traffic from port 53 (DNS)—with a destination port from the ephemeral range—from the address of the Amazon DNS server; otherwise, internal DNS resolution from your instances fails. If your firewall doesn't automatically allow DNS query responses, then you need to allow traffic from the IP address of the Amazon DNS server. To get the IP address of the Amazon DNS server, use the following command from within your instance:

• Windows

```
ipconfig /all | findstr /c:"DNS Servers"
```

Public IPv4 Addresses and External DNS Hostnames

A public IP address is an IPv4 address that's reachable from the Internet. You can use public addresses for communication between your instances and the Internet.

Each instance that receives a public IP address is also given an external DNS hostname; for example, `ec2-203-0-113-25.compute-1.amazonaws.com`. We resolve an external DNS hostname to the public IP address of the instance outside the network of the instance, and to the private IPv4 address of the instance from within the network of the instance. The public IP address is mapped to the primary private IP address through network address translation (NAT). For more information about NAT, see [RFC 1631: The IP Network Address Translator \(NAT\)](#).

When you launch an instance in EC2-Classic, we automatically assign a public IP address to the instance from the EC2-Classic public IPv4 address pool. You cannot modify this behavior. When you launch an instance into a VPC, your subnet has an attribute that determines whether instances launched into that subnet receive a public IP address from the EC2-VPC public IPv4 address pool. By default, we assign a public IP address to instances launched in a default VPC, and we don't assign a public IP address to instances launched in a nondefault subnet.

You can control whether your instance in a VPC receives a public IP address by doing the following:

- Modifying the public IP addressing attribute of your subnet. For more information, see [Modifying the Public IPv4 Addressing Attribute for Your Subnet](#) in the *Amazon VPC User Guide*.
- Enabling or disabling the public IP addressing feature during launch, which overrides the subnet's public IP addressing attribute. For more information, see [Assigning a Public IPv4 Address During Instance Launch](#) (p. 700).

A public IP address is assigned to your instance from Amazon's pool of public IPv4 addresses, and is not associated with your AWS account. When a public IP address is disassociated from your instance, it is released back into the public IPv4 address pool, and you cannot reuse it.

You cannot manually associate or disassociate a public IP address from your instance. Instead, in certain cases, we release the public IP address from your instance, or assign it a new one:

- We release the public IP address for your instance when it's stopped or terminated. Your stopped instance receives a new public IP address when it's restarted.
- We release the public IP address for your instance when you associate an Elastic IP address with your instance, or when you associate an Elastic IP address with the primary network interface (eth0) of your instance in a VPC. When you disassociate the Elastic IP address from your instance, it receives a new public IP address.
- If the public IP address of your instance in a VPC has been released, it will not receive a new one if there is more than one network interface attached to your instance.

If you require a persistent public IP address that can be associated to and from instances as you require, use an Elastic IP address instead. For example, if you use dynamic DNS to map an existing DNS name to a new instance's public IP address, it might take up to 24 hours for the IP address to propagate through the Internet. As a result, new instances might not receive traffic while terminated instances continue to receive requests. To solve this problem, use an Elastic IP address. You can allocate your own Elastic IP address, and associate it with your instance. For more information, see [Elastic IP Addresses](#) (p. 709).

If your instance is in a VPC and you assign it an Elastic IP address, it receives an IPv4 DNS hostname if DNS hostnames are enabled. For more information, see [Using DNS with Your VPC](#) in the *Amazon VPC User Guide*.

Note

Instances that access other instances through their public NAT IP address are charged for regional or Internet data transfer, depending on whether the instances are in the same region.

Elastic IP Addresses (IPv4)

An Elastic IP address is a public IPv4 address that you can allocate to your account. You can associate it to and from instances as you require, and it's allocated to your account until you choose to release it. For more information about Elastic IP addresses and how to use them, see [Elastic IP Addresses \(p. 709\)](#).

We do not support Elastic IP addresses for IPv6.

Amazon DNS Server

Amazon provides a DNS server that resolves Amazon-provided IPv4 DNS hostnames to IPv4 addresses. In EC2-Classic, the Amazon DNS server is located at `172.16.0.23`. In EC2-VPC, the Amazon DNS server is located at the base of your VPC network range plus two. For more information, see [Amazon DNS Server](#) in the *Amazon VPC User Guide*.

IPv6 Addresses

You can optionally associate an IPv6 CIDR block with your VPC, and associate IPv6 CIDR blocks with your subnets. The IPv6 CIDR block for your VPC is automatically assigned from Amazon's pool of IPv6 addresses; you cannot choose the range yourself. For more information, see the following topics in the *Amazon VPC User Guide*:

- [VPC and Subnet Sizing for IPv6](#)
- [Associating an IPv6 CIDR Block with Your VPC](#)
- [Associating an IPv6 CIDR Block with Your Subnet](#)

IPv6 addresses are globally unique, and therefore reachable over the Internet. Your instance in a VPC receives an IPv6 address if an IPv6 CIDR block is associated with your VPC and subnet, and if one of the following is true:

- Your subnet is configured to automatically assign an IPv6 address to an instance during launch. For more information, see [Modifying the IPv6 Addressing Attribute for Your Subnet](#).
- You assign an IPv6 address to your instance during launch.
- You assign an IPv6 address to the primary network interface of your instance after launch.
- You assign an IPv6 address to a network interface in the same subnet, and attach the network interface to your instance after launch.

When your instance receives an IPv6 address during launch, the address is associated with the primary network interface (eth0) of the instance. You can disassociate the IPv6 address from the network interface. We do not support IPv6 DNS hostnames for your instance.

An IPv6 address persists when you stop and start your instance, and is released when you terminate your instance. You cannot reassign an IPv6 address while it's assigned to another network interface—you must first unassign it.

You can assign additional IPv6 addresses to your instance by assigning them to a network interface attached to your instance. The number of IPv6 addresses you can assign to a network interface and the number of network interfaces you can attach to an instance varies per instance type. For more information, see [IP Addresses Per Network Interface Per Instance Type \(p. 717\)](#).

IP Address Differences Between EC2-Classic and EC2-VPC

The following table summarizes the differences between IP addresses for instances launched in EC2-Classic, instances launched in a default subnet, and instances launched in a nondefault subnet.

Characteristic	EC2-Classic	Default Subnet	Nondefault Subnet
Public IP address (from Amazon's public IPv4 address pool)	Your instance receives a public IP address.	Your instance receives a public IP address by default, unless you specify otherwise during launch, or you modify the subnet's public IP address attribute.	Your instance doesn't receive a public IP address by default, unless you specify otherwise during launch, or you modify the subnet's public IP address attribute.
Private IPv4 address	Your instance receives a private IP address from the EC2-Classic range each time it's started.	Your instance receives a static private IP address from the IPv4 address range of your default subnet.	Your instance receives a static private IP address from the IPv4 address range of your subnet.
Multiple IPv4 addresses	We select a single private IP address for your instance; multiple IP addresses are not supported.	You can assign multiple private IP addresses to your instance.	You can assign multiple private IP addresses to your instance.
Network interfaces	IP addresses are associated with the instance; network interfaces aren't supported.	IP addresses are associated with a network interface. Each instance has one or more network interfaces.	IP addresses are associated with a network interface. Each instance has one or more network interfaces.
Elastic IP address (IPv4)	An Elastic IP address is disassociated from your instance when you stop it.	An Elastic IP address remains associated with your instance when you stop it.	An Elastic IP address remains associated with your instance when you stop it.
DNS hostnames (IPv4)	DNS hostnames are enabled by default.	DNS hostnames are enabled by default.	DNS hostnames are disabled by default, except if you've created your VPC using the VPC wizard in the Amazon VPC console.
IPv6 address	Not supported. Your instance cannot receive an IPv6 address.	Your instance does not receive an IPv6 address by default unless you've associated an IPv6 CIDR block with your VPC and subnet, and either specified an IPv6 address during launch, or modified your subnet's IPv6 addressing attribute.	Your instance does not receive an IPv6 address by default unless you've associated an IPv6 CIDR block with your VPC and subnet, and either specified an IPv6 address during launch, or modified your subnet's IPv6 addressing attribute.

Working with IP Addresses for Your Instance

You can view the IP addresses assigned to your instance, assign a public IPv4 address to your instance during launch, or assign an IPv6 address to your instance during launch.

Contents

- [Determining Your Public, Private, and Elastic IP Addresses](#) (p. 698)
- [Determining Your IPv6 Addresses](#) (p. 699)
- [Assigning a Public IPv4 Address During Instance Launch](#) (p. 700)
- [Assigning an IPv6 Address to an Instance](#) (p. 701)
- [Unassigning an IPv6 Address From an Instance](#) (p. 702)

Determining Your Public, Private, and Elastic IP Addresses

You can use the Amazon EC2 console to determine the private IPv4 addresses, public IPv4 addresses, and Elastic IP addresses of your instances. You can also determine the public IPv4 and private IPv4 addresses of your instance from within your instance by using instance metadata. For more information, see [Instance Metadata and User Data](#) (p. 271).

To determine your instance's private IPv4 addresses using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select your instance. In the details pane, get the private IPv4 address from the **Private IPs** field, and get the internal DNS hostname from the **Private DNS** field.
4. (VPC only) If you have one or more secondary private IPv4 addresses assigned to network interfaces that are attached to your instance, get those IP addresses from the **Secondary private IPs** field.
5. (VPC only) Alternatively, in the navigation pane, choose **Network Interfaces**, and then select the network interface that's associated with your instance.
6. Get the primary private IP address from the **Primary private IPv4 IP** field, and the internal DNS hostname from the **Private DNS (IPv4)** field.
7. If you've assigned secondary private IP addresses to the network interface, get those IP addresses from the **Secondary private IPv4 IPs** field.

To determine your instance's public IPv4 addresses using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select your instance. In the details pane, get the public IP address from the **IPv4 Public IP** field, and get the external DNS hostname from the **Public DNS (IPv4)** field.
4. If an Elastic IP address has been associated with the instance, get the Elastic IP address from the **Elastic IPs** field.

Note

If you've associated an Elastic IP address with your instance, the **IPv4 Public IP** field also displays the Elastic IP address.

5. (VPC only) Alternatively, in the navigation pane, choose **Network Interfaces**, and then select a network interface that's associated with your instance.
6. Get the public IP address from the **IPv4 Public IP** field. An asterisk (*) indicates the public IPv4 address or Elastic IP address that's mapped to the primary private IPv4 address.

Note

The public IPv4 address is displayed as a property of the network interface in the console, but it's mapped to the primary private IPv4 address through NAT. Therefore, if you inspect the properties of your network interface on your instance, for example, through `ifconfig` (Linux) or `ipconfig` (Windows), the public IPv4 address is not displayed. To determine your instance's public IPv4 address from within the instance, you can use instance metadata.

To determine your instance's IPv4 addresses using instance metadata

1. Connect to your instance.
2. Use the following command to access the private IP address:

- **Linux**

```
C:\> curl http://169.254.169.254/latest/meta-data/local-ipv4
```

- **Windows**

```
C:\> wget http://169.254.169.254/latest/meta-data/local-ipv4
```

3. Use the following command to access the public IP address:

- **Linux**

```
C:\> curl http://169.254.169.254/latest/meta-data/public-ipv4
```

- **Windows**

```
C:\> wget http://169.254.169.254/latest/meta-data/public-ipv4
```

Note that if an Elastic IP address is associated with the instance, the value returned is that of the Elastic IP address.

Determining Your IPv6 Addresses

(VPC only) You can use the Amazon EC2 console to determine the IPv6 addresses of your instances.

To determine your instance's IPv6 addresses using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select your instance. In the details pane, get the IPv6 addresses from the **IPv6 IPs** field.

To determine your instance's IPv6 addresses using instance metadata

1. Connect to your instance.
2. Use the following command to view the IPv6 address (you can get the MAC address from `http://169.254.169.254/latest/meta-data/network/interfaces/macs/`):

- **Linux**

```
C:\> curl http://169.254.169.254/latest/meta-data/network/interfaces/macs/mac-address/ipv6s
```

- **Windows**

```
C:\> wget http://169.254.169.254/latest/meta-data/network/interfaces/macs/mac-address/ipv6s
```

Assigning a Public IPv4 Address During Instance Launch

If you launch an instance in EC2-Classic, it is assigned a public IPv4 address by default. You can't modify this behavior.

In a VPC, all subnets have an attribute that determines whether instances launched into that subnet are assigned a public IP address. By default, nondefault subnets have this attribute set to false, and default subnets have this attribute set to true. When you launch an instance, a public IPv4 addressing feature is also available for you to control whether your instance is assigned a public IPv4 address; you can override the default behavior of the subnet's IP addressing attribute. The public IPv4 address is assigned from Amazon's pool of public IPv4 addresses, and is assigned to the network interface with the device index of eth0. This feature depends on certain conditions at the time you launch your instance.

Important

You can't manually disassociate the public IP address from your instance after launch. Instead, it's automatically released in certain cases, after which you cannot reuse it. For more information, see [Public IPv4 Addresses and External DNS Hostnames \(p. 695\)](#). If you require a persistent public IP address that you can associate or disassociate at will, assign an Elastic IP address to the instance after launch instead. For more information, see [Elastic IP Addresses \(p. 709\)](#).

To access the public IP addressing feature when launching an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Launch Instance**.
3. Select an AMI and an instance type, and then choose **Next: Configure Instance Details**.
4. On the **Configure Instance Details** page, for **Network**, select a VPC . The **Auto-assign Public IP** list is displayed. Choose **Enable** or **Disable** to override the default setting for the subnet.

Important

You cannot auto-assign a public IP address if you specify more than one network interface. Additionally, you cannot override the subnet setting using the auto-assign public IP feature if you specify an existing network interface for eth0.

5. Follow the steps on the next pages of the wizard to complete your instance's setup. For more information about the wizard configuration options, see [Launching an Instance \(p. 244\)](#). On the final **Review Instance Launch** page, review your settings, and then choose **Launch** to choose a key pair and launch your instance.
6. On the **Instances** page, select your new instance and view its public IP address in **IPv4 Public IP** field in the details pane.

The public IP addressing feature is only available during launch. However, whether you assign a public IP address to your instance during launch or not, you can associate an Elastic IP address with your instance after it's launched. For more information, see [Elastic IP Addresses \(p. 709\)](#). You can also modify your subnet's public IPv4 addressing behavior. For more information, see [Modifying the Public IPv4 Addressing Attribute for Your Subnet](#).

To enable or disable the public IP addressing feature using the command line

- You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- Use the `--associate-public-ip-address` or the `--no-associate-public-ip-address` option with the `run-instances` command (AWS CLI)
- Use the `-AssociatePublicIp` parameter with the `New-EC2Instance` command (AWS Tools for Windows PowerShell)

Assigning an IPv6 Address to an Instance

If your VPC and subnet have IPv6 CIDR blocks associated with them, you can assign an IPv6 address to your instance during or after launch. The IPv6 address is assigned from the IPv6 address range of the subnet, and is assigned to the network interface with the device index of eth0.

To assign an IPv6 address to an instance during launch

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Select an AMI, an instance type, and choose **Next: Configure Instance Details**.

Note

Ensure that you select an instance type that supports IPv6 addresses. For more information, see [Instance Types \(p. 117\)](#).

3. On the **Configure Instance Details** page, for **Network**, select a VPC and for **Subnet**, select a subnet. For **Auto-assign IPv6 IP**, choose **Enable**.
4. Follow the remaining steps in the wizard to launch your instance.

Alternatively, you can assign an IPv6 address to your instance after launch.

To assign an IPv6 address to your instance after launch

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select your instance, choose **Actions, Manage IP Addresses**.
4. Under **IPv6 Addresses**, choose **Assign new IP**. You can specify an IPv6 address from the range of the subnet, or leave the **Auto-assign** value to let Amazon choose an IPv6 address for you.
5. Choose **Save**.

Note

If you launched your instance using Amazon Linux 2016.09.0 or later, or Windows Server 2008 R2 or later, your instance is configured for IPv6, and no additional steps are needed to ensure that the IPv6 address is recognized on the instance. If you launched your instance from an older AMI, you may have to configure your instance manually. For more information, see [Configure IPv6 on Your Instances](#) in the *Amazon VPC User Guide*.

To assign an IPv6 address using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- Use the `--ipv6-addresses` option with the `run-instances` command (AWS CLI)
- Use the `Ipv6Addresses` property for `-NetworkInterface` in the `New-EC2Instance` command (AWS Tools for Windows PowerShell)
- `assign-ipv6-addresses` (AWS CLI)
- `Register-EC2Ipv6AddressList` (AWS Tools for Windows PowerShell)

Unassigning an IPv6 Address From an Instance

You can unassign an IPv6 address from an instance using the Amazon EC2 console.

To unassign an IPv6 address from an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select your instance, choose **Actions, Manage IP Addresses**.
4. Under **IPv6 Addresses**, choose **Unassign** for the IPv6 address to unassign.
5. Choose **Yes, Update**.

To unassign an IPv6 address using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [unassign-ipv6-addresses](#) (AWS CLI)
- [Unregister-EC2Ipv6AddressList](#) (AWS Tools for Windows PowerShell).

Multiple IP Addresses

In EC2-VPC, you can specify multiple private IPv4 and IPv6 addresses for your instances. The number of network interfaces and private IPv4 and IPv6 addresses that you can specify for an instance depends on the instance type. For more information, see [IP Addresses Per Network Interface Per Instance Type \(p. 717\)](#).

It can be useful to assign multiple IP addresses to an instance in your VPC to do the following:

- Host multiple websites on a single server by using multiple SSL certificates on a single server and associating each certificate with a specific IP address.
- Operate network appliances, such as firewalls or load balancers, that have multiple IP addresses for each network interface.
- Redirect internal traffic to a standby instance in case your instance fails, by reassigning the secondary IP address to the standby instance.

Contents

- [How Multiple IP Addresses Work \(p. 702\)](#)
- [Working with Multiple IPv4 Addresses \(p. 703\)](#)
- [Working with Multiple IPv6 Addresses \(p. 707\)](#)

How Multiple IP Addresses Work

The following list explains how multiple IP addresses work with network interfaces:

- You can assign a secondary private IPv4 address to any network interface. The network interface can be attached to or detached from the instance.
- You can assign multiple IPv6 addresses to a network interface that's in a subnet that has an associated IPv6 CIDR block.
- You must choose the secondary IPv4 from the IPv4 CIDR block range of the subnet for the network interface.

- You must choose IPv6 addresses from the IPv6 CIDR block range of the subnet for the network interface.
- Security groups apply to network interfaces, not to IP addresses. Therefore, IP addresses are subject to the security group of the network interface in which they're specified.
- Multiple IP addresses can be assigned and unassigned to network interfaces attached to running or stopped instances.
- Secondary private IPv4 addresses that are assigned to a network interface can be reassigned to another one if you explicitly allow it.
- An IPv6 address cannot be reassigned to another network interface; you must first unassign the IPv6 address from the existing network interface.
- When assigning multiple IP addresses to a network interface using the command line tools or API, the entire operation fails if one of the IP addresses can't be assigned.
- Primary private IPv4 addresses, secondary private IPv4 addresses, Elastic IP addresses, and IPv6 addresses remain with the network interface when it is detached from an instance or attached to another instance.
- Although you can't move the primary network interface from an instance, you can reassign the secondary private IPv4 address of the primary network interface to another network interface.
- You can move any additional network interface from one instance to another.

The following list explains how multiple IP addresses work with Elastic IP addresses (IPv4 only):

- Each private IPv4 address can be associated with a single Elastic IP address, and vice versa.
- When a secondary private IPv4 address is reassigned to another interface, the secondary private IPv4 address retains its association with an Elastic IP address.
- When a secondary private IPv4 address is unassigned from an interface, an associated Elastic IP address is automatically disassociated from the secondary private IPv4 address.

Working with Multiple IPv4 Addresses

You can assign a secondary private IPv4 address to an instance, associate an Elastic IPv4 address with a secondary private IPv4 address, and unassign a secondary private IPv4 address.

Contents

- [Assigning a Secondary Private IPv4 Address \(p. 703\)](#)
- [Configuring the Operating System on Your Instance to Recognize the Secondary Private IPv4 Address \(p. 705\)](#)
- [Associating an Elastic IP Address with the Secondary Private IPv4 Address \(p. 705\)](#)
- [Viewing Your Secondary Private IPv4 Addresses \(p. 706\)](#)
- [Unassigning a Secondary Private IPv4 Address \(p. 706\)](#)

Assigning a Secondary Private IPv4 Address

You can assign the secondary private IPv4 address to the network interface for an instance as you launch the instance, or after the instance is running. This section includes the following procedures.

- [To assign a secondary private IPv4 address when launching an instance in EC2-VPC \(p. 704\)](#)
- [To assign a secondary IPv4 address during launch using the command line \(p. 704\)](#)
- [To assign a secondary private IPv4 address to a network interface \(p. 704\)](#)
- [To assign a secondary private IPv4 to an existing instance using the command line \(p. 705\)](#)

To assign a secondary private IPv4 address when launching an instance in EC2-VPC

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Launch Instance**.
3. Select an AMI, then choose an instance type and choose **Next: Configure Instance Details**.
4. In the **Configure Instance Details** page, for **Network**, select a VPC and for **Subnet**, select a subnet.
5. In the **Network Interfaces** section, do the following, and then choose **Next: Add Storage**:
 - To add another network interface, choose **Add Device**. The console enables you to specify up to two network interfaces when you launch an instance. After you launch the instance, choose **Network Interfaces** in the navigation pane to add additional network interfaces. The total number of network interfaces that you can attach varies by instance type. For more information, see [IP Addresses Per Network Interface Per Instance Type \(p. 717\)](#).

Important

When you add a second network interface, the system can no longer auto-assign a public IPv4 address. You will not be able to connect to the instance over IPv4 unless you assign an Elastic IP address to the primary network interface (eth0). You can assign the Elastic IP address after you complete the Launch wizard. For more information, see [Working with Elastic IP Addresses \(p. 712\)](#).

- For each network interface, under **Secondary IP addresses**, choose **Add IP**, and then enter a private IP address from the subnet range, or accept the default `Auto-assign` value to let Amazon select an address.
6. On the next **Add Storage** page, you can specify volumes to attach to the instance besides the volumes specified by the AMI (such as the root device volume), and then choose **Next: Tag Instance**.
 7. On the **Tag Instance** page, specify tags for the instance, such as a user-friendly name, and then choose **Next: Configure Security Group**.
 8. On the **Configure Security Group** page, select an existing security group or create a new one. Choose **Review and Launch**.
 9. On the **Review Instance Launch** page, review your settings, and then choose **Launch** to choose a key pair and launch your instance. If you're new to Amazon EC2 and haven't created any key pairs, the wizard prompts you to create one.

Important

After you have added a secondary private IP address to a network interface, you must connect to the instance and configure the secondary private IP address on the instance itself. For more information, see [Configuring the Operating System on Your Instance to Recognize the Secondary Private IPv4 Address \(p. 705\)](#).

To assign a secondary IPv4 address during launch using the command line

- You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).
 - The `--secondary-private-ip-addresses` option with the `run-instances` command (AWS CLI)
 - Define `-NetworkInterface` and specify the `PrivateIpAddresses` parameter with the `New-EC2Instance` command (AWS Tools for Windows PowerShell).

To assign a secondary private IPv4 address to a network interface

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

2. In the navigation pane, choose **Network Interfaces**, and then select the network interface attached to the instance.
3. Choose **Actions, Manage IP Addresses**.
4. Under **IPv4 Addresses**, choose **Assign new IP**.
5. Enter a specific IPv4 address that's within the subnet range for the instance, or leave the field blank to let Amazon select an IP address for you.
6. (Optional) Choose **Allow reassignment** to allow the secondary private IP address to be reassigned if it is already assigned to another network interface.
7. Choose **Yes, Update**.

Alternatively, you can assign a secondary private IPv4 address to an instance. Choose **Instances** in the navigation pane, select the instance, and then choose **Actions, Networking, Manage IP Addresses**. You can configure the same information as you did in the steps above. The IP address is assigned to the primary network interface (eth0) for the instance.

To assign a secondary private IPv4 to an existing instance using the command line

- You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).
 - [assign-private-ip-addresses](#) (AWS CLI)
 - [Register-EC2PrivateIpAddress](#) (AWS Tools for Windows PowerShell)

Configuring the Operating System on Your Instance to Recognize the Secondary Private IPv4 Address

After you assign a secondary private IPv4 address to your instance, you need to configure the operating system on your instance to recognize the secondary private IP address.

For information about configuring a Windows instance, see [Configuring a Secondary Private IPv4 Address for Your Windows Instance in a VPC \(p. 378\)](#).

Associating an Elastic IP Address with the Secondary Private IPv4 Address

To associate an Elastic IP address with a secondary private IPv4 address in EC2-VPC

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Choose **Actions, Associate Address**.
4. In the **Associate Address** dialog box, for **Network Interface**, select the network interface and for **Private IP address**, select the secondary IP address.
5. Choose **Associate**.

To associate an Elastic IP address with a secondary private IPv4 address using the command line

- You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).
 - [associate-address](#) (AWS CLI)
 - [Register-EC2Address](#) (AWS Tools for Windows PowerShell)

Viewing Your Secondary Private IPv4 Addresses

To view the private IPv4 addresses assigned to a network interface in EC2-VPC

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the network interface with private IP addresses to view.
4. On the **Details** tab in the details pane, check the **Primary private IPv4 IP** and **Secondary private IPv4 IPs** fields for the primary private IPv4 address and any secondary private IPv4 addresses assigned to the network interface.

To view the private IPv4 addresses assigned to an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance with private IPv4 addresses to view.
4. On the **Description** tab in the details pane, check the **Private IPs** and **Secondary private IPs** fields for the primary private IPv4 address and any secondary private IPv4 addresses assigned to the instance through its network interface.

Unassigning a Secondary Private IPv4 Address

If you no longer require a secondary private IPv4 address, you can unassign it from the instance or the network interface. When a secondary private IPv4 address is unassigned from a network interface, the Elastic IP address (if it exists) is also disassociated.

To unassign a secondary private IPv4 address from an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select an instance, choose **Actions, Networking, Manage IP Addresses**.
4. Under **IPv4 Addresses**, choose **Unassign** for the IPv4 address to unassign.
5. Choose **Yes, Update**.

To unassign a secondary private IPv4 address from a network interface

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the network interface, choose **Actions, Manage IP Addresses**.
4. Under **IPv4 Addresses**, choose **Unassign** for the IPv4 address to unassign.
5. Choose **Yes, Update**.

To unassign a secondary private IPv4 address using the command line

- You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).
 - [unassign-private-ip-addresses](#) (AWS CLI)
 - [Unregister-EC2PrivateIpAddress](#) (AWS Tools for Windows PowerShell)

Working with Multiple IPv6 Addresses

You can assign multiple IPv6 addresses to your instance, view the IPv6 addresses assigned to your instance, and unassign IPv6 addresses from your instance.

Contents

- [Assigning Multiple IPv6 Addresses \(p. 707\)](#)
- [Viewing Your IPv6 Addresses \(p. 708\)](#)
- [Unassigning an IPv6 Address \(p. 709\)](#)

Assigning Multiple IPv6 Addresses

You can assign one or more IPv6 addresses to your instance during launch or after launch. To assign an IPv6 address to an instance, the VPC and subnet in which you launch the instance must have an associated IPv6 CIDR block. For more information, see [VPCs and Subnets](#) in the *Amazon VPC User Guide*.

To assign multiple IPv6 addresses during launch

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the dashboard, choose **Launch Instance**.
3. Select an AMI, choose an instance type, and choose **Next: Configure Instance Details**. Ensure that you choose an instance type that support IPv6. For more information, see [Instance Types \(p. 117\)](#).
4. On the **Configure Instance Details** page, select a VPC from the **Network** list, and a subnet from the **Subnet** list.
5. In the **Network Interfaces** section, do the following, and then choose **Next: Add Storage**:
 - To assign a single IPv6 address to the primary network interface (eth0), under **IPv6 IPs**, choose **Add IP**. To add a secondary IPv6 address, choose **Add IP** again. You can enter an IPv6 address from the range of the subnet, or leave the default **Auto-assign** value to let Amazon choose an IPv6 address from the subnet for you.
 - Choose **Add Device** to add another network interface and repeat the steps above to add one or more IPv6 addresses to the network interface. The console enables you to specify up to two network interfaces when you launch an instance. After you launch the instance, choose **Network Interfaces** in the navigation pane to add additional network interfaces. The total number of network interfaces that you can attach varies by instance type. For more information, see [IP Addresses Per Network Interface Per Instance Type \(p. 717\)](#).
6. Follow the next steps in the wizard to attach volumes and tag your instance.
7. On the **Configure Security Group** page, select an existing security group or create a new one. If you want your instance to be reachable over IPv6, ensure that your security group has rules that allow access from IPv6 addresses. For more information, see [Security Group Rules Reference \(p. 613\)](#). Choose **Review and Launch**.
8. On the **Review Instance Launch** page, review your settings, and then choose **Launch** to choose a key pair and launch your instance. If you're new to Amazon EC2 and haven't created any key pairs, the wizard prompts you to create one.

You can use the **Instances** screen Amazon EC2 console to assign multiple IPv6 addresses to an existing instance. This assigns the IPv6 addresses to the primary network interface (eth0) for the instance. To assign a specific IPv6 address to the instance, ensure that the IPv6 address is not already assigned to another instance or network interface.

To assign multiple IPv6 addresses to an existing instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select your instance, choose **Actions, Manage IP Addresses**.
4. Under **IPv6 Addresses**, choose **Assign new IP** for each IPv6 address you want to add. You can specify an IPv6 address from the range of the subnet, or leave the **Auto-assign** value to let Amazon choose an IPv6 address for you.
5. Choose **Yes, Update**.

Alternatively, you can assign multiple IPv6 addresses to an existing network interface. The network interface must have been created in a subnet that has an associated IPv6 CIDR block. To assign a specific IPv6 address to the network interface, ensure that the IPv6 address is not already assigned to another network interface.

To assign multiple IPv6 addresses to a network interface

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select your network interface, choose **Actions, Manage IP Addresses**.
4. Under **IPv6 Addresses**, choose **Assign new IP** for each IPv6 address you want to add. You can specify an IPv6 address from the range of the subnet, or leave the **Auto-assign** value to let Amazon choose an IPv6 address for you.
5. Choose **Yes, Update**.

CLI Overview

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- **Assign an IPv6 address during launch:**
 - Use the `--ipv6-addresses` or `--ipv6-address-count` options with the [run-instances](#) command (AWS CLI)
 - Define `-NetworkInterface` and specify the `Ipv6Addresses` or `Ipv6AddressCount` parameters with the [New-EC2Instance](#) command (AWS Tools for Windows PowerShell).
- **Assign an IPv6 address to a network interface:**
 - [assign-ipv6-addresses](#) (AWS CLI)
 - [Register-EC2Ipv6AddressList](#) (AWS Tools for Windows PowerShell)

Viewing Your IPv6 Addresses

You can view the IPv6 addresses for an instance or for a network interface.

To view the IPv6 addresses assigned to an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select your instance. In the details pane, review the **IPv6 IPs** field.

To view the IPv6 addresses assigned to a network interface

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

2. In the navigation pane, choose **Network Interfaces**.
3. Select your network interface. In the details pane, review the **IPv6 IPs** field.

CLI Overview

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- **View the IPv6 addresses for an instance:**
 - [describe-instances](#) (AWS CLI)
 - [Get-EC2Instance](#) (AWS Tools for Windows PowerShell).
- **View the IPv6 addresses for a network interface:**
 - [describe-network-interfaces](#) (AWS CLI)
 - [Get-EC2NetworkInterface](#) (AWS Tools for Windows PowerShell)

Unassigning an IPv6 Address

You can unassign an IPv6 address from the primary network interface of an instance, or you can unassign an IPv6 address from a network interface.

To unassign an IPv6 address from an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select your instance, choose **Actions, Manage IP Addresses**.
4. Under **IPv6 Addresses**, choose **Unassign** for the IPv6 address to unassign.
5. Choose **Yes, Update**.

To unassign an IPv6 address from a network interface

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select your network interface, choose **Actions, Manage IP Addresses**.
4. Under **IPv6 Addresses**, choose **Unassign** for the IPv6 address to unassign.
5. Choose **Save**.

CLI Overview

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [unassign-ipv6-addresses](#) (AWS CLI)
- [Unregister-EC2Ipv6AddressList](#) (AWS Tools for Windows PowerShell).

Elastic IP Addresses

An *Elastic IP address* is a static IPv4 address designed for dynamic cloud computing. An Elastic IP address is associated with your AWS account. With an Elastic IP address, you can mask the failure of an instance or software by rapidly remapping the address to another instance in your account.

An Elastic IP address is a public IPv4 address, which is reachable from the Internet. If your instance does not have a public IPv4 address, you can associate an Elastic IP address with your instance to enable communication with the Internet; for example, to connect to your instance from your local computer.

We currently do not support Elastic IP addresses for IPv6.

Topics

- [Elastic IP Address Basics \(p. 710\)](#)
- [Elastic IP Address Differences for EC2-Classic and EC2-VPC \(p. 710\)](#)
- [Working with Elastic IP Addresses \(p. 712\)](#)
- [Using Reverse DNS for Email Applications \(p. 716\)](#)
- [Elastic IP Address Limit \(p. 716\)](#)

Elastic IP Address Basics

The following are the basic characteristics of an Elastic IP address:

- To use an Elastic IP address, you first allocate one to your account, and then associate it with your instance or a network interface.
- When you associate an Elastic IP address with an instance or its primary network interface, the instance's public IPv4 address (if it had one) is released back into Amazon's pool of public IPv4 addresses. You cannot reuse a public IPv4 address. For more information, see [Public IPv4 Addresses and External DNS Hostnames \(p. 695\)](#).
- You can disassociate an Elastic IP address from a resource, and reassociate it with a different resource.
- A disassociated Elastic IP address remains allocated to your account until you explicitly release it.
- To ensure efficient use of Elastic IP addresses, we impose a small hourly charge if an Elastic IP address is not associated with a running instance, or if it is associated with a stopped instance or an unattached network interface. While your instance is running, you are not charged for one Elastic IP address associated with the instance, but you are charged for any additional Elastic IP addresses associated with the instance. For more information, see [Amazon EC2 Pricing](#).
- An Elastic IP address is for use in a specific region only.
- When you associate an Elastic IP address with an instance that previously had a public IPv4 address, the public DNS hostname of the instance changes to match the Elastic IP address.
- We resolve a public DNS hostname to the public IPv4 address or the Elastic IP address of the instance outside the network of the instance, and to the private IPv4 address of the instance from within the network of the instance.

If your account supports EC2-Classic, the use and behavior of Elastic IP addresses for EC2-Classic and EC2-VPC may differ. For more information, see [Elastic IP Address Differences for EC2-Classic and EC2-VPC \(p. 710\)](#).

Elastic IP Address Differences for EC2-Classic and EC2-VPC

If your account supports EC2-Classic, there's one pool of Elastic IP addresses for use with the EC2-Classic platform and another for use with the EC2-VPC platform. You can't associate an Elastic IP address that you allocated for use with a VPC with an instance in EC2-Classic, and vice-versa. However, you can migrate an Elastic IP address you've allocated for use in the EC2-Classic platform

Amazon Elastic Compute Cloud
User Guide for Windows Instances
Elastic IP Address Differences
for EC2-Classic and EC2-VPC

to the EC2-VPC platform. You cannot migrate an Elastic IP address to another region. For more information about EC2-Classic and EC2-VPC, see [Supported Platforms \(p. 672\)](#).

When you associate an Elastic IP address with an instance in EC2-Classic, a default VPC, or an instance in a nondefault VPC in which you assigned a public IPv4 to the eth0 network interface during launch, the instance's current public IPv4 address is released back into the public IP address pool. If you disassociate an Elastic IP address from the instance, the instance is automatically assigned a new public IPv4 address within a few minutes. However, if you have attached a second network interface to an instance in a VPC, the instance is not automatically assigned a new public IPv4 address. For more information about public IPv4 addresses, see [Public IPv4 Addresses and External DNS Hostnames \(p. 695\)](#).

For information about using an Elastic IP address with an instance in a VPC, see [Elastic IP Addresses](#) in the *Amazon VPC User Guide*.

The following table lists the differences between Elastic IP addresses on EC2-Classic and EC2-VPC. For more information about the differences between private and public IP addresses, see [IP Address Differences Between EC2-Classic and EC2-VPC \(p. 697\)](#).

Characteristic	EC2-Classic	EC2-VPC
Allocating an Elastic IP address	When you allocate an Elastic IP address, it's for use in EC2-Classic; however, you can migrate an Elastic IP address to the EC2-VPC platform. For more information, see Migrating an Elastic IP Address from EC2-Classic to EC2-VPC (p. 712) .	When you allocate an Elastic IP address, it's for use only in a VPC.
Associating an Elastic IP address	You associate an Elastic IP address with an instance.	An Elastic IP address is a property of a network interface. You can associate an Elastic IP address with an instance by updating the network interface attached to the instance. For more information, see Elastic Network Interfaces (p. 716) .
Reassociating an Elastic IP address	If you try to associate an Elastic IP address that's already associated with another instance, the address is automatically associated with the new instance.	If your account supports EC2-VPC only, and you try to associate an Elastic IP address that's already associated with another instance, the address is automatically associated with the new instance. If you're using a VPC in an EC2-Classic account, and you try to associate an Elastic IP address that's already associated with another instance, it succeeds only if you allowed reassociation.
Stopping an instance	If you stop an instance, its Elastic IP address is disassociated, and you must reassociate the Elastic IP address when you restart the instance.	If you stop an instance, its Elastic IP address remains associated.
Assigning multiple IP addresses	Instances support only a single private IPv4 address and a corresponding Elastic IP address.	Instances support multiple IPv4 addresses, and each one can have a corresponding Elastic IP address. For more information, see Multiple IP Addresses (p. 702) .

Migrating an Elastic IP Address from EC2-Classic to EC2-VPC

If your account supports EC2-Classic, you can migrate Elastic IP addresses that you've allocated for use in the EC2-Classic platform to the EC2-VPC platform, within the same region. This can assist you to migrate your resources from EC2-Classic to a VPC; for example, you can launch new web servers in your VPC, and then use the same Elastic IP addresses that you used for your web servers in EC2-Classic for your new VPC web servers.

After you've migrated an Elastic IP address to EC2-VPC, you cannot use it in the EC2-Classic platform; however, if required, you can restore it to EC2-Classic. After you've restored an Elastic IP address to EC2-Classic, you cannot use it in EC2-VPC until you migrate it again. You can only migrate an Elastic IP address from EC2-Classic to EC2-VPC. You cannot migrate an Elastic IP address that was originally allocated for use in EC2-VPC to EC2-Classic.

To migrate an Elastic IP address, it must not be associated with an instance. For more information about disassociating an Elastic IP address from an instance, see [Disassociating an Elastic IP Address and Reassociating it with a Different Instance](#) (p. 714).

You can migrate as many EC2-Classic Elastic IP addresses as you can have in your account. However, when you migrate an Elastic IP address to EC2-VPC, it counts against your Elastic IP address limit for EC2-VPC. You cannot migrate an Elastic IP address if it will result in you exceeding your limit. Similarly, when you restore an Elastic IP address to EC2-Classic, it counts against your Elastic IP address limit for EC2-Classic. For more information, see [Elastic IP Address Limit](#) (p. 716).

You cannot migrate an Elastic IP address that has been allocated to your account for less than 24 hours.

For more information, see [Moving an Elastic IP Address](#) (p. 714).

Working with Elastic IP Addresses

The following sections describe how you can work with Elastic IP addresses.

Topics

- [Allocating an Elastic IP Address](#) (p. 712)
- [Describing Your Elastic IP Addresses](#) (p. 713)
- [Associating an Elastic IP Address with a Running Instance](#) (p. 713)
- [Disassociating an Elastic IP Address and Reassociating it with a Different Instance](#) (p. 714)
- [Moving an Elastic IP Address](#) (p. 714)
- [Releasing an Elastic IP Address](#) (p. 715)

Allocating an Elastic IP Address

You can allocate an Elastic IP address using the Amazon EC2 console or the command line. If your account supports EC2-Classic, you can allocate an address for use in EC2-Classic or in EC2-VPC.

To allocate an Elastic IP address for use in EC2-VPC using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Choose **Allocate New Address**.
4. (EC2-Classic accounts) In the **Allocate New Address** dialog box, select **VPC** from **EIP used in**, and then choose **Yes, Allocate**. Close the confirmation dialog box.

5. (VPC-only accounts) Choose **Yes, Allocate**, and close the confirmation dialog box.

To allocate an Elastic IP address for use in EC2-Classical using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Choose **Allocate New Address**.
4. Select **EC2**, and then choose **Yes, Allocate**. Close the confirmation dialog box.

To allocate an Elastic IP address using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [allocate-address](#) (AWS CLI)
- [New-EC2Address](#) (AWS Tools for Windows PowerShell)

Describing Your Elastic IP Addresses

You can describe an Elastic IP address using the Amazon EC2 or the command line.

To describe your Elastic IP addresses using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Select a filter from the Resource Attribute list to begin searching. You can use multiple filters in a single search.

To describe your Elastic IP addresses using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [describe-addresses](#) (AWS CLI)
- [Get-EC2Address](#) (AWS Tools for Windows PowerShell)

Associating an Elastic IP Address with a Running Instance

You can associate an Elastic IP address to an instance using the Amazon EC2 console or the command line.

(VPC only) If you're associating an Elastic IP address with your instance to enable communication with the Internet, you must also ensure that your instance is in a public subnet. For more information, see [Internet Gateways](#) in the *Amazon VPC User Guide*.

To associate an Elastic IP address with an instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Select an Elastic IP address, choose **Actions**, and then select **Associate Address**.
4. In the **Associate Address** dialog box, select the instance from **Instance** and then choose **Associate**.

To associate an Elastic IP address using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [associate-address](#) (AWS CLI)
- [Register-EC2Address](#) (AWS Tools for Windows PowerShell)

Disassociating an Elastic IP Address and Reassociating it with a Different Instance

You can disassociate an Elastic IP address and then reassociate it using the Amazon EC2 console or the command line.

To disassociate and reassociate an Elastic IP address using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Select the Elastic IP address, choose **Actions**, and then select **Disassociate Address**.
4. Choose **Yes, Disassociate** when prompted for confirmation.
5. Select the address that you disassociated in the previous step. For **Actions**, choose **Associate Address**.
6. In the **Associate Address** dialog box, select the new instance from **Instance**, and then choose **Associate**.

To disassociate an Elastic IP address using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [disassociate-address](#) (AWS CLI)
- [Unregister-EC2Address](#) (AWS Tools for Windows PowerShell)

To associate an Elastic IP address using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [associate-address](#) (AWS CLI)
- [Register-EC2Address](#) (AWS Tools for Windows PowerShell)

Moving an Elastic IP Address

Currently, you can migrate an Elastic IP address to EC2-VPC or restore it to EC2-Classical using the Amazon EC2 Query API, an AWS SDK, or the AWS CLI only.

After you've performed the command to move or restore your Elastic IP address, the process of migrating the Elastic IP address can take a few minutes. Use the [describe-moving-addresses](#) command to check whether your Elastic IP address is still moving, or has completed moving.

If the Elastic IP address is in a moving state for longer than 5 minutes, contact <http://aws.amazon.com/premiumsupport/>.

To move an Elastic IP address using the Amazon EC2 Query API or AWS CLI

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [move-address-to-vpc](#) (AWS CLI)
- [MoveAddressToVpc](#) (Amazon EC2 Query API)
- [Move-EC2AddressToVpc](#) (AWS Tools for Windows PowerShell)

To restore an Elastic IP address to EC2-Classic using the Amazon EC2 Query API or AWS CLI

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [restore-address-to-classic](#) (AWS CLI)
- [RestoreAddressToClassic](#) (Amazon EC2 Query API)
- [Restore-EC2AddressToClassic](#) (AWS Tools for Windows PowerShell)

To describe the status of your moving addresses using the Amazon EC2 Query API or AWS CLI

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [describe-moving-addresses](#) (AWS CLI)
- [DescribeMovingAddresses](#) (Amazon EC2 Query API)
- [Get-EC2Address](#) (AWS Tools for Windows PowerShell)

To retrieve the allocation ID for your migrated Elastic IP address in EC2-VPC

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [describe-addresses](#) (AWS CLI)
- [DescribeAddresses](#) (Amazon EC2 Query API)
- [Get-EC2Address](#) (AWS Tools for Windows PowerShell)

Releasing an Elastic IP Address

If you no longer need an Elastic IP address, we recommend that you release it (the address must not be associated with an instance). You incur charges for any Elastic IP address that's allocated for use with EC2-Classic but not associated with an instance.

You can release an Elastic IP address using the Amazon EC2 console or the command line.

To release an Elastic IP address using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Select the Elastic IP address, choose **Actions**, and then select **Release Addresses**. Choose **Yes, Release** when prompted.

To release an Elastic IP address using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [release-address](#) (AWS CLI)
- [Remove-EC2Address](#) (AWS Tools for Windows PowerShell)

Using Reverse DNS for Email Applications

If you intend to send email to third parties from an instance, we suggest you provision one or more Elastic IP addresses and provide them to us. AWS works with ISPs and Internet anti-spam organizations to reduce the chance that your email sent from these addresses will be flagged as spam.

In addition, assigning a static reverse DNS record to your Elastic IP address used to send email can help avoid having email flagged as spam by some anti-spam organizations. Note that a corresponding forward DNS record (record type A) pointing to your Elastic IP address must exist before we can create your reverse DNS record.

If a reverse DNS record is associated with an Elastic IP address, the Elastic IP address is locked to your account and cannot be released from your account until the record is removed.

To remove email sending limits, or to provide us with your Elastic IP addresses and reverse DNS records, go to the [Request to Remove Email Sending Limitations](#) page.

Elastic IP Address Limit

By default, all AWS accounts are limited to 5 Elastic IP addresses per region, because public (IPv4) Internet addresses are a scarce public resource. We strongly encourage you to use an Elastic IP address primarily for the ability to remap the address to another instance in the case of instance failure, and to use DNS hostnames for all other inter-node communication.

If you feel your architecture warrants additional Elastic IP addresses, please complete the [Amazon EC2 Elastic IP Address Request Form](#). We will ask you to describe your use case so that we can understand your need for additional addresses.

Elastic Network Interfaces

An elastic network interface (referred to as a *network interface* in this documentation) is a virtual network interface that you can attach to an instance in a VPC. Network interfaces are available only for instances running in a VPC.

A network interface can include the following attributes:

- A primary private IPv4 address
- One or more secondary private IPv4 addresses
- One Elastic IP address (IPv4) per private IPv4 address
- One public IPv4 address
- One or more IPv6 addresses
- One or more security groups
- A MAC address
- A source/destination check flag
- A description

You can create a network interface, attach it to an instance, detach it from an instance, and attach it to another instance. The attributes of a network interface follow it as it's attached or detached from an instance and reattached to another instance. When you move a network interface from one instance to another, network traffic is redirected to the new instance.

Every instance in a VPC has a default network interface, called the *primary network interface* (eth0). You cannot detach a primary network interface from an instance. You can create and attach additional network interfaces. The maximum number of network interfaces that you can use varies by instance type. For more information, see [IP Addresses Per Network Interface Per Instance Type](#) (p. 717).

Private IPv4 addresses for network interfaces

The primary network interface for an instance is assigned a primary private IPv4 address from the IPv4 address range of your VPC. You can assign additional private IPv4 addresses to a network interface.

Public IPv4 addresses for network interfaces

In a VPC, all subnets have a modifiable attribute that determines whether network interfaces created in that subnet (and therefore instances launched into that subnet) are assigned a public IPv4 address. For more information, see [IP Addressing Behavior for Your Subnet](#) in the *Amazon VPC User Guide*. The public IPv4 address is assigned from Amazon's pool of public IPv4 addresses. When you launch an instance, the IP address is assigned to the primary network interface (eth0) that's created.

When you create a network interface, it inherits the public IPv4 addressing attribute from the subnet. If you later modify the public IPv4 addressing attribute of the subnet, the network interface keeps the setting that was in effect when it was created. If you launch an instance and specify an existing network interface for eth0, the public IPv4 addressing attribute is determined by the network interface.

For more information, see [Public IPv4 Addresses and External DNS Hostnames](#) (p. 695).

IPv6 addresses for network interfaces

You can associate an IPv6 CIDR block with your VPC and subnet, and assign one or more IPv6 addresses from the subnet range to a network interface.

All subnets have a modifiable attribute that determines whether network interfaces created in that subnet (and therefore instances launched into that subnet) are automatically assigned an IPv6 address from the range of the subnet. For more information, see [IP Addressing Behavior for Your Subnet](#) in the *Amazon VPC User Guide*. When you launch an instance, the IPv6 address is assigned to the primary network interface (eth0) that's created.

For more information, see [IPv6 Addresses](#) (p. 696).

Contents

- [IP Addresses Per Network Interface Per Instance Type](#) (p. 717)
- [Scenarios for Network Interfaces](#) (p. 720)
- [Best Practices for Configuring Network Interfaces](#) (p. 722)
- [Working with Network Interfaces](#) (p. 722)

IP Addresses Per Network Interface Per Instance Type

The following table lists the maximum number of network interfaces per instance type, and the maximum number of private IPv4 addresses and IPv6 addresses per network interface. The limit for IPv6 addresses is separate from the limit for private IPv4 addresses per network interface. Not all instance types support IPv6 addressing. Network interfaces, multiple private IPv4 addresses, and IPv6 addresses are only available for instances running in a VPC. For more information, see [Multiple IP](#)

Amazon Elastic Compute Cloud
User Guide for Windows Instances
IP Addresses Per Network Interface Per Instance Type

[Addresses \(p. 702\)](#). For more information about IPv6 in VPC, see [IP Addressing in Your VPC](#) in the *Amazon VPC User Guide*.

Instance Type	Maximum Network Interfaces	IPv4 Addresses per Interface	IPv6 Addresses per Interface
c1.medium	2	6	IPv6 not supported.
c1.xlarge	4	15	IPv6 not supported.
c3.large	3	10	8
c3.xlarge	4	15	8
c3.2xlarge	4	15	8
c3.4xlarge	8	30	8
c3.8xlarge	8	30	8
c4.large	3	10	8
c4.xlarge	4	15	8
c4.2xlarge	4	15	8
c4.4xlarge	8	30	8
c4.8xlarge	8	30	8
cc2.8xlarge	8	30	IPv6 not supported.
cgl.4xlarge	8	30	IPv6 not supported.
cr1.8xlarge	8	30	IPv6 not supported.
d2.xlarge	4	15	8
d2.2xlarge	4	15	8
d2.4xlarge	8	30	8
d2.8xlarge	8	30	8
g2.2xlarge	4	15	IPv6 not supported.
g2.8xlarge	8	30	IPv6 not supported.
h1.4xlarge	8	30	IPv6 not supported.
hs1.8xlarge	8	30	IPv6 not supported.
i2.xlarge	4	15	8

Amazon Elastic Compute Cloud
User Guide for Windows Instances
IP Addresses Per Network Interface Per Instance Type

Instance Type	Maximum Network Interfaces	IPv4 Addresses per Interface	IPv6 Addresses per Interface
i2.2xlarge	4	15	8
i2.4xlarge	8	30	8
i2.8xlarge	8	30	8
m1.small	2	4	IPv6 not supported.
m1.medium	2	6	IPv6 not supported.
m1.large	3	10	IPv6 not supported.
m1.xlarge	4	15	IPv6 not supported.
m2.xlarge	4	15	IPv6 not supported.
m2.2xlarge	4	30	IPv6 not supported.
m2.4xlarge	8	30	IPv6 not supported.
m3.medium	2	6	IPv6 not supported.
m3.large	3	10	IPv6 not supported.
m3.xlarge	4	15	IPv6 not supported.
m3.2xlarge	4	30	IPv6 not supported.
m4.large	2	10	8
m4.xlarge	4	15	8
m4.2xlarge	4	15	8
m4.4xlarge	8	30	8
m4.10xlarge	8	30	8
m4.16xlarge	8	30	8
p2.xlarge	4	15	8
p2.8xlarge	8	30	8
p2.16xlarge	8	30	8
r3.large	3	10	8

Instance Type	Maximum Network Interfaces	IPv4 Addresses per Interface	IPv6 Addresses per Interface
r3.xlarge	4	15	8
r3.2xlarge	4	15	8
r3.4xlarge	8	30	8
r3.8xlarge	8	30	8
r4.large	3	10	8
r4.xlarge	4	15	8
r4.2xlarge	4	15	8
r4.4xlarge	8	30	8
r4.8xlarge	8	30	8
r4.16xlarge	15	50	8
t1.micro	2	2	IPv6 not supported.
t2.nano	2	2	2
t2.micro	2	2	2
t2.small	2	4	4
t2.medium	3	6	6
t2.large	3	12	8
t2.xlarge	3	15	8
t2.2xlarge	3	15	8
x1.16xlarge	8	30	8
x1.32xlarge	8	30	8

Scenarios for Network Interfaces

Attaching multiple network interfaces to an instance is useful when you want to:

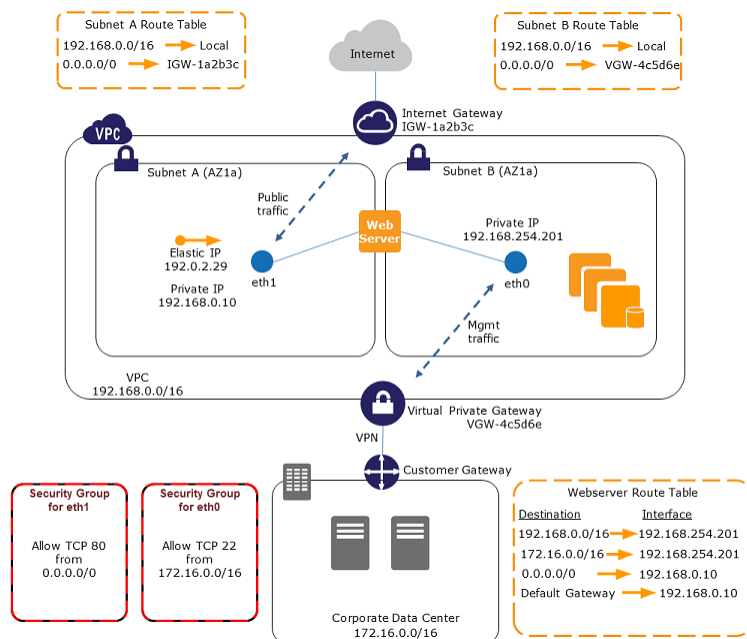
- Create a management network.
- Use network and security appliances in your VPC.
- Create dual-homed instances with workloads/roles on distinct subnets.
- Create a low-budget, high-availability solution.

Creating a Management Network

You can create a management network using network interfaces. In this scenario, the secondary network interface on the instance handles public-facing traffic and the primary network interface handles back-end management traffic and is connected to a separate subnet in your VPC that has

more restrictive access controls. The public-facing interface, which may or may not be behind a load balancer, has an associated security group that allows access to the server from the Internet (for example, allow TCP port 80 and 443 from 0.0.0.0/0, or from the load balancer) while the private facing interface has an associated security group allowing RDP access only from an allowed range of IP addresses either within the VPC or from the Internet, a private subnet within the VPC or a virtual private gateway.

To ensure failover capabilities, consider using a secondary private IPv4 for incoming traffic on a network interface. In the event of an instance failure, you can move the interface and/or secondary private IPv4 address to a standby instance.



Use Network and Security Appliances in Your VPC

Some network and security appliances, such as load balancers, network address translation (NAT) servers, and proxy servers prefer to be configured with multiple network interfaces. You can create and attach secondary network interfaces to instances in a VPC that are running these types of applications and configure the additional interfaces with their own public and private IP addresses, security groups, and source/destination checking.

Creating Dual-homed Instances with Workloads/Roles on Distinct Subnets

You can place a network interface on each of your web servers that connects to a mid-tier network where an application server resides. The application server can also be dual-homed to a back-end network (subnet) where the database server resides. Instead of routing network packets through the dual-homed instances, each dual-homed instance receives and processes requests on the front end, initiates a connection to the back end, and then sends requests to the servers on the back-end network.

Create a Low Budget High Availability Solution

If one of your instances serving a particular function fails, its network interface can be attached to a replacement or hot standby instance pre-configured for the same role in order to rapidly recover the service. For example, you can use a network interface as your primary or secondary network interface

to a critical service such as a database instance or a NAT instance. If the instance fails, you (or more likely, the code running on your behalf) can attach the network interface to a hot standby instance. Because the interface maintains its private IP addresses, Elastic IP addresses, and MAC address, network traffic begins flowing to the standby instance as soon as you attach the network interface to the replacement instance. Users experience a brief loss of connectivity between the time the instance fails and the time that the network interface is attached to the standby instance, but no changes to the VPC route table or your DNS server are required.

Best Practices for Configuring Network Interfaces

- You can attach a network interface to an instance when it's running (hot attach), when it's stopped (warm attach), or when the instance is being launched (cold attach).
- You can detach secondary (ethN) network interfaces when the instance is running or stopped. However, you can't detach the primary (eth0) interface.
- You can attach a network interface in one subnet to an instance in another subnet in the same VPC; however, both the network interface and the instance must reside in the same Availability Zone.
- When launching an instance from the CLI or API, you can specify the network interfaces to attach to the instance for both the primary (eth0) and additional network interfaces.
- Launching an Amazon Linux or Windows Server instance with multiple network interfaces automatically configures interfaces, private IPv4 addresses, and route tables on the operating system of the instance.
- A warm or hot attach of an additional network interface may require you to manually bring up the second interface, configure the private IPv4 address, and modify the route table accordingly. Instances running Amazon Linux or Windows Server automatically recognize the warm or hot attach and configure themselves.
- Attaching another network interface to an instance (for example, a NIC teaming configuration) cannot be used as a method to increase or double the network bandwidth to or from the dual-homed instance.
- If you attach two or more network interfaces from the same subnet to an instance, you may encounter networking issues such as asymmetric routing. If possible, use a secondary private IPv4 address on the primary network interface instead. For more information, see [Assigning a Secondary Private IPv4 Address \(p. 703\)](#). If you need to use multiple network interfaces, you must configure the network interfaces to use static routing. For more information, see [Configure a Secondary Elastic Network Interface \(p. 382\)](#).

Working with Network Interfaces

You can work with network interfaces using the Amazon EC2 console.

Contents

- [Creating a Network Interface \(p. 723\)](#)
- [Deleting a Network Interface \(p. 723\)](#)
- [Viewing Details about a Network Interface \(p. 724\)](#)
- [Monitoring IP Traffic \(p. 724\)](#)
- [Attaching a Network Interface When Launching an Instance \(p. 724\)](#)
- [Attaching a Network Interface to a Stopped or Running Instance \(p. 725\)](#)
- [Detaching a Network Interface from an Instance \(p. 726\)](#)
- [Changing the Security Group \(p. 726\)](#)
- [Changing the Source/Destination Checking \(p. 727\)](#)
- [Associating an Elastic IP Address \(IPv4\) \(p. 727\)](#)
- [Disassociating an Elastic IP Address \(IPv4\) \(p. 728\)](#)

- [Assigning an IPv6 Address](#) (p. 728)
- [Unassigning an IPv6 Address](#) (p. 729)
- [Changing Termination Behavior](#) (p. 729)
- [Adding or Editing a Description](#) (p. 730)
- [Adding or Editing Tags](#) (p. 730)

Creating a Network Interface

You can create a network interface using the Amazon EC2 console or the command line.

To create a network interface using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Choose **Create Network Interface**.
4. For **Description**, enter a descriptive name.
5. For **Subnet**, select the subnet. Note that you can't move the network interface to another subnet after it's created, and you can only attach the interface to instances in the same Availability Zone.
6. For **Private IP** (or **IPv4 Private IP**), enter the primary private IPv4 address. If you don't specify an IPv4 address, we select an available private IPv4 address from within the selected subnet.
7. (IPv6 only) If you selected a subnet that has an associated IPv6 CIDR block, you can optionally specify an IPv6 address in the **IPv6 IP** field.
8. For **Security groups**, select one or more security groups.
9. Choose **Yes, Create**.

To create a network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2](#) (p. 3).

- [create-network-interface](#) (AWS CLI)
- [New-EC2NetworkInterface](#) (AWS Tools for Windows PowerShell)

Deleting a Network Interface

You must first detach a network interface from an instance before you can delete it. Deleting a network interface releases all attributes associated with the interface and releases any private IP addresses or Elastic IP addresses to be used by another instance.

You can delete a network interface using the Amazon EC2 console or the command line.

To delete a network interface using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select a network interface and choose **Delete**.
4. In the **Delete Network Interface** dialog box, choose **Yes, Delete**.

To delete a network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2](#) (p. 3).

- [delete-network-interface](#) (AWS CLI)
- [Remove-EC2NetworkInterface](#) (AWS Tools for Windows PowerShell)

Viewing Details about a Network Interface

You can describe a network interface using the Amazon EC2 console or the command line.

To describe a network interface using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the network interface.
4. View the details on the **Details** tab.

To describe a network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2](#) (p. 3).

- [describe-network-interfaces](#) (AWS CLI)
- [Get-EC2NetworkInterface](#) (AWS Tools for Windows PowerShell)

To describe a network interface attribute using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2](#) (p. 3).

- [describe-network-interface-attribute](#) (AWS CLI)
- [Get-EC2NetworkInterfaceAttribute](#) (AWS Tools for Windows PowerShell)

Monitoring IP Traffic

You can enable a VPC flow log on your network interface to capture information about the IP traffic going to and from the interface. After you've created a flow log, you can view and retrieve its data in Amazon CloudWatch Logs.

For more information, see [VPC Flow Logs](#) in the *Amazon VPC User Guide*.

Attaching a Network Interface When Launching an Instance

You can specify an existing network interface or attach an additional network interface when you launch an instance. You can do this using the Amazon EC2 console or the command line.

Note

If an error occurs when attaching a network interface to your instance, this causes the instance launch to fail.

To attach a network interface when launching an instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Launch Instance**.
3. Select an AMI and instance type and choose **Next: Configure Instance Details**.
4. On the **Configure Instance Details** page, select a VPC for **Network**, and a subnet for **Subnet**.

5. In the **Network Interfaces** section, the console enables you to specify up to two network interfaces (new, existing, or a combination) when you launch an instance. You can also enter a primary IPv4 address and one or more secondary IPv4 addresses for any new interface.

You can add additional network interfaces to the instance after you launch it. The total number of network interfaces that you can attach varies by instance type. For more information, see [IP Addresses Per Network Interface Per Instance Type](#) (p. 717).

Note

You cannot auto-assign a public IPv4 address to your instance if you specify more than one network interface.

6. (IPv6 only) If you're launching an instance into a subnet that has an associated IPv6 CIDR block, you can specify IPv6 addresses for any network interfaces that you attach. Under **IPv6 IPs**, choose **Add IP**. To add a secondary IPv6 address, choose **Add IP** again. You can enter an IPv6 address from the range of the subnet, or leave the default **Auto-assign** value to let Amazon choose an IPv6 address from the subnet for you.
7. Choose **Next: Add Storage**.
8. On the **Add Storage** page, you can specify volumes to attach to the instance besides the volumes specified by the AMI (such as the root device volume), and then choose **Next: Tag Instance**.
9. On the **Tag Instance** page, specify tags for the instance, such as a user-friendly name, and then choose **Next: Configure Security Group**.
10. On the **Configure Security Group** page, you can select a security group or create a new one. Choose **Review and Launch**.

Note

If you specified an existing network interface in step 5, the instance is associated with the security group for that network interface, regardless of any option you select in this step.

11. On the **Review Instance Launch** page, details about the primary and additional network interface are displayed. Review the settings, and then choose **Launch** to choose a key pair and launch your instance. If you're new to Amazon EC2 and haven't created any key pairs, the wizard prompts you to create one.

To attach a network interface when launching an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2](#) (p. 3).

- [run-instances](#) (AWS CLI)
- [New-EC2Instance](#) (AWS Tools for Windows PowerShell)

Attaching a Network Interface to a Stopped or Running Instance

You can attach a network interface to any of your stopped or running instances in your VPC using either the **Instances** or **Network Interfaces** page of the Amazon EC2 console, or using a command line interface.

Note

If the public IPv4 address on your instance is released, it does not receive a new one if there is more than one network interface attached to the instance. For more information about the behavior of public IPv4 addresses, see [Public IPv4 Addresses and External DNS Hostnames](#) (p. 695).

To attach a network interface to an instance using the Instances page

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

2. In the navigation pane, choose **Instances**.
3. Choose **Actions, Networking, Attach Network Interface**.
4. In the **Attach Network Interface** dialog box, select the network interface and choose **Attach**.

To attach a network interface to an instance using the Network Interfaces page

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the network interface and choose **Attach**.
4. In the **Attach Network Interface** dialog box, select the instance and choose **Attach**.

To attach a network interface to an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [attach-network-interface](#) (AWS CLI)
- [Add-EC2NetworkInterface](#) (AWS Tools for Windows PowerShell)

Detaching a Network Interface from an Instance

You can detach a secondary network interface at any time, using either the **Instances** or **Network Interfaces** page of the Amazon EC2 console, or using a command line interface.

To detach a network interface from an instance using the Instances page

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Choose **Actions, Networking, Detach Network Interface**.
4. In the **Detach Network Interface** dialog box, select the network interface and choose **Detach**.

To detach a network interface from an instance using the Network Interfaces page

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the network interface and choose **Detach**.
4. In the **Detach Network Interface** dialog box, choose **Yes, Detach**. If the network interface fails to detach from the instance, choose **Force detachment**, and then try again.

To detach a network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [detach-network-interface](#) (AWS CLI)
- [Dismount-EC2NetworkInterface](#) (AWS Tools for Windows PowerShell)

Changing the Security Group

You can change the security groups that are associated with a network interface. When you create the security group, be sure to specify the same VPC as the subnet for the interface.

You can change the security group for your network interfaces using the Amazon EC2 console or the command line.

Note

To change security group membership for interfaces owned by other services, such as Elastic Load Balancing, use the console or command line interface for that service.

To change the security group of a network interface using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the network interface and choose **Actions, Change Security Groups**.
4. In the **Change Security Groups** dialog box, select the security groups to use, and choose **Save**.

To change the security group of a network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [modify-network-interface-attribute](#) (AWS CLI)
- [Edit-EC2NetworkInterfaceAttribute](#) (AWS Tools for Windows PowerShell)

Changing the Source/Destination Checking

The Source/Destination Check attribute controls whether source/destination checking is enabled on the instance. Disabling this attribute enables an instance to handle network traffic that isn't specifically destined for the instance. For example, instances running services such as network address translation, routing, or a firewall should set this value to `disabled`. The default value is `enabled`.

You can change source/destination checking using the Amazon EC2 console or the command line.

To change source/destination checking for a network interface using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the network interface and choose **Actions, Change Source/Dest Check**.
4. In the dialog box, choose **Enabled** (if enabling) or **Disabled** (if disabling), and **Save**.

To change source/destination checking for a network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [modify-network-interface-attribute](#) (AWS CLI)
- [Edit-EC2NetworkInterfaceAttribute](#) (AWS Tools for Windows PowerShell)

Associating an Elastic IP Address (IPv4)

If you have an Elastic IP address (IPv4), you can associate it with one of the private IPv4 addresses for the network interface. You can associate one Elastic IP address with each private IPv4 address.

You can associate an Elastic IP address using the Amazon EC2 console or the command line.

To associate an Elastic IP address using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the network interface and choose **Actions, Associate Address**.
4. In the **Associate Elastic IP Address** dialog box, select the Elastic IP address from the **Address** list.
5. For **Associate to private IP address**, select the private IPv4 address to associate with the Elastic IP address.
6. Choose **Allow reassociation** to allow the Elastic IP address to be associated with the specified network interface if it's currently associated with another instance or network interface, and then choose **Associate Address**.

To associate an Elastic IP address using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [associate-address](#) (AWS CLI)
- [Register-EC2Address](#) (AWS Tools for Windows PowerShell)

Disassociating an Elastic IP Address (IPv4)

If the network interface has an Elastic IP address (IPv4) associated with it, you can disassociate the address, and then either associate it with another network interface or release it back to the address pool. Note that this is the only way to associate an Elastic IP address with an instance in a different subnet or VPC using a network interface, as network interfaces are specific to a particular subnet.

You can disassociate an Elastic IP address using the Amazon EC2 console or the command line.

To disassociate an Elastic IP address using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the network interface and choose **Actions, Disassociate Address**.
4. In the **Disassociate IP Address** dialog box, choose **Yes, Disassociate**.

To disassociate an Elastic IP address using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [disassociate-address](#) (AWS CLI)
- [Unregister-EC2Address](#) (AWS Tools for Windows PowerShell)

Assigning an IPv6 Address

You can assign one or more IPv6 addresses to a network interface. The network interface must be in a subnet that has an associated IPv6 CIDR block. To assign a specific IPv6 address to the network interface, ensure that the IPv6 address is not already assigned to another network interface.

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

2. In the navigation pane, choose **Network Interfaces** and select the network interface.
3. Choose **Actions, Manage IP Addresses**.
4. Under **IPv6 Addresses**, choose **Assign new IP**. Specify an IPv6 address from the range of the subnet, or leave the **Auto-assign** value to let Amazon choose one for you.
5. Choose **Yes, Update**.

To assign an IPv6 address to a network interface using the command line

- You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).
 - [assign-ipv6-addresses](#) (AWS CLI)
 - [Register-EC2Ipv6AddressList](#) (AWS Tools for Windows PowerShell)

Unassigning an IPv6 Address

You can unassign an IPv6 address from a network interface using the Amazon EC2 console.

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces** and select the network interface.
3. Choose **Actions, Manage IP Addresses**.
4. Under **IPv6 Addresses**, choose **Unassign** for the IPv6 address to remove.
5. Choose **Yes, Update**.

To unassign an IPv6 address from a network interface using the command line

- You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).
 - [unassign-ipv6-addresses](#) (AWS CLI)
 - [Unregister-EC2Ipv6AddressList](#) (AWS Tools for Windows PowerShell)

Changing Termination Behavior

You can set the termination behavior for a network interface attached to an instance so that it is automatically deleted when you delete the instance to which it's attached.

Note

By default, network interfaces that are automatically created and attached to instances using the console are set to terminate when the instance terminates. However, network interfaces created using the command line interface aren't set to terminate when the instance terminates.

You can change the terminating behavior for a network interface using the Amazon EC2 console or the command line.

To change the termination behavior for a network interface using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the network interface and choose **Actions, Change Termination Behavior**.
4. In the **Change Termination Behavior** dialog box, select the **Delete on termination** check box if you want the network interface to be deleted when you terminate an instance.

To change the termination behavior for a network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- `modify-network-interface-attribute` (AWS CLI)
- `Edit-EC2NetworkInterfaceAttribute` (AWS Tools for Windows PowerShell)

Adding or Editing a Description

You can change the description for a network interface using the Amazon EC2 console or the command line.

To change the description for a network interface using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the network interface and choose **Actions, Change Description**.
4. In the **Change Description** dialog box, enter a description for the network interface, and then choose **Save**.

To change the description for a network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- `modify-network-interface-attribute` (AWS CLI)
- `Edit-EC2NetworkInterfaceAttribute` (AWS Tools for Windows PowerShell)

Adding or Editing Tags

Tags are metadata that you can add to a network interface. Tags are private and are only visible to your account. Each tag consists of a key and an optional value. For more information about tags, see [Tagging Your Amazon EC2 Resources \(p. 859\)](#).

You can tag a resource using the Amazon EC2 console or the command line.

To add or edit tags for a network interface using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the network interface.
4. In the details pane, choose **Tags, Add/Edit Tags**.
5. In the **Add/Edit Tags** dialog box, choose **Create Tag** for each tag to create, and enter a key and optional value. When you're done, choose **Save**.

To add or edit tags for a network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- `create-tags` (AWS CLI)

- [New-EC2Tag](#) (AWS Tools for Windows PowerShell)

Placement Groups

A *placement group* is a logical grouping of instances within a single Availability Zone. Placement groups are recommended for applications that benefit from low network latency, high network throughput, or both. To provide the lowest latency, and the highest packet-per-second network performance for your placement group, choose an instance type that supports enhanced networking. For more information, see [Enhanced Networking](#) (p. 737).

First, you create a placement group and then you launch multiple instances into the placement group. We recommend that you launch the number of instances that you need in the placement group in a single launch request and that you use the same instance type for all instances in the placement group. If you try to add more instances to the placement group later, or if you try to launch more than one instance type in the placement group, you increase your chances of getting an insufficient capacity error.

There is no charge for creating a placement group.

If you stop an instance in a placement group and then start it again, it still runs in the placement group. However, the start fails if there isn't enough capacity for the instance.

If you receive a capacity error when launching an instance in a placement group, stop and restart the instances in the placement group, and then try the launch again.

Contents

- [Placement Group Limitations](#) (p. 731)
- [Launching Instances into a Placement Group](#) (p. 732)
- [Deleting a Placement Group](#) (p. 733)

Placement Group Limitations

Placement groups have the following limitations:

- A placement group can't span multiple Availability Zones.
- The name you specify for a placement group must be unique within your AWS account.
- The following are the only instance types that you can use when you launch an instance into a placement group:
 - **General purpose:** m4.large | m4.xlarge | m4.2xlarge | m4.4xlarge | m4.10xlarge | m4.16xlarge
 - **Compute optimized:** c4.large | c4.xlarge | c4.2xlarge | c4.4xlarge | c4.8xlarge | c3.large | c3.xlarge | c3.2xlarge | c3.4xlarge | c3.8xlarge | cc2.8xlarge
 - **Memory optimized:** cr1.8xlarge | r3.large | r3.xlarge | r3.2xlarge | r3.4xlarge | r3.8xlarge | r4.large | r4.xlarge | r4.2xlarge | r4.4xlarge | r4.8xlarge | r4.16xlarge | x1.16xlarge | x1.32xlarge
 - **Storage optimized:** d2.xlarge | d2.2xlarge | d2.4xlarge | d2.8xlarge | hi1.4xlarge | hs1.8xlarge | i2.xlarge | i2.2xlarge | i2.4xlarge | i2.8xlarge
 - **Accelerated computing:** cg1.4xlarge | g2.2xlarge | g2.8xlarge | p2.xlarge | p2.8xlarge | p2.16xlarge
- The maximum network throughput speed of traffic between two instances in a placement group is limited by the slower of the two instances. For applications with high-throughput requirements, choose an instance type with 10 Gbps or 20 Gbps network connectivity. For more information about instance type network performance, see the [Amazon EC2 Instance Types Matrix](#).

- Although launching multiple instance types into a placement group is possible, this reduces the likelihood that the required capacity will be available for your launch to succeed. We recommend using the same instance type for all instances in a placement group.
- You can't merge placement groups. Instead, you must terminate the instances in one placement group, and then relaunch those instances into the other placement group.
- A placement group can span peered VPCs; however, you will not get full-bisection bandwidth between instances in peered VPCs. For more information about VPC peering connections, see [VPC Peering](#) in the *Amazon VPC User Guide*.
- You can't move an existing instance into a placement group. You can create an AMI from your existing instance, and then launch a new instance from the AMI into a placement group.
- Reserved Instances provide a capacity reservation for EC2 instances in an Availability Zone. The capacity reservation can be used by instances in a placement group that are assigned to the same Availability Zone. However, it is not possible to explicitly reserve capacity for a placement group.
- To ensure that network traffic remains within the placement group, members of the placement group must address each other via their private IPv4 addresses or IPv6 addresses (if applicable). If members address each other using their public IPv4 addresses, throughput drops to 5 Gbps or less.
- Network traffic to and from resources outside the placement group is limited to 5 Gbps.

Launching Instances into a Placement Group

We suggest that you create an AMI specifically for the instances that you'll launch into a placement group.

To launch instances into a placement group using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Create an AMI for your instances.
 - a. From the Amazon EC2 dashboard, choose **Launch Instance**. After you complete the wizard, choose **Launch**.
 - b. Connect to your instance. (For more information, see [Connecting to Your Windows Instance](#) (p. 254).)
 - c. Install software and applications on the instance, copy data, or attach additional Amazon EBS volumes.
 - d. (Optional) If your instance type supports enhanced networking, ensure that this feature is enabled by following the procedures in [Enhanced Networking on Windows](#) (p. 737).
 - e. In the navigation pane, choose **Instances**, select your instance, choose **Actions, Image, Create Image**. Provide the information requested by the **Create Image** dialog box, and then choose **Create Image**.
 - f. (Optional) You can terminate this instance if you have no further use for it.
3. Create a placement group.
 - a. In the navigation pane, choose **Placement Groups**.
 - b. Choose **Create Placement Group**.
 - c. In the **Create Placement Group** dialog box, provide a name for the placement group that is unique in the AWS account you're using, and then choose **Create**.

When the status of the placement group is `available`, you can launch instances into the placement group.
4. Launch instances into your placement group.
 - a. In the navigation pane, choose **Instances**.
 - b. Choose **Launch Instance**. Complete the wizard as directed, taking care to do the following:

- On the **Choose an Amazon Machine Image (AMI)** page, select the **My AMIs** tab, and then select the AMI that you created.
- On the **Choose an Instance Type** page, select an instance type that can be launched into a placement group.
- On the **Configure Instance Details** page, enter the total number of instances that you'll need in this placement group, as you might not be able to add instances to the placement group later on.
- On the **Configure Instance Details** page, select the placement group that you created from **Placement group**. If you do not see the **Placement group** list on this page, verify that you have selected an instance type that can be launched into a placement group, as this option is not available otherwise.

To launch instances into a placement group using the command line

1. Create an AMI for your instances using one of the following commands:
 - `create-image` (AWS CLI)
 - `New-EC2Image` (AWS Tools for Windows PowerShell)
2. Create a placement group using one of the following commands:
 - `create-placement-group` (AWS CLI)
 - `New-EC2PlacementGroup` (AWS Tools for Windows PowerShell)
3. Launch instances into your placement group using one of the following options:
 - `--placement` with `run-instances` (AWS CLI)
 - `-PlacementGroup` with `New-EC2Instance` (AWS Tools for Windows PowerShell)

Deleting a Placement Group

You can delete a placement group if you need to replace it or no longer need a placement group. Before you can delete your placement group, you must terminate all instances that you launched into the placement group.

To delete a placement group using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select and terminate all instances in the placement group. (You can verify that the instance is in a placement group before you terminate it by checking the value of **Placement Group** in the details pane.)
4. In the navigation pane, choose **Placement Groups**.
5. Select the placement group, and then choose **Delete Placement Group**.
6. When prompted for confirmation, choose **Yes, Delete**.

To delete a placement group using the command line

You can use one of the following sets of commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- `terminate-instances` and `delete-placement-group` (AWS CLI)
- `Stop-EC2Instance` and `Remove-EC2PlacementGroup` (AWS Tools for Windows PowerShell)

Network Maximum Transmission Unit (MTU) for Your EC2 Instance

The maximum transmission unit (MTU) of a network connection is the size, in bytes, of the largest permissible packet that can be passed over the connection. The larger the MTU of a connection, the more data that can be passed in a single packet. Ethernet packets consist of the frame, or the actual data you are sending, and the network overhead information that surrounds it.

Ethernet frames can come in different formats, and the most common format is the standard Ethernet v2 frame format. It supports 1500 MTU, which is the largest Ethernet packet size supported over most of the Internet. The maximum supported MTU for an instance depends on its instance type. All Amazon EC2 instance types support 1500 MTU, and many current instance sizes support 9001 MTU, or jumbo frames.

Contents

- [Jumbo Frames \(9001 MTU\) \(p. 734\)](#)
- [Path MTU Discovery \(p. 734\)](#)
- [Check the Path MTU Between Two Hosts \(p. 735\)](#)
- [Check and Set the MTU on your Amazon EC2 Instance \(p. 735\)](#)
- [Troubleshooting \(p. 737\)](#)

Jumbo Frames (9001 MTU)

Jumbo frames allow more than 1500 bytes of data by increasing the payload size per packet, and thus increasing the percentage of the packet that is not packet overhead. Fewer packets are needed to send the same amount of usable data. However, outside of a given AWS region (EC2-Classic), a single VPC, or a VPC peering connection, you will experience a maximum path of 1500 MTU. VPN connections and traffic sent over an Internet gateway are limited to 1500 MTU. If packets are over 1500 bytes, they are fragmented, or they are dropped if the `Don't Fragment` flag is set in the IP header.

Jumbo frames should be used with caution for Internet-bound traffic or any traffic that leaves a VPC. Packets are fragmented by intermediate systems, which slows down this traffic. To use jumbo frames inside a VPC and not slow traffic that's bound for outside the VPC, you can configure the MTU size by route, or use multiple elastic network interfaces with different MTU sizes and different routes.

For instances that are collocated inside a placement group, jumbo frames help to achieve the maximum network throughput possible, and they are recommended in this case. For more information, see [Placement Groups \(p. 731\)](#).

The following instances support jumbo frames:

- Compute optimized: C3, C4, CC2
- General purpose: M3, M4, T2
- Accelerated computing: CG1, G2, P2
- Memory optimized: CR1, R3, R4, X1
- Storage optimized: D2, HI1, HS1, I2

Path MTU Discovery

Path MTU Discovery is used to determine the path MTU between two devices. The path MTU is the maximum packet size that's supported on the path between the originating host and the receiving

host. If a host sends a packet that's larger than the MTU of the receiving host or that's larger than the MTU of a device along the path, the receiving host or device returns the following ICMP message: Destination Unreachable: Fragmentation Needed and Don't Fragment was Set (Type 3, Code 4). This instructs the original host to adjust the MTU until the packet can be transmitted.

By default, security groups do not allow any inbound ICMP traffic. To ensure that your instance can receive this message and the packet does not get dropped, you must add a **Custom ICMP Rule** with the **Destination Unreachable** protocol to the inbound security group rules for your instance. For more information, see the [Adding Rules to a Security Group \(p. 611\)](#) and [API and Command Overview \(p. 612\)](#) sections in the Amazon EC2 Security Groups topic.

Important

Modifying your instance's security group to allow path MTU discovery does not guarantee that jumbo frames will not be dropped by some routers. An Internet gateway in your VPC will forward packets up to 1500 bytes only. 1500 MTU packets are recommended for Internet traffic.

Check the Path MTU Between Two Hosts

You can check the path MTU between two hosts using the **mturoute.exe** command, which you can download and install from <http://www.elifulkerson.com/projects/mturoute.php>.

To check path MTU with mtroute.exe

1. Download **mturoute.exe** from <http://www.elifulkerson.com/projects/mturoute.php>.
2. Open a command prompt window and change to the directory where you downloaded **mturoute.exe**.
3. Use the following command to check the path MTU between your Amazon EC2 instance and another host. You can use a DNS name or an IP address as the destination; this example checks the path MTU between an EC2 instance and `www.elifulkerson.com`.

```
PS C:\Users\Administrator\Downloads> .\mturoute.exe www.elifulkerson.com
* ICMP Fragmentation is not permitted. *
* Speed optimization is enabled. *
* Maximum payload is 10000 bytes. *
+ ICMP payload of 1472 bytes succeeded.
- ICMP payload of 1473 bytes is too big.
Path MTU: 1500 bytes.
```

In this example, the path MTU is 1500.

Check and Set the MTU on your Amazon EC2 Instance

Some AMIs are configured to use jumbo frames on instance that support them, and others are configured to use standard frame sizes. You may want to use jumbo frames for network traffic within your VPC or you may want to use standard frames for Internet traffic. Whatever your use case, we recommend verifying that your instance will behave the way you expect it to. You can use the procedures in this section to check your network interface's MTU setting and modify it if needed.

To check the MTU setting on a Windows instance

- If your instance uses a Windows operating system, you can review the MTU value with the **netsh** command. Run the following command to determine the current MTU value:

```
PS C:\Users\Administrator> netsh interface ipv4 show subinterface
```

MTU	MediaSenseState	Bytes In	Bytes Out	Interface
9001	1	317337	692805	Ethernet

In the resulting output, look for the entry titled "Ethernet," "Ethernet 2," or "Local Area Connection."

In the above example, the *9001* in the **MTU** column indicates that this instance uses jumbo frames.

To set the MTU value on a Windows instance

1. If your instance uses a Windows operating system, you can set the MTU value with the **netsh** command. Run the following command to set the desired MTU value.

Note

These steps vary based on the network drivers your Windows instance uses; make sure to execute the correct command for your driver version. For more information, see [Paravirtual Drivers \(p. 352\)](#).

- For Windows instances that use AWS PV drivers or the Intel network driver for enhanced networking (for example, Windows Server 2012 R2), execute the following command to set the MTU to 1500.

```
PS C:\Users\Administrator> netsh interface ipv4 set subinterface  
"Ethernet" mtu=1500 store=persistent  
Ok.
```

To set the MTU to 9001, execute the following commands.

```
PS C:\Users\Administrator> netsh interface ipv4 set subinterface  
"Ethernet" mtu=9001 store=persistent  
Ok.
```

To finish setting the MTU to 9001, execute the following command. This command is not necessary if setting the MTU to 1500.

```
PS C:\Users\Administrator> - Set-NetAdapterAdvancedProperty -Name  
"Ethernet" -RegistryKeyword "*JumboPacket" -RegistryValue 9014
```

Note

If you receive an `Element not found` error, replace *Ethernet* with the `Interface` column output from the [To check the MTU setting on a Windows instance \(p. 735\)](#) procedure that matches your interface.

- For Windows instances that use Citrix PV drivers, first ensure that your PV drivers are up to date by following the procedures in [Upgrading PV Drivers on Your Windows AMI \(p. 356\)](#). Then, execute the following command to set the MTU to 1500. Citrix PV drivers interpret MTU to mean max frame size, so you must subtract 18 from your `mtu` setting to set the correct value. For example, to set 1500 MTU, use 1482 in the command below, and to set 9001 MTU, use 8983 instead.

```
PS C:\Users\Administrator> netsh interface ipv4 set subinterface  
"Local Area Connection" mtu=1482 store=persistent
```


Ok.

Note

If you receive an `Element not found` error, replace `Local Area Connection` with the `Interface` column output from the [To check the MTU setting on a Windows instance \(p. 735\)](#) procedure that matches your interface.

2. (Optional) Reboot your instance and verify that the MTU setting is correct.

Troubleshooting

If you experience connectivity issues between your EC2 instance and an Amazon Redshift cluster when using jumbo frames, see [Queries Appear to Hang](#) in the *Amazon Redshift Cluster Management Guide*

Enhanced Networking on Windows

Enhanced networking uses single root I/O virtualization (SR-IOV) to provide high-performance networking capabilities on [supported instance types \(p. 737\)](#). SR-IOV is a method of device virtualization that provides higher I/O performance and lower CPU utilization when compared to traditional virtualized network interfaces. Enhanced networking provides higher bandwidth, higher packet per second (PPS) performance, and consistently lower inter-instance latencies. There is no additional charge for using enhanced networking.

Contents

- [Enhanced Networking Types \(p. 737\)](#)
- [Enabling Enhanced Networking on Your Instance \(p. 738\)](#)
- [Enabling Enhanced Networking with the Intel 82599 VF Interface on Windows Instances in a VPC \(p. 738\)](#)
- [Enabling Enhanced Networking with the Elastic Network Adapter \(ENA\) on Windows Instances in a VPC \(p. 740\)](#)

Enhanced Networking Types

Depending on your instance type, enhanced networking can be enabled using one of the following mechanisms:

Intel 82599 Virtual Function (VF) interface

The Intel 82599 Virtual Function interface supports network speeds of up to 10 Gbps for supported instance types. For more information, see the [Instance Type Matrix](#).

C3, C4, D2, I2, R3, and M4 (excluding `m4.16xlarge`) instances use the Intel 82599 VF interface for enhanced networking.

Elastic Network Adapter (ENA)

The Elastic Network Adapter (ENA) supports network speeds of up to 20 Gbps.

P2, R4, X1, and `m4.16xlarge` instances use the Elastic Network Adapter for enhanced networking.

Enabling Enhanced Networking on Your Instance

If your instance type supports the Intel 82599 VF interface for enhanced networking, follow the procedures in [Enabling Enhanced Networking with the Intel 82599 VF Interface on Windows Instances in a VPC](#) (p. 738).

If your instance type supports the Elastic Network Adapter for enhanced networking, follow the procedures in [Enabling Enhanced Networking with the Elastic Network Adapter \(ENA\) on Windows Instances in a VPC](#) (p. 740).

Enabling Enhanced Networking with the Intel 82599 VF Interface on Windows Instances in a VPC

Amazon EC2 provides enhanced networking capabilities to C3, C4, D2, I2, R3, and M4 (excluding m4.16xlarge) instances with the Intel 82599 VF interface, which uses the Intel `ixgbevf` driver.

To prepare for enhanced networking with the Intel 82599 VF interface, set up your instance as follows:

- Launch the instance from a 64-bit HVM AMI for Windows Server 2012 or Windows Server 2008 R2. (You can't enable enhanced networking on Windows Server 2008 and Windows Server 2003, and enhanced networking is already enabled on Windows Server 2012 R2.) Windows Server Enhanced networking is already enabled for Windows Server 2012 R2 AMIs. However, Windows Server 2012 R2 includes Intel driver 1.0.15.3 and we recommend that you upgrade that driver to the latest version using the `Pnputil.exe` utility.
- Launch the instance in a VPC. (You can't enable enhanced networking if the instance is in EC2-Classic.)
- Install and configure the [AWS CLI](#) on any computer you choose, preferably your local desktop or laptop. For more information, see [Accessing Amazon EC2](#) (p. 3). Enhanced networking cannot be managed from the Amazon EC2 console.
- If you have important data on the instance that you want to preserve, you should back that data up now by creating an AMI from your instance. Updating kernels and kernel modules, as well as enabling the `sriovNetSupport` attribute, may render incompatible instances or operating systems unreachable; if you have a recent backup, your data will still be retained if this happens.

Contents

- [Testing Whether Enhanced Networking with the Intel 82599 VF Interface is Enabled](#) (p. 738)
- [Enabling Enhanced Networking with the Intel 82599 VF Interface on Windows](#) (p. 739)

Testing Whether Enhanced Networking with the Intel 82599 VF Interface is Enabled

To test whether enhanced networking with the Intel 82599 VF interface is already enabled, verify that the driver is installed on your instance and that the `sriovNetSupport` attribute is set.

Driver

To verify that the driver is installed, connect to your instance and open Device Manager. You should see "Intel(R) 82599 Virtual Function" listed under **Network adapters**.

Instance Attribute (`sriovNetSupport`)

To check whether an instance has the enhanced networking `sriovNetSupport` attribute set, use the following command:

- [describe-instance-attribute](#) (AWS CLI)

```
C:\> aws ec2 describe-instance-attribute --instance-id instance_id --attribute sriovNetSupport
```

If the attribute isn't set, `SriovNetSupport` is empty; otherwise, it is set as follows:

```
"SriovNetSupport": {  
  "Value": "simple"  
},
```

Image Attribute (sriovNetSupport)

To check whether an AMI already has the enhanced networking `sriovNetSupport` attribute set, use the following command:

- [describe-image-attribute](#) (AWS CLI)

```
C:\> aws ec2 describe-image-attribute --image-id ami_id --attribute sriovNetSupport
```

Note

This command only works for images that you own. You receive an `AuthFailure` error for images that do not belong to your account.

If the attribute isn't set, `SriovNetSupport` is empty; otherwise, it is set as follows:

```
"SriovNetSupport": {  
  "Value": "simple"  
},
```

Enabling Enhanced Networking with the Intel 82599 VF Interface on Windows

If you launched your instance and it does not have enhanced networking enabled already, you must download and install the required network adapter driver on your instance, and then set the `sriovNetSupport` instance attribute to activate enhanced networking. You can only enable this attribute on supported instance types. For more information, see [Enhanced Networking Types](#) (p. 737).

Important

Windows Server Enhanced networking is already enabled for Windows Server 2012 R2 AMIs. However, Windows Server 2012 R2 includes Intel driver 1.0.15.3 and we recommend that you upgrade that driver to the latest version using the `Pnputil.exe` utility as described here.

To enable enhanced networking

1. Connect to your instance and log in as the local administrator.
2. From the instance, install the driver as follows:
 - a. Download the Intel network adapter driver for your operating system.
 - [Windows Server 2008 R2](#)

- [Windows Server 2012](#)
 - [Windows Server 2012 R2](#)
- In the **Download** folder, locate the `PROWinx64.exe` file. Rename this file `PROWinx64.zip`.
 - Open a context (right-click) menu on `PROWinx64.zip` and choose **Extract All**. Specify a destination path and choose **Extract**.
 - Open a command prompt window, go to the folder with the extracted files, and use the `pnputil` utility to add and install the INF file in the driver store.

Windows Server 2012 R2

```
C:\> pnputil -i -a PROXGB\Winx64\NDIS64\vxn64x64.inf
```

Windows Server 2012

```
C:\> pnputil -i -a PROXGB\Winx64\NDIS63\vxn63x64.inf
```

Windows Server 2008 R2

```
C:\> pnputil -a PROXGB\Winx64\NDIS62\vxn62x64.inf
```

- From your local computer, stop the instance using the Amazon EC2 console or the following command: [stop-instances](#) (AWS CLI). If your instance is managed by AWS OpsWorks, you should stop the instance in the AWS OpsWorks console so that the instance state remains in sync.
- From a command prompt window, enable the enhanced networking attribute using the following command.

Warning

There is no way to disable the enhanced networking attribute after you've enabled it.

- [modify-instance-attribute](#) (AWS CLI)

```
C:\> aws ec2 modify-instance-attribute --instance-id instance_id --  
sriov-net-support simple
```

- (Optional) Create an AMI from the instance, as described in [Creating an Amazon EBS-Backed Windows AMI \(p. 77\)](#). The AMI inherits the enhanced networking attribute from the instance. Therefore, you can use this AMI to launch another instance with enhanced networking enabled by default.
- From your local computer, start the instance using the Amazon EC2 console or the following command: [start-instances](#) (AWS CLI). If your instance is managed by AWS OpsWorks, you should start the instance in the AWS OpsWorks console so that the instance state remains in sync.

Enabling Enhanced Networking with the Elastic Network Adapter (ENA) on Windows Instances in a VPC

To prepare for enhanced networking with the ENA network adapter, set up your instance as follows:

- Launch the instance in a VPC. (You can't enable enhanced networking if the instance is in EC2-Classic.)

- Install and configure the [AWS CLI](#) on any computer you choose, preferably your local desktop or laptop. For more information, see [Accessing Amazon EC2 \(p. 3\)](#). Enhanced networking cannot be managed from the Amazon EC2 console.
- If you have important data on the instance that you want to preserve, you should back that data up now by creating an AMI from your instance. Updating kernels and kernel modules, as well as enabling the `enaSupport` attribute, may render incompatible instances or operating systems unreachable; if you have a recent backup, your data will still be retained if this happens.

Contents

- [Testing Whether Enhanced Networking with ENA Is Enabled \(p. 741\)](#)
- [Enabling Enhanced Networking with ENA on Windows \(p. 742\)](#)

Testing Whether Enhanced Networking with ENA Is Enabled

To test whether enhanced networking with ENA is already enabled, verify that the driver is installed on your instance and that the `enaSupport` attribute is set.

Instance Attribute (enaSupport)

To check whether an instance already has the enhanced networking `enaSupport` attribute set, use the following command:

- [describe-instances](#) (AWS CLI)

```
C:\> aws ec2 describe-instances --instance-id instance_id --query  
Reservations[ ].Instances[ ].EnaSupport
```

If the `enaSupport` attribute isn't set, the returned JSON is empty; otherwise, it is set as follows:

```
[  
  true  
]
```

Image Attribute (enaSupport)

To check whether an AMI already has the enhanced networking `enaSupport` attribute set, use the following command:

- [describe-image-attribute](#) (AWS CLI)

```
C:\> aws ec2 describe-image-attribute --image-id ami_id --attribute  
enaSupport
```

Note

This command only works for images that you own. You receive an `AuthFailure` error for images that do not belong to your account.

If the attribute isn't set, `EnaSupport` is empty; otherwise, it is set as follows:

```
{  
  "EnaSupport": {
```

```
        "Value": true
      },
      "ImageId": "ami_id"
    }
  }
```

Enabling Enhanced Networking with ENA on Windows

If you launched your instance and it does not have enhanced networking enabled already, you must download and install the required network adapter driver on your instance, and then set the `enaSupport` instance attribute to activate enhanced networking. You can only enable this attribute on supported instance types. For more information, see [Enhanced Networking Types \(p. 737\)](#).

To enable enhanced networking with ENA

1. Connect to your instance and log in as the local administrator.
2. From the instance, install the driver as follows:
 - a. Download the Amazon ENA adapter driver [package](#).
 - b. Extract the zip archive.
 - c. Open a Command Prompt window and navigate to the folder containing the driver for your Windows version, which must be one of the following:
 - Windows Server 2008 R2
 - Windows Server 2012
 - Windows Server 2012 R2
 - d. In the folder corresponding to your OS version, you should see three driver files: `ena.cat`, `ena.inf`, and `ena.sys`. Install the driver with the following command:

```
C:\> pnputil -i -a ena.inf
```

This should yield the following output if the installation is successful:

```
Microsoft PnP Utility

Processing inf :          ena.inf
Successfully installed the driver on a device on the system.
Driver package added successfully.
Published name :          oem9.inf

Total attempted:          1
Number successfully imported: 1
```

3. From your local computer, stop the instance using the Amazon EC2 console or the following command: [stop-instances](#) (AWS CLI). If your instance is managed by AWS OpsWorks, you should stop the instance in the AWS OpsWorks console so that the instance state remains in sync.
4. Enable ENA support on your instance.

Note

You cannot enable ENA support on the instance unless you have previously installed the ENA driver as described above.

- a. From your local computer, check the EC2 instance ENA-support attribute on your instance by running the following command. EnaSupport is set to false by default.

```
C:\> aws ec2 describe-instances --instance-id "instance-id" --query  
Reservations[ ].Instances[ ].EnaSupport
```

If the attribute is not enabled, the output will be "[]".

- b. To enable ENA support, run the following command, which returns no output:

```
C:\> aws ec2 modify-instance-attribute --instance-id "instance-id" --  
ena-support
```

Note

If you encounter problems when you restart the instance, you can also disable ENA support with the following command:

```
C:\> aws ec2 modify-instance-attribute --instance-id "instance-  
id" --no-ena-support
```

- c. Verify that the attribute has been set to `true` by again running the **describe-instances** command as shown above. You should now see:

```
[  
  true  
]
```

5. From your local computer, start the instance using the Amazon EC2 console or the following command: [start-instances](#) (AWS CLI). If your instance is managed by AWS OpsWorks, you should start the instance in the AWS OpsWorks console so that the instance state remains in sync.
6. On the instance, validate that the ENA driver is installed and working.
 - a. Right-click the network icon and choose **Open Network and Sharing Center**.
 - b. Choose the Ethernet adapter, for example, **Ethernet 2**.
 - c. Choose **Details**. The **Network Connection Details** window, the **Description** field should have the value **Amazon Elastic Network Adapter**.
7. (Optional) Create an AMI from the instance. The AMI will inherit the enhanced networking `enaSupport` attribute from the instance. Therefore, you can use this AMI to launch another instance with enhanced networking with ENA enabled by default.

If your instance is an EBS-backed instance, create a new AMI as described in [Creating an Amazon EBS-Backed Windows AMI](#).

If your instance is an instance store-backed instance, create a new AMI as described in [Creating an Instance Store-Backed Windows AMI](#). To enable enhanced networking by default on instances created from the AMI, be sure to include the `--ena-support` flag when you register it.

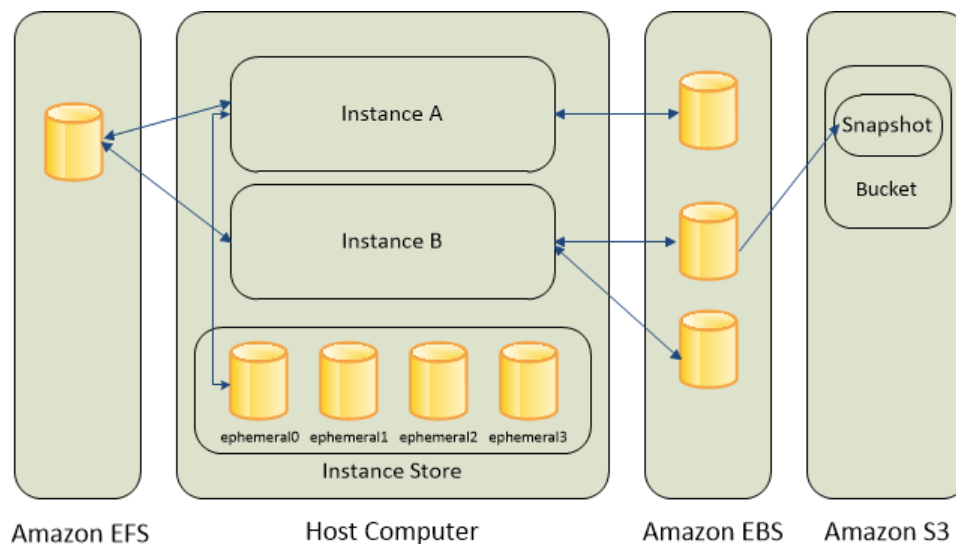
Storage

Amazon EC2 provides you with flexible, cost effective, and easy-to-use data storage options for your instances. Each option has a unique combination of performance and durability. These storage options can be used independently or in combination to suit your requirements.

After reading this section, you should have a good understanding about how you can use the data storage options supported by Amazon EC2 to meet your specific requirements. These storage options include the following:

- [Amazon Elastic Block Store \(Amazon EBS\)](#) (p. 745)
- [Amazon EC2 Instance Store](#) (p. 822)
- [Amazon Elastic File System \(Amazon EFS\)](#) (p. 829)
- [Amazon Simple Storage Service \(Amazon S3\)](#) (p. 829)

The following figure shows the relationship between these types of storage.



Amazon EBS

Amazon EBS provides durable, block-level storage volumes that you can attach to a running instance. You can use Amazon EBS as a primary storage device for data that requires frequent and granular updates. For example, Amazon EBS is the recommended storage option when you run a database on an instance.

An EBS volume behaves like a raw, unformatted, external block device that you can attach to a single instance. The volume persists independently from the running life of an instance. After an EBS volume is attached to an instance, you can use it like any other physical hard drive. As illustrated in the previous figure, multiple volumes can be attached to an instance. You can also detach an EBS volume from one instance and attach it to another instance. EBS volumes can also be created as encrypted volumes using the Amazon EBS encryption feature. For more information, see [Amazon EBS Encryption \(p. 799\)](#).

To keep a backup copy of your data, you can create a *snapshot* of an EBS volume, which is stored in Amazon S3. You can create an EBS volume from a snapshot, and attach it to another instance. For more information, see [Amazon Elastic Block Store \(Amazon EBS\) \(p. 745\)](#).

Amazon EC2 Instance Store

Many instances can access storage from disks that are physically attached to the host computer. This disk storage is referred to as *instance store*. Instance store provides temporary block-level storage for instances. The data on an instance store volume persists only during the life of the associated instance; if you stop or terminate an instance, any data on instance store volumes is lost. For more information, see [Amazon EC2 Instance Store \(p. 822\)](#).

Amazon EFS File System

Amazon EFS provides scalable file storage for use with Amazon EC2. You can create an EFS file system and configure your instances to mount the file system. You can use an EFS file system as a common data source for workloads and applications running on multiple instances. For more information, see [Amazon Elastic File System \(Amazon EFS\) \(p. 829\)](#).

Amazon S3

Amazon S3 provides access to reliable and inexpensive data storage infrastructure. It is designed to make web-scale computing easier by enabling you to store and retrieve any amount of data, at any time, from within Amazon EC2 or anywhere on the web. For example, you can use Amazon S3 to store backup copies of your data and applications. Amazon EC2 uses Amazon S3 to store EBS snapshots and instance store-backed AMIs. For more information, see [Amazon Simple Storage Service \(Amazon S3\) \(p. 829\)](#).

Adding Storage

Every time you launch an instance from an AMI, a root storage device is created for that instance. The root storage device contains all the information necessary to boot the instance. You can specify storage volumes in addition to the root device volume when you create an AMI or launch an instance using *block device mapping*. For more information, see [Block Device Mapping \(p. 833\)](#).

You can also attach EBS volumes to a running instance. For more information, see [Attaching an Amazon EBS Volume to an Instance \(p. 766\)](#).

Amazon Elastic Block Store (Amazon EBS)

Amazon Elastic Block Store (Amazon EBS) provides block level storage volumes for use with EC2 instances. EBS volumes are highly available and reliable storage volumes that can be attached to any running instance that is in the same Availability Zone. EBS volumes that are attached to an EC2 instance are exposed as storage volumes that persist independently from the life of the instance. With Amazon EBS, you pay only for what you use. For more information about Amazon EBS pricing, see the Projecting Costs section of the [Amazon Elastic Block Store page](#).

Amazon EBS is recommended when data must be quickly accessible and requires long-term persistence. EBS volumes are particularly well-suited for use as the primary storage for file systems,

databases, or for any applications that require fine granular updates and access to raw, unformatted, block-level storage. Amazon EBS is well suited to both database-style applications that rely on random reads and writes, and to throughput-intensive applications that perform long, continuous reads and writes.

For simplified data encryption, you can launch your EBS volumes as encrypted volumes. Amazon EBS encryption offers you a simple encryption solution for your EBS volumes without the need for you to build, manage, and secure your own key management infrastructure. When you create an encrypted EBS volume and attach it to a supported instance type, data stored at rest on the volume, disk I/O, and snapshots created from the volume are all encrypted. The encryption occurs on the servers that hosts EC2 instances, providing encryption of data-in-transit from EC2 instances to EBS storage. For more information, see [Amazon EBS Encryption \(p. 799\)](#).

Amazon EBS encryption uses AWS Key Management Service (AWS KMS) master keys when creating encrypted volumes and any snapshots created from your encrypted volumes. The first time you create an encrypted EBS volume in a region, a default master key is created for you automatically. This key is used for Amazon EBS encryption unless you select a Customer Master Key (CMK) that you created separately using the AWS Key Management Service. Creating your own CMK gives you more flexibility, including the ability to create, rotate, disable, define access controls, and audit the encryption keys used to protect your data. For more information, see the [AWS Key Management Service Developer Guide](#).

You can attach multiple volumes to the same instance within the limits specified by your AWS account. Your account has a limit on the number of EBS volumes that you can use, and the total storage available to you. For more information about these limits, and how to request an increase in your limits, see [Request to Increase the Amazon EBS Volume Limit](#).

Contents

- [Features of Amazon EBS \(p. 746\)](#)
- [Amazon EBS Volumes \(p. 747\)](#)
- [Amazon EBS Snapshots \(p. 788\)](#)
- [Amazon EBS–Optimized Instances \(p. 795\)](#)
- [Amazon EBS Encryption \(p. 799\)](#)
- [Amazon EBS Volume Performance on Windows Instances \(p. 803\)](#)
- [Amazon CloudWatch Events for Amazon EBS \(p. 816\)](#)

Features of Amazon EBS

- You can create EBS General Purpose SSD (`gp2`), Provisioned IOPS SSD (`io1`), Throughput Optimized HDD (`st1`), and Cold HDD (`sc1`) volumes up to 16 TiB in size. You can mount these volumes as devices on your Amazon EC2 instances. You can mount multiple volumes on the same instance, but each volume can be attached to only one instance at a time. For more information, see [Creating an Amazon EBS Volume \(p. 761\)](#).
- With General Purpose SSD (`gp2`) volumes, you can expect base performance of 3 IOPS/GiB, with the ability to burst to 3,000 IOPS for extended periods of time. `gp2` volumes are ideal for a broad range of use cases such as boot volumes, small and medium-size databases, and development and test environments. `gp2` volumes support up to 10,000 IOPS and 160 MB/s of throughput. For more information, see [General Purpose SSD \(`gp2`\) Volumes \(p. 752\)](#).
- With Provisioned IOPS SSD (`io1`) volumes, you can provision a specific level of I/O performance. `io1` volumes support up to 20,000 IOPS and 320 MB/s of throughput. This allows you to predictably scale to tens of thousands of IOPS per EC2 instance. For more information, see [Provisioned IOPS SSD \(`io1`\) Volumes \(p. 754\)](#).
- Throughput Optimized HDD (`st1`) volumes provide low-cost magnetic storage that defines performance in terms of throughput rather than IOPS. With throughput of up to 500 MiB/s, this

volume type is a good fit for large, sequential workloads such as Amazon EMR, ETL, data warehouses, and log processing. For more information, see [Throughput Optimized HDD \(st1\) Volumes](#) (p. 754).

- Cold HDD (sc1) volumes provide low-cost magnetic storage that defines performance in terms of throughput rather than IOPS. With throughput of up to 250 MiB/s, sc1 is a good fit ideal for large, sequential, cold-data workloads. If you require infrequent access to your data and are looking to save costs, sc1 provides inexpensive block storage. For more information, see [Cold HDD \(sc1\) Volumes](#) (p. 757).
- EBS volumes behave like raw, unformatted block devices. You can create a file system on top of these volumes, or use them in any other way you would use a block device (like a hard drive). For more information on creating file systems and mounting volumes, see [Making an Amazon EBS Volume Available for Use](#) (p. 767).
- You can use encrypted EBS volumes to meet a wide range of data-at-rest encryption requirements for regulated/audited data and applications. For more information, see [Amazon EBS Encryption](#) (p. 799).
- You can create point-in-time snapshots of EBS volumes, which are persisted to Amazon S3. Snapshots protect data for long-term durability, and they can be used as the starting point for new EBS volumes. The same snapshot can be used to instantiate as many volumes as you wish. These snapshots can be copied across AWS regions. For more information, see [Amazon EBS Snapshots](#) (p. 788).
- EBS volumes are created in a specific Availability Zone, and can then be attached to any instances in that same Availability Zone. To make a volume available outside of the Availability Zone, you can create a snapshot and restore that snapshot to a new volume anywhere in that region. You can copy snapshots to other regions and then restore them to new volumes there, making it easier to leverage multiple AWS regions for geographical expansion, data center migration, and disaster recovery. For more information, see [Creating an Amazon EBS Snapshot](#) (p. 789), [Restoring an Amazon EBS Volume from a Snapshot](#) (p. 763), and [Copying an Amazon EBS Snapshot](#) (p. 791).
- A large repository of public data set snapshots can be restored to EBS volumes and seamlessly integrated into AWS cloud-based applications. For more information, see [Using Public Data Sets](#) (p. 848).
- Performance metrics, such as bandwidth, throughput, latency, and average queue length, are available through the AWS Management Console. These metrics, provided by Amazon CloudWatch, allow you to monitor the performance of your volumes to make sure that you are providing enough performance for your applications without paying for resources you don't need. For more information, see [Amazon EBS Volume Performance on Windows Instances](#) (p. 803).

Amazon EBS Volumes

An Amazon EBS volume is a durable, block-level storage device that you can attach to a single EC2 instance. You can use EBS volumes as primary storage for data that requires frequent updates, such as the system drive for an instance or storage for a database application, or for throughput-intensive applications that perform continuous disk scans. EBS volumes persist independently from the running life of an EC2 instance. After a volume is attached to an instance, you can use it like any other physical hard drive. Amazon EBS provides the following volume types: General Purpose SSD (gp2), Provisioned IOPS SSD (io1), Throughput Optimized HDD (st1), Cold HDD (sc1), and Magnetic (standard). They differ in performance characteristics and price, allowing you to tailor your storage performance and cost to the needs of your applications. For more information, see [Amazon EBS Volume Types](#) (p. 749).

Contents

- [Benefits of Using EBS Volumes](#) (p. 748)
- [Amazon EBS Volume Types](#) (p. 749)
- [Creating an Amazon EBS Volume](#) (p. 761)

- [Restoring an Amazon EBS Volume from a Snapshot \(p. 763\)](#)
- [Attaching an Amazon EBS Volume to an Instance \(p. 766\)](#)
- [Making an Amazon EBS Volume Available for Use \(p. 767\)](#)
- [Viewing Volume Information \(p. 769\)](#)
- [Monitoring the Status of Your Volumes \(p. 769\)](#)
- [Detaching an Amazon EBS Volume from an Instance \(p. 781\)](#)
- [Deleting an Amazon EBS Volume \(p. 783\)](#)
- [Expanding the Storage Space of an EBS Volume on Windows \(p. 783\)](#)

Benefits of Using EBS Volumes

EBS volumes provide several benefits that are not supported by instance store volumes.

- **Data availability**

When you create an EBS volume in an Availability Zone, it is automatically replicated within that zone to prevent data loss due to failure of any single hardware component. After you create a volume, you can attach it to any EC2 instance in the same Availability Zone. After you attach a volume, it appears as a native block device similar to a hard drive or other physical device. At that point, the instance can interact with the volume just as it would with a local drive; the instance can format the EBS volume with a file system, such as NTFS, and then install applications.

An EBS volume can be attached to only one instance at a time within the same Availability Zone. However, multiple volumes can be attached to a single instance. If you attach multiple volumes to a device that you have named, you can stripe data across the volumes for increased I/O and throughput performance.

You can get monitoring data for your EBS volumes at no additional charge (this includes data for the root device volumes for EBS-backed instances). For more information, see [Monitoring Volumes with CloudWatch \(p. 769\)](#).

- **Data persistence**

An EBS volume is off-instance storage that can persist independently from the life of an instance. You continue to pay for the volume usage as long as the data persists.

By default, EBS volumes that are attached to a running instance automatically detach from the instance with their data intact when that instance is terminated. The volume can then be reattached to a new instance, enabling quick recovery. If you are using an EBS-backed instance, you can stop and restart that instance without affecting the data stored in the attached volume. The volume remains attached throughout the stop-start cycle. This enables you to process and store the data on your volume indefinitely, only using the processing and storage resources when required. The data persists on the volume until the volume is deleted explicitly. The physical block storage used by deleted EBS volumes is overwritten with zeroes before it is allocated to another account. If you are dealing with sensitive data, you should consider encrypting your data manually or storing the data on a volume protected by Amazon EBS encryption. For more information, see [Amazon EBS Encryption \(p. 799\)](#).

By default, EBS volumes that are created and attached to an instance at launch are deleted when that instance is terminated. You can modify this behavior by changing the value of the flag `DeleteOnTermination` to `false` when you launch the instance. This modified value causes the volume to persist even after the instance is terminated, and enables you to attach the volume to another instance.

- **Data encryption**

For simplified data encryption, you can create encrypted EBS volumes with the Amazon EBS encryption feature. All EBS volume types support encryption. You can use encrypted EBS volumes

to meet a wide range of data-at-rest encryption requirements for regulated/audited data and applications. Amazon EBS encryption uses 256-bit Advanced Encryption Standard algorithms (AES-256) and an Amazon-managed key infrastructure. The encryption occurs on the server that hosts the EC2 instance, providing encryption of data-in-transit from the EC2 instance to Amazon EBS storage. For more information, see [Amazon EBS Encryption \(p. 799\)](#).

Amazon EBS encryption uses AWS Key Management Service (AWS KMS) master keys when creating encrypted volumes and any snapshots created from your encrypted volumes. The first time you create an encrypted EBS volume in a region, a default master key is created for you automatically. This key is used for Amazon EBS encryption unless you select a customer master key (CMK) that you created separately using AWS KMS. Creating your own CMK gives you more flexibility, including the ability to create, rotate, disable, define access controls, and audit the encryption keys used to protect your data. For more information, see the [AWS Key Management Service Developer Guide](#).

- **Snapshots**

Amazon EBS provides the ability to create snapshots (backups) of any EBS volume and write a copy of the data in the volume to Amazon S3, where it is stored redundantly in multiple Availability Zones. The volume does not need to be attached to a running instance in order to take a snapshot. As you continue to write data to a volume, you can periodically create a snapshot of the volume to use as a baseline for new volumes. These snapshots can be used to create multiple new EBS volumes, expand the size of a volume, or move volumes across Availability Zones. Snapshots of encrypted EBS volumes are automatically encrypted.

When you create a new volume from a snapshot, it's an exact copy of the original volume at the time the snapshot was taken. EBS volumes that are restored from encrypted snapshots are automatically encrypted. By optionally specifying a different volume size or a different Availability Zone, you can use this functionality to increase the size of an existing volume or to create duplicate volumes in new Availability Zones. The snapshots can be shared with specific AWS accounts or made public. When you create snapshots, you incur charges in Amazon S3 based on the volume's total size. For a successive snapshot of the volume, you are only charged for any additional data beyond the volume's original size.

Snapshots are incremental backups, meaning that only the blocks on the volume that have changed after your most recent snapshot are saved. If you have a volume with 100 GiB of data, but only 5 GiB of data have changed since your last snapshot, only the 5 GiB of modified data is written to Amazon S3. Even though snapshots are saved incrementally, the snapshot deletion process is designed so that you need to retain only the most recent snapshot in order to restore the volume.

To help categorize and manage your volumes and snapshots, you can tag them with metadata of your choice. For more information, see [Tagging Your Amazon EC2 Resources \(p. 859\)](#).

Amazon EBS Volume Types

Amazon EBS provides the following volume types, which differ in performance characteristics and price, so that you can tailor your storage performance and cost to the needs of your applications. The volumes types fall into two categories:

- SSD-backed volumes optimized for transactional workloads involving frequent read/write operations with small I/O size, where the dominant performance attribute is IOPS
- HDD-backed volumes optimized for large streaming workloads where throughput (measured in MiB/s) is a better performance measure than IOPS

The following table describes the use cases and performance characteristics for each volume type:

Amazon Elastic Compute Cloud
User Guide for Windows Instances
EBS Volumes

	Solid-State Drives (SSD)		Hard disk Drives (HDD)	
Volume Type	General Purpose SSD (gp2)*	Provisioned IOPS SSD (io1)	Throughput Optimized HDD (st1)	Cold HDD (sc1)
Description	General purpose SSD volume that balances price and performance for a wide variety of transactional workloads	Highest-performance SSD volume designed for mission-critical applications	Low cost HDD volume designed for frequently accessed, throughput-intensive workloads	Lowest cost HDD volume designed for less frequently accessed workloads
Use Cases	<ul style="list-style-type: none"> Recommended for most workloads System boot volumes Virtual desktops Low-latency interactive apps Development and test environments 	<ul style="list-style-type: none"> Critical business applications that require sustained IOPS performance, or more than 10,000 IOPS or 160 MiB/s of throughput per volume Large database workloads, such as: <ul style="list-style-type: none"> MongoDB Cassandra Microsoft SQL Server MySQL PostgreSQL Oracle 	<ul style="list-style-type: none"> Streaming workloads requiring consistent, fast throughput at a low price Big data Data warehouses Log processing Cannot be a boot volume 	<ul style="list-style-type: none"> Throughput-oriented storage for large volumes of data that is infrequently accessed Scenarios where the lowest storage cost is important Cannot be a boot volume
API Name	gp2	io1	st1	sc1
Volume Size	1 GiB - 16 TiB	4 GiB - 16 TiB	500 GiB - 16 TiB	500 GiB - 16 TiB
Max. IOPS**/ Volume	10,000	20,000	500	250
Max. Throughput/ Volume†	160 MiB/s	320 MiB/s	500 MiB/s	250 MiB/s
Max. IOPS/ Instance	65,000	65,000	65,000	65,000
Max. Throughput/ Instance	1,250 MiB/s	1,250 MiB/s	1,250 MiB/s	1,250 MiB/s
Dominant Performance Attribute	IOPS	IOPS	MiB/s	MiB/s

*Default volume type

**gp2/io1 based on 16KiB I/O size, st1/sc1 based on 1 MiB I/O size

† To achieve this throughput, you must have an instance that supports it, such as `r3.8xlarge` or `x1.32xlarge`.

The following table describes previous-generation EBS volume types. If you need higher performance or performance consistency than previous-generation volumes can provide, we recommend that you consider using General Purpose SSD (gp2) or other current volume types. For more information, see [Previous Generation Volumes](#).

Previous Generation Volumes	
Volume Type	EBS Magnetic
Description	Previous generation HDD
Use Cases	Workloads where data is infrequently accessed
API Name	standard
Volume Size	1 GiB-1 TiB
Max. IOPS/Volume	40-200
Max. Throughput/Volume	40-90 MiB/s
Max. IOPS/Instance	48,000
Max. Throughput/Instance	1,250 MiB/s
Dominant Performance Attribute	IOPS

Note

The following Amazon EBS volume considerations apply to Windows boot volumes:

- Windows 2003 instances do not boot if the boot volume is 2 TiB (2048 GiB) or greater.
- Windows boot volumes must use an MBR partition table, which limits the usable space to 2 TiB, regardless of volume size.
- Windows boot volumes of 2 TiB (2048 GiB) that have been converted to use a dynamic MBR partition table display an error when examined with Disk Manager.

The following Amazon EBS volume considerations apply to Windows data (non-boot) volumes:

- Windows volumes 2 TiB (2048 GiB) or greater must use a GPT partition table to access the entire volume.
- Amazon EBS volumes over 2048 GiB that are attached to Windows instances at launch are automatically formatted with a GPT partition table.
- Amazon EBS volumes attached to Windows instances after launch must be manually initialized with a GPT partition table. For more information, see [Making an Amazon EBS Volume Available for Use](#).

There are several factors that can affect the performance of EBS volumes, such as instance configuration, I/O characteristics, and workload demand. For more information about getting the most out of your EBS volumes, see [Amazon EBS Volume Performance on Windows Instances \(p. 803\)](#).

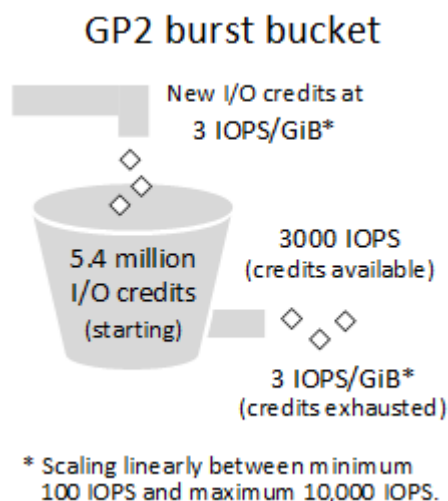
For more information about pricing for these volume types, see [Amazon EBS Pricing](#).

General Purpose SSD (gp2) Volumes

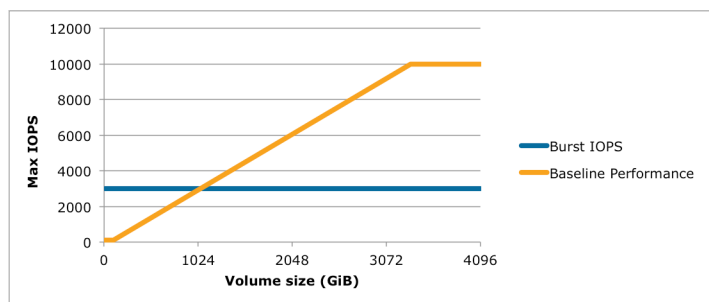
General Purpose SSD (gp2) volumes offer cost-effective storage that is ideal for a broad range of workloads. These volumes deliver single-digit millisecond latencies and the ability to burst to 3,000 IOPS for extended periods of time. Between a minimum of 100 IOPS (at 33.33 GiB and below) and a maximum of 10,000 IOPS (at 3,334 GiB and above), baseline performance scales linearly at 3 IOPS per GiB of volume size. A gp2 volume can range in size from 1 GiB to 16 TiB.

I/O Credits and Burst Performance

The performance of gp2 volumes is tied to volume size, which determines the baseline performance level of the volume and how quickly it accumulates I/O credits; larger volumes have higher baseline performance levels and accumulate I/O credits faster. I/O credits represent the available bandwidth that your gp2 volume can use to burst large amounts of I/O when more than the baseline performance is needed. The more credits your volume has for I/O, the more time it can burst beyond its baseline performance level and the better it performs when more performance is needed. The following diagram shows the burst-bucket behavior for gp2.



Each volume receives an initial I/O credit balance of 5.4 million I/O credits, which is enough to sustain the maximum burst performance of 3,000 IOPS for 30 minutes. This initial credit balance is designed to provide a fast initial boot cycle for boot volumes and to provide a good bootstrapping experience for other applications. Volumes earn I/O credits at the baseline performance rate of 3 IOPS per GiB of volume size. For example, a 100 GiB gp2 volume has a baseline performance of 300 IOPS.



When your volume requires more than the baseline performance I/O level, it draws on I/O credits in the credit balance to burst to the required performance level, up to a maximum of 3,000 IOPS. Volumes larger than 1,000 GiB have a baseline performance that is equal or greater than the maximum burst

performance, and their I/O credit balance never depletes. When your volume uses fewer I/O credits than it earns in a second, unused I/O credits are added to the I/O credit balance. The maximum I/O credit balance for a volume is equal to the initial credit balance (5.4 million I/O credits).

The following table lists several volume sizes and the associated baseline performance of the volume (which is also the rate at which it accumulates I/O credits), the burst duration at the 3,000 IOPS maximum (when starting with a full credit balance), and the time in seconds that the volume would take to refill an empty credit balance.

Volume size (GiB)	Baseline performance (IOPS)	Maximum burst duration @ 3,000 IOPS (seconds)	Seconds to fill empty credit balance
1	100	1862	54,000
100	300	2,000	18,000
214 (Min. size for max. throughput)	642	2,290	8,412
250	750	2,400	7,200
500	1,500	3,600	3,600
750	2,250	7,200	2,400
1,000	3,000	N/A*	N/A*
3,334 (Min. size for max. IOPS)	10,000	N/A*	N/A*
16,384 (16 TiB, max. volume size)	10,000	N/A*	N/A*

* Bursting and I/O credits are only relevant to volumes under 1,000 GiB, where burst performance exceeds baseline performance.

The burst duration of a volume is dependent on the size of the volume, the burst IOPS required, and the credit balance when the burst begins. This is shown in the following equation:

$$\text{Burst duration} = \frac{(\text{Credit balance})}{(\text{Burst IOPS}) - 3(\text{Volume size in GiB})}$$

What happens if I empty my I/O credit balance?

If your `gp2` volume uses all of its I/O credit balance, the maximum IOPS performance of the volume will remain at the baseline IOPS performance level (the rate at which your volume earns credits) and the volume's maximum throughput is reduced to the baseline IOPS multiplied by the maximum I/O size. Throughput can never exceed 160 MiB/s. When I/O demand drops below the baseline level and unused credits are added to the I/O credit balance, the maximum IOPS performance of the volume will again exceed the baseline. For example, a 100 GiB `gp2` volume with an empty credit balance has a baseline performance of 300 IOPS and a throughput limit of 75 MiB/s (300 I/O operations per second * 256 KiB per I/O operation = 75 MiB/s). The larger a volume is, the greater the baseline performance is and the faster it replenishes the credit balance. For more information about how IOPS are measured, see [I/O Characteristics](#).

If you notice that your volume performance is frequently limited to the baseline level (due to an empty I/O credit balance), you should consider using a larger `gp2` volume (with a higher baseline performance

level) or switching to an `io1` volume for workloads that require sustained IOPS performance greater than 10,000 IOPS.

For information about using CloudWatch metrics and alarms to monitor your burst bucket balance, see [Monitoring the Burst Bucket Balance for `gp2`, `st1`, and `sc1` Volumes](#) (p. 761).

Throughput Performance

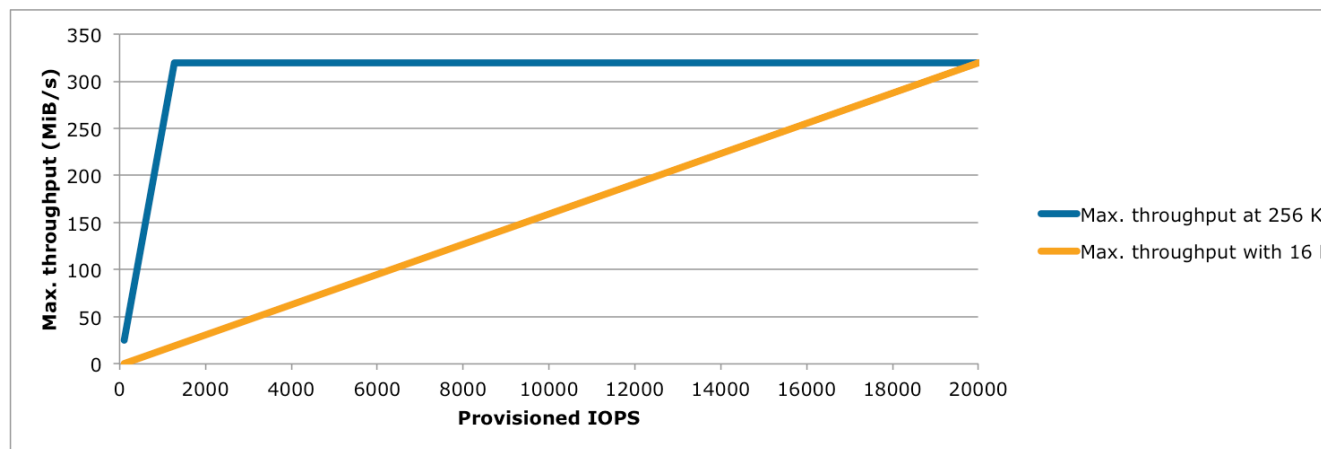
The throughput limit for `gp2` volumes is 128 MiB/s for volumes less than or equal to 170 GiB and 160 MiB/s for volumes over 170 GiB.

Provisioned IOPS SSD (`io1`) Volumes

Provisioned IOPS SSD (`io1`) volumes are designed to meet the needs of I/O-intensive workloads, particularly database workloads, that are sensitive to storage performance and consistency. Instead of using a bucket and credit model to calculate performance, an `io1` volume allows you to specify a consistent IOPS rate when you create the volume, and Amazon EBS delivers within 10 percent of the provisioned IOPS performance 99.9 percent of the time over a given year.

An `io1` volume can range in size from 4 GiB to 16 TiB and you can provision up to 20,000 IOPS per volume. The maximum ratio of provisioned IOPS to requested volume size (in GiB) is 50:1. For example, a 100 GiB volume can be provisioned with up to 5,000 IOPS. Any volume 400 GiB in size or greater allows provisioning up to the 20,000 IOPS maximum.

The throughput limit of `io1` volumes is 256 KiB for each IOPS provisioned, up to a maximum of 320 MiB/s (at 1,280 IOPS).



Your per-I/O latency experience depends on the IOPS provisioned and your workload pattern. For the best per-I/O latency experience, we recommend that you provision an IOPS-to-GiB ratio greater than 2:1. For example, a 2,000 IOPS volume should be smaller than 1,000 GiB.

Note

Some AWS accounts created before 2012 might have access to Availability Zones in `us-east-1`, `us-west-1`, or `ap-northeast-1` that do not support Provisioned IOPS SSD (`io1`) volumes. If you are unable to create an `io1` volume (or launch an instance with an `io1` volume in its block device mapping) in one of these regions, try a different Availability Zone in the region. You can verify that an Availability Zone supports `io1` volumes by creating a 4 GiB `io1` volume in that zone.

Throughput Optimized HDD (`st1`) Volumes

Throughput Optimized HDD (`st1`) volumes provide low-cost magnetic storage that defines performance in terms of throughput rather than IOPS. This volume type is a good fit for large,

sequential workloads such as Amazon EMR, ETL, data warehouses, and log processing. Bootable `st1` volumes are not supported.

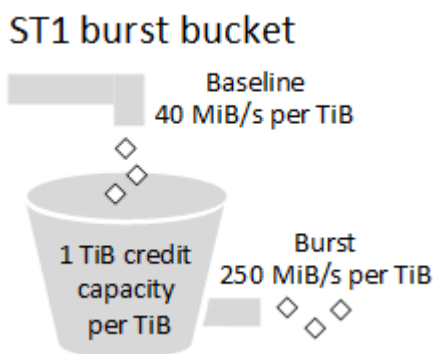
Note

This volume type is optimized for workloads involving large, sequential I/O, and we recommend that customers with workloads performing small, random I/O use `gp2`. For more information, see [Inefficiency of Small Read/Writes on HDD](#) (p. 760).

Throughput Credits and Burst Performance

Like `gp2`, `st1` uses a burst-bucket model for performance. Volume size determines the baseline throughput of your volume, which is the rate at which the volume accumulates throughput credits. Volume size also determines the burst throughput of your volume, which is the rate at which you can spend credits when they are available. Larger volumes have higher baseline and burst throughput. The more credits your volume has, the longer it will be able to drive I/O at the burst level.

The following diagram shows the burst-bucket behavior for `st1`.



Subject to throughput and throughput-credit caps, the available throughput of an `st1` volume is expressed by the following formula:

$$\text{(Volume size)} \times \text{(Credit accumulation rate per TiB)} = \text{Baseline Throughput}$$

For a 1 TiB `st1` volume, burst throughput is limited to 250 MiB/s, the bucket fills with credits at 40 MiB/s, and it can hold up to 1 TiB-worth of credits.

Larger volumes scale these limits linearly, with throughput capped at a maximum of 500 MiB/s. After the bucket is depleted, throughput is limited to the baseline rate of 40 MiB/s per TiB.

On volume sizes ranging from 0.5 to 16 TiB, baseline throughput varies from 20 to a cap of 500 MiB/s, which is reached at 12.5 TiB because

$$12.5 \text{ TiB} \times \frac{40 \text{ MiB/s}}{1 \text{ TiB}} = 500 \text{ MiB/s}$$

Burst throughput varies from 125 MiB/s to a cap of 500 MiB/s, which is reached at 2 TiB because

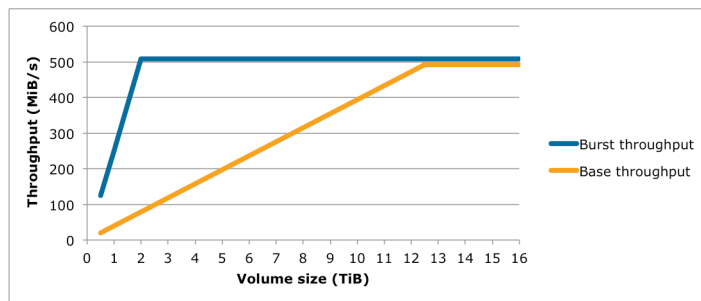
$$2 \text{ TiB} \times \frac{250 \text{ MiB/s}}{1 \text{ TiB}} = 500 \text{ MiB/s}$$

1 TiB

The following table states the full range of base and burst throughput values for `st1`:

Volume Size (TiB)	ST1 Base Throughput (MiB/s)	ST1 Burst Throughput (MiB/s)
0.5	20	125
1	40	250
2	80	500
3	120	500
4	160	500
5	200	500
6	240	500
7	280	500
8	320	500
9	360	500
10	400	500
11	440	500
12	480	500
12.5	500	500
13	500	500
14	500	500
15	500	500
16	500	500

The following diagram plots the table values:



Note

Throughput for an `st1` volume is also capped at the baseline while a snapshot is being created.

For information about using CloudWatch metrics and alarms to monitor your burst bucket balance, see [Monitoring the Burst Bucket Balance for gp2, st1, and sc1 Volumes \(p. 761\)](#).

Cold HDD (sc1) Volumes

Cold HDD (sc1) volumes provide low-cost magnetic storage that defines performance in terms of throughput rather than IOPS. With a lower throughput limit than st1, sc1 is a good fit ideal for large, sequential cold-data workloads. If you require infrequent access to your data and are looking to save costs, sc1 provides inexpensive block storage. Bootable sc1 volumes are not supported.

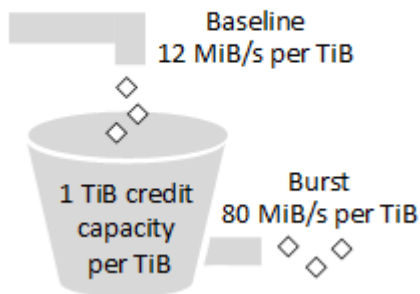
Note

This volume type is optimized for workloads involving large, sequential I/O, and we recommend that customers with workloads performing small, random I/O use gp2. For more information, see [Inefficiency of Small Read/Writes on HDD \(p. 760\)](#).

Throughput Credits and Burst Performance

Like gp2, sc1 uses a burst-bucket model for performance. Volume size determines the baseline throughput of your volume, which is the rate at which the volume accumulates throughput credits. Volume size also determines the burst throughput of your volume, which is the rate at which you can spend credits when they are available. Larger volumes have higher baseline and burst throughput. The more credits your volume has, the longer it will be able to drive I/O at the burst level.

SC1 burst bucket



Subject to throughput and throughput-credit caps, the available throughput of an sc1 volume is expressed by the following formula:

$$(\text{Volume size}) \times (\text{Credit accumulation rate per TiB}) = \text{Baseline Throughput}$$

For a 1 TiB sc1 volume, burst throughput is limited to 80 MiB/s, the bucket fills with credits at 12 MiB/s, and it can hold up to 1 TiB-worth of credits.

Larger volumes scale these limits linearly, with throughput capped at a maximum of 250 MiB/s. After the bucket is depleted, throughput is limited to the baseline rate of 12 MiB/s per TiB.

On volume sizes ranging from 0.5 to 16 TiB, baseline throughput varies from 6 MiB/s to a maximum of 192 MiB/s, which is reached at 16 TiB because

$$16 \text{ TiB} \times \frac{12 \text{ MiB/s}}{1 \text{ TiB}} = 192 \text{ MiB/s}$$

Burst throughput varies from 40 MiB/s to a cap of 250 MiB/s, which is reached at 3.125 TiB because

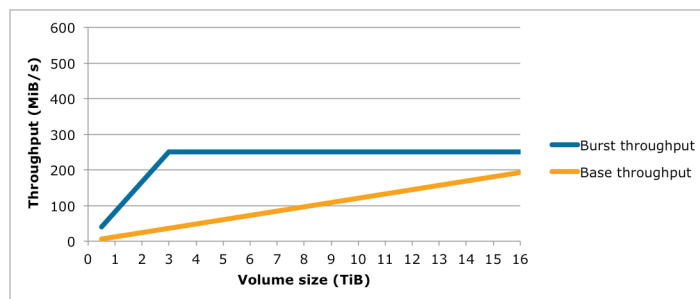
$$250 \text{ MiB/s}$$

$$3.125 \text{ TiB} \times \frac{\text{-----}}{1 \text{ TiB}} = 250 \text{ MiB/s}$$

The following table states the full range of base and burst throughput values for `sc1`:

Volume Size (TiB)	SC1 Base Throughput (MiB/s)	SC1 Burst Throughput (MiB/s)
0.5	6	40
1	12	80
2	24	160
3	36	240
3.125	37.5	250
4	48	250
5	60	250
6	72	250
7	84	250
8	96	250
9	108	250
10	120	250
11	132	250
12	144	250
13	156	250
14	168	250
15	180	250
16	192	250

The following diagram plots the table values:



Magnetic (standard)

Magnetic volumes are backed by magnetic drives and are suited for workloads where data is accessed infrequently, and scenarios where low-cost storage for small volume sizes is important. These volumes

deliver approximately 100 IOPS on average, with burst capability of up to hundreds of IOPS, and they can range in size from 1 GiB to 1 TiB.

Note

Magnetic is a Previous Generation Volume. For new applications, we recommend using one of the newer volume types. For more information, see [Previous Generation Volumes](#).

For information about using CloudWatch metrics and alarms to monitor your burst bucket balance, see [Monitoring the Burst Bucket Balance for gp2, st1, and sc1 Volumes \(p. 761\)](#).

Performance Considerations When Using HDD Volumes

For optimal throughput results using HDD volumes, plan your workloads with the following considerations in mind.

Throughput Optimized HDD vs. Cold HDD

The `st1` and `sc1` bucket sizes vary according to volume size, and a full bucket contains enough tokens for a full volume scan. However, larger `st1` and `sc1` volumes take longer for the volume scan to complete due to per-instance and per-volume throughput limits. Volumes attached to smaller instances are limited to the per-instance throughput rather than the `st1` or `sc1` throughput limits.

Both `st1` and `sc1` are designed for performance consistency of 90% of burst throughput 99% of the time. Non-compliant periods are approximately uniformly distributed, targeting 99% of expected total throughput each hour.

The following table shows ideal scan times for volumes of various size, assuming full buckets and sufficient instance throughput.

In general, scan times are expressed by this formula:

$$\frac{\text{Volume size}}{\text{Throughput}} = \text{Scan time}$$

For example, taking the performance consistency guarantees and other optimizations into account, an `st1` customer with a 5 TiB volume can expect to complete a full volume scan in 2.91 to 3.27 hours.

$$\frac{5 \text{ TiB}}{500 \text{ MiB/s}} = \frac{5 \text{ TiB}}{0.00047684 \text{ TiB/s}} = 10,486 \text{ s} = 2.91 \text{ hours (optimal)}$$

$$2.91 \text{ hours} + \frac{2.91 \text{ hours}}{(0.90)(0.99)} = 3.27 \text{ hours (minimum expected)}$$

(0.90)(0.99) <-- From expected performance of 90% of burst 99% of the time

Similarly, an `sc1` customer with a 5 TiB volume can expect to complete a full volume scan in 5.83 to 6.54 hours.

$$\frac{5 \text{ TiB}}{0.000238418 \text{ TiB/s}} = 20972 \text{ s} = 5.83 \text{ hours (optimal)}$$

$$5.83 \text{ hours} + \frac{5.83 \text{ hours}}{(0.90)(0.99)} = 6.54 \text{ hours (minimum expected)}$$

Volume Size (TiB)	ST1 Scan Time with Burst (Hours)*	SC1 Scan Time with Burst (Hours)*
1	1.17	3.64
2	1.17	3.64
3	1.75	3.64
4	2.33	4.66
5	2.91	5.83
6	3.50	6.99
7	4.08	8.16
8	4.66	9.32
9	5.24	10.49
10	5.83	11.65
11	6.41	12.82
12	6.99	13.98
13	7.57	15.15
14	8.16	16.31
15	8.74	17.48
16	9.32	18.64

* These scan times assume an average queue depth (rounded to the nearest whole number) of four or more when performing 1 MiB of sequential I/O.

Therefore if you have a throughput-oriented workload that needs to complete scans quickly (up to 500 MiB/s), or requires several full volume scans a day, use `st1`. If you are optimizing for cost, your data is relatively infrequently accessed, and you don't need more than 250 MiB/s of scanning performance, then use `sc1`.

Inefficiency of Small Read/Writes on HDD

The performance model for `st1` and `sc1` volumes is optimized for sequential I/Os, favoring high-throughput workloads, offering acceptable performance on workloads with mixed IOPS and throughput, and discouraging workloads with small, random I/O.

For example, an I/O request of 1 MiB or less counts as a 1 MiB I/O credit. However, if the I/Os are sequential, they are merged into 1 MiB I/O blocks and count only as a 1 MiB I/O credit.

Limitations on per-Instance Throughput

Throughput for `st1` and `sc1` volumes will always be determined by the smaller of the following:

- Throughput limits of the volume
- Throughput limits of the instance

As for all Amazon EBS volumes, we recommend that you select an appropriate EBS-optimized EC2 instance in order to avoid network bottlenecks. For more information, see [Amazon EBS-Optimized Instances](#).

Monitoring the Burst Bucket Balance for `gp2`, `st1`, and `sc1` Volumes

You can monitor the burst-bucket level for `gp2`, `st1`, and `sc1` volumes using the EBS `BurstBalance` metric available in Amazon CloudWatch. This metric shows the percentage of I/O credits (for `gp2`) or throughput credits (for `st1` and `sc1`) remaining in the burst bucket. For more information about the `BurstBalance` metric and other metrics related to I/O, see [I/O Characteristics and Monitoring](#). CloudWatch also allows you to set an alarm that notifies you when the `BurstBalance` value falls to a certain level. For more information about CloudWatch alarms, see [Creating Amazon CloudWatch Alarms](#).

Creating an Amazon EBS Volume

You can create an Amazon EBS volume that you can then attach to any EC2 instance within the same Availability Zone. You can choose to create an encrypted EBS volume, but encrypted volumes can only be attached to selected instance types. For more information, see [Supported Instance Types](#) (p. 800).

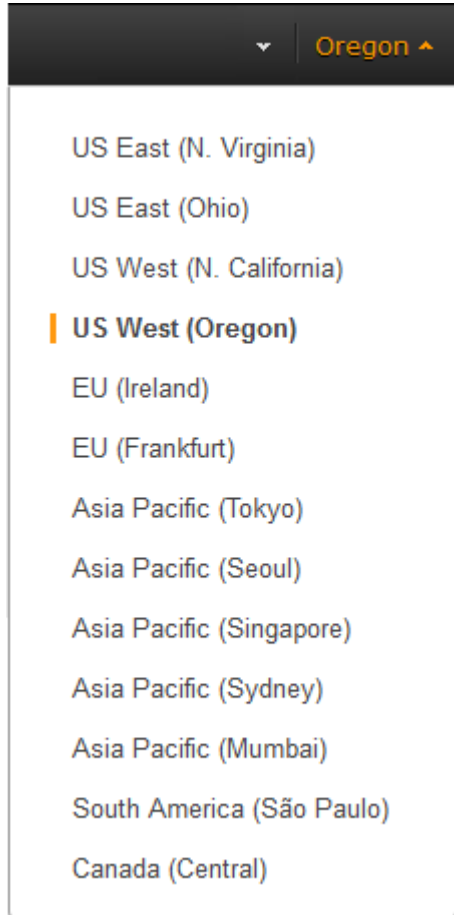
You can also create and attach EBS volumes when you launch instances by specifying a block device mapping. For more information, see [Launching an Instance](#) (p. 244) and [Block Device Mapping](#) (p. 833). You can restore volumes from previously created snapshots. For more information, see [Restoring an Amazon EBS Volume from a Snapshot](#) (p. 763).

If you are creating a volume for a high-performance storage scenario, you should make sure to use a Provisioned IOPS SSD (`io1`) volume and attach it to an instance with enough bandwidth to support your application, such as an EBS-optimized instance or an instance with 10 Gigabit network connectivity. The same advice holds for Throughput Optimized HDD (`st1`) and Cold HDD (`sc1`) volumes. For more information, see [Amazon EC2 Instance Configuration](#) (p. 805).

New EBS volumes receive their maximum performance the moment that they are available and do not require initialization (formerly known as pre-warming). However, storage blocks on volumes that were restored from snapshots must be initialized (pulled down from Amazon S3 and written to the volume) before you can access the block. This preliminary action takes time and can cause a significant increase in the latency of an I/O operation the first time each block is accessed. For most applications, amortizing this cost over the lifetime of the volume is acceptable. Performance is restored after the data is accessed once. For more information, see [Initializing Amazon EBS Volumes](#) (p. 810).

To create an EBS volume using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select the region in which you would like to create your volume. This choice is important because some Amazon EC2 resources can be shared between regions, while others can't. For more information, see [Resource Locations](#) (p. 851).



3. In the navigation pane, under **ELASTIC BLOCK STORE**, choose **Volumes**.
4. Above the upper pane, choose **Create Volume**.
5. In the **Create Volume** dialog box, for **Volume Type**, choose **General Purpose SSD, Provisioned IOPS SSD**, or **Magnetic**. For more information, see [Amazon EBS Volume Types \(p. 749\)](#).

Note

Some AWS accounts created before 2012 might have access to Availability Zones in us-east-1, us-west-1, or ap-northeast-1 that do not support Provisioned IOPS SSD (i_o1) volumes. If you are unable to create an i_o1 volume (or launch an instance with an i_o1 volume in its block device mapping) in one of these regions, try a different Availability Zone in the region. You can verify that an Availability Zone supports i_o1 volumes by creating a 4 GiB i_o1 volume in that zone.

6. For **Size**, enter the size of the volume, in GiB.

Note

The following Amazon EBS volume considerations apply to Windows boot volumes:

- Windows 2003 instances do not boot if the boot volume is 2 TiB (2048 GiB) or greater.
- Windows boot volumes must use an MBR partition table, which limits the usable space to 2 TiB, regardless of volume size.
- Windows boot volumes of 2 TiB (2048 GiB) that have been converted to use a dynamic MBR partition table display an error when examined with Disk Manager.

The following Amazon EBS volume considerations apply to Windows data (non-boot) volumes:

- Windows volumes 2 TiB (2048 GiB) or greater must use a GPT partition table to access the entire volume.
 - Amazon EBS volumes over 2048 GiB that are attached to Windows instances at launch are automatically formatted with a GPT partition table.
 - Amazon EBS volumes attached to Windows instances after launch must be manually initialized with a GPT partition table. For more information, see [Making an Amazon EBS Volume Available for Use](#).
7. For `io1` volumes, in the **IOPS** field, enter the maximum number of input/output operations per second (IOPS) that the volume should support.
 8. For **Availability Zone**, select the Availability Zone in which to create the volume.
 9. (Optional) To create an encrypted volume, select the **Encrypted** box and choose the master key you want to use when encrypting the volume. You can choose the default master key for your account, or you can choose any customer master key (CMK) that you have previously created using the AWS Key Management Service. Available keys are visible in the **Master Key** menu, or you can paste the full ARN of any key that you have access to. For more information, see the [AWS Key Management Service Developer Guide](#).

Note

Encrypted volumes can only be attached to selected instance types. For more information, see [Supported Instance Types \(p. 800\)](#).

10. Choose **Yes, Create**.

Important

If you receive one of the following errors, the current volume creation would exceed the default storage limit for your account:

```
Maximum number of active volumes bytes, 20, exceeded.  
Maximum number of active gp2 volumes bytes, 20, exceeded.  
Maximum number of active io1 volumes bytes, 20, exceeded.
```

To view the default service limits for Amazon EBS, see [Amazon Elastic Block Store \(Amazon EBS\) Limits](#) in the *Amazon Web Services General Reference*. To request an increase in your storage limits, see [Request to Increase the Amazon EBS Volume Limit](#).

To create an EBS volume using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [create-volume](#) (AWS CLI)
- [New-EC2Volume](#) (AWS Tools for Windows PowerShell)

Restoring an Amazon EBS Volume from a Snapshot

You can restore an Amazon EBS volume with data from a snapshot stored in Amazon S3. You need to know the ID of the snapshot you wish to restore your volume from and you need to have access permissions for the snapshot. For more information on snapshots, see [Amazon EBS Snapshots \(p. 788\)](#).

New volumes created from existing EBS snapshots load lazily in the background. This means that after a volume is created from a snapshot, there is no need to wait for all of the data to transfer from Amazon S3 to your EBS volume before your attached instance can start accessing the volume and all its data. If your instance accesses data that hasn't yet been loaded, the volume immediately downloads the requested data from Amazon S3, and continues loading the rest of the data in the background.

EBS volumes that are restored from encrypted snapshots are automatically encrypted. Encrypted volumes can only be attached to selected instance types. For more information, see [Supported Instance Types \(p. 800\)](#).

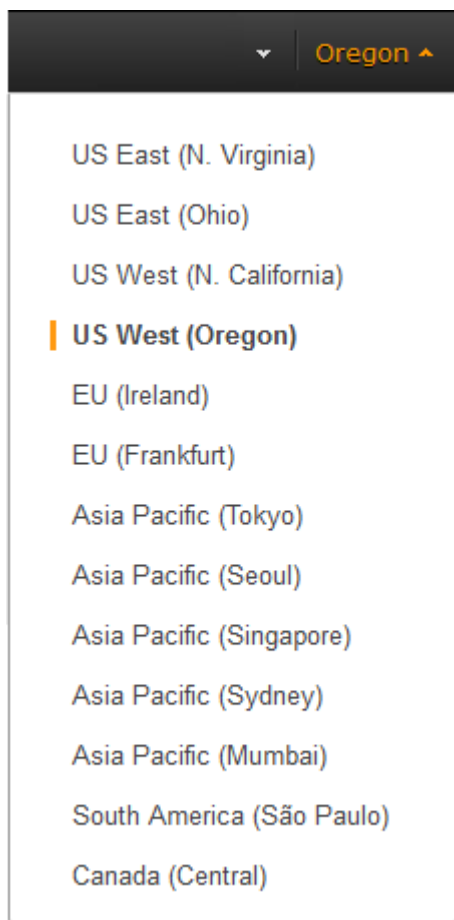
Because of security constraints, you cannot directly restore an EBS volume from a shared encrypted snapshot that you do not own. You must first create a copy of the snapshot, which you will own. You can then restore a volume from that copy. For more information, see [Amazon EBS Encryption](#).

New EBS volumes receive their maximum performance the moment that they are available and do not require initialization (formerly known as pre-warming). However, storage blocks on volumes that were restored from snapshots must be initialized (pulled down from Amazon S3 and written to the volume) before you can access the block. This preliminary action takes time and can cause a significant increase in the latency of an I/O operation the first time each block is accessed. Performance is restored after the data is accessed once.

For most applications, amortizing the initialization cost over the lifetime of the volume is acceptable. If you need to ensure that your restored volume always functions at peak capacity in production, you can force the immediate initialization of the entire volume using **dd** or **fiio**. For more information, see [Initializing Amazon EBS Volumes \(p. 810\)](#).

To restore an EBS volume from a snapshot using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select the region that your snapshot is located in. This choice is important because some Amazon EC2 resources can be shared between regions, while others can't. For more information, see [Resource Locations \(p. 851\)](#). If you need to restore the snapshot to a volume in a different region, you can copy your snapshot to the new region and then restore it to a volume in that region. For more information, see [Copying an Amazon EBS Snapshot \(p. 791\)](#).



3. In the navigation pane, choose **Volumes, Create Volume**.
4. In the **Create Volume** dialog box, for **Volume Type**, choose **General Purpose SSD, Provisioned IOPS SSD**, or **Magnetic**. For more information, see [Amazon EBS Volume Types \(p. 749\)](#).

Note

Some AWS accounts created before 2012 might have access to Availability Zones in us-east-1, us-west-1, or ap-northeast-1 that do not support Provisioned IOPS SSD (io1) volumes. If you are unable to create an io1 volume (or launch an instance with an io1 volume in its block device mapping) in one of these regions, try a different Availability Zone in the region. You can verify that an Availability Zone supports io1 volumes by creating a 4 GiB io1 volume in that zone.

5. For **Snapshot**, start typing the ID or description of the snapshot from which you are restoring the volume, and select it from the list of suggested options.

Note

Volumes that are restored from encrypted snapshots can only be attached to instances that support Amazon EBS encryption. For more information, see [Supported Instance Types \(p. 800\)](#).

6. For **Size**, enter the size of the volume in GiB, or verify that the default size of the snapshot is adequate.

If you specify both a volume size and a snapshot ID, the size must be equal to or greater than the snapshot size. When you select a volume type and a snapshot ID, minimum and maximum sizes for the volume are shown next to the **Size** list. Any AWS Marketplace product codes from the snapshot are propagated to the volume.

Note

The following Amazon EBS volume considerations apply to Windows boot volumes:

- Windows 2003 instances do not boot if the boot volume is 2 TiB (2048 GiB) or greater.
- Windows boot volumes must use an MBR partition table, which limits the usable space to 2 TiB, regardless of volume size.
- Windows boot volumes of 2 TiB (2048 GiB) that have been converted to use a dynamic MBR partition table display an error when examined with Disk Manager.

The following Amazon EBS volume considerations apply to Windows data (non-boot) volumes:

- Windows volumes 2 TiB (2048 GiB) or greater must use a GPT partition table to access the entire volume.
 - Amazon EBS volumes over 2048 GiB that are attached to Windows instances at launch are automatically formatted with a GPT partition table.
 - Amazon EBS volumes attached to Windows instances after launch must be manually initialized with a GPT partition table. For more information, see [Making an Amazon EBS Volume Available for Use](#).
7. For `io1` volumes, in the **IOPS** field, enter the maximum number of input/output operations per second (IOPS) that the volume can support.
 8. In the **Availability Zone** list, select the Availability Zone in which to create the volume. EBS volumes can only be attached to EC2 instances within the same Availability Zone.
 9. Choose **Yes, Create**.

Important

If you restored a snapshot to a larger volume than the default for that snapshot, you need to extend the file system on the volume to take advantage of the extra space. For more information, see [Expanding the Storage Space of an EBS Volume on Windows \(p. 783\)](#).

After you've restored a volume from a snapshot, you can attach it to an instance to begin using it. For more information, see [Attaching an Amazon EBS Volume to an Instance \(p. 766\)](#).

To restore an EBS volume using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- `create-volume` (AWS CLI)
- `New-EC2Volume` (AWS Tools for Windows PowerShell)

Attaching an Amazon EBS Volume to an Instance

You can attach an EBS volumes to one of your instances that is in the same Availability Zone as the volume.

Prerequisites

- Determine the device names that you'll use. For more information, see [Device Naming on Windows Instances \(p. 832\)](#).
- Determine how many volumes you can attach to your instance. For more information, see [Instance Volume Limits \(p. 831\)](#).
- If a volume is encrypted, it can only be attached to an instance that supports Amazon EBS encryption. For more information, see [Supported Instance Types \(p. 800\)](#).

- If a volume has an AWS Marketplace product code:
 - The volume can only be attached to a stopped instance.
 - You must be subscribed to the AWS Marketplace code that is on the volume.
 - The configuration (instance type, operating system) of the instance must support that specific AWS Marketplace code. For example, you cannot take a volume from a Windows instance and attach it to a Linux instance.
 - AWS Marketplace product codes are copied from the volume to the instance.

To attach an EBS volume to an instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Volumes**.
3. Select a volume and choose **Actions, Attach Volume**.
4. In the **Attach Volume** dialog box, start typing the name or ID of the instance to attach the volume to for **Instance**, and select it from the list of suggestion options (only instances that are in the same Availability Zone as the volume are displayed).
5. You can keep the suggested device name, or enter a different supported device name.

Important

The block device driver for the instance assigns the actual volume name when mounting the volume, and the name assigned can be different from the name that Amazon EC2 recommends.

6. Choose **Attach**.
7. Connect to your instance and make the volume available. For more information, see [Making an Amazon EBS Volume Available for Use \(p. 767\)](#).

To attach an EBS volume to an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [attach-volume](#) (AWS CLI)
- [Add-EC2Volume](#) (AWS Tools for Windows PowerShell)

Making an Amazon EBS Volume Available for Use

After you attach an Amazon EBS volume to your instance, it is exposed as a block device. You can format the volume with any file system and then mount it. After you make the EBS volume available for use, you can access it in the same ways that you access any other volume. Any data written to this file system is written to the EBS volume and is transparent to applications using the device.

Note that you can take snapshots of your EBS volume for backup purposes or to use as a baseline when you create another volume. For more information, see [Amazon EBS Snapshots \(p. 788\)](#).

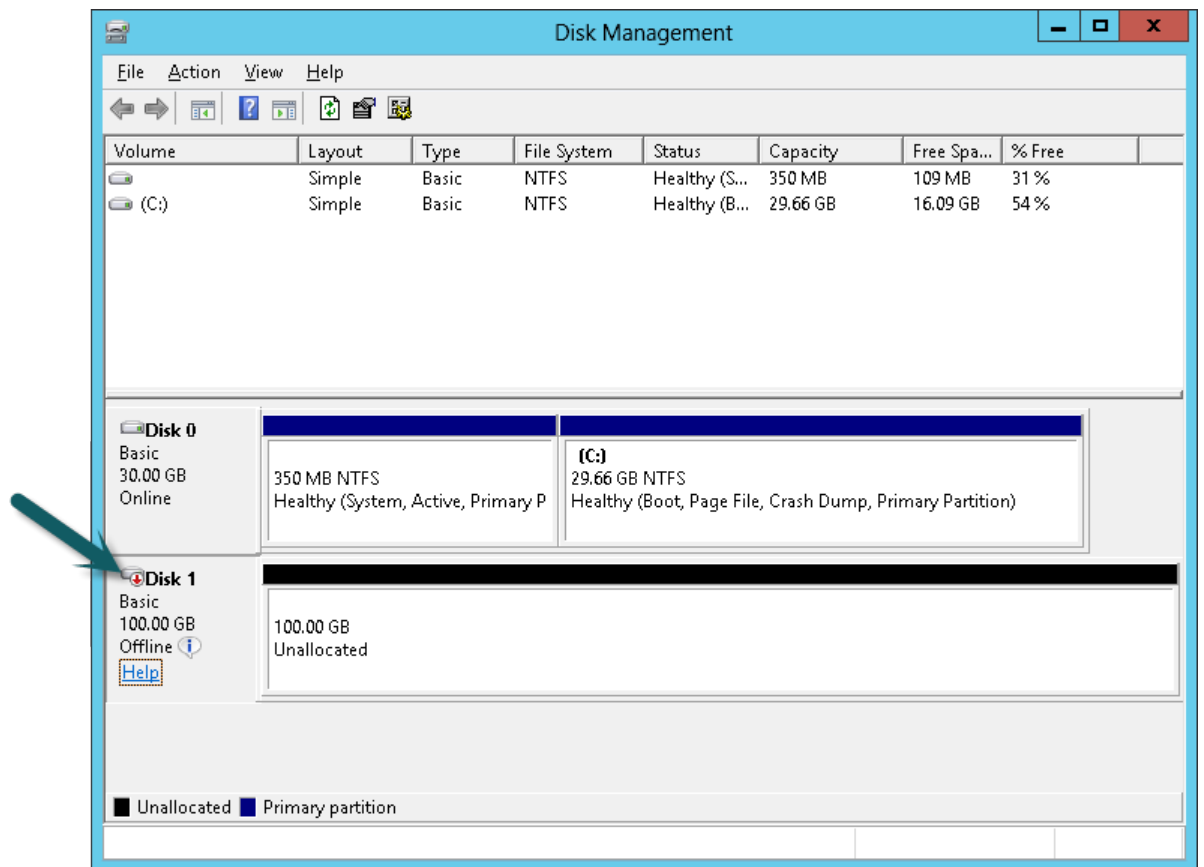
Making the Volume Available on Windows

Use the following procedure to make the volume available. Note that you can get directions for volumes on a Linux instance from [Making the Volume Available on Linux](#) in the *Amazon EC2 User Guide for Linux Instances*.

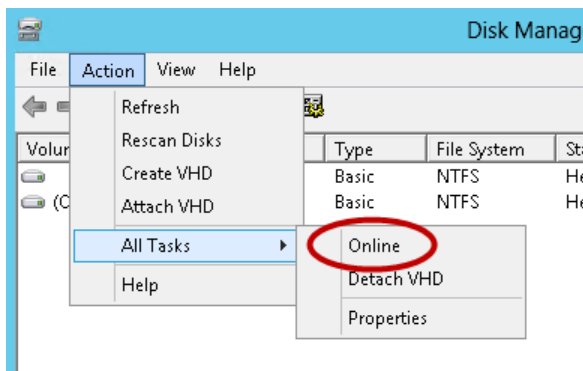
To use an EBS volume

1. Log in to your Windows instance using Remote Desktop. For more information, see, [Connecting to Your Windows Instance \(p. 254\)](#).

2. Start the Disk Management utility. On Windows Server 2012, on the taskbar, open the context (right-click) menu for the Windows logo and choose **Disk Management**. On Windows Server 2003/2008, choose **Start, Administrative Tools, Computer Management, and Disk Management**.
3. In the lower pane, select the disk that represents the new EBS volume.



4. On the **Disk Management** menu, choose **Action, All Tasks, Online**.



5. (Conditional) A new disk needs to be initialized before it can be used.

Caution

If you're mounting a volume that already has data on it (for example, a public data set, or a volume that you created from a snapshot), make sure that you don't reformat the volume and delete the existing data.

To initialize a new disk:

- a. In the Disk Management utility, select the new EBS volume disk.
- b. On the **Disk Management** menu, choose **Action, All Tasks, Initialize Disk**.
- c. In the **Initialize Disk** dialog, select the disk to initialize, select the desired partition style, and choose **OK**.

Viewing Volume Information

You can view descriptive information for your Amazon EBS volumes in a selected region at a time in the AWS Management Console. You can also view detailed information about a single volume, including the size, volume type, whether or not the volume is encrypted, which master key was used to encrypt the volume, and the specific instance to which the volume is attached.

To view information about an EBS volume using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Volumes**.
3. To view more information about a volume, select it.

To view information about an EBS volume using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [describe-volumes](#) (AWS CLI)
- [Get-EC2Volume](#) (AWS Tools for Windows PowerShell)

Monitoring the Status of Your Volumes

Amazon Web Services (AWS) automatically provides data, such as Amazon CloudWatch metrics and volume status checks, that you can use to monitor your Amazon Elastic Block Store (Amazon EBS) volumes.

Contents

- [Monitoring Volumes with CloudWatch \(p. 769\)](#)
- [Monitoring Volumes with Status Checks \(p. 772\)](#)
- [Monitoring Volume Events \(p. 774\)](#)
- [Working with an Impaired Volume \(p. 776\)](#)
- [Working with the AutoEnableIO Volume Attribute \(p. 779\)](#)

Monitoring Volumes with CloudWatch

CloudWatch metrics are statistical data that you can use to view, analyze, and set alarms on the operational behavior of your volumes.

The following table describes the types of monitoring data available for your Amazon EBS volumes.

Type	Description
Basic	Data is available automatically in 5-minute periods at no charge. This includes data for the root device volumes for EBS-backed instances.

Type	Description
Detailed	Provisioned IOPS SSD (io1) volumes automatically send one-minute metrics to CloudWatch.

When you get data from CloudWatch, you can include a `Period` request parameter to specify the granularity of the returned data. This is different than the period that we use when we collect the data (5-minute periods). We recommend that you specify a period in your request that is equal to or larger than the collection period to ensure that the returned data is valid.

You can get the data using either the CloudWatch API or the Amazon EC2 console. The console takes the raw data from the CloudWatch API and displays a series of graphs based on the data. Depending on your needs, you might prefer to use either the data from the API or the graphs in the console.

Amazon EBS Metrics

Amazon Elastic Block Store (Amazon EBS) sends data points to CloudWatch for several metrics. Amazon EBS General Purpose SSD (gp2), Throughput Optimized HDD (st1), Cold HDD (sc1), and Magnetic (standard) volumes automatically send five-minute metrics to CloudWatch. Provisioned IOPS SSD (io1) volumes automatically send one-minute metrics to CloudWatch. For more information about how to monitor Amazon EBS, see [Monitoring the Status of Your Volumes](#) in the *Amazon EC2 User Guide for Linux Instances*.

The `AWS/EBS` namespace includes the following metrics.

Metric	Description
VolumeReadBytes VolumeWriteBytes	Provides information on the I/O operations in a specified period of time. The <code>Sum</code> statistic reports the total number of bytes transferred during the period. The <code>Average</code> statistic reports the average size of each I/O operation during the period. The <code>SampleCount</code> statistic reports the total number of I/O operations during the period. The <code>Minimum</code> and <code>Maximum</code> statistics are not relevant for this metric. Data is only reported to Amazon CloudWatch when the volume is active. If the volume is idle, no data is reported to Amazon CloudWatch. Units: Bytes
VolumeReadOps VolumeWriteOps	The total number of I/O operations in a specified period of time. Note To calculate the average I/O operations per second (IOPS) for the period, divide the total operations in the period by the number of seconds in that period. Units: Count
VolumeTotalReadTime VolumeTotalWriteTime	The total number of seconds spent by all operations that completed in a specified period of time. If multiple requests are submitted at the same time, this total could be greater than the length of the period. For example, for a period of 5 minutes (300 seconds): if 700 operations completed during that period, and each operation took 1 second, the value would be 700 seconds. Units: Seconds
VolumeIdleTime	The total number of seconds in a specified period of time when no read or write operations were submitted.

Metric	Description
	Units: Seconds
VolumeQueueLength	The number of read and write operation requests waiting to be completed in a specified period of time. Units: Count
VolumeThroughputPercentage	Used with Provisioned IOPS SSD volumes only. The percentage of I/O operations per second (IOPS) delivered of the total IOPS provisioned for an Amazon EBS volume. Provisioned IOPS SSD volumes deliver within 10 percent of the provisioned IOPS performance 99.9 percent of the time over a given year. Note During a write, if there are no other pending I/O requests in a minute, the metric value will be 100 percent. Also, a volume's I/O performance may become degraded temporarily due to an action you have taken (e.g., creating a snapshot of a volume during peak usage, running the volume on a non-EBS-optimized instance, accessing data on the volume for the first time). Units: Percent
VolumeConsumedReadWriteIOPS	Used with Provisioned IOPS SSD volumes only. The total amount of read and write operations (normalized to 256K capacity units) consumed in a specified period of time. I/O operations that are smaller than 256K each count as 1 consumed IOPS. I/O operations that are larger than 256K are counted in 256K capacity units. For example, a 1024K I/O would count as 4 consumed IOPS. Units: Count
BurstBalance	Used with General Purpose SSD (gp2), Throughput Optimized HDD (st1), and Cold HDD (sc1) volumes only. Provides information about the percentage of I/O credits (for gp2) or throughput credits (for st1 and sc1) remaining in the burst bucket. Data is reported to CloudWatch only when the volume is active. If the volume is not attached, no data is reported. Units: Percent

Dimensions for Amazon EBS Metrics

The only dimension that Amazon EBS sends to CloudWatch is the volume ID. This means that all available statistics are filtered by volume ID.

Graphs in the Amazon EC2 Console

After you create a volume, you can view the volume's monitoring graphs in the Amazon EC2 console. Select a volume on the **Volumes** page in the console and choose **Monitoring**. The following table lists the graphs that are displayed. The column on the right describes how the raw data metrics from the CloudWatch API are used to produce each graph. The period for all the graphs is 5 minutes.

Graph	Description using raw metrics
Read Bandwidth (KiB/s)	Sum(VolumeReadBytes) / Period / 1024

Graph	Description using raw metrics
Write Bandwidth (KiB/s)	$\text{Sum}(\text{VolumeWriteBytes}) / \text{Period} / 1024$
Read Throughput (Ops/s)	$\text{Sum}(\text{VolumeReadOps}) / \text{Period}$
Write Throughput (Ops/s)	$\text{Sum}(\text{VolumeWriteOps}) / \text{Period}$
Avg Queue Length (ops)	$\text{Avg}(\text{VolumeQueueLength})$
% Time Spent Idle	$\text{Sum}(\text{VolumeldleTime}) / \text{Period} * 100$
Avg Read Size (KiB/op)	$\text{Avg}(\text{VolumeReadBytes}) / 1024$
Avg Write Size (KiB/op)	$\text{Avg}(\text{VolumeWriteBytes}) / 1024$
Avg Read Latency (ms/op)	$\text{Avg}(\text{VolumeTotalReadTime}) * 1000$
Avg Write Latency (ms/op)	$\text{Avg}(\text{VolumeTotalWriteTime}) * 1000$

For the average latency graphs and average size graphs, the average is calculated over the total number of operations (read or write, whichever is applicable to the graph) that completed during the period.

Monitoring Volumes with Status Checks

Volume status checks enable you to better understand, track, and manage potential inconsistencies in the data on an Amazon EBS volume. They are designed to provide you with the information that you need to determine whether your Amazon EBS volumes are impaired, and to help you control how a potentially inconsistent volume is handled.

Volume status checks are automated tests that run every 5 minutes and return a pass or fail status. If all checks pass, the status of the volume is `ok`. If a check fails, the status of the volume is `impaired`. If the status is `insufficient-data`, the checks may still be in progress on the volume. You can view the results of volume status checks to identify any impaired volumes and take any necessary actions.

When Amazon EBS determines that a volume's data is potentially inconsistent, the default is that it disables I/O to the volume from any attached EC2 instances, which helps to prevent data corruption. After I/O is disabled, the next volume status check fails, and the volume status is `impaired`. In addition, you'll see an event that lets you know that I/O is disabled, and that you can resolve the impaired status of the volume by enabling I/O to the volume. We wait until you enable I/O to give you the opportunity to decide whether to continue to let your instances use the volume, or to run a consistency check using a command, such as `fsck` (Linux) or `chkdsk` (Windows), before doing so.

Note

Volume status is based on the volume status checks, and does not reflect the volume state. Therefore, volume status does not indicate volumes in the `error` state (for example, when a volume is incapable of accepting I/O.)

If the consistency of a particular volume is not a concern for you, and you'd prefer that the volume be made available immediately if it's impaired, you can override the default behavior by configuring the volume to automatically enable I/O. If you enable the `AutoEnableIO` volume attribute, the volume status check continues to pass. In addition, you'll see an event that lets you know that the volume was determined to be potentially inconsistent, but that its I/O was automatically enabled. This enables you to check the volume's consistency or replace it at a later time.

The I/O performance status check compares actual volume performance to the expected performance of a volume and alerts you if the volume is performing below expectations. This status check is only available for `io1` volumes that are attached to an instance and is not valid for General Purpose SSD

(gp2), Throughput Optimized HDD (st1), Cold HDD (sc1), or Magnetic (standard) volumes. The I/O performance status check is performed once every minute and CloudWatch collects this data every 5 minutes, so it may take up to 5 minutes from the moment you attach a io1 volume to an instance for this check to report the I/O performance status.

Important

While initializing io1 volumes that were restored from snapshots, the performance of the volume may drop below 50 percent of its expected level, which causes the volume to display a warning state in the **I/O Performance** status check. This is expected, and you can ignore the warning state on io1 volumes while you are initializing them. For more information, see [Initializing Amazon EBS Volumes \(p. 810\)](#).

The following table lists statuses for Amazon EBS volumes.

Volume status	I/O enabled status	I/O performance status (only available for Provisioned IOPS volumes)
ok	Enabled (I/O Enabled or I/O Auto-Enabled)	Normal (Volume performance is as expected)
warning	Enabled (I/O Enabled or I/O Auto-Enabled)	Degraded (Volume performance is below expectations) Severely Degraded (Volume performance is well below expectations)
impaired	Enabled (I/O Enabled or I/O Auto-Enabled) Disabled (Volume is offline and pending recovery, or is waiting for the user to enable I/O)	Stalled (Volume performance is severely impacted) Not Available (Unable to determine I/O performance because I/O is disabled)
insufficient-data	Enabled (I/O Enabled or I/O Auto-Enabled) Insufficient Data	Insufficient Data

To view and work with status checks, you can use the Amazon EC2 console, the API, or the command line interface.

To view status checks in the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Volumes**.
3. On the **EBS Volumes** page, use the **Volume Status** column lists the operational status of each volume.
4. To view an individual volume's status, select the volume, and choose **Status Checks**.

Volumes: | vol-d882c69b



IO operations have been disabled since 16 hours and 58 minutes ago ago. Data inconsistencies

Description

Status Checks

Monitoring

Tags

Volume Status **impaired**

IO Status **Disabled**

Since **December 23, 2013 7:06:41 PM UTC+2**

Description **Awaiting Action: Enable IO**

Auto-Enabled IO **Disabled** [Edit](#)

[Find out more](#) about working with volume status checks and events.

If you need technical assistance with your volume, post your issue to the [Developer Forums](#) or visit our

5. If you have a volume with a failed status check (status is `impaired`), see [Working with an Impaired Volume](#) (p. 776).

Alternatively, you can use the **Events** pane to view all events for your instances and volumes in a single pane. For more information, see [Monitoring Volume Events](#) (p. 774).

To view volume status information with the command line

You can use one of the following commands to view the status of your Amazon EBS volumes. For more information about these command line interfaces, see [Accessing Amazon EC2](#) (p. 3).

- `describe-volume-status` (AWS CLI)
- `Get-EC2VolumeStatus` (AWS Tools for Windows PowerShell)

Monitoring Volume Events

When Amazon EBS determines that a volume's data is potentially inconsistent, it disables I/O to the volume from any attached EC2 instances by default. This causes the volume status check to fail, and creates a volume status event that indicates the cause of the failure.

To automatically enable I/O on a volume with potential data inconsistencies, change the setting of the `AutoEnableIO` volume attribute. For more information about changing this attribute, see [Working with an Impaired Volume](#) (p. 776).

Each event includes a start time that indicates the time at which the event occurred, and a duration that indicates how long I/O for the volume was disabled. The end time is added to the event when I/O for the volume is enabled.

Volume status events include one of the following descriptions:

Awaiting Action: Enable IO

Volume data is potentially inconsistent. I/O is disabled for the volume until you explicitly enable it. The event description changes to **IO Enabled** after you explicitly enable I/O.

IO Enabled

I/O operations were explicitly enabled for this volume.

IO Auto-Enabled

I/O operations were automatically enabled on this volume after an event occurred. We recommend that you check for data inconsistencies before continuing to use the data.

Normal

For *io1* volumes only. Volume performance is as expected.

Degraded

For *io1* volumes only. Volume performance is below expectations.

Severely Degraded

For *io1* volumes only. Volume performance is well below expectations.

Stalled

For *io1* volumes only. Volume performance is severely impacted.

You can view events for your volumes using the Amazon EC2 console, the API, or the command line interface.

To view events for your volumes in the console


1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Events**.
3. All instances and volumes that have events are listed. You can filter by volume to view only volume status. You can also filter on specific status types.
4. Select a volume to view its specific event.

Actions ▾

Filter: Volume resources ▾ All event types ▾ Ongoing and scheduled ▾

<input type="checkbox"/>	Resource Name ▾	Resource Type ▾	Resource Id ▾	Availability Zone ▾	Event Type ▾	Event Status ▾
<input type="checkbox"/>		volume	vol-0381c540	us-east-1d	potential-data-i...	Awa
<input checked="" type="checkbox"/>		volume	vol-3682c675	us-east-1d	potential-data-i...	Awa

Event: vol-3682c675

 IO operations have been disabled since 30 days, 15 hours and 22 minutes ago. Data inconsistency detected.

Availability Zone	us-east-1d
Event Type	potential-data-inconsistency
Event Status	Awaiting Action: Enable IO
IO status	IO Disabled
Attached to	i-93aae4ea
Start Time	December 23, 2013 7:09:20 PM UTC+2
End time	

Find out more about [monitoring volume events](#).

If you have a volume where I/O is disabled, see [Working with an Impaired Volume \(p. 776\)](#). If you have a volume where I/O performance is below normal, this might be a temporary condition due to an action you have taken (e.g., creating a snapshot of a volume during peak usage, running the volume on an instance that cannot support the I/O bandwidth required, accessing data on the volume for the first time, etc.).

To view events for your volumes with the command line

You can use one of the following commands to view event information for your Amazon EBS volumes. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [describe-volume-status](#) (AWS CLI)
- [Get-EC2VolumeStatus](#) (AWS Tools for Windows PowerShell)

Working with an Impaired Volume

This section discusses your options if a volume is impaired because the volume's data is potentially inconsistent.

Options

- [Option 1: Perform a Consistency Check on the Volume Attached to its Instance \(p. 777\)](#)

- [Option 2: Perform a Consistency Check on the Volume Using Another Instance \(p. 778\)](#)
- [Option 3: Delete the Volume If You No Longer Need It \(p. 779\)](#)

Option 1: Perform a Consistency Check on the Volume Attached to its Instance

The simplest option is to enable I/O and then perform a data consistency check on the volume while the volume is still attached to its Amazon EC2 instance.

To perform a consistency check on an attached volume

1. Stop any applications from using the volume.
2. Enable I/O on the volume.
 - a. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
 - b. In the navigation pane, choose **Volumes**.
 - c. Select the volume on which to enable I/O operations.
 - d. In the details pane, choose **Enable Volume IO**.

Volumes: | vol-d882c69b

IO operations have been disabled since 16 hours and 58 minutes ago ago. Data inconsistency

Description	Status Checks	Monitoring	Tags
Volume ID	vol-d882c69b		
Capacity	100 GiB		
Created	November 21, 2013 3:42:01 PM UTC+2		
State	available		
Volume type	io1		
Product codes	-		

- e. In **Enable Volume IO**, choose **Yes, Enable**.
3. Check the data on the volume.
 - a. Run the **fsck** (Linux) or **chkdsk** (Windows) command.
 - b. (Optional) Review any available application or system logs for relevant error messages.
 - c. If the volume has been impaired for more than 20 minutes you can contact support. Choose **Troubleshoot**, and then on the **Troubleshoot Status Checks** dialog box, choose **Contact Support** to submit a support case.

To enable I/O for a volume with the command line

You can use one of the following commands to view event information for your Amazon EBS volumes. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [enable-volume-io](#) (AWS CLI)
- [Enable-EC2VolumeIO](#) (AWS Tools for Windows PowerShell)

Option 2: Perform a Consistency Check on the Volume Using Another Instance

Use the following procedure to check the volume outside your production environment.

Important

This procedure may cause the loss of write I/Os that were suspended when volume I/O was disabled.

To perform a consistency check on a volume in isolation

1. Stop any applications from using the volume.
2. Detach the volume from the instance.
 - a. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
 - b. In the navigation pane, choose **Volumes**.
 - c. Select the volume to detach.
 - d. Choose **Actions, Force Detach Volume**. You'll be prompted for confirmation.
3. Enable I/O on the volume.
 - a. In the navigation pane, choose **Volumes**.
 - b. Select the volume that you detached in the previous step.
 - c. In the details pane, choose **Enable Volume IO**.

Volumes: | vol-d882c69b

Volume ID	vol-d882c69b
Capacity	100 GiB
Created	November 21, 2013 3:42:01 PM UTC+2
State	available
Volume type	io1
Product codes	-

- d. In the **Enable Volume IO** dialog box, choose **Yes, Enable**.
4. Attach the volume to another instance. For information, see [Launch Your Instance \(p. 244\)](#) and [Attaching an Amazon EBS Volume to an Instance \(p. 766\)](#).
 5. Check the data on the volume.
 - a. Run the **fsck** (Linux) or **chkdsk** (Windows) command.
 - b. (Optional) Review any available application or system logs for relevant error messages.
 - c. If the volume has been impaired for more than 20 minutes, you can contact support. Choose **Troubleshoot**, and then in the troubleshooting dialog box, choose **Contact Support** to submit a support case.

To enable I/O for a volume with the command line

You can use one of the following commands to view event information for your Amazon EBS volumes. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [enable-volume-io](#) (AWS CLI)
- [Enable-EC2VolumeIO](#) (AWS Tools for Windows PowerShell)

Option 3: Delete the Volume If You No Longer Need It

If you want to remove the volume from your environment, simply delete it. For information about deleting a volume, see [Deleting an Amazon EBS Volume \(p. 783\)](#).

If you have a recent snapshot that backs up the data on the volume, you can create a new volume from the snapshot. For information about creating a volume from a snapshot, see [Restoring an Amazon EBS Volume from a Snapshot \(p. 763\)](#).

Working with the AutoEnableIO Volume Attribute

When Amazon EBS determines that a volume's data is potentially inconsistent, it disables I/O to the volume from any attached EC2 instances by default. This causes the volume status check to fail, and creates a volume status event that indicates the cause of the failure. If the consistency of a particular volume is not a concern, and you prefer that the volume be made available immediately if it's impaired, you can override the default behavior by configuring the volume to automatically enable I/O. If you enable the `AutoEnableIO` volume attribute, I/O between the volume and the instance is automatically re-enabled and the volume's status check will pass. In addition, you'll see an event that lets you know that the volume was in a potentially inconsistent state, but that its I/O was automatically enabled. When this event occurs, you should check the volume's consistency and replace it if necessary. For more information, see [Monitoring Volume Events \(p. 774\)](#).

This section explains how to view and modify the `AutoEnableIO` attribute of a volume using the Amazon EC2 console, the command line interface, or the API.

To view the AutoEnableIO attribute of a volume in the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Volumes**.
3. Select the volume.
4. In the lower pane, choose **Status Checks**.
5. In the **Status Checks** tab, **Auto-Enable IO** displays the current setting for your volume, either `Enabled` or `Disabled`.

Volumes: | vol-d882c69b



IO operations have been disabled since 16 hours and 58 minutes ago ago. Data inconsistencies

Description

Status Checks

Monitoring

Tags

Volume Status impaired

IO Status Disabled

Since December 23, 2013 7:06:41 PM UTC+2

Description Awaiting Action: Enable IO

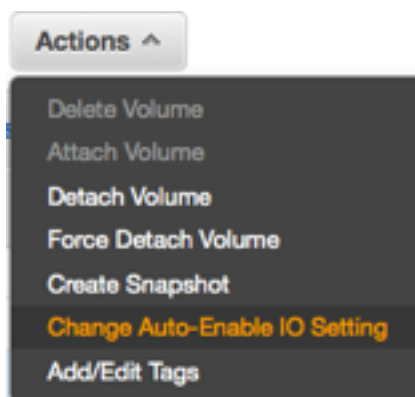
Auto-Enabled IO Disabled [Edit](#)

[Find out more](#) about working with volume status checks and events.

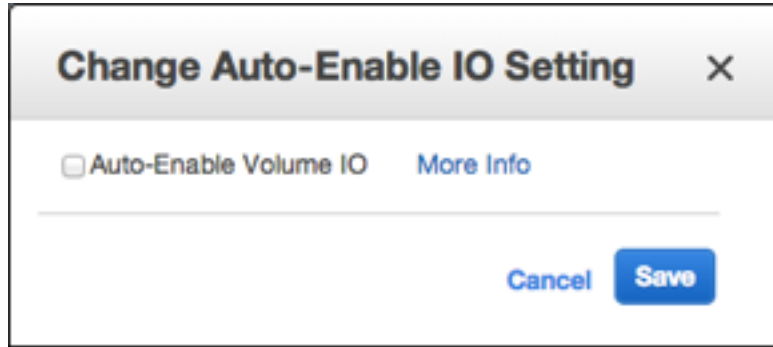
If you need technical assistance with your volume, post your issue to the [Developer Forums](#) or visit our

To modify the AutoEnableIO attribute of a volume in the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Volumes**.
3. Select the volume.
4. At the top of the **Volumes** page, choose **Actions**.
5. Choose **Change Auto-Enable IO Setting**.



6. In the **Change Auto-Enable IO Setting** dialog box, select the **Auto-Enable Volume IO** option to automatically enable I/O for an impaired volume. To disable the feature, clear the option.



7. Choose **Save**.

Alternatively, instead of completing steps 4-6 in the previous procedure, choose **Status Checks, Edit**.

To view or modify the `AutoEnableIO` attribute of a volume with the command line

You can use one of the following commands to view the `AutoEnableIO` attribute of your Amazon EBS volumes. For more information about these command line interfaces, see [Accessing Amazon EC2](#) (p. 3).

- [describe-volume-attribute](#) (AWS CLI)
- [Get-EC2VolumeAttribute](#) (AWS Tools for Windows PowerShell)

To modify the `AutoEnableIO` attribute of a volume, you can use one of the commands below.

- [modify-volume-attribute](#) (AWS CLI)
- [Edit-EC2VolumeAttribute](#) (AWS Tools for Windows PowerShell)

Detaching an Amazon EBS Volume from an Instance

You can detach an Amazon EBS volume from an instance explicitly or by terminating the instance. However, if the instance is running, you must first unmount the volume from the instance.;

If an EBS volume is the root device of an instance, you must stop the instance before you can detach the volume.

When a volume with an AWS Marketplace product code is detached from an instance, the product code is no longer associated with the instance.

Important

After you detach a volume, you are still charged for volume storage as long as the storage amount exceeds the limit of the AWS Free Tier. You must delete a volume to avoid incurring further charges. For more information, see [Deleting an Amazon EBS Volume](#) (p. 783).

This example unmounts the volume and then explicitly detaches it from the instance. This is useful when you want to terminate an instance or attach a volume to a different instance. To verify that the volume is no longer attached to the instance, see [Viewing Volume Information](#) (p. 769).

Note that you can reattach a volume that you detached (without unmounting it), but it might not get the same mount point and the data on the volume might be out of sync if there were writes to the volume in progress when it was detached.

To detach an EBS volume using the console

1. Unmount the volume. Choose **Disk Management**, right-click the volume, and then choose **Change Drive Letter and Path**. Select the mount point and choose **Remove**.
2. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
3. In the navigation pane, choose **Volumes**.
4. Select a volume and choose **Actions, Detach Volume**.
5. In the confirmation dialog box, choose **Yes, Detach**.

To detach an EBS volume from an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [detach-volume](#) (AWS CLI)
- [Dismount-EC2Volume](#) (AWS Tools for Windows PowerShell)

Troubleshooting

This section deals with common problems encountered when detaching volumes, and how to resolve them.

Note

To guard against the possibility of data loss, take a snapshot of your volume before attempting to unmount it. Forced detachment of a stuck volume can cause damage to the file system or the data it contains or an inability to attach a new volume using the same device name, unless you reboot the instance.

- If you encounter problems while detaching a volume through the Amazon EC2 console, it may be helpful to use the **describe-volumes** CLI command to diagnose the issue. For more information, see [describe-volumes](#).
- If your volume stays in the `detaching` state, you can force the detachment by choosing **Force Detach**. Use this option only as a last resort to detach a volume from a failed instance, or if you are detaching a volume with the intention of deleting it. The instance doesn't get an opportunity to flush file system caches or file system metadata. If you use this option, you must perform file system check and repair procedures.
- If you've tried to force the volume to detach multiple times over several minutes and it stays in the `detaching` state, you can post a request for help to the [Amazon EC2 forum](#). To help expedite a resolution, include the volume ID and describe the steps that you've already taken.
- When you attempt to detach a volume that is still mounted, the volume can become stuck in the `busy` state while it is trying to detach. The following output from **describe-volumes** shows an example of this condition:

```
[ec2-user ~]$ aws ec2 describe-volumes --region us-west-2 --volume-ids
vol-1234abcd
{
  "Volumes": [
    {
      "AvailabilityZone": "us-west-2b",
      "Attachments": [
        {
          "AttachTime": "2016-07-21T23:44:52.000Z",
          "InstanceId": "i-fedc9876",
          "VolumeId": "vol-1234abcd",
          "State": "busy",
```

```
        "DeleteOnTermination": false,  
        "Device": "/dev/sdf"  
    }  
    ....
```

When you encounter this state, detachment can be delayed indefinitely until you unmount the volume, force detachment, reboot the instance, or all three.

Deleting an Amazon EBS Volume

After you no longer need an Amazon EBS volume, you can delete it. After deletion, its data is gone and the volume can't be attached to any instance. However, before deletion, you can store a snapshot of the volume, which you can use to re-create the volume later.

To delete a volume, it must be in the `available` state (not attached to an instance).

To delete an EBS volume using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Volumes**.
3. Select a volume and choose **Actions, Delete Volume**.
4. In the confirmation dialog box, choose **Yes, Delete**.

To delete an EBS volume using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- `delete-volume` (AWS CLI)
- `Remove-EC2Volume` (AWS Tools for Windows PowerShell)

Expanding the Storage Space of an EBS Volume on Windows

You can increase the storage space of an existing EBS volume without losing the data on the volume. To do this, you migrate your data to a larger volume and then extend the file system on the volume to recognize the newly-available space. After you verify that your new volume is working properly, you can delete the old volume.

Tasks

- [Migrating Your Data to a Larger Volume \(p. 784\)](#)
- [Extending a Windows File System \(p. 785\)](#)
- [Deleting the Old Volume \(p. 788\)](#)

If you need to expand the storage space of a volume on a Linux instance, see [Expanding the Storage Space of a Volume](#) in the *Amazon EC2 User Guide for Linux Instances*.

If you create a larger volume, you will be charged for the additional storage. For more information, see the *Amazon Elastic Block Store* section on the [Amazon EC2 Pricing](#) page.

Note

If your storage needs demand a larger EBS volume than AWS provides, you may want to use RAID 0 to "stripe" a single logical volume across multiple physical volumes. For more information see [RAID Configuration on Windows](#).

Migrating Your Data to a Larger Volume

You must stop your instance to expand the storage space. When you stop and start an instance, be aware of the following:

- If you've attached instance store (ephemeral) volumes to your instance, any data on these volumes is erased. Therefore, if you have any data on that you want to keep, back it up to persistent storage. For more information, see [Amazon EC2 Instance Store \(p. 822\)](#) and [Amazon EBS Volumes \(p. 747\)](#).
- If your instance is running in a VPC and has a public IPv4 address, we release the address and give it a new public IPv4 address. The instance retains its private IPv4 addresses and any Elastic IP addresses.
- If your instance is running in EC2-Classic, we give it new public and private IPv4 addresses, and disassociate any Elastic IP address that's associated with the instance. You must re-associate any Elastic IP address after you restart your instance.
- If your instance is in an Auto Scaling group, the Auto Scaling service marks the stopped instance as unhealthy, and may terminate it and launch a replacement instance. To prevent this, you can temporarily suspend the Auto Scaling processes for the group. For more information, see [Suspend and Resume Auto Scaling Processes](#) in the *Auto Scaling User Guide*.

To migrate your data to a larger volume

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances** and then locate the instance with the volume that you want to expand.
3. Make a note of the instance ID and Availability Zone. You will specify this information when you attach a new volume to the instance later in this procedure.
4. Verify that the instance **Shutdown Behavior** is set to **Stop** and not **Terminate**.
 - a. Choose the instance.
 - b. From the context-menu (right-click) choose **Instance Settings**, and then choose **Change Shutdown Behavior**.
 - c. If the **Shutdown behavior** is set to **Terminate**, choose **Stop**, and then choose **Apply**.

If the **Shutdown behavior** is already set to **Stop**, then choose **Cancel**.
5. Stop the instance. For more information about how to stop an instance, see [Stopping and Starting Your Instances \(p. 260\)](#).

Warning

When you stop an instance, the data on any instance store volumes is erased. Therefore, if you have any data on instance store volumes that you want to keep, be sure to back it up to persistent storage.

6. Create a snapshot of the volume to expand.
 - a. In the navigation pane, choose **Volumes**, and then locate the volume you want to expand.
 - b. From the context-menu (right-click) choose the volume that you want to expand, and then choose **Create Snapshot**.
 - c. Enter information in the **Name** and **Description** fields, and then choose **Yes, Create**.
7. Create a new volume from the snapshot.
 - a. In the navigation pane, choose **Snapshots**.
 - b. When the status of the snapshot that you just created is set to **completed**, choose the snapshot, and then from the context-menu (right-click) choose **Create Volume**.

- c. In the **Create Volume** dialog box, choose the desired volume type and enter the new volume size. You must also set the **Availability Zone** to match the instance Availability Zone. Choose **Yes, Create**.

Important

If you do not set the **Availability Zone** to match the instance then you will not be able to attach the new volume to the instance.

8. Detach the old volume.
 - a. In the navigation pane, choose **Volumes**, and then choose the old volume from the list. Make a note of the device name in the **Attachment Information** field. You will specify this information when you attach a new volume to the instance later in this procedure. The information appears in the following format:

```
i-xxxxxxxxxxxxxxxxxxxx (instance_name):device_name
```

- b. From the context-menu (right-click) choose the old volume, and then choose **Detach Volume**.
 - c. In the **Detach Volume** dialog box, choose **Yes, Detach**. It may take several minutes for the volume to detach.
9. Attach the newly expanded volume
 - a. In the navigation pane, choose **Volumes**.
 - b. From the context-menu (right-click) choose the new volume, and then choose **Attach Volume**.
 - c. Start typing the name or ID of the instance in the **Instance** field, and then choose the instance.
 - d. Enter the same device name retrieved in [Step 8.a \(p. 785\)](#), and then choose **Yes, Attach**. It is important to attach the new volume to the exact location you noted above (for example `/dev/sda1`).
10. Restart the instance.
 - a. In the navigation pane, choose **Instances** and then choose the instance you want to restart.
 - b. From the context-menu (right-click) choose **Instance State**, and then choose **Start**.
 - c. In the **Start Instances** dialog box, choose **Yes, Start**. If the instance fails to start, and the volume being expanded is a root volume, verify that you attached the expanded volume using the same device name as the original volume, for example `/dev/sda1`.

After the instance has started, you can check the file system size to see if your instance recognizes the larger volume space.

If the size does not reflect your newly-expanded volume, you must extend the file system of your device so that your instance can use the new space. For more information, see [Extending a Windows File System \(p. 785\)](#).

You may have to bring the volume online in order to use it. For more information, see [Making an Amazon EBS Volume Available for Use \(p. 767\)](#). You do not need to reformat the volume.

Note

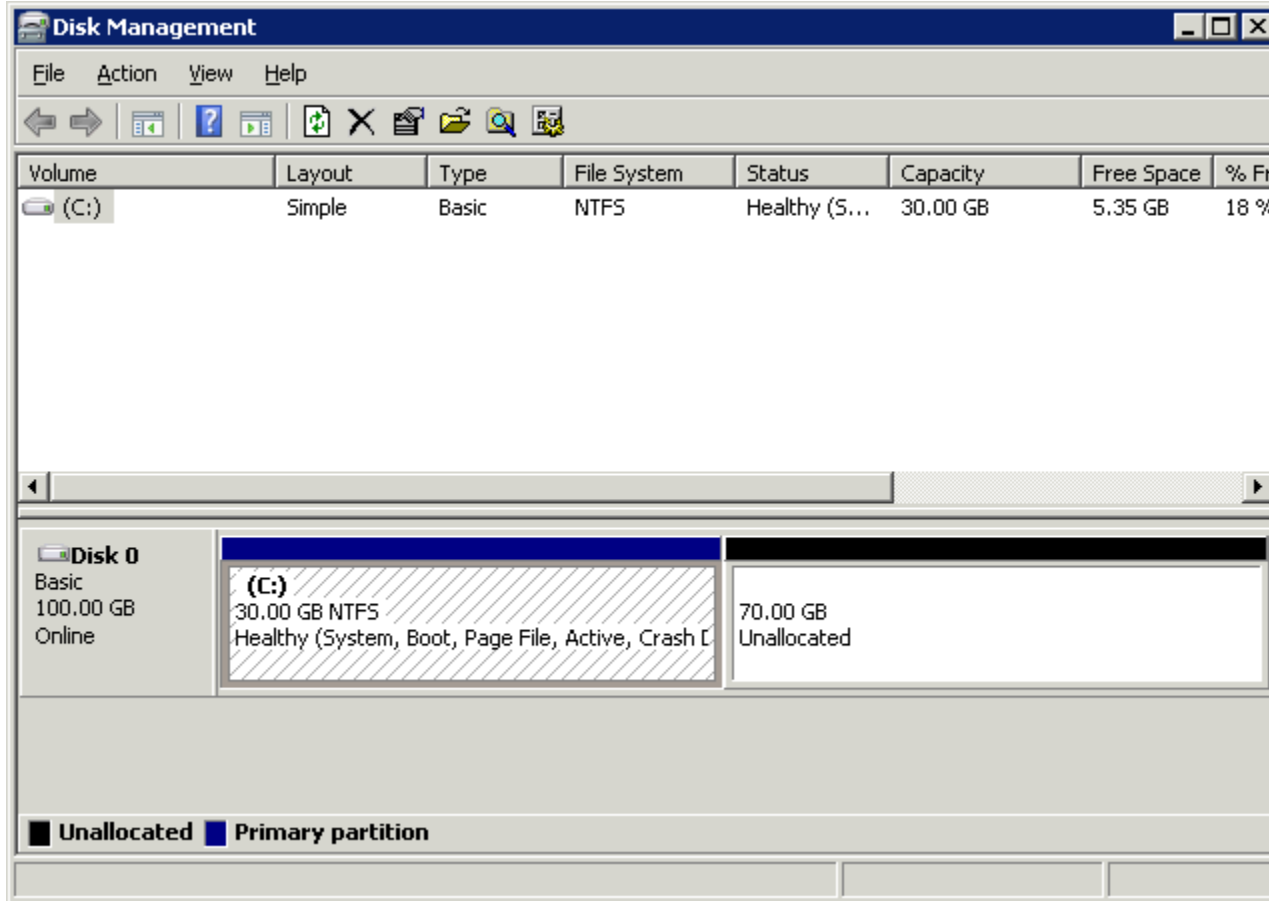
After you are done resizing your volume, you should delete the snapshot you created in the procedure above to avoid incurring storage costs.

Extending a Windows File System

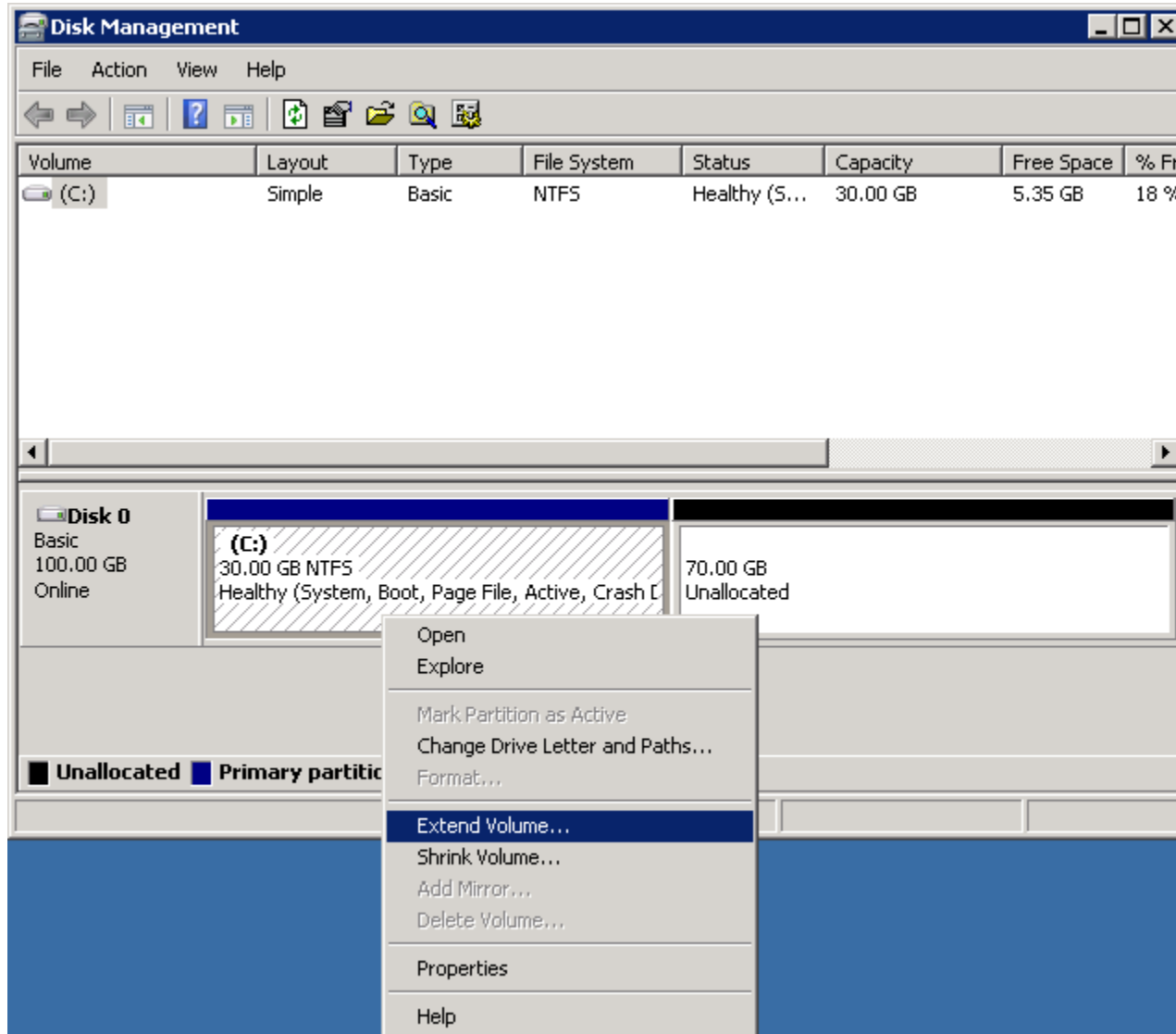
In Windows, you use the Disk Management utility to extend the disk size to the new size of the volume.

To extend a Windows file system

1. Log in to your Windows instance using Remote Desktop.
2. In the **Run** dialog, type **diskmgmt.msc** and press **Enter**. The Disk Management utility opens.

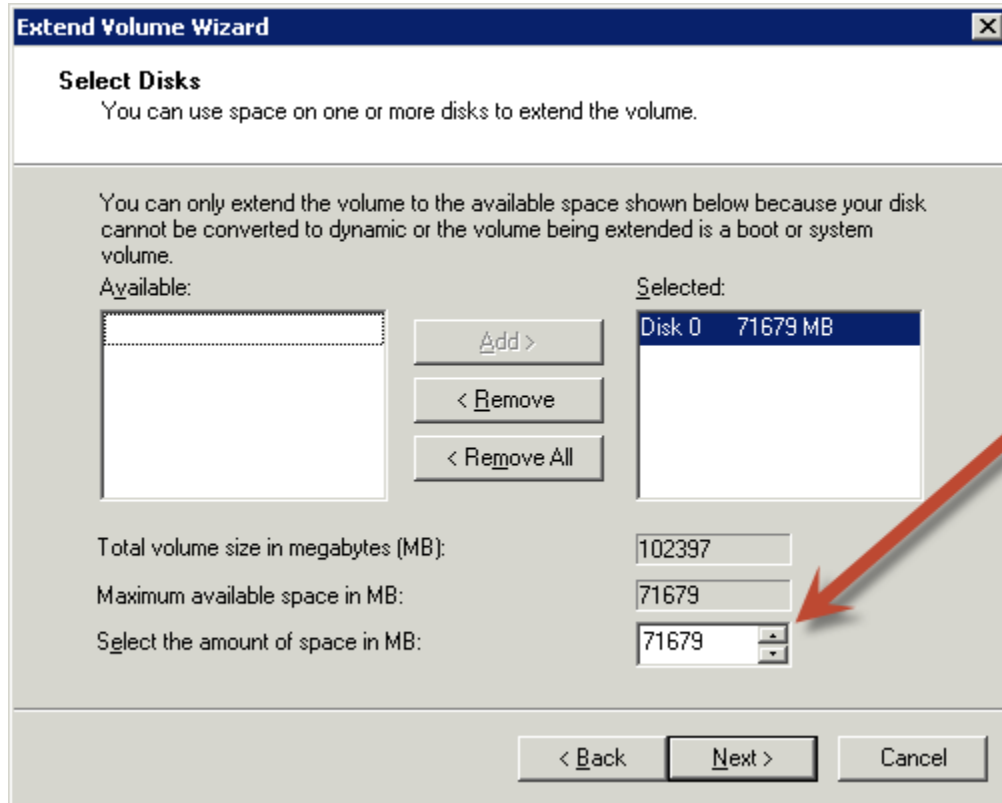


3. Right-click the expanded drive and select **Extend Volume**.



4. In the Extend Volume Wizard, choose **Next**, then set the **Select the amount of space in MB** field to the number of megabytes by which to extend the volume. Normally, you set this to the maximum available space. Note that the highlighted text under **Selected** is the amount of space that will be added, not the final size the volume will have.

Complete the wizard.



Deleting the Old Volume

After the new volume has been attached and extended in the instance, you can delete the old volume if it is no longer needed.

To delete the old volume

1. In the Amazon EC2 console, choose **Volumes** in the navigation pane and then choose the volume you want to delete.
2. From the context-menu (right-click) choose **Delete Volume**.
3. In the **Delete Volume** dialog box, choose **Yes, Delete**.

Amazon EBS Snapshots

You can back up the data on your EBS volumes to Amazon S3 by taking point-in-time snapshots. Snapshots are incremental backups, which means that only the blocks on the device that have changed after your most recent snapshot are saved. This minimizes the time required to create the snapshot and saves on storage costs. When you delete a snapshot, only the data unique to that snapshot is removed. Active snapshots contain all of the information needed to restore your data (from the time the snapshot was taken) to a new EBS volume.

Contents

- [Snapshot Overview \(p. 789\)](#)
- [Creating an Amazon EBS Snapshot \(p. 789\)](#)
- [Deleting an Amazon EBS Snapshot \(p. 791\)](#)

- [Copying an Amazon EBS Snapshot \(p. 791\)](#)
- [Viewing Amazon EBS Snapshot Information \(p. 793\)](#)
- [Sharing an Amazon EBS Snapshot \(p. 794\)](#)

Snapshot Overview

When you create an EBS volume, you can create it based on an existing snapshot. The new volume begins as an exact replica of the original volume that was used to create the snapshot. When you create a volume from an existing snapshot, it loads lazily in the background so that you can begin using them right away. If you access a piece of data that hasn't been loaded yet, the volume immediately downloads the requested data from Amazon S3, and then continues loading the rest of the volume's data in the background. For more information, see [Creating an Amazon EBS Snapshot \(p. 789\)](#).

By modifying their access permissions, snapshots can be shared across AWS accounts. You can make copies of your own snapshots as well as snapshots that have been shared with you. For more information, see [Sharing an Amazon EBS Snapshot \(p. 794\)](#).

EBS snapshots broadly support EBS encryption:

- Snapshots of encrypted volumes are automatically encrypted.
- Volumes that are created from encrypted snapshots are automatically encrypted.
- When you copy an unencrypted snapshot that you own, you can encrypt it during the copy process.
- When you copy an encrypted snapshot that you own, you can reencrypt it with a different key during the copy process.

For more information, see [Amazon EBS Encryption](#).

Snapshots are constrained to the region in which they are created. After you have created a snapshot of an EBS volume, you can use it to create new volumes in the same region. For more information, see [Restoring an Amazon EBS Volume from a Snapshot \(p. 763\)](#). You can also copy snapshots across regions, making it easier to leverage multiple regions for geographical expansion, data center migration, and disaster recovery. You can copy any accessible snapshots that have a `completed` status. For more information, see [Copying an Amazon EBS Snapshot \(p. 791\)](#).

Creating an Amazon EBS Snapshot

After writing data to an EBS volume, you can periodically create a snapshot of the volume to use as a baseline for new volumes or for data backup. If you make periodic snapshots of a volume, the snapshots are incremental so that only the blocks on the device that have changed after your last snapshot are saved in the new snapshot. Even though snapshots are saved incrementally, the snapshot deletion process is designed so that you need to retain only the most recent snapshot in order to restore the volume.

Snapshots occur asynchronously; the point-in-time snapshot is created immediately, but the status of the snapshot is `pending` until the snapshot is complete (when all of the modified blocks have been transferred to Amazon S3), which can take several hours for large initial snapshots or subsequent snapshots where many blocks have changed. While it is completing, an in-progress snapshot is not affected by ongoing reads and writes to the volume.

Important

Although you can take a snapshot of a volume while a previous snapshot of that volume is in the `pending` status, having multiple `pending` snapshots of a volume may result in reduced volume performance until the snapshots complete.

There is a limit of 5 `pending` snapshots for a single `gp2`, `io1`, or Magnetic volume, and 1 `pending` snapshot for a single `st1` or `sc1` volume. If you receive a `ConcurrentSnapshotLimitExceeded` error while trying to create multiple concurrent snapshots of the same volume, wait for one or more of the `pending` snapshots to complete before creating another snapshot of that volume.

Snapshots that are taken from encrypted volumes are automatically encrypted. Volumes that are created from encrypted snapshots are also automatically encrypted. The data in your encrypted volumes and any associated snapshots is protected both at rest and in motion. For more information, see [Amazon EBS Encryption](#).

By default, only you can create volumes from snapshots that you own. However, you can share your unencrypted snapshots with specific AWS accounts, or you can share them with the entire AWS community by making them public. For more information, see [Sharing an Amazon EBS Snapshot \(p. 794\)](#).

You can share an encrypted snapshot only with specific AWS accounts. For others to use your shared, encrypted snapshot, you must also share the CMK key that was used to encrypt it. Users with access to your encrypted snapshot must create their own personal copy of it and then use that copy to restore the volume. Your copy of a shared, encrypted snapshot can also be re-encrypted with a different key. For more information, see [Sharing an Amazon EBS Snapshot \(p. 794\)](#).

When a snapshot is created from a volume with an AWS Marketplace product code, the product code is propagated to the snapshot.

You can take a snapshot of an attached volume that is in use. However, snapshots only capture data that has been written to your Amazon EBS volume at the time the snapshot command is issued. This might exclude any data that has been cached by any applications or the operating system. If you can pause any file writes to the volume long enough to take a snapshot, your snapshot should be complete. However, if you can't pause all file writes to the volume, you should unmount the volume from within the instance, issue the snapshot command, and then remount the volume to ensure a consistent and complete snapshot. You can remount and use your volume while the snapshot status is `pending`.

To create a snapshot for Amazon EBS volumes that serve as root devices, you should stop the instance before taking the snapshot.

To unmount the volume in Windows, open Disk Management, right-click the volume to unmount, and select **Change Drive Letter and Path**. Select the mount point to remove, and then click **Remove**.

After you've created a snapshot, you can tag it to help you manage it later. For example, you can add tags describing the original volume from which the snapshot was created, or the device name that was used to attach the original volume to an instance. For more information, see [Tagging Your Amazon EC2 Resources \(p. 859\)](#).

To create a snapshot using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Snapshots** in the navigation pane.
3. Choose **Create Snapshot**.
4. In the **Create Snapshot** dialog box, select the volume to create a snapshot for, and then choose **Create**.

To create a snapshot using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- `create-snapshot` (AWS CLI)

- [New-EC2Snapshot](#) (AWS Tools for Windows PowerShell)

Deleting an Amazon EBS Snapshot

When you delete a snapshot, only the data exclusive to that snapshot is removed. Deleting previous snapshots of a volume does not affect your ability to restore volumes from later snapshots of that volume.

If you make periodic snapshots of a volume, the snapshots are incremental so that only the blocks on the device that have changed since your last snapshot are saved in the new snapshot. Even though snapshots are saved incrementally, the snapshot deletion process is designed so that you need to retain only the most recent snapshot in order to restore the volume.

Note that you can't delete a snapshot of the root device of an EBS volume used by a registered AMI. You must first deregister the AMI before you can delete the snapshot. For more information, see [Deregistering Your AMI](#) (p. 92).

To delete a snapshot using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Snapshots** in the navigation pane.
3. Select a snapshot and then choose **Delete** from the **Actions** list.
4. Choose **Yes, Delete**.

To delete a snapshot using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2](#) (p. 3).

- [delete-snapshot](#) (AWS CLI)
- [Remove-EC2Snapshot](#) (AWS Tools for Windows PowerShell)

Copying an Amazon EBS Snapshot

With Amazon EBS, you can create point-in-time snapshots of volumes which we store for you in Amazon Simple Storage Service (Amazon S3). After you've created a snapshot and it has finished copying to Amazon S3 (when the snapshot status is `completed`), you can copy it from one AWS region to another, or within the same region. Amazon S3 server-side encryption (256-bit AES) protects a snapshot's data-in-transit during copying. The snapshot copy receives a snapshot ID different from the original snapshot's ID.

Note

To copy an Amazon Relational Database Service (Amazon RDS) snapshot, see [Copying a DB Snapshot](#) in the Amazon Relational Database Service User Guide.

You can use a copy of a snapshot in the following ways:

- **Geographic expansion:** Launch your applications in a new region.
- **Migration:** Move an application to a new region, to enable better availability and minimize cost.
- **Disaster recovery:** Back up your data and logs across different geographical locations at regular intervals. In case of disaster, you can restore your applications using point-in-time backups stored in the secondary region. This minimizes data loss and recovery time.
- **Encryption:** Encrypt a previously unencrypted snapshot, change the key with which the snapshot is encrypted, or, for encrypted snapshots that have been shared with you, create a copy that you own in order to restore the volume from it.

- **Data retention and auditing requirements:** Copy your encrypted EBS snapshots from one AWS account to another to preserve data logs or other files for auditing or data retention. Using a different account helps prevent accidental snapshot deletions, and protects you if your main AWS account is compromised.

Note

Snapshots created by the CopySnapshot action have an arbitrary volume ID that should not be used for any purpose.

User-defined tags are not copied from the source snapshot to the new snapshot. After the copy operation is complete, you can apply user-defined tags to the new snapshot. For more information, see [Tagging Your Amazon EC2 Resources \(p. 859\)](#).

You can have up to five snapshot copy requests in progress to a single destination per account. You can copy any accessible snapshots that have a `completed` status, including shared snapshots and snapshots that you've created. You can also copy AWS Marketplace, VM Import/Export, and AWS Storage Gateway snapshots, but you must verify that the snapshot is supported in the destination region.

When you copy a snapshot, you are only charged for the data transfer and storage used to copy the snapshot data across regions and to store the copied snapshot in the destination region. You are not charged if the snapshot copy fails. However, if you cancel a snapshot copy that is not yet complete, or delete the source snapshot while the copy is in progress, you are charged for the bandwidth of the data transferred.

The first snapshot copy to another region is always a full copy. Each subsequent snapshot copy is incremental (which makes the copy process faster), meaning that only the blocks in the snapshot that have changed after your last snapshot copy to the same destination are transferred. Support for incremental snapshots is specific to a region pair where a previous complete snapshot copy of the source volume is already available in the destination region, and it is limited to the default EBS CMK for encrypted snapshots. For example, if you copy an unencrypted snapshot from the US East (N. Virginia) region to the US West (Oregon) region, the first snapshot copy of the volume is a full copy and subsequent snapshot copies of the same volume transferred between the same regions are incremental.

Note

Snapshot copies within a single region do not copy any data at all as long as the following conditions apply:

- The encryption status of the snapshot copy does not change during the copy operation
- For encrypted snapshots, both the source snapshot and the copy are encrypted with the default EBS CMK

If you would like another account to be able to copy your snapshot, you must either modify the snapshot permissions to allow access to that account or make the snapshot public so that all AWS accounts may copy it. For more information, see [Sharing an Amazon EBS Snapshot \(p. 794\)](#).

Encrypted Snapshots

When you copy a snapshot, you can choose to encrypt the copy (if the original snapshot was not encrypted) or you can specify a CMK different from the original one, and the resulting copied snapshot will use the new CMK. However, changing the encryption status of a snapshot or using a non-default EBS CMK during a copy operation always results in a full copy (not incremental), which may incur greater data transfer and storage charges.

To copy an encrypted snapshot from another account, you must have permissions to use the snapshot and you must have permissions to use the customer master key (CMK) that was used to encrypt the original snapshot. For more information, see [Sharing an Amazon EBS Snapshot \(p. 794\)](#).

Note

When copying an encrypted snapshot that was shared with you, you should consider re-encrypting the snapshot during the copy process with a different key that you control. This protects you if the original key is compromised, or if the owner revokes the key for any reason, which could cause you to lose access to the volume you created.

To copy a snapshot using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Snapshots**.
3. Select the snapshot to copy, and then choose **Copy** from the **Actions** list.
4. In the **Copy Snapshot** dialog box, update the following as necessary:
 - **Destination region:** Select the region where you want to write the copy of the snapshot.
 - **Description:** By default, the description includes information about the source snapshot so that you can identify a copy from the original. You can change this description as necessary.
 - **Encryption:** If the source snapshot is not encrypted, you can choose to encrypt the copy. You cannot decrypt an encrypted snapshot.
 - **Master Key:** The customer master key (CMK) that will be used to encrypt this snapshot. You can select from master keys in your account or type/paste the ARN of a key from a different account. You can create a new master encryption key in the IAM console.
5. Choose **Copy**.
6. In the **Copy Snapshot** confirmation dialog box, choose **Snapshots** to go to the **Snapshots** page in the region specified, or choose **Close**.

To view the progress of the copy process later, switch to the destination region, and then refresh the **Snapshots** page. Copies in progress are listed at the top of the page.

To check for failure

If you attempt to copy an encrypted snapshot without having permissions to use the encryption key, the operation will fail silently. The error state will not be displayed in the console until you refresh the page. You can also check the state of the snapshot from the command line. For example:

```
C:\> aws ec2 describe-snapshots --snapshot-id snap-0123abcd
```

If the copy failed because of insufficient key permissions, you will see the following message:

```
"StateMessage": "Given key ID is not accessible"
```

Note

When copying an encrypted snapshot, you must have describe permissions on the default CMK. Explicitly denying these permissions will result in copy failure.

To copy a snapshot using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [copy-snapshot](#) (AWS CLI)
- [Copy-EC2Snapshot](#) (AWS Tools for Windows PowerShell)

Viewing Amazon EBS Snapshot Information

You can view detailed information about your snapshots.

To view snapshot information using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Snapshots** in the navigation pane.
3. To reduce the list, choose an option from the **Filter** list. For example, to view only your snapshots, choose **Owned By Me**. You can filter your snapshots further by using the advanced search options. Choose the search bar to view the filters available.
4. To view more information about a snapshot, choose it.

To view snapshot information using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [describe-snapshots](#) (AWS CLI)
- [Get-EC2Snapshot](#) (AWS Tools for Windows PowerShell)

Sharing an Amazon EBS Snapshot

You can share your unencrypted snapshots with your co-workers or others in the AWS community by modifying the permissions of the snapshot. Users that you have authorized can quickly use your unencrypted shared snapshots as the basis for creating their own EBS volumes. If you choose, you can also make your unencrypted snapshots available publicly to all AWS users.

You can share an encrypted snapshot with specific AWS accounts, though you cannot make it public. For others to use the snapshot, you must also share the custom CMK key used to encrypt it. Cross-account permissions may be applied to a custom key either when it is created or at a later time. Users with access can copy your snapshot and create their own EBS volumes based on your snapshot while your original snapshot remains unaffected.

Important

When you share a snapshot (whether by sharing it with another AWS account or making it public to all), you are giving others access to all the data on your snapshot. Share snapshots only with people with whom you want to share *all* your snapshot data.

Several technical and policy restrictions apply to sharing snapshots:

- Snapshots are constrained to the region in which they were created. If you would like to share a snapshot with another region, you need to copy the snapshot to that region. For more information about copying snapshots, see [Copying an Amazon EBS Snapshot \(p. 791\)](#).
- If your snapshot uses the longer resource ID format, you can only share it with another account that also supports longer IDs. For more information, see [Resource IDs](#).
- AWS prevents you from sharing snapshots that were encrypted with your default CMK. Snapshots that you intend to share must instead be encrypted with a custom CMK. For information about creating keys, see [Creating Keys](#).
- Users of your shared CMK who will be accessing encrypted snapshots must be granted `DescribeKey` and `ReEncrypt` permissions. For information about managing and sharing CMK keys, see [Controlling Access to Customer Master Keys](#).
- If you have access to a shared encrypted snapshot and you wish to restore a volume from it, you must create a personal copy of the snapshot and then use that copy to restore the volume. We recommend that you re-encrypt the snapshot during the copy process with a different key that you control. This protects your access to the volume if the original key is compromised, or if the owner revokes the key for any reason.

To modify snapshot permissions using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Snapshots** in the navigation pane.
3. Select a snapshot and then choose **Modify Permissions** from the **Actions** list.
4. Choose whether to make the snapshot public or to share it with specific AWS accounts:
 - To make the snapshot public, choose **Public**.

This is not a valid option for encrypted snapshots or snapshots with AWS Marketplace product codes.

- To expose the snapshot to only specific AWS accounts, choose **Private**, enter the ID of the AWS account (without hyphens) in the **AWS Account Number** field, and choose **Add Permission**. Repeat until you've added all the required AWS accounts.

Important

If your snapshot is encrypted, you must ensure that the following are true:

- The snapshot is encrypted with a custom CMK, not your default CMK. If you attempt to change the permissions of a snapshot encrypted with your default CMK, the console will display an error message.
- You are sharing the custom CMK with the accounts that have access to your snapshot.

5. Choose **Save**.

To view and modify snapshot permissions using the command line

To view the `createVolumePermission` attribute of a snapshot, you can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [describe-snapshot-attribute](#) (AWS CLI)
- [Get-EC2SnapshotAttribute](#) (AWS Tools for Windows PowerShell)

To modify the `createVolumePermission` attribute of a snapshot, you can use one of the following commands.

- [modify-snapshot-attribute](#) (AWS CLI)
- [Edit-EC2SnapshotAttribute](#) (AWS Tools for Windows PowerShell)

Amazon EBS–Optimized Instances

An Amazon EBS–optimized instance uses an optimized configuration stack and provides additional, dedicated capacity for Amazon EBS I/O. This optimization provides the best performance for your EBS volumes by minimizing contention between Amazon EBS I/O and other traffic from your instance.

EBS–optimized instances deliver dedicated bandwidth to Amazon EBS, with options between 500 Mbps and 10,000 Mbps, depending on the instance type you use. When attached to an EBS–optimized instance, General Purpose SSD (`gp2`) volumes are designed to deliver within 10% of their baseline and burst performance 99% of the time in a given year, and Provisioned IOPS SSD (`io1`) volumes are designed to deliver within 10% of their provisioned performance 99.9% of the time in a given year. Both Throughput Optimized HDD (`st1`) and Cold HDD (`sc1`) guarantee performance consistency of 90% of burst throughput 99% of the time in a given year. Non-compliant periods are approximately uniformly distributed, targeting 99% of expected total throughput each hour. For more information, see [Amazon EBS Volume Types \(p. 749\)](#).

When you enable EBS optimization for an instance that is not EBS-optimized by default, you pay an additional low, hourly fee for the dedicated capacity. For pricing information, see [EBS-optimized Instances](#) on the Amazon EC2 On-Demand Pricing page.

Contents

- [Instance Types that Support EBS Optimization](#) (p. 796)
- [Enabling EBS Optimization at Launch](#) (p. 798)
- [Modifying EBS Optimization for a Running Instance](#) (p. 798)

Instance Types that Support EBS Optimization

The following table shows which instance types support EBS optimization, the dedicated bandwidth to Amazon EBS, the maximum number of IOPS the instance can support if you are using a 16 KiB I/O size, and the typical maximum aggregate throughput that can be achieved on that connection in MiB/s with a streaming read workload and 128 KiB I/O size. Choose an EBS-optimized instance that provides more dedicated EBS throughput than your application needs; otherwise, the connection between Amazon EBS and Amazon EC2 can become a performance bottleneck.

Note that some instance types are EBS-optimized by default. For instances that are EBS-optimized by default, there is no need to enable EBS optimization and there is no effect if you disable EBS optimization using the CLI or API. You can enable EBS optimization for the other instance types that support EBS optimization when you launch the instances, or enable EBS optimization after the instances are running.

Instance type	EBS-optimized by default	Max. bandwidth (Mbps)*	Expected throughput (MB/s)**	Max. IOPS (16 KB I/O size)**
c1.xlarge		1,000	125	8,000
c3.xlarge		500	62.5	4,000
c3.2xlarge		1,000	125	8,000
c3.4xlarge		2,000	250	16,000
c4.large	Yes	500	62.5	4,000
c4.xlarge	Yes	750	93.75	6,000
c4.2xlarge	Yes	1,000	125	8,000
c4.4xlarge	Yes	2,000	250	16,000
c4.8xlarge	Yes	4,000	500	32,000
d2.xlarge	Yes	750	93.75	6,000
d2.2xlarge	Yes	1,000	125	8,000
d2.4xlarge	Yes	2,000	250	16,000
d2.8xlarge	Yes	4,000	500	32,000
g2.2xlarge		1,000	125	8,000
i2.xlarge		500	62.5	4,000
i2.2xlarge		1,000	125	8,000

Amazon Elastic Compute Cloud
User Guide for Windows Instances
EBS Optimization

Instance type	EBS-optimized by default	Max. bandwidth (Mbps)*	Expected throughput (MB/s)**	Max. IOPS (16 KB I/O size)**
i2.4xlarge		2,000	250	16,000
m1.large		500	62.5	4,000
m1.xlarge		1,000	125	8,000
m2.2xlarge		500	62.5	4,000
m2.4xlarge		1,000	125	8,000
m3.xlarge		500	62.5	4,000
m3.2xlarge		1,000	125	8,000
m4.large	Yes	450	56.25	3,600
m4.xlarge	Yes	750	93.75	6,000
m4.2xlarge	Yes	1,000	125	8,000
m4.4xlarge	Yes	2,000	250	16,000
m4.10xlarge	Yes	4,000	500	32,000
m4.16xlarge	Yes	10,000	1,250	65,000
p2.xlarge	Yes	750	93.75	6,000
p2.8xlarge	Yes	5,000	625	32,500
p2.16xlarge	Yes	10,000	1,250	65,000
r3.xlarge		500	62.5	4,000
r3.2xlarge		1,000	125	8,000
r3.4xlarge		2,000	250	16,000
r4.large	Yes	400	50	3,000
r4.xlarge	Yes	800	100	6,000
r4.2xlarge	Yes	1600	200	12,000
r4.4xlarge	Yes	3000	400	16,000
r4.8xlarge	Yes	6000	800	32,000
r4.16xlarge	Yes	12,000	1500	65,000
x1.16xlarge	Yes	5,000	625	32,500
x1.32xlarge	Yes	10,000	1,250	65,000

* These instance types must be launched as EBS-optimized to consistently achieve this level of performance.

** This value is a rounded approximation based on a 100% read-only workload and it is provided as a baseline configuration aid. EBS-optimized connections are full-duplex, and can drive more throughput

and IOPS in a 50/50 read/write workload where both communication lanes are used. In some cases, network, file system, and Amazon EBS encryption overhead can reduce the maximum throughput and IOPS available.

Note that some instances with 10-gigabit network interfaces, such as `i2.8xlarge` and `r3.8xlarge` do not offer EBS-optimization, and therefore do not have dedicated EBS bandwidth available and are not listed here. On these instances, network traffic and Amazon EBS traffic is shared on the same 10-gigabit network interface. Some other 10-gigabit network instances, such as `c4.8xlarge` and `d2.8xlarge` offer dedicated EBS bandwidth in addition to a 10-gigabit interface which is used exclusively for network traffic.

Enabling EBS Optimization at Launch

You can enable EBS optimization for an instance by setting its EBS-optimized attribute.

To enable EBS optimization when launching an instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Click **Launch Instance**. In **Step 1: Choose an Amazon Machine Image (AMI)**, select an AMI.
3. In **Step 2: Choose an Instance Type**, select an instance type that is listed as supporting EBS optimization.
4. In **Step 3: Configure Instance Details**, complete the fields that you need and select **Launch as EBS-optimized instance**. If the instance type that you selected in the previous step doesn't support EBS optimization, this option is not present. If the instance type that you selected is EBS-optimized by default, this option is selected and you can't deselect it.
5. Follow the directions to complete the wizard and launch your instance.

To enable EBS optimization when launching an instance using the command line

You can use one of the following options with the corresponding command. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- `--ebs-optimized` with `run-instances` (AWS CLI)
- `-EbsOptimized` with `New-EC2Instance` (AWS Tools for Windows PowerShell)

Modifying EBS Optimization for a Running Instance

You can enable or disable EBS optimization for a running instance by modifying its EBS-optimized instance attribute.

To enable EBS optimization for a running instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, click **Instances**, and select the instance.
3. Click **Actions**, select **Instance State**, and then click **Stop**.

Warning

When you stop an instance, the data on any instance store volumes is erased. Therefore, if you have any data on instance store volumes that you want to keep, be sure to back it up to persistent storage.

4. In the confirmation dialog box, click **Yes, Stop**. It can take a few minutes for the instance to stop.
5. With the instance still selected, click **Actions**, select **Instance Settings**, and then click **Change Instance Type**.
6. In the **Change Instance Type** dialog box, do one of the following:

- If the instance type of your instance is EBS-optimized by default, **EBS-optimized** is selected and you can't deselect it. You can click **Cancel**, because EBS optimization is already enabled for the instance.
 - If the instance type of your instance supports EBS optimization, select **EBS-optimized**, and then click **Apply**.
 - If the instance type of your instance does not support EBS optimization, **EBS-optimized** is deselected and you can't select it. You can select an instance type from **Instance Type** that supports EBS optimization, select **EBS-optimized**, and then click **Apply**.
7. Click **Actions**, select **Instance State**, and then click **Start**.

To enable EBS optimization for a running instance using the command line

You can use one of the following options with the corresponding command. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- `--ebs-optimized` with [modify-instance-attribute](#) (AWS CLI)
- `-EbsOptimized` with [Edit-EC2InstanceAttribute](#) (AWS Tools for Windows PowerShell)

Amazon EBS Encryption

Amazon EBS encryption offers you a simple encryption solution for your EBS volumes without the need for you to build, maintain, and secure your own key management infrastructure. When you create an encrypted EBS volume and attach it to a supported instance type, the following types of data are encrypted:

- Data at rest inside the volume
- All data moving between the volume and the instance
- All snapshots created from the volume

The encryption occurs on the servers that host EC2 instances, providing encryption of data-in-transit from EC2 instances to EBS storage.

Amazon EBS encryption uses AWS Key Management Service (AWS KMS) customer master keys (CMK) when creating encrypted volumes and any snapshots created from them. The first time you create an encrypted volume in a region, a default CMK is created for you automatically. This key is used for Amazon EBS encryption unless you select a CMK that you created separately using AWS KMS. Creating your own CMK gives you more flexibility, including the ability to create, rotate, and disable keys to define access controls, and to audit the encryption keys used to protect your data. For more information, see the [AWS Key Management Service Developer Guide](#).

This feature is supported with all EBS volume types (General Purpose SSD [[gp2](#)], Provisioned IOPS SSD [[io1](#)], Throughput Optimized HDD [[st1](#)], Cold HDD [[sc1](#)], and Magnetic [[standard](#)]), and you can expect the same IOPS performance on encrypted volumes as you would with unencrypted volumes, with a minimal effect on latency. You can access encrypted volumes the same way that you access unencrypted volumes; encryption and decryption are handled transparently and they require no additional action from you, your EC2 instance, or your application.

Snapshots that are taken from encrypted volumes are automatically encrypted. Volumes that are created from encrypted snapshots are also automatically encrypted. Public snapshots of encrypted volumes are not supported, but you can share an encrypted snapshot with specific accounts if you take the following steps:

1. Use a custom CMK, not your default CMK, to encrypt your volume.
2. Give the specific accounts access to the custom CMK.

3. Create the snapshot.
4. Give the specific accounts access to the snapshot.

For more information, see [Sharing an Amazon EBS Snapshot](#).

Amazon EBS encryption is only available on certain instance types. You can attach both encrypted and unencrypted volumes to a supported instance type. For more information, see [Supported Instance Types](#) (p. 800).

Contents

- [Encryption Key Management](#) (p. 800)
- [Supported Instance Types](#) (p. 800)
- [Changing the Encryption State of Your Data](#) (p. 801)
- [Amazon EBS Encryption and CloudWatch Events](#) (p. 803)

Encryption Key Management

Amazon EBS encryption handles key management for you. Each newly-created volume is encrypted with a unique 256-bit key; any snapshots of this volume and any subsequent volumes created from those snapshots also share that key. These keys are protected by our own key management infrastructure, which implements strong logical and physical security controls to prevent unauthorized access. Your data and associated keys are encrypted using the industry-standard AES-256 algorithm.

You cannot change the CMK that is associated with an existing snapshot or encrypted volume. However, you can associate a different CMK during a snapshot copy operation (including encrypting a copy of an unencrypted snapshot) and the resulting copied snapshot will use the new CMK.

Amazon's overall key management infrastructure uses Federal Information Processing Standards (FIPS) 140-2 approved cryptographic algorithms and is consistent with National Institute of Standards and Technology (NIST) 800-57 recommendations.

Each AWS account has a unique master key that is stored completely separate from your data, on a system that is surrounded with strong physical and logical security controls. Each encrypted volume (and its subsequent snapshots) is encrypted with a unique volume encryption key that is then encrypted with a region-specific secure master key. The volume encryption keys are used in memory on the server that hosts your EC2 instance; they are never stored on disk in plain text.

Supported Instance Types

Amazon EBS encryption is available on the instance types listed in the table below. These instance types leverage the Intel AES New Instructions (AES-NI) instruction set to provide faster and simpler data protection. You can attach both encrypted and unencrypted volumes to these instance types simultaneously.

Instance family	Instance types that support Amazon EBS encryption
General purpose	m3.medium m3.large m3.xlarge m3.2xlarge m4.large m4.xlarge m4.2xlarge m4.4xlarge m4.10xlarge m4.16xlarge t2.nano t2.micro t2.small t2.medium t2.large t2.xlarge t2.2xlarge
Compute optimized	c4.large c4.xlarge c4.2xlarge c4.4xlarge c4.8xlarge c3.large c3.xlarge c3.2xlarge c3.4xlarge c3.8xlarge
Memory optimized	r3.4xlarge r3.8xlarge r4.large r4.xlarge r3.large r3.xlarge r3.2xlarge r1.8xlarge

Instance family	Instance types that support Amazon EBS encryption
	r4.2xlarge r4.4xlarge r4.8xlarge r4.16xlarge x1.16xlarge x1.32xlarge
Storage optimized	d2.xlarge d2.2xlarge d2.4xlarge d2.8xlarge i2.xlarge i2.2xlarge i2.4xlarge i2.8xlarge
Accelerated computing	g2.2xlarge g2.8xlarge p2.xlarge p2.8xlarge p2.16xlarge

For more information about these instance types, see [Instance Type Details](#).

Changing the Encryption State of Your Data

There is no direct way to encrypt an existing unencrypted volume, or to remove encryption from an encrypted volume. However, you can migrate data between encrypted and unencrypted volumes. You can also apply a new encryption status while copying a snapshot:

- While copying an unencrypted snapshot of an unencrypted volume, you can encrypt the copy. Volumes restored from this encrypted copy will also be encrypted.
- While copying an encrypted snapshot of an encrypted volume, you can re-encrypt the copy using a different CMK. Volumes restored from the encrypted copy will only be accessible using the newly applied CMK.

Migrate Data between Encrypted and Unencrypted Volumes

When you have access to both an encrypted and unencrypted volume, you can freely transfer data between them. EC2 carries out the encryption or decryption operations transparently.

To migrate data between encrypted and unencrypted volumes

1. Create your destination volume (encrypted or unencrypted, depending on your need) by following the procedures in [Creating an Amazon EBS Volume \(p. 761\)](#).
2. Attach the destination volume to the instance that hosts the data to migrate. For more information, see [Attaching an Amazon EBS Volume to an Instance \(p. 766\)](#).
3. Make the destination volume available by following the procedures in [Making an Amazon EBS Volume Available for Use \(p. 767\)](#). For Linux instances, you can create a mount point at `/mnt/destination` and mount the destination volume there.
4. Copy the data from your source directory to the destination volume. It may be most convenient to use a bulk-copy utility for this.

Linux

Use the **rsync** command as follows to copy the data from your source to the destination volume. In this example, the source data is located in `/mnt/source` and the destination volume is mounted at `/mnt/destination`.

```
[ec2-user ~]$ sudo rsync -avh --progress /mnt/source/ /mnt/destination/
```

Windows

At a command prompt, use the **robocopy** command to copy the data from your source to the destination volume. In this example, the source data is located in `D:\` and the destination volume is mounted at `E:\`.


```
PS C:\Users\Administrator> robocopy D:\ E:\ /e /copyall /eta
```

Apply Encryption While Copying a Snapshot

Because you can apply encryption to a snapshot while copying it, another path to encrypting your data is the following procedure.

To encrypt a volume's data by means of snapshot copying

1. Create a snapshot of your unencrypted EBS volume. This snapshot is also unencrypted.
2. Copy the snapshot while applying encryption parameters. The resulting target snapshot is encrypted.
3. Restore the encrypted snapshot to a new volume, which is also encrypted.

For more information, see [Copying an Amazon EBS Snapshot](#).

Re-Encrypt a Snapshot with a New CMK

The ability to encrypt a snapshot during copying also allows you to re-encrypt an already-encrypted snapshot that you own. In this operation, the plaintext of your snapshot will be encrypted using a new CMK that you provide. Volumes restored from the resulting copy will only be accessible using the new CMK.

In a related scenario, you may choose to re-encrypt a snapshot that has been shared with you. Before you can restore a volume from a shared encrypted snapshot, you must create your own copy of it. By default, the copy will be encrypted with the key shared by the snapshot's owner. However, we recommend that you re-encrypt the snapshot during the copy process with a different key that you control. This protects your access to the volume if the original key is compromised, or if the owner revokes the key for any reason.

The following procedure demonstrates how to re-encrypt a snapshot that you own.

To re-encrypt a snapshot using the console

1. Create a custom CMK. For more information, see [AWS Key Management Service Developer Guide](#).
2. Create an EBS volume encrypted with (for this example) your default CMK.
3. Create a snapshot of your encrypted EBS volume. This snapshot is also encrypted with your default CMK.
4. On the **Snapshots** page, choose **Actions**, then choose **Copy**.
5. In the **Copy Snapshot** window, supply the complete ARN for your custom CMK (in the form `arn:aws:kms:us-east-1:012345678910:key/abcd1234-a123-456a-a12b-a123b4cd56ef`) in the **Master Key** field, or choose it from the menu. Click **Copy**.

The resulting copy of the snapshot—and all volumes restored from it—will be encrypted with your custom CMK.

The following procedure demonstrates how to re-encrypt a shared encrypted snapshot as you copy it. For this to work, you need access permissions to both the shared encrypted snapshot and to the CMK that encrypted it.

To copy and re-encrypt a shared snapshot using the console

1. Choose the shared encrypted snapshot on the **Snapshots** page, choose **Actions**, then choose **Copy**.

2. In the **Copy Snapshot** window, supply the complete ARN for a CMK that you own (in the form `arn:aws:kms:us-east-1:012345678910:key/abcd1234-a123-456a-a12b-a123b4cd56ef`) in the **Master Key** field, or choose it from the menu. Click **Copy**.

The resulting copy of the snapshot—and all volumes restored from it—will be encrypted with the CMK that you supplied. Changes to the original shared snapshot, its encryption status, or the shared CMK will have no effect on your copy.

For more information, see [Copying an Amazon EBS Snapshot](#).

Amazon EBS Encryption and CloudWatch Events

EBS supports Amazon CloudWatch Events for certain encryption-related scenarios. For more information, see [Amazon CloudWatch Events for Amazon EBS](#).

Amazon EBS Volume Performance on Windows Instances

Several factors, including I/O characteristics and the configuration of your instances and volumes, can affect the performance of Amazon EBS. Customers who follow the guidance on our Amazon EBS and Amazon EC2 product detail pages typically achieve good performance out of the box. However, there are some cases where you may need to do some tuning in order to achieve peak performance on the platform. This topic discusses general best practices as well as performance tuning that is specific to certain use cases. We recommend that you tune performance with information from your actual workload, in addition to benchmarking, to determine your optimal configuration. After you learn the basics of working with EBS volumes, it's a good idea to look at the I/O performance you require and at your options for increasing Amazon EBS performance to meet those requirements.

Contents

- [Amazon EBS Performance Tips \(p. 803\)](#)
- [Amazon EC2 Instance Configuration \(p. 805\)](#)
- [I/O Characteristics and Monitoring \(p. 808\)](#)
- [Initializing Amazon EBS Volumes \(p. 810\)](#)
- [RAID Configuration on Windows \(p. 811\)](#)

Amazon EBS Performance Tips

These tips represent best practices for getting optimal performance from your EBS volumes in a variety of user scenarios.

Use EBS-Optimized Instances

On instances without support for EBS-optimized throughput, network traffic can contend with traffic between your instance and your EBS volumes; on EBS-optimized instances, the two types of traffic are kept separate. Some EBS-optimized instance configurations incur an extra cost (such as C3, R3, and M3), while others are always EBS-optimized at no extra cost (such as M4, C4, and D2). For more information, see [Amazon EC2 Instance Configuration \(p. 805\)](#).

Understand How Performance is Calculated

When you measure the performance of your EBS volumes, it is important to understand the units of measure involved and how performance is calculated. For more information, see [I/O Characteristics and Monitoring \(p. 808\)](#).

Understand Your Workload

There is a relationship between the maximum performance of your EBS volumes, the size and number of I/O operations, and the time it takes for each action to complete. Each of these factors (performance, I/O, and latency) affects the others, and different applications are more sensitive to one factor or another.

Be Aware of the Performance Penalty When Initializing Volumes from Snapshots

There is a significant increase in latency when you first access each block of data on a new EBS volume that was restored from a snapshot. You can avoid this performance hit by accessing each block prior to putting the volume into production. This process is called *initialization* (formerly known as pre-warming). For more information, see [Initializing Amazon EBS Volumes \(p. 810\)](#).

Factors That Can Degrade HDD Performance

When you create a snapshot of a Throughput Optimized HDD (*st1*) or Cold HDD (*sc1*) volume, performance may drop as far as the volume's baseline value while the snapshot is in progress. This behavior is specific to these volume types. Other factors that can limit performance include driving more throughput than the instance can support, the performance penalty encountered while initializing volumes restored from a snapshot, and excessive amounts of small, random I/O on the volume. For more information about calculating throughput for HDD volumes, see [Amazon EBS Volume Types](#).

Your performance can also be impacted if your application isn't sending enough I/O requests. This can be monitored by looking at your volume's queue length and I/O size. The queue length is the number of pending I/O requests from your application to your volume. For maximum consistency, HDD-backed volumes must maintain a queue length (rounded to the nearest whole number) of 4 or more when performing 1 MiB sequential I/O. For more information about ensuring consistent performance of your volumes, see [I/O Characteristics and Monitoring \(p. 808\)](#).

Increase Read-Ahead for High-Throughput, Read-Heavy Workloads on *st1* and *sc1*

Some workloads are read-heavy and access the block device through the operating system page cache (for example, from a file system). In this case, to achieve the maximum throughput, we recommend that you configure the read-ahead setting to 1 MiB. This is a per-block-device setting that should only be applied to your HDD volumes. The following examples assume that you are on an Amazon Linux instance.

To examine the current value of read-ahead for your block devices, use the following command:

```
[ec2-user ~]$ sudo blockdev --report /dev/<device>
```

Block device information is returned in the following format:

RO	RA	SSZ	BSZ	StartSec	Size	Device
rw	256	512	4096	4096	8587820544	/dev/<device>

The device shown reports a read-ahead value of 256 bytes (the default). Multiply this number by the sector size (512 bytes) to obtain the size of the read-ahead buffer, which in this case is 128 KiB. To set the buffer value to 1 MiB, use the following command:

```
[ec2-user ~]$ sudo blockdev --setra 2048 /dev/<device>
```

Verify that the read-ahead setting now displays 2,048 by running the first command again.

Only use this setting when your workload consists of large, sequential I/Os. If it consists mostly of small, random I/Os, this setting will actually degrade your performance. In general, if your workload consists mostly of small or random I/Os, you should consider using a General Purpose SSD (`gp2`) volume rather than `st1` or `sc1`.

Use RAID 0 to Maximize Utilization of Instance Resources

Some instance types can drive more I/O throughput than what you can provision for a single EBS volume. You can join multiple `gp2`, `io1`, `st1`, or `sc1` volumes together in a RAID 0 configuration to use the available bandwidth for these instances. For more information, see [RAID Configuration on Windows \(p. 811\)](#).

Track Performance with Amazon CloudWatch

Amazon Web Services provides performance metrics for Amazon EBS that you can analyze and view with Amazon CloudWatch and status checks that you can use to monitor the health of your volumes. For more information, see [Monitoring the Status of Your Volumes \(p. 769\)](#).

Amazon EC2 Instance Configuration

When you plan and configure EBS volumes for your application, it is important to consider the configuration of the instances that you will attach the volumes to. In order to get the most performance out of your EBS volumes, you should attach them to an instance with enough bandwidth to support your volumes, such as an EBS-optimized instance or an instance with 10 Gigabit network connectivity. This is especially important when you stripe multiple volumes together in a RAID configuration.

Use EBS-Optimized or 10 Gigabit Network Instances

Any performance-sensitive workloads that require minimal variability and dedicated Amazon EC2 to Amazon EBS traffic, such as production databases or business applications, should use volumes that are attached to an EBS-optimized instance or an instance with 10 Gigabit network connectivity. EC2 instances that do not meet this criteria offer no guarantee of network resources. The only way to ensure sustained reliable network bandwidth between your EC2 instance and your EBS volumes is to launch the EC2 instance as EBS-optimized or choose an instance type with 10 Gigabit network connectivity. To see which instance types include 10 Gigabit network connectivity, see [Instance Type Details](#). For information about configuring EBS-optimized instances, see [Amazon EBS—Optimized Instances](#).

Choose an EC2 Instance with Enough Bandwidth

Launching an instance that is EBS-optimized provides you with a dedicated connection between your EC2 instance and your EBS volume. However, it is still possible to provision EBS volumes that exceed the available bandwidth for certain instance types, especially when multiple volumes are striped in a RAID configuration. The following table shows which instance types are available to be launched as EBS-optimized, the dedicated throughput to instance types are available to be launched as EBS-optimized, the dedicated bandwidth to Amazon EBS, the maximum amount of IOPS the instance can support if you are using a 16 KB I/O size, and the approximate I/O bandwidth available on that connection in MB/s. Be sure to choose an EBS-optimized instance that provides more dedicated EBS throughput than your application needs; otherwise, the Amazon EBS to Amazon EC2 connection will become a performance bottleneck.

Note

The table below and the following examples use 16 KB as an I/O size for explanatory purposes only; your application I/O size may vary (Amazon EBS measures each I/O operation per second that is 256 KiB or smaller as one IOPS). For more information about IOPS and the relationship between I/O size and volume throughput limits, see [I/O Characteristics and Monitoring \(p. 808\)](#).

Amazon Elastic Compute Cloud
User Guide for Windows Instances
EBS Performance

Instance type	EBS-optimized by default	Max. bandwidth (Mbps)*	Expected throughput (MB/s)**	Max. IOPS (16 KB I/O size)**
c1.xlarge		1,000	125	8,000
c3.xlarge		500	62.5	4,000
c3.2xlarge		1,000	125	8,000
c3.4xlarge		2,000	250	16,000
c4.large	Yes	500	62.5	4,000
c4.xlarge	Yes	750	93.75	6,000
c4.2xlarge	Yes	1,000	125	8,000
c4.4xlarge	Yes	2,000	250	16,000
c4.8xlarge	Yes	4,000	500	32,000
d2.xlarge	Yes	750	93.75	6,000
d2.2xlarge	Yes	1,000	125	8,000
d2.4xlarge	Yes	2,000	250	16,000
d2.8xlarge	Yes	4,000	500	32,000
g2.2xlarge		1,000	125	8,000
i2.xlarge		500	62.5	4,000
i2.2xlarge		1,000	125	8,000
i2.4xlarge		2,000	250	16,000
m1.large		500	62.5	4,000
m1.xlarge		1,000	125	8,000
m2.2xlarge		500	62.5	4,000
m2.4xlarge		1,000	125	8,000
m3.xlarge		500	62.5	4,000
m3.2xlarge		1,000	125	8,000
m4.large	Yes	450	56.25	3,600
m4.xlarge	Yes	750	93.75	6,000
m4.2xlarge	Yes	1,000	125	8,000
m4.4xlarge	Yes	2,000	250	16,000
m4.10xlarge	Yes	4,000	500	32,000
m4.16xlarge	Yes	10,000	1,250	65,000
p2.xlarge	Yes	750	93.75	6,000

Amazon Elastic Compute Cloud
User Guide for Windows Instances
EBS Performance

Instance type	EBS-optimized by default	Max. bandwidth (Mbps)*	Expected throughput (MB/s)**	Max. IOPS (16 KB I/O size)**
p2.8xlarge	Yes	5,000	625	32,500
p2.16xlarge	Yes	10,000	1,250	65,000
r3.xlarge		500	62.5	4,000
r3.2xlarge		1,000	125	8,000
r3.4xlarge		2,000	250	16,000
r4.large	Yes	400	50	3,000
r4.xlarge	Yes	800	100	6,000
r4.2xlarge	Yes	1600	200	12,000
r4.4xlarge	Yes	3000	400	16,000
r4.8xlarge	Yes	6000	800	32,000
r4.16xlarge	Yes	12,000	1500	65,000
x1.16xlarge	Yes	5,000	625	32,500
x1.32xlarge	Yes	10,000	1,250	65,000

* These instance types must be launched as EBS-optimized to consistently achieve this level of performance.

** This value is a rounded approximation based on a 100% read-only workload and it is provided as a baseline configuration aid. EBS-optimized connections are full-duplex, and can drive more throughput and IOPS in a 50/50 read/write workload where both communication lanes are used. In some cases, network, file system, and Amazon EBS encryption overhead can reduce the maximum throughput and IOPS available.

Note that some instances with 10-gigabit network interfaces, such as `i2.8xlarge`, `c3.8xlarge`, and `r3.8xlarge`, do not offer EBS-optimization, and therefore do not have dedicated EBS bandwidth available and are not listed here. However, you can use all of that bandwidth for traffic to Amazon EBS if your application isn't pushing other network traffic that contends with Amazon EBS. Some other 10-gigabit network instances, such as `c4.8xlarge` and `d2.8xlarge` offer dedicated Amazon EBS bandwidth in addition to a 10-gigabit interface which is used exclusively for network traffic.

The `m1.large` instance has a maximum 16 KB IOPS value of 4,000, but unless this instance type is launched as EBS-optimized, that value is an absolute best-case scenario and is not guaranteed; to consistently achieve 4,000 16 KB IOPS, you must launch this instance as EBS-optimized. However, if a 4,000 IOPS `io1` volume is attached to an EBS-optimized `m1.large` instance, the Amazon EC2 to Amazon EBS connection bandwidth limit prevents this volume from providing the 320 MB/s maximum aggregate throughput available to it. In this case, we must use an EBS-optimized EC2 instance that supports at least 320 MB/s of throughput, such as the `c4.8xlarge` instance type.

Volumes of type General Purpose SSD (`gp2`) have a throughput limit between 128 MB/s and 160 MB/s per volume (depending on volume size), which pairs well with a 1,000 Mbps EBS-optimized connection. Instance types that offer more than 1,000 Mbps of throughput to Amazon EBS can use more than one `gp2` volume to take advantage of the available throughput. Volumes of type Provisioned IOPS SSD (`io1`) have a throughput limit range of 256 KiB for each IOPS provisioned,

up to a maximum of 320 MiB/s (at 1,280 IOPS). For more information, see [Amazon EBS Volume Types \(p. 749\)](#).

Instance types with 10 Gigabit network connectivity support up to 800 MB/s of throughput and 48,000 16K IOPS for unencrypted Amazon EBS volumes and up to 25,000 16K IOPS for encrypted Amazon EBS volumes. Because the maximum `io1` value for EBS volumes is 20,000 for `io1` volumes and 10,000 for `gp2` volumes, you can use several EBS volumes simultaneously to reach the level of I/O performance available to these instance types. For more information about which instance types include 10 Gigabit network connectivity, see [Instance Type Details](#).

You should use EBS-optimized instances when available to get the full performance benefits of Amazon EBS `gp2` and `io1` volumes. For more information, see [Amazon EBS-Optimized Instances \(p. 795\)](#).

I/O Characteristics and Monitoring

On a given volume configuration, certain I/O characteristics drive the performance behavior for your EBS volumes. SSD-backed volumes—General Purpose SSD (`gp2`) and Provisioned IOPS SSD (`io1`)—deliver consistent performance whether an I/O operation is random or sequential. HDD-backed volumes—Throughput Optimized HDD (`st1`) and Cold HDD (`sc1`)—deliver optimal performance only when I/O operations are large and sequential. To understand how SSD and HDD volumes will perform in your application, it is important to know the connection between demand on the volume, the quantity of IOPS available to it, the time it takes for an I/O operation to complete, and the volume's throughput limits.

IOPS

IOPS are a unit of measure representing input/output operations per second. The operations are measured in KiB, and the underlying drive technology determines the maximum amount of data that a volume type counts as a single I/O. I/O size is capped at 256 KiB for SSD volumes and 1,024 KiB for HDD volumes because SSD volumes handle small or random I/O much more efficiently than HDD volumes.

When small I/O operations are physically contiguous, Amazon EBS attempts to merge them into a single I/O up to the maximum size. For example, for SSD volumes, a single 1,024 KiB I/O operation counts as 4 operations ($1,024 \div 256 = 4$), while 8 contiguous I/O operations at 32 KiB each count as 1 operation ($8 \times 32 = 256$). However, 8 random I/O operations at 32 KiB each count as 8 operations. Each I/O operation under 32 KiB counts as 1 operation.

Similarly, for HDD-backed volumes, both a single 1,024 KiB I/O operation and 8 sequential 128 KiB operations would count as one operation. However, 8 random 128 KiB I/O operations would count as 8 operations.

Consequently, when you create an SSD-backed volume supporting 3,000 IOPS (either by provisioning an `io1` volume at 3,000 IOPS or by sizing a `gp2` volume at 1000 GiB), and you attach it to an EBS-optimized instance that can provide sufficient bandwidth, you can transfer up to 3,000 I/Os of data per second, with throughput determined by I/O size.

Volume Queue Length and Latency

The volume queue length is the number of pending I/O requests for a device. Latency is the true end-to-end client time of an I/O operation, in other words, the time elapsed between sending an I/O to EBS and receiving an acknowledgement from EBS that the I/O read or write is complete. Queue length must be correctly calibrated with I/O size and latency to avoid creating bottlenecks either on the guest operating system or on the network link to EBS.

Optimal queue length varies for each workload, depending on your particular application's sensitivity to IOPS and latency. If your workload is not delivering enough I/O requests to fully use the performance available to your EBS volume, then your volume might not deliver the IOPS or throughput that you have provisioned.

Transaction-intensive applications are sensitive to increased I/O latency and are well-suited for SSD-backed `io1` and `gp2` volumes. You can maintain high IOPS while keeping latency down by maintaining a low queue length and a high number of IOPS available to the volume. Consistently driving more IOPS to a volume than it has available can cause increased I/O latency.

Throughput-intensive applications are less sensitive to increased I/O latency, and are well-suited for HDD-backed `st1` and `sc1` volumes. You can maintain high throughput to HDD-backed volumes by maintaining a high queue length when performing large, sequential I/O.

I/O size and volume throughput limits

For SSD-backed volumes, if your I/O size is very large, you may experience a smaller number of IOPS than you provisioned because you are hitting the throughput limit of the volume. For example, a `gp2` volume under 1000 GiB with burst credits available has an IOPS limit of 3,000 and a volume throughput limit of 160 MiB/s. If you are using a 256 KiB I/O size, your volume reaches its throughput limit at 640 IOPS (640 x 256 KiB = 160 MiB). For smaller I/O sizes (such as 16 KiB), this same volume can sustain 3,000 IOPS because the throughput is well below 160 MiB/s. (These examples assume that your volume's I/O is not hitting the throughput limits of the instance.) For more information about the throughput limits for each EBS volume type, see [Amazon EBS Volume Types \(p. 749\)](#).

For smaller I/O operations, you may see a higher-than-provisioned IOPS value as measured from inside your instance. This happens when the instance operating system merges small I/O operations into a larger operation before passing them to Amazon EBS.

If your workload uses sequential I/Os on HDD-backed `st1` and `sc1` volumes, you may experience a higher than expected number of IOPS as measured from inside your instance. This happens when the instance operating system merges sequential I/Os and counts them in 1,024 KiB-sized units. If your workload uses small or random I/Os, you may experience a lower throughput than you expect. This is because we count each random, non-sequential I/O toward the total IOPS count, which can cause you to hit the volume's IOPS limit sooner than expected.

Whatever your EBS volume type, if you are not experiencing the IOPS or throughput you expect in your configuration, ensure that your EC2 instance bandwidth is not the limiting factor. You should always use a current-generation, EBS-optimized instance (or one that includes 10 Gb/s network connectivity) for optimal performance. For more information, see [Amazon EC2 Instance Configuration \(p. 805\)](#). Another possible cause for not experiencing the expected IOPS is that you are not driving enough I/O to the EBS volumes.

Monitor I/O Characteristics with CloudWatch

You can monitor these I/O characteristics with each volume's [CloudWatch metrics](#). Important metrics to consider include:

- `BurstBalance`
- `VolumeReadBytes`
- `VolumeWriteBytes`
- `VolumeReadOps`
- `VolumeWriteOps`
- `VolumeQueueLength`

`BurstBalance` displays the burst bucket balance for `gp2`, `st1`, and `sc1` volumes as a percentage of the remaining balance. When your burst bucket is depleted, volume I/O credits (for `gp2` volumes) or volume throughput credits (for `st1` and `sc1` volumes) is throttled to the baseline. Check the `BurstBalance` value to determine whether your volume is being throttled for this reason.

HDD-backed `st1` and `sc1` volumes are designed to perform best with workloads that take advantage of the 1,024 KiB maximum I/O size. To determine your volume's average I/O size, divide

`VolumeWriteBytes` by `VolumeWriteOps`. The same calculation applies to read operations. If average I/O size is below 64 KiB, increasing the size of the I/O operations sent to an `st1` or `sc1` volume should improve performance.

Note

If average I/O size is at or near 44 KiB, you may be using an instance or kernel without support for indirect descriptors. Any Linux kernel 3.8 and above has this support, as well as any current-generation instance.

If your I/O latency is higher than you require, check `VolumeQueueLength` to make sure your application is not trying to drive more IOPS than you have provisioned. If your application requires a greater number of IOPS than your volume can provide, you should consider using a larger `gp2` volume with a higher base performance level or an `io1` volume with more provisioned IOPS to achieve faster latencies.

For more information about Amazon EBS I/O characteristics, see the [Amazon EBS: Designing for Performance](#) re:Invent presentation on this topic.

Initializing Amazon EBS Volumes

New EBS volumes receive their maximum performance the moment that they are available and do not require initialization (formerly known as pre-warming). However, storage blocks on volumes that were restored from snapshots must be initialized (pulled down from Amazon S3 and written to the volume) before you can access the block. This preliminary action takes time and can cause a significant increase in the latency of an I/O operation the first time each block is accessed. For most applications, amortizing this cost over the lifetime of the volume is acceptable. Performance is restored after the data is accessed once.

You can avoid this performance hit in a production environment by reading from all of the blocks on your volume before you use it; this process is called *initialization*. For a new volume created from a snapshot, you should read all the blocks that have data before using the volume.

Important

While initializing `io1` volumes that were restored from snapshots, the performance of the volume may drop below 50 percent of its expected level, which causes the volume to display a `warning` state in the **I/O Performance** status check. This is expected, and you can ignore the `warning` state on `io1` volumes while you are initializing them. For more information, see [Monitoring Volumes with Status Checks \(p. 772\)](#).

Initializing Amazon EBS Volumes on Windows

New EBS volumes receive their maximum performance the moment that they are available and do not require initialization (formerly known as pre-warming). For volumes that have been restored from snapshots, use `dd` for Windows to read from all of the blocks on a volume. All existing data on the volume will be preserved.

To install `dd` for Windows

The `dd` for the Windows program provides a similar experience to the `dd` program that is commonly available for Linux and Unix systems, and it allows you to initialize Amazon EBS volumes that have been restored from snapshots. At the time of this writing, the most recent beta version contains the `/dev/null` virtual device that is required to initialize volumes restored from snapshots. Full documentation for the program is available at <http://www.chrysocome.net/dd>.

1. Download the most recent binary version of `dd` for Windows from <http://www.chrysocome.net/dd>. You must use version 0.6 beta 3 or newer to initialize restored volumes.
2. (Optional) Create a folder for command line utilities that is easy to locate and remember, such as `C:\bin`. If you already have a designated folder for command line utilities, you can use that folder instead in the following step.

3. Unzip the binary package and copy the `dd.exe` file to your command line utilities folder (for example, `C:\bin`).
4. Add the command line utilities folder to your `Path` environment variable so you can execute the programs in that folder from anywhere.

Important

The following steps don't update the environment variables in your current command prompt windows. The command prompt windows that you open after you complete these steps will contain the updates. This is why it's necessary for you to open a new command prompt window to verify that your environment is set up properly.

- a. Choose **Start**, open the context (right-click) menu for **Computer**, and then choose **Properties**.
- b. Choose **Advanced system settings, Environment Variables**.
- c. For **System Variables**, select the variable **Path** and choose **Edit**.
- d. For **Variable value**, append a semicolon and the location of your command line utility folder (`;C:\bin\`) to the end of the existing value.
- e. Choose **OK** to close the **Edit System Variable** window.

To initialize a volume using `dd` for Windows

1. Use the `wmic` command to list the available disks on your system.

```
C:\>wmic diskdrive get size,deviceid
DeviceID          Size
\\.\PHYSICALDRIVE2 80517265920
\\.\PHYSICALDRIVE1 80517265920
\\.\PHYSICALDRIVE0 128849011200
\\.\PHYSICALDRIVE3 107372805120
```

Identify the disk to initialize in the following steps. The `C:` drive is on `\\.\PHYSICALDRIVE0`. You can use the `diskmgmt.msc` utility to compare drive letters to disk drive numbers if you are not sure which drive number to use.

2. Execute the following command to read all blocks on the specified device (and send the output to the `/dev/null` virtual device). This command safely initializes your existing data.

```
C:\>dd if=\\.\PHYSICALDRIVE $n$  of=/dev/null bs=1M --progress --size
```

Note

You may see an error if `dd` attempts to read beyond the end of the volume. This can be safely ignored.

3. When the operation completes, you are ready to use your new volume. For more information, see [Making an Amazon EBS Volume Available for Use \(p. 767\)](#).

RAID Configuration on Windows

With Amazon EBS, you can use any of the standard RAID configurations that you can use with a traditional bare metal server, as long as that particular RAID configuration is supported by the operating system for your instance. This is because all RAID is accomplished at the software level. For greater I/O performance than you can achieve with a single volume, RAID 0 can stripe multiple volumes together; for on-instance redundancy, RAID 1 can mirror two volumes together.

Amazon EBS volume data is replicated across multiple servers in an Availability Zone to prevent the loss of data from the failure of any single component. This replication makes Amazon EBS volumes

ten times more reliable than typical commodity disk drives. For more information, see [Amazon EBS Availability and Durability](#) in the Amazon EBS product detail pages.

Note

You should avoid booting from a RAID volume. Grub is typically installed on only one device in a RAID array, and if one of the mirrored devices fails, you may be unable to boot the operating system.

If you need to create a RAID array on a Linux instance, see [RAID Configuration on Linux](#) in the *Amazon EC2 User Guide for Linux Instances*.

Contents

- [RAID Configuration Options \(p. 812\)](#)
- [Creating a RAID Array on Windows \(p. 813\)](#)

RAID Configuration Options

The following table compares the common RAID 0 and RAID 1 options.

Configuratio	Use	Advantages	Disadvantages
RAID 0	When I/O performance is more important than fault tolerance; for example, as in a heavily used database (where data replication is already set up separately).	I/O is distributed across the volumes in a stripe. If you add a volume, you get the straight addition of throughput.	Performance of the stripe is limited to the worst performing volume in the set. Loss of a single volume results in a complete data loss for the array.
RAID 1	When fault tolerance is more important than I/O performance; for example, as in a critical application.	Safer from the standpoint of data durability.	Does not provide a write performance improvement; requires more Amazon EC2 to Amazon EBS bandwidth than non-RAID configurations because the data is written to multiple volumes simultaneously.

Important

RAID 5 and RAID 6 are not recommended for Amazon EBS because the parity write operations of these RAID modes consume some of the IOPS available to your volumes. Depending on the configuration of your RAID array, these RAID modes provide 20-30% fewer usable IOPS than a RAID 0 configuration. Increased cost is a factor with these RAID modes as well; when using identical volume sizes and speeds, a 2-volume RAID 0 array can outperform a 4-volume RAID 6 array that costs twice as much.

Creating a RAID 0 array allows you to achieve a higher level of performance for a file system than you can provision on a single Amazon EBS volume. A RAID 1 array offers a "mirror" of your data for extra redundancy. Before you perform this procedure, you need to decide how large your RAID array should be and how many IOPS you want to provision.

The resulting size of a RAID 0 array is the sum of the sizes of the volumes within it, and the bandwidth is the sum of the available bandwidth of the volumes within it. The resulting size and bandwidth of a RAID 1 array is equal to the size and bandwidth of the volumes in the array. For example, two 500 GiB Amazon EBS volumes with 4,000 provisioned IOPS each will create a 1000 GiB RAID 0 array with an available bandwidth of 8,000 IOPS and 640 MB/s of throughput or a 500 GiB RAID 1 array with an available bandwidth of 4,000 IOPS and 320 MB/s of throughput.

This documentation provides basic RAID setup examples. For more information about RAID configuration, performance, and recovery, see the Linux RAID Wiki at https://raid.wiki.kernel.org/index.php/Linux_Raid.

Creating a RAID Array on Windows

Use the following procedure to create the RAID array. Note that you can get directions for Linux instances from [Creating a RAID Array on Linux](#) in the *Amazon EC2 User Guide for Linux Instances*.

To create a RAID array on Windows

1. Create the Amazon EBS volumes for your array. For more information, see [Creating an Amazon EBS Volume](#) (p. 761).

Important

Create volumes with identical size and IOPS performance values for your array. Make sure you do not create an array that exceeds the available bandwidth of your EC2 instance. For more information, see [Amazon EC2 Instance Configuration](#) (p. 805).

2. Attach the Amazon EBS volumes to the instance that you want to host the array. For more information, see [Attaching an Amazon EBS Volume to an Instance](#) (p. 766).
3. Connect to your Windows instance. For more information, see [Connecting to Your Windows Instance](#) (p. 254).
4. Open a command prompt and type the **diskpart** command.

```
PS C:\Users\Administrator> diskpart

Microsoft DiskPart version 6.1.7601
Copyright (C) 1999-2008 Microsoft Corporation.
On computer: WIN-BM6QPPL51CO
```

5. At the DISKPART prompt, list the available disks with the following command.

```
DISKPART> list disk

Disk ###  Status              Size          Free           Dyn  Gpt
-----  -
Disk 0    Online              30 GB         0 B
Disk 1    Online              8 GB          0 B
Disk 2    Online              8 GB          0 B
Disk 3    Online              8 GB          0 B
Disk 4    Online              8 GB          0 B
Disk 5    Online             419 GB        0 B
Disk 6    Online             419 GB        0 B
```

Identify the disks you want to use in your array and take note of their disk numbers.

6. Each disk you want to use in your array must be an online dynamic disk that does not contain any existing volumes. Use the following steps to convert basic disks to dynamic disks and to delete any existing volumes.
 - a. Select a disk you want to use in your array with the following command, substituting *n* with your disk number.

```
DISKPART> select disk n

Disk n is now the selected disk.
```

- b. If the selected disk is listed as *Offline*, bring it online by running the **online disk** command.

- c. If the selected disk does not have an asterisk in the `Dyn` column in the previous **list disk** command output, you need to convert it to a dynamic disk.

```
DISKPART> convert dynamic
```

Note

If you receive an error that the disk is write protected, you can clear the read-only flag with the **ATTRIBUTE DISK CLEAR READONLY** command and then try the dynamic disk conversion again.

- d. Use the **detail disk** command to check for existing volumes on the selected disk.

```
DISKPART> detail disk

XENSRC PVDISK SCSI Disk Device
Disk ID: 2D8BF659
Type      : SCSI
Status    : Online
Path      : 0
Target    : 1
LUN ID    : 0
Location Path : PCIROOT(0)#PCI(0300)#SCSI(P00T01L00)
Current Read-only State : No
Read-only  : No
Boot Disk  : No
Pagefile Disk : No
Hibernation File Disk : No
Crashdump Disk : No
Clustered Disk : No

   Volume ###  Ltr  Label           Fs      Type          Size      Status
   -----  ---  -----  ---  ---  -----  ---
   Volume 2     D    NEW VOLUME     FAT32   Simple        8189 MB   Healthy
```

Note any volume numbers on the disk. In this example, the volume number is 2. If there are no volumes, you can skip the next step.

- e. (Only required if volumes were identified in the previous step) Select and delete any existing volumes on the disk that you identified in the previous step.

Warning

This destroys any existing data on the volume.

- i. Select the volume, substituting `n` with your volume number.

```
DISKPART> select volume n
Volume n is the selected volume.
```

- ii. Delete the volume.

```
DISKPART> delete volume

DiskPart successfully deleted the volume.
```

- iii. Repeat these substeps for each volume you need to delete on the selected disk.
- f. Repeat [Step 6 \(p. 813\)](#) for each disk you want to use in your array.

7. Verify that the disks you want to use are now dynamic.

```
DISKPART> list disk
```

Disk ###	Status	Size	Free	Dyn	Gpt
Disk 0	Online	30 GB	0 B		
Disk 1	Online	8 GB	0 B	*	
Disk 2	Online	8 GB	0 B	*	
Disk 3	Online	8 GB	0 B	*	
* Disk 4	Online	8 GB	0 B	*	
Disk 5	Online	419 GB	0 B		
Disk 6	Online	419 GB	0 B		

8. Create your raid array. On Windows, a RAID 0 volume is referred to as a striped volume and a RAID 1 volume is referred to as a mirrored volume.

(Striped volumes only) To create a striped volume array on disks 1 and 2, use the following command (note the `stripe` option to stripe the array):

```
DISKPART> create volume stripe disk=1,2
```

DiskPart successfully created the volume.

(Mirrored volumes only) To create a mirrored volume array on disks 3 and 4, use the following command (note the `mirror` option to mirror the array):

```
DISKPART> create volume mirror disk=3,4
```

DiskPart successfully created the volume.

9. Verify your new volume.

```
DISKPART> list volume
```

Volume ###	Ltr	Label	Fs	Type	Size	Status
Volume 0	C		NTFS	Partition	29 GB	Healthy
* Volume 1			RAW	Mirror	8190 MB	Healthy
Volume 2			RAW	Stripe	15 GB	Healthy
Volume 5	Z	Temporary S	NTFS	Partition	419 GB	Healthy
Volume 6	Y	Temporary S	NTFS	Partition	419 GB	Healthy

Note that for this example the `Type` column lists a `Mirror` volume and a `Stripe` volume.

10. Select and format your volume so that you can begin using it.

- a. Select the volume you want to format, substituting `n` with your volume number.

```
DISKPART> select volume n
```

Volume `n` is the selected volume.

- b. Format the volume.

Note

To perform a full format, omit the `quick` option.

```
DISKPART> format quick recommended label="My new volume"

100 percent completed

DiskPart successfully formatted the volume.
```

- c. Assign an available drive letter to your volume.

```
DISKPART> assign letter f

DiskPart successfully assigned the drive letter or mount point.
```

Your new volume is now ready to use.

Amazon CloudWatch Events for Amazon EBS

Amazon EBS emits notifications based on Amazon CloudWatch Events for a variety of snapshot and encryption status changes. With CloudWatch Events, you can establish rules that trigger programmatic actions in response to a change in snapshot or encryption key state. For example, when a snapshot is created, you can trigger an AWS Lambda function to share the completed snapshot with another account or copy it to another region for disaster-recovery purposes.

For more information, see [Using Events](#) in the *Amazon CloudWatch User Guide*.

Event Definitions and Examples

This section defines the supported Amazon EBS events and provides examples of event output for specific scenarios. Events in CloudWatch are represented as JSON objects. For more information about the format and content of event objects, see [Events and Event Patterns](#) in the *Amazon CloudWatch Events User Guide*.

The fields that are unique to EBS events are contained in the "detail" section of the JSON objects shown below. The "event" field contains the event name. The "result" field contains the completed status of the action that triggered the event.

Create Snapshot (`createSnapshot`)

The `createSnapshot` event is sent to your AWS account when an action to create a snapshot completes. This event can have a result of either `succeeded` or `failed`.

Event Data

The listing below is an example of a JSON object emitted by EBS for a successful `createSnapshot` event. The `source` field contains the ARN of the source volume. The `StartTime` and `EndTime` fields indicate when creation of the snapshot started and completed.

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "EBS Snapshot Notification",
```

```
"source": "aws.ec2",
"account": "012345678901",
"time": "yyyy-mm-ddTth:mm:ssZ",
"region": "us-east-1",
"resources": [
  "arn:aws:ec2::us-west-2:snapshot/snap-01234567"
],
"detail": {
  "event": "createSnapshot",
  "result": "succeeded",
  "cause": "",
  "request-id": "",
  "snapshot_id": "arn:aws:ec2::us-west-2:snapshot/snap-01234567",
  "source": "arn:aws:ec2::us-west-2:volume/vol-01234567",
  "StartTime": "yyyy-mm-ddTth:mm:ssZ",
  "EndTime": "yyyy-mm-ddTth:mm:ssZ" }
}
```

Copy Snapshot (copySnapshot)

The `copySnapshot` event is sent to your AWS account when an action to copy a snapshot completes. This event can have a result of either `succeeded` or `failed`.

Event Data

The listing below is an example of a JSON object emitted by EBS after a failed `copySnapshot` event. The cause for the failure was an invalid source snapshot ID. The value of `snapshot_id` is the ARN of the failed snapshot. The value of `source` is the ARN of the source snapshot. `StartTime` and `EndTime` represent when the copy-snapshot action started and ended.

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "EBS Snapshot Notification",
  "source": "aws.ec2",
  "account": "123456789012",
  "time": "yyyy-mm-ddTth:mm:ssZ",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2::us-west-2:snapshot/snap-01234567"
  ],
  "detail": {
    "event": "copySnapshot",
    "result": "failed",
    "cause": "Source snapshot ID is not valid",
    "request-id": "",
    "snapshot_id": "arn:aws:ec2::us-west-2:snapshot/snap-01234567",
    "source": "arn:aws:ec2::eu-west-1:snapshot/snap-76543210",
    "StartTime": "yyyy-mm-ddTth:mm:ssZ",
    "EndTime": "yyyy-mm-ddTth:mm:ssZ"
  }
}
```

Share Snapshot (shareSnapshot)

The `shareSnapshot` event is sent to your AWS account when another account shares a snapshot with it. The result is always `succeeded`.

Event Data

The listing below is an example of a JSON object emitted by EBS after a completed `shareSnapshot` event. The value of `source` is the AWS account number of the user that shared the snapshot with you. `StartTime` and `EndTime` represent when the share-snapshot action started and ended. The `shareSnapshot` event is emitted only when a private snapshot is shared with another user. Sharing a public snapshot does not trigger the event.

```
{
  "version": "0",
  "id": "01234567-01234-0123-0123-012345678901",
  "detail-type": "EBS Snapshot Notification",
  "source": "aws.ec2",
  "account": "012345678901",
  "time": "yyyy-mm-ddTth:mm:ssZ",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2::us-west-2:snapshot/snap-01234567"
  ],
  "detail": {
    "event": "shareSnapshot",
    "result": "succeeded",
    "cause": "",
    "request-id": "",
    "snapshot_id": "arn:aws:ec2::us-west-2:snapshot/snap-01234567",
    "source": "012345678901",
    "StartTime": "yyyy-mm-ddTth:mm:ssZ",
    "EndTime": "yyyy-mm-ddTth:mm:ssZ"
  }
}
```

Invalid Encryption Key on Volume Attach or Reattach (`attachVolume`, `reattachVolume`)

The `attachVolume` event is sent to your AWS account when it fails to attach or reattach a volume to an instance due to an invalid KMS key.

Note

You can use a KMS key to encrypt an EBS volume. If the key used to encrypt the volume becomes invalid, EBS will emit an event if that key is later used to create, attach, or reattach to a volume.

Event Data

The listing below is an example of a JSON object emitted by EBS after a failed `attachVolume` event. The cause for the failure was a KMS key pending deletion.

Note

AWS may attempt to reattach to a volume following routine server maintenance.

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "detail-type": "EBS Volume Notification",
  "source": "aws.ec2",
  "account": "012345678901",
  "time": "yyyy-mm-ddTth:mm:ssZ",
  "region": "us-east-1",
}
```

```
"resources": [
  "arn:aws:ec2:us-east-1:0123456789ab:volume/vol-01234567",
  "arn:aws:kms:us-
east-1:0123456789ab:key/01234567-0123-0123-0123-0123456789ab"
],
"detail": {
  "event": "attachVolume",
  "result": "failed",
  "cause": "arn:aws:kms:us-
east-1:0123456789ab:key/01234567-0123-0123-0123-0123456789ab is pending
deletion.",
  "request-id": ""
}
}
```

The listing below is an example of a JSON object emitted by EBS after a failed `reattachVolume` event. The cause for the failure was a KMS key pending deletion.

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "detail-type": "EBS Volume Notification",
  "source": "aws.ec2",
  "account": "012345678901",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2:us-east-1:0123456789ab:volume/vol-01234567",
    "arn:aws:kms:us-
east-1:0123456789ab:key/01234567-0123-0123-0123-0123456789ab"
  ],
  "detail": {
    "event": "reattachVolume",
    "result": "failed",
    "cause": "arn:aws:kms:us-
east-1:0123456789ab:key/01234567-0123-0123-0123-0123456789ab is pending
deletion.",
    "request-id": ""
  }
}
```

Invalid Encryption Key on Create Volume (`createVolume`)

The `createVolume` event is sent to your AWS account when it fails to create a volume due to an invalid KMS key.

Note

You can use a KMS key to encrypt an EBS volume. If the key used to encrypt the volume becomes invalid, EBS will emit an event if that key is later used to create, attach, or reattach to a volume.

Event Data

The listing below is an example of a JSON object emitted by EBS after a failed `createVolume` event. The cause for the failure was a disabled KMS key.

```
{
```

```
"version": "0",
"id": "01234567-0123-0123-0123-0123456789ab",
"detail-type": "EBS Volume Notification",
"source": "aws.ec2",
"account": "012345678901",
"time": "yyyy-mm-ddThh:mm:ssZ",
"region": "sa-east-1",
"resources": [
  "arn:aws:ec2:sa-east-1:0123456789ab:volume/vol-01234567",
],
"detail": {
  "event": "createVolume",
  "result": "failed",
  "cause": "arn:aws:kms:sa-east-1:0123456789ab:key/01234567-0123-0123-0123-0123456789ab is disabled.",
  "request-id": "01234567-0123-0123-0123-0123456789ab",
}
```

The following is an example of a JSON object that is emitted by EBS after a failed `createVolume` event. The cause for the failure was a KMS key pending import.

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "detail-type": "EBS Volume Notification",
  "source": "aws.ec2",
  "account": "012345678901",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "sa-east-1",
  "resources": [
    "arn:aws:ec2:sa-east-1:0123456789ab:volume/vol-01234567",
  ],
  "detail": {
    "event": "createVolume",
    "result": "failed",
    "cause": "arn:aws:kms:sa-east-1:0123456789ab:key/01234567-0123-0123-0123-0123456789ab is pending import.",
    "request-id": "01234567-0123-0123-0123-0123456789ab",
  }
}
```

Using Amazon Lambda To Handle CloudWatch Events

You can use Amazon EBS and CloudWatch Events to automate your data-backup workflow. This requires you to create an IAM policy, a AWS Lambda function to handle the event, and an Amazon CloudWatch Events rule that matches incoming events and routes them to the Lambda function.

The following procedure uses the `createSnapshot` event to automatically copy a completed snapshot to another region for disaster recovery.

To copy a completed snapshot to another region

1. Create an IAM policy, such as the one shown in the following example, to provide permissions to execute a `CopySnapshot` action and write to the CloudWatch Events log. Assign the policy to the IAM user that will handle the CloudWatch event.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "arn:aws:logs:*:*:*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CopySnapshot"
      ],
      "Resource": "*"
    }
  ]
}
```

2. Define a function in Lambda that will be available from the CloudWatch console. The sample Lambda function below, written in Node.js, is invoked by CloudWatch when a matching createSnapshot event is emitted by Amazon EBS (signifying that a snapshot was completed). When invoked, the function copies the snapshot from us-east-2 to us-east-1.

```
// Sample Lambda function to copy an EBS snapshot to a different region

var AWS = require('aws-sdk');
var ec2 = new AWS.EC2();

// define variables
var destinationRegion = 'us-east-1';
var sourceRegion = 'us-east-2';
console.log ('Loading function');

//main function
exports.handler = (event, context, callback) => {

  // Get the EBS snapshot ID from the CloudWatch event details
  var snapshotArn = event.detail.snapshot_id.split('/');
  const snapshotId = snapshotArn[1];
  const description = `Snapshot copy from ${snapshotId} in
  ${sourceRegion}.`;
  console.log ("snapshotId:", snapshotId);

  // Load EC2 class and update the configuration to use destination
  region to initiate the snapshot.
  AWS.config.update({region: destinationRegion});
  var ec2 = new AWS.EC2();

  // Prepare variables for ec2.modifySnapshotAttribute call
  const copySnapshotParams = {
    Description: description,
    DestinationRegion: destinationRegion,
    SourceRegion: sourceRegion,
```

```
        SourceSnapshotId: snapshotId
    };

    // Execute the copy snapshot and log any errors
    ec2.copySnapshot(copySnapshotParams, (err, data) => {
        if (err) {
            const errorMessage = `Error copying snapshot ${snapshotId} to
region ${destinationRegion}.`;
            console.log(errorMessage);
            console.log(err);
            callback(errorMessage);
        } else {
            const successMessage = `Successfully started copy of snapshot
${snapshotId} to region ${destinationRegion}.`;
            console.log(successMessage);
            console.log(data);
            callback(null, successMessage);
        }
    });
};
```

To ensure that your Lambda function is available from the CloudWatch console, create it in the region where the CloudWatch event will occur. For more information, see the [AWS Lambda Developer Guide](#).

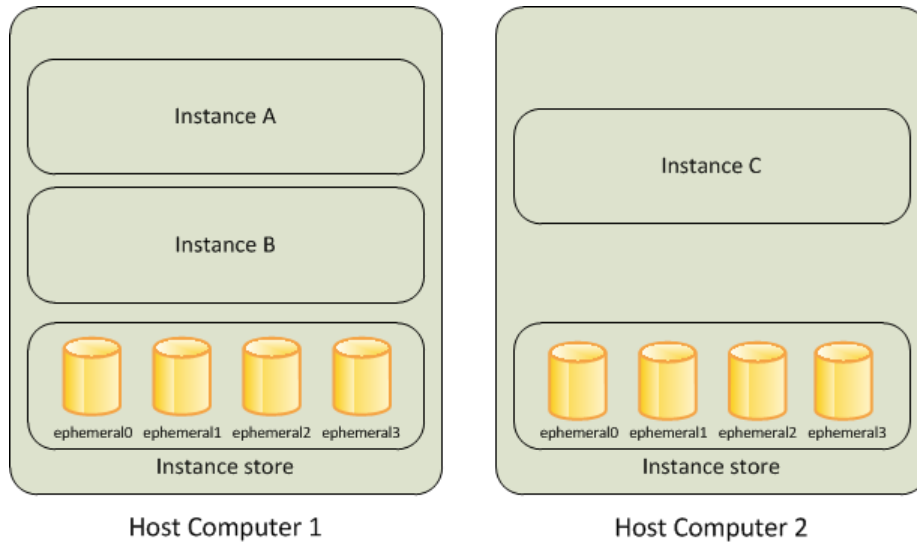
3. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
4. Choose **Events**, **Create rule**, **Select event source**, and **Amazon EBS Snapshots**.
5. For **Specific Event(s)**, choose **createSnapshot** and for **Specific Result(s)**, choose **succeeded**.
6. For **Rule target**, find and choose the sample function that you previously created.
7. Choose **Target**, **Add Target**.
8. For **Lambda function**, select the Lambda function that you previously created and choose **Configure details**.
9. On the **Configure rule details** page, type values for **Name** and **Description**. Select the **State** check box to activate the function (setting it to **Enabled**).
10. Choose **Create rule**.

Your rule should now appear on the **Rules** tab. In the example shown, the event that you configured should be emitted by EBS the next time you copy a snapshot.

Amazon EC2 Instance Store

An *instance store* provides temporary block-level storage for your instance. This storage is located on disks that are physically attached to the host computer. Instance store is ideal for temporary storage of information that changes frequently, such as buffers, caches, scratch data, and other temporary content, or for data that is replicated across a fleet of instances, such as a load-balanced pool of web servers.

An instance store consists of one or more instance store volumes exposed as block devices. The size of an instance store varies by instance type. The virtual devices for instance store volumes are ephemeral[0-23]. Instance types that support one instance store volume have ephemeral0. Instance types that support two instance store volumes have ephemeral0 and ephemeral1, and so on. While an instance store is dedicated to a particular instance, the disk subsystem is shared among instances on a host computer.



Contents

- [Instance Store Lifetime \(p. 823\)](#)
- [Instance Store Volumes \(p. 823\)](#)
- [Add Instance Store Volumes to Your EC2 Instance \(p. 826\)](#)
- [SSD Instance Store Volumes \(p. 828\)](#)

Instance Store Lifetime

You can specify instance store volumes for an instance only when you launch it, though you may be able to resize an instance and add additional ephemeral storage during that process. For information about resizing instances, see [Resizing Your Instance](#).

The data in an instance store persists only during the lifetime of its associated instance. If an instance reboots (intentionally or unintentionally), data in the instance store persists. However, data in the instance store is lost under the following circumstances:

- The underlying disk drive fails
- The instance stops
- The instance terminates

Therefore, do not rely on instance store for valuable, long-term data. Instead, you can build a degree of redundancy (for example, RAID 1/5/6), or use a file system (for example, HDFS and MapR-FS) that supports redundancy and fault tolerance. You can also back up data periodically to more durable data storage solutions such as Amazon S3 or Amazon EBS.

You can't detach an instance store volume from one instance and attach it to a different instance. If you create an AMI from an instance, the data on its instance store volumes isn't preserved and isn't present on the instance store volumes of the instances that you launch from the AMI.

Instance Store Volumes

The instance type determines the size of the instance store available and the type of hardware used for the instance store volumes. Instance store volumes are included as part of the instance's hourly cost. You must specify the instance store volumes that you'd like to use when you launch the instance, and

then format and mount them before using them. You can't make an instance store volume available after you launch the instance. For more information, see [Add Instance Store Volumes to Your EC2 Instance](#) (p. 826).

Some instance types use solid state drives (SSD) to deliver very high random I/O performance. This is a good option when you need storage with very low latency, but you don't need the data to persist when the instance terminates or you can take advantage of fault-tolerant architectures. For more information, see [SSD Instance Store Volumes](#) (p. 828).

The following table provides the quantity, size, type, and performance optimizations of instance store volumes available on each supported instance type. For a complete list of instance types, including EBS-only types, see [Amazon EC2 Instance Types](#).

Instance Type	Instance Store Volumes	Type	Needs Initialization*	TRIM Support**
c1.medium	1 x 350 GB	HDD	✓	
c1.xlarge	4 x 420 GB (1,680 GB)	HDD	✓	
c3.large	2 x 16 GB (32 GB)	SSD	✓	
c3.xlarge	2 x 40 GB (80 GB)	SSD	✓	
c3.2xlarge	2 x 80 GB (160 GB)	SSD	✓	
c3.4xlarge	2 x 160 GB (320 GB)	SSD	✓	
c3.8xlarge	2 x 320 GB (640 GB)	SSD	✓	
cc2.8xlarge	4 x 840 GB (3,360 GB)	HDD	✓	
cg1.4xlarge	2 x 840 GB (1,680 GB)	HDD	✓	
cr1.8xlarge	2 x 120 GB (240 GB)	SSD	✓	
d2.xlarge	3 x 2,000 GB (6 TB)	HDD		
d2.2xlarge	6 x 2,000 GB (12 TB)	HDD		
d2.4xlarge	12 x 2,000 GB (24 TB)	HDD		
d2.8xlarge	24 x 2,000 GB (48 TB)	HDD		
g2.2xlarge	1 x 60 GB	SSD	✓	
g2.8xlarge	2 x 120 GB (240 GB)	SSD	✓	

Amazon Elastic Compute Cloud
User Guide for Windows Instances
Instance Store Volumes

Instance Type	Instance Store Volumes	Type	Needs Initialization*	TRIM Support**
hi1.4xlarge	2 x 1,024 GB (2,048 GB)	SSD		
hs1.8xlarge	24 x 2,000 GB (48 TB)	HDD	✓	
i2.xlarge	1 x 800 GB	SSD		✓
i2.2xlarge	2 x 800 GB (1,600 GB)	SSD		✓
i2.4xlarge	4 x 800 GB (3,200 GB)	SSD		✓
i2.8xlarge	8 x 800 GB (6,400 GB)	SSD		✓
m1.small	1 x 160 GB	HDD	✓	
m1.medium	1 x 410 GB	HDD	✓	
m1.large	2 x 420 GB (840 GB)	HDD	✓	
m1.xlarge	4 x 420 GB (1,680 GB)	HDD	✓	
m2.xlarge	1 x 420 GB	HDD	✓	
m2.2xlarge	1 x 850 GB	HDD	✓	
m2.4xlarge	2 x 840 GB (1,680 GB)	HDD	✓	
m3.medium	1 x 4 GB	SSD	✓	
m3.large	1 x 32 GB	SSD	✓	
m3.xlarge	2 x 40 GB (80 GB)	SSD	✓	
m3.2xlarge	2 x 80 GB (160 GB)	SSD	✓	
r3.large	1 x 32 GB	SSD		✓
r3.xlarge	1 x 80 GB	SSD		✓
r3.2xlarge	1 x 160 GB	SSD		✓
r3.4xlarge	1 x 320 GB	SSD		✓
r3.8xlarge	2 x 320 GB (640 GB)	SSD		✓
x1.16xlarge	1 x 1,920 GB	SSD		
x1.32xlarge	2 x 1,920 GB (3,840 GB)	SSD		

* Volumes attached to certain instances will suffer a first-write penalty unless initialized. For more information about initializing instance store volumes, see [Optimizing Disk Performance for Instance Store Volumes](#).

Add Instance Store Volumes to Your EC2 Instance

You specify the EBS volumes and instance store volumes for your instance using a block device mapping. Each entry in a block device mapping includes a device name and the volume that it maps to. The default block device mapping is specified by the AMI you use. Alternatively, you can specify a block device mapping for the instance when you launch it. For more information, see [Block Device Mapping \(p. 833\)](#).

A block device mapping always specifies the root volume for the instance. The root volume is either an Amazon EBS volume or an instance store volume. For more information, see [Storage for the Root Device \(p. 64\)](#). The root volume is mounted automatically. For instances with an instance store volume for the root volume, the size of this volume varies by AMI, but the maximum size is 10 GB.

You can use a block device mapping to specify additional EBS volumes when you launch your instance, or you can attach additional EBS volumes after your instance is running. For more information, see [Amazon EBS Volumes \(p. 747\)](#).

You can specify the instance store volumes for your instance only when you launch an instance. You can't attach instance store volumes to an instance after you've launched it.

The number and size of available instance store volumes for your instance varies by instance type. Some instance types do not support instance store volumes. For more information about the instance store volumes support by each instance type, see [Instance Store Volumes \(p. 823\)](#). If the instance type you choose for your instance supports instance store volumes, you must add them to the block device mapping for the instance when you launch it. After you launch the instance, you must ensure that the instance store volumes for your instance are formatted and mounted before you can use them. Note that the root volume of an instance store-backed instance is mounted automatically.

Contents

- [Adding Instance Store Volumes to an AMI \(p. 826\)](#)
- [Adding Instance Store Volumes to an Instance \(p. 827\)](#)
- [Making Instance Store Volumes Available on Your Instance \(p. 828\)](#)

Adding Instance Store Volumes to an AMI

You can create an AMI with a block device mapping that includes instance store volumes. After you add instance store volumes to an AMI, any instance that you launch from the AMI includes these instance store volumes. Note that when you launch an instance, you can omit volumes specified in the AMI block device mapping and add new volumes.

Important

For M3 instances, specify instance store volumes in the block device mapping of the instance, not the AMI. Amazon EC2 might ignore instance store volumes that are specified only in the block device mapping of the AMI.

To add instance store volumes to an Amazon EBS-backed AMI using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select an instance, choose **Actions**, select **Image**, and then select **Create Image**.
4. In the **Create Image** dialog, add a meaningful name and description for your image.

- For each instance store volume to add, choose **Add New Volume**, select an instance store volume from **Type**, and select a device name from **Device**. (For more information, see [Device Naming on Windows Instances \(p. 832\)](#).) The number of available instance store volumes depends on the instance type.

Type	Device	Snapshot	Size (GiB)
Root	/dev/xvda	snap-bfb086e1	8
Instance Store 0	/dev/sdb	N/A	N/A
Instance Store 1	/dev/sdc	Search (case-insensitive)	8

- Click **Create Image**.

To add instance store volumes to an AMI using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- `create-image` or `register-image` (AWS CLI)
- `New-EC2Image` and `Register-EC2Image` (AWS Tools for Windows PowerShell)

Adding Instance Store Volumes to an Instance

When you launch an instance, the default block device mapping is provided by the specified AMI. If you need additional instance store volumes, you must add them to the instance as you launch it. Note that you can also omit devices specified in the AMI block device mapping.

Important

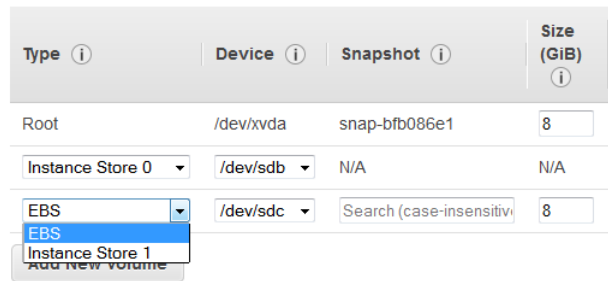
For M3 instances, you might receive instance store volumes even if you do not specify them in the block device mapping for the instance.

Important

For HS1 instances, no matter how many instance store volumes you specify in the block device mapping of an AMI, the block device mapping for an instance launched from the AMI automatically includes the maximum number of supported instance store volumes. You must explicitly remove the instance store volumes that you don't want from the block device mapping for the instance before you launch it.

To update the block device mapping for an instance using the console

- Open the Amazon EC2 console.
- From the dashboard, choose **Launch Instance**.
- In **Step 1: Choose an Amazon Machine Image (AMI)**, select the AMI to use and choose **Select**.
- Follow the wizard to complete **Step 1: Choose an Amazon Machine Image (AMI)**, **Step 2: Choose an Instance Type**, and **Step 3: Configure Instance Details**.
- In **Step 4: Add Storage**, modify the existing entries as needed. For each instance store volume to add, click **Add New Volume**, select an instance store volume from **Type**, and select a device name from **Device**. The number of available instance store volumes depends on the instance type.



6. Complete the wizard to launch the instance.

To update the block device mapping for an instance using the command line

You can use one of the following options commands with the corresponding command. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- `--block-device-mappings` with `run-instances` (AWS CLI)
- `-BlockDeviceMapping` with `New-EC2Instance` (AWS Tools for Windows PowerShell)

Making Instance Store Volumes Available on Your Instance

After you launch an instance, the instance store volumes are available to the instance, but you can't access them until they are mounted. For Linux instances, the instance type determines which instance store volumes are mounted for you and which are available for you to mount yourself. For Windows instances, the EC2Config service mounts the instance store volumes for an instance. The block device driver for the instance assigns the actual volume name when mounting the volume, and the name assigned can be different than the name that Amazon EC2 recommends.

Many instance store volumes are pre-formatted with the ext3 file system. SSD-based instance store volumes that support TRIM instruction are not pre-formatted with any file system. However, you can format volumes with the file system of your choice after you launch your instance. For more information, see [Instance Store Volume TRIM Support \(p. 828\)](#). For Windows instances, the EC2Config service reformats the instance store volumes with the NTFS file system.

You can confirm that the instance store devices are available from within the instance itself using instance metadata. For more information, see [Viewing the Instance Block Device Mapping for Instance Store Volumes \(p. 842\)](#).

For Windows instances, you can also view the instance store volumes using Windows Disk Management. For more information, see [Listing the Disks Using Windows Disk Management \(p. 843\)](#).

SSD Instance Store Volumes

The following instances support instance store volumes that use solid state drives (SSD) to deliver very high random I/O performance: C3, G2, H1, I2, M3, R3, and X1. For more information about the instance store volumes support by each instance type, see [Instance Store Volumes \(p. 823\)](#).

Like other instance store volumes, you must map the SSD instance store volumes for your instance when you launch it, and the data on an SSD instance volume persists only for the life of its associated instance. For more information, see [Add Instance Store Volumes to Your EC2 Instance \(p. 826\)](#).

Instance Store Volume TRIM Support

The following instances support SSD volumes with TRIM: I2, R3.

Important

Instances running Windows Server 2012 R2 support TRIM as of AWS PV Driver version 7.3.0. Instances running earlier versions of Windows Server do not support TRIM.

With instance store volumes that support TRIM, you can use the TRIM command to notify the SSD controller when you no longer need data that you've written. This provides the controller with more free space, which can reduce write amplification and increase performance. For more information about using TRIM commands, see the documentation for the operating system for your instance.

Instance store volumes that support TRIM are fully trimmed before they are allocated to your instance. These volumes are not formatted with a file system when an instance launches, so you must format them before they can be mounted and used. For faster access to these volumes, you should specify the file system-specific option that skips the TRIM operation when you format them. On Linux, you should also add the `discard` option to your mount command or `/etc/fstab` file entries for the devices that support TRIM so that they use this feature effectively. On Windows, use the following command: `fsutil behavior set DisableDeleteNotify 1`.

Amazon Elastic File System (Amazon EFS)

Amazon EFS provides scalable file storage for use with Amazon EC2. You can create an EFS file system and configure your instances to mount the file system. You can use an EFS file system as a common data source for workloads and applications running on multiple instances. For more information, see the [Amazon Elastic File System product page](#).

Important

Amazon EFS is not supported on Windows instances.

Amazon Simple Storage Service (Amazon S3)

Amazon S3 is a repository for Internet data. Amazon S3 provides access to reliable, fast, and inexpensive data storage infrastructure. It is designed to make web-scale computing easy by enabling you to store and retrieve any amount of data, at any time, from within Amazon EC2 or anywhere on the web. Amazon S3 stores data objects redundantly on multiple devices across multiple facilities and allows concurrent read or write access to these data objects by many separate clients or application threads. You can use the redundant data stored in Amazon S3 to recover quickly and reliably from instance or application failures.

Amazon EC2 uses Amazon S3 for storing Amazon Machine Images (AMIs). You use AMIs for launching EC2 instances. In case of instance failure, you can use the stored AMI to immediately launch another instance, thereby allowing for fast recovery and business continuity.

Amazon EC2 also uses Amazon S3 to store snapshots (backup copies) of the data volumes. You can use snapshots for recovering data quickly and reliably in case of application or system failures. You can also use snapshots as a baseline to create multiple new data volumes, expand the size of an existing data volume, or move data volumes across multiple Availability Zones, thereby making your data usage highly scalable. For more information about using data volumes and snapshots, see [Amazon Elastic Block Store \(p. 745\)](#).

Objects are the fundamental entities stored in Amazon S3. Every object stored in Amazon S3 is contained in a bucket. Buckets organize the Amazon S3 namespace at the highest level and identify the account responsible for that storage. Amazon S3 buckets are similar to Internet domain names. Objects stored in the buckets have a unique key value and are retrieved using a HTTP URL address. For example, if an object with a key value `/photos/mygarden.jpg` is stored in the `myawsbucket` bucket, then it is addressable using the URL `http://myawsbucket.s3.amazonaws.com/photos/mygarden.jpg`.

For more information about the features of Amazon S3, see the [Amazon S3 product page](#).

Amazon S3 and Amazon EC2

Given the benefits of Amazon S3 for storage, you may decide to use this service to store files and data sets for use with EC2 instances. There are several ways to move data to and from Amazon S3 to your instances. In addition to the examples discussed below, there are a variety of tools that people have written that you can use to access your data in Amazon S3 from your computer or your instance. Some of the common ones are discussed in the AWS forums.

If you have permission, you can copy a file to or from Amazon S3 and your instance using one of the following methods.

GET or wget

The **wget** utility is an HTTP and FTP client that allows you to download public objects from Amazon S3. It is installed by default in Amazon Linux and most other distributions, and available for download on Windows. To download an Amazon S3 object, use the following command, substituting the URL of the object to download.

```
wget http://s3.amazonaws.com/my_bucket/my_folder/my_file.ext
```

This method requires that the object you request is public; if the object is not public, you receive an ERROR 403: Forbidden message. If you receive this error, open the Amazon S3 console and change the permissions of the object to public. For more information, see the [Amazon Simple Storage Service Developer Guide](#).

AWS Command Line Interface

The AWS Command Line Interface (AWS CLI) is a unified tool to manage your AWS services. With just one tool to download and configure, you can control multiple AWS services from the command line and automate them through scripts. The AWS CLI allows users to authenticate themselves and download restricted items from Amazon S3 and also to upload items. For more information, such as how to install and configure the tools, see the [AWS Command Line Interface detail page](#).

The **aws s3 cp** command is similar to the Unix **cp** command (the syntax is: **aws s3 cp source destination**). You can copy files from Amazon S3 to your instance, you can copy files from your instance to Amazon S3, and you can even copy files from one Amazon S3 location to another.

Use the following command to copy an object from Amazon S3 to your instance.

```
C:\> aws s3 cp s3://my_bucket/my_folder/my_file.ext my_copied_file.ext
```

Use the following command to copy an object from your instance back into Amazon S3.

```
C:\> aws s3 cp my_copied_file.ext s3://my_bucket/my_folder/my_file.ext
```

Use the following command to copy an object from one Amazon S3 location to another.

```
C:\> aws s3 cp s3://my_bucket/my_folder/my_file.ext  
s3://my_bucket/my_folder/my_file2.ext
```

The **aws s3 sync** command can synchronize an entire Amazon S3 bucket to a local directory location. This can be helpful for downloading a data set and keeping the local copy up-to-date with the remote set. The command syntax is: **aws s3 sync source destination**. If you have the proper permissions

on the Amazon S3 bucket, you can push your local directory back up to the cloud when you are finished by reversing the source and destination locations in the command.

Use the following command to download an entire Amazon S3 bucket to a local directory on your instance.

```
C:\> aws s3 sync s3://remote_S3_bucket local_directory
```

AWS Tools for Windows PowerShell

Windows instances have the benefit of a graphical browser that you can use to access the Amazon S3 console directly; however, for scripting purposes, Windows users can also use the [AWS Tools for Windows PowerShell](#) to move objects to and from Amazon S3.

Use the following command to copy an Amazon S3 object to your Windows instance.

```
PS C:\> Copy-S3Object -BucketName my_bucket -Key my_folder/my_file.ext -  
LocalFile my_copied_file.ext
```

Amazon S3 API

If you are a developer, you can use an API to access data in Amazon S3. For more information, see the [Amazon Simple Storage Service Developer Guide](#). You can use this API and its examples to help develop your application and integrate it with other APIs and SDKs, such as the `boto` Python interface.

Instance Volume Limits

The maximum number of volumes that your instance can have depends on the operating system. When considering how many volumes to add to your instance, you should consider whether you need increased I/O bandwidth or increased storage capacity.

Contents

- [Linux-Specific Volume Limits \(p. 831\)](#)
- [Windows-Specific Volume Limits \(p. 831\)](#)
- [Bandwidth vs Capacity \(p. 832\)](#)

Linux-Specific Volume Limits

Attaching more than 40 volumes can cause boot failures. Note that this number includes the root volume, plus any attached instance store volumes and EBS volumes. If you experience boot problems on an instance with a large number of volumes, stop the instance, detach any volumes that are not essential to the boot process, and then reattach the volumes after the instance is running.

Important

Attaching more than 40 volumes to a Linux instance is supported on a best effort basis only and is not guaranteed.

Windows-Specific Volume Limits

The following table shows the volume limits for Windows instances based on the driver used. Note that these numbers include the root volume, plus any attached instance store volumes and EBS volumes.

Important

Attaching more than the following volumes to a Windows instance is supported on a best effort basis only and is not guaranteed.

Driver	Volume Limit
AWS PV	26
Citrix PV	26
Red Hat PV	17

We do not recommend that you give a Windows instance more than 26 volumes with AWS PV or Citrix PV drivers, as it is likely to cause performance issues.

To determine which PV drivers your instance is using, or to upgrade your Windows instance from Red Hat to Citrix PV drivers, see [Upgrading PV Drivers on Your Windows AMI \(p. 356\)](#).

For more information about how device names related to volumes, see [Mapping Disks to Volumes on Your Windows EC2 Instance \(p. 842\)](#).

Bandwidth vs Capacity

For consistent and predictable bandwidth use cases, use EBS-optimized or 10 Gigabit network connectivity instances and General Purpose SSD or Provisioned IOPS SSD volumes. Follow the guidance in [Amazon EC2 Instance Configuration \(p. 805\)](#) to match the IOPS you have provisioned for your volumes to the bandwidth available from your instances for maximum performance. For RAID configurations, many administrators find that arrays larger than 8 volumes have diminished performance returns due to increased I/O overhead. Test your individual application performance and tune it as required.

Device Naming on Windows Instances

When you attach a volume to your instance, you include a device name for the volume. This device name is used by Amazon EC2. The block device driver for the instance assigns the actual volume name when mounting the volume, and the name assigned can be different from the name that Amazon EC2 uses.

Contents

- [Available Device Names \(p. 832\)](#)
- [Device Name Considerations \(p. 833\)](#)

For information about device names on Linux instances, see [Device Naming on Linux Instances](#) in the *Amazon EC2 User Guide for Linux Instances*.

Available Device Names

The following table lists the available device names for Windows instances. The number of volumes that you can attach to your instance is determined by the operating system. For more information, see [Instance Volume Limits \(p. 831\)](#).

Xen Driver Type	Available	Reserved for Root	Used for Instance Store Volumes	Recommended for EBS Volumes
AWS PV, Citrix PV	xvd[b-z]	/dev/sda1	xvd[a-e]	xvd[f-z]

Xen Driver Type	Available	Reserved for Root	Used for Instance Store Volumes	Recommended for EBS Volumes
	xvd[b-c][a-z] /dev/sda1 /dev/sd[b-e]		xvdc[a-x] (hs1.8xlarge)	Warning If you map an EBS volume with the name xvda, Windows does not recognize the volume.
Red Hat PV	xvd[a-z] xvd[b-c][a-z] /dev/sda1 /dev/sd[b-e]	/dev/sda1	xvd[a-e] xvdc[a-x] (hs1.8xlarge)	xvd[f-p]

Note that you can determine the root device name for your particular AMI with the following AWS CLI command:

```
aws ec2 describe-images --image-ids image_id --query Images[.].RootDeviceName
```

For more information about instance store volumes, see [Amazon EC2 Instance Store \(p. 822\)](#). For information about the root device storage, see [Root Device Volume \(p. 8\)](#).

Device Name Considerations

Keep the following in mind when selecting a device name:

- Although you can attach your EBS volumes using the device names used to attach instance store volumes, we strongly recommend that you don't because the behavior can be unpredictable.
- Amazon EC2 Windows AMIs come with an additional service installed, the **Ec2Config Service**. The Ec2Config service runs as a local system and performs various functions to prepare an instance when it first boots up. After the devices have been mapped with the drives, the Ec2Config service then initializes and mounts the drives. The root drive is initialized and mounted as C:\. The instance store volumes that come attached to the instance are initialized and mounted as Z:\, Y:\, and so on. By default, when an EBS volume is attached to a Windows instance, it can show up as any drive letter on the instance. You can change the settings of the Ec2Config service to set the drive letters of the EBS volumes per your specifications. For more information, see [Configuring a Windows Instance Using the EC2Config Service \(p. 283\)](#) and [Mapping Disks to Volumes on Your Windows EC2 Instance \(p. 842\)](#).

Block Device Mapping

Each instance that you launch has an associated root device volume, either an Amazon EBS volume or an instance store volume. You can use block device mapping to specify additional EBS volumes

or instance store volumes to attach to an instance when it's launched. You can also attach additional EBS volumes to a running instance; see [Attaching an Amazon EBS Volume to an Instance \(p. 766\)](#). However, the only way to attach instance store volumes to an instance is to use block device mapping to attach them as the instance is launched.

For more information about root device volumes, see [Root Device Volume \(p. 8\)](#).

Contents

- [Block Device Mapping Concepts \(p. 834\)](#)
- [AMI Block Device Mapping \(p. 836\)](#)
- [Instance Block Device Mapping \(p. 838\)](#)

Block Device Mapping Concepts

A *block device* is a storage device that moves data in sequences of bytes or bits (blocks). These devices support random access and generally use buffered I/O. Examples include hard disks, CD-ROM drives, and flash drives. A block device can be physically attached to a computer or accessed remotely as if it were physically attached to the computer. Amazon EC2 supports two types of block devices:

- Instance store volumes (virtual devices whose underlying hardware is physically attached to the host computer for the instance)
- EBS volumes (remote storage devices)

A *block device mapping* defines the block devices (instance store volumes and EBS volumes) to attach to an instance. You can specify a block device mapping as part of creating an AMI so that the mapping is used by all instances launched from the AMI. Alternatively, you can specify a block device mapping when you launch an instance, so this mapping overrides the one specified in the AMI from which you launched the instance.

Contents

- [Block Device Mapping Entries \(p. 834\)](#)
- [Block Device Mapping Instance Store Caveats \(p. 835\)](#)
- [Example Block Device Mapping \(p. 835\)](#)
- [How Devices Are Made Available in the Operating System \(p. 836\)](#)

Block Device Mapping Entries

When you create a block device mapping, you specify the following information for each block device that you need to attach to the instance:

- The device name used within Amazon EC2. For more information, see [Device Naming on Windows Instances \(p. 832\)](#).

Important

The block device driver for the instance assigns the actual volume name when mounting the volume, and the name assigned can be different from the name that Amazon EC2 recommends.

- [Instance store volumes] The virtual device: `ephemeral[0-23]`. Note that the number and size of available instance store volumes for your instance varies by instance type.
- [EBS volumes] The ID of the snapshot to use to create the block device (`snap-xxxxxxx`). This value is optional as long as you specify a volume size.

- [EBS volumes] The size of the volume, in GiB. The specified size must be greater than or equal to the size of the specified snapshot.
- [EBS volumes] Whether to delete the volume on instance termination (`true` or `false`). The default value is `true` for the root device volume and `false` for attached volumes. When you create an AMI, its block device mapping inherits this setting from the instance. When you launch an instance, it inherits this setting from the AMI.
- [EBS volumes] The volume type, which can be `gp2` for General Purpose SSD, `io1` for Provisioned IOPS SSD, `st1` for Throughput Optimized HDD, `sc1` for Cold HDD, or `standard` for Magnetic. The default value is `gp2` in the Amazon EC2 console, and `standard` in the AWS SDKs and the AWS CLI.
- [EBS volumes] The number of input/output operations per second (IOPS) that the volume supports. (Not used with `gp2`, `st1`, `sc1`, or `standard` volumes.)

Block Device Mapping Instance Store Caveats

There are several caveats to consider when launching instances with AMIs that have instance store volumes in their block device mappings.

- Some instance types include more instance store volumes than others, and some instance types contain no instance store volumes at all. If your instance type supports one instance store volume, and your AMI has mappings for two instance store volumes, then the instance launches with one instance store volume.
- Instance store volumes can only be mapped at launch time. You cannot stop an instance without instance store volumes (such as the `t2.micro`), change the instance to a type that supports instance store volumes, and then restart the instance with instance store volumes. However, you can create an AMI from the instance and launch it on an instance type that supports instance store volumes, and map those instance store volumes to the instance.
- If you launch an instance with instance store volumes mapped, and then stop the instance and change it to an instance type with fewer instance store volumes and restart it, the instance store volume mappings from the initial launch still show up in the instance metadata. However, only the maximum number of supported instance store volumes for that instance type are available to the instance.

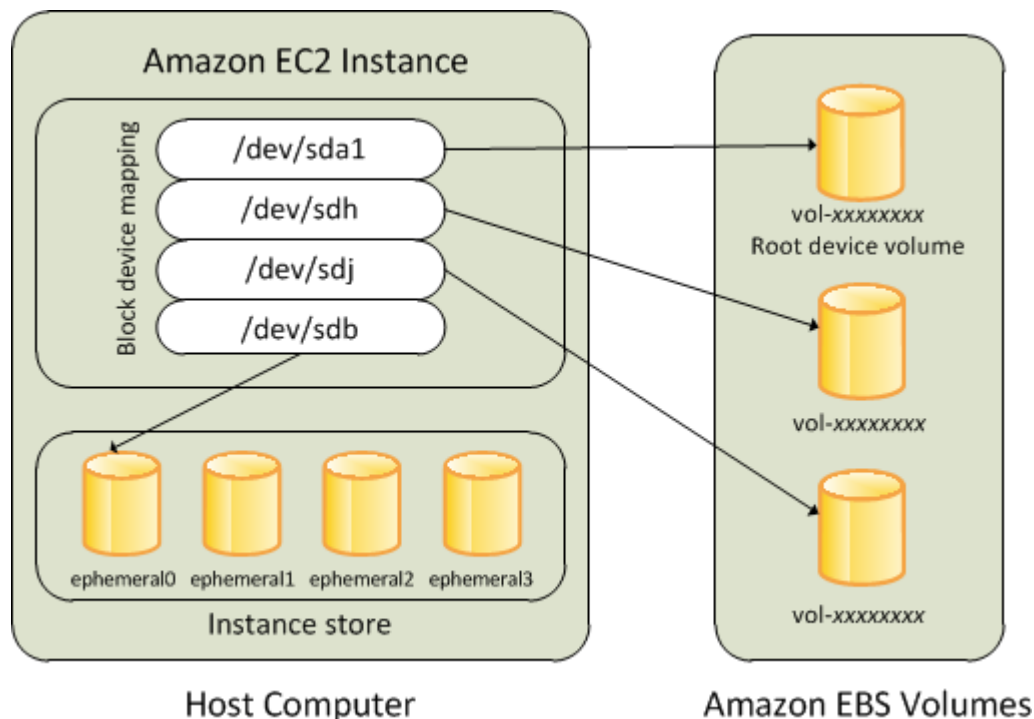
Note

When an instance is stopped, all data on the instance store volumes is lost.

- Depending on instance store capacity at launch time, M3 instances may ignore AMI instance store block device mappings at launch unless they are specified at launch. You should specify instance store block device mappings at launch time, even if the AMI you are launching has the instance store volumes mapped in the AMI, to ensure that the instance store volumes are available when the instance launches.

Example Block Device Mapping

This figure shows an example block device mapping for an EBS-backed instance. It maps `/dev/sdb` to `ephemeral0` and maps two EBS volumes, one to `/dev/sdh` and the other to `/dev/sdj`. It also shows the EBS volume that is the root device volume, `/dev/sda1`.



Note that this example block device mapping is used in the example commands and APIs in this topic. You can find example commands and APIs that create block device mappings in [Specifying a Block Device Mapping for an AMI \(p. 837\)](#) and [Updating the Block Device Mapping when Launching an Instance \(p. 839\)](#).

How Devices Are Made Available in the Operating System

Device names like `/dev/sdh` and `xvdh` are used by Amazon EC2 to describe block devices. The block device mapping is used by Amazon EC2 to specify the block devices to attach to an EC2 instance. After a block device is attached to an instance, it must be mounted by the operating system before you can access the storage device. When a block device is detached from an instance, it is unmounted by the operating system and you can no longer access the storage device.

With a Windows instance, the device names specified in the block device mapping are mapped to their corresponding block devices when the instance first boots, and then the Ec2Config service initializes and mounts the drives. The root device volume is mounted as `C:\`. The instance store volumes are mounted as `Z:\`, `Y:\`, and so on. When an EBS volume is mounted, it can be mounted using any available drive letter. However, you can configure how the Ec2Config Service assigns drive letters to EBS volumes; for more information, see [Configuring a Windows Instance Using the EC2Config Service \(p. 283\)](#).

AMI Block Device Mapping

Each AMI has a block device mapping that specifies the block devices to attach to an instance when it is launched from the AMI. An AMI that Amazon provides includes a root device only. To add more block devices to an AMI, you must create your own AMI.

Contents

- [Specifying a Block Device Mapping for an AMI \(p. 837\)](#)
- [Viewing the EBS Volumes in an AMI Block Device Mapping \(p. 838\)](#)

Specifying a Block Device Mapping for an AMI

There are two ways to specify volumes in addition to the root volume when you create an AMI. If you've already attached volumes to a running instance before you create an AMI from the instance, the block device mapping for the AMI includes those same volumes. For EBS volumes, the existing data is saved to a new snapshot, and it's this new snapshot that's specified in the block device mapping. For instance store volumes, the data is not preserved.

For an EBS-backed AMI, you can add EBS volumes and instance store volumes using a block device mapping. For an instance store-backed AMI, you can add instance store volumes only by modifying the block device mapping entries in the image manifest file when registering the image.

Note

For M3 instances, you must specify instance store volumes in the block device mapping for the instance when you launch it. When you launch an M3 instance, instance store volumes specified in the block device mapping for the AMI may be ignored if they are not specified as part of the instance block device mapping.

To add volumes to an AMI using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, choose **Instances**.
3. Select an instance and choose **Actions, Image, Create Image**.
4. In the **Create Image** dialog box, choose **Add New Volume**.
5. Select a volume type from the **Type** list and a device name from the **Device** list. For an EBS volume, you can optionally specify a snapshot, volume size, and volume type.
6. Choose **Create Image**.

To add volumes to an AMI using the command line

Use the [create-image](#) AWS CLI command to specify a block device mapping for an EBS-backed AMI. Use the [register-image](#) AWS CLI command to specify a block device mapping for an instance store-backed AMI.

Specify the block device mapping using the following parameter:

```
--block-device-mappings [mapping, ...]
```

To add an instance store volume, use the following mapping:

```
{
  "DeviceName": "xvdb",
  "VirtualName": "ephemeral0"
}
```

To add an empty 100 GiB Magnetic volume, use the following mapping:

```
{
  "DeviceName": "xvdg",
  "Ebs": {
    "VolumeSize": 100
  }
}
```

To add an EBS volume based on a snapshot, use the following mapping:

```
{
  "DeviceName": "xvdh",
  "Ebs": {
    "SnapshotId": "snap-xxxxxxxx"
  }
}
```

To omit a mapping for a device, use the following mapping:

```
{
  "DeviceName": "xvdj",
  "NoDevice": ""
}
```

Alternatively, you can use the `-BlockDeviceMapping` parameter with the following commands (AWS Tools for Windows PowerShell):

- [New-EC2Image](#)
- [Register-EC2Image](#)

Viewing the EBS Volumes in an AMI Block Device Mapping

You can easily enumerate the EBS volumes in the block device mapping for an AMI.

To view the EBS volumes for an AMI using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, choose **AMIs**.
3. Choose **EBS images** from the **Filter** list to get a list of EBS-backed AMIs.
4. Select the desired AMI, and look at the **Details** tab. At a minimum, the following information is available for the root device:
 - **Root Device Type** (ebs)
 - **Root Device Name** (for example, `/dev/sda1`)
 - **Block Devices** (for example, `/dev/sda1=snap-1234567890abcdef0:8:true`)

If the AMI was created with additional EBS volumes using a block device mapping, the **Block Devices** field displays the mapping for those additional volumes as well. (Recall that this screen doesn't display instance store volumes.)

To view the EBS volumes for an AMI using the command line

Use the [describe-images](#) (AWS CLI) command or [Get-EC2Image](#) (AWS Tools for Windows PowerShell) command to enumerate the EBS volumes in the block device mapping for an AMI.

Instance Block Device Mapping

By default, an instance that you launch includes any storage devices specified in the block device mapping of the AMI from which you launched the instance. You can specify changes to the block

device mapping for an instance when you launch it, and these updates overwrite or merge with the block device mapping of the AMI. However,

Limits

- For the root volume, you can only modify the following: volume size, volume type, and the **Delete on Termination** flag.
- When you modify an EBS volume, you can't decrease its size. Therefore, you must specify a snapshot whose size is equal to or greater than the size of the snapshot specified in the block device mapping of the AMI.

Contents

- [Updating the Block Device Mapping when Launching an Instance \(p. 839\)](#)
- [Updating the Block Device Mapping of a Running Instance \(p. 840\)](#)
- [Viewing the EBS Volumes in an Instance Block Device Mapping \(p. 841\)](#)
- [Viewing the Instance Block Device Mapping for Instance Store Volumes \(p. 842\)](#)

Updating the Block Device Mapping when Launching an Instance

You can add EBS volumes and instance store volumes to an instance when you launch it. Note that updating the block device mapping for an instance doesn't make a permanent change to the block device mapping of the AMI from which it was launched.

To add volumes to an instance using the console

1. Open the Amazon EC2 console.
2. From the dashboard, choose **Launch Instance**.
3. On the **Choose an Amazon Machine Image (AMI)** page, select the AMI to use and choose **Select**.
4. Follow the wizard to complete the **Choose an Instance Type** and **Configure Instance Details** pages.
5. On the **Add Storage** page, you can modify the root volume, EBS volumes, and instance store volumes as follows:
 - To change the size of the root volume, locate the **Root** volume under the **Type** column, and change its **Size** field.
 - To suppress an EBS volume specified by the block device mapping of the AMI used to launch the instance, locate the volume and click its **Delete** icon.
 - To add an EBS volume, choose **Add New Volume**, choose **EBS** from the **Type** list, and fill in the fields (**Device**, **Snapshot**, and so on).
 - To suppress an instance store volume specified by the block device mapping of the AMI used to launch the instance, locate the volume, and choose its **Delete** icon.
 - To add an instance store volume, choose **Add New Volume**, select **Instance Store** from the **Type** list, and select a device name from **Device**.
6. Complete the remaining wizard pages, and choose **Launch**.

To add volumes to an instance using the command line

Use the `run-instances` AWS CLI command to specify a block device mapping for an instance.

Specify the block device mapping using the following parameter:

```
--block-device-mappings [mapping, ...]
```

For example, suppose that an EBS-backed AMI specifies the following block device mapping:

- xvdb=ephemeral0
- xvdh=snap-1234567890abcdef0
- xvdj=:100

To prevent `xvdj` from attaching to an instance launched from this AMI, use the following mapping:

```
{  
  "DeviceName": "xvdj",  
  "NoDevice": ""  
}
```

To increase the size of `xvdh` to 300 GiB, specify the following mapping. Notice that you don't need to specify the snapshot ID for `xvdh`, because specifying the device name is enough to identify the volume.

```
{  
  "DeviceName": "xvdh",  
  "Ebs": {  
    "VolumeSize": 300  
  }  
}
```

To attach an additional instance store volume, `xvdc`, specify the following mapping. If the instance type doesn't support multiple instance store volumes, this mapping has no effect.

```
{  
  "DeviceName": "xvdc",  
  "VirtualName": "ephemeral1"  
}
```

Alternatively, you can use the `-BlockDeviceMapping` parameter with the [New-EC2Instance](#) command (AWS Tools for Windows PowerShell).

Updating the Block Device Mapping of a Running Instance

You can use the following [modify-instance-attribute](#) AWS CLI command to update the block device mapping of a running instance. Note that you do not need to stop the instance before changing this attribute.

```
C:\> aws ec2 modify-instance-attribute --instance-id i-1a2b3c4d --block-device-mappings file://mapping.json
```

For example, to preserve the root volume at instance termination, specify the following in `mapping.json`:

```
[  
  {  
    "DeviceName": "/dev/sda1",
```

```
"Ebs": {  
  "DeleteOnTermination": false  
}  
]  
]
```

Alternatively, you can use the `-BlockDeviceMapping` parameter with the [Edit-EC2InstanceAttribute](#) command (AWS Tools for Windows PowerShell).

Viewing the EBS Volumes in an Instance Block Device Mapping

You can easily enumerate the EBS volumes mapped to an instance.

Note

For instances launched before the release of the 2009-10-31 API, AWS can't display the block device mapping. You must detach and reattach the volumes so that AWS can display the block device mapping.

To view the EBS volumes for an instance using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, choose **Instances**.
3. In the search bar, type **Root Device Type**, and then choose **EBS**. This displays a list of EBS-backed instances.
4. Select the desired instance and look at the details displayed in the **Description** tab. At a minimum, the following information is available for the root device:

- **Root device type** (*ebs*)
- **Root device** (for example, */dev/sda1*)
- **Block devices** (for example, */dev/sda1*, *xvdh*, and *xvdj*)

If the instance was launched with additional EBS volumes using a block device mapping, the **Block devices** field displays those additional volumes as well as the root device. (Recall that this dialog box doesn't display instance store volumes.)

Root device type	<i>ebs</i>
Root device	<i>/dev/sda1</i>
Block devices	<i>/dev/sda1</i> <i>/dev/sdf</i>

5. To display additional information about a block device, select its entry next to **Block devices**. This displays the following information for the block device:
 - **EBS ID** (*vol-xxxxxxx*)
 - **Root device type** (*ebs*)
 - **Attachment time** (*yyyy-mmT hh:mm:ss.ssTZD*)
 - **Block device status** (*attaching, attached, detaching, detached*)
 - **Delete on termination** (*Yes, No*)

To view the EBS volumes for an instance using the command line

Use the [describe-instances](#) (AWS CLI) command or [Get-EC2Instance](#) (AWS Tools for Windows PowerShell) command to enumerate the EBS volumes in the block device mapping for an instance.

Viewing the Instance Block Device Mapping for Instance Store Volumes

When you view the block device mapping for your instance, you can see only the EBS volumes, not the instance store volumes. You can use instance metadata to query the complete block device mapping. The base URI for all requests for instance metadata is `http://169.254.169.254/latest/`.

First, connect to your running instance. For Windows instances, install `wget` on the instance if it is not installed already.

Use this query on a running instance to get its block device mapping.

```
C:\> wget http://169.254.169.254/latest/meta-data/block-device-mapping/
```

The response includes the names of the block devices for the instance. For example, the output for an instance store-backed `m1.small` instance looks like this.

```
ami  
ephemeral0  
root  
swap
```

The `ami` device is the root device as seen by the instance. The instance store volumes are named `ephemeral[0-23]`. The swap device is for the page file. If you've also mapped EBS volumes, they appear as `ebs1`, `ebs2`, and so on.

To get details about an individual block device in the block device mapping, append its name to the previous query, as shown here.

```
C:\> wget http://169.254.169.254/latest/meta-data/block-device-mapping/  
ephemeral0
```

For more information, see [Instance Metadata and User Data \(p. 271\)](#).

Mapping Disks to Volumes on Your Windows EC2 Instance

Your Windows EC2 instance comes with an EBS volume that serves as the root volume. If your Windows instance uses AWS PV or Citrix PV drivers, you can optionally add up to 25 volumes, making a total of 26 volumes. For more information, see [Instance Volume Limits \(p. 831\)](#)

Depending on the instance type of your instance, you'll have from 0 to 24 possible instance store volumes available to the instance. To use any of the instance store volumes that are available to your instance, you must specify them when you create your AMI or launch your instance. You can also add EBS volumes when you create your AMI or launch your instance, or attach them while your instance is running.

When you add a volume to your instance, you specify the device name that Amazon EC2 uses. For more information, see [Device Naming on Windows Instances \(p. 832\)](#). AWS Windows Amazon Machine Images (AMIs) contain a set of drivers that are used by Amazon EC2 to map instance store and EBS volumes to Windows disks and drive letters. If you launch an instance from a Windows AMI

that uses Citrix paravirtualized (PV) or AWS PV drivers, you can use the relationships described on this page to map your Windows disks to your instance store and EBS volumes. If your Windows AMI uses Red Hat PV drivers, you can update your instance to use the Citrix drivers. For more information, see [Upgrading PV Drivers on Your Windows AMI \(p. 356\)](#).

Contents

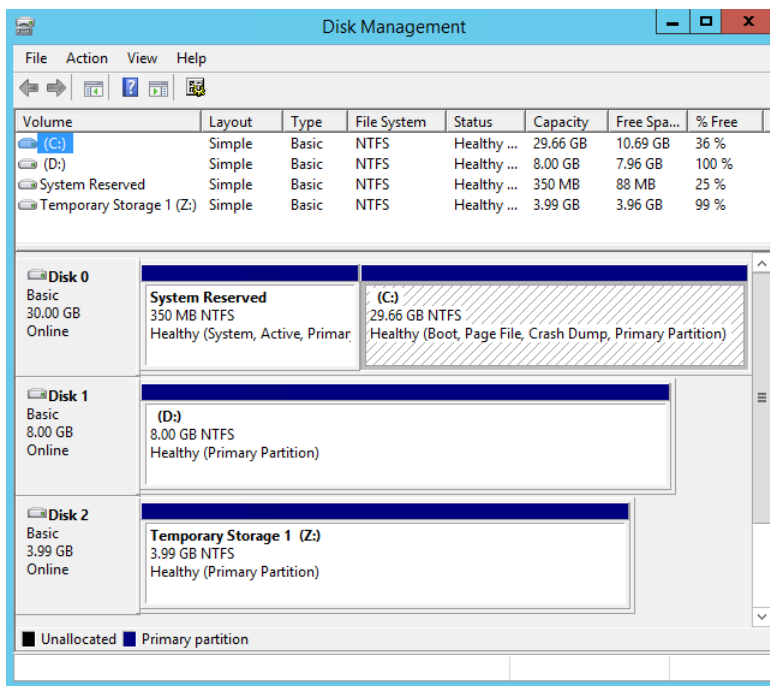
- [Listing the Disks Using Windows Disk Management \(p. 843\)](#)
- [Listing the Disks Using Windows PowerShell \(p. 844\)](#)
- [Disk Device to Device Name Mapping \(p. 846\)](#)

Listing the Disks Using Windows Disk Management

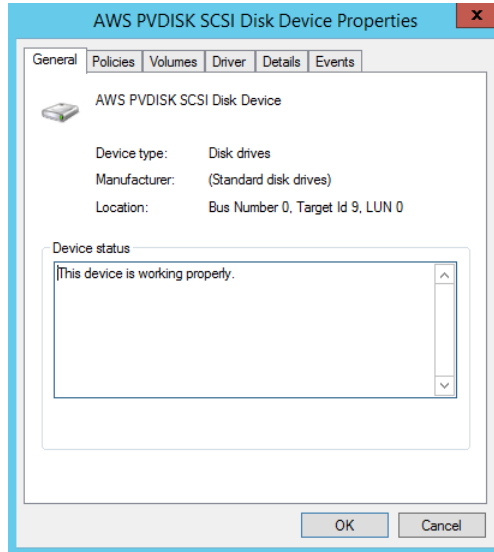
You can find the disks on your Windows instance using Windows Disk Management.

To find the disks on your Windows instance

1. Log in to your Windows instance using Remote Desktop. For more information, see, [Connecting to Your Windows Instance \(p. 254\)](#).
2. Start the Disk Management utility. On Windows Server 2012, on the taskbar, right-click the Windows logo, and then select **Disk Management**. On Windows Server 2008, click **Start**, point to **Administrative Tools**, select **Computer Management**, and then select **Disk Management**.
3. Review the disks. Disk 0 is the root volume, which is an EBS volume mounted as `C:\`. If there are no other disks shown, then your instance does not come with instance store volumes, and you didn't specify any EBS volumes when you created the AMI or launched the instance. Otherwise, you'll see additional disks. For example, the following disks are available if you launch an `m3.medium` instance with an additional empty EBS volume. Disk 1 is the EBS volume, and Disk 2 is the instance store volume.

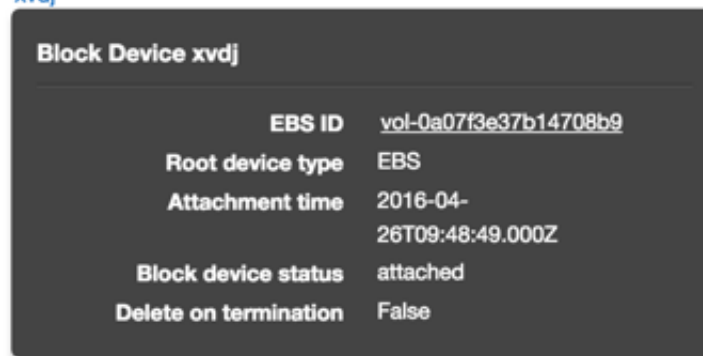


4. Right-click the gray pane labeled Disk 1, and then select **Properties**. Note the value of **Location** and look it up in the tables in [Disk Device to Device Name Mapping \(p. 846\)](#). For example, the following disk has the location `Bus Number 0, Target Id 9, LUN 0`. According to the table for EBS volumes, the device name for this location is `xvdj`.



- To map the device name of an EBS volume to its volume ID, open the Amazon EC2 console on your computer. In the navigation pane, select **Instances**, and then select your instance. Under **Block devices**, click the device name, and locate **EBS ID**. For this example, the volume ID is `vol-0a07f3e37b14708b9`.

Block devices `/dev/sda1`
`xvdj`



Note that the Amazon EC2 console shows only the EBS volumes.

Listing the Disks Using Windows PowerShell

The following PowerShell script lists each disk and its corresponding device name and volume.

```
# List the Windows disks

# Create a hash table that maps each device to a SCSI target
$Map = @{"0" = '/dev/sda1'}
for($x = 1; $x -le 25; $x++) {$Map.add($x.ToString(),
  [String]::Format("xvd{0}",[char](97 + $x)))}
for($x = 26; $x -le 51; $x++) {$Map.add($x.ToString(),
  [String]::Format("xvda{0}",[char](71 + $x)))}
for($x = 52; $x -le 77; $x++) {$Map.add($x.ToString(),
  [String]::Format("xvdb{0}",[char](45 + $x)))}
```

Amazon Elastic Compute Cloud
User Guide for Windows Instances
Listing the Disks Using Windows PowerShell

```
for($x = 78; $x -le 103; $x++) {$Map.add($x.ToString(),
 [String]::Format("xvdc{0}",[char](19 + $x)))}
for($x = 104; $x -le 129; $x++) {$Map.add($x.ToString(),
 [String]::Format("xvdd{0}",[char]($x - 7)))}

Try {
    # Use the metadata service to discover which instance the script is
    running on
    $InstanceId = (Invoke-WebRequest '169.254.169.254/latest/meta-data/
instance-id').Content
    $AZ = (Invoke-WebRequest '169.254.169.254/latest/meta-data/placement/
availability-zone').Content
    $Region = $AZ.Substring(0, $AZ.Length -1)

    #Get the volumes attached to this instance
    $BlockDeviceMappings = (Get-EC2Instance -Region $Region -Instance
$InstanceId).Instances.BlockDeviceMappings
}
Catch
{
    Write-Host "Could not access the AWS API, therefore, VolumeId is not
available.
Verify that you provided your access keys." -ForegroundColor Yellow
}

Get-WmiObject -Class Win32_DiskDrive | % {
    $Drive = $_
    # Find the partitions for this drive
    Get-WmiObject -Class Win32_DiskDriveToDiskPartition | Where-Object
{$_ .Antecedent -eq $Drive.Path.Path} | %{
        $D2P = $_
        # Get details about each partition
        $Partition = Get-WmiObject -Class Win32_DiskPartition | Where-Object
{$_ .Path.Path -eq $D2P.Dependent}
        # Find the drive that this partition is linked to
        $Disk = Get-WmiObject -Class Win32_LogicalDiskToPartition | Where-
Object {$_ .Antecedent -in $D2P.Dependent} | %{
            $L2P = $_
            #Get the drive letter for this partition, if there is one
            Get-WmiObject -Class Win32_LogicalDisk | Where-Object
{$_ .Path.Path -in $L2P.Dependent}
        }
        $BlockDeviceMapping = $BlockDeviceMappings | Where-Object
{$_ .DeviceName -eq $Map[$Drive.SCSITargetId.ToString()]}

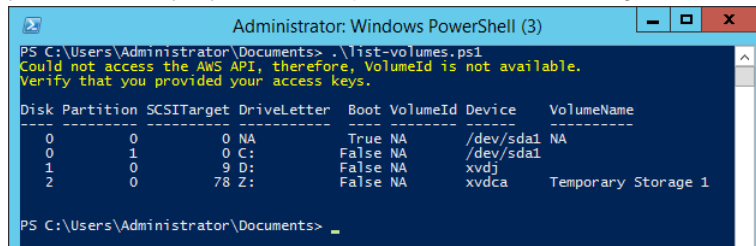
        # Display the information in a table
        New-Object PSObject -Property @{
            Device = $Map[$Drive.SCSITargetId.ToString()];
            Disk = [Int]::Parse($Partition.Name.Split(",")[0].Replace("Disk
#", ""));
            Boot = $Partition.BootPartition;
            Partition = [Int]::Parse($Partition.Name.Split(",")[1].Replace("
Partition #", ""));
            SCSITarget = $Drive.SCSITargetId;
            DriveLetter = If($Disk -eq $NULL) {"NA"} else {$Disk.DeviceID};
            VolumeName = If($Disk -eq $NULL) {"NA"} else {$Disk.VolumeName};
            VolumeId = If($BlockDeviceMapping -eq $NULL) {"NA"} else
{$BlockDeviceMapping.Ebs.VolumeId}
        }
    }
```

```
}
} | Sort-Object Disk, Partition | Format-Table -AutoSize -Property Disk,
  Partition, SCSITarget, DriveLetter, Boot,
  VolumeId, Device, VolumeName
```

Before you run this script, be sure to run the following command to enable PowerShell script execution.

```
Set-ExecutionPolicy RemoteSigned
```

Copy the script and save it as a .ps1 file on the Windows instance. If you run the script without setting your access keys, you'll see output similar to the following.



If you specified an IAM role with a policy that allows access to Amazon EC2 when you launched the instance, or if you set up your credentials on the Windows instance as described in [Using AWS Credentials](#) in the *AWS Tools for Windows PowerShell User Guide*, you'll get the volume ID (vol-xxxxxxx) for the EBS volumes in the VolumeId column instead of NA.

Disk Device to Device Name Mapping

The following table describes how the Citrix PV and AWS PV drivers map instance store volumes to Windows volumes. The number of available instance store volumes is determined by the instance type. For more information, see [Instance Store Volumes \(p. 823\)](#).

Location	Device Name
Bus Number 0, Target ID 78, LUN 0	xvdca
Bus Number 0, Target ID 79, LUN 0	xvdcb
Bus Number 0, Target ID 80, LUN 0	xvdcc
Bus Number 0, Target ID 81, LUN 0	xvdcd
Bus Number 0, Target ID 82, LUN 0	xvdce
Bus Number 0, Target ID 83, LUN 0	xvdcf
Bus Number 0, Target ID 84, LUN 0	xvdcg
Bus Number 0, Target ID 85, LUN 0	xvdch
Bus Number 0, Target ID 86, LUN 0	xvdci
Bus Number 0, Target ID 87, LUN 0	xvdcj
Bus Number 0, Target ID 88, LUN 0	xvdck
Bus Number 0, Target ID 89, LUN 0	xvdcl

The following table describes how the Citrix PV and AWS PV drivers map EBS volumes to Windows volumes. For more information, see [Device Naming on Windows Instances \(p. 832\)](#).

Location	Device Name
Bus Number 0, Target ID 0, LUN 0	/dev/sda1
Bus Number 0, Target ID 1, LUN 0	xvdb
Bus Number 0, Target ID 2, LUN 0	xvdc
Bus Number 0, Target ID 3, LUN 0	xvdd
Bus Number 0, Target ID 4, LUN 0	xvde
Bus Number 0, Target ID 5, LUN 0	xvdf
Bus Number 0, Target ID 6, LUN 0	xvdg
Bus Number 0, Target ID 7, LUN 0	xvdh
Bus Number 0, Target ID 8, LUN 0	xvdi
Bus Number 0, Target ID 9, LUN 0	xvdj
Bus Number 0, Target ID 10, LUN 0	xvdk
Bus Number 0, Target ID 11, LUN 0	xvdl
Bus Number 0, Target ID 12, LUN 0	xvdm
Bus Number 0, Target ID 13, LUN 0	xvdn
Bus Number 0, Target ID 14, LUN 0	xvdo
Bus Number 0, Target ID 15, LUN 0	xvdp
Bus Number 0, Target ID 16, LUN 0	xvdq
Bus Number 0, Target ID 17, LUN 0	xvdr
Bus Number 0, Target ID 18, LUN 0	xvds
Bus Number 0, Target ID 19, LUN 0	xvdt
Bus Number 0, Target ID 20, LUN 0	xvdu
Bus Number 0, Target ID 21, LUN 0	xvdv
Bus Number 0, Target ID 22, LUN 0	xvdw
Bus Number 0, Target ID 23, LUN 0	xvdx
Bus Number 0, Target ID 24, LUN 0	xvdy
Bus Number 0, Target ID 25, LUN 0	xvdz

Using Public Data Sets

Amazon Web Services provides a repository of public data sets that can be seamlessly integrated into AWS cloud-based applications. Amazon stores the data sets at no charge to the community and, as with all AWS services, you pay only for the compute and storage you use for your own applications.

Contents

- [Public Data Set Concepts \(p. 848\)](#)
- [Finding Public Data Sets \(p. 848\)](#)
- [Creating a Public Data Set Volume from a Snapshot \(p. 849\)](#)
- [Attaching and Mounting the Public Data Set Volume \(p. 850\)](#)

Public Data Set Concepts

Previously, large data sets such as the mapping of the Human Genome and the US Census data required hours or days to locate, download, customize, and analyze. Now, anyone can access these data sets from an EC2 instance and start computing on the data within minutes. You can also leverage the entire AWS ecosystem and easily collaborate with other AWS users. For example, you can produce or use prebuilt server images with tools and applications to analyze the data sets. By hosting this important and useful data with cost-efficient services such as Amazon EC2, AWS hopes to provide researchers across a variety of disciplines and industries with tools to enable more innovation, more quickly.

For more information, go to the [Public Data Sets on AWS Page](#).

Available Public Data Sets

Public data sets are currently available in the following categories:

- **Biology**—Includes Human Genome Project, GenBank, and other content.
- **Chemistry**—Includes multiple versions of PubChem and other content.
- **Economics**—Includes census data, labor statistics, transportation statistics, and other content.
- **Encyclopedic**—Includes Wikipedia content from multiple sources and other content.

Finding Public Data Sets

Before you can use a public data set, you must locate the data set and determine which format the data set is hosted in. The data sets are available in two possible formats: Amazon EBS snapshots or Amazon S3 buckets.

To find a public data set and determine its format

1. Go to the [Public Data Sets Page](#) to see a listing of all available public data sets. You can also enter a search phrase on this page to query the available public data set listings.
2. Click the name of a data set to see its detail page.
3. On the data set detail page, look for a snapshot ID listing to identify an Amazon EBS formatted data set or an Amazon S3 URL.

Data sets that are in snapshot format are used to create new EBS volumes that you attach to an EC2 instance. For more information, see [Creating a Public Data Set Volume from a Snapshot \(p. 849\)](#).

For data sets that are in Amazon S3 format, you can use the AWS SDKs or the HTTP query API to access the information, or you can use the AWS CLI to copy or synchronize the data to and from your instance. For more information, see [Amazon S3 and Amazon EC2 \(p. 830\)](#).

You can also use Amazon EMR to analyze and work with public data sets. For more information, see [What is Amazon EMR?](#).

Creating a Public Data Set Volume from a Snapshot

To use a public data set that is in snapshot format, you create a new volume, specifying the snapshot ID of the public data set. You can create your new volume using the AWS Management Console as follows. If you prefer, you can use the [create-volume](#) AWS CLI command instead.

To create a public data set volume from a snapshot

1. Open the Amazon EC2 console.
2. From the navigation bar, select the region that your data set snapshot is located in.

Important

Snapshot IDs are constrained to a single region, and you cannot create a volume from a snapshot that is located in another region. In addition, you can only attach an EBS volume to an instance in the same Availability Zone. For more information, see [Resource Locations \(p. 851\)](#).

If you need to create this volume in a different region, you can copy the snapshot to your required region and then restore it to a volume in that region. For more information, see [Copying an Amazon EBS Snapshot \(p. 791\)](#).

3. In the navigation pane, click **Volumes**.
4. Above the upper pane, click **Create Volume**.
5. In the **Create Volume** dialog box, in the **Type** list, select **General Purpose SSD, Provisioned IOPS SSD**, or **Magnetic**. For more information, see [Amazon EBS Volume Types \(p. 749\)](#).
6. In the **Snapshot** field, start typing the ID or description of the snapshot for your data set. Select the snapshot from the list of suggested options.

Note

If the snapshot ID you are expecting to see does not appear, you may have a different region selected in the Amazon EC2 console. If the data set you identified in [Finding Public Data Sets \(p. 848\)](#) does not specify a region on its detail page, it is likely contained in the `us-east-1` US East (N. Virginia) region.

7. In the **Size** field, enter the size of the volume (in GiB or TiB), or verify that the default size of the snapshot is adequate.

Note

If you specify both a volume size and a snapshot ID, the size must be equal to or greater than the snapshot size. When you select a volume type and a snapshot ID, minimum and maximum sizes for the volume are shown next to the **Size** list.

8. For Provisioned IOPS SSD volumes, in the **IOPS** field, enter the maximum number of input/output operations per second (IOPS) that the volume can support.
9. In the **Availability Zone** list, select the Availability Zone in which to launch the instance.

Important

EBS volumes can only be attached to instances in the same Availability Zone.

10. Click **Yes, Create**.

Important

If you created a larger volume than the default size for that snapshot (by specifying a size in [Step 7 \(p. 849\)](#)), you need to extend the file system on the volume to take advantage of the extra space. For more information, see [Expanding the Storage Space of an EBS Volume on Windows \(p. 783\)](#).

Attaching and Mounting the Public Data Set Volume

After you have created your new data set volume, you need to attach it to an EC2 instance to access the data (this instance must also be in the same Availability Zone as the new volume). For more information, see [Attaching an Amazon EBS Volume to an Instance \(p. 766\)](#).

After you have attached the volume to an instance, you need to mount the volume on the instance. For more information, see [Making an Amazon EBS Volume Available for Use \(p. 767\)](#).

Resources and Tags

Amazon EC2 provides different *resources* that you can create and use. Some of these resources include images, instances, volumes, and snapshots. When you create a resource, we assign the resource a unique resource ID.

Some resources can be tagged with values that you define, to help you organize and identify them.

The following topics describe resources and tags, and how you can work with them.

Topics

- [Resource Locations \(p. 851\)](#)
- [Resource IDs \(p. 852\)](#)
- [Listing and Filtering Your Resources \(p. 856\)](#)
- [Tagging Your Amazon EC2 Resources \(p. 859\)](#)
- [Amazon EC2 Service Limits \(p. 869\)](#)
- [Amazon EC2 Usage Reports \(p. 870\)](#)

Resource Locations

The following table describes which Amazon EC2 resources are global, regional, or based on Availability Zone.

Resource	Type	Description
AWS account	Global	You can use the same AWS account in all regions.
Key pairs	Global or Regional	You can use the key pairs that you create using Amazon EC2 only in the region where you created them. You can create and upload an RSA key pair that you can use in all regions. For more information, see Amazon EC2 Key Pairs and Windows Instances (p. 602) .

Resource	Type	Description
Amazon EC2 resource identifiers	Regional	Each resource identifier, such as an AMI ID, instance ID, EBS volume ID, or EBS snapshot ID, is tied to its region and can be used only in the region where you created the resource.
User-supplied resource names	Regional	Each resource name, such as a security group name or key pair name, is tied to its region and can be used only in the region where you created the resource. Although you can create resources with the same name in multiple regions, they aren't related to each other.
AMIs	Regional	An AMI is tied to the region where its files are located within Amazon S3. You can copy an AMI from one region to another. For more information, see Copying an AMI (p. 87) .
Elastic IP addresses	Regional	An Elastic IP address is tied to a region and can be associated only with an instance in the same region.
Security groups	Regional	A security group is tied to a region and can be assigned only to instances in the same region. You can't enable an instance to communicate with an instance outside its region using security group rules. Traffic from an instance in another region is seen as WAN bandwidth.
EBS snapshots	Regional	An EBS snapshot is tied to its region and can only be used to create volumes in the same region. You can copy a snapshot from one region to another. For more information, see Copying an Amazon EBS Snapshot (p. 791) .
EBS volumes	Availability Zone	An Amazon EBS volume is tied to its Availability Zone and can be attached only to instances in the same Availability Zone.
Instances	Availability Zone	An instance is tied to the Availability Zones in which you launched it. However, note that its instance ID is tied to the region.

Resource IDs

When resources are created, we assign each resource a unique resource ID. You can use resource IDs to find your resources in the Amazon EC2 console. If you are using a command line tool or the Amazon EC2 API to work with Amazon EC2, resource IDs are required for certain commands. For example, if you are using the [stop-instances](#) AWS CLI command to stop an instance, you must specify the instance ID in the command.

Resource ID Length

A resource ID takes the form of a resource identifier (such as `snap` for a snapshot) followed by a hyphen and a unique combination of letters and numbers. Starting in January 2016, we're gradually introducing longer length IDs for some Amazon EC2 and Amazon EBS resource types. The length of the alphanumeric character combination was in an 8-character format; the new IDs are in a 17-character format, for example, `i-1234567890abcdef0` for an instance ID.

Supported resource types will have an opt-in period, during which you can enable the longer ID format. After you've enabled longer IDs for a resource type, any new resources that you create are created with a longer ID unless you explicitly disable the longer ID format. A resource ID does not change after it's created; therefore, your existing resources with shorter IDs are not affected. Similarly, if you disable longer IDs for a resource type, any resources that you created with the longer IDs are not affected.

All supported resource types will have a deadline date, after which all new resources of this type default to the longer ID format, and you can no longer disable the longer ID format. You can enable or disable longer IDs per IAM user and IAM role. By default, an IAM user or role defaults to the same settings as the root user.

Depending on when you created your account, supported resource types may default to using longer IDs. However, you can opt out of using longer IDs until the deadline date for that resource type. For more information, see [Longer EC2 and EBS Resource IDs](#) in the *Amazon EC2 FAQs*.

Resources created with longer IDs are visible to all IAM users and IAM roles, regardless of individual settings and provided that they have permissions to view the relevant resource types.

Topics

- [Working with Longer IDs \(p. 853\)](#)
- [Controlling Access to Longer ID Settings \(p. 856\)](#)

Working with Longer IDs

You can view and modify the longer ID settings for yourself, or for a different IAM user, IAM role, or the root user of the account.

Topics

- [Viewing and Modifying Your Longer ID Settings \(p. 853\)](#)
- [Viewing and Modifying Longer ID Settings for Users or Roles \(p. 855\)](#)

Viewing and Modifying Your Longer ID Settings

You can use the Amazon EC2 console or the AWS CLI to view the resource types that support long IDs, and enable or disable the longer ID format for yourself. The procedures in this section apply to the IAM user or IAM role that's logged into the console or that makes the request; they do not apply to the entire AWS account.

To view and modify the longer ID settings using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation bar at the top of the screen, the current region is displayed. Select the region for which you want to view or change the longer ID settings. Settings are not shared between regions.
3. From the dashboard, under **Account Attributes**, choose **Resource ID length management**. The resource types that support longer IDs are listed. The date at which you're automatically switched over to using longer IDs for each resource type is displayed in the **Deadline** column.
4. To enable the longer ID format for a supported resource type, choose the check box for the **Use Longer IDs** column. To disable the longer ID format, clear the check box.

Important

If you're logged in as the root user, these settings apply to the entire account, unless an IAM user or role logs in and explicitly overrides these settings for themselves. Resources created with longer IDs are visible to all IAM users, regardless of individual settings and provided that they have permissions to view the relevant resource types.

To view and modify longer ID settings using the AWS CLI

To view the longer ID settings of all supported resources, use the [describe-id-format](#) AWS CLI command:

```
aws ec2 describe-id-format

{
  "Statuses": [
    {
      "Deadline": "2016-11-01T13:00:00.000Z",
      "UseLongIds": false,
      "Resource": "instance"
    },
    {
      "Deadline": "2016-11-01T13:00:00.000Z",
      "UseLongIds": true,
      "Resource": "reservation"
    },
    {
      "Deadline": "2016-11-01T13:00:00.000Z",
      "UseLongIds": false,
      "Resource": "volume"
    },
    {
      "Deadline": "2016-11-01T13:00:00.000Z",
      "UseLongIds": false,
      "Resource": "snapshot"
    }
  ]
}
```

The results apply to the IAM user, IAM role, or root user that makes the request; they do not apply to the entire AWS account. The results above indicate that the `instance`, `reservation`, `volume`, and `snapshot` resource types can be enabled or disabled for longer IDs; the `reservation` resource is already enabled. The `Deadline` field indicates the date (in UTC) at which you will be automatically switched over to using longer IDs for that resource. If a deadline date is not yet available, this value is not returned.

To enable longer IDs for a specified resource, use the [modify-id-format](#) AWS CLI command:

```
aws ec2 modify-id-format --resource resource-type --use-long-ids
```

To disable longer IDs for a specified resource, use the [modify-id-format](#) AWS CLI command:

```
aws ec2 modify-id-format --resource resource-type --no-use-long-ids
```

If you're using these actions as the root user, then these settings apply to the entire account, unless an IAM user or role explicitly overrides these settings for themselves. These commands are per-region only. To modify the settings for other regions, use the `--region` parameter in the command.

Note

In the 2015-10-01 version of the Amazon EC2 API, if you call `describe-id-format` or `modify-id-format` using IAM role credentials, the results apply to the entire AWS account, and not the specific IAM role. In the current version of the Amazon EC2 API, the results apply to the IAM role only.

Alternatively, you can use the following commands:

To describe the ID format

- [DescribeIdFormat](#) (Amazon EC2 API)
- [Get-EC2IdFormat](#) (AWS Tools for Windows PowerShell)

To modify the ID format

- [ModifyIdFormat](#) (Amazon EC2 API)
- [Edit-EC2IdFormat](#) (AWS Tools for Windows PowerShell)

Viewing and Modifying Longer ID Settings for Users or Roles

You can view supported resource types and enable the longer ID settings for a specific IAM user, IAM role, or the root user of your account by using the [describe-identity-id-format](#) and [modify-identity-id-format](#) AWS CLI commands. To use these commands, you must specify the ARN of an IAM user, IAM role, or root account user in the request. For example, the ARN of the role 'EC2Role' in account 123456789012 is `arn:aws:iam::123456789012:role/EC2Role`. For more information, see [Principal](#) in the *IAM User Guide*.

To view the longer ID settings of all supported resources for a specific IAM user or IAM role, use the following AWS CLI command:

```
aws ec2 describe-identity-id-format --principal-arn arn-of-iam-principal
```

To enable the longer ID settings for a resource type for a specific IAM user or IAM role, use the following AWS CLI command:

```
aws ec2 modify-identity-id-format --principal-arn arn-of-iam-principal --  
resource resource-type --use-long-ids
```

These commands apply to the ARN specified in the request, they do not apply to the IAM user, IAM role, or root user that made the request.

You can enable the longer ID settings for all IAM users, IAM roles, and the root user of your account by using the following AWS CLI command:

```
aws ec2 modify-identity-id-format --principal-arn all --resource resource-  
type --use-long-ids
```

Alternatively, you can use the following commands:

To describe the ID format

- [DescribeIdentityIdFormat](#) (Amazon EC2 API)
- [Get-EC2IdentityIdFormat](#) (AWS Tools for Windows PowerShell)

To modify the ID format

- [ModifyIdentityIdFormat](#) (Amazon EC2 API)
- [Edit-EC2IdentityIdFormat](#) (AWS Tools for Windows PowerShell)

Controlling Access to Longer ID Settings

By default, IAM users and roles do not have permission to use the `ec2:DescribeIdFormat`, `ec2:DescribeIdentityIdFormat`, `ec2:ModifyIdFormat`, and `ec2:ModifyIdentityIdFormat` actions unless they're explicitly granted permission through their associated IAM policies. For example, an IAM role may have permission to use all Amazon EC2 actions through an `"Action": "ec2:*"` element in the policy statement.

To prevent IAM users and roles from viewing or modifying the longer resource ID settings for themselves or other users and roles in your account, ensure that the IAM policy contains the following statement:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "ec2:ModifyIdFormat",
        "ec2:DescribeIdFormat",
        "ec2:ModifyIdentityIdFormat",
        "ec2:DescribeIdentityIdFormat"
      ],
      "Resource": "*"
    }
  ]
}
```

We do not support resource-level permissions for the `ec2:DescribeIdFormat`, `ec2:DescribeIdentityIdFormat`, `ec2:ModifyIdFormat`, and `ec2:ModifyIdentityIdFormat` actions.

Listing and Filtering Your Resources

You can get a list of some types of resource using the Amazon EC2 console. You can get a list of each type of resource using its corresponding command or API action. If you have many resources, you can filter the results to include only the resources that match certain criteria.

Topics

- [Advanced Search \(p. 856\)](#)
- [Listing Resources Using the Console \(p. 857\)](#)
- [Filtering Resources Using the Console \(p. 858\)](#)
- [Listing and Filtering Using the CLI and API \(p. 859\)](#)

Advanced Search

Advanced search allows you to search using a combination of filters to achieve precise results. You can filter by keywords, user-defined tag keys, and predefined resource attributes.

The specific search types available are:

- **Search by keyword**

To search by keyword, type or paste what you're looking for in the search box, and then choose Enter. For example, to search for a specific instance, you can type the instance ID.

- **Search by fields**

You can also search by fields, tags, and attributes associated with a resource. For example, to find all instances in the stopped state:

1. In the search box, start typing **Instance state**. As you type, you'll see a list of suggested fields.
2. Select **Instance State** from the list.
3. Select **Stopped** from the list of suggested values.
4. To further refine your list, select the search box for more search options.

- **Advanced search**

You can create advanced queries by adding multiple filters. For example, you can search by tags and see instances for the Flying Mountain project running in the Production stack, and then search by attributes to see all t2.micro instances, or all instances in us-west-2a, or both.

- **Inverse search**

You can search for resources that do not match a specified value. For example, to list all instances that are not terminated, search by the **Instance State** field, and prefix the Terminated value with an exclamation mark (!).

- **Partial search**

When searching by field, you can also enter a partial string to find all resources that contain the string in that field. For example, search by **Instance Type**, and then type **t2** to find all t2.micro, t2.small or t2.medium instances.

- **Regular expression**

Regular expressions are useful when you need to match the values in a field with a specific pattern. For example, search by the Name tag, and then type **^s.*** to see all instances with a Name tag that starts with an 's'. Regular expression search is not case-sensitive.

After you have the precise results of your search, you can bookmark the URL for easy reference. In situations where you have thousands of instances, filters and bookmarks can save you a great deal of time; you don't have to run searches repeatedly.

Combining search filters

In general, multiple filters with the same key field (e.g., tag:Name, search, Instance State) are automatically joined with OR. This is intentional, as the vast majority of filters would not be logical if they were joined with AND. For example, you would get zero results for a search on Instance State=running AND Instance State=stopped. In many cases, you can granulate the results by using complementary search terms on different key fields, where the AND rule is automatically applied instead. If you search for tag: Name:=All values and tag:Instance State=running, you get search results that contain both those criteria. To fine-tune your results, simply remove one filter in the string until the results fit your requirements.

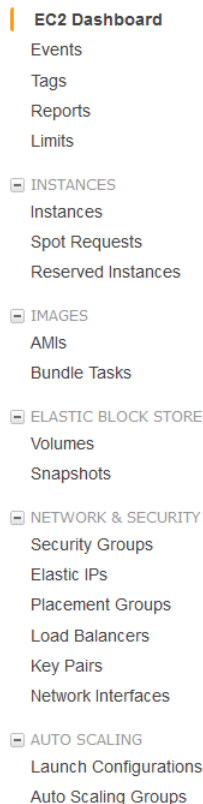
Listing Resources Using the Console

You can view the most common Amazon EC2 resource types using the console. To view additional resources, use the command line interface or the API actions.

To list EC2 resources using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

2. In the navigation pane, choose the option that corresponds to the resource, such as **AMIs** or **Instances**.



3. The page displays all the available resources.

Filtering Resources Using the Console

You can perform filtering and sorting of the most common resource types using the Amazon EC2 console. For example, you can use the search bar on the instances page to sort instances by tags, attributes, or keywords.

You can also use the search field on each page to find resources with specific attributes or values. You can use regular expressions to search on partial or multiple strings. For example, to find all instances that are using the MySG security group, enter `MySG` in the search field. The results will include any values that contain `MySG` as a part of the string, such as `MySG2` and `MySG3`. To limit your results to `MySG` only, enter `\bMySG\b` in the search field. To list all the instances whose type is either `m1.small` or `m1.large`, enter `m1.small|m1.large` in the search field.

To list volumes in the `us-east-1b` Availability Zone with a status of `available`

1. In the navigation pane, choose **Volumes**.
2. Click on the search box, select **Attachment Status** from the menu, and then select **Detached**. (A detached volume is available to be attached to an instance in the same Availability Zone.)
3. Click on the search box again, select **State**, and then select **Available**.
4. Click on the search box again, select **Availability Zone**, and then select `us-east-1b`.
5. Any volumes that meet this criteria are displayed.

To list public 64-bit Windows AMIs backed by Amazon EBS

1. In the navigation pane, choose **AMIs**.
2. In the **Filter** pane, select **Public images**, **EBS images**, and then **Windows** from the **Filter** lists.
3. Enter `x86_64` in the search field.
4. Any AMIs that meet this criteria are displayed.

Listing and Filtering Using the CLI and API

Each resource type has a corresponding CLI command or API request that you use to list resources of that type. For example, you can list Amazon Machine Images (AMI) using `ec2-describe-images` or `DescribeImages`. The response contains information for all your resources.

The resulting lists of resources can be long, so you might want to filter the results to include only the resources that match certain criteria. You can specify multiple filter values, and you can also specify multiple filters. For example, you can list all the instances whose type is either `m1.small` or `m1.large`, and that have an attached EBS volume that is set to delete when the instance terminates. The instance must match all your filters to be included in the results.

Note

If you use a tag filter, the response includes the tags for your resources; otherwise, tags may be omitted in the response.

You can also use wildcards with the filter values. An asterisk (*) matches zero or more characters, and a question mark (?) matches exactly one character. For example, you can use `*database*` as a filter value to get all EBS snapshots that include `database` in the description. If you were to specify `database` as the filter value, then only snapshots whose description equals `database` would be returned. Filter values are case sensitive. We support only exact string matching, or substring matching (with wildcards). If a resulting list of resources is long, using an exact string filter may return the response faster.

Tip

Your search can include the literal values of the wildcard characters; you just need to escape them with a backslash before the character. For example, a value of `*amazon?\` searches for the literal string `*amazon?\`.

For a list of supported filters per Amazon EC2 resource, see the relevant documentation:

- For the AWS CLI, see the relevant `describe` command in the [AWS Command Line Interface Reference](#).
- For Windows PowerShell, see the relevant `Get` command in the [AWS Tools for Windows PowerShell Reference](#).
- For the Query API, see the relevant `Describe` API action in the [Amazon EC2 API Reference](#).

Tagging Your Amazon EC2 Resources

To help you manage your instances, images, and other Amazon EC2 resources, you can optionally assign your own metadata to each resource in the form of *tags*. This topic describes tags and shows you how to create them.

Contents

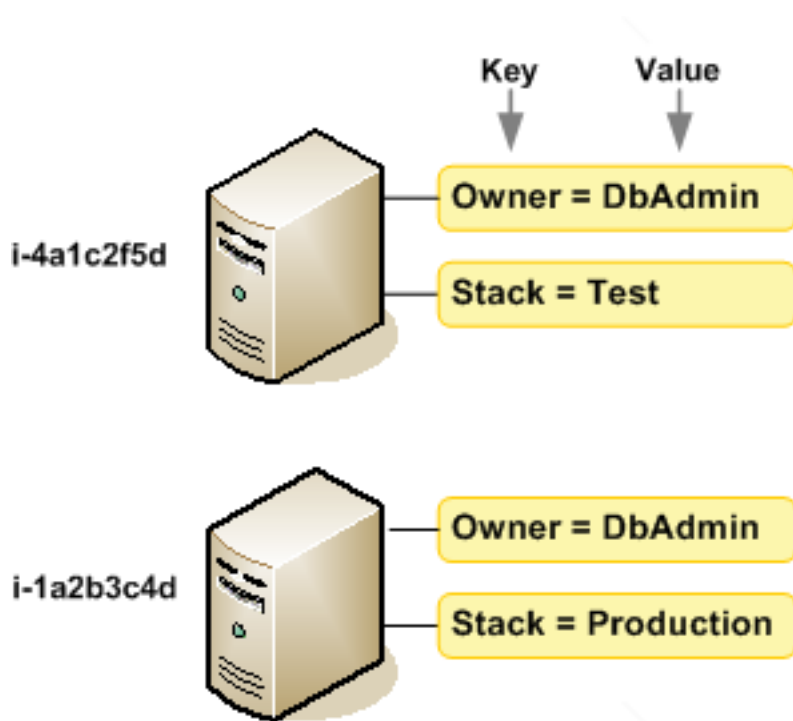
- [Tag Basics \(p. 860\)](#)
- [Tag Restrictions \(p. 861\)](#)
- [Tagging Your Resources for Billing \(p. 862\)](#)

- [Working with Tags Using the Console \(p. 863\)](#)
- [Working with Tags Using the CLI or API \(p. 868\)](#)

Tag Basics

Tags enable you to categorize your AWS resources in different ways, for example, by purpose, owner, or environment. This is useful when you have many resources of the same type — you can quickly identify a specific resource based on the tags you've assigned to it. Each tag consists of a key and an optional value, both of which you define. For example, you could define a set of tags for your account's Amazon EC2 instances that helps you track each instance's owner and stack level. We recommend that you devise a set of tag keys that meets your needs for each resource type. Using a consistent set of tag keys makes it easier for you to manage your resources. You can search and filter the resources based on the tags you add.

The following diagram illustrates how tagging works. In this example, you've assigned two tags to each of your instances, one called `Owner` and another called `Stack`. Each of the tags also has an associated value.



Tags don't have any semantic meaning to Amazon EC2 and are interpreted strictly as a string of characters. Also, tags are not automatically assigned to your resources.

You can work with tags using the AWS Management Console, the Amazon EC2 command line interface (CLI), the AWS CLI, and the Amazon EC2 API.

You can assign tags only to resources that already exist. You cannot assign tags when you create a resource; for example, when you use the `run-instances` AWS CLI command. When you use the Amazon EC2 console, some resource creation screens enable you to specify tags which are applied immediately after the resource is created. If you add a tag that has the same key as an existing tag on that resource, the new value overwrites the old value. You can edit tag keys and values, and you can

remove tags from a resource at any time. You can set a tag's value to the empty string, but you can't set a tag's value to null.

If you're using AWS Identity and Access Management (IAM), you can control which users in your AWS account have permission to create, edit, or delete tags. For more information about IAM, see [Controlling Access to Amazon EC2 Resources \(p. 619\)](#).

Tag Restrictions

The following basic restrictions apply to tags:

- Maximum number of tags per resource—50
- Maximum key length—127 Unicode characters in UTF-8
- Maximum value length—255 Unicode characters in UTF-8
- Tag keys and values are case sensitive.
- Do not use the `aws:` prefix in your tag names or values because it is reserved for AWS use. You can't edit or delete tag names or values with this prefix. Tags with this prefix do not count against your tags per resource limit.
- If your tagging schema will be used across multiple services and resources, remember that other services may have restrictions on allowed characters. Generally allowed characters are: letters, spaces, and numbers representable in UTF-8, plus the following special characters: `+ - = . _ : / @`.

You can't terminate, stop, or delete a resource based solely on its tags; you must specify the resource identifier. For example, to delete snapshots that you tagged with a tag key called `DeleteMe`, you must use the `DeleteSnapshots` action with the resource identifiers of the snapshots, such as `snap-1234567890abcdef0`. To identify resources by their tags, you can use the `DescribeTags` action to list all of your tags and their associated resources. You can also filter by resource type or tag keys and values. You can't call `DeleteSnapshots` with a filter that specified the tag. For more information about using filters when listing your resources, see [Listing and Filtering Your Resources \(p. 856\)](#).

You can tag public or shared resources, but the tags you assign are available only to your AWS account and not to the other accounts sharing the resource.

You can't tag all resources, and some you can only tag using API actions or the command line. The following table lists all Amazon EC2 resources and the tagging restrictions that apply to them, if any. Resources with tagging restrictions of None can be tagged with API actions, the CLI, and the console.

Resource	Tagging support	Tagging restrictions
AMI	Yes	None
Bundle task	No	
Customer gateway	Yes	None
Dedicated Host	No	
DHCP option	Yes	None
EBS volume	Yes	None
Instance store volume	No	
Elastic IP	No	
Egress-only Internet gateway	No	
Instance	Yes	None

Resource	Tagging support	Tagging restrictions
Internet gateway	Yes	None
Key pair	No	
NAT gateway	No	
Network ACL	Yes	None
Network interface	Yes	None
Placement group	No	
Reserved Instance	Yes	None
Reserved Instance listing	No	
Route table	Yes	None
Spot instance request	Yes	None
Security group - EC2-Classical	Yes	None
Security group - VPC	Yes	None
Snapshot	Yes	None
Subnet	Yes	None
Virtual private gateway	Yes	None
VPC	Yes	None
VPC endpoint	No	
VPC flow log	No	
VPC peering connection	Yes	None
VPN connection	Yes	None

For more information about tagging using the AWS Management Console, see [Working with Tags Using the Console \(p. 863\)](#). For more information about tagging using the API or command line, see [Working with Tags Using the CLI or API \(p. 868\)](#).

Tagging Your Resources for Billing

You can use tags to organize your AWS bill to reflect your own cost structure. To do this, sign up to get your AWS account bill with tag key values included. For more information about setting up a cost allocation report with tags, see [Setting Up Your Monthly Cost Allocation Report](#) in *About AWS Account Billing*. To see the cost of your combined resources, you can organize your billing information based on resources that have the same tag key values. For example, you can tag several resources with a specific application name, and then organize your billing information to see the total cost of that application across several services. For more information, see [Cost Allocation and Tagging](#) in *About AWS Account Billing*.

Note

If you've just enabled reporting, the current month's data will be available for viewing in about 24 hours.

Working with Tags Using the Console

Using the Amazon EC2 console, you can see which tags are in use across all of your Amazon EC2 resources in the same region. You can view tags by resource and by resource type, and you can also view how many items of each resource type are associated with a specified tag. You can also use the Amazon EC2 console to apply or remove tags from one or more resources at a time.

For ease of use and best results, use Tag Editor in the AWS Management Console, which provides a central, unified way to create and manage your tags. For more information, see [Working with Tag Editor](#) in [Getting Started with the AWS Management Console](#).

Contents

- [Displaying Tags](#) (p. 863)
- [Adding and Deleting Tags on an Individual Resource](#) (p. 864)
- [Adding and Deleting Tags to a Group of Resources](#) (p. 865)
- [Adding a Tag When You Launch an Instance](#) (p. 867)
- [Filtering a List of Resources by Tag](#) (p. 868)

Displaying Tags

You can display tags in two different ways in the Amazon EC2 console. You can display the tags for an individual resource or for all resources.

To display tags for individual resources

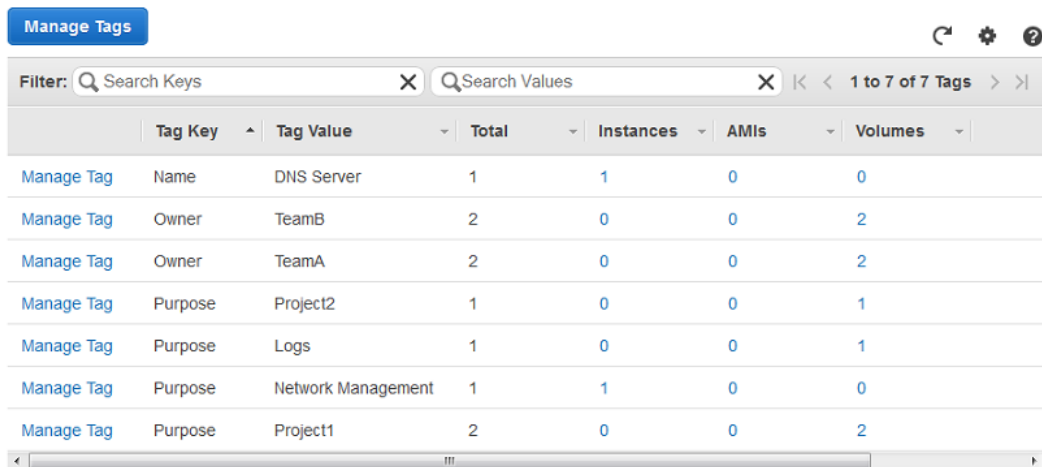
When you select a resource-specific page in the Amazon EC2 console, it displays a list of those resources. For example, if you select **Instances** from the navigation pane, the console displays a list of Amazon EC2 instances. When you select a resource from one of these lists (e.g., an instance), if the resource supports tags, you can view and manage its tags. On most resource pages, you can view the tags in the **Tags** tab on the details pane.

You can add a column to the resource list that displays all values for tags with the same key. This column enables you to sort and filter the resource list by the tag. There are two ways to add a new column to the resource list to display your tags.

- On the **Tags** tab, select **Show Column**. A new column will be added to the console.
- Choose the **Show/Hide Columns** gear-shaped icon, and in the **Show/Hide Columns** dialog box, select the tag key under **Your Tag Keys**.

To display tags for all resources

You can display tags across all resources by selecting **Tags** from the navigation pane in the Amazon EC2 console. The following image shows the **Tags** pane, which lists all tags in use by resource type.



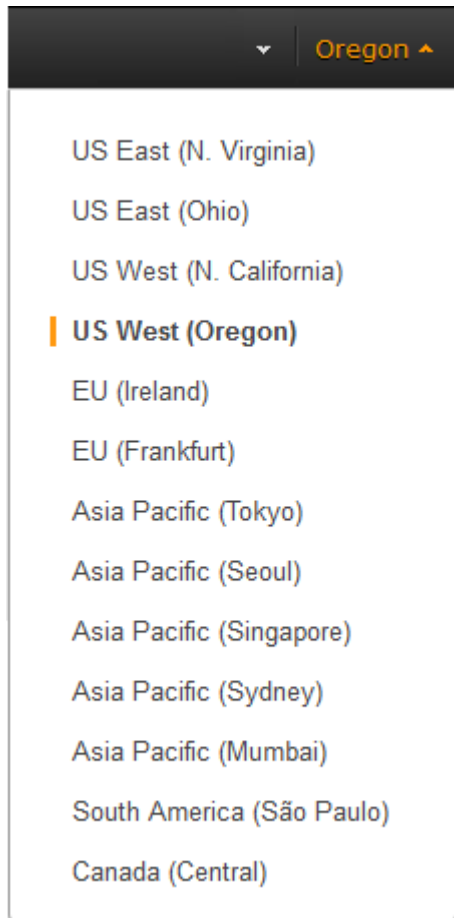
	Tag Key	Tag Value	Total	Instances	AMIs	Volumes
Manage Tag	Name	DNS Server	1	1	0	0
Manage Tag	Owner	TeamB	2	0	0	2
Manage Tag	Owner	TeamA	2	0	0	2
Manage Tag	Purpose	Project2	1	0	0	1
Manage Tag	Purpose	Logs	1	0	0	1
Manage Tag	Purpose	Network Management	1	1	0	0
Manage Tag	Purpose	Project1	2	0	0	2

Adding and Deleting Tags on an Individual Resource

You can manage tags for an individual resource directly from the resource's page.

To add a tag to an individual resource

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select the region that meets your needs. This choice is important because some Amazon EC2 resources can be shared between regions, while others can't. For more information, see [Resource Locations \(p. 851\)](#).



3. In the navigation pane, select a resource type (for example, **Instances**).
4. Select the resource from the resource list.
5. Select the **Tags** tab in the details pane.
6. Choose the **Add/Edit Tags** button.
7. In the **Add/Edit Tags** dialog box, specify the key and value for each tag, and then choose **Save**.

To delete a tag from an individual resource

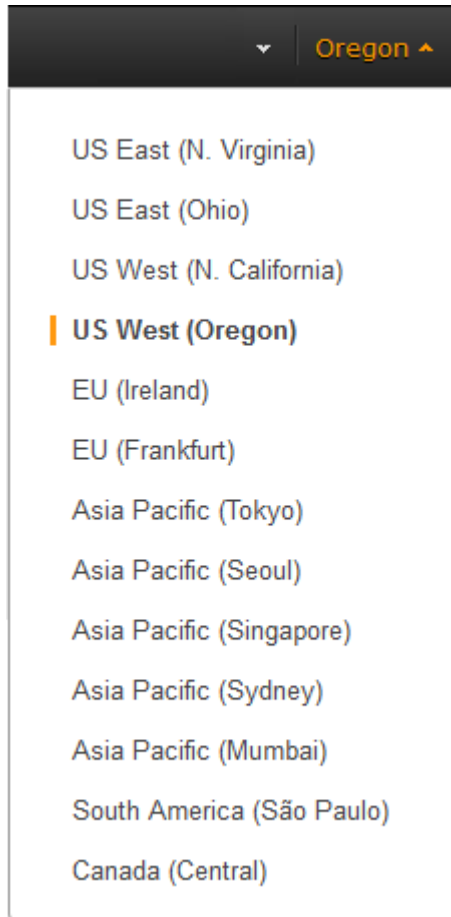
1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select the region that meets your needs. This choice is important because some Amazon EC2 resources can be shared between regions, while others can't. For more information, see [Resource Locations](#) (p. 851).
3. In the navigation pane, choose a resource type (for example, **Instances**).
4. Select the resource from the resource list.
5. Select the **Tags** tab in the details pane.
6. Choose **Add/Edit Tags**, select the **Delete** icon for the tag, and choose **Save**.

Adding and Deleting Tags to a Group of Resources

To add a tag to a group of resources

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

2. From the navigation bar, select the region that meets your needs. This choice is important because some Amazon EC2 resources can be shared between regions, while others can't. For more information, see [Resource Locations](#) (p. 851).



3. In the navigation pane, choose **Tags**.
4. At the top of the content pane, choose **Manage Tags**.
5. From the **Filter** drop-down list, select the type of resource (for example, instances) that you want to add tags to.
6. In the resources list, select the check box next to each resource that you want to add tags to.
7. In the **Key** and **Value** boxes under **Add Tag**, type the tag key and values you want, and then choose **Add Tag**.

Note

If you add a new tag with the same tag key as an existing tag, the new tag overwrites the existing tag.

To remove a tag from a group of resources

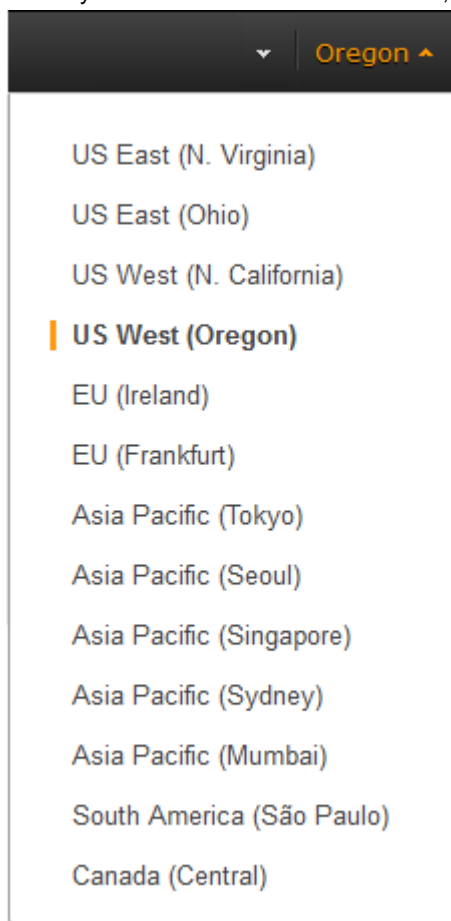
1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select the region that meets your needs. This choice is important because some Amazon EC2 resources can be shared between regions, while others can't. For more information, see [Resource Locations](#) (p. 851).
3. In the navigation pane, choose **Tags**.
4. At the top of the content pane, choose **Manage Tags**.

5. To view the tags in use, select the **Show/Hide Columns** gear-shaped icon, and in the **Show/Hide Columns** dialog box, select the tag keys you want to view, and then choose **Close**.
6. From the **Filter** drop-down list, select the type of resource (for example, instances) that you want to remove tags from.
7. In the resource list, select the check box next to each resource that you want to remove tags from.
8. Under **Remove Tag**, type the tag's name in the **Key** box, and then choose **Remove Tag**.

Adding a Tag When You Launch an Instance

To add a tag using the Launch Wizard

1. From the navigation bar, select the region for the instance. This choice is important because some Amazon EC2 resources can be shared between regions, while others can't. Select the region that meets your needs. For more information, see [Resource Locations \(p. 851\)](#).



2. Choose **Launch Instance**.
3. The **Choose an Amazon Machine Image (AMI)** page displays a list of basic configurations called Amazon Machine Images (AMIs). Choose the AMI that you want to use and choose **Select**. For more information about selecting an AMI, see [Finding a Windows AMI \(p. 67\)](#).
4. On the **Configure Instance Details** page, configure the instance settings as necessary, and then choose **Next: Add Storage**.
5. On the **Add Storage** page, you can specify additional storage volumes for your instance. Choose **Next: Tag Instance** when done.

6. On the **Tag Instance** page, specify tags for the instance by providing key and value combinations. Choose **Create Tag** to add more than one tag to your instance. Choose **Next: Configure Security Group** when you are done.
7. On the **Configure Security Group** page, you can choose from an existing security group that you own, or let the wizard create a new security group for you. Choose **Review and Launch** when you are done.
8. Review your settings. When you're satisfied with your selections, choose **Launch**. Select an existing key pair or create a new one, select the acknowledgment check box, and then choose **Launch Instances**.

Filtering a List of Resources by Tag

You can filter your list of resources based on one or more tag keys and tag values.

To filter a list of resources by tag

1. Display a column for the tag as follows:
 - a. Select one of the resources.
 - b. Select the **Tags** tab in the details pane.
 - c. Locate the tag in the list and choose **Show Column**.
2. Choose the filter icon in the top right corner of the column for the tag to display the filter list.
3. Select the tag values, and then choose **Apply Filter** to filter the results list.

Note

For more information about filters see [Listing and Filtering Your Resources \(p. 856\)](#).

Working with Tags Using the CLI or API

Use the following to add, update, list, and delete the tags for your resources. The corresponding documentation provides examples.

Task	AWS CLI	AWS Tools for Windows PowerShell	API Action
Add or overwrite one or more tags.	create-tags	New-EC2Tag	CreateTags
Delete one or more tags.	delete-tags	Remove-EC2Tag	DeleteTags
Describe one or more tags.	describe-tags	Get-EC2Tag	DescribeTags

You can also filter a list of resources according to their tags. The following examples demonstrate how to filter your instances using tags with the [describe-instances](#) command.

Example 1: Describe instances with the specified tag key

The following command describes the instances with a Stack tag, regardless of the value of the tag.

```
aws ec2 describe-instances --filters Name=tag-key,Values=Stack
```

Example 2: Describe instances with the specified tag

The following command describes the instances with the tag Stack=production.

```
aws ec2 describe-instances --filters Name=tag:Stack,Values=production
```

Example 3: Describe instances with the specified tag value

The following command describes the instances with a tag with the value production, regardless of the tag key.

```
aws ec2 describe-instances --filters Name=tag-value,Values=production
```

Important

If you describe resources without using a tag filter, the results may not return the tags for your resources. To ensure that tags are returned in results, we recommend that you either describe tags (and use a resource filter if necessary), or describe your resources and use one or more tag filters.

Amazon EC2 Service Limits

Amazon EC2 provides different *resources* that you can use. These resources include images, instances, volumes, and snapshots. When you create your AWS account, we set default limits on these resources on a per-region basis. For example, there is a limit on the number of instances that you can launch in a region. Therefore, when you launch an instance in the US West (Oregon) Region, the request must not cause your usage to exceed your current instance limit in that region.

The Amazon EC2 console provides limit information for the resources managed by the Amazon EC2 and Amazon VPC consoles. You can request an increase for many of these limits. Use the limit information that we provide to manage your AWS infrastructure. Plan to request any limit increases in advance of the time that you'll need them.

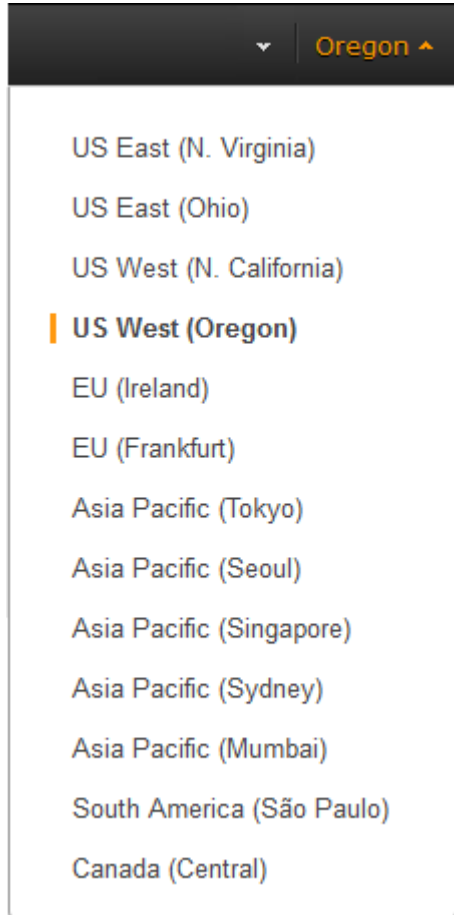
For more information about the limits for other services, see [AWS Service Limits](#) in the *Amazon Web Services General Reference*.

Viewing Your Current Limits

Use the **EC2 Service Limits** page in the Amazon EC2 console to view the current limits for resources provided by Amazon EC2 and Amazon VPC, on a per-region basis.

To view your current limits

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select a region.



3. From the navigation pane, choose **Limits**.
4. Locate the resource in the list. The **Current Limit** column displays the current maximum for that resource for your account.

Requesting a Limit Increase

Use the **Limits** page in the Amazon EC2 console to request an increase in the limits for resources provided by Amazon EC2 or Amazon VPC, on a per-region basis.

To request a limit increase

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select a region.
3. From the navigation pane, choose **Limits**.
4. Locate the resource in the list. Choose **Request limit increase**.
5. Complete the required fields on the limit increase form. We'll respond to you using the contact method that you specified.

Amazon EC2 Usage Reports

The usage reports provided by Amazon EC2 enable you to analyze the usage of your instances in depth. The data in the usage reports is updated multiple times each day. You can filter the reports by

AWS account, region, Availability Zone, operating system, instance type, purchasing option, tenancy, and tags.

To get usage and cost data for an account, you must have its account credentials and enable detailed billing reports with resources and tags for the account. If you're using consolidated billing and are logged into the payer account, you can view data for the payer account and all its linked accounts. If you're using consolidated billing and are logged into one of the linked accounts, you can only view data for that linked account. For information about consolidated billing, see [Pay Bills for Multiple Accounts with Consolidated Billing](#).

Topics

- [Available Reports](#) (p. 871)
- [Getting Set Up for Usage Reports](#) (p. 871)
- [Granting IAM Users Access to the Amazon EC2 Usage Reports](#) (p. 872)
- [Instance Usage Report](#) (p. 873)
- [Reserved Instance Utilization Reports](#) (p. 876)

Available Reports

You can generate the following reports:

- [Instance usage report](#) (p. 873). This report covers your usage of On-Demand instances, Spot instances, and Reserved Instances.
- [Reserved Instances utilization report](#) (p. 876). This report covers the usage of your capacity reservation.

To access the reports, open the AWS Management Console. In the navigation pane, choose **Reports** then choose the report you'd like to view.

Getting Set Up for Usage Reports

Before you begin, enable detailed billing reports with resources and tags as shown in the following procedure. After you complete this procedure, we'll start collecting usage data for your instances. If you've already enabled detailed billing reports, you can access the usage data that we've been collecting since you enabled them.

Important

To complete these procedures, you must log in using your AWS account credentials. You can't complete these procedures if you log in using IAM user credentials.

To enable detailed billing reports

1. Select an existing Amazon S3 bucket to receive your usage data. Be sure to manage access to this bucket as it contains your billing data. (We don't require that you keep these files; in fact, you can delete them immediately if you don't need them.) If you don't have a bucket, create one as follows:
 - a. Open the Amazon S3 console.
 - b. Select **Create Bucket**.
 - c. In the **Create a Bucket** dialog box, enter a name for your bucket (for example, *username-ec2-usage-data*), select a region, and then choose **Create**. For more information about the requirements for bucket names, see [Creating a Bucket](#) in the *Amazon Simple Storage Service Console User Guide*.

2. Open the Billing and Cost Management console at <https://console.aws.amazon.com/billing/home?#>.
3. Choose **Preferences** in the navigation pane.
4. Select **Receive Billing Reports**.
5. Specify the name of your Amazon S3 bucket in **Save to S3 Bucket**, and then select **Verify**.
6. Grant AWS permission to publish usage data to your Amazon S3 bucket.
 - a. Under **Receive Billing Reports**, choose **sample policy**. Copy the sample policy. Notice that the sample policy uses the bucket name you specified.
 - b. Open the Amazon S3 console in another browser tab. Select your bucket, choose **Properties**, and then expand **Permissions**. In the **Permissions** section, choose **Add bucket policy**. Paste the sample policy into the text area and choose **Save**. In the **Permissions** section, choose **Save**.
 - c. Return to the browser tab with the sample policy and choose **Done**.
7. Under **Report**, select **Detailed billing report with resources and tags**.
8. Choose **Save preferences**.

Note

It can take up to a day before you can see your data in the reports.

You can categorize your instances using tags. After you tag your instances, you must enable reporting on these tags.

To enable usage reporting by tag

1. Tag your instances. For best results, ensure that you add each tag you plan to use for reporting to each of your instances. For more information about how to tag an instance, see [Tagging Your Amazon EC2 Resources \(p. 859\)](#).
2. Open the Billing and Cost Management console at <https://console.aws.amazon.com/billing/home?#>.
3. Select **Preferences** in the navigation pane.
4. Under **Report**, choose **Manage report tags**.
5. The page displays the list of tags that you've created. Select the tags that you'd like to use to filter or group your instance usage data, and then click **Save**. We automatically exclude any tags that you don't select from your instance usage report.

Note

We apply these changes only to the data for the current month. It can take up to a day for these changes to take effect.

Granting IAM Users Access to the Amazon EC2 Usage Reports

By default, IAM users can't access the Amazon EC2 usage reports. You must create an IAM policy that grants IAM users permission to access these reports.

The following policy allows users to view both Amazon EC2 usage reports.

```
{
  "Version": "2012-10-17",
  "Statement": [ {
```

```
"Effect": "Allow",
"Action": "ec2-reports:*",
"Resource": "*"
}
]
```

The following policy allows users to view the instance usage report.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "ec2-reports:ViewInstanceUsageReport",
    "Resource": "*"
  }
]
```

The following policy allows users to view the Reserved Instances utilization report.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "ec2-reports:ViewReservedInstanceUtilizationReport",
    "Resource": "*"
  }
]
```

For more information, see [Permissions and Policies](#) in the *IAM User Guide*.

Instance Usage Report

You can use the instance usage report to view your instance usage and cost trends. You can see your usage data in either instance hours or cost. You can choose to see hourly, daily and monthly aggregates of your usage data. You can filter or group the report by region, Availability Zone, instance type, AWS account, platform, tenancy, purchase option, or tag. After you configure a report, you can bookmark it so that it's easy to get back to later.

Here's an example of some of the questions that you can answer by creating an instance usage report:

- How much am I spending on instances of each instance type?
- How many instance hours are being used by a particular department?
- How is my instance usage distributed across Availability Zones?
- How is my instance usage distributed across AWS accounts?

Topics

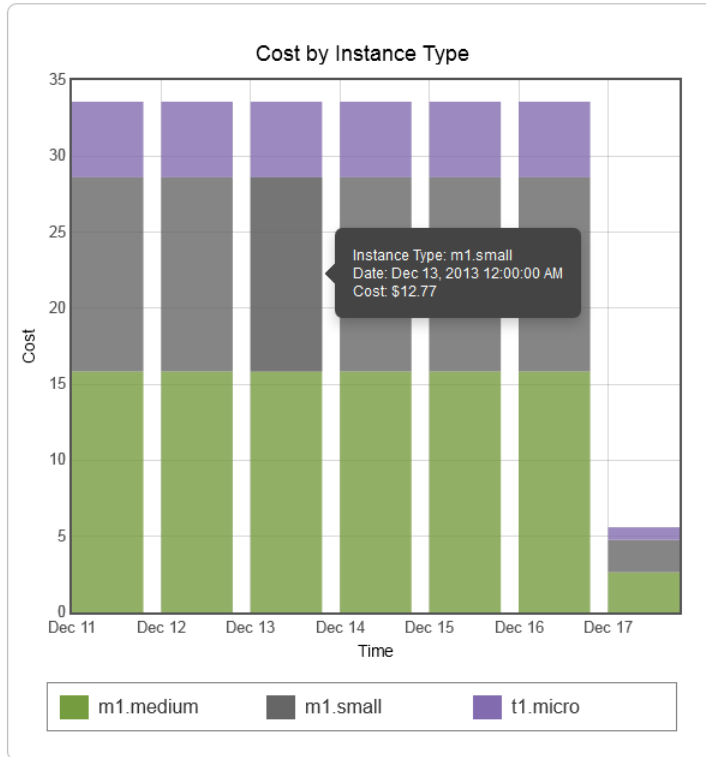
- [Report Formats](#) (p. 874)
- [Viewing Your Instance Usage](#) (p. 875)
- [Bookmarking a Customized Report](#) (p. 876)

- [Exporting Your Usage Data \(p. 876\)](#)

Report Formats

We display the usage data that you request as both a graph and a table.

For example, the following graph displays cost by instance type. The key for the graph indicates which color represents which instance type. To get detailed information about a segment of a bar, hover over it.



The corresponding table displays one column for each instance type. Notice that we include a color band in the column head that is the same color as the instance type in the graph.

Time (UTC)	m1.medium	m1.small	t1.micro
12/11/13	\$15.84	\$12.77	\$4.97
12/12/13	\$15.84	\$12.77	\$4.97
12/13/13	\$15.84	\$12.77	\$4.97
12/14/13	\$15.84	\$12.77	\$4.97
12/15/13	\$15.84	\$12.77	\$4.97
12/16/13	\$15.84	\$12.77	\$4.97
12/17/13	\$2.64	\$2.13	\$0.83
Total	\$97.68	\$78.75	\$30.65

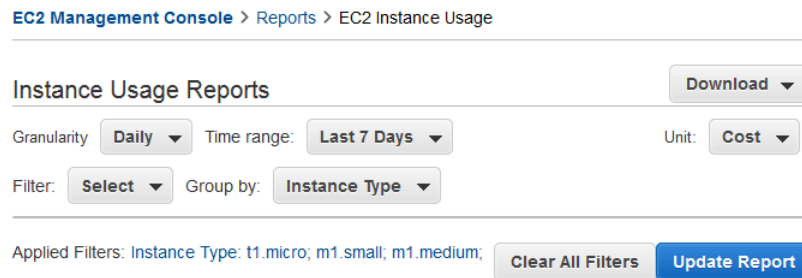
Viewing Your Instance Usage

The following procedures demonstrate how to generate usage reports using some of the capabilities we provide.

Before you begin, you must get set up. For more information, see [Getting Set Up for Usage Reports](#) (p. 871).

To filter and group your instance usage by instance type

1. Open the Amazon EC2 console.
2. In the navigation pane, choose **Reports** and then select **EC2 Instance Usage Report**.
3. Select an option for **Unit**. To view the time that your instances have been running, in hours, select `Instance Hours`. To view the cost of your instance usage, select `Cost`.
4. Select options for **Granularity** and **Time range**.
 - To view the data summarized for each hour in the time range, select `Hourly` granularity. You can select a time range of up to 2 days when viewing hourly data.
 - To view the data summarized for each day in the time range, select `Daily` granularity. You can select a time range of up to 2 months when viewing daily data.
 - To view the data summarized for each month in the time range, select `Monthly` granularity.
5. In the **Filter** list, select `Instance Type`. In the **Group by** list, select `Instance Type`.
6. In the filter area, select one or more instance types and then select **Update Report**. The filters you specify appear under **Applied Filters**.



Notice that you can return to the Amazon EC2 console by choosing either **Reports** or **EC2 Management Console** at the top of the page.

To group your instance usage based on tags

1. Open the Instance Usage Reports page.
2. Select an option for **Unit**. To view the time that your instances have been running, in hours, select `Instance Hours`. To view the cost of your instance usage, select `Cost`.
3. Select options for **Granularity** and **Time range**.
 - To view the data summarized for each hour in the time range, select `Hourly` granularity. You can select a time range of up to 2 days when viewing hourly data.
 - To view the data summarized for each day in the time range, select `Daily` granularity. You can select a time range of up to 2 months when viewing daily data.
 - To view the data summarized for each month in the time range, select `Monthly` granularity.
4. In the **Group by** list, select **Tag**.
5. Choose the **Key Name** box, select a name from the list, and then choose **Update Report**. If there are no items in this list, you must enable usage reporting by tag. For more information, see [To enable usage reporting by tag](#) (p. 872).

Instance Usage Reports

Granularity: **Daily** Time range: **Last 14 Days** Unit: **Instance Hours**

Filter: **Select** Group by: **Tag** Key Name: Project

Applied Filters: None

Clear All Filters **Update Report**

Bookmarking a Customized Report

You might want to generate a customized report again. Do this by bookmarking the report.

To bookmark a custom report

1. Select the options and filters for your report. Each selection you make adds a parameter to the console URL. For example, `granularity=Hourly` and `Filters=filter_list`.
2. Using your browser, add the console URL as a bookmark.
3. To generate the same report in the future, use the bookmark that you created.

Exporting Your Usage Data

You might want to include your report graph or table in other reports. Do this by exporting the data.

To export usage data

1. Select the options and filters for your report.
2. To export the usage data from the table as a `.csv` file, choose **Download** and select **CSV Only**.
3. To export the graphical usage data as a `.png` file, choose **Download** and select **Graph Only**.

Reserved Instance Utilization Reports

The Reserved Instance utilization report describes the utilization over time of each group (or *bucket*) of Amazon EC2 Reserved Instances that you own. Each bucket has a unique combination of region, Availability Zone, instance type, tenancy, offering type, and platform. You can specify the time range that the report covers, from a custom range to weeks, months, a year, or three years. The available data depends on when you enable detailed billing reports for the account (see [Getting Set Up for Usage Reports \(p. 871\)](#)). The Reserved Instance utilization report compares the Reserved Instance prices paid for instance usage in the bucket with On-Demand prices and shows your savings for the time range covered by the report.

To get usage and cost data for an account, you must have its account credentials and enable detailed billing reports with resources and tags for the account. If you're using consolidated billing and are logged into the payer account, you can view data for the payer account and all its linked accounts. If you're using consolidated billing and are logged into one of the linked accounts, you can only view data for that linked account. For information about consolidated billing, see [Pay Bills for Multiple Accounts with Consolidated Billing](#).

Note

The Reserved Instance buckets aggregate Reserved Instances across EC2-VPC and EC2-Classic network platform types in the same way that your bill is calculated. Additionally, Reserved Instances in a bucket may have different upfront and hourly prices.

Here are examples of some of the questions that you can answer using the Reserved Instance utilization report:

- How well am I utilizing my Reserved Instances?
- Are my Reserved Instances helping me save money?

Before you begin, you must get set up. For more information, see [Getting Set Up for Usage Reports \(p. 871\)](#).

Topics

- [Getting to Know the Report \(p. 877\)](#)
- [Viewing Your Reserved Instance Utilization \(p. 878\)](#)
- [Bookmarking a Customized Report \(p. 879\)](#)
- [Exporting Your Usage Data \(p. 880\)](#)
- [Options Reference \(p. 880\)](#)

Getting to Know the Report

The Reserved Instance utilization report displays your requested utilization data in graph and table formats.

To access the report, open the AWS Management Console. In the navigation pane, choose **Reports** and then select **EC2 Reserved Instance Usage Report**.

The report aggregates Reserved Instance usage data for a given period by bucket. In the report, each row in the table represents a bucket and provides the following metrics:

- **Count**—The highest number of Reserved Instances owned at the same time during the period of the report.
- **Usage Cost**—The total Reserved Instance usage fees applied to instance usage covered by the Reserved Instance bucket.
- **Total Cost**—The usage cost plus the amortized upfront fee for the usage period associated with the Reserved Instance bucket.

Note

If the bucket contains a Reserved Instance that you sold in the Reserved Instance Marketplace and that Reserved Instance was active at any point during the period of the report, the total cost of the bucket might be inflated and your savings might be underestimated.

- **Savings**—The difference between what your usage for the period would have cost at On-Demand prices and what it actually cost using Reserved Instances (Total Cost).
- **Average Utilization**—The average hourly utilization rate for the Reserved Instance bucket over the period.
- **Maximum Utilization**—The highest utilization rate of any hour during the period covered by the report.

For each row—or Reserved Instance bucket—in the table, the graph represents data based on your selected **Show** metric over the selected **Time range** for the report. Each point in the graph represents a metric at a point in time. For information about report options, see [Options Reference \(p. 880\)](#).

A color band at the edge of each selected row in the table corresponds to a report line in the graph. You can show a row in the graph by selecting the checkbox at the beginning of the row.

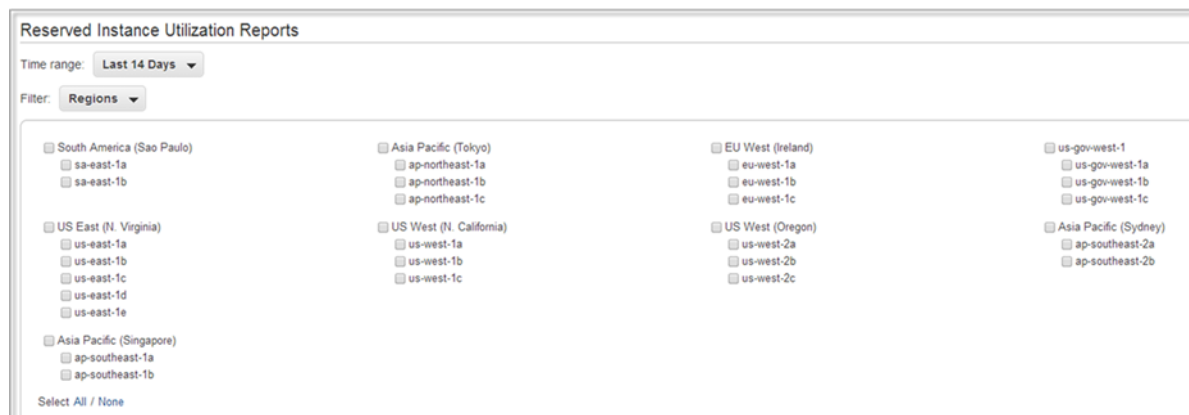
By default, the Reserved Instance utilization report returns data over the last 14 days for all Reserved Instance buckets. The graph shows the average utilization for the first five buckets in the table. You can customize the report graph to show different utilization (average utilization, maximum utilization) or cost (total cost, usage cost) data over a period ranging from 7 days to weeks, months, or years.

Customizing the Report

You can customize the Reserved Instance utilization report with **Time range** and **Filter** options.

Time range provides a list of common relative time ranges, ranging from **Last 7 Days** to **Last 3 Years**. Select the time range that works best for your needs, and then click **Update Report** to apply the change. To apply a time range that is not on the list, select **Custom** and enter the start date and end date for which you want to run the report.

Filter lets you scope your Reserved Instance utilization report by one or more of the following Reserved Instance qualities: region, instance type, accounts, platforms, tenancy, and offering types. For example, you can filter by region or by specific Availability Zones in a region, or both. To filter by region, select **Regions**, then select the regions and Availability Zones you want to include in the report, and choose **Update Report**.



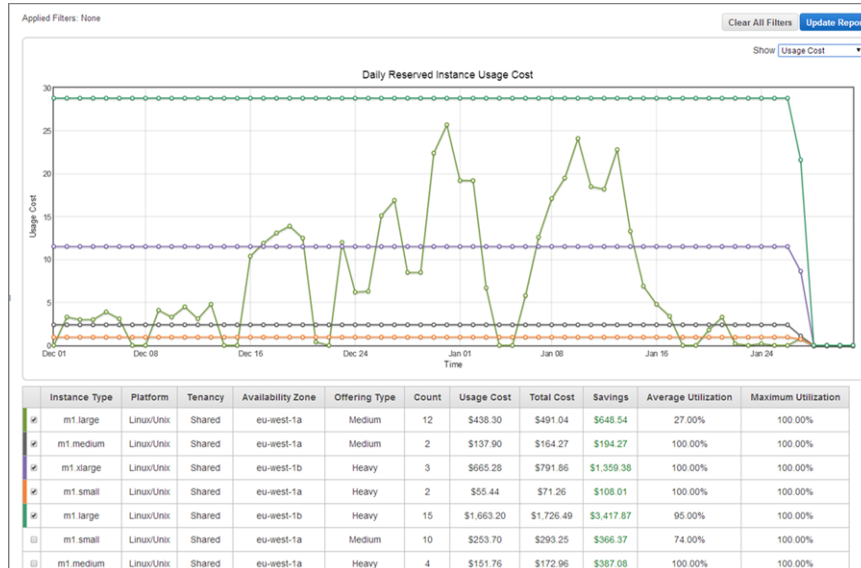
The report will return all results if no filter is applied.

For information about report options, see [Options Reference \(p. 880\)](#).

Viewing Your Reserved Instance Utilization

In this section, we will highlight aspects of your Reserved Instance utilization that the graph and table capture. For the purposes of this discussion, we'll use the following report, which is based on test data.

Amazon Elastic Compute Cloud User Guide for Windows Instances Reserved Instance Utilization



This Reserved Instance utilization report displays the average utilization of Reserved Instances in the last two months. This report reveals the following information about the account's Reserved Instances and how they have been utilized.

- Average Utilization

Most of the Reserved Instances in the table were utilized well. Standouts were the two m1.medium medium utilization Reserved Instances (row 2), which were utilized all the time at 100% average utilization, and the m1.xlarge (row 3) and m1.small (row 4) heavy utilization Reserved Instances, which also were utilized all the time. In contrast, the high-count heavy utilization Reserved Instances (row 5) had lower average utilization rates.

It is also worth noting that the 12 m1.large medium utilization Reserved Instances (row 1) were utilized on average only 27 percent of the time.

- Maximum Utilization

At some point during the two-month period, all of the Reserved Instances were used 100 percent.

- Savings

All across the board, the report shows that for this test account, using Reserved Instances instead of On-Demand instances results in savings for the account owner.

- Question

Does the account have too many m1.large medium utilization Reserved Instances (row 1)?

Bookmarking a Customized Report

You might want to generate a customized report again. Do this by bookmarking the report.

To bookmark a custom report

1. Select the options and filters for your report. Each selection you make adds a parameter to the console URL. For example, `granularity=Hourly` and `Filters=filter_list`.
2. Using your browser, add the console URL as a bookmark.
3. To generate the same report in the future, use the bookmark that you created.

Exporting Your Usage Data

You might want to include your report graph or table in other reports. Do this by exporting the data.

To export usage data

1. Select the options and filters for your report.
2. To export the usage data from the table as a `.csv` file, choose **Download** and select **CSV Only**.
3. To export the graphical usage data as a `.png` file, choose **Download** and select **Graph Only**.

Options Reference

Use the **Show** options to specify the metric to be displayed by the report graph.

- Average Utilization

Shows the average of the utilization rates for each hour over the selected time range, where the utilization rate of a bucket for an hour is the number of instance hours used for that hour divided by the total number of Reserved Instances owned in that hour.

- Maximum Utilization

Shows the highest of the utilization rates of any hour over the selected time range, where the utilization rate of a bucket for an hour is the number of instance hours used for that hour divided by the total number of Reserved Instances owned in that hour.

- Total Cost

Shows the usage cost plus the amortized portion of the upfront cost of the Reserved Instances in the bucket over the period for which the report is generated.

- Usage Cost

Shows the total cost based on hourly fees for a selected bucket of Reserved Instances.

Use **Time range** to specify the period on which the report will be based.

Note

All times are specified in UTC time.

- Last 7 Days

Shows data for usage that took place during the current and previous six calendar days. Can be used with daily or monthly granularities.

- Last 14 Days

Shows data for usage that took place during the current and previous 13 calendar days. Can be used with daily or monthly granularities.

- This Month

Shows data for usage that took place during the current calendar month. Can be used with daily or monthly granularities.

- Last 3 Months

Shows data for usage that took place during the current and previous two calendar months. Can be used with daily or monthly granularities.

- Last 6 Months

Shows data for usage that took place during the current and previous five calendar months. Can be used with monthly granularities.

- Last 12 Months

Shows data for usage that took place during the current and previous 11 calendar months. Can be used with monthly granularity.

- This Year

Shows data for usage that took place during the current calendar year. Can be used with monthly granularity.

- Last 3 Years

Shows data for usage that took place during the current and previous two calendar years. Can be used with monthly granularity.

- Custom

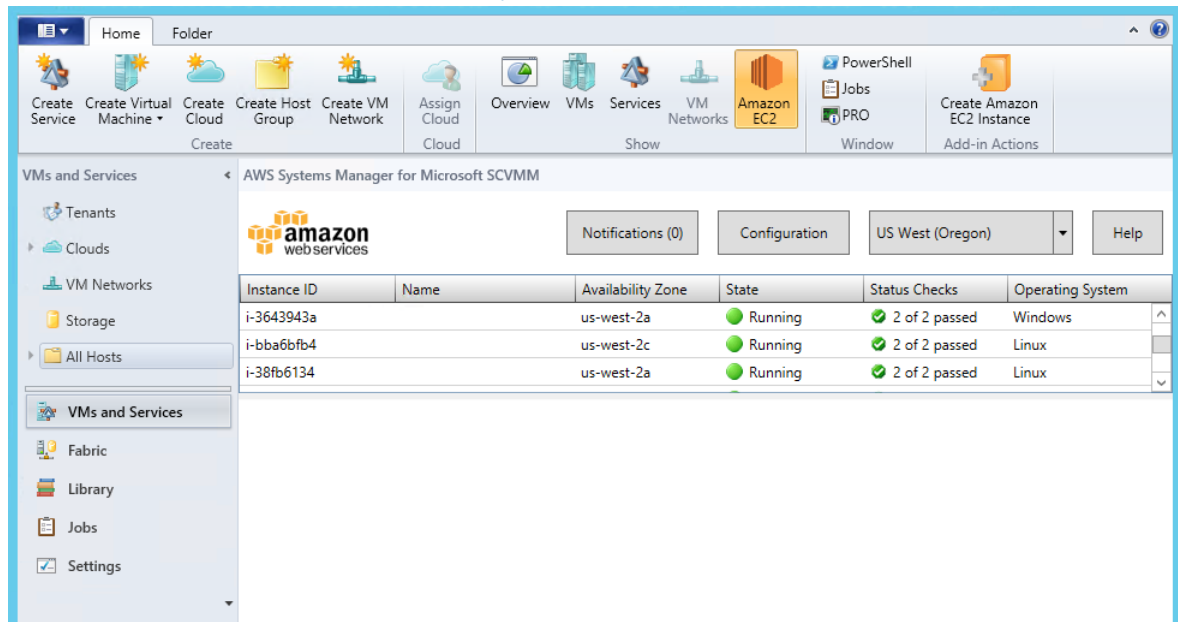
Shows data for the time range for the entered **Start** and **End** dates specified in the following format: mm/dd/yyyy. Can be used with hourly, daily, or monthly granularities, but you can only specify a maximum time range of two days for hourly data, two months for daily data, and three years for monthly data.

Use **Filter** to scope the data displayed in the report.

- Regions
- Instance Type
- Accounts
- Platforms
- Tenancy
- Offering Types

AWS Systems Manager for Microsoft System Center VMM

Amazon Web Services (AWS) Systems Manager for Microsoft System Center Virtual Machine Manager (SCVMM) provides a simple, easy-to-use interface for managing AWS resources, such as EC2 instances, from Microsoft SCVMM. It is implemented as an add-in for the VMM console. For more information, see [AWS Add-ins for Microsoft System Center](#).



Features

- Administrators can grant permissions to users so that they can manage EC2 instances from SCVMM.
- Users can launch, view, reboot, stop, start, and terminate instances, if they have the required permissions.
- Users can get the passwords for their Windows instances and connect to them using RDP.

- Users can get the public DNS names for their Linux instances and connect to them using SSH.
- Users can import their Hyper-V Windows virtual machines from SCVMM to Amazon EC2.

Limitations

- Users must have an account that they can use to log in to SCVMM.
- You can't launch EC2 instances into EC2-Classic; you must launch them into a VPC.
- You can't import Linux virtual machines from SCVMM to Amazon EC2.
- This is not a comprehensive tool for creating and managing AWS resources. The add-in enables SCVMM users to get started quickly with the basic tasks for managing their EC2 instances. Future releases might support managing additional AWS resources.

Requirements

- An AWS account
- Microsoft System Center VMM 2012 R2 or System Center VMM 2012 SP1 with the latest update roll-up

Getting Started

To get started, see the following documentation:

- [Setting Up](#) (p. 883)
- [Managing EC2 Instances](#) (p. 887)
- [Troubleshooting](#) (p. 894)

Setting Up AWS Systems Manager for Microsoft SCVMM

When you set up AWS Systems Manager, users in your organization can access your AWS resources. The process involves creating accounts, deploying the add-in, and providing your credentials.

Tasks

- [Sign Up for AWS](#) (p. 883)
- [Set Up Access for Users](#) (p. 884)
- [Deploy the Add-In](#) (p. 886)
- [Provide Your AWS Credentials](#) (p. 886)

Sign Up for AWS

When you sign up for Amazon Web Services, your AWS account is automatically signed up for all services in AWS. You are charged only for the services that you use.

If you have an AWS account already, skip to the next task. If you don't have an AWS account, use the following procedure to create one.

To sign up for an AWS account

1. Open <http://aws.amazon.com/>, and then choose **Create an AWS Account**.
2. Follow the online instructions.

Part of the sign-up procedure involves receiving a phone call and entering a PIN using the phone keypad.

Set Up Access for Users

The first time that you use AWS Systems Manager, you must provide AWS credentials. To enable multiple users to access the same AWS account using unique credentials and permissions, create an IAM user for each user. You can create one or more groups with policies that grant permissions to perform limited tasks. Then you can create one or more IAM users, and add each user to the appropriate group.

To create an Administrators group

1. Open the Identity and Access Management (IAM) console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Groups** and then choose **Create New Group**.
3. In the **Group Name** box, specify **Administrators** and then choose **Next Step**.
4. On the **Attach Policy** page, select the **AdministratorAccess** AWS managed policy.
5. Choose **Next Step** and then choose **Create Group**.

To create a group with limited access to Amazon EC2

1. Open the Identity and Access Management (IAM) console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Groups** and then choose **Create New Group**.
3. In the **Group Name** box, specify a meaningful name for the group and then choose **Next Step**.
4. On the **Attach Policy** page, do not select an AWS managed policy — choose **Next Step**, and then choose **Create Group**.
5. Choose the name of the group you've just created. On the **Permissions** tab, choose **Inline Policies**, and then **click here**.
6. Select the **Custom Policy** radio button and then choose **Select**.
7. Enter a name for the policy and a policy document that grants limited access to Amazon EC2, and then choose **Apply Policy**. For example, you can specify one of the following custom policies.

Grant users in this group permission to view information about EC2 instances only

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:Describe*",
        "iam:ListInstanceProfiles"
      ],
      "Resource": "*"
    }
  ]
}
```

```
]
}
```

Grant users in this group permission to perform all operations on EC2 instances that are supported by the add-in

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:ListInstanceProfiles", "iam:PassRole",
        "ec2:Describe*", "ec2:CreateKeyPair",
        "ec2:CreateTags", "ec2>DeleteTags",
        "ec2:RunInstances", "ec2:GetPasswordData",
        "ec2:RebootInstances", "ec2:StartInstances",
        "ec2:StopInstances", "ec2:TerminateInstances"
      ],
      "Resource": "*"
    }
  ]
}
```

Grant users in this group permission to import a VM to Amazon EC2

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListAllMyBuckets", "s3:CreateBucket",
        "s3>DeleteBucket", "s3>DeleteObject",
        "s3:GetBucketLocation", "s3:GetObject",
        "s3:ListBucket", "s3:PutObject",
        "ec2:DescribeTags", "ec2:CancelConversionTask",
        "ec2:DescribeConversionTasks", "ec2:DescribeInstanceAttribute",
        "ec2:CreateImage", "ec2:AttachVolume",
        "ec2:ImportInstance", "ec2:ImportVolume",
        "dynamodb:DescribeTable", "dynamodb:CreateTable",
        "dynamodb:Scan", "dynamodb:PutItem", "dynamodb:UpdateItem"
      ],
      "Resource": "*"
    }
  ]
}
```

To create an IAM user, get the user's AWS credentials, and grant the user permissions

1. In the navigation pane, choose **Users** and then choose **Add user**.
2. Enter a user name.
3. Select the type of access this set of users will have. Select **Programmatic access** and **AWS Management Console access** if this user must also access the AWS Management Console.
4. For **Console password type**, choose one of the following:

- **Autogenerated password.** Each user gets a randomly generated password that meets the current password policy in effect (if any). You can view or download the passwords when you get to the **Final** page.
 - **Custom password.** Each user is assigned the password that you type in the box.
5. Choose **Next: Permissions**.
 6. On the **Set permissions** page, choose **Add user to group**. Select the appropriate group.
 7. Choose **Next: Review**, then **Create user**.
 8. To view the users' access keys (access key IDs and secret access keys), choose **Show** next to each password and secret access key that you want to see. To save the access keys, choose **Download .csv** and then save the file to a safe location.

Note

You cannot retrieve the secret access key after you complete this step; if you misplace it you must create a new one.

9. Choose **Close**.

Deploy the Add-In

Add-ins for System Center VMM are distributed as `.zip` files. To deploy the add-in, use the following procedure.

To deploy the add-in

1. From your instance, go to [AWS Systems Manager for Microsoft System Center Virtual Machine Manager](#) and click **SCVMM**. Save the `aws-systems-manager-1.5.zip` file to your instance.
2. Open the VMM console.
3. In the navigation pane, click **Settings** and then click **Console Add-Ins**.
4. On the ribbon, click **Import Console Add-in**.
5. On the **Select an Add-in** page, click **Browse** and select the `aws-systems-manager-1.5.zip` file for the add-in that you downloaded.
6. Ignore any warnings that there are assemblies in the add-in that are not signed by a trusted authority. Select **Continue installing this add-in anyway** and then click **Next**.
7. On the **Summary** page, click **Finish**.
8. When the add-in is imported, the status of the job is `Completed`. You can close the **Jobs** window.

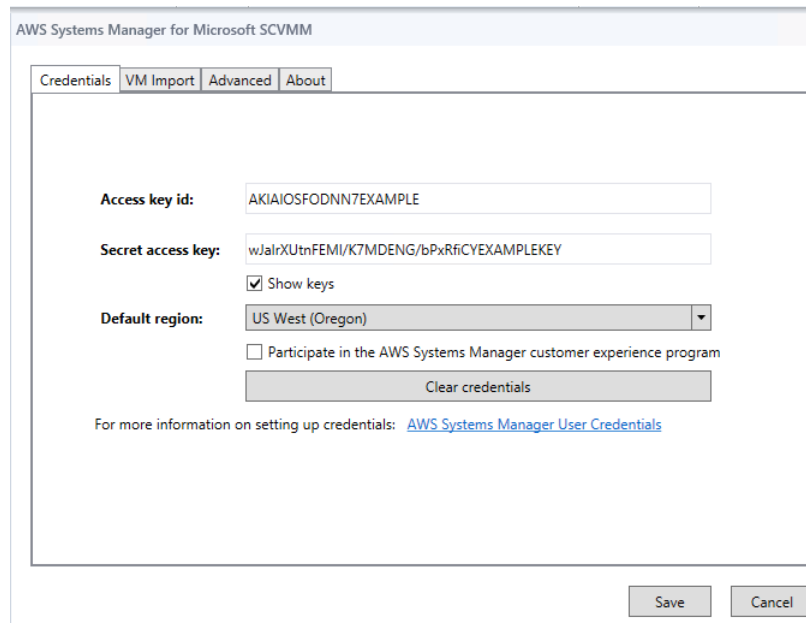
Provide Your AWS Credentials

When you use the AWS Systems Manager for the first time, you must provide your AWS credentials. Your access keys identify you to AWS. There are two types of access keys: access key IDs (for example, AKIAIOSFODNN7EXAMPLE) and secret access keys (for example, wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY). You should have stored your access keys in a safe place when you received them.

To provide your AWS credentials

1. Open the VMM console.
2. In the navigation pane, click **VMs and Services**.
3. On the ribbon, click **Amazon EC2**.

4. On the **Credentials** tab, specify your AWS credentials, select a default region, and then click **Save**.



The screenshot shows the 'Credentials' tab in the AWS Systems Manager for Microsoft SCVMM console. The interface includes a title bar 'AWS Systems Manager for Microsoft SCVMM' and a navigation menu with 'Credentials', 'VM Import', 'Advanced', and 'About'. The main content area contains the following fields and controls:

- Access key id:** A text input field containing 'AKIAIOSFODNN7EXAMPLE'.
- Secret access key:** A text input field containing 'wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY'.
- Show keys:** A checked checkbox.
- Default region:** A dropdown menu currently set to 'US West (Oregon)'.
- Participate in the AWS Systems Manager customer experience program:** An unchecked checkbox.
- Clear credentials:** A button to reset the credential fields.
- Help link:** A link to 'AWS Systems Manager User Credentials'.

At the bottom right of the form are 'Save' and 'Cancel' buttons.

To change these credentials at any time, click **Configuration**.

Managing EC2 Instances Using AWS Systems Manager for Microsoft SCVMM

After you log in to the AWS Systems Manager using your AWS credentials, you can manage your EC2 instances.

Tasks

- [Creating an EC2 Instance \(p. 887\)](#)
- [Viewing Your Instances \(p. 889\)](#)
- [Connecting to Your Instance \(p. 890\)](#)
- [Rebooting Your Instance \(p. 890\)](#)
- [Stopping Your Instance \(p. 891\)](#)
- [Starting Your Instance \(p. 891\)](#)
- [Terminating Your Instance \(p. 891\)](#)

Creating an EC2 Instance

The permissions that you've been granted by your administrator determine whether you can create instances.

Prerequisites

- A virtual private cloud (VPC) with a subnet in the Availability Zone where you'll launch the instance. For more information about creating a VPC, see the [Amazon VPC Getting Started Guide](#).

To create an EC2 instance

1. Open SCVMM.
2. On the ribbon, click **Create Amazon EC2 Instance**.
3. Complete the **Create Amazon EC2 Instance** dialog box as follows:
 - a. Select a region for your instance. By default, we select the region that you configured as your default region.
 - b. Select a template (known as an AMI) for your instance. To use an AMI provided by Amazon, select **Windows** or **Linux** and then select an AMI from **Image**. To use an AMI that you created, select **My images** and then select the AMI from **Image**.
 - c. Select an instance type for the instance. First, select one of the latest instance families from **Family**, and then select an instance type from **Instance type**. To include previous generation instance families in the list, select **Show previous generations**. For more information, see [Amazon EC2 Instances](#) and [Previous Generation Instances](#).
 - d. Create or select a key pair. To create a key pair, select `Create a new key pair` from **Key pair name** and enter a name for the key pair in the highlighted field (for example, `my-key-pair`).
 - e. (Optional) Under **Advanced settings**, specify a display name for the instance.
 - f. (Optional) Under **Advanced settings**, select a VPC from **Network (VPC)**. Note that this list includes all VPCs for the region, including VPCs created using the Amazon VPC console and the default VPC (if it exists). If you have a default VPC in this region, we select it by default. If the text is "There is no VPC available for launch or import operations in this region", then you must create a VPC in this region using the Amazon VPC console.
 - g. (Optional) Under **Advanced settings**, select a subnet from **Subnet**. Note that this list includes all subnets for the selected VPC, including any default subnets. If this list is empty, you must add a subnet to the VPC using the Amazon VPC console, or select a different VPC. Otherwise, we select a subnet for you.
 - h. (Optional) Under **Advanced settings**, create a security group or select one or more security groups. If you select `Create default security group`, we create a security group that grants RDP and SSH access to everyone, which you can modify using the Amazon EC2 or Amazon VPC console. You can enter a name for this security group in the **Group name** box.
 - i. (Optional) Under **Advanced settings**, select an IAM role. If this list is empty, you can create a role using the IAM console.

The screenshot shows the 'Create Amazon EC2 Instance' wizard. The 'Key pair name' field is highlighted with a red border and contains the text 'my-key-pair'. The 'Advanced settings' section is expanded, showing details for the root volume, network, and security groups.

4. Click **Create**. If you are creating a key pair, you are prompted to save the `.pem` file. Save this file in a secure place; you'll need it to log in to your instance. You'll receive confirmation that the instance has launched. Click **Close**.

After you've created your instance, it appears in the list of instances for the region in which you launched it. Initially, the status of the instance is `pending`. After the status changes to `running`, your instance is ready for use.

You can manage the lifecycle of your instance using AWS Systems Manager, as described on this page. To perform other tasks, such as the following, you must use the AWS Management Console:

- [Attach an Amazon EBS volume to your instance \(p. 766\)](#)
- [Associate an Elastic IP address with your instance \(p. 713\)](#)
- [Enable termination protection \(p. 266\)](#)

Viewing Your Instances

The permissions that your administrator grants you determine whether you can view instances and get detailed information about them.

To view your instances and get detailed information

1. Open AWS Systems Manager.
2. From the region list, select a region.
3. From the list of instances, select one or more instances.

4. In the lower pane, click the down arrow next to each instance to view detailed information about the instance.

^ i-343e9f3a (my-instance)

Virtual machine information		Networking	
Instance ID:	i-343e9f3a	Public DNS name:	
Name:	my-instance	Public IP address:	
State:	Running	Private DNS name:	ip-10-0-0-147.us-west-2.compute.internal
Launch time:	1/20/2015 12:26:48 PM -08:00 (1 minute ago)	Private IP address:	10.0.0.147
Instance type:	m3.medium	Vpc ID:	vpc-f1663d98
Tenancy:	default	Subnet ID:	subnet-c9663da0
Image ID:	ami-29d18719	Network interfaces:	eni-89b0bed0
Operating system:	Windows		

Connecting to Your Instance

You can log in to an EC2 instance if you have the private key (.pem file) for the key pair that was specified when launching the instance. The tool that you'll use to connect to your instance depends on whether the instance is a Windows instance or a Linux instance.

To connect to a Windows EC2 instance

1. Open AWS Systems Manager.
2. From the list of instances, select the instance, right-click, and then click **Retrieve Windows Password**.
3. In the **Retrieve Default Windows Administrator Password** dialog box, click **Browse**. Select the private key file for the key pair and then click **Open**.
4. Click **Decrypt Password**. Save the password or copy it to the clipboard.
5. Select the instance, right-click, and then click **Connect via RDP**. When prompted for credentials, use the name of the administrator account and the password that you saved in the previous step.
6. Because the certificate is self-signed, you might get a warning that the security certificate is not from a trusted certifying authority. Click **Yes** to continue.

If the connection fails, see [Troubleshooting Windows Instances](#) in the *Amazon EC2 User Guide for Windows Instances*.

To connect to a Linux EC2 instance

1. Open AWS Systems Manager.
2. From the list of instances, select the instance.
3. In the lower pane, click the down arrow next to the instance ID to view detailed information about the instance.
4. Locate the public DNS name. You'll need this information to connect to your instance.
5. Connect to the instance using PuTTY. For step-by-step instructions, see [Connect to Your Linux Instance from Windows Using PuTTY](#) in the *Amazon EC2 User Guide for Linux Instances*.

Rebooting Your Instance

The permissions that you've been granted by your administrator determine whether you can reboot instances.

To reboot your instance

1. Open AWS Systems Manager.

2. From the list of instances, select the instance.
3. Right-click the instance, and then click **Reset (Reboot)**.
4. When prompted for confirmation, click **Yes**.

Stopping Your Instance

The permissions that you've been granted by your administrator determine whether you can stop instances.

To stop your instance

1. Open AWS Systems Manager.
2. From the list of instances, select the instance.
3. Right-click the instance, and then click **Shut Down (Stop)**.
4. When prompted for confirmation, click **Yes**.

Starting Your Instance

The permissions that you've been granted by your administrator determine whether you can start instances.

To start your instance

1. Open AWS Systems Manager.
2. From the list of instances, select the instance.
3. Right-click the instance, and then click **Power On (Start)**.
4. When prompted for confirmation, click **Yes**.

If you get a quota error when you try to start an instance, you have reached your concurrent running instance limit. The default limit for your AWS account is 20. If you need additional running instances, complete the form at [Request to Increase Amazon EC2 Instance Limit](#).

Terminating Your Instance

The permissions that you've been granted by your administrator determine whether you can terminate instances.

To terminate your instance

1. Open AWS Systems Manager.
2. From the list of instances, select the instance.
3. Right-click the instance, and then click **Delete (Terminate)**.
4. When prompted for confirmation, click **Yes**.

Importing Your Virtual Machine Using AWS Systems Manager for Microsoft SCVMM

You can launch an EC2 instance from a virtual machine that you import from SCVMM to Amazon EC2.

Important

You can't import Linux virtual machines from SCVMM to Amazon EC2.

Contents

- [Prerequisites](#) (p. 892)
- [Importing Your Virtual Machine](#) (p. 892)
- [Checking the Import Task Status](#) (p. 893)
- [Backing Up Your Imported Instance](#) (p. 893)

Prerequisites

- Ensure that your VM is ready. For more information, see [Prepare Your VM](#) in the *VM Import/Export User Guide*.
- In AWS Systems Manager, click **Configuration**, select the **VM Import** tab, and review the following settings:
 - **S3 bucket prefix**: We create a bucket for disk images to be uploaded before they are imported. The name of the bucket starts with the prefix listed here and includes the region (for example, `us-west-2`). To delete the disk images after they are imported, select **Clean up S3 bucket after import**.
 - **VM image export path**: A location for the disk images exported from the VM. To delete the disk images after they are imported, select **Clean up export path after import**.
 - **Alternate Hyper-V PowerShell module path**: The location of the Hyper-V PowerShell module, if it's not installed in the standard location. For more information, see [Installing the Hyper-V Management Tools](#) in the Microsoft TechNet Library.

Importing Your Virtual Machine

The permissions that you've been granted by your administrator determine whether you can import HyperV Windows virtual machines from SCVMM to AWS.

To import your virtual machine

1. Open SCVMM.
2. On the ribbon, click **VMs**. Select your virtual machine from the list.
3. On the ribbon, click **Import VM to Amazon EC2**.
4. Complete the **Import Virtual Machine** dialog box as follows:
 - a. Select a region for the instance. By default, we select the region that you configured as your default region.
 - b. Select an instance type for the instance. First, select one of the latest instance families from **Family**, and then select an instance type from **Instance type**. To include previous generation instance families in the list, select **Show previous generations**. For more information, see [Amazon EC2 Instances](#) and [Previous Generation Instances](#).
 - c. Select a VPC from **Network (VPC)**. Note that this list includes all VPCs for the region, including VPCs created using the Amazon VPC console and the default VPC (if it exists). If you have a default VPC in this region, we select it by default. If the text is "There is no VPC available for launch or import operations in this region", then you must create a VPC in this region using the Amazon VPC console.
 - d. Select a subnet from **Subnet**. Note that this list includes all subnets for the selected VPC, including any default subnets. If this list is empty, you must add a subnet to the VPC using the Amazon VPC console, or select a different VPC. Otherwise, we select a subnet for you.

The screenshot shows the 'Import Virtual Machine' dialog box. It has a title bar with the text 'Import Virtual Machine' and a close button. Below the title bar is the Amazon Web Services logo and the text 'To import a virtual machine into Amazon EC2, complete the fields, and then click Import.' The dialog is divided into two main sections: 'Virtual Machine' and 'Amazon EC2 Options'. The 'Virtual Machine' section contains the following fields: Name (my-virtual-machine), Id (729D71D0-E0FE-414F-8C78-AE3EB549CBC6), Host (my-host), and Hardware (Processors: 1, Memory: 512 MB). The 'Amazon EC2 Options' section contains the following fields: Region (US West (Oregon)), Architecture (64-bit selected, 32-bit unselected), Family (General purpose), Instance type (m3.xlarge), Network (VPC: vpc-f1663d98), and Subnet (subnet-cb663da2). At the bottom of the dialog are two buttons: 'Import' and 'Cancel'.

5. Click **Import**. If you haven't specified the required information in the **VM Import** tab, you'll receive an error asking you to provide the required information. Otherwise, you'll receive confirmation that the import task has started. Click **Close**.

Checking the Import Task Status

The import task can take several hours to complete. To view the current status, open AWS System Manager and click **Notifications**.

You'll receive the following notifications as the import task progresses:

- Import VM: Created Import VM Task
- Import VM: Export VM Disk Image Done
- Import VM: Upload to S3
- Import VM: Image Conversion Starting
- Import VM: Image Conversion Done
- Import VM: Import Complete

Note that you'll receive the **Import VM: Upload to S3**, **Import VM: Image Conversion Starting**, and **Import VM: Image Conversion Done** notifications for each disk image converted.

If the import task fails, you'll receive the notification **Import VM: Import Failed**. For more information about troubleshooting issues with import tasks, see [Errors Importing a VM \(p. 895\)](#).

Backing Up Your Imported Instance

After the import operation completes, the instance runs until it is terminated. If your instance is terminated, you can't connect to or recover the instance. To ensure that you can start a new instance with the same software as an imported instance if needed, create an Amazon Machine Image (AMI) from the imported instance. For more information, see [Creating an Amazon EBS-Backed Windows AMI \(p. 77\)](#).

Troubleshooting AWS Systems Manager for Microsoft SCVMM

The following are common errors and troubleshooting steps.

Contents

- [Error: Add-in cannot be installed \(p. 894\)](#)
- [Installation Errors \(p. 894\)](#)
- [Checking the Log File \(p. 895\)](#)
- [Errors Importing a VM \(p. 895\)](#)
- [Uninstalling the Add-In \(p. 895\)](#)

Error: Add-in cannot be installed

If you receive the following error, try installing [KB2918659](#) on the computer running the VMM console. For more information, see [Description of System Center 2012 SP1 Update Rollup 5](#). Note that you don't need to install all the updates listed in this article to address this issue, just KB2918659.

```
Add-in cannot be installed
The assembly "Amazon.Scvm.Addin" referenced to by add-in component "AWS
Systems Manager for
Microsoft SCVMM" could not be found in the add-in package. This could be due
to the following
reasons:
1. The assembly was not included with the add-in package.
2. The AssemblyName attribute for the add-in does not match the name of the
add-in assembly.
3. The assembly file is corrupt and cannot be loaded.
```

Installation Errors

If you receive one of the following errors during installation, it is likely due to an issue with SCVMM:

```
Could not update managed code add-in pipeline due to the following error:
Access to the path 'C:\Program Files\Microsoft System Center 2012\Virtual
Machine Manager
\Bin\AddInPipeline\PipelineSegments.store' is denied.
```

```
Could not update managed code add-in pipeline due to the following error:
The required folder 'C:\Program Files\Microsoft System Center 2012\Virtual
Machine Manager
\Bin\AddInPipeline\HostSideAdapters' does not exist.
```

```
Add-in cannot be installed
The assembly "Microsoft.SystemCenter.VirtualMachineManager.UIAddIns.dll"
referenced by the
add-in assembly "Amazon.Scvm.AddIn" could not be found in the add-in
package. Make sure
that this assembly was included with the add-in package.
```

Try one of the following steps to work around this issue:

- Grant authenticated users permission to read and execute the `C:\Program Files\Microsoft System Center 2012\Virtual Machine Manager\Bin\AddInPipeline` folder. In Windows Explorer, right-click the folder, select **Properties**, and then select the **Security** tab.
- Close the SCVMM console and start it one time as an administrator. From the **Start** menu, locate SCVMM, right-click, and then select **Run as administrator**.

Checking the Log File

If you have a problem using the add-in, check the generated log file, `%APPDATA%\Amazon\SCVMM\ec2addin.log`, for useful information.

Errors Importing a VM

The log file, `%APPDATA%\Amazon\SCVMM\ec2addin.log`, contains detailed information about the status of an import task. The following are common errors that you might see in the log file when you import your VM from SCVMM to Amazon EC2.

Error: Unable to extract Hyper-V VirtualMachine object

Solution: Configure the path to the Hyper-V PowerShell module.

Error: You do not have permission to perform the operation

Solution: Contact your administrator.

Uninstalling the Add-In

If you need to uninstall the add-in, use the following procedure.

To uninstall the add-in

1. Open the VMM console.
2. Select the **Settings** workspace, and then click **Console Add-Ins**.
3. Select **AWS Systems Manager for Microsoft SCVMM**.
4. On the ribbon, click **Remove**.
5. When prompted for confirmation, click **Yes**.

If you reinstall the add-in after uninstalling it and receive the following error, delete the path as suggested by the error message.

```
Error (27301)
There was an error while installing the add-in. Please ensure that the
following path does not
exist and then try the installation again.

C:\Program Files\Microsoft System Center 2012\Virtual Machine Manager\Bin
\AddInPipeline\
AddIns\EC2WINDOWS...
```

AWS Management Pack for Microsoft System Center

Amazon Web Services (AWS) offers a complete set of infrastructure and application services for running almost anything in the cloud—from enterprise applications and big data projects to social games and mobile apps. The AWS Management Pack for Microsoft System Center provides availability and performance monitoring capabilities for your applications running in AWS.

The AWS Management Pack allows Microsoft System Center Operations Manager to access your AWS resources (such as instances and volumes), so that it can collect performance data and monitor your AWS resources. The AWS Management Pack is an extension to System Center Operations Manager. There are two versions of the AWS Management Pack: one for System Center 2012 — Operations Manager and another for System Center Operations Manager 2007 R2.

The AWS Management Pack uses Amazon CloudWatch metrics and alarms to monitor your AWS resources. Amazon CloudWatch metrics appear in Microsoft System Center as performance counters and Amazon CloudWatch alarms appear as alerts.

You can monitor the following resources:

- EC2 instances
- EBS volumes
- ELB load balancers
- Auto Scaling groups and Availability Zones
- Elastic Beanstalk applications
- CloudFormation stacks
- CloudWatch Alarms
- CloudWatch Custom Metrics

Contents

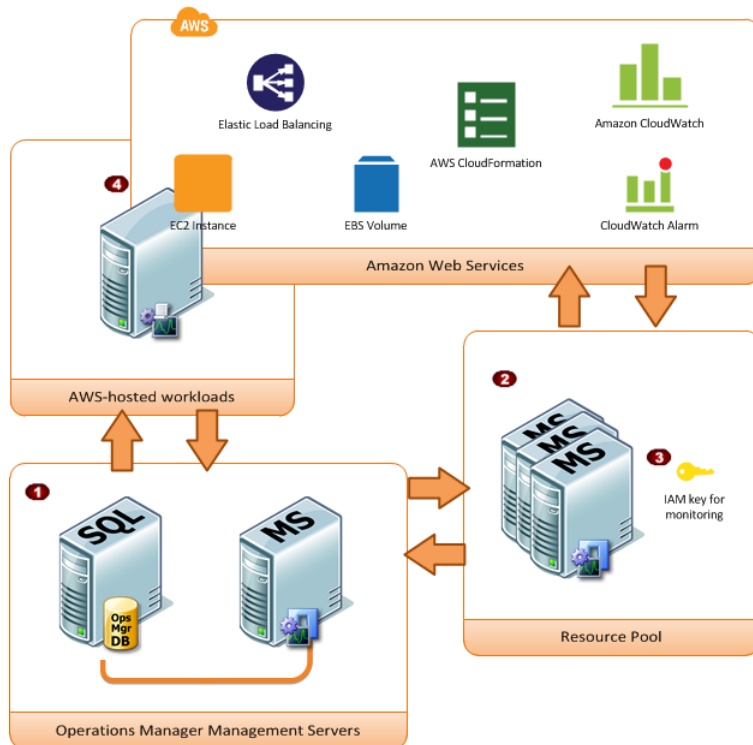
- [Overview of AWS Management Pack for System Center 2012 \(p. 897\)](#)
- [Overview of AWS Management Pack for System Center 2007 R2 \(p. 898\)](#)
- [Downloading the AWS Management Pack \(p. 899\)](#)
- [Deploying the AWS Management Pack \(p. 900\)](#)

- [Using the AWS Management Pack \(p. 910\)](#)
- [Upgrading the AWS Management Pack \(p. 929\)](#)
- [Uninstalling the AWS Management Pack \(p. 930\)](#)
- [Troubleshooting the AWS Management Pack \(p. 931\)](#)

Overview of AWS Management Pack for System Center 2012

The AWS Management Pack for System Center 2012 — Operations Manager uses a resource pool that contains one or more management servers to discover and monitor your AWS resources. You can add management servers to the pool as you increase the number of AWS resources that you use.

The following diagram shows the main components of AWS Management Pack.



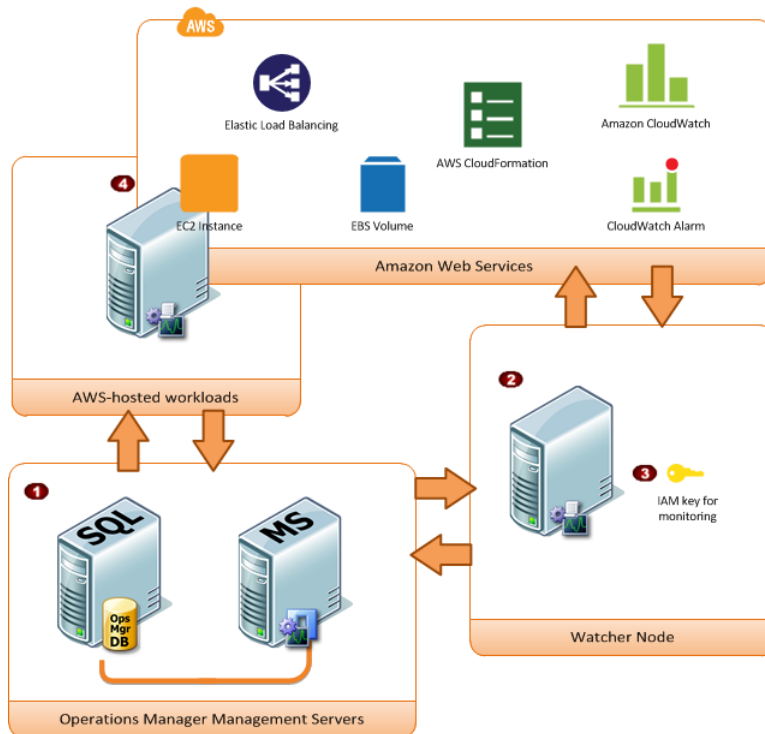
Item	Component	Description
1	Operations Manager infrastructure	One or more management servers and their dependencies, such as Microsoft SQL Server and a Microsoft Active Directory domain. These servers can either be deployed on-premises or in the AWS cloud; both scenarios are supported.
2	Resource pool	One or more management servers used for communicating with AWS using the AWS SDK for .NET. These servers must have Internet connectivity.
3	AWS credentials	An access key ID and a secret access key used by the management servers to make AWS API calls. You must

Item	Component	Description
		specify these credentials while you configure the AWS Management Pack. We recommend that you create an IAM user with read-only privileges and use those credentials. For more information about creating an IAM user, see Adding a New User to Your AWS Account in the <i>IAM User Guide</i> .
4	EC2 instances	Virtual computers running in the AWS cloud. Some instances might have the Operations Manager Agent installed, others might not. When you install Operations Manager Agent you can see the operating system and application health apart from the instance health.

Overview of AWS Management Pack for System Center 2007 R2

The AWS Management Pack for System Center Operations Manager 2007 R2 uses a designated computer that connects to your System Center environment and has Internet access, called a *watcher node*, to call AWS APIs to remotely discover and collect information about your AWS resources.

The following diagram shows the main components of AWS Management Pack.



Item	Component	Description
1	Operations Manager infrastructure	One or more management servers and their dependencies, such as Microsoft SQL Server and a Microsoft Active Directory domain. These servers can either be deployed

Item	Component	Description
		on-premises or in the AWS cloud; both scenarios are supported.
2	Watcher node	A designated agent-managed computer used for communicating with AWS using the AWS SDK for .NET. It can either be deployed on-premises or in the AWS cloud, but it must be an agent-managed computer, and it must have Internet connectivity. You can use exactly one watcher node to monitor an AWS account. However, one watcher node can monitor multiple AWS accounts. For more information about setting up a watcher node, see Deploying Windows Agents in the Microsoft System Center documentation.
3	AWS credentials	An access key ID and a secret access key used by the watcher node to make AWS API calls. You must specify these credentials while you configure the AWS Management Pack. We recommend that you create an IAM user with read-only privileges and use those credentials. For more information about creating an IAM user, see Adding a New User to Your AWS Account in the <i>IAM User Guide</i> .
4	EC2 instances	Virtual computers running in the AWS cloud. Some instances might have the Operations Manager Agent installed, others might not. When you install the Operations Manager Agent you can see the operating system and application health apart from the instance health.

Downloading the AWS Management Pack

To get started, download the AWS Management Pack. The AWS Management Pack is free. You might incur charges for Amazon CloudWatch, depending on how you configure monitoring or how many AWS resources you monitor.

System Center 2012

Before you download the AWS Management Pack, ensure that your systems meet the following system requirements and prerequisites.

System Requirements

- System Center Operations Manager 2012 R2 or System Center Operations Manager 2012 SP1
- Cumulative Update 1 or later. You must deploy the update to the management servers monitoring AWS resources, as well as agents running the watcher nodes and agents to be monitored by the AWS Management Pack. We recommend that you deploy the latest available Operations Manager updates on all computers monitoring AWS resources.
- Microsoft.Unix.Library MP version 7.3.2026.0 or later

Prerequisites

- Your data center must have at least one management server configured with Internet connectivity. The management servers must have the Microsoft .NET Framework version 4.5 or later and PowerShell 2.0 or later installed.

- The action account for the management server must have local administrator privileges on the management server.

To download the AWS Management Pack

1. On the [AWS Add-Ins for Microsoft System Center](#) website, click **SCOM 2012**.
2. Save `AWS-SCOM-MP-2.5.zip` to your computer and unzip it.

Continue with [Deploying the AWS Management Pack \(p. 900\)](#).

System Center 2007 R2

Before you download the AWS Management Pack, ensure that your systems meet the following system requirements and prerequisites.

System Requirements

- System Center Operations Manager 2007 R2
- Microsoft.Unix.Library MP version 6.1.7000.256 or later

Prerequisites

- Your data center must have an agent-managed computer with Internet connectivity that you designate as the watcher node. The watcher node must have the following Agent Proxy option enabled: **Allow this agent to act as a proxy and discover managed objects on other computers**. The watcher node must have the Microsoft .NET Framework version 3.5.1 or later and PowerShell 2.0 or later installed.
- The action account for the watcher node must have local administrator privileges on the watcher node.
- You must ensure that your watcher node has the agent installed, has Internet access, and can communicate with the management servers in your data center. For more information, see [Deploying Windows Agents](#) in the Microsoft System Center documentation.

To download the AWS Management Pack

1. On the [AWS Add-Ins for Microsoft System Center](#) website, click **SCOM 2007**.
2. Save `AWS-MP-Setup-2.5.msi` to your computer.

Continue with [Deploying the AWS Management Pack \(p. 900\)](#).

Deploying the AWS Management Pack

Before you can deploy the AWS Management Pack, you must download it. For more information, see [Downloading the AWS Management Pack \(p. 899\)](#).

Tasks

- [Step 1: Installing the AWS Management Pack \(p. 901\)](#)
- [Step 2: Configuring the Watcher Node \(p. 902\)](#)
- [Step 3: Create an AWS Run As Account \(p. 902\)](#)
- [Step 4: Run the Add Monitoring Wizard \(p. 905\)](#)
- [Step 5: Configure Ports and Endpoints \(p. 909\)](#)

Step 1: Installing the AWS Management Pack

After you download the AWS Management Pack, you must configure it to monitor one or more AWS accounts.

System Center 2012

To install the AWS Management Pack

1. In the Operations console, on the **Go** menu, click **Administration**, and then click **Management Packs**.
2. In the **Actions** pane, click **Import Management Packs**.
3. On the **Select Management Packs** page, click **Add**, and then click **Add from disk**.
4. In the **Select Management Packs to import** dialog box, select the `Amazon.AmazonWebServices.mpb` file from the location where you downloaded it, and then click **Open**.
5. On the **Select Management Packs** page, under **Import list**, select the **Amazon Web Services** management pack, and then click **Install**.

Note

System Center Operations Manager doesn't import any management packs in the **Import** list that display an **Error** icon.

6. The **Import Management Packs** page shows the progress for the import process. If a problem occurs, select the management pack in the list to view the status details. Click **Close**.

System Center 2007 R2

To install the AWS Management Pack

The management pack is distributed as a Microsoft System Installer file, `AWS-MP-Setup.msi`. It contains the required DLLs for the watcher node, root management server, and Operations console, as well as the `Amazon.AmazonWebServices.mp` file.

1. Run `AWS-MP-Setup.msi`.

Note

If your root management server, Operations console, and watcher node are on different computers, you must run the installer on each computer.

2. On the **Welcome to the Amazon Web Services Management Pack Setup Wizard** screen, click **Next**.
3. On the **End-User License Agreement** screen, read the license agreement, and, if you accept the terms, select the **I accept the terms in the License Agreement** check box, and then click **Next**.
4. On the **Custom Setup** screen, select the features you want to install, and then click **Next**.

Operations Console

Installs `Amazon.AmazonWebServices.UI.Pages.dll` and registers it in the Global Assembly Cache (GAC), and then installs `Amazon.AmazonWebServices.mp`.

Root Management Server

Installs `Amazon.AmazonWebServices.Modules.dll`, `Amazon.AmazonWebServices.SCOM.SDK.dll` and the AWS SDK for .NET (`AWSSDK.dll`), and then registers them in the GAC.

AWS Watcher Node

Installs `Amazon.AmazonWebServices.Modules.dll` and `Amazon.AmazonWebServices.SCOM.SDK.dll`, and then installs the AWS SDK for .NET (`AWSSDK.dll`) and registers it in the GAC.

5. On the **Ready to install Amazon Web Services Management Pack** screen, click **Install**.
6. On the **Completed the Amazon Web Services Management Pack Setup Wizard** screen, click **Finish**.

Note

The required DLLs are copied and registered in the GAC, and the management pack file (*.mp) is copied to the `Program Files (x86)/Amazon Web Services Management Pack` folder on the computer running the Operations console. Next, you must import the management pack into System Center.

7. In the Operations console, on the **Go** menu, click **Administration**, and then click **Management Packs**.
8. In the **Actions** pane, click **Import Management Packs**.
9. On the **Select Management Packs** page, click **Add**, and then click **Add from disk**.
10. In the **Select Management Packs to import** dialog box, change the directory to `C:\Program Files (x86)\Amazon Web Services Management Pack`, select the `Amazon.AmazonWebServices.mp` file, and then click **Open**.
11. On the **Select Management Packs** page, under **Import list**, select the **Amazon Web Services** management pack, and then click **Install**.

Note

System Center Operations Manager doesn't import any management packs in the **Import** list that display an **Error** icon.

12. The **Import Management Packs** page shows the progress for the import process. If a problem occurs, select the management pack in the list to view the status details. Click **Close**.

Step 2: Configuring the Watcher Node

On System Center Operations Manager 2007 R2, the watcher node runs discoveries that go beyond the watcher node computer, so you must enable the proxy agent option on the watcher node. The proxy agent allows those discoveries to access the objects on other computers.

Note

If your system is configured with a large number of resources, we recommend that you configure one management server as a Watcher Node. Having a separate Watcher Node management server can improve performance.

If you're using System Center 2012 — Operations Manager, you can skip this step.

To enable the proxy agent on System Center Operations Manager 2007 R2

1. In the Operations console, on the **Go** menu, click **Administration**.
2. In the **Administration** workspace, under **Device Management**, click **Agent Managed**.
3. In the **Agent Managed** list, right-click the watcher node, and then click **Properties**.
4. In the **Agent Properties** dialog box, click the **Security** tab, select **Allow this agent to act as proxy and discover managed objects on other computers**, and then click **OK**.

Step 3: Create an AWS Run As Account

You must set up credentials that grant AWS Management Pack access to your AWS resources.

To create an AWS Run As account

1. We recommend that you create an IAM user with the minimum access rights required (for example, the **ReadOnlyAccess** AWS managed policy works in most cases). You'll need the

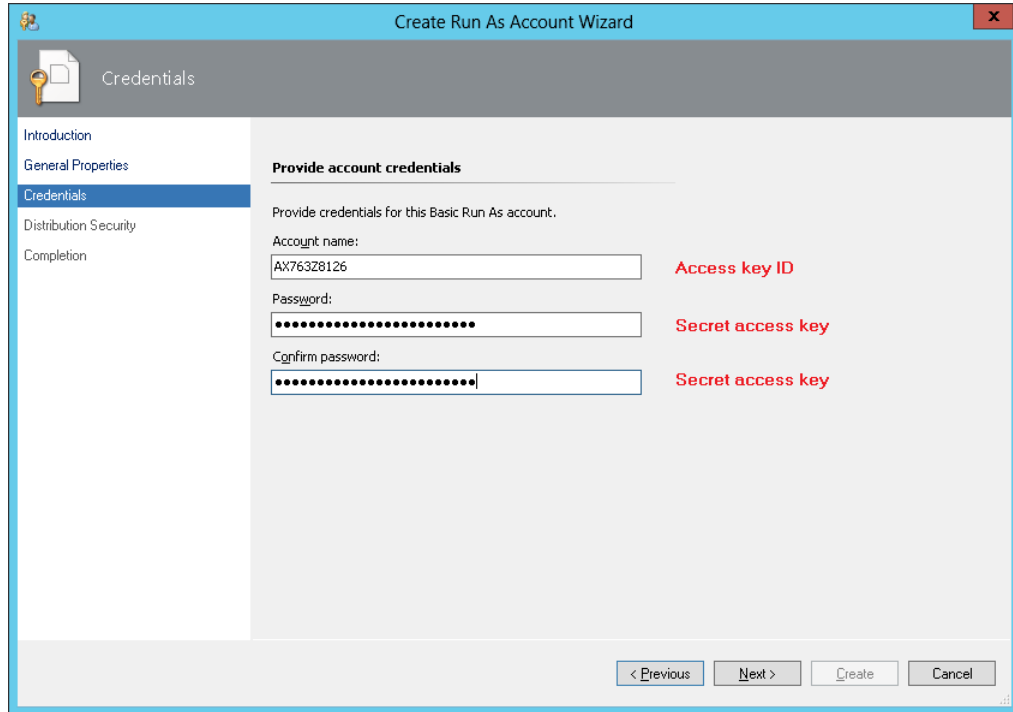
access keys (access key ID and secret access key) for this user to complete this procedure. For more information, see [Administering Access Keys for IAM Users](#) in the *IAM User Guide*.

2. In the Operations console, on the **Go** menu, click **Administration**.
3. In the **Administration** workspace, expand the **Run As Configuration** node, and then select **Accounts**.
4. Right-click the **Accounts** pane, and then click **Create Run As Account**.
5. In the **Create Run As Account Wizard**, on the **General Properties** page, in the **Run As account type** list, select **Basic Authentication**.
6. Enter a display name (for example, "My IAM Account") and a description, and then click **Next**.

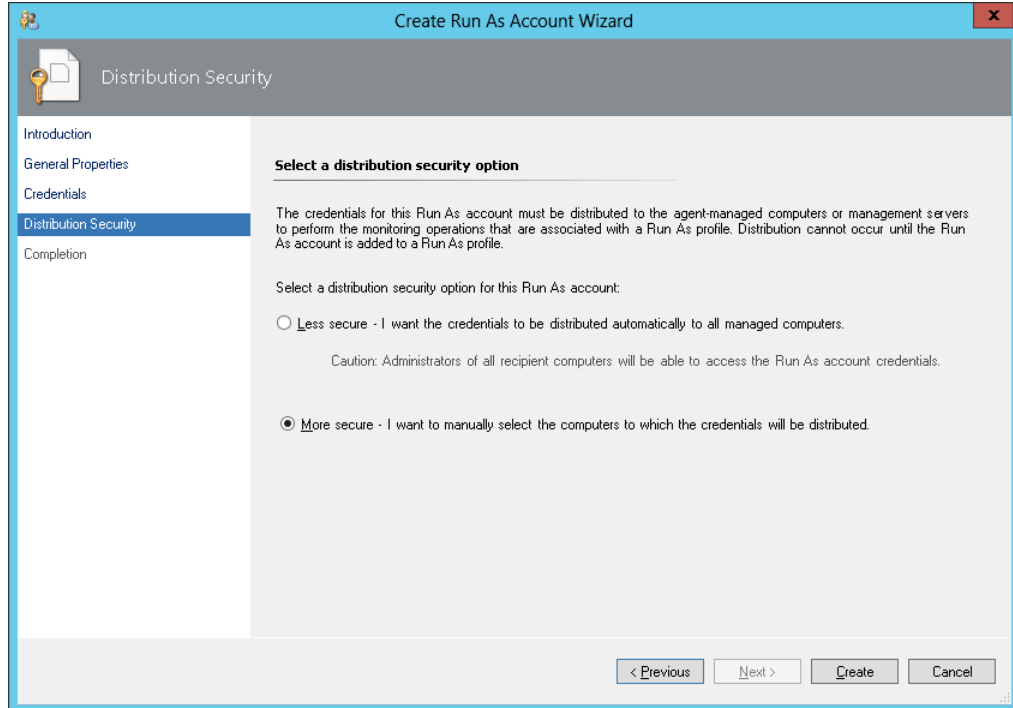
The screenshot shows the 'Create Run As Account Wizard' window, specifically the 'General Properties' page. The left sidebar contains a navigation menu with 'Introduction', 'General Properties' (selected), 'Credentials', 'Distribution Security', and 'Completion'. The main content area is titled 'Specify general properties for the Run As account' and includes the instruction: 'Select the type of Run As account that you want to create, and then provide a display name and description.' Below this, there are three input fields: 'Run As account type' (a dropdown menu with 'Basic Authentication' selected), 'Display name' (a text box containing 'AWS Environment Credentials'), and 'Description (optional)' (a text area). A red rectangular box highlights the 'Run As account type' dropdown and the 'Display name' text box. At the bottom of the wizard, there are four buttons: '< Previous', 'Next >', 'Create', and 'Cancel'.

7. On the **Credentials** page, enter the access key ID in the **Account name** box and the secret access key in the **Password** box, and then click **Next**.

Amazon Elastic Compute Cloud
User Guide for Windows Instances
Step 3: Create an AWS Run As Account

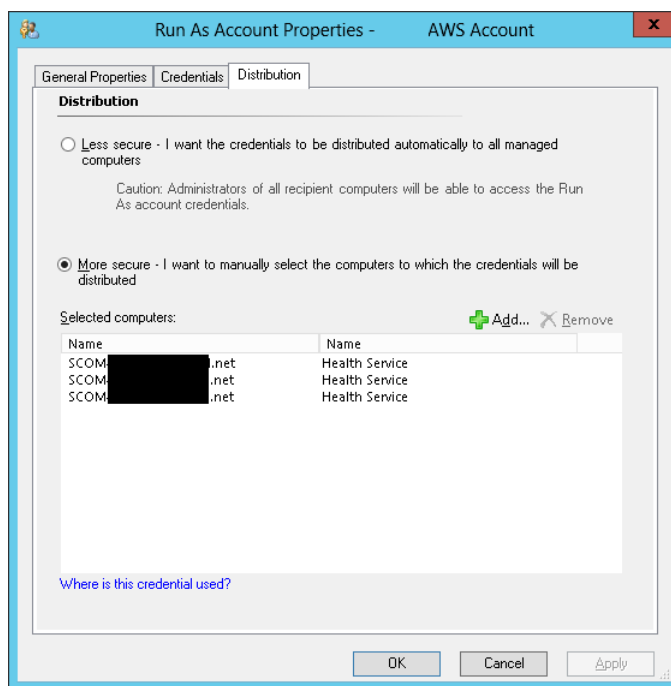


8. On the **Distribution Security** page, select **More secure - I want to manually select the computers to which the credentials will be distributed**, and then click **Create**.



9. Click **Close**.
10. In the list of accounts, select the account that you just created.
11. In the **Actions** pane, click **Properties**.

12. In the **Properties** dialog box, verify that the **More Secure** option is selected and that all management servers to be used to monitor your AWS resources are listed.



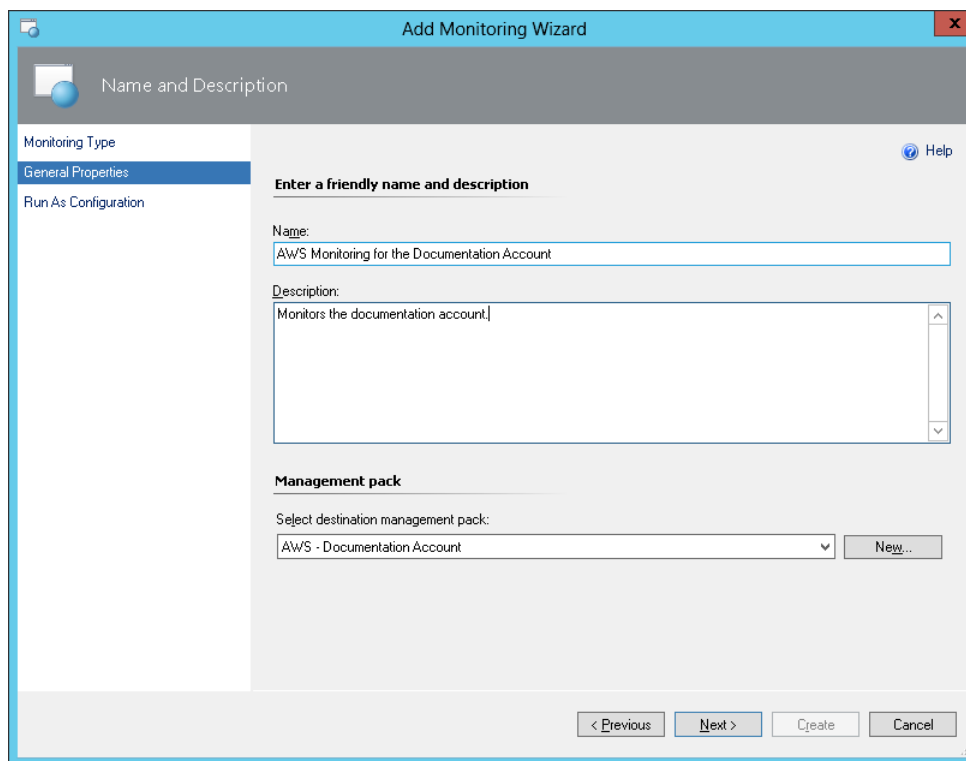
Step 4: Run the Add Monitoring Wizard

You can configure the AWS Management Pack to monitor a particular AWS account by using the Add Monitoring Wizard, which is available in the **Authoring** workspace of the Operations console. This wizard creates a management pack that contains the settings for the AWS account to monitor. You must run this wizard to monitor each AWS account. For example, if you want to monitor two AWS accounts, you must run the wizard twice.

System Center 2012

To run the Add Monitoring Wizard on System Center 2012 — Operations Manager

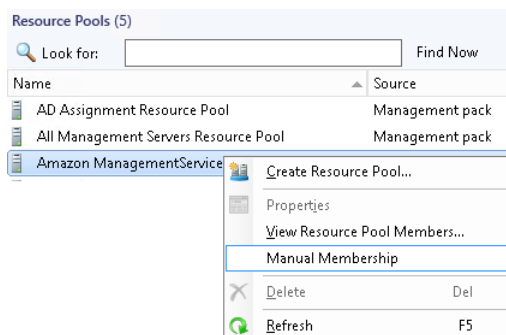
1. In the Operations console, on the **Go** menu, click **Authoring**.
2. In the **Authoring** workspace, expand the **Management Pack Templates** node, right-click **Amazon Web Services**, and then click **Add Monitoring Wizard**.
3. In the **Add Monitoring Wizard**, in the **Select the monitoring type** list, select **Amazon Web Services**, and then click **Next**.
4. On the **General Properties** page, in the **Name** box, enter a name (for example, "My AWS Resources"). In the **Description** box, enter a description.
5. In the **Select destination management pack** list, select an existing management pack (or click **New** to create one) where you want to save the settings. Click **Next**.



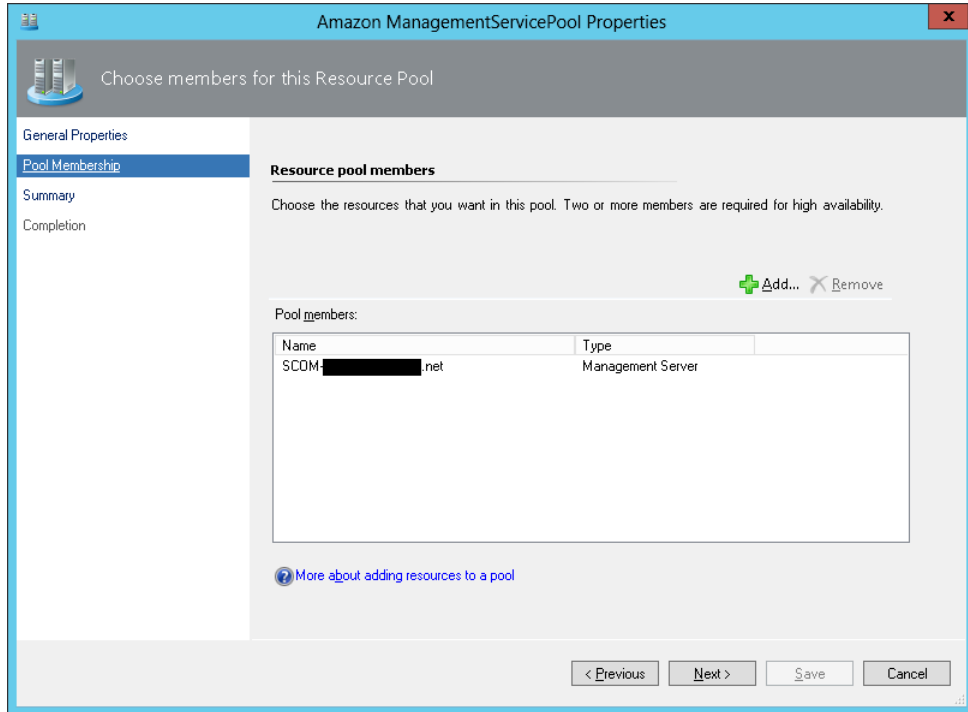
Note

By default, when you create a management pack object, disable a rule or monitor, or create an override, Operations Manager saves the setting to the default management pack. As a best practice, you should create a separate management pack for each sealed management pack that you want to customize, instead of saving your customized settings to the default management pack.

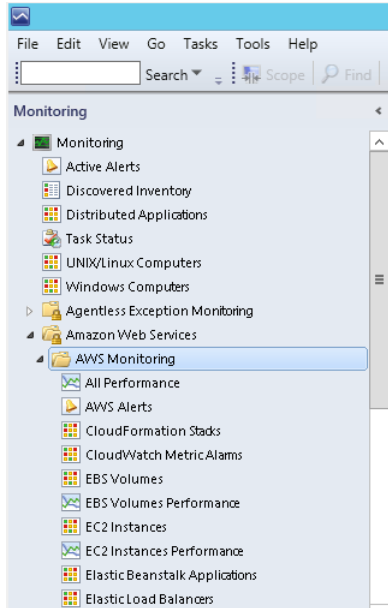
6. The AWS Management Pack automatically creates a resource pool and adds the management servers to it. To control server membership, make the following changes:
 - a. Click **Administration** on the **Go** menu.
 - b. Click the **Resource Pools** node.
 - c. Right-click the **AWS Resource Pool** in the **Resource Pools** pane and select **Manual Membership**.



- d. Right-click the **AWS Resource Pool** in the **Resource Pools** pane and select **Properties**.
 - e. On the **Pool Membership** page, remove the management servers that should not monitor AWS resources.



7. After the AWS Management Pack is configured, it shows up as a sub-folder of the Amazon Web Services folder in the **Monitoring** workspace of the Operations console.

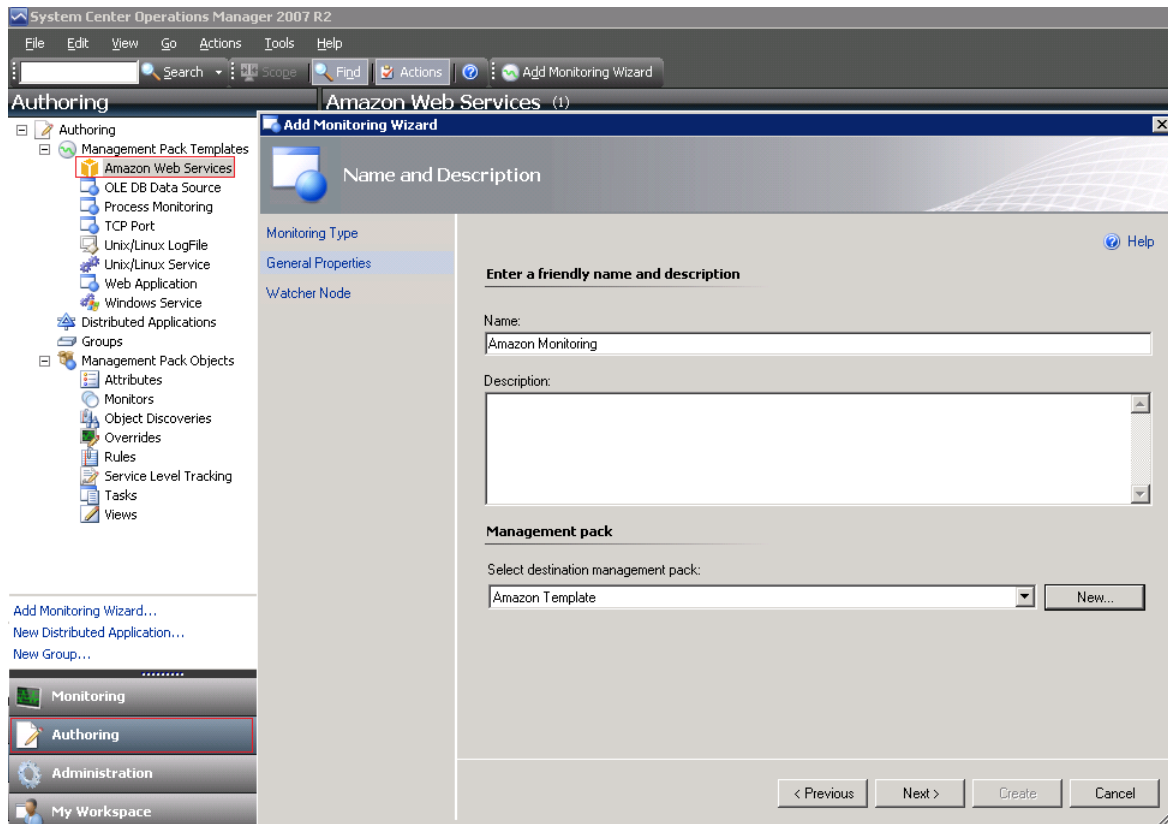


System Center 2007 R2

To run the Add Monitoring Wizard on System Center Operations Manager 2007

1. In the Operations console, on the **Go** menu, click **Authoring**.

2. In the **Authoring** workspace, expand the **Management Pack Templates** node, right-click **Amazon Web Services**, and then click **Add Monitoring Wizard**.
3. In the **Add Monitoring Wizard**, in the **Select the monitoring type list**, select **Amazon Web Services**, and then click **Next**.
4. On the **General Properties** page, in the **Name** box, enter a name (for example, "My AWS Resources"). In the **Description** box, enter a description.
5. In the **Select destination management pack** drop-down list, select an existing management pack (or click **New** to create a new one) where you want to save the settings. Click **Next**.



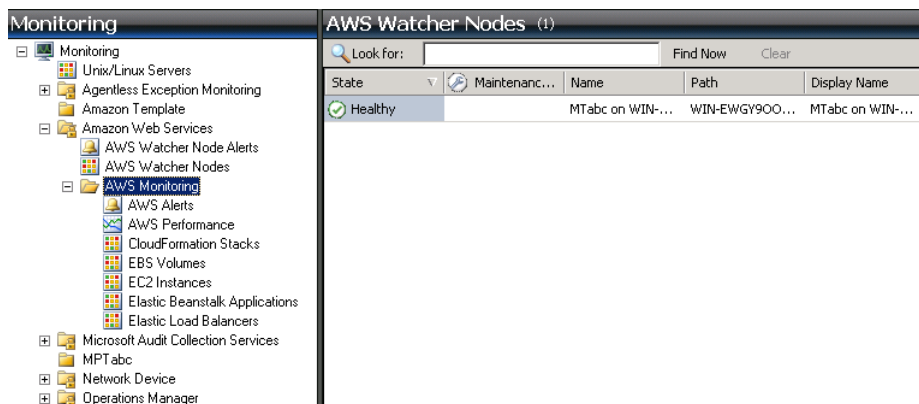
Note

By default, when you create a management pack object, disable a rule or monitor, or create an override, Operations Manager saves the setting to the default management pack. As a best practice, you should create a separate management pack for each sealed management pack that you want to customize, instead of saving your customized settings to the default management pack.

6. On the **Watcher Node Configuration** page, in the **Watcher Node** list, select an agent-managed computer to act as the watcher node.
7. In the **Select AWS Run As account** drop-down list, select the Run As account that you created earlier, and then click **Create**.
8. After the AWS Management Pack is configured, it first discovers the watcher node. To verify that the watcher node was discovered successfully, navigate to the **Monitoring** workspace in the Operations console. You should see a new **Amazon Web Services** folder and an **Amazon Watcher Nodes** subfolder under it. This subfolder displays the watcher nodes. The AWS Management Pack automatically checks and monitors the watcher node connectivity to AWS. When the watcher node is discovered, it shows up in this list. When the watcher node is ready, its state changes to **Healthy**.

Note

To establish connectivity with AWS, the AWS Management Pack requires that you deploy the AWS SDK for .NET, modules, and scripts to the watcher node. This can take about ten minutes. If the watcher node doesn't appear, or if you see the state as `Not Monitored`, verify your Internet connectivity and IAM permissions. For more information, see [Troubleshooting the AWS Management Pack \(p. 931\)](#).



9. After the watcher node is discovered, dependent discoveries are triggered, and the AWS resources are added to the **Monitoring** workspace of the Operations console.

Note

The discovery of AWS resources should finish within twenty minutes. This process can take more time, based on your Operations Manager environment, your AWS environment, the load on the management server, and the load on the watcher node. For more information, see [Troubleshooting the AWS Management Pack \(p. 931\)](#).

Step 5: Configure Ports and Endpoints

The AWS Management Pack for Microsoft System Center must be able to communicate with AWS services to monitor the performance of those services and provide alerts in System Center. For monitoring to succeed, you must configure the firewall on the Management Pack servers to allow outbound HTTP calls on ports 80 and 443 to the AWS endpoints for the following services.

This enables monitoring for the following AWS services:

- Amazon Elastic Compute Cloud (EC2)
- Elastic Load Balancing
- Auto Scaling
- AWS Elastic Beanstalk
- Amazon CloudWatch
- AWS CloudFormation

The AWS Management Pack uses the public APIs in the AWS SDK for .NET to retrieve information from these services over ports 80 and 443. Log on to each server and enable outbound firewall rules for ports 80 and 443.

If your firewall application supports more detailed settings you can configure specific endpoints for each service. An endpoint is a URL that is the entry point for a web service. For example, `ec2.us-west-2.amazonaws.com` is an entry point for the Amazon EC2 service. To configure endpoints on your firewall, [locate the specific endpoint URLs](#) for the AWS services you are running and specify those endpoints in your firewall application.

Using the AWS Management Pack

You can use the AWS Management Pack to monitor the health of your AWS resources.

Contents

- [Views \(p. 910\)](#)
- [Discoveries \(p. 924\)](#)
- [Monitors \(p. 925\)](#)
- [Rules \(p. 926\)](#)
- [Events \(p. 926\)](#)
- [Health Model \(p. 927\)](#)
- [Customizing the AWS Management Pack \(p. 928\)](#)

Views

The AWS Management Pack provides the following views, which are displayed in the **Monitoring** workspace of the Operations console.

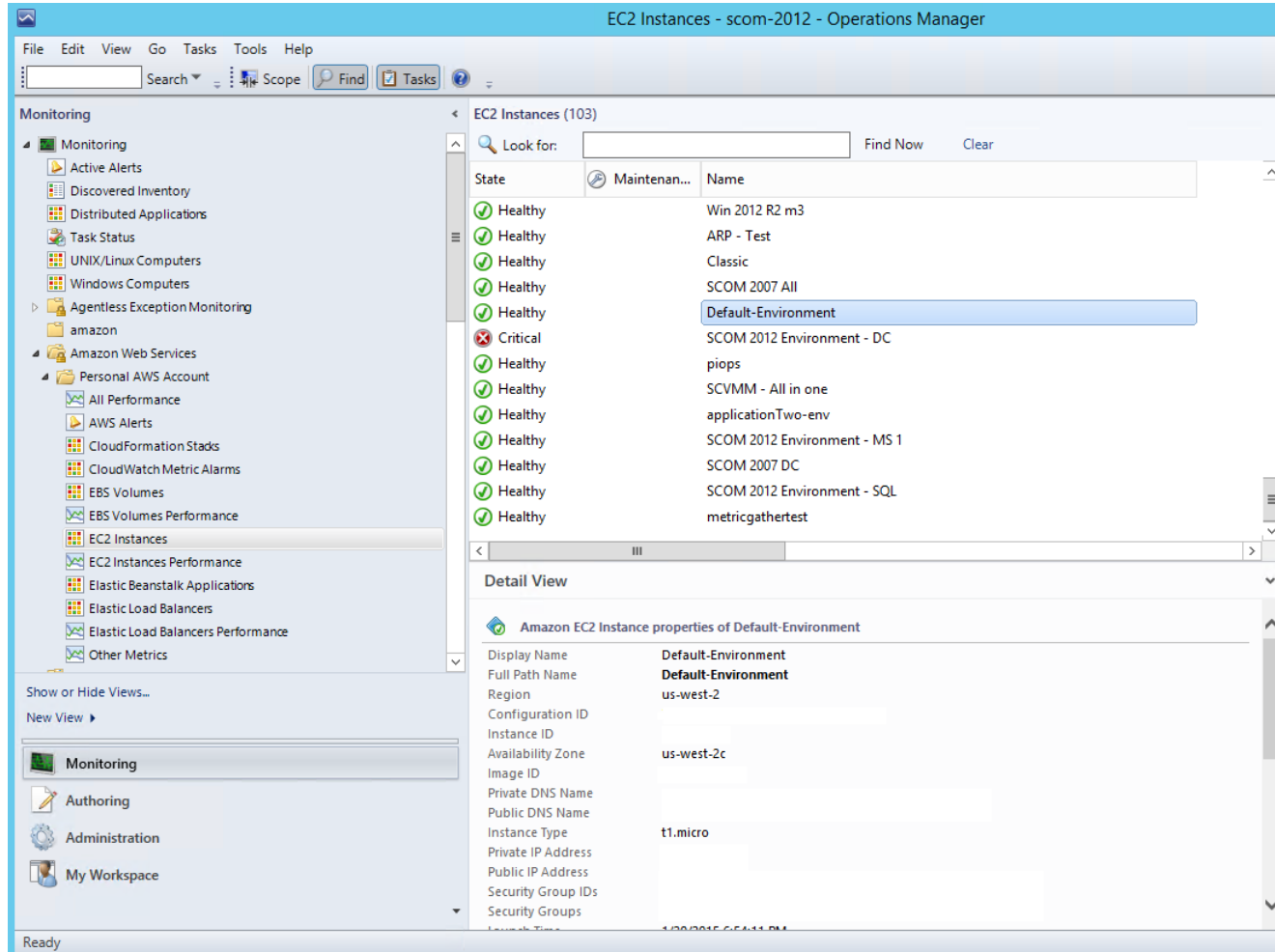
Views

- [EC2 Instances \(p. 910\)](#)
- [Amazon EBS Volumes \(p. 912\)](#)
- [Elastic Load Balancers \(p. 914\)](#)
- [AWS Elastic Beanstalk Applications \(p. 916\)](#)
- [AWS CloudFormation Stacks \(p. 918\)](#)
- [Amazon Performance Views \(p. 920\)](#)
- [Amazon CloudWatch Metric Alarms \(p. 921\)](#)
- [AWS Alerts \(p. 922\)](#)
- [Watcher Nodes \(System Center Operations Manager 2007 R2\) \(p. 923\)](#)

EC2 Instances

View the health state of the EC2 instances for a particular AWS account, from all Availability Zones and regions. The view also includes EC2 instances running in a virtual private cloud (VPC). The AWS Management Pack retrieves tags, so you can search and filter the list using those tags.

Amazon Elastic Compute Cloud User Guide for Windows Instances Views

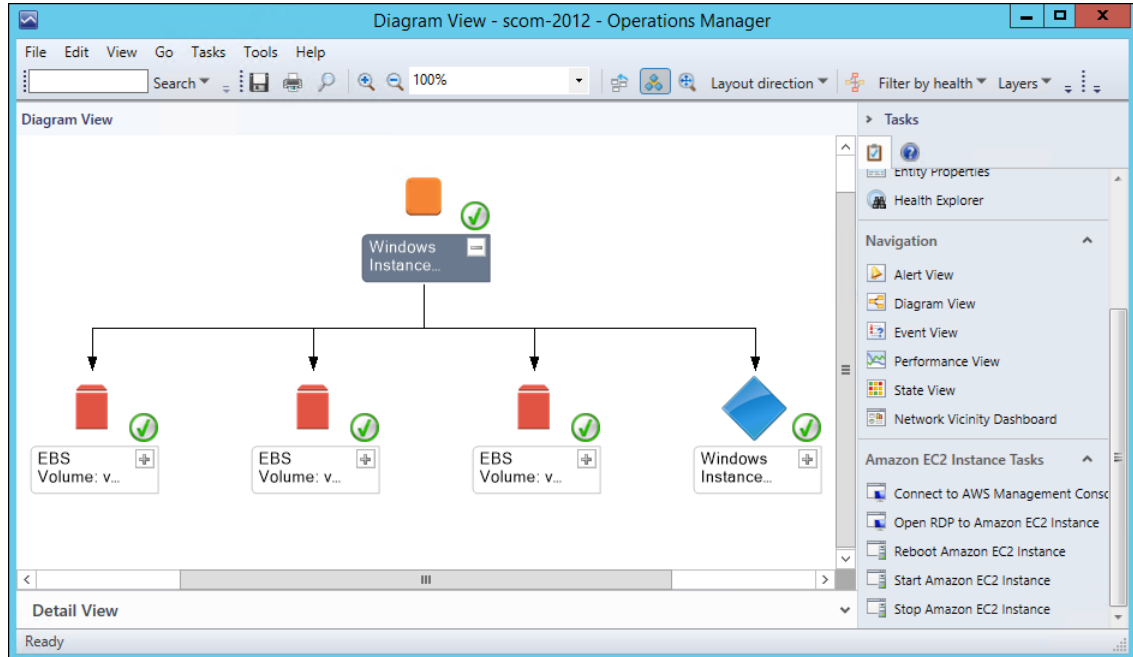


When you select an EC2 instance, you can perform instance health tasks:

- **Open Amazon Console:** Launches the AWS Management Console in a web browser.
- **Open RDP to Amazon EC2 Instance:** Opens an RDP connection to the selected Windows instance.
- **Reboot Amazon EC2 Instance:** Reboots the selected EC2 instance.
- **Start Amazon EC2 Instance:** Starts the selected EC2 instance.
- **Stop Amazon EC2 Instance:** Stops the selected EC2 instance.

EC2 Instances Diagram View

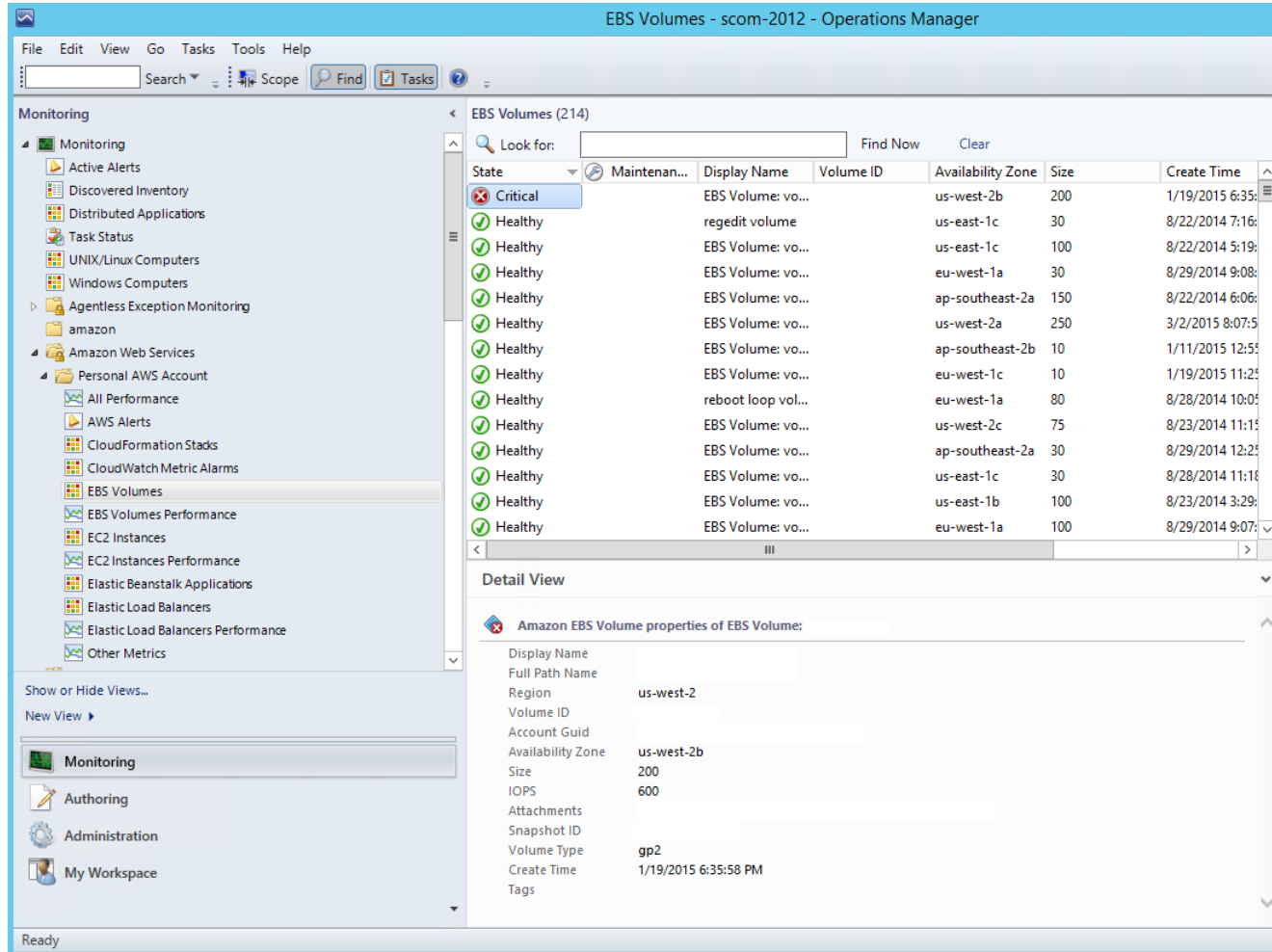
Shows the relationship of an instance with other components.



Amazon EBS Volumes

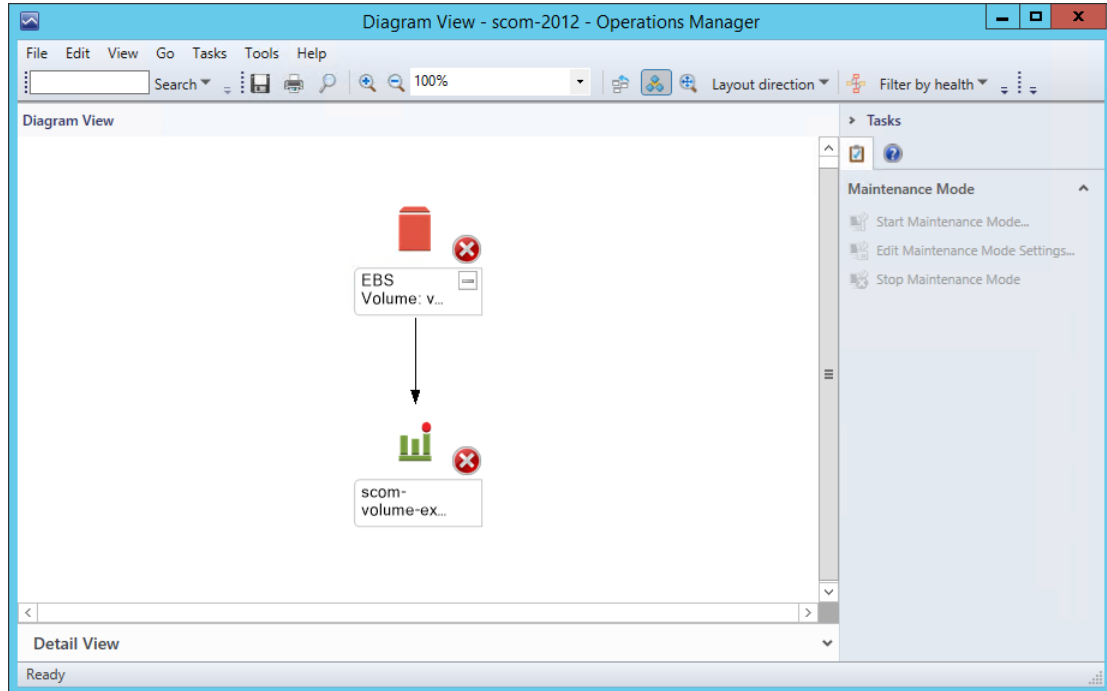
Shows the health state of all the Amazon EBS volumes for a particular AWS account from all Availability Zones and regions.

Amazon Elastic Compute Cloud User Guide for Windows Instances Views



Amazon EBS Volumes Diagram View

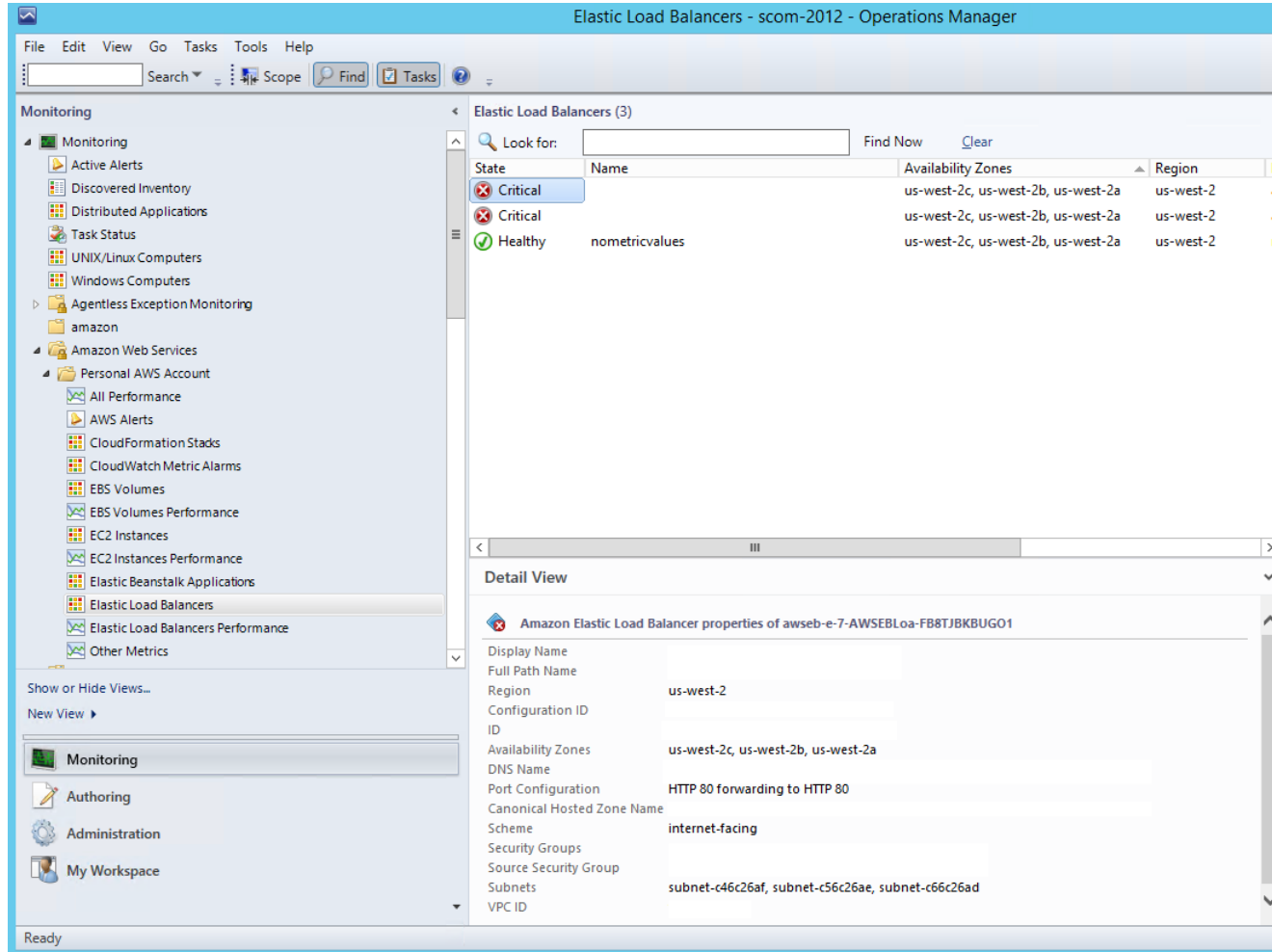
Shows an Amazon EBS volume and any associated alarms. The following illustration shows an example:



Elastic Load Balancers

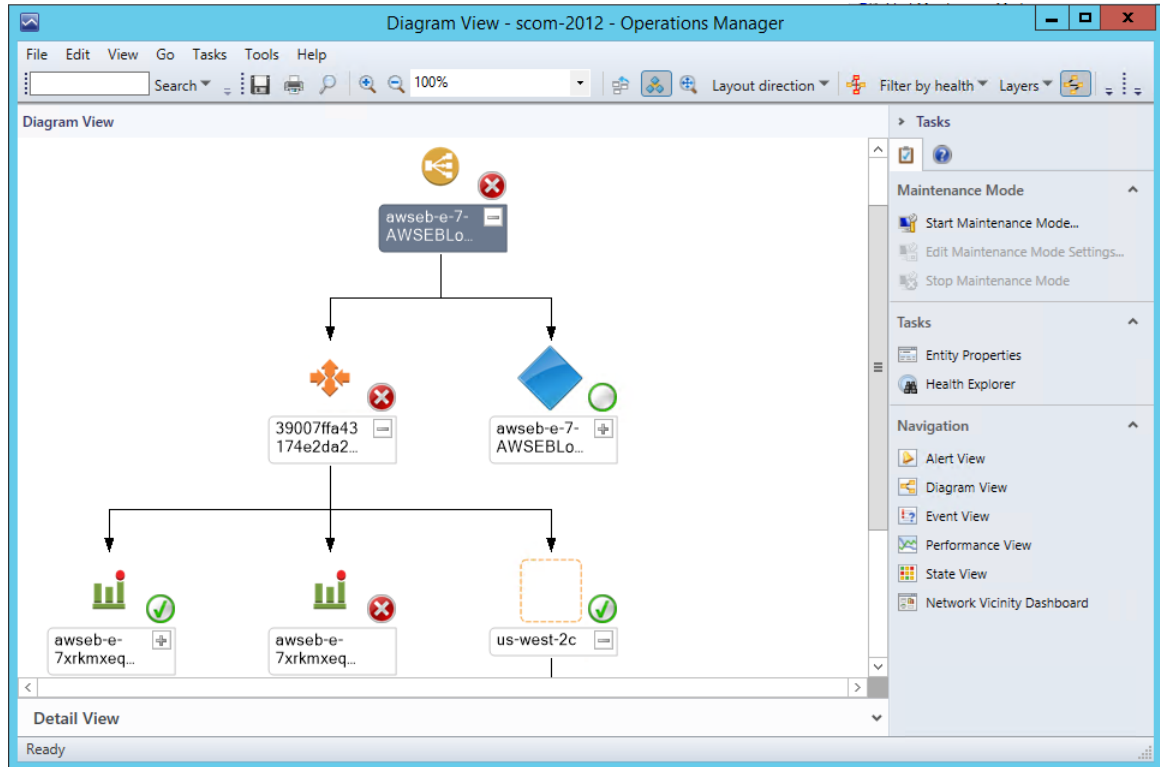
Shows the health state of all the load balancers for a particular AWS account from all regions.

Amazon Elastic Compute Cloud User Guide for Windows Instances Views



Elastic Load Balancing Diagram View

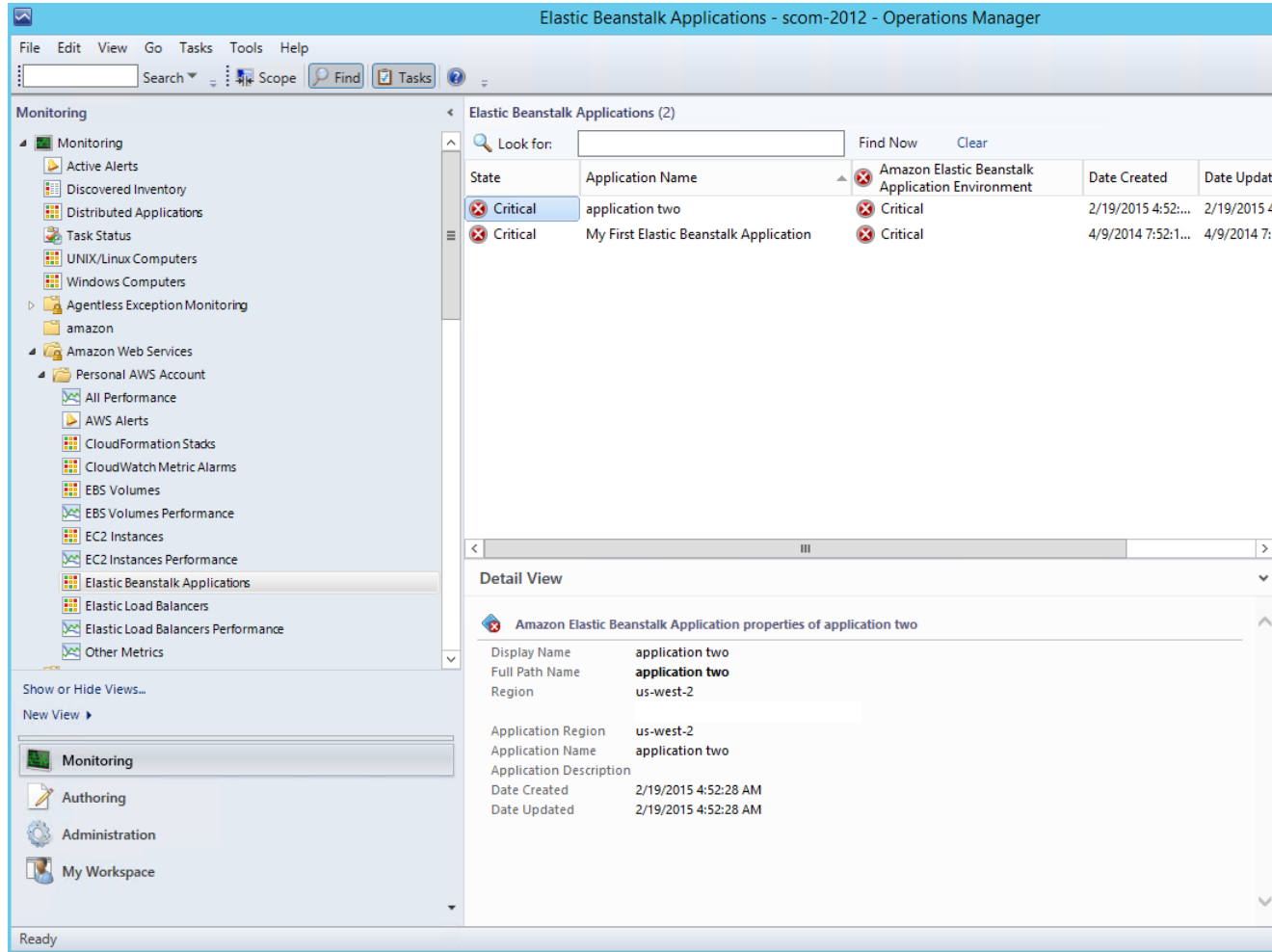
Shows the Elastic Load Balancing relationship with other components. The following illustration shows an example:



AWS Elastic Beanstalk Applications

Shows the state of all discovered AWS Elastic Beanstalk applications.

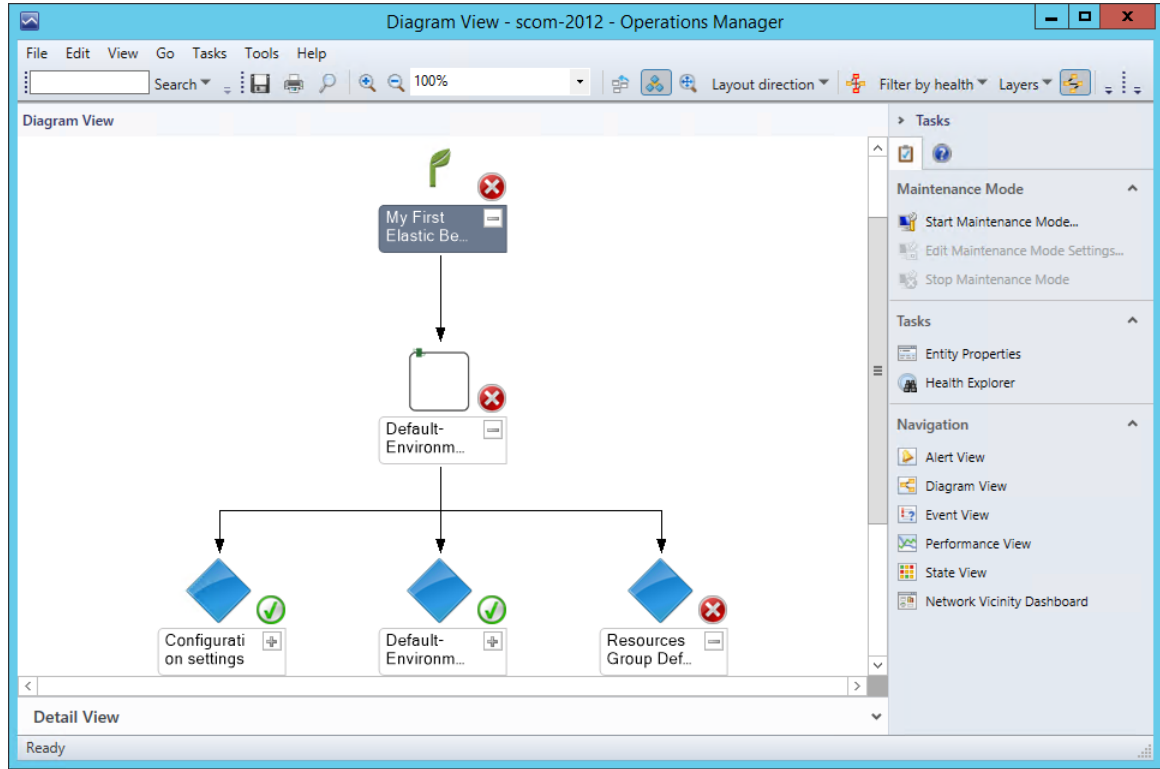
Amazon Elastic Compute Cloud User Guide for Windows Instances Views



AWS Elastic Beanstalk Applications Diagram View

Shows the AWS Elastic Beanstalk application, application environment, application configuration, and application resources objects.

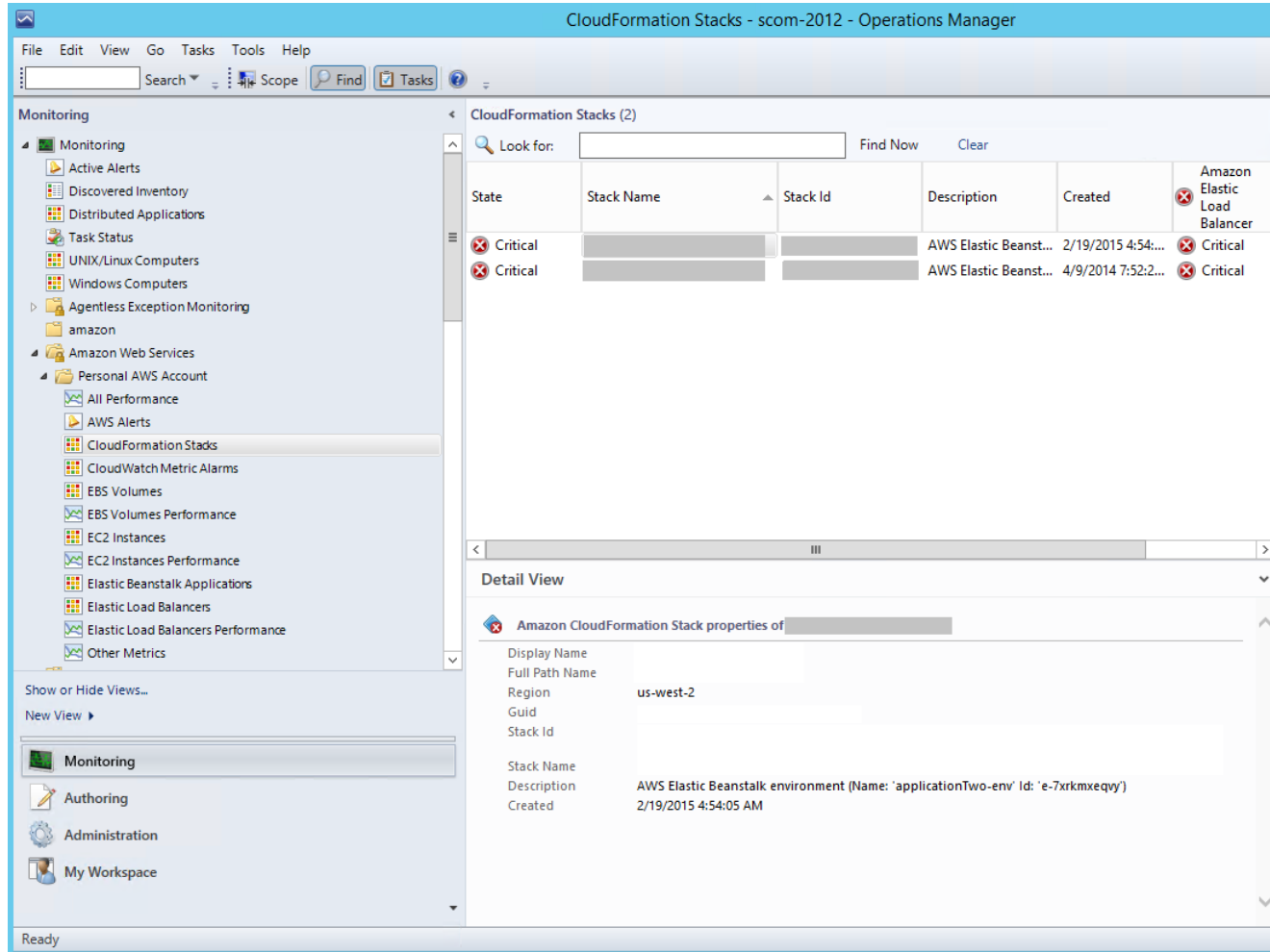
Amazon Elastic Compute Cloud User Guide for Windows Instances Views



AWS CloudFormation Stacks

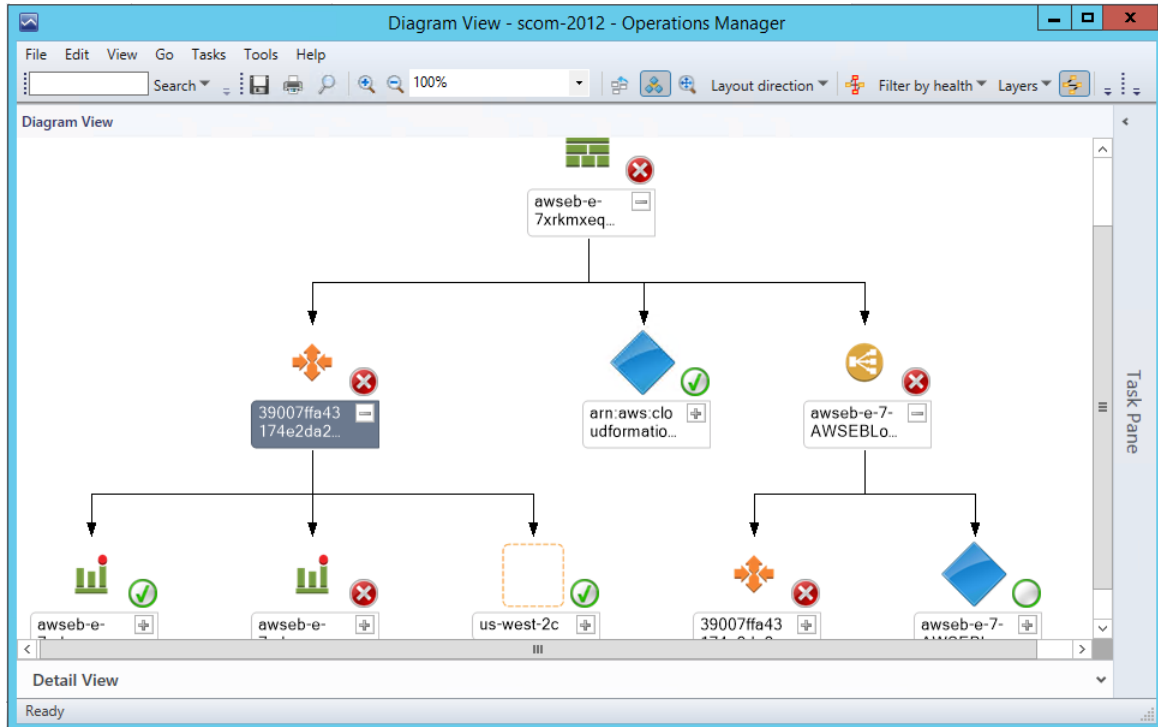
Shows the health state of all the AWS CloudFormation stacks for a particular AWS account from all regions.

Amazon Elastic Compute Cloud User Guide for Windows Instances Views



AWS CloudFormation Stacks Diagram View

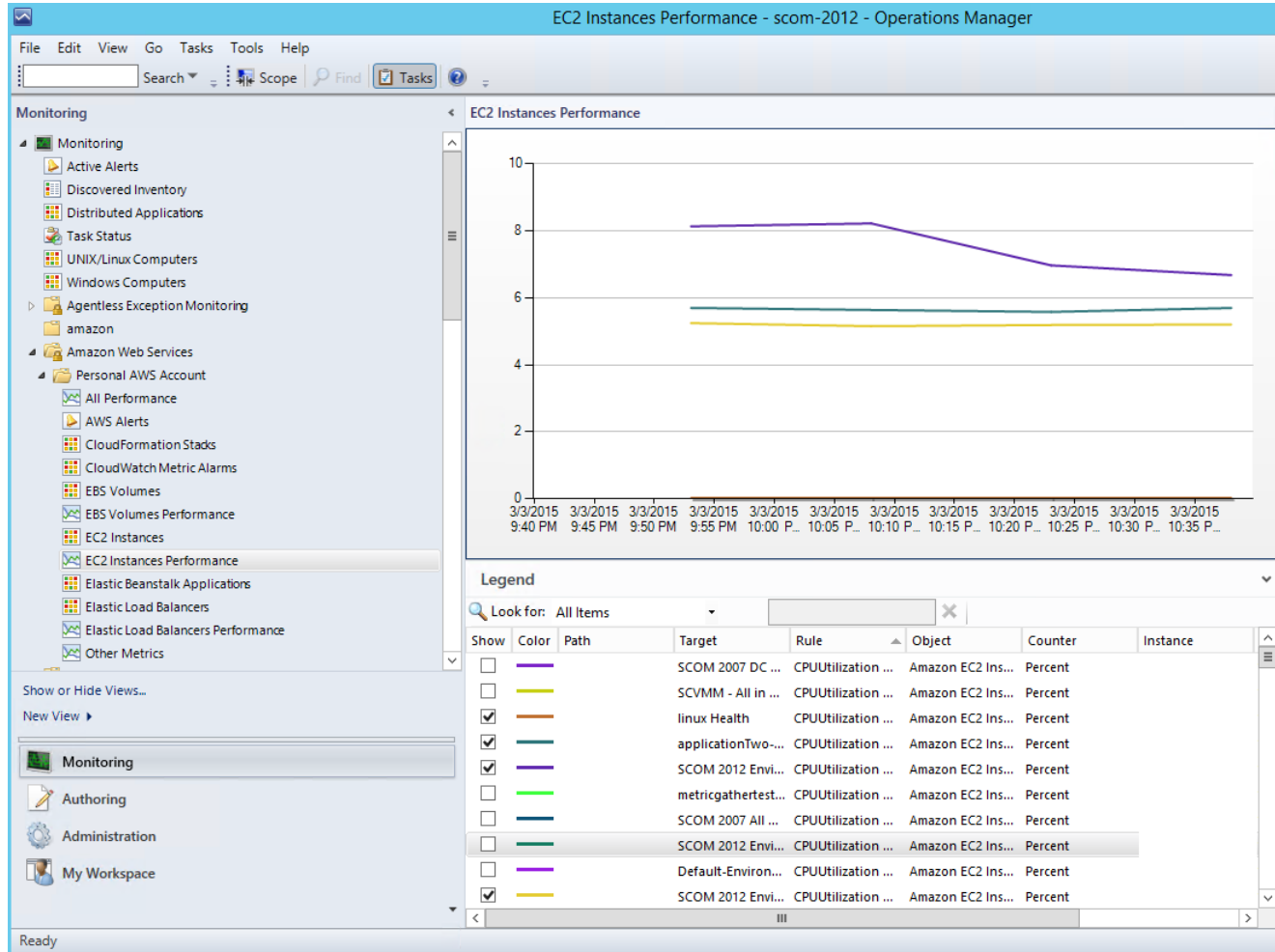
Shows the AWS CloudFormation stack relationship with other components. An AWS CloudFormation stack might contain Amazon EC2 or Elastic Load Balancing resources. The following illustration shows an example:



Amazon Performance Views

Shows the Amazon CloudWatch metrics for Amazon EC2, Amazon EBS, and Elastic Load Balancing, custom metrics, and metrics created from CloudWatch alarms. In addition, there are separate performance views for each resource. The **Other Metrics** performance view contains custom metrics, and metrics created from CloudWatch alarms. For more information about these metrics, see the [CloudWatch Metrics, Namespaces, and Dimensions Reference](#) in the *Amazon CloudWatch Developer Guide*. The following illustration shows an example.

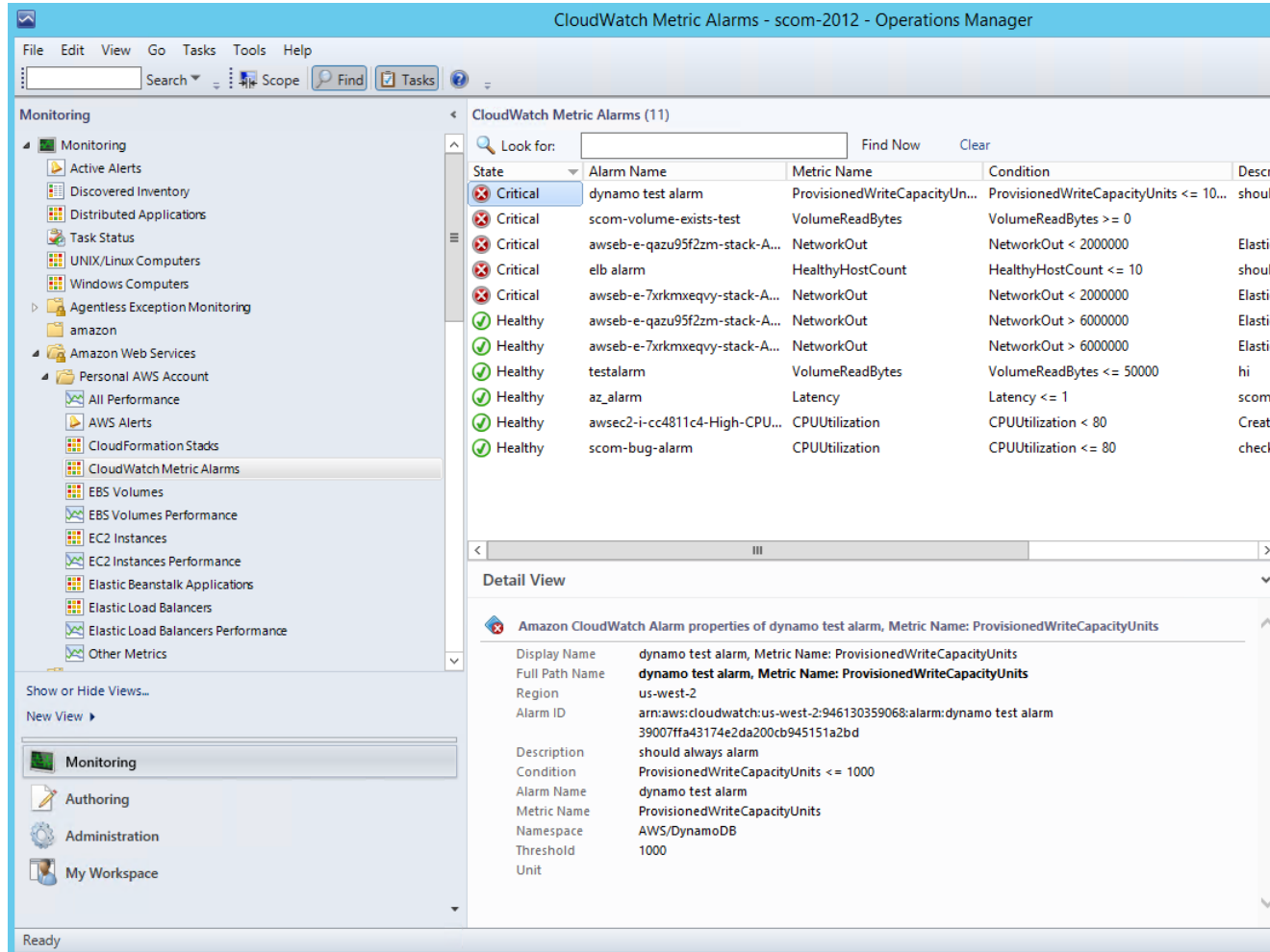
Amazon Elastic Compute Cloud User Guide for Windows Instances Views



Amazon CloudWatch Metric Alarms

Shows Amazon CloudWatch alarms related to the discovered AWS resources.

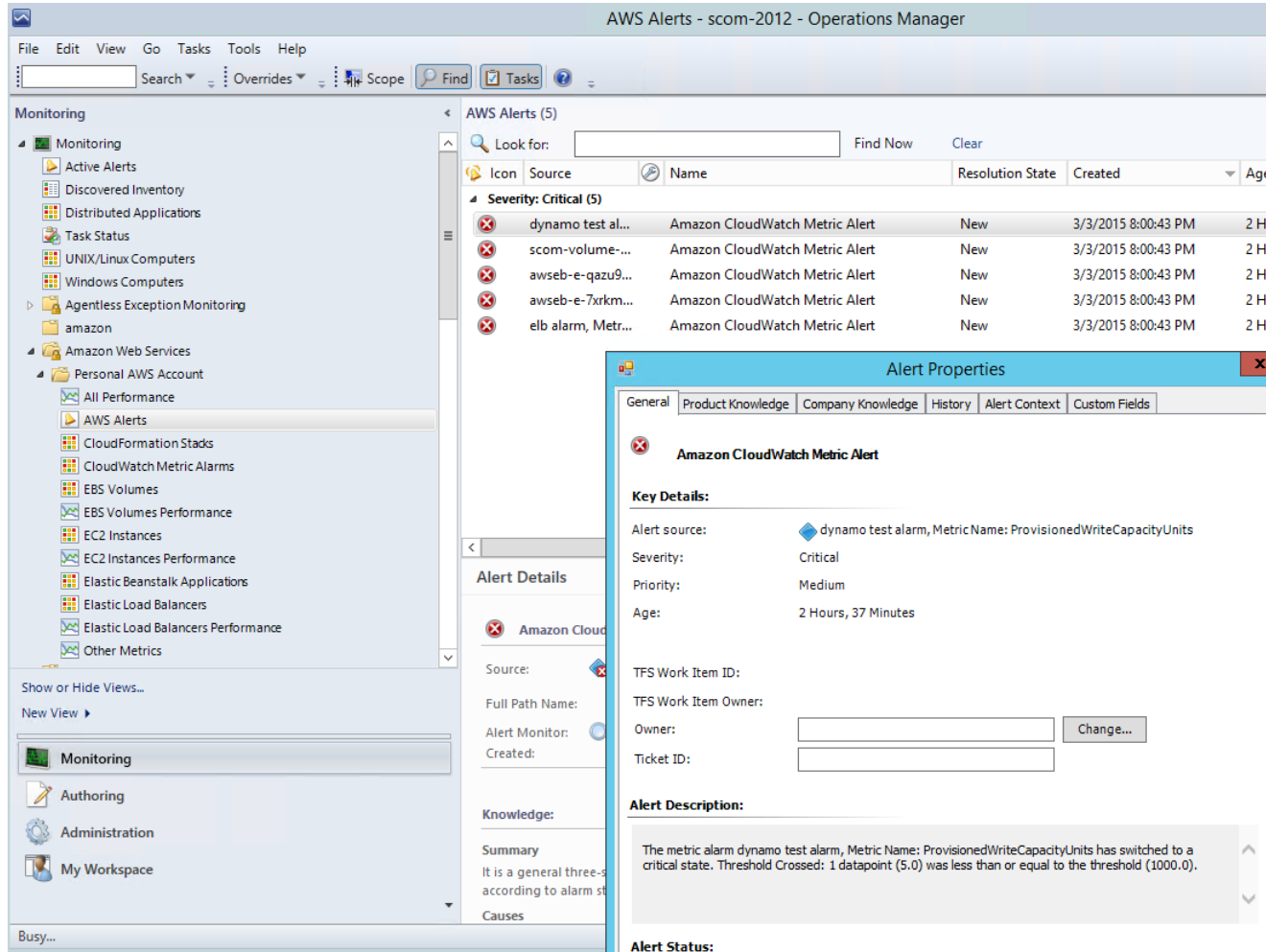
Amazon Elastic Compute Cloud User Guide for Windows Instances Views



AWS Alerts

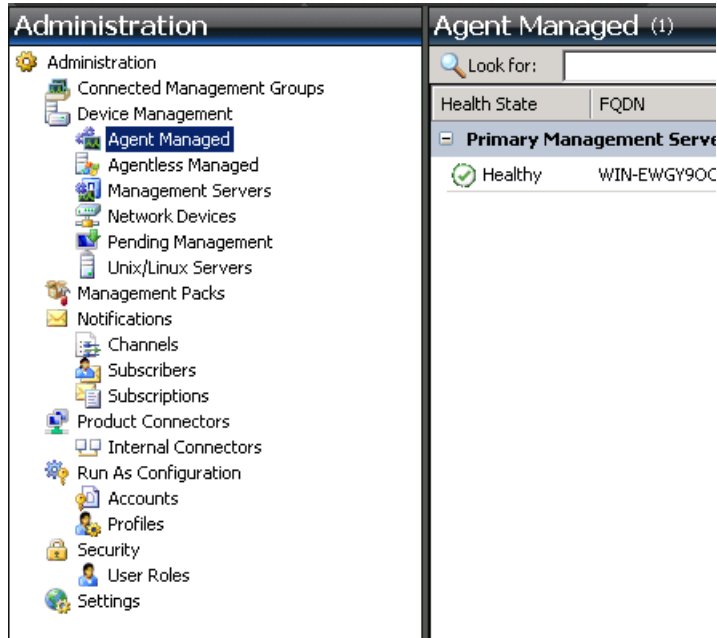
Shows the alerts that the AWS management pack produces when the health of an object is in a critical state.

Amazon Elastic Compute Cloud User Guide for Windows Instances Views



Watcher Nodes (System Center Operations Manager 2007 R2)

View the health state of the watcher nodes across all of the AWS accounts that are being monitored. A **Healthy** state means that the watcher node is configured correctly and can communicate with AWS.



Discoveries

Discoveries are the AWS resources that are monitored by the AWS Management Pack. The AWS Management Pack discovers the following objects:

- Amazon EC2 instances
- EBS volumes
- ELB load balancers
- AWS CloudFormation stacks
- Amazon CloudWatch alarms
- AWS Elastic Beanstalk applications
- Auto Scaling groups and Availability Zones

Amazon CloudWatch metrics are generated for the following resources:

- Amazon EC2 instance
- EBS volume
- Elastic Load Balancing
- Custom Amazon CloudWatch metrics
- Metrics from existing Amazon CloudWatch alarms

For Amazon CloudWatch metrics discovery, the following guidelines apply:

- AWS CloudFormation stacks do not have any default Amazon CloudWatch metrics.
- Stopped Amazon EC2 instances or unused Amazon EBS volumes do not generate data for their default Amazon CloudWatch metrics.
- After starting an Amazon EC2 instance, it can take up to 30 minutes for the Amazon CloudWatch metrics to appear in Operations Manager.
- Amazon CloudWatch retains the monitoring data for two weeks, even if your AWS resources have been terminated. This data appears in Operations Manager.

- An existing Amazon CloudWatch alarm for a resource that is not supported will create a metric and be associated with the Amazon CloudWatch alarm. These metric can be viewed in the Other Metrics performance view.

The AWS Management Pack also discovers the following relationships:

- AWS CloudFormation stack and its Elastic Load Balancing or Amazon EC2 resources
- Elastic Load Balancing load balancer and its EC2 instances
- Amazon EC2 instance and its EBS volumes
- Amazon EC2 instance and its operating system
- AWS Elastic Beanstalk application and its environment, configuration, and resources

The AWS Management Pack automatically discovers the relationship between an EC2 instance and the operating system running on it. To discover this relationship, the Operations Manager Agent must be installed and configured on the instance and the corresponding operating system management pack must be imported in Operations Manager.

Discoveries run on the management servers in the resource pool (System Center 2012) or the watcher node (System Center 2007 R2).

Discovery	Interval (seconds)
Amazon Resources Discovery (SCOM 2012) Discovers EC2 instances, Amazon EBS volumes, load balancers, and CloudFront stacks.	14400
AWS Elastic Beanstalk Discovery Discovers AWS Elastic Beanstalk and its relationship with environment, resources, and configuration.	14400
CloudWatch Alarms Discovery Discovers alarms generated using CloudWatch metrics.	900
Custom CloudWatch Metric Discovery Discovers custom CloudWatch metrics.	14400
Watcher Node Discovery (SCOM 2007 R2) Targets the root management server and creates the watcher node objects.	14400

Monitors

Monitors are used to measure the health of your AWS resources. Monitors run on the management servers in the resource pool (System Center 2012) or the watcher node (System Center 2007 R2).

Monitor	Interval (seconds)
AWS CloudFormation Stack Status	900
Amazon CloudWatch Metric Alarm	300

Monitor	Interval (seconds)
Amazon EBS Volume Status	900
Amazon EC2 Instance Status	900
Amazon EC2 Instance System Status	900
AWS Elastic Beanstalk Status	900
Watcher Node to Amazon Cloud Connectivity (SCOM 2007 R2)	900

Rules

Rules create alerts (based on Amazon CloudWatch metrics) and collect data for analysis and reporting.

Rule	Interval (seconds)
<p>AWS Resource Discovery Rule (SCOM 2007 R2)</p> <p>Targets the watcher node and uses the AWS API to discover objects for the following AWS resources: EC2 instances, EBS volumes, load balancers, and AWS CloudFormation stacks. (CloudWatch metrics or alarms are not discovered). After discovery is complete, view the objects in the Not Monitored state.</p>	14400
Amazon Elastic Block Store Volume Performance Metrics Data Collection Rule	900
Amazon EC2 Instance Performance Metrics Data Collection Rule	900
Elastic Load Balancing Balancing Performance Metrics Data Collection Rule	900
Custom CloudWatch Metric Data Collection Rule	900

Events

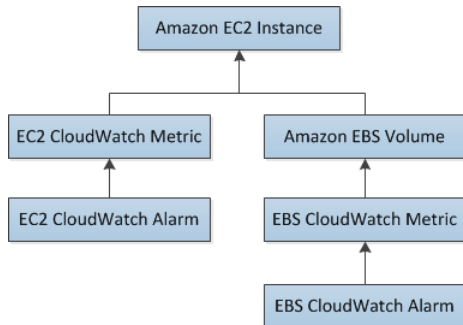
Events report on activities that involve the monitored resources. Events are written to the Operations Manager event log.

Event ID	Description
4101	Amazon EC2 Instance Discovery (General Discovery) finished
4102	Elastic Load Balancing Metrics Discovery, Amazon EBS Volume Metrics Discovery, Amazon EC2 Instance Metrics Discovery finished
4103	Amazon CloudWatch Metric Alarms Discovery finished
4104	Amazon Windows Computer Discovery finished
4105	Collecting Amazon Metrics Alarm finished
4106	EC2 Instance Computer Relation Discovery finished

Event ID	Description
4107	Collecting AWS CloudFormation Stack State finished
4108	Collecting Watcher Node Availability State finished
4109	Amazon Metrics Collection Rule finished
4110	Task to change Amazon Instance State finished
4111	EC2 Instance Status Monitor State finished
4112	Amazon EBS Volume Status Monitor State finished
4113	Amazon EC2 Instance Scheduled Events Monitor State calculated
4114	Amazon EBS Scheduled Events Monitor State calculated
4115	Elastic Beanstalk Discovery finished
4116	Elastic Beanstalk Environment Status State calculated
4117	Elastic Beanstalk Environment Operational State calculated
4118	Elastic Beanstalk Environment Configuration State calculated

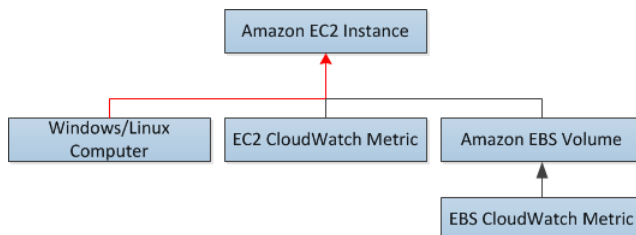
Health Model

The following illustration shows the health model defined by the AWS Management Pack.



The health state for a CloudWatch alarm is rolled up to its corresponding CloudWatch metric. The health state for a CloudWatch metric for Amazon EC2 is rolled up to the EC2 instance. Similarly, the health state for the CloudWatch metrics for Amazon EBS is rolled up to the Amazon EBS volume. The health states for the Amazon EBS volumes used by an EC2 instance are rolled up to the EC2 instance.

When the relationship between an EC2 instance and its operating system has been discovered, the operating system health state is rolled up to the EC2 instance.

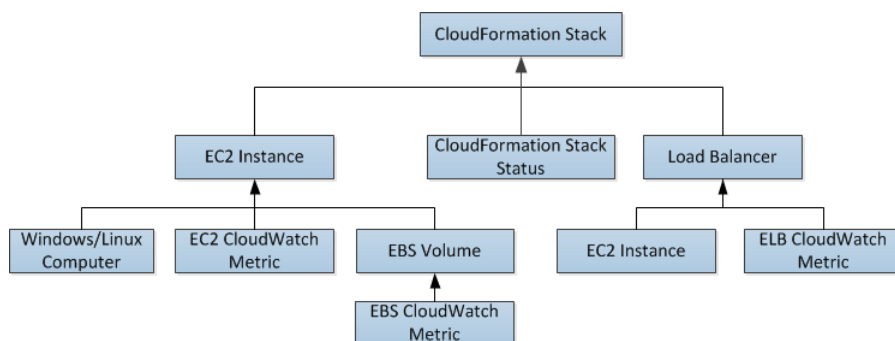


The health state of an AWS CloudFormation stack depends on the status of the AWS CloudFormation stack itself and the health states of its resources, namely the load balancers and EC2 instances.

The following table illustrates how the status of the AWS CloudFormation stack corresponds to its health state.

Health State	AWS CloudFormation Stack Status	Notes
Error	CREATE_FAILED DELETE_IN_PROGRESS DELETE_FAILED UPDATE_ROLLBACK_FAILED	Most likely usable
Warning	UPDATE_ROLLBACK_IN_PROGRESS UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS UPDATE_ROLLBACK_COMPLETE	Recovering after some problem
Healthy	CREATE_COMPLETE UPDATE_IN_PROGRESS UPDATE_COMPLETE_CLEANUP_IN_PROGRESS UPDATE_COMPLETE	Usable

The full health model for an AWS CloudFormation stack is as follows:



Customizing the AWS Management Pack

To change the frequency of discoveries, rules, and monitors, you can override the interval time (in seconds).

To change frequency

1. In the **Operations Manager** toolbar, click **Go**, and then click **Authoring**.
2. In the **Authoring** pane, expand **Management Pack Objects** and then click the object to change (for example, **Object Discoveries**, **Rules**, or **Monitors**).
3. In the toolbar, click **Scope**.
4. In the **Scope Management Pack Objects** dialog box, click **View all targets**.
5. To limit the scope to Amazon objects, type Amazon in the **Look for** field.

6. Select the object want to configure and click **OK**.
7. In the **Operations Manager** center pane, right-click the object to configure, click **Overrides**, and then click the type of override you want to configure.
8. Use the **Override Properties** dialog box to configure different values and settings for objects.

Tip

To disable a discovery, rule, or monitoring object right-click the object to disable in the **Operations Manager** center pane, click **Overrides**, and then click **Disable the Rule**. You might disable rules if, for example, you do not run AWS Elastic Beanstalk applications or use custom Amazon CloudWatch metrics.

For information about creating overrides, see [Tuning Monitoring by Using Targeting and Overrides](#) on the *Microsoft TechNet* website.

For information about creating custom rules and monitors, see [Authoring for System Center 2012 - Operations Manager](#) or [System Center Operations Manager 2007 R2 Management Pack Authoring Guide](#) on the *Microsoft TechNet* website.

Upgrading the AWS Management Pack

The procedure that you'll use to update AWS Management Pack depends on the version of System Center.

System Center 2012

To upgrade the AWS Management Pack

1. On the [AWS Add-Ins for Microsoft System Center](#) website, click **SCOM 2012**. Download `AWS-SCOM-MP-2.0-2.5.zip` to your computer and unzip it. The `.zip` file includes `Amazon.AmazonWebServices.mpb`.
2. In the Operations console, on the **Go** menu, click **Administration**, and then click **Management Packs**.
3. In the **Tasks** pane, click **Import Management Packs**.
4. On the **Select Management Packs** page, click **Add**, and then click **Add from disk**.
5. In the **Select Management Packs to import** dialog box, select the `Amazon.AmazonWebServices.mpb` file from the location where you downloaded it, and then click **Open**.
6. On the **Select Management Packs** page, under **Import list**, select the **Amazon Web Services** management pack, and then click **Install**.

If the **Install** button is disabled, upgrading to the current version is not supported and you must uninstall the AWS Management Pack before you can install the current version. For more information, see [Uninstalling the AWS Management Pack](#) (p. 930).

System Center 2007 R2

To upgrade the AWS Management Pack

1. On the Management Server, go to the [AWS Add-Ins for Microsoft System Center](#) website and click **SCOM 2007**. Save `AWS-MP-Setup-2.5.msi`, and then run it.
2. Click **Next** and follow the directions to upgrade the components that you installed previously.

3. If your root management server, Operations console, and watcher node are on different computers, you must download and run the setup program on each computer.
4. On the watcher node, open a Command Prompt window as an administrator and run the following commands.

```
C:\> net stop HealthService
The System Center Management service is stopping.
The System Center Management service was stopped successfully.

C:\> net start HealthService
The System Center Management service is starting.
The System Center Management service was started successfully.
```

5. In the Operations console, on the **Go** menu, click **Administration**, and then click **Management Packs**.
6. In the **Actions** pane, click **Import Management Packs**.
7. On the **Select Management Packs** page, click **Add**, and then click **Add from disk**.
8. In the **Select Management Packs to import** dialog box, change the directory to `C:\Program Files (x86)\Amazon Web Services Management Pack`, select the `Amazon.AmazonWebServices.mp` file, and then click **Open**.
9. On the **Select Management Packs** page, under **Import list**, select the **Amazon Web Services** management pack, and then click **Install**.

If the **Install** button is disabled, upgrading to the current version is not supported and you must uninstall AWS Management Pack first. For more information, see [Uninstalling the AWS Management Pack \(p. 930\)](#).

Uninstalling the AWS Management Pack

If you need to uninstall the AWS Management Pack, use the following procedure.

System Center 2012

To uninstall the AWS Management Pack

1. In the Operations console, on the **Go** menu, click **Administration**, and then click **Management Packs**.
2. Right-click **Amazon Web Services** and select **Delete**.
3. In the **Dependent Management Packs** dialog box, note the dependent management packs, and then click **Close**.
4. Right-click the dependent management pack and select **Delete**.
5. Right-click **Amazon Web Services** and select **Delete**.

System Center 2007 R2

To uninstall the AWS Management Pack

1. Complete steps 1 through 5 described for System Center 2012 in the previous section.
2. From Control Panel, open Programs and Features. Select `Amazon Web Services Management Pack` and then click **Uninstall**.

3. If your root management server, Operations console, and watcher node are on different computers, you must repeat this process on each computer.

Troubleshooting the AWS Management Pack

The following are common errors, events, and troubleshooting steps.

Contents

- [Errors 4101 and 4105 \(p. 931\)](#)
- [Error 4513 \(p. 931\)](#)
- [Event 623 \(p. 931\)](#)
- [Events 2023 and 2120 \(p. 932\)](#)
- [Event 6024 \(p. 932\)](#)
- [General Troubleshooting for System Center 2012 — Operations Manager \(p. 932\)](#)
- [General Troubleshooting for System Center 2007 R2 \(p. 933\)](#)

Errors 4101 and 4105

If you receive one of the following errors, you must upgrade the AWS Management Pack. For more information, see [Upgrading the AWS Management Pack \(p. 929\)](#).

```
Error 4101
Exception calling "DescribeVolumes" with "1" argument(s): "AWS was not able
to validate the
provided access credentials"
```

```
Error 4105
Exception calling "DescribeApplications" with "0" argument(s): "The security
token included
in the request is invalid"
```

Error 4513

If you receive one of the following error, you must upgrade the AWS Management Pack. For more information, see [Upgrading the AWS Management Pack \(p. 929\)](#).

```
Error 4513
The callback method DeliverDataToModule failed with exception "Resolution of
the dependency
failed, type = "Amazon.SCOM.SDK.Interfaces.IMonitorSdk", name = "(none)".
Exception occurred while: Calling constructor
Amazon.SCOM.SDK.CloudWatch.AwsMonitorSdk
(System.String awsAccessKey, System.String awsSecretKey).
Exception is: InvalidOperationException - Collection was modified;
enumeration operation
may not execute.
```

Event 623

If you find the following event in the Windows event log, follow the solution described in [KB975057](#).

```
Event ID: 623
HealthService (process_id) The version store for instance instance ("name")
has reached
its maximum size of size MB. It is likely that a long-running transaction is
preventing
cleanup of the version store and causing it to build up in size. Updates will
be rejected
until the long-running transaction has been completely committed or rolled
back.
Possible long-running transaction:
SessionId: id
Session-context: value
Session-context ThreadId: id
Cleanup: value
```

Events 2023 and 2120

If you find the following events in the Windows event log, see [Event ID 2023 and 2120](#) for more information.

```
Event ID: 2023
The Health Service has removed some items from the send queue for management
group "Servers"
since it exceeded the maximum allowed size of size megabytes.
```

```
Event ID: 2120
The Health Service has deleted one or more items for management group
"Servers" which could
not be sent in 1440 minutes.
```

Event 6024

If you find the following event in the Windows event log, see [Health Service Restarts](#) for more information.

```
Event ID: 6024
LaunchRestartHealthService.js : Launching Restart Health Service. Health
Service exceeded
Process\Handle Count or Private Bytes threshold.
```

General Troubleshooting for System Center 2012 — Operations Manager

Try the following to resolve any issues.

- Verify that you have installed the latest Update Rollup for System Center 2012 — Operations Manager. The AWS Management Pack requires at least Update Rollup 1.
- Ensure that you have configured the AWS Management Pack after importing it by running the Add Monitoring Wizard. For more information, see [Step 1: Installing the AWS Management Pack \(p. 901\)](#).
- Verify that you have waited long enough for the AWS resources to be discovered (10–20 minutes).

- Verify that the management servers are configured properly.
 - Management servers must have Internet connectivity.
 - The action account for a management server must have local administrator privileges on the management server.
 - The management server must have the .NET Framework 4.5. or later.
- Verify that the AWS Run As account is valid.
 - The values for the access key ID and secret access key are correct.
 - The access keys are active: In the AWS Management Console, click your name in the navigation bar and then click **Security Credentials**.
 - The IAM user has at least read-only access permission. Note that read-only access allows the user actions that do not change the state of a resource, such as monitoring, but do not allow the user actions like launching or stopping an instance.
 - If an Amazon CloudWatch metric shows as **Not Monitored**, check whether at least one Amazon CloudWatch alarm has been defined for that Amazon CloudWatch metric.
 - For further troubleshooting, use the information in the event logs.
 - Check the Operations Manager event log on the management server. For more information, see [Events \(p. 926\)](#) for a list of the events that the AWS Management Pack writes to the Operations Manager event log.

General Troubleshooting for System Center 2007 R2

Try the following to resolve any issues.

- Ensure that you have configured the AWS Management Pack after importing it by running the Add Monitoring Wizard. For more information, see [Step 1: Installing the AWS Management Pack \(p. 901\)](#).
- Verify that you have waited long enough for the AWS resources to be discovered (10–20 minutes).
- Verify that the watcher node is configured properly.
 - The proxy agent is enabled. For more information, see [Step 2: Configuring the Watcher Node \(p. 902\)](#).
 - The watcher node has Internet connectivity.
 - The action account for the watcher node has local administrator privileges.
 - The watcher node must have the .NET Framework 3.5.1 or later.
- Verify that the watcher node is healthy and resolve all alerts. For more information, see [Views \(p. 910\)](#).
- Verify that the AWS Run As account is valid.
 - The values for the access key ID and secret access key are correct.
 - The access keys are active: In the AWS Management Console, click your name in the navigation bar and then click **Security Credentials**.
 - The IAM user has at least read-only access permission. Note that read-only access allows the user actions that do not change the state of a resource, such as monitoring, but do not allow the user actions like launching or stopping an instance.
 - If an Amazon CloudWatch metric shows as **Not Monitored**, check whether at least one Amazon CloudWatch alarm has been defined for that Amazon CloudWatch metric.
 - For further troubleshooting, use the information in the event logs.
 - Check the Operations Manager event log on the management server as well as the watcher node. For more information, see [Events \(p. 926\)](#) for a list of the events that the AWS Management Pack writes to the Operations Manager event log.

AWS Diagnostics for Windows Server - Beta

AWS Diagnostics for Windows Server is a easy-to-use tool that you run on an Amazon EC2 Windows Server instance to diagnose and troubleshoot possible problems. It is valuable not just for collecting log files and troubleshooting issues, but also proactively searching for possible areas of concern. For example, this tool can diagnose configuration issues between the Windows Firewall and the AWS security groups that might affect your applications. It can even examine EBS boot volumes from other instances and collect relevant logs for troubleshooting Windows Server instances using that volume.

One use for AWS Diagnostics for Windows Server is diagnosing problems with Key Management Service (KMS) activations. KMS activation can fail if you have changed the DNS server, added instances to a domain, or if the server time is out of sync. In this case, instead of trying to examine your configuration settings manually and debugging the issue, run the AWS Diagnostics for Windows Server tool to give you the information you need about possible issues.

The tool can also find differences between the rules in an security group and the Windows Firewall. If you provide your AWS user credentials to describe your security groups, the AWS Diagnostics for Windows Server tool is able verify whether the ports listed in a security group are allowed through the Windows Firewall. You eliminate the need to look at firewall rules manually and verify them against the security group rules.

The AWS Diagnostics for Windows Server tool is free and can be downloaded and installed from [AWS Diagnostics for Windows Server - Beta](#).

AWS Diagnostics for Windows Server has two different modules: a data collector module that collects data from all different sources, and an analyzer module that parses the data collected against a series of predefined rules to identify issues and provide suggestions.

The AWS Diagnostics for Windows Server tool only runs on Windows Server running on an EC2 instance. When the tool starts, it checks whether it is running on an EC2 instance. If the check fails, the tool displays the `EC2InstanceCheckFailed` error message.

Analysis Rules

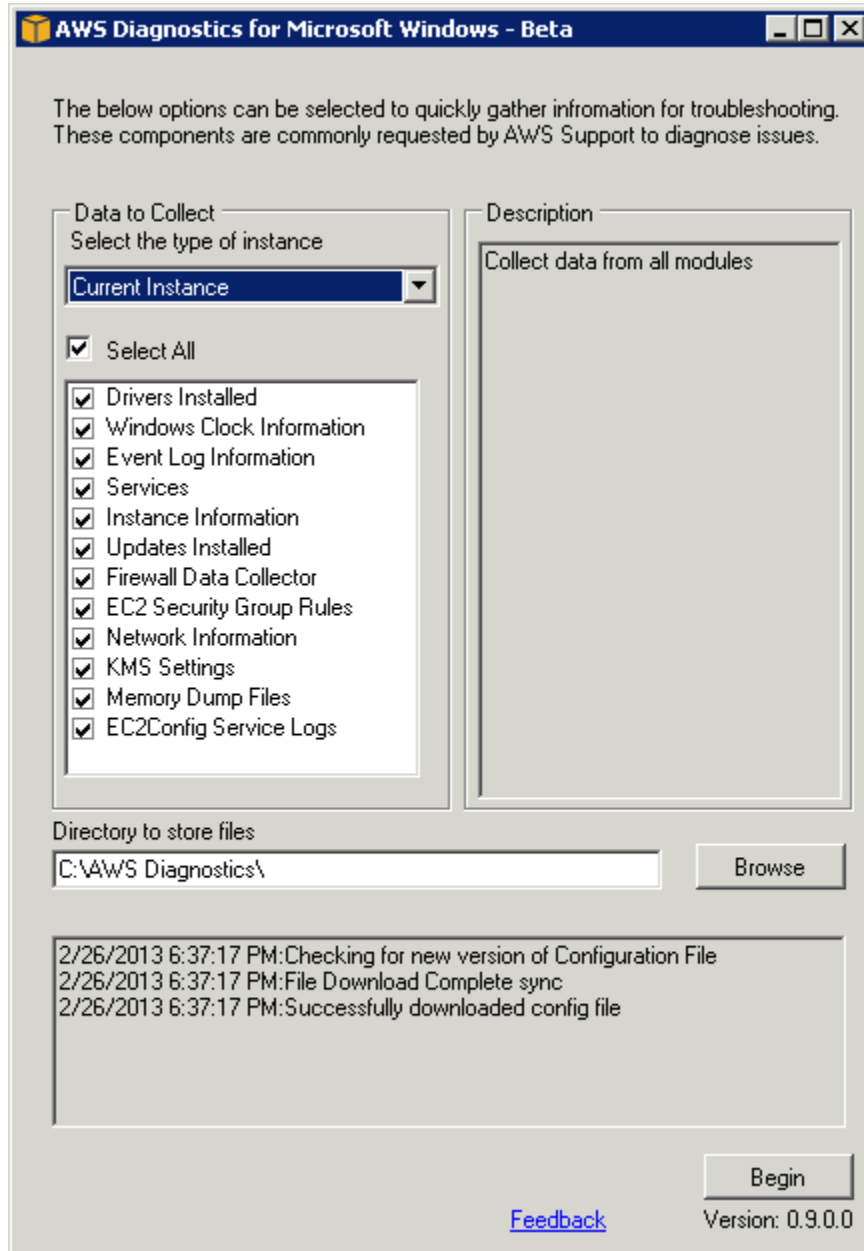
AWS Diagnostics for Windows Server provides the following analysis rules:

- Check for activation status and KMS settings
- Check for proper route table entries for metadata and KMS access
- Compare security group rules with Windows Firewall rules
- Check the version of the PV driver (RedHat or Citrix)
- Check whether the `RealTimeIsUniversal` registry key is set
- Check the default gateway settings if using multiple NICs
- Bug check code in mini dump files

Even if the analyzer doesn't report any problems, the data collected by the tool might still be useful. You can view the data files created by the tool to look for problems or provide these files to AWS Support to help resolve a support case.

Analyzing the Current Instance

To analyze the current instance, run the AWS Diagnostics for Windows Server tool and select **Current Instance** for the type of instance. In the **Data to Collect** section of the main window, specify the data that AWS Diagnostics for Windows Server collects.



Data	Description
Drivers Installed	Collects information about all drivers installed on the instance.
Windows Clock Information	Collects current time and time zone information for the instance.
Event Log Information	Collects critical, error, and warning messages from the event logs.
Services	Collects information about the services that are installed on the instance.

Data	Description
Instance Information	Collects information from the instance metadata and local environment variables.
Updates Installed	Collects information about the updates that are installed on the instance.
Firewall Data Collector	Collects information about the Windows Firewall settings.
EC2 Security Group Rules	Collects information about the rules in the Amazon EC2 security groups associated with the instance.
Network Information	Collects route table and IP address information for the instance.
KMS Settings	Collects Key Management Service settings.
Memory Dump Files	Collects any memory dump files that exist on the instance.
EC2Config Service Logs	Collects log files generated by the EC2Config service.

Collecting Data From an Offline Instance

The **Offline Instance** option is useful when you want to debug a problem with a Windows instance that is either unable to boot up or is preventing you from running the AWS Diagnostics for Windows Server tool on it. In this case, you can detach the EBS boot volume from that instance and attach it to another Windows instance.

To collect data from an offline instance

1. Stop the faulty instance, if it is not stopped already.
2. Detach the EBS boot volume from the faulty instance.
3. Attach the EBS boot volume to another working Windows instance that has AWS Diagnostics for Windows Server installed on it
4. Mount the volume in the working instance, assigning it a drive letter (for example, F:).
5. Run the AWS Diagnostics for Windows Server tool on the working instance and select **Offline Instance**.
6. Choose the drive letter of the newly mounted volume (for example, F:).
7. Click **Begin**.

The AWS Diagnostics for Windows Server tool scans the volume and collects troubleshooting information based on the log files that are on the volume. For offline instances, the data collected is a fixed set, and no analysis of the data is performed.

Data File Storage

By default, the AWS Diagnostics for Windows Server tool places its data files in the directory from which you launch the tool. You can choose where to save the data files that are collected by the AWS

Diagnostics for Windows Server tool. Within the chosen directory, the tool creates a directory named `DataCollected`. Each time it runs, the tool also creates a separate directory with the current date and time stamp. Each data collection module produces an XML file that contains information for that data set. Finally, the tool creates a ZIP file archive containing copies of all of the data files generated. You can provide this archive to an AWS support engineer if needed.

Troubleshooting Windows Instances

The following procedures and tips can help you troubleshoot problems with your Amazon EC2 Windows instances.

Topics

- [Troubleshoot an Unreachable Instance \(p. 939\)](#)
- [Common Issues \(p. 946\)](#)
- [Common Messages \(p. 955\)](#)

If you need additional help, you can post a question to the [Amazon EC2 forum](#). Be sure to post the ID of your instance and any error messages, including error messages available through console output.

To get additional information for troubleshooting problems with your instance, use [AWS Diagnostics for Windows Server - Beta \(p. 934\)](#). For information about troubleshooting issues with PV drivers, see [Troubleshooting PV Drivers \(p. 363\)](#).

Troubleshoot an Unreachable Instance

If you are unable to reach your instance via SSH or RDP, you can capture a screenshot of your instance and view it as an image. This provides visibility as to the status of the instance, and allows for quicker troubleshooting.

There is no data transfer cost for this screenshot. The image is generated in JPG format, no larger than 100kb. This section includes the following information.

- [How to Take a Screenshot of an Unreachable Instance \(p. 939\)](#)
- [Common Screenshots \(p. 940\)](#)

How to Take a Screenshot of an Unreachable Instance

To access the instance console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the left navigation pane, choose **Instances**.

3. Select the instance to capture.
4. Choose **Actions, Instance Settings**.
5. Choose **Get Instance Screenshot**.

Right-click on the image to download and save it.

To capture a screenshot using the command line

You can use one of the following commands. The returned output is base64-encoded. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [get-console-screenshot](#) (AWS CLI)
- [GetConsoleScreenshot](#) (Amazon EC2 Query API)

For API calls, the returned content is base64-encoded. For command line tools, the decoding is performed for you.

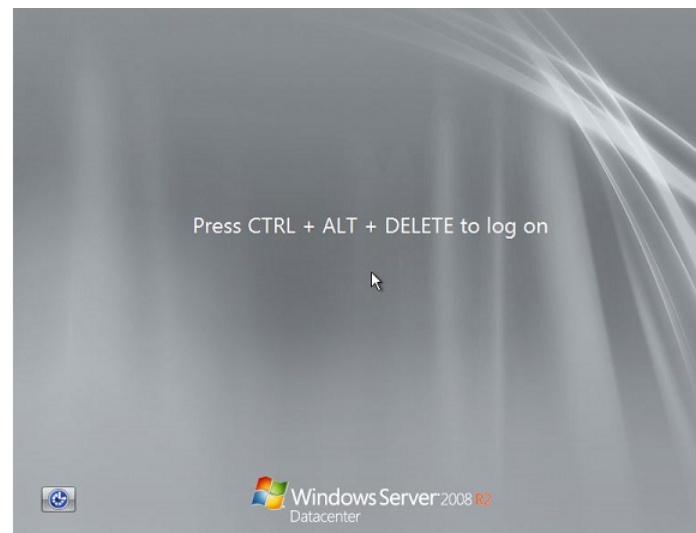
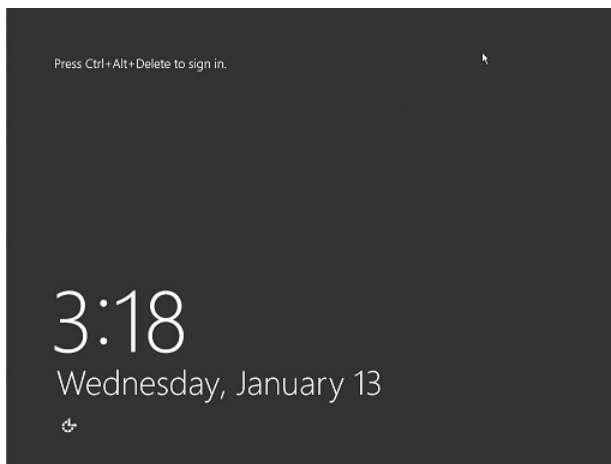
Common Screenshots

You can use the following information to help you troubleshoot an unreachable instance based on screenshots returned by the service.

- [Log On Screen \(Ctrl+Alt+Delete\) \(p. 940\)](#)
- [Recovery Console Screen \(p. 943\)](#)
- [Windows Boot Manager Screen \(p. 944\)](#)
- [Sysprep Screen \(p. 944\)](#)
- [Getting Ready Screen \(p. 945\)](#)
- [Windows Update Screen \(p. 946\)](#)
- [Chkdsk \(p. 946\)](#)

Log On Screen (Ctrl+Alt+Delete)

Console Screenshot Service returned the following.



If an instance becomes unreachable during log on, there could be a problem with your network configuration or Windows Remote Desktop Services. An instance can also be unresponsive if a process is using large amounts of CPU.

Network Configuration

Use the following information, to verify that your AWS, Microsoft Windows, and local (or on-premises) network configurations aren't blocking access to the instance.

AWS Network Configuration

Configuration	Verify
Security group configuration	Verify that port 3389 is open for your security group. Verify you are connecting to the right public IP address. If the instance was not associated with an Elastic IP, the public IP changes after the instance stops/starts. For more information, see Remote Desktop can't connect to the remote computer (p. 951) .
VPC configuration (Network ACLs)	Verify that the access control list (ACL) for your Amazon VPC is not blocking access. For information, see Network ACLs in the Amazon VPC User Guide .
VPN configuration	If you are connecting to your VPC using a virtual private network (VPN), verify VPN tunnel connectivity. For more information, see How do I troubleshoot VPN tunnel connectivity to an Amazon VPC?

Windows Network Configuration

Configuration	Verify
Windows Firewall	Verify that Windows Firewall isn't blocking connections to your instance. Disable Windows Firewall as described in bullet 7 of the remote desktop troubleshooting section, Remote Desktop can't connect to the remote computer (p. 951) .
Advanced TCP/IP configuration (Use of static IP)	The instance may be unresponsive because you configured a static IP address. For a VPC, Create a network interface (p. 723) and attach it to the instance (p. 725) . For EC2 Classic, enable DHCP.

Local or On-Premises Network Configuration

Verify that a local network configuration isn't blocking access. Try to connect to another instance in the same VPC as your unreachable instance. If you can't access another instance, work with your local network administrator to determine whether a local policy is restricting access.

Remote Desktop Service Issue

If the instance can't be reached during log on, there could a problem with Remote Desktop Services (RDS) on the instance.

Remote Desktop Services Configuration

Configuration	Verify
RDS is running	Verify that RDS is running on the instance. Connect to the instance using the Microsoft Management Console (MMC) Services snap-in (services.msc). In the list of services, verify that Remote Desktop Services is Running . If it isn't, start it and then set the startup type to Automatic . If you can't connect to the instance by using the Services snap-in, detach the root volume from the instance, take a snapshot of the volume or create an AMI from it, attach the original volume to another instance in the same availability zone as a secondary volume, and modify the Start registry key. When you are finished, reattach the root volume to the original instance. For more information about detaching volumes, see Detaching an Amazon EBS Volume from an Instance (p. 781) .
RDS is enabled	Even if the service is started, it may be disabled. Detach the root volume from the instance, take a snapshot of the volume or create an AMI from it, attach the original volume to another instance in the same availability zone as a secondary volume, and enable the service by modifying the Terminal Server registry key as described in the following articles: <ul style="list-style-type: none">• Enable Remote desktop via the registry• Windows Server Hacks: Remotely Enable Remote Desktop When you are finished, reattach the root volume to the original instance. For more information about detaching volumes, see Detaching an Amazon EBS Volume from an Instance (p. 781) .

High CPU

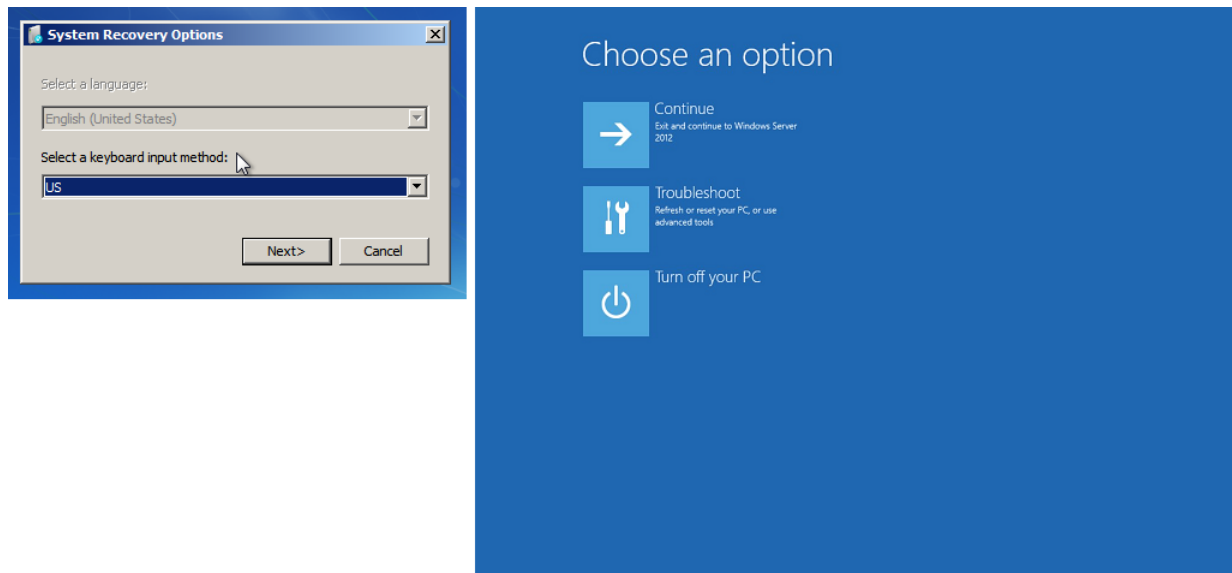
Check the **CPUUtilization (Maximum)** metric on your instance by using Amazon CloudWatch. If **CPUUtilization (Maximum)** is a high number, wait for the CPU to go down and try connecting again. High CPU usage can be caused by:

- Windows Update
- Security Software Scan
- Custom Startup Script
- Task Scheduler

For more information about the **CPUUtilization (Maximum)** metric, see [Get Statistics for a Specific EC2 Instance](#) in the *Amazon CloudWatch User Guide*. For additional troubleshooting tips, see [High CPU usage shortly after Windows starts](#) (p. 950).

Recovery Console Screen

Console Screenshot Service returned the following.



The operating system may boot into the Recovery console and get stuck in this state if the `bootstatuspolicy` is not set to `ignoreallfailures`. Use the following procedure to change the `bootstatuspolicy` configuration to `ignoreallfailures`.

Note

By default, the policy configuration for AWS-provided public Windows AMIs is set to `ignoreallfailures`.

1. Stop the unreachable instance.
2. Create a snapshot of the root volume. The root volume is attached to the instance as `/dev/sda1`.

Detach the root volume from the unreachable instance, take a snapshot of the volume or create an AMI from it, and attach it to another instance in the same Availability Zone as a secondary volume. For more information, see [Detaching an Amazon EBS Volume from an Instance](#) (p. 781).

Warning

If your temporary instance is based on the same AMI that the original instance is based on, and the operating system is later than Windows Server 2003, you must complete additional steps or you won't be able to boot the original instance after you restore its root volume because of a disk signature collision. Alternatively, select a different AMI for the temporary instance. For example, if the original instance uses the AWS Windows AMI for Windows Server 2008 R2, launch the temporary instance using the AWS Windows AMI for Windows Server 2012 or Windows Server 2003. (To find an AMI for Windows Server 2003, search for an AMI using the name `Windows_Server-2003-R2_SP2`.) If you must create a temporary instance based on the same AMI, see Step 6 in [Remote Desktop can't connect to the remote computer](#) (p. 951) for the additional steps you must complete to avoid a disk signature collision.

3. Log in to the instance and execute the following command from a command prompt to change the `bootstatuspolicy` configuration to `ignoreallfailures`:

```
bcdedit /store <Drive Letter>:\boot\bcd /set {default} bootstatuspolicy ignoreallfailures
```

4. Reattach the volume to the unreachable instance and start the instance again.

Windows Boot Manager Screen

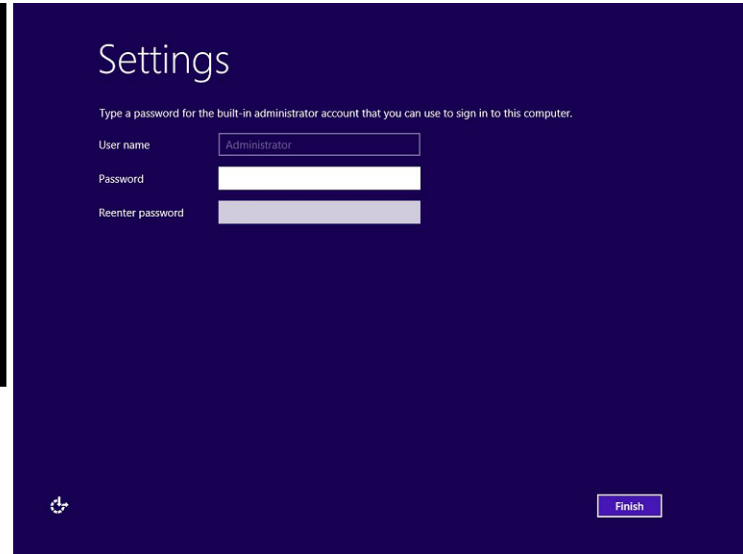
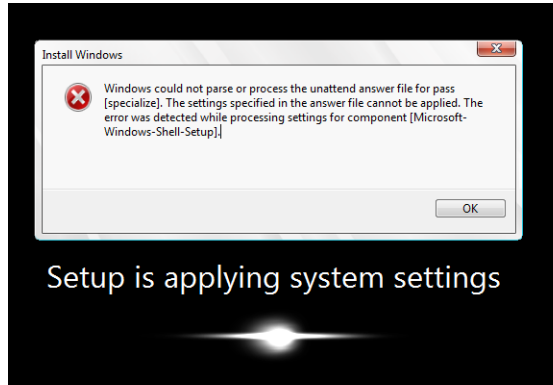
Console Screenshot Service returned the following.



The operating system experienced a fatal corruption in the system file and/or the registry. When the instance is stuck in this state, you should recover the instance from a recent backup AMI or launch a replacement instance. If you need to access data on the instance, detach any root volumes from the unreachable instance, take a snapshot of those volume or create an AMI from them, and attach them to another instance in the same Availability Zone as a secondary volume. For more information, see [Detaching an Amazon EBS Volume from an Instance \(p. 781\)](#).

Sysprep Screen

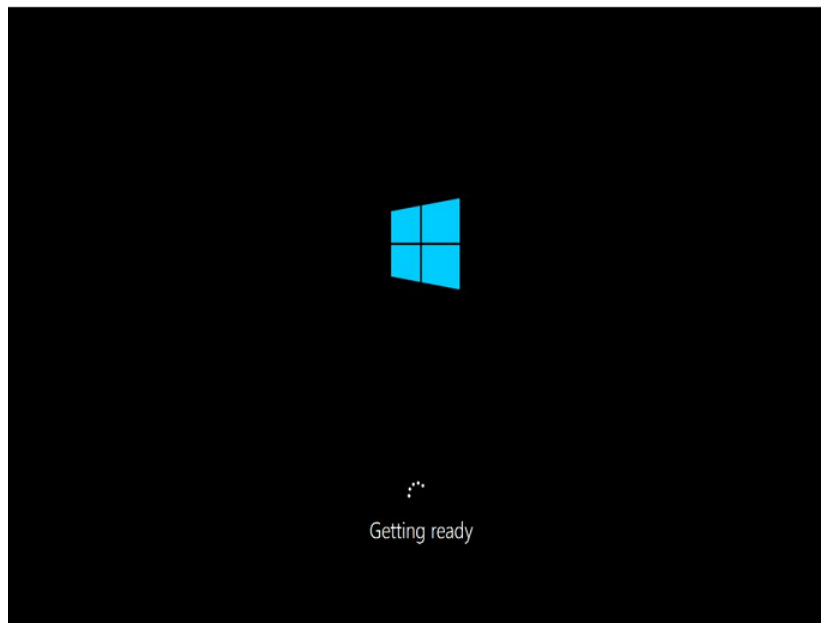
Console Screenshot Service returned the following.



You may see this screen if you did not use the EC2Config Service to call sysprep.exe or if the operating system failed while running Sysprep. To solve this problem, [Create a Standard Amazon Machine Image Using Sysprep \(p. 111\)](#).

Getting Ready Screen

Console Screenshot Service returned the following.



Refresh the Instance Console Screenshot Service repeatedly to verify that the progress ring is spinning. If the ring is spinning, wait for the operating system to start up. You can also check the **CPU Utilization (Maximum)** metric on your instance by using Amazon CloudWatch to see if the operating system is active. If the progress ring is not spinning, the instance may be stuck at the boot process. Reboot the instance. If rebooting does not solve the problem, recover the instance from a recent backup AMI or launch a replacement instance. If you need to access data on the instance, detach the root volume from the unreachable instance, take a snapshot of the volume or create an AMI from it. Then attach it to another instance in the same Availability Zone as a secondary volume. For

EBS volumes don't initialize on Windows Server 2016 AMIs

Instances created from Windows Server 2012 R2 and earlier Amazon Machine Images (AMIs) use the EC2Config service for a variety of startup tasks, including initializing EBS volumes. To accommodate the change from .NET Framework to .NET Core, the EC2Config service has been deprecated on Windows Server 2016 AMIs and replaced by EC2Launch. EC2Launch is a bundle of Windows PowerShell scripts that perform many of the tasks performed by the EC2Config service. By default, EC2Launch does not initialize secondary volumes. You can configure EC2Launch to initialize disks automatically by either scheduling the script to run or by calling EC2Launch in user data.

To map drive letters to volumes

1. On the instance you want to configure, open the following file in a simple text editor.

C:\ProgramData\Amazon\EC2-Windows\Launch\Config\DriveLetterMapping.json

2. Specify the volume settings as in the following example:

```
{
  "driveLetterMapping": [
    {
      "volumeName": "Temporary Storage 0",
      "driveLetter": "H"
    }
  ]
}
```

3. Save your changes.
4. In Windows PowerShell, run the following command so that the system schedules the script to run as a Windows Scheduled Task.

```
C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeDisks.ps1
```

The script will execute once when the instance boots.

Note

You can also initialize attached disks at the instance launch by adding the following path to the PowerShell script in Amazon EC2 userdata.

```
<powershell>
C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeDisks.ps1
</powershell>
```

For more information about EC2Launch, see [Configuring a Windows Instance Using EC2Launch](#) (p. 319).

Boot an EC2 Windows Instance into Directory Services Restore Mode (DSRM)

If an instance running Microsoft Active Directory experiences a system failure or other critical issues you can troubleshoot the instance by booting into a special version of Safe Mode called *Directory Services Restore Mode* (DSRM). In DSRM you can repair or recover Active Directory.

Driver Support for DSRM

How you enable DSRM and boot into the instance depends on the drivers the instance is running. In the EC2 console you can view driver version details for an instance from the System Log. The following tables shows which drivers are supported for DSRM.

Driver Versions	DSRM Supported?	Next Steps
Citrix PV 5.9	No	Restore the instance from a backup. You cannot enable DSRM.
AWS PV 7.2.0	No	Though DSRM is not supported for this driver, you can still detach the root volume from the instance, take a snapshot of the volume or create an AMI from it, and attach it to another instance in the same availability zone as a secondary volume. You can then enable DSRM (as described in this section).
AWS PV 7.2.2 and later	Yes	Detach the root volume, attach it to another instance, and enable DSRM (as described in this section).
Enhanced Networking (Intel 82599 Virtual Function)	Yes	Detach the root volume, attach it to another instance, and enable DSRM (as described in this section).

Note

By default, Enhanced Networking is enabled on the following Windows Server 2012 R2 instance types:

- C3
- C4
- D2
- I2
- R3

For more information about instance types, see [Amazon EC2 Instances](#). For information about how to enable Enhanced Networking for other Windows Server instances, see [Enabling Enhanced Networking on Windows Instances in a VPC](#). For more information about upgrading AWS PV drivers, see [Upgrading PV Drivers on Your Windows AMI \(p. 356\)](#).

Configure an Instance to Boot into DSRM

EC2 Windows instances do not have network connectivity before the operating system is running. For this reason, you cannot press the F8 button on your keyboard to select a boot option. You must use one of the following procedures to boot an EC2 Windows Server instance into DSRM.

Boot an Online Instance into DSRM

If you suspect that Active Directory has been corrupted and the instance is still running, you can configure the instance to boot into DSRM using either the System Configuration dialog box or the command prompt. Choose one of the following methods. If your instance is not online (unavailable) see the next section:

To boot an online instance into DSRM using the System Configuration dialog box

1. In the **Run** dialog box type `msconfig` and select **Enter**.
2. Choose the **Boot** tab.
3. Under **Boot options** choose **Safe boot**.
4. Choose **Active Directory repair** and then choose **OK**. The system prompts you to reboot the server.

To boot an online instance into DSRM using the command prompt

1. Open a command prompt.
2. Type `bcdedit /set safeboot dsrepair` and select **Enter**.

Boot an Offline Instance into DSRM

If an instance is offline and unreachable you must detach the root volume and attach it to another instance to enable DSRM mode.

To boot an offline instance into DSRM

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Locate the affected instance. Open the context (right-click) menu for the instance, choose **Instance State**, and then choose **Stop**.
4. Choose **Launch Instance** and create a temporary instance in the same Availability Zone as the affected instance. Choose an instance type that uses a different version of Windows. For example, if your instance is Windows Server 2008 R1, then choose a Windows Server 2008 R2 instance.

Important

If you do not create the instance in the same Availability Zone as the affected instance you will not be able to attach the root volume of the affected instance to the new instance.

5. In the navigation pane, choose **Volumes**.
6. Locate the root volume of the affected instance. **Detach** the volume and **attach** it to the temporary instance you created earlier. Attach it with the default device name (xvdf).
7. Use Remote Desktop to connect to the temporary instance, and then use the Disk Management utility to [make the volume available for use](#).
8. Open a command prompt and run the following command. Replace *D* with the actual drive letter of the secondary volume you just attached:

```
bcdedit /store D:\Boot\BCD /set {default} safeboot dsrepair
```

9. In the Disk Management Utility, choose the drive you attached earlier, open the context (right-click) menu, and choose **Offline**.
10. In the EC2 console, detach the affected volume from the temporary instance and reattach it to your original instance with the device name `/dev/sda1`. You must specify this device name to designate the volume as a root volume.
11. **Start** the instance.

12. After the instance passes the health checks in the EC2 console, connect to the instance using Remote Desktop and verify that it boots into DSRM mode.

Note

Delete or stop the *temporary* instance you created in this procedure.

High CPU usage shortly after Windows starts

If Windows Update is set to **Check for updates but let me choose whether to download and install them** (the default instance setting) this check can consume anywhere from 50 - 99% of the CPU on the instance. If this CPU consumption causes problems for your applications, you can manually change Windows Update settings in **Control Panel** or you can use the following script in the Amazon EC2 user data field:

```
reg add "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\WindowsUpdate\Auto
Update" /v
        AUOptions /t REG_DWORD /d 3 /f net stop wuauserv net start
wuauserv
```

When you execute this script specify a value for /d. The default value is 3. Possible values include the following:

1. Never check for updates
2. Check for updates but let me choose whether to download and install them
3. Download updates but let me choose whether to install them
4. Install updates automatically

To modify the user data for a Amazon EBS-backed instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**, and select the instance.
3. Choose **Actions**, select **Instance State**, and then choose **Stop**.
4. In the confirmation dialog box, select **Yes, Stop**. It can take a few minutes for the instance to stop.
5. With the instance still selected, select **Actions**, select **Instance Settings**, and then choose **View/Change User Data**. Note that you can't change the user data if the instance is running, but you can view it.
6. In the **View/Change User Data** dialog box, update the user data, and then choose **Save**.

After you modify the user data for your instance, you can execute it. For more information, see [Executing Scripts with User Data \(p. 274\)](#).

No console output

For Windows instances, the instance console displays the output from the EC2Config service running on the instance. The output logs the status of tasks performed during the Windows boot process. If Windows boots successfully, the last message logged is `Windows is Ready to use`. Note that you can also display event log messages in the console, but this feature is not enabled by default. For more information, see [Ec2 Service Properties \(p. 287\)](#).

To get the console output for your instance using the Amazon EC2 console, select the instance, click **Actions**, select **Instance Settings**, and then click **Get System Log**. To get the console output

using the command line, use one of the following commands: [get-console-output](#) (AWS CLI) or [Get-EC2ConsoleOutput](#) (AWS Tools for Windows PowerShell).

If the console output is empty, it could indicate an issue with the EC2Config service, such as a misconfigured configuration file, or that Windows failed to boot properly. To fix the issue, download and install the latest version of EC2Config. For more information, see [Installing the Latest Version of EC2Config](#) (p. 295).

Instance terminates immediately

After you launch an instance, we recommend that you check its status to confirm that it goes from the `pending` status to the `running` status, and not the `terminated` status.

If the instance terminates immediately, you can use the Amazon EC2 console or command line to get information about the reason that the instance terminated.

To get the reason that an instance terminated using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, choose **Instances** to display the instance details.
3. Select your instance.
4. In the **Description** tab, locate the reason next to the label **State transition reason**. If the instance is still running, there's typically no reason listed. If you've explicitly stopped or terminated the instance, the reason is `User initiated shutdown`.

To get the reason that an instance terminated using the command line

Use the [describe-instances](#) command (AWS CLI) with the ID of the instance. Look for the `StateReason` element in the output.

Remote Desktop can't connect to the remote computer

Try the following to resolve issues related to connecting to your instance:

- Verify that you're using the correct public DNS hostname. (In the Amazon EC2 console, select the instance and check **Public DNS (IPv4)** in the details pane.) If your instance is in a VPC and you do not see a public DNS name, you must enable DNS hostnames. For more information, see [Using DNS with Your VPC](#) in the *Amazon VPC User Guide*.
- Verify that your instance has a public IPv4 address. If not, you can associate an Elastic IP address with your instance. For more information, see [Elastic IP Addresses](#) (p. 709).
- To connect to your instance using an IPv6 address, check that your local computer has an IPv6 address and is configured to use IPv6. If you launched an instance from a Windows Server 2008 SP2 AMI or earlier, your instance is not automatically configured to recognize an IPv6 address assigned to the instance. For more information, see [Configure IPv6 on Your Instances](#) in the *Amazon VPC User Guide*.
- Verify that your security group has a rule that allows RDP access. For more information, see [Create a Security Group](#) (p. 17).
- If you copied the password but get the error `Your credentials did not work`, try typing them manually when prompted. It's possible that you missed a character or got an extra whitespace character when you copied the password.
- Verify that the instance has passed status checks. For more information, see [Status Checks for Your Instances](#) (p. 567) and [Troubleshooting Instances with Failed Status Checks](#) (*Amazon EC2 User Guide for Linux Instances*).

- [EC2-VPC] Verify that the route table for the subnet has a route that sends all traffic destined outside the VPC to the Internet gateway for the VPC. For more information, see [Creating a Custom Route Table](#) (Internet Gateways) in the *Amazon VPC User Guide*.
- Verify that Windows Firewall, or other firewall software, is not blocking RDP traffic to the instance. We recommend that you disable Windows Firewall and control access to your instance using security group rules.

To disable Windows Firewall on a Windows instance that you can't connect to

1. Stop the affected instance and detach its root volume.
2. Launch a temporary instance in the same Availability Zone as the affected instance.

Warning

If your temporary instance is based on the same AMI that the original instance is based on, and the operating system is later than Windows Server 2003, you must complete additional steps or you won't be able to boot the original instance after you restore its root volume because of a disk signature collision. Alternatively, select a different AMI for the temporary instance. For example, if the original instance uses the AWS Windows AMI for Windows Server 2008 R2, launch the temporary instance using the AWS Windows AMI for Windows Server 2012 or Windows Server 2003. (To find an AMI for Windows Server 2003, search for an AMI using the name `Windows_Server-2003-R2_SP2`.)

3. Attach the root volume from the affected instance to this temporary instance. Connect to the temporary instance, open the **Disk Management** utility, and bring the drive online.
4. Open **Regedit** and select **HKEY_LOCAL_MACHINE**. From the **File** menu, choose **Load Hive**. Select the drive, open the file `Windows\System32\config\SYSTEM`, and specify a key name when prompted (you can use any name).
5. Select the key you just loaded and navigate to `ControlSet001\Services\SharedAccess\Parameters\FirewallPolicy`. For each key with a name of the form `xxxxProfile`, select the key and change `EnableFirewall` from 1 to 0. Select the key again, and from the **File** menu, choose **Unload Hive**.
6. (Optional) If your temporary instance is based on the same AMI that the original instance is based on, and the operating system is later than Windows Server 2003, you must complete the following steps or you won't be able to boot the original instance after you restore its root volume because of a disk signature collision.

Warning

The following procedure describes how to edit the Windows Registry using Registry Editor. If you are not familiar with the Registry or how to safely make changes using Registry Editor, read about the Registry on [Microsoft TechNet](#).

- a. Open a command prompt, type **regedit.exe**, and press Enter.
- b. In the **Registry Editor**, choose **HKEY_LOCAL_MACHINE** from the context menu (right-click), and then choose **Find**.
- c. Type **Windows Boot Manager** and then choose **Find Next**.
- d. Choose the key named `11000001`. This key is a sibling of the key you found in the previous step.
- e. In the right pane, choose `Element` and then choose **Modify** from the context menu (right-click).
- f. Locate the four-byte disk signature at offset `0x38` in the data. Reverse the bytes to create the disk signature, and write it down. For example, the disk signature represented by the following data is `E9EB3AA5`:

```
...  
0030 00 00 00 00 01 00 00 00  
0038 A5 3A EB E9 00 00 00 00
```



```
0040 00 00 00 00 00 00 00 00  
...
```

- g. In a Command Prompt window, run the following command to start Microsoft DiskPart.

```
C:\> diskpart
```

- h. Run the following DiskPart command to select the volume. (You can verify that the disk number is 1 using the **Disk Management** utility.)

```
DISKPART> select disk 1  
  
Disk 1 is now the selected disk.
```

- i. Run the following DiskPart command to get the disk signature.

```
DISKPART> uniqueid disk  
  
Disk ID: 0C764FA8
```

- j. If the disk signature shown in the previous step doesn't match the disk signature from BCD that you wrote down earlier, use the following DiskPart command to change the disk signature so that it matches:

```
DISKPART> uniqueid disk id=E9EB3AA5
```

7. Using the **Disk Management** utility, bring the drive offline.

Note

The drive is automatically offline if the temporary instance is running the same operating system as the affected instance, so you won't need to bring it offline manually.

8. Detach the volume from the temporary instance. You can terminate the temporary instance if you have no further use for it.
9. Restore the root volume of the affected instance by attaching it as `/dev/sda1`.
10. Start the instance.
- Verify that the password has not expired. If the password has expired, you can reset it. For more information, see [Resetting an Administrator Password that's Lost or Expired \(p. 371\)](#).
 - If you attempt to connect using a user account that you created on the instance and receive the error `The user cannot connect to the server due to insufficient access privileges`, verify that you granted the user the right to log on locally. For more information, see <http://technet.microsoft.com/en-us/library/ee957044.aspx>.
 - If you attempt more than the maximum allowed concurrent RDP sessions, your session is terminated with the message `Your Remote Desktop Services session has ended. Another user connected to the remote computer, so your connection was lost.` By default, you are allowed two concurrent RDP sessions to your instance.

RDP displays a black screen instead of the desktop

Try the following to resolve this issue:

- Check the console output for additional information. To get the console output for your instance using the Amazon EC2 console, select the instance, choose **Actions**, select **Instance Settings**, and then choose **Get System Log**.
- Verify that you are running the latest version of your RDP client.

- Try the default settings for the RDP client. For more information, see [Remote Session Environment](#) in the *Microsoft TechNet Library*.
- If you are using Remote Desktop Connection, try starting it with the `/admin` option as follows.

```
C:\> mstsc /v:instance /admin
```

- If the server is running a full-screen application, it might have stopped responding. Use Ctrl+Shift+Esc to start Windows Task Manager, and then close the application.
- If the server is over-utilized, it might have stopped responding. To monitor the instance using the Amazon EC2 console, select the instance and then select the **Monitoring** tab. If you need to change the instance type to a larger size, see [Resizing Your Instance](#) (p. 147).

Instance loses network connectivity or scheduled tasks don't run when expected

If you restart your instance and it loses network connectivity, it's possible that the instance has the wrong time.

By default, Windows instances use Coordinated Universal Time (UTC). If you set the time for your instance to a different time zone and then restart it, the time becomes offset and the instance temporarily loses its IP address. The instance regains network connectivity eventually, but this can take several hours. The amount of time that it takes for the instance to regain network connectivity depends on the difference between UTC and the other time zone.

This same time issue can also result in scheduled tasks not running when you expect them to. In this case, the scheduled tasks do not run when expected because the instance has the incorrect time.

To use a time zone other than UTC persistently, you must set the **RealTimeIsUniversal** registry key. Without this key, an instance uses UTC after you restart it.

Important

Windows Server 2003 doesn't support the **RealTimeIsUniversal** registry key. Therefore, the instance always uses UTC after a restart.

To resolve time issues that cause a loss of network connectivity

1. Ensure that you are running the recommended PV drivers. For more information, see [Upgrading PV Drivers on Your Windows AMI](#) (p. 356).
2. Verify that the following registry key exists and is set to 1: **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\TimeZoneInformation\RealTimeIsUniversal**

Insufficient Instance Capacity

If you get an `InsufficientInstanceCapacity` error when you try to launch an instance, AWS does not currently have enough available capacity to service your request.

Try the following:

- Wait a few minutes and then submit your request again; capacity can shift frequently.
- Submit a new request with a reduced number of instances. For example, if you're making a single request to launch 15 instances, try making 3 requests for 5 instances, or 15 requests for 1 instance instead.
- Submit a new request without specifying an Availability Zone.

- Submit a new request using a different instance type (which you can resize at a later stage). For more information, see [Resizing Your Instance](#) (p. 147).
- Try purchasing Reserved Instances. Reserved Instances are a long-term capacity reservation. For more information, see [Amazon EC2 Reserved Instances](#).

Instance Limit Exceeded

If you get an `InstanceLimitExceeded` error when you try to launch an instance, you have reached your concurrent running instance limit. For new AWS accounts, the default limit is 20. If you need additional running instances, complete the form at [Request to Increase Amazon EC2 Instance Limit](#).

Windows Server 2012 R2 not available on the network

For information about troubleshooting a Windows Server 2012 R2 instance that is not available on the network, see [Windows Server 2012 R2 loses network and storage connectivity after an instance reboot](#) (p. 363).

Common Messages

This section includes tips to help you troubleshoot issues based on common messages.

Topics

- ["Password is not available"](#) (p. 955)
- ["Password not available yet"](#) (p. 956)
- ["Cannot retrieve Windows password"](#) (p. 956)
- ["Waiting for the metadata service"](#) (p. 956)
- ["Unable to activate Windows"](#) (p. 959)
- ["Windows is not genuine \(0x80070005\)"](#) (p. 960)
- ["No Terminal Server License Servers available to provide a license"](#) (p. 960)

"Password is not available"

To connect to a Windows instance using Remote Desktop, you must specify an account and password. The accounts and passwords provided are based on the AMI that you used to launch the instance. You can either retrieve the auto-generated password for the Administrator account, or use the account and password that were in use in the original instance from which the AMI was created.

If your Windows instance isn't configured to generate a random password, you'll receive the following message when you retrieve the auto-generated password using the console:

```
Password is not available.  
The instance was launched from a custom AMI, or the default password has  
changed. A  
password cannot be retrieved for this instance. If you have forgotten your  
password, you can  
reset it using the Amazon EC2 configuration service. For more information,  
see Passwords for a
```

```
Windows Server instance.
```

Check the console output for the instance to see whether the AMI that you used to launch it was created with password generation disabled. If password generation is disabled, the console output contains the following:

```
Ec2SetPassword: Disabled
```

If password generation is disabled and you don't remember the password for the original instance, you can reset the password for this instance. For more information, see [Resetting an Administrator Password that's Lost or Expired](#) (p. 371).

"Password not available yet"

To connect to a Windows instance using Remote Desktop, you must specify an account and password. The accounts and passwords provided are based on the AMI that you used to launch the instance. You can either retrieve the auto-generated password for the Administrator account, or use the account and password that were in use in the original instance from which the AMI was created.

Your password should be available within a few minutes. If the password isn't available, you'll receive the following message when you retrieve the auto-generated password using the console:

```
Password not available yet.  
Please wait at least 4 minutes after launching an instance before trying to  
retrieve the  
auto-generated password.
```

If it's been longer than four minutes and you still can't get the password, it's possible that EC2Config is disabled. Verify by checking whether the console output is empty. For more information, see [No console output](#) (p. 950).

Also verify that the AWS Identity and Access Management (IAM) account being used to access the Management Portal has the `ec2:GetPasswordData` action allowed. For more information about IAM permissions, see [What is IAM?](#)

"Cannot retrieve Windows password"

To retrieve the auto-generated password for the Administrator account, you must use the private key for the key pair that you specified when you launched the instance. If you didn't specify a key pair when you launched the instance, you'll receive the following message.

```
Cannot retrieve Windows password
```

You can terminate this instance and launch a new instance using the same AMI, making sure to specify a key pair.

"Waiting for the metadata service"

A Windows instance must obtain information from its instance metadata before it can activate itself. By default, the `WaitForMetadataAvailable` setting ensures that the EC2Config service waits for the instance metadata to be accessible before continuing with the boot process. For more information, see [Instance Metadata and User Data](#) (p. 271).

If the instance is failing the instance reachability test, try the following to resolve this issue.

- [EC2-VPC] Check the CIDR block for your VPC. A Windows instance cannot boot correctly if it's launched into a VPC that has an IP address range from 224.0.0.0 to 255.255.255.255 (Class D and Class E IP address ranges). These IP address ranges are reserved, and should not be assigned to host devices. We recommend that you create a VPC with a CIDR block from the private (non-publicly routable) IP address ranges as specified in [RFC 1918](#).
- It's possible that the system has been configured with a static IP address. Try the following:
 - [EC2-VPC] [Create a network interface \(p. 723\)](#) and [attach it to the instance \(p. 725\)](#).
 - [EC2-Classic] Enable DHCP.
- **To enable DHCP on a Windows instance that you can't connect to**
 1. Stop the affected instance and detach its root volume.
 2. Launch a temporary instance in the same Availability Zone as the affected instance.

Warning

If your temporary instance is based on the same AMI that the original instance is based on, and the operating system is later than Windows Server 2003, you must complete additional steps or you won't be able to boot the original instance after you restore its root volume because of a disk signature collision. Alternatively, select a different AMI for the temporary instance. For example, if the original instance uses the AWS Windows AMI for Windows Server 2008 R2, launch the temporary instance using the AWS Windows AMI for Windows Server 2012 or Windows Server 2003. (To find an AMI for Windows Server 2003, search for an AMI using the name `Windows_Server-2003-R2_SP2`.)

3. Attach the root volume from the affected instance to this temporary instance. Connect to the temporary instance, open the **Disk Management** utility, and bring the drive online.
4. From the temporary instance, open **Regedit** and select **HKEY_LOCAL_MACHINE**. From the **File** menu, choose **Load Hive**. Select the drive, open the file `Windows\System32\config\SYSTEM`, and specify a key name when prompted (you can use any name).
5. Select the key that you just loaded and navigate to `ControlSet001\Services\Tcpip\Parameters\Interfaces`. Each network interface is listed by a GUID. Select the correct network interface. If DHCP is disabled and a static IP address assigned, `EnableDHCP` is set to 0. To enable DHCP, set `EnableDHCP` to 1, and delete the following keys if they exist: `NameServer`, `SubnetMask`, `IPAddress`, and `DefaultGateway`. Select the key again, and from the **File** menu, choose **Unload Hive**.

Note

If you have multiple network interfaces, you'll need to identify the correct interface to enable DHCP. To identify the correct network interface, review the following key values `NameServer`, `SubnetMask`, `IPAddress`, and `DefaultGateway`. These values display the static configuration of the previous instance.

6. (Optional) If DHCP is already enabled, it's possible that you don't have a route to the metadata service. Updating EC2Config can resolve this issue.
 - a. [Download](#) and install the latest version of the EC2Config service. For more information about installing this service, see [Installing the Latest Version of EC2Config \(p. 295\)](#).
 - b. Extract the files from the `.zip` file to the `Temp` directory on the drive you attached.
 - c. Open **Regedit** and select **HKEY_LOCAL_MACHINE**. From the **File** menu, choose **Load Hive**. Select the drive, open the file `Windows\System32\config\SOFTWARE`, and specify a key name when prompted (you can use any name).
 - d. Select the key that you just loaded and navigate to `Microsoft\Windows\CurrentVersion`. Select the `RunOnce` key. (If this key doesn't exist, right-click `CurrentVersion`, point to **New**, select **Key**, and name the key `RunOnce`.) Right-click, point to **New**, and select **String Value**. Enter `Ec2Install` as the name and `C:\Temp\Ec2Install.exe -q` as the data.

- e. Select the key again, and from the **File** menu, choose **Unload Hive**.
7. (Optional) If your temporary instance is based on the same AMI that the original instance is based on, and the operating system is later than Windows Server 2003, you must complete the following steps or you won't be able to boot the original instance after you restore its root volume because of a disk signature collision.

Warning

The following procedure describes how to edit the Windows Registry using Registry Editor. If you are not familiar with the Registry or how to safely make changes using Registry Editor, read about the Registry on [Microsoft TechNet](#).

- a. Open a command prompt, type **regedit.exe**, and press Enter.
- b. In the **Registry Editor**, choose **HKEY_LOCAL_MACHINE** from the context menu (right-click), and then choose **Find**.
- c. Type **Windows Boot Manager** and then choose **Find Next**.
- d. Choose the key named 11000001. This key is a sibling of the key you found in the previous step.
- e. In the right pane, choose **Element** and then choose **Modify** from the context menu (right-click).
- f. Locate the four-byte disk signature at offset 0x38 in the data. Reverse the bytes to create the disk signature, and write it down. For example, the disk signature represented by the following data is E9EB3AA5:

```
...  
0030 00 00 00 00 01 00 00 00  
0038 A5 3A EB E9 00 00 00 00  
0040 00 00 00 00 00 00 00 00  
...
```

- g. In a Command Prompt window, run the following command to start Microsoft DiskPart.

```
C:\> diskpart
```

- h. Run the following DiskPart command to select the volume. (You can verify that the disk number is 1 using the **Disk Management** utility.)

```
DISKPART> select disk 1  
  
Disk 1 is now the selected disk.
```

- i. Run the following DiskPart command to get the disk signature.

```
DISKPART> uniqueid disk  
  
Disk ID: 0C764FA8
```

- j. If the disk signature shown in the previous step doesn't match the disk signature from BCD that you wrote down earlier, use the following DiskPart command to change the disk signature so that it matches:

```
DISKPART> uniqueid disk id=E9EB3AA5
```

8. Using the **Disk Management** utility, bring the drive offline.

Note

The drive is automatically offline if the temporary instance is running the same operating system as the affected instance, so you won't need to bring it offline manually.

9. Detach the volume from the temporary instance. You can terminate the temporary instance if you have no further use for it.
10. Restore the root volume of the affected instance by attaching the volume as `/dev/sda1`.
11. Start the affected instance.

If you are connected to the instance, open an Internet browser from the instance and enter the following URL for the metadata server:

```
http://169.254.169.254/latest/meta-data/
```

If you can't contact the metadata server, try the following to resolve the issue:

- [Download](#) and install the latest version of the EC2Config service. For more information about installing this service, see [Installing the Latest Version of EC2Config \(p. 295\)](#).
- Check whether the Windows instance is running RedHat PV drivers. If so, update to Citrix PV drivers. For more information, see [Upgrading PV Drivers on Your Windows AMI \(p. 356\)](#).
- Verify that the firewall, IPsec, and proxy settings do not block outgoing traffic to the metadata service (169.254.169.254) or the KMS servers (the addresses are specified in `TargetKMSserver` elements in `C:\Program Files\Amazon\Ec2ConfigService\Settings\ActivationSettings.xml`).
- Verify that you have a route to the metadata service (169.254.169.254) using the following command.

```
C:\> route print
```

- Check for network issues that might affect the Availability Zone for your instance. Go to <http://status.aws.amazon.com/>.

"Unable to activate Windows"

Windows instances use the AWS Key Management Service (AWS KMS) for activation. You can receive this message: `A problem occurred when Windows tried to activate. Error Code 0xC004F074`, if your instance can't reach the KMS server. Windows must be activated every 180 days. EC2Config attempts to contact the KMS server before the activation period expires to ensure that Windows remains activated.

If you encounter a Windows Activation issue, use the following procedure to resolve this issue.

1. [Download](#) and install the latest version of the EC2Config service. For more information about installing this service, see [Installing the Latest Version of EC2Config \(p. 295\)](#).
2. Log onto the instance and open the following file: `C:\Program Files\Amazon\Ec2ConfigService\Settings\config.xml`.
3. Locate the **Ec2WindowsActivate** plugin in the `config.xml` file. Change the state to **Enabled** and save your changes.
4. In the Windows Services snap-in, restart the EC2Config service or reboot the instance.

If this does not resolve the activation issue, follow these additional steps.

1. Set the KMS target: **C:\> slmgr.vbs /skms 169.254.169.250:1688**
2. Activate Windows: **C:\> slmgr.vbs /ato**

If you are still receiving an activation error, verify the following information.

- Verify that you have routes to the KMS servers. Open `C:\Program Files\Amazon\Ec2ConfigService\Settings\ActivationSettings.xml` and locate the `TargetKMSServer` elements. Run the following command and check whether the addresses for these KMS servers are listed.

```
C:\> route print
```

- Verify that the KMS client key is set. Run the following command and check the output.

```
C:\> C:\Windows\System32\slmgr.vbs /dlv
```

If the output contains `Error: product key not found`, the KMS client key isn't set. If the KMS client key isn't set, look up the client key as described in this Microsoft TechNet article: <http://technet.microsoft.com/en-us/library/jj612867.aspx>, and then run the following command to set the KMS client key.

```
C:\> C:\Windows\System32\slmgr.vbs /ipk client_key
```

- Verify that the system has the correct time and time zone. If you are using Windows Server 2008 or later and a time zone other than UTC, add the following registry key and set it to 1 to ensure that the time is correct: **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\TimeZoneInformation\RealTimeIsUniversal**.
- If Windows Firewall is enabled, temporarily disable it using the following command.

```
C:\> netsh advfirewall set allprofiles state off
```

"Windows is not genuine (0x80070005)"

Windows instances use KMS for activation. If an instance is unable to complete the activation process, it reports that the copy of Windows is not genuine.

Try the suggestions for "[Unable to activate Windows](#)" (p. 959).

"No Terminal Server License Servers available to provide a license"

By default, Windows Server is licensed for two simultaneous users through Remote Desktop. If you need to provide more than two users with simultaneous access to your Windows instance through Remote Desktop, you can purchase a Remote Desktop Services client access license (CAL) and install the Remote Desktop Session Host and Remote Desktop Licensing Server roles.

Check for the following issues:

- You've exceeded the maximum number of concurrent RDP sessions.
- You've installed the Windows Remote Desktop Services role.
- Licensing has expired. If the licensing has expired, you can't connect to your Windows instance as a user. You can try the following:

Amazon Elastic Compute Cloud
User Guide for Windows Instances
"No Terminal Server License Servers
available to provide a license"

- Connect to the instance from the command line using an `/admin` parameter, for example:

```
C:\> mstsc /v:instance /admin
```

For more information, go to the following Microsoft article: [Use command line parameters with Remote Desktop Connection](#).

- Stop the instance, detach its Amazon EBS volumes, and attach them to another instance in the same Availability Zone to recover your data.

Document History

The following table describes important additions to the Amazon EC2 documentation. We also update the documentation frequently to address the feedback that you send us.

Current API version: 2016-11-15.

Feature	API Version	Description	Release Date
IPv6 support	2016-11-15	You can associate an IPv6 CIDR with your VPC and subnets, and assign IPv6 addresses to instances in your VPC. For more information, see Amazon EC2 Instance IP Addressing (p. 693) .	1 December 2016
R4 instances	2016-09-15	R4 instances represents the next generation of memory optimized instances. R4 instances are well-suited for memory-intensive, latency-sensitive workloads such as business intelligence (BI), data mining and analysis, in-memory databases, distributed web scale in-memory caching, and application performance real-time processing of unstructured big data. For more information, see Memory Optimized Instances (p. 124)	30 November 2016
New t2.xlarge and t2.2xlarge instance types	2016-09-15	T2 instances are designed to provide moderate base performance and the capability to burst to significantly higher performance as required by your workload. They are intended for applications that need responsiveness, high performance for limited periods of time, and a low cost. For more information, see T2 Instances (p. 121) .	30 November 2016
P2 instances	2016-09-15	P2 instances use NVIDIA Tesla K80 GPUs and are designed for general purpose GPU computing using the CUDA or OpenCL programming models. For more information, see Windows Accelerated Computing Instances (p. 127) .	29 September 2016
m4.16xlarge instances	2016-04-01	Expands the range of the general-purpose M4 family with the introduction of m4.16xlarge instances, with 64 vCPUs and 256 GiB of RAM.	6 September 2016

Feature	API Version	Description	Release Date
Automatic scaling for Spot fleet		You can now set up scaling policies for your Spot fleet. For more information, see Automatic Scaling for Spot Fleet (p. 215) .	1 September 2016
Run Command support for managed instances	2016-04-01	Amazon EC2 Run Command now supports the management of on-premises servers and virtual machines (VMs) and VMs from other cloud providers. For more information, see Setting Up Systems Manager in Hybrid Environments (p. 397) .	30 June 2016
Elastic Network Adapter (ENA)	2016-04-01	You can now use ENA for enhanced networking. For more information, see Enhanced Networking Types (p. 737) .	28 June 2016
Enhanced support for viewing and modifying longer IDs	2016-04-01	You can now view and modify longer ID settings for other IAM users, IAM roles, or the root user. For more information, see Resource IDs (p. 852) .	23 June 2016
Copy encrypted Amazon EBS snapshots between AWS accounts	2016-04-01	You can now copy encrypted EBS snapshots between AWS accounts. For more information, see Copying an Amazon EBS Snapshot (p. 791) .	21 June 2016
Capture a screenshot of an instance console	2015-10-01	You can now obtain additional information when debugging instances that are unreachable. For more information, see Troubleshoot an Unreachable Instance (p. 939) .	24 May 2016
X1 instances	2015-10-01	Memory-optimized instances designed for running in-memory databases, big data processing engines, and high performance computing (HPC) applications. For more information, see Memory Optimized Instances (p. 124) .	18 May 2016
Two new EBS volume types	2015-10-01	You can now create Throughput Optimized HDD (st1) and Cold HDD (sc1) volumes. For more information, see Amazon EBS Volume Types (p. 749) .	19 April 2016
Added new NetworkPacketsIn and NetworkPacketsOut metrics for Amazon EC2		Added new NetworkPacketsIn and NetworkPacketsOut metrics for Amazon EC2. For more information, see Instance Metrics (p. 577) .	23 March 2016
CloudWatch metrics for Spot fleet		You can now get CloudWatch metrics for your Spot fleet. For more information, see CloudWatch Metrics for Spot Fleet (p. 213) .	21 March 2016

Feature	API Version	Description	Release Date
Scheduled Instances	2015-10-01	Scheduled Reserved Instances (Scheduled Instances) enable you to purchase capacity reservations that recur on a daily, weekly, or monthly basis, with a specified start time and duration. For more information, see Scheduled Reserved Instances (p. 176) .	13 January 2016
Longer resource IDs	2015-10-01	We're gradually introducing longer length IDs for some Amazon EC2 and Amazon EBS resource types. During the opt-in period, you can enable the longer ID format for supported resource types. For more information, see Resource IDs (p. 852) .	13 January 2016
ClassicLink DNS support	2015-10-01	You can enable ClassicLink DNS support for your VPC so that DNS hostnames that are addressed between linked EC2-Classic instances and instances in the VPC resolve to private IP addresses and not public IP addresses. For more information, see Enabling ClassicLink DNS Support (p. 679) .	11 January 2016
New <code>t2.nano</code> instance type	2015-10-01	T2 instances are designed to provide moderate base performance and the capability to burst to significantly higher performance as required by your workload. They are intended for applications that need responsiveness, high performance for limited periods of time, and a low cost. For more information, see T2 Instances (p. 121) .	15 December 2015
Dedicated hosts	2015-10-01	An Amazon EC2 Dedicated host is a physical server with instance capacity dedicated for your use. For more information, see Dedicated Hosts (p. 226) .	23 November 2015
Spot instance duration	2015-10-01	You can now specify a duration for your Spot instances. For more information, see Specifying a Duration for Your Spot Instances (p. 192) .	6 October 2015
Spot fleet modify request	2015-10-01	You can now modify the target capacity of your Spot fleet request. For more information, see Modifying a Spot Fleet Request (p. 203) .	29 September 2015
Spot fleet diversified allocation strategy	2015-04-15	You can now allocate Spot instances in multiple Spot pools using a single Spot fleet request. For more information, see Spot Fleet Allocation Strategy (p. 185) .	15 September 2015
Spot fleet instance weighting	2015-04-15	You can now define the capacity units that each instance type contributes to your application's performance, and adjust your bid price for each Spot pool accordingly. For more information, see Spot Fleet Instance Weighting (p. 186) .	31 August 2015

Feature	API Version	Description	Release Date
New reboot alarm action and new IAM role for use with alarm actions		Added the reboot alarm action and new IAM role for use with alarm actions. For more information, see Create Alarms That Stop, Terminate, Reboot, or Recover an Instance (p. 592).	23 July 2015
New <code>t2.large</code> instance type		T2 instances are designed to provide moderate base performance and the capability to burst to significantly higher performance as required by your workload. They are intended for applications that need responsiveness, high performance for limited periods of time, and a low cost. For more information, see T2 Instances (p. 121).	16 June 2015
M4 instances		The next generation of general-purpose instances that provide a balance of compute, memory, and network resources. M4 instances are powered by a custom Intel 2.4 GHz Intel® Xeon® E5 2676v3 (Haswell) processor with AVX2.	11 June 2015
Spot fleets	2015-04-15	You can manage a collection, or fleet, of Spot instances instead of managing separate Spot instance requests. For more information, see How Spot Fleet Works (p. 185).	18 May 2015
Migrate Elastic IP addresses to EC2-Classic	2015-04-15	You can migrate an Elastic IP address that you've allocated for use in the EC2-Classic platform to the EC2-VPC platform. For more information, see Migrating an Elastic IP Address from EC2-Classic to EC2-VPC (p. 712).	15 May 2015
Importing VMs with multiple disks as AMIs	2015-03-01	The VM Import process now supports importing VMs with multiple disks as AMIs. For more information, see Importing a VM as an Image Using VM Import/Export in the <i>VM Import/Export User Guide</i> .	23 April 2015
New <code>g2.8xlarge</code> instance type		The new <code>g2.8xlarge</code> instance is backed by four high-performance NVIDIA GPUs, making it well suited for GPU compute workloads including large scale rendering, transcoding, machine learning, and other server-side workloads that require massive parallel processing power.	7 April 2015

Feature	API Version	Description	Release Date
D2 instances		<p>Next generation Amazon EC2 dense-storage instances that are optimized for applications requiring sequential access to large amount of data on direct attached instance storage. D2 instances are designed to offer best price/performance in the dense-storage family. Powered by 2.4 GHz Intel® Xeon® E5 2676v3 (Haswell) processors, D2 instances improve on HS1 instances by providing additional compute power, more memory, and Enhanced Networking. In addition, D2 instances are available in four instance sizes with 6TB, 12TB, 24TB, and 48TB storage options.</p> <p>For more information, see D2 Instances (p. 133).</p>	24 March 2015
Amazon EC2 Simple Systems Manager (SSM)		<p>SSM enables you to configure and manage your EC2 instances. For more information, see Managing Windows Instance Configuration (p. 326) and Joining a Windows Instance to an AWS Directory Service Domain (p. 331).</p>	17 February 2015
AWS Systems Manager for Microsoft SCVMM 1.5		<p>You can now use AWS Systems Manager for Microsoft SCVMM to launch an instance and to import a VM from SCVMM to Amazon EC2. For more information, see Creating an EC2 Instance (p. 887) and Importing Your Virtual Machine (p. 892).</p>	21 January 2015
Automatic recovery for EC2 instances		<p>You can create an Amazon CloudWatch alarm that monitors an Amazon EC2 instance and automatically recovers the instance if it becomes impaired due to an underlying hardware failure or a problem that requires AWS involvement to repair. A recovered instance is identical to the original instance, including the instance ID, IP addresses, and all instance metadata.</p> <p>For more information, see Recover Your Instance (p. 269).</p>	12 January 2015

Feature	API Version	Description	Release Date
C4 instances		<p>Next-generation compute-optimized instances that provide very high CPU performance at an economical price. C4 instances are based on custom 2.9 GHz Intel® Xeon® E5-2666 v3 (Haswell) processors. With additional Turbo boost, the processor clock speed in C4 instances can reach as high as 3.5GHz with 1 or 2 core turbo. Expanding on the capabilities of C3 compute-optimized instances, C4 instances offer customers the highest processor performance among EC2 instances. These instances are ideally suited for high-traffic web applications, ad serving, batch processing, video encoding, distributed analytics, high-energy physics, genome analysis, and computational fluid dynamics.</p> <p>For more information, see C4 Instances (p. 129).</p>	11 January 2015
ClassicLink	2014-10-01	<p>ClassicLink enables you to link your EC2-Classic instance to a VPC in your account. You can associate VPC security groups with the EC2-Classic instance, enabling communication between your EC2-Classic instance and instances in your VPC using private IP addresses. For more information, see ClassicLink (p. 673).</p>	7 January 2015
Spot instance termination notices		<p>The best way to protect against Spot instance interruption is to architect your application to be fault tolerant. In addition, you can take advantage of Spot instance termination notices, which provide a two-minute warning before Amazon EC2 must terminate your Spot instance.</p> <p>For more information, see Spot Instance Termination Notices (p. 222).</p>	5 January 2015
AWS Systems Manager for Microsoft SCVMM		<p>AWS Systems Manager for Microsoft SCVMM provides a simple, easy-to-use interface for managing AWS resources, such as EC2 instances, from Microsoft SCVMM. For more information, see AWS Systems Manager for Microsoft System Center VMM (p. 882).</p>	29 October 2014
DescribeVolumes pagination support	2014-09-01	<p>The <code>DescribeVolumes</code> API call now supports the pagination of results with the <code>MaxResults</code> and <code>NextToken</code> parameters. For more information, see DescribeVolumes in the <i>Amazon EC2 API Reference</i>.</p>	23 October 2014

Feature	API Version	Description	Release Date
Added support for Amazon CloudWatch Logs		You can use Amazon CloudWatch Logs to monitor, store, and access your system, application, and custom log files from your instances or other sources. You can then retrieve the associated log data from CloudWatch Logs using the Amazon CloudWatch console, the CloudWatch Logs commands in the AWS CLI, or the CloudWatch Logs SDK. For more information, see Configuring a Windows Instance Using the EC2Config Service (p. 283) . For more information about CloudWatch Logs, see Monitoring System, Application, and Custom Log Files in the Amazon CloudWatch User Guide.	10 July 2014
T2 instances	2014-06-15	T2 instances are designed to provide moderate base performance and the capability to burst to significantly higher performance as required by your workload. They are intended for applications that need responsiveness, high performance for limited periods of time, and a low cost. For more information, see T2 Instances (p. 121) .	30 June 2014
New EC2 Service Limits page		Use the EC2 Service Limits page in the Amazon EC2 console to view the current limits for resources provided by Amazon EC2 and Amazon VPC, on a per-region basis.	19 June 2014
Amazon EBS General Purpose SSD Volumes	2014-05-01	General Purpose SSD volumes offer cost-effective storage that is ideal for a broad range of workloads. These volumes deliver single-digit millisecond latencies, the ability to burst to 3,000 IOPS for extended periods of time, and a base performance of 3 IOPS/GiB. General Purpose SSD volumes can range in size from 1 GiB to 1 TiB. For more information, see General Purpose SSD (gp2) Volumes (p. 752) .	16 June 2014
Windows Server 2012 R2		AMIs for Windows Server 2012 R2 use the new AWS PV drivers. For more information, see AWS PV Drivers (p. 353) .	3 June 2014
AWS Management Pack		AWS Management Pack now supports for System Center Operations Manager 2012 R2. For more information, see AWS Management Pack for Microsoft System Center (p. 896) .	22 May 2014

Feature	API Version	Description	Release Date
Amazon EBS encryption	2014-05-01	Amazon EBS encryption offers seamless encryption of EBS data volumes and snapshots, eliminating the need to build and maintain a secure key management infrastructure. EBS encryption enables data at rest security by encrypting your data using Amazon-managed keys. The encryption occurs on the servers that host EC2 instances, providing encryption of data as it moves between EC2 instances and EBS storage. For more information, see Amazon EBS Encryption (p. 799) .	21 May 2014
R3 instances	2014-02-01	Next generation memory-optimized instances with the best price point per GiB of RAM and high performance. These instances are ideally suited for relational and NoSQL databases, in-memory analytics solutions, scientific computing, and other memory-intensive applications that can benefit from the high memory per vCPU, high compute performance, and enhanced networking capabilities of R3 instances. For more information about the hardware specifications for each Amazon EC2 instance type, see Amazon EC2 Instances .	9 April 2014
Amazon EC2 Usage Reports		Amazon EC2 Usage Reports is a set of reports that shows cost and usage data of your usage of EC2. For more information, see Amazon EC2 Usage Reports (p. 870) .	28 January 2014
Additional M3 instances	2013-10-15	The M3 instance sizes <code>m3.medium</code> and <code>m3.large</code> are now supported. For more information about the hardware specifications for each Amazon EC2 instance type, see Amazon EC2 Instances .	20 January 2014
I2 instances	2013-10-15	These instances provide very high IOPS. I2 instances also support enhanced networking that delivers improved inter-instance latencies, lower network jitter, and significantly higher packet per second (PPS) performance. For more information, see I2 Instances (p. 131) .	19 December 2013
Updated M3 instances	2013-10-15	The M3 instance sizes, <code>m3.xlarge</code> and <code>m3.2xlarge</code> now support instance store with SSD volumes. For more information about the hardware specifications for each Amazon EC2 instance type, see Amazon EC2 Instances .	19 December 2013
Resource-level permissions for RunInstances	2013-10-15	You can now create policies in AWS Identity and Access Management to control resource-level permissions for the Amazon EC2 RunInstances API action. For more information and example policies, see Controlling Access to Amazon EC2 Resources (p. 619) .	20 November 2013

Feature	API Version	Description	Release Date
C3 instances	2013-10-15	<p>Compute-optimized instances that provide very high CPU performance at an economical price. C3 instances also support enhanced networking that delivers improved inter-instance latencies, lower network jitter, and significantly higher packet per second (PPS) performance. These instances are ideally suited for high-traffic web applications, ad serving, batch processing, video encoding, distributed analytics, high-energy physics, genome analysis, and computational fluid dynamics.</p> <p>For more information about the hardware specifications for each Amazon EC2 instance type, see Amazon EC2 Instances.</p>	14 November 2013
Launching an instance from the AWS Marketplace		You can now launch an instance from the AWS Marketplace using the Amazon EC2 launch wizard. For more information, see Launching an AWS Marketplace Instance (p. 251) .	11 November 2013
G2 instances	2013-10-01	These instances are ideally suited for video creation services, 3D visualizations, streaming graphics-intensive applications, and other server-side workloads requiring massive parallel processing power. For more information, see Windows Accelerated Computing Instances (p. 127) .	4 November 2013
New launch wizard		There is a new and redesigned EC2 launch wizard. For more information, see Launching an Instance (p. 244) .	10 October 2013
Modifying Amazon EC2 Reserved Instances	2013-08-15	You can now modify Reserved Instances in a region.	11 September 2013
Assigning a public IP address	2013-07-15	You can now assign a public IP address when you launch an instance in a VPC. For more information, see Assigning a Public IPv4 Address During Instance Launch (p. 700) .	20 August 2013
Granting resource-level permissions	2013-06-15	Amazon EC2 supports new Amazon Resource Names (ARNs) and condition keys. For more information, see IAM Policies for Amazon EC2 (p. 621) .	8 July 2013
Incremental Snapshot Copies	2013-02-01	You can now perform incremental snapshot copies. For more information, see Copying an Amazon EBS Snapshot (p. 791) .	11 June 2013

Feature	API Version	Description	Release Date
AWS Management Pack		The AWS Management Pack links Amazon EC2 instances and the Windows or Linux operating systems running inside them. The AWS Management Pack is an extension to Microsoft System Center Operations Manager. For more information, see AWS Management Pack for Microsoft System Center (p. 896) .	8 May 2013
New Tags page		There is a new Tags page in the Amazon EC2 console. For more information, see Tagging Your Amazon EC2 Resources (p. 859) .	04 April 2013
Additional EBS-optimized instance types	2013-02-01	The following instance types can now be launched as EBS-optimized instances: c1.xlarge, m2.2xlarge, m3.xlarge, and m3.2xlarge. For more information, see Amazon EBS-Optimized Instances (p. 795) .	19 March 2013
PV Drivers		To learn how to upgrade the paravirtualized (PV) drivers on your Windows AMI, see Upgrading PV Drivers on Your Windows AMI (p. 356) .	March 2013
AWS Diagnostics for Windows Server		The topic AWS Diagnostics for Windows Server - Beta (p. 934) describes how to diagnose and troubleshoot possible issues using the AWS Diagnostics for Windows Server.	March 2013
Copy an AMI from one region to another	2013-02-01	You can copy an AMI from one region to another, enabling you to launch consistent instances in more than one AWS region quickly and easily. For more information, see Copying an AMI (p. 87) .	11 March 2013
Launch instances into a default VPC	2013-02-01	Your AWS account is capable of launching instances into either the EC2-Classic or EC2-VPC platform, or only into the EC2-VPC platform, on a region-by-region basis. If you can launch instances only into EC2-VPC, we create a default VPC for you. When you launch an instance, we launch it into your default VPC, unless you create a nondefault VPC and specify it when you launch the instance. For more information, see Supported Platforms (p. 672) .	11 March 2013
High-memory cluster (cr1.8xlarge) instance type	2012-12-01	Have large amounts of memory coupled with high CPU and network performance. These instances are well suited for in-memory analytics, graph analysis, and scientific computing applications.	21 January 2013

Feature	API Version	Description	Release Date
High storage (hs1.8xlarge) instance type	2012-12-01	High storage instances provide very high storage density and high sequential read and write performance per instance. They are well-suited for data warehousing, Hadoop/MapReduce, and parallel file systems. For more information, see HS1 Instances (p. 135) .	20 December 2012
EBS snapshot copy	2012-12-01	You can use snapshot copies to create backups of data, to create new Amazon EBS volumes, or to create Amazon Machine Images (AMIs). For more information, see Copying an Amazon EBS Snapshot (p. 791) .	17 December 2012
Updated EBS metrics and status checks for Provisioned IOPS SSD volumes	2012-10-01	Updated the EBS metrics to include two new metrics for Provisioned IOPS SSD volumes. For more information, see Monitoring Volumes with CloudWatch (p. 769) . Also added new status checks for Provisioned IOPS SSD volumes. For more information, see Monitoring Volumes with Status Checks (p. 772) .	20 November 2012
Support for Windows Server 2012		<p>Amazon EC2 now provides you with several pre-configured Windows Server 2012 AMIs. These AMIs are immediately available for use in every region and for every 64-bit instance type. The AMIs support the following languages:</p> <ul style="list-style-type: none"> • English • Chinese Simplified • Chinese Traditional • Chinese Traditional Hong Kong • Japanese • Korean • Portuguese • Portuguese Brazil • Czech • Dutch • French • German • Hungarian • Italian • Polish • Russian • Spanish • Swedish • Turkish 	19 November 2012

Feature	API Version	Description	Release Date
M3 instances	2012-10-01	There are new M3 extra-large and M3 double-extra-large instance types. For more information about the hardware specifications for each Amazon EC2 instance type, see Amazon EC2 Instances .	31 October 2012
Spot instance request status	2012-10-01	Spot instance request status makes it easy to determine the state of your Spot requests.	14 October 2012
Amazon EC2 Reserved Instance Marketplace	2012-08-15	The Reserved Instance Marketplace matches sellers who have Amazon EC2 Reserved Instances that they no longer need with buyers who are looking to purchase additional capacity. Reserved Instances bought and sold through the Reserved Instance Marketplace work like any other Reserved Instances, except that they can have less than a full standard term remaining and can be sold at different prices.	11 September 2012
Provisioned IOPS SSD for Amazon EBS	2012-07-20	Provisioned IOPS SSD volumes deliver predictable, high performance for I/O intensive workloads, such as database applications, that rely on consistent and fast response times. For more information, see Amazon EBS Volume Types (p. 749) .	31 July 2012
High I/O instances for Amazon EC2	2012-06-15	High I/O instances provides very high, low latency, disk I/O performance using SSD-based local instance storage. For more information, see H1 Instances (p. 134) .	18 July 2012
IAM roles on Amazon EC2 instances	2012-06-01	IAM roles for Amazon EC2 provide: <ul style="list-style-type: none"> • AWS access keys for applications running on Amazon EC2 instances. • Automatic rotation of the AWS access keys on the Amazon EC2 instance. • Granular permissions for applications running on Amazon EC2 instances that make requests to your AWS services. 	11 June 2012
Spot instance features that make it easier to get started and handle the potential of interruption.		You can now manage your Spot instances as follows: <ul style="list-style-type: none"> • Place bids for Spot instances using Auto Scaling launch configurations, and set up a schedule for placing bids for Spot instances. For more information, see Launching Spot Instances in Your Auto Scaling Group in the <i>Auto Scaling User Guide</i>. • Get notifications when instances are launched or terminated. • Use AWS CloudFormation templates to launch Spot instances in a stack with AWS resources. 	7 June 2012

Feature	API Version	Description	Release Date
EC2 instance export and timestamps for status checks for Amazon EC2	2012-05-01	Added support for exporting Windows Server instances that you originally imported into EC2. Added support for timestamps on instance status and system status to indicate the date and time that a status check failed.	25 May 2012
EC2 instance export, and timestamps in instance and system status checks for Amazon VPC	2012-05-01	Added support for EC2 instance export to Citrix Xen, Microsoft Hyper-V, and VMware vSphere. Added support for timestamps in instance and system status checks.	25 May 2012
Cluster Compute Eight Extra Large instances	2012-04-01	Added support for <code>cc2.8xlarge</code> instances in a VPC.	26 April 2012
AWS Marketplace AMIs	2012-04-01	Added support for AWS Marketplace AMIs.	19 April 2012
Medium instances, support for 64-bit on all AMIs	2011-12-15	Added support for a new instance type and 64-bit information.	7 March 2012
Reserved Instance pricing tiers	2011-12-15	Added a new section discussing how to take advantage of the discount pricing that is built into the Reserved Instance pricing tiers.	5 March 2012
Elastic Network Interfaces (ENIs) for EC2 instances in Amazon Virtual Private Cloud	2011-12-01	Added new section about elastic network interfaces (ENIs) for EC2 instances in a VPC. For more information, see Elastic Network Interfaces (p. 716) .	21 December 2011
New offering types for Amazon EC2 Reserved Instances	2011-11-01	You can choose from a variety of Reserved Instance offerings that address your projected use of the instance.	01 December 2011
Amazon EC2 instance status	2011-11-01	You can view additional details about the status of your instances, including scheduled events planned by AWS that might have an impact on your instances. These operational activities include instance reboots required to apply software updates or security patches, or instance retirements required where there are hardware issues. For more information, see Monitoring the Status of Your Instances (p. 567) .	16 November 2011
Amazon EC2 Cluster Compute Instance Type		Added support for Cluster Compute Eight Extra Large (<code>cc2.8xlarge</code>) to Amazon EC2.	14 November 2011
Spot instances in Amazon VPC	2011-07-15	Added information about the support for Spot instances in Amazon VPC. With this update, users can launch Spot instances a virtual private cloud (VPC). By launching Spot instances in a VPC, users of Spot instances can enjoy the benefits of Amazon VPC.	11 October 2011

Feature	API Version	Description	Release Date
Simplified VM import process for users of the CLI tools	2011-07-15	The VM Import process is simplified with the enhanced functionality of <code>ImportInstance</code> and <code>ImportVolume</code> , which now will perform the upload of the images into Amazon EC2 after creating the import task. In addition, with the introduction of <code>ResumeImport</code> , users can restart an incomplete upload at the point the task stopped.	15 September 2011
Support for importing in VHD file format		VM Import can now import virtual machine image files in VHD format. The VHD file format is compatible with the Citrix Xen and Microsoft Hyper-V virtualization platforms. With this release, VM Import now supports RAW, VHD and VMDK (VMware ESX-compatible) image formats. For more information, see the VM Import/Export User Guide .	24 August 2011
Support for Windows Server 2003 R2		VM Import now supports Windows Server 2003 (R2). With this release, VM Import supports all versions of Windows Server supported by Amazon EC2.	24 August 2011
Update to the Amazon EC2 VM Import Connector for VMware vCenter		Added information about the 1.1 version of the Amazon EC2 VM Import Connector for VMware vCenter virtual appliance (Connector). This update includes proxy support for Internet access, better error handling, improved task progress bar accuracy, and several bug fixes.	27 June 2011
Spot instances Availability Zone pricing changes	2011-05-15	Added information about the Spot instances Availability Zone pricing feature. In this release, we've added new Availability Zone pricing options as part of the information returned when you query for Spot instance requests and Spot price history. These additions make it easier to determine the price required to launch a Spot instance into a particular Availability Zone.	26 May 2011
AWS Identity and Access Management		Added information about AWS Identity and Access Management (IAM), which enables users to specify which Amazon EC2 actions a user can use with Amazon EC2 resources in general. For more information, see Controlling Access to Amazon EC2 Resources (p. 619).	26 April 2011

Feature	API Version	Description	Release Date
Dedicated instances		Launched within your Amazon Virtual Private Cloud (Amazon VPC), Dedicated Instances are instances that are physically isolated at the host hardware level. Dedicated Instances let you take advantage of Amazon VPC and the AWS cloud, with benefits including on-demand elastic provisioning and pay only for what you use, while isolating your Amazon EC2 compute instances at the hardware level. For more information, see Dedicated Instances (p. 236) .	27 March 2011
Reserved Instances updates to the AWS Management Console		Updates to the AWS Management Console make it easier for users to view their Reserved Instances and purchase additional Reserved Instances, including Dedicated Reserved Instances.	27 March 2011
Support for Windows Server 2008 R2		Amazon EC2 now provides you with several pre-configured Windows Server 2008 R2 AMIs. These AMIs are immediately available for use in every region and in most 64-bit instance types, excluding t1.micro and HPC families. The AMIs will support multiple languages.	15 March 2011
Metadata information	2011-01-01	Added information about metadata to reflect changes in the 2011-01-01 release. For more information, see Instance Metadata and User Data (p. 271) and Instance Metadata Categories (p. 277) .	11 March 2011
Amazon EC2 VM Import Connector for VMware vCenter		Added information about the Amazon EC2 VM Import Connector for VMware vCenter virtual appliance (Connector). The Connector is a plug-in for VMware vCenter that integrates with VMware vSphere Client and provides a graphical user interface that you can use to import your VMware virtual machines to Amazon EC2.	3 March 2011
Force volume detachment		You can now use the AWS Management Console to force the detachment of an Amazon EBS volume from an instance. For more information, see Detaching an Amazon EBS Volume from an Instance (p. 781) .	23 February 2011
Instance termination protection		You can now use the AWS Management Console to prevent an instance from being terminated. For more information, see Enabling Termination Protection for an Instance (p. 266) .	23 February 2011
VM Import	2010-11-15	Added information about VM Import, which allows you to import a virtual machine or volume into Amazon EC2. For more information, see the VM Import/Export User Guide .	15 December 2010

Feature	API Version	Description	Release Date
Basic monitoring for instances	2010-08-31	Added information about basic monitoring for EC2 instances.	12 December 2010
Cluster GPU instances	2010-08-31	Amazon EC2 offers cluster GPU instances (cg1.4xlarge) for high-performance computing (HPC) applications. For more information about the hardware specifications for each Amazon EC2 instance type, see Amazon EC2 Instances .	14 November 2010
Filters and Tags	2010-08-31	Added information about listing, filtering, and tagging resources. For more information, see Listing and Filtering Your Resources (p. 856) and Tagging Your Amazon EC2 Resources (p. 859) .	19 September 2010
Idempotent Instance Launch	2010-08-31	Added information about ensuring idempotency when running instances.	19 September 2010
Micro instances	2010-06-15	Amazon EC2 offers the <code>t1.micro</code> instance type for certain types of applications. For more information, see T1 Micro Instances (p. 136) .	8 September 2010
AWS Identity and Access Management for Amazon EC2		Amazon EC2 now integrates with AWS Identity and Access Management (IAM). For more information, see Controlling Access to Amazon EC2 Resources (p. 619) .	2 September 2010
Cluster instances	2010-06-15	Amazon EC2 offers cluster compute instances for high-performance computing (HPC) applications. For more information about the hardware specifications for each Amazon EC2 instance type, see Amazon EC2 Instances .	12 July 2010
Amazon VPC IP Address Designation	2010-06-15	Amazon VPC users can now specify the IP address to assign an instance launched in a VPC.	12 July 2010
Amazon CloudWatch Monitoring for Amazon EBS Volumes		Amazon CloudWatch monitoring is now automatically available for Amazon EBS volumes. For more information, see Monitoring Volumes with CloudWatch (p. 769) .	14 June 2010
High-memory extra large instances	2009-11-30	Amazon EC2 now supports a High-Memory Extra Large (m2.xlarge) instance type. For more information about the hardware specifications for each Amazon EC2 instance type, see Amazon EC2 Instances .	22 February 2010
Reserved Instances with Windows		Amazon EC2 now supports Reserved Instances with Windows.	22 February 2010

AWS Glossary

For the latest AWS terminology, see the [AWS Glossary](#) in the *AWS General Reference*.