

# Microsoft Active Directory Domain Services on the AWS Cloud

## Quick Start Reference Deployment

*Santiago Cardenas*  
*Solutions Architect, AWS Quick Start Reference Team*

*March 2014*  
*([last update](#): July 2016)*

This guide is also available in HTML format at  
<http://docs.aws.amazon.com/quickstart/latest/active-directory-ds/>.



## Contents

Overview .....	4
AD DS on AWS.....	4
Cost and Licenses.....	4
AWS Services.....	5
Deployment Scenarios and Architecture .....	6
Scenario 1: Deploy and Manage Your Own AD DS on AWS .....	6
Scenario 2: Extend On-Premises AD DS Installation to the AWS Cloud .....	9
Scenario 3: Deploy AD DS with AWS Directory Service on the AWS Cloud .....	11
Design Considerations .....	13
Amazon VPC Configuration.....	13
Security Group Ingress Traffic.....	14
Setting up Secure Administrative Access Using Remote Desktop Gateway.....	15
Active Directory Design .....	16
Site Topology .....	16
Highly Available Directory Domain Services.....	17
Read-Only and Writable Domain Controllers .....	18
Active Directory DNS and DHCP Inside the Amazon VPC .....	19
DNS Settings on Windows Server Instances .....	21
Deployment Steps .....	22
Step 1. Prepare Your AWS Account .....	22
Step 2. Launch the Quick Start .....	25
Step 3. Post-Deployment Tasks (Scenario 2 Only) .....	33
Connecting Your On-Premises Network to Amazon VPC .....	33
Deploying Additional Domain Controllers in the AWS Cloud .....	36

---

Configuring Active Directory Sites and Services .....	37
Configuring DNS Resolution.....	37
Troubleshooting .....	38
Security .....	39
Additional Resources .....	42
Send Us Feedback .....	43
Document Revisions.....	44

## About This Guide

This Quick Start reference deployment guide discusses architectural considerations and configuration steps for deploying a highly available Microsoft Active Directory Domain Services (AD DS) environment on the Amazon Web Services (AWS) cloud. It also provides links for viewing and launching [AWS CloudFormation](#) templates that automate the deployment.

The guide is for IT infrastructure architects and administrators who want to design and deploy a solution to launch AD DS in the AWS cloud, or extend their on-premises AD DS into the AWS cloud.

[Quick Starts](#) are automated reference deployments for key enterprise workloads on the AWS cloud. Each Quick Start launches, configures, and runs the AWS compute, network, storage, and other services required to deploy a specific workload on AWS, using AWS best practices for security and availability.

# Overview

## AD DS on AWS

Amazon Web Services (AWS) provides a comprehensive set of services and tools for deploying Microsoft Windows-based workloads on its reliable and secure cloud infrastructure. Microsoft Active Directory Domain Services (AD DS) and Domain Name System (DNS) are core Windows services that provide the foundation for many enterprise class Microsoft-based solutions, including Microsoft SharePoint, Microsoft Exchange, and .NET applications.

This Quick Start is for organizations running workloads in the AWS cloud that require secure, low-latency connectivity to AD DS and DNS services. After reading this guide, IT infrastructure personnel should have a good understanding of how to design and deploy a solution to launch AD DS in the AWS cloud, or extend their on-premises AD DS into the AWS cloud.

This Quick Start assumes that you're already familiar with Active Directory and DNS. For details, please consult the Microsoft product documentation.

This guide focuses on infrastructure configuration topics that require careful consideration when you are planning and deploying AD DS, domain controller instances, and DNS services in the AWS cloud. We don't cover general Windows Server installation and software configuration tasks. For general software configuration guidance and best practices, consult the Microsoft product documentation.

## Cost and Licenses

You are responsible for the cost of the AWS services used while running this Quick Start reference deployment. There is no additional cost for using the Quick Start. For cost estimates, please use the [AWS Simple Monthly Calculator](#), and see the pricing pages for each AWS service you will be using in this Quick Start for full details.

This Quick Start launches the Amazon Machine Image (AMI) for Microsoft Windows Server 2012 R2 and includes the license for the Windows Server operating system. The AMI is updated on a regular basis with the latest service pack for the operating system, so you don't have to install any updates. The Windows Server AMI doesn't require Client Access

Licenses (CALs) and includes two Microsoft Remote Desktop Services licenses. For details, see [Microsoft Licensing on AWS](#).

## AWS Services

The core AWS components used by this Quick Start include the following AWS services. (If you are new to AWS, see the [Getting Started section](#) of the AWS documentation.)

- [Amazon VPC](#) – The Amazon Virtual Private Cloud (Amazon VPC) service lets you provision a private, isolated section of the AWS cloud where you can launch AWS services and other resources in a virtual network that you define. You have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways.
- [Amazon EC2](#) – The Amazon Elastic Compute Cloud (Amazon EC2) service enables you to launch virtual machine instances with a variety of operating systems. You can choose from existing Amazon Machine Images (AMIs) or import your own virtual machine images.
- [NAT Gateway](#) – NAT gateways are network address translation (NAT) devices, which provide outbound Internet access to instances in a private subnets, but prevent the Internet from accessing those instances. NAT gateways provide better availability and bandwidth than NAT instances. The NAT Gateway service is a managed service that takes care of administering NAT gateways for you.
- [AWS Direct Connect](#) – The AWS Direct Connect service enables you to establish a private connection between AWS and your on-premises data center. With this connection in place, you can create virtual interfaces to establish private connectivity to multiple Amazon VPCs, bypassing Internet service providers in your network path.
- [AWS Directory Service](#) – The AWS Directory Service makes it easy to set up and operate a new directory in the AWS cloud. This Quick Start supports AWS Directory Service for Microsoft Active Directory (Enterprise Edition), which provides most of the features offered by Microsoft Active Directory plus integration with AWS applications.

# Deployment Scenarios and Architecture

This Quick Start provides separate AWS CloudFormation templates to support three deployment scenarios. For each scenario, you also have the option to create a new Amazon VPC or use your existing Amazon VPC infrastructure. Choose the scenario that best fits your needs.

- **Scenario 1: Deploy and manage your own AD DS installation on the AWS cloud.** The AWS CloudFormation template for this scenario builds the AWS cloud infrastructure, and sets up and configures AD DS and AD-integrated DNS on the AWS cloud. It doesn't include AWS Directory Service, so you handle all AD DS maintenance and monitoring tasks yourself. You can also choose to deploy the Quick Start into your existing VPC infrastructure.
- **Scenario 2: Extend your on-premises AD DS to the AWS cloud.** The AWS CloudFormation template for this scenario builds the base AWS cloud infrastructure for AD DS, and you perform several manual steps to extend your existing network to AWS and to promote your domain controllers. As in scenario 1, you manage all AD DS tasks yourself. You can also choose to deploy the Quick Start into your existing VPC infrastructure.
- **Scenario 3: Deploy AD DS with AWS Directory Service on the AWS cloud.** The AWS CloudFormation template for this scenario builds the base AWS cloud infrastructure, and deploys AWS Directory Service for Microsoft AD, which offers managed AD DS functionality on the AWS cloud. AWS Directory Service takes care of AD DS tasks such as building a highly available directory topology, monitoring domain controllers, and configuring backups and snapshots. As with the first two scenarios, you can choose to deploy the Quick Start into an existing VPC infrastructure.

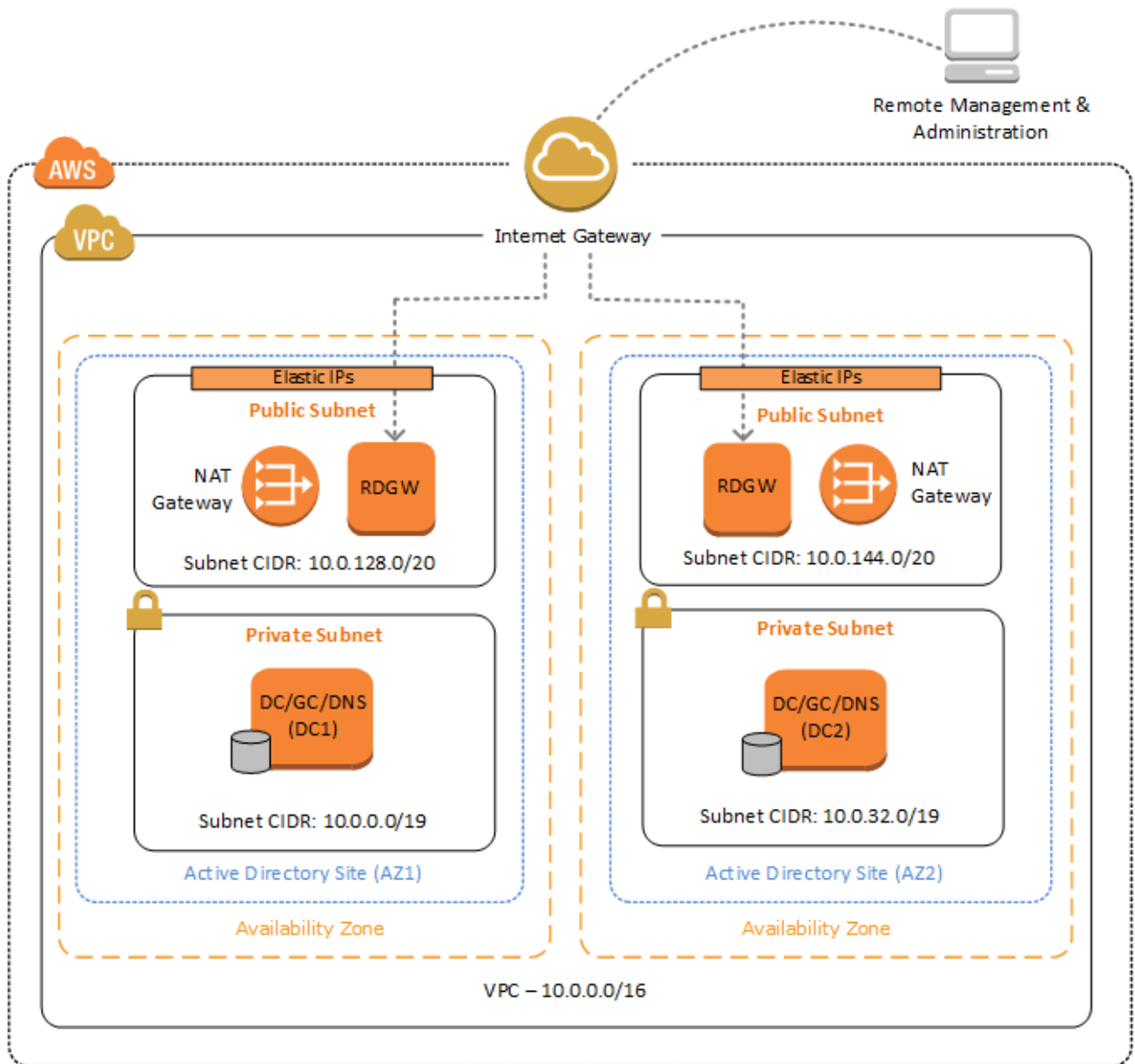
The following sections discuss the Quick Start architecture for each scenario, and explain the automation provided by the Quick Start template.

## Scenario 1: Deploy and Manage Your Own AD DS on AWS

This scenario is based on a new installation of AD DS in the AWS cloud without AWS Directory Service. The AWS CloudFormation templates that automate this deployment perform the following tasks to set up the architecture illustrated in Figure 1:

- Sets up the Amazon VPC, including private and public subnets in two Availability Zones.\*
- Configures two NAT gateways in the public subnets.\*
- Configures private and public routes.\*
- Launches Windows Server 2012 R2 Amazon Machine Images (AMIs), and sets up and configures AD DS and AD-integrated DNS.
- Configures security groups and rules for traffic between instances.
- Sets up and configures Active Directory Sites and Subnets.
- Enables ingress traffic into the Amazon VPC for administrative access to Remote Desktop Gateway.

\* The template that deploys the Quick Start into an existing VPC skips the tasks marked by asterisks.



**Figure 1: Quick Start Architecture for Highly Available AD DS on AWS**

In this architecture:

- Domain controllers are deployed into two private Amazon VPC subnets in separate Availability Zones, making AD DS highly available.



- NAT gateways are deployed to public subnets, providing outbound Internet access for instances in private subnets.
- Remote Desktop gateways are deployed to each public subnet for secure remote access to instances in private subnets.

Windows Server 2012 R2 is used for the Remote Desktop Gateway and domain controller instances. The AWS CloudFormation template bootstraps each instance, deploying the required components, finalizing the configuration to create a new AD forest, and promoting instances in two Availability Zones to Active Directory domain controllers.

To deploy this stack, follow the step-by-step instructions in the [Deployment Steps](#) section. After deploying this stack, you can move on to deploying your AD DS-dependent servers into the Amazon VPC.

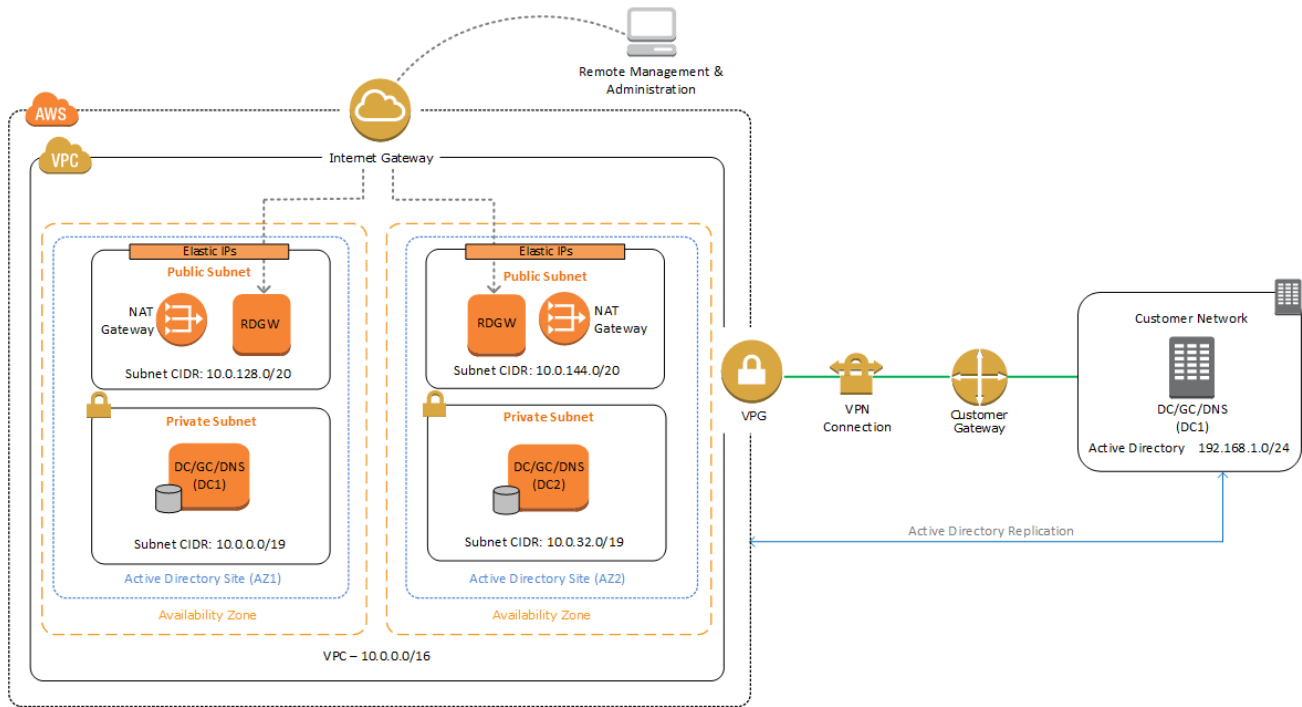
## Scenario 2: Extend On-Premises AD DS Installation to the AWS Cloud

This scenario is for users who want to use their existing installation of AD DS and extend their on-premises network to the Amazon VPC, when a new deployment of AD DS is not an option. The AWS CloudFormation templates that automate this deployment perform these tasks:

- Sets up the Amazon VPC, including private and public subnets in two Availability Zones.\*
- Configures two NAT gateways in the public subnets.\*
- Configures private and public routes.\*
- Launches Windows Server 2012 R2 AMIs.
- Configures security groups and rules for traffic between instances.
- Enables ingress traffic into the VPC for administrative access to Remote Desktop Gateway.

\* The template that deploys the Quick Start into an existing VPC skips the tasks marked by asterisks.

The AWS CloudFormation template deploys the architecture shown in Figure 2, except for the virtual private gateway and VPN connection, which you can create manually.



**Figure 2: Quick Start Architecture for Extending Your On-Premises AD DS to AWS**

This scenario provides an example of using an Amazon VPC and a virtual private gateway to enable communication with your own network over an IPsec VPN tunnel. Active Directory is deployed in the customer data center, and Windows servers are deployed into two Amazon VPC subnets. After deploying the VPN connection, you can promote the Windows instances to domain controllers in the on-premises Active Directory forest, making AD DS highly available in the AWS cloud.

After you deploy the VPN connection and promote your servers to domain controllers, you can launch additional instances into the empty Amazon VPC subnets in the web, application, or database tier. These instances will have access to cloud-based domain controllers for secure, low-latency directory services and DNS. All network traffic, including AD DS communication, authentication requests, and Active Directory replication, is secured either within the private subnets or across the VPN tunnel.

## Scenario 3: Deploy AD DS with AWS Directory Service on the AWS Cloud

This scenario is similar to scenario 1, except that it includes AWS Directory Service to provision and manage AD DS on the AWS cloud. Instead of fully managing AD DS yourself, you rely on AWS Directory Service for tasks such as building a highly available directory topology, monitoring domain controllers, and configuring backups and snapshots.

AWS Directory Service deploys AD DS across multiple Availability Zones, and automatically detects and replaces domain controllers that fail. AWS Directory Service also handles time-consuming tasks such as patch management, software updates, data replication, snapshot backups, replication monitoring, and point-in-time restores. For more information about AWS Directory Service, see [product details](#) and the [AWS documentation](#).

The AWS CloudFormation templates that automate this deployment perform these tasks:

- Sets up the Amazon VPC, including private and public subnets in two Availability Zones.\*
- Configures two NAT gateways in the public subnets.\*
- Configures private and public routes.\*
- Launches Windows Server 2012 R2 AMIs.
- Configures security groups and rules for traffic between instances.
- Enables ingress traffic into the VPC for administrative access to Remote Desktop Gateway.
- Sets up AWS Directory Service to provision and manage AD DS in the private subnets.

\* The template that deploys the Quick Start into an existing VPC skips the tasks marked by asterisks.

The architecture for this scenario is illustrated in Figure 3.

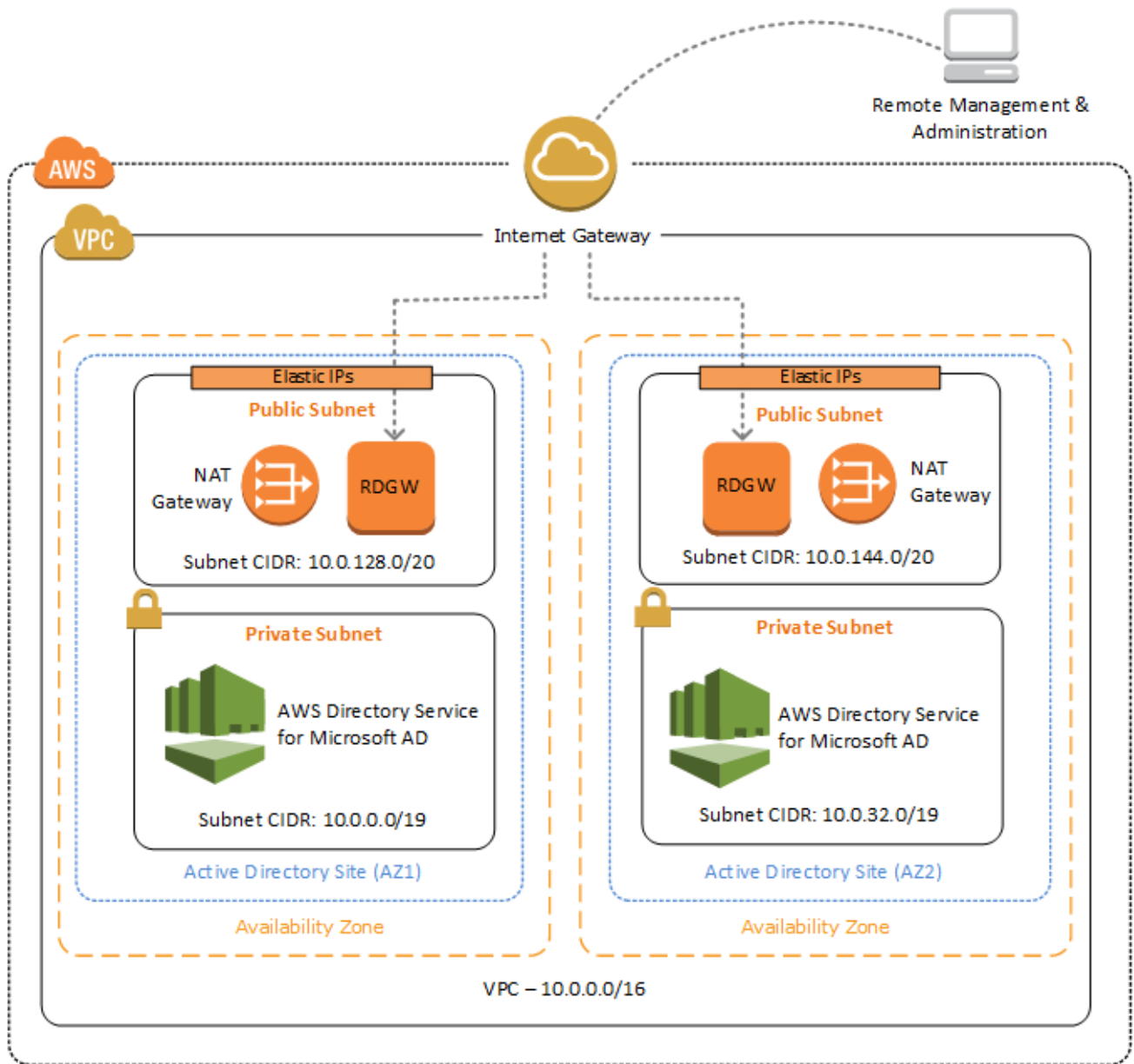


Figure 3: Quick Start Architecture for Deploying AD DS with AWS Directory Service

# Design Considerations

Deploying a functional AD DS deployment in the AWS cloud requires a good understanding of specific AWS services. In this section, we discuss key considerations for both new AD DS deployments and extensions of existing AD DC deployments to the AWS cloud. We discuss how to use Amazon VPC to define your networks in the cloud, and cover domain controller placement, Active Directory Sites and Services configuration, and how DNS and DHCP work in Amazon VPC.

## Amazon VPC Configuration

With Amazon VPC, you can define a virtual network topology closely resembling a traditional network that you might operate on your own premises. An Amazon VPC can span multiple Availability Zones, which enables you to place independent infrastructure in physically separate locations. A Multi-AZ deployment provides high availability and fault tolerance. In the scenarios in this guide, we place domain controllers in two Availability Zones to provide highly available, low latency access to AD DS services in the AWS cloud.

Each scenario is automated by two templates: one that builds a new VPC for the deployment, and the other that deploys into an existing VPC. To accommodate highly available AD DS in the AWS cloud, the Quick Start builds (or requires, in the case of the existing VPC template) a base Amazon VPC configuration that complies with the following AWS best practices:

- Domain controllers should be placed in a minimum of two Availability Zones to provide high availability.
- Domain controllers and other non-Internet facing servers should be placed in private subnets.
- Instances launched by the deployment templates provided in this guide will require Internet access to connect to the AWS CloudFormation endpoint during the bootstrapping process. To support this configuration, public subnets are used to host NAT gateways for outbound Internet access. Remote Desktop gateways are also deployed into the public subnets for remote administration. Other components such as reverse proxy servers can be placed into these public subnets, if needed.

This Amazon VPC architecture uses two Availability Zones, each with its own distinct public and private subnets. We recommend that you leave plenty of unallocated address space to support the growth of your environment over time and to reduce the complexity of your Amazon VPC subnet design. This Quick Start uses a default Amazon VPC configuration that provides plenty of address space by using the minimum number of private and public subnets. In addition, we've defined optional *protected* and *spare* subnets for each Availability Zone. By default, this Quick Start uses the following CIDR ranges:

<b>VPC</b>	<b>10.0.0.0/16</b>
<b>Private subnets A</b>	<b>10.0.0.0/17</b>
Availability Zone 1	10.0.0.0/19
Availability Zone 2	10.0.32.0/19
<b>Public subnets</b>	<b>10.0.128.0/18</b>
Availability Zone 1	10.0.128.0/20
Availability Zone 2	10.0.144.0/20
<b>Private subnets B with dedicated custom network ACL</b>	<b>10.0.192.0/19</b>
Availability Zone 1	10.0.192.0/21
Availability Zone 2	10.0.200.0/21
<b>Spare subnet capacity</b>	<b>10.0.224.0/19</b>
Availability Zone 1	10.0.224.0/21
Availability Zone 2	10.0.232.0/21

If you have sensitive workloads that should be completely isolated from the Internet, you can create new Amazon VPC subnets using these optional address spaces. This also goes for the spare address space that can be used later, if needed. For background information and more details on this approach, see [Building a Modular and Scalable Virtual Network Architecture with Amazon VPC](#).

## Security Group Ingress Traffic

When launched, Amazon EC2 instances must be associated with a security group, which acts as a stateful firewall. You have complete control over the network traffic entering or leaving the security group, and you can build granular rules that are scoped by protocol, port number, and source/destination IP address or subnet. By default, all egress traffic

from the security group is permitted. However, ingress traffic must be configured to allow the appropriate traffic to reach your instances.

The [Securing the Microsoft Platform on Amazon Web Services](#) whitepaper discusses the different methods for securing your AWS infrastructure. Recommendations include providing isolation between application tiers by using security groups. We recommend that you tightly control ingress traffic in order to reduce the attack surface of your Amazon EC2 instances.

If you're deploying and managing your own AD DS installation, domain controllers and member servers will require several security group rules to allow traffic for services such as AD DS replication, user authentication, Windows Time services, and Distributed File System (DFS), among others. You should also consider restricting these rules to specific IP subnets that are used within your Amazon VPC.

We provide an example of how to implement these rules for each application tier later in this guide as part of the AWS CloudFormation template for each scenario. For a detailed list of port mappings used by the AWS CloudFormation templates, see the [Security](#) section of this guide.

For a complete list of ports, see [Active Directory and Active Directory Domain Services Port Requirements](#) in the Microsoft TechNet library. For step-by-step guidance for implementing rules, see [Adding Rules to a Security Group](#) in the *Amazon EC2 User Guide*.

## Setting up Secure Administrative Access Using Remote Desktop Gateway

As you design your architecture for highly available AD DS, you should also design for highly available and secure remote access. The Quick Start templates handle this by deploying Remote Desktop (RD) Gateway in each Availability Zone. In case of an Availability Zone outage, this architecture allows access to the resources that may have failed over to the other Availability Zone.

RD Gateway uses the Remote Desktop Protocol (RDP) over HTTPS to establish a secure, encrypted connection between remote administrators on the Internet and Windows-based Amazon EC2 instances without the need for a virtual private network (VPN) connection.

This configuration helps reduce the attack surface on your Windows-based Amazon EC2 instances while providing a remote administration solution for administrators.

The AWS CloudFormation templates provided with this Quick Start automatically deploy the architecture and configuration outlined in the [Remote Desktop Gateway Quick Start](#).

After you've launched your AD infrastructure by following the deployment steps in this guide, you will initially connect to your instances by using a standard RDP TCP port 3389 connection. You can then follow the steps in the [Remote Desktop Gateway Quick Start](#) to secure future connections via HTTPS.

## Active Directory Design

If you're managing your own AD DS infrastructure ([scenario 1](#) or [scenario 2](#)), review the following sections for key design considerations.

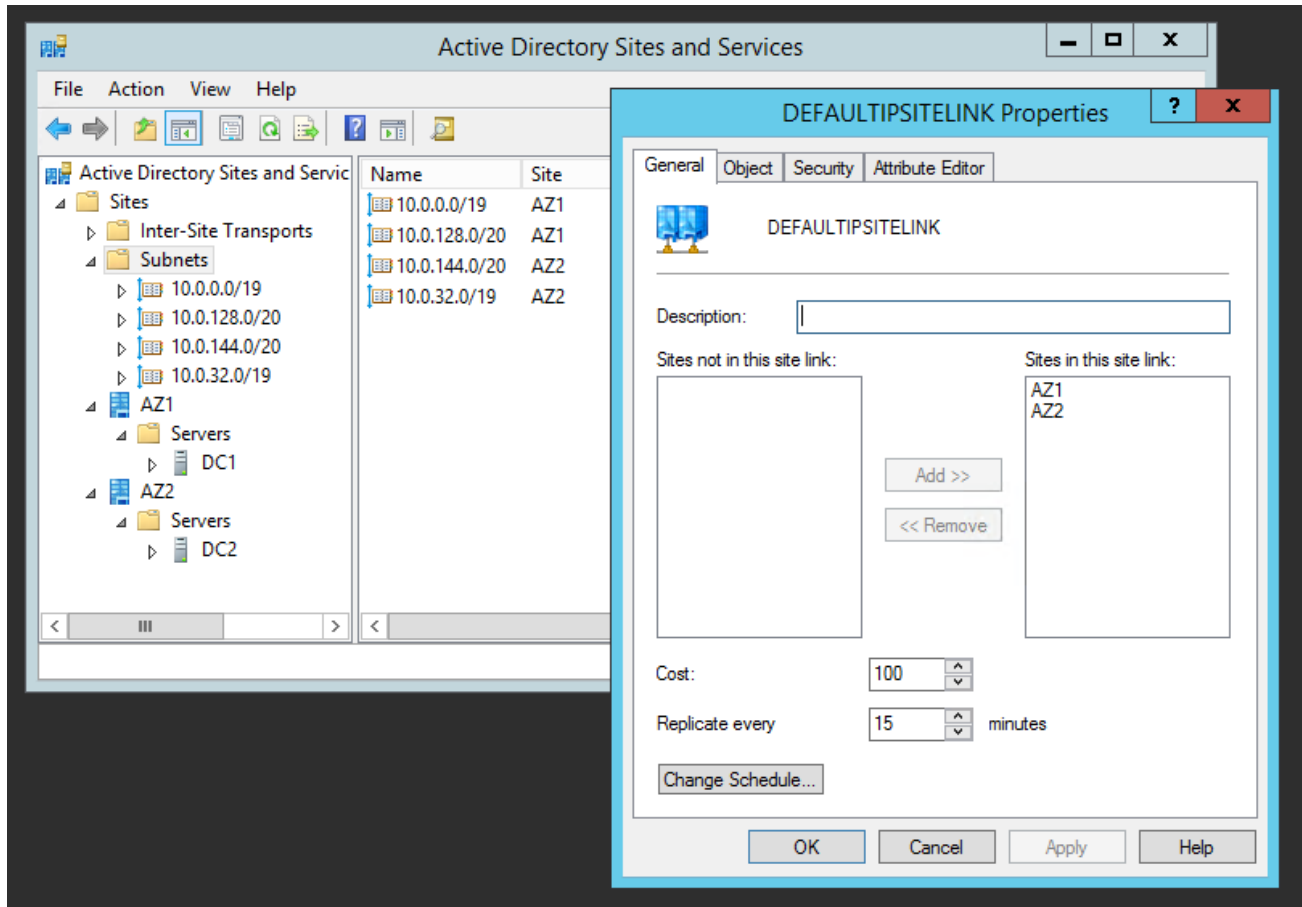
### Site Topology

Active Directory site topology allows you to logically define your physical and virtual networks. Active Directory replication sends directory changes from one domain controller to another, until all domain controllers have been updated. Site topology controls Active Directory replication between domain controllers within the same site and across site boundaries. Replication traffic between sites is compressed and replication is performed on a schedule based on a site link. Additionally, domain controllers use the site topology to provide client affinity, meaning that clients located within a specific site will prefer domain controllers in the same site.

Site topology is a crucial design consideration when running AD DS in the AWS cloud. A well-designed site topology allows you to define subnets that can be associated with the Availability Zones within your Amazon VPC. These associations help ensure that traffic—such as directory service queries, AD DS replication, and client authentication—uses the most efficient path to a domain controller. They also provide you with granular control over replication traffic.

Figure 4 shows an example of site and subnet definitions for a typical AD DS architecture running within an Amazon VPC. Active Directory sites (AZ1 and AZ2) have been created in the Active Directory Sites and Services snap-in. Subnets have been defined and associated with their respective site objects.





**Figure 4: Active Directory Sites and Services Configuration**

By creating Active Directory sites that represent each Availability Zone in the Amazon VPC, subnets associated with those sites can help ensure that domain-joined instances will primarily use a domain controller closest to them. This is also a key design configuration for maintaining a highly available AD DS deployment.

### Highly Available Directory Domain Services

Even in the smallest AD DS deployments, we recommend implementing at least two domain controllers in your AWS cloud environment. This design provides fault tolerance and prevents a single domain controller failure from affecting the availability of the AD DS. In order to provide higher availability, we recommend that you implement domain controllers in at least two Availability Zones.

To further support the high availability of your architecture and help mitigate the impact of a possible disaster, we also recommend placing global catalog servers and Active Directory DNS servers in each Availability Zone. Global catalogs provide a mechanism for forestwide searches and are required for logon authentication in forests with multiple domains. If you do not have a global catalog and a DNS server in each Availability Zone, AD DS queries and authentication traffic could cross Availability Zones. Although this is not technically an issue during normal operations, full AD DS service availability could be impacted by a single Availability Zone failure.

To implement these recommendations, we suggest that you make each domain controller a global catalog and DNS server. This configuration allows AD DS in each Availability Zone to operate independently, and helps ensure that AD DS availability is not affected in the unlikely event of disaster. If an Availability Zone in this architecture is cut off from other resources in the region, instances within the Availability Zone still have a local domain controller that can authenticate users, perform service directory lookups, and resolve DNS queries.

The requirements of a smaller environment might make a single Availability Zone more appealing. Even though a single Availability Zone AD DS design is not our recommendation, we realize that this may be the chosen architecture. In this case, we recommend that you deploy at least two domain controllers in your Availability Zone to provide redundancy.

The AWS CloudFormation template provided for [scenario 1](#) will build out an Active Directory Sites and Services configuration for you automatically that will support a highly available AD DS architecture. If you plan to deploy AD DS manually, make sure that you properly map subnets to the correct site to help ensure that AD DS traffic uses the best possible path.

For detailed guidance on creating sites, adding global catalog servers, and creating and managing site links, see the [Microsoft Active Directory Sites and Services](#) documentation.

## Read-Only and Writable Domain Controllers

Read-only domain controllers (RODCs) hold a copy of the AD DS database and respond to authentication requests, but applications or other servers cannot write to them. RODCs are typically deployed in locations where physical security cannot be guaranteed. For example,

in an on-premises scenario, you might deploy an RODC in a remote branch office where the physical server cannot be protected by a secure, locked closet or server room.

Writable domain controllers operate in a multi-master model; changes can be made on any writable server in the forest, and those changes are replicated to servers throughout the entire forest. Several key functions and Microsoft enterprise applications require connectivity to a writable domain controller.

If you are planning to deploy enterprise application servers into the AWS cloud, an RODC may not be a viable option. For example, an RODC cannot process a password reset for an end user, and Microsoft Exchange Server cannot use an RODC to perform directory look-ups. Make sure you understand the requirements of the application, the dependencies on AD DS, and compatibility before considering RODCs.

### Active Directory DNS and DHCP Inside the Amazon VPC

With an Amazon VPC, Dynamic Host Configuration Protocol (DHCP) services are provided by default for your instances. DHCP scopes do not need to be managed; they are created for the Amazon VPC subnets you define when you deploy your solution. These DHCP services cannot be disabled, so you'll need to use them rather than deploying your own DHCP server.

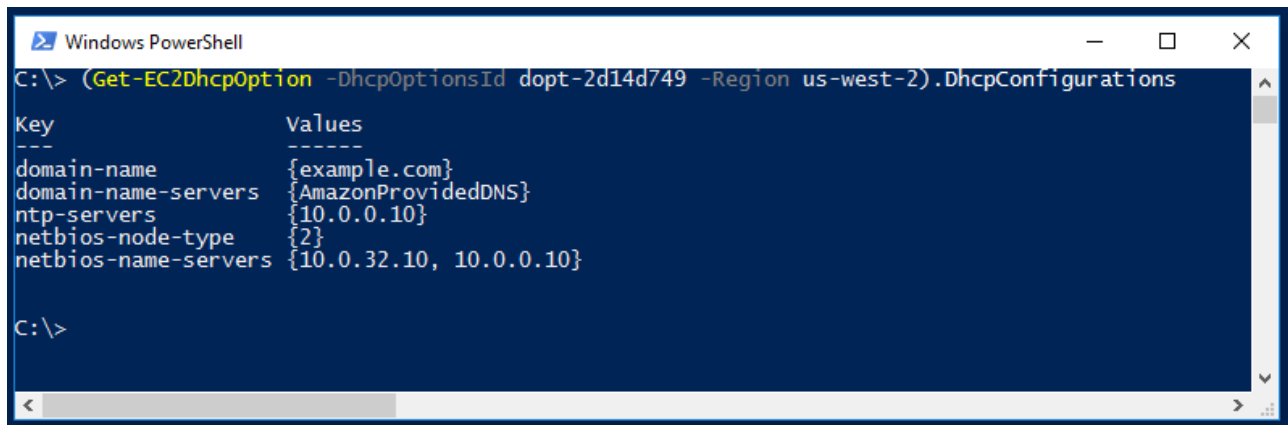
The Amazon VPC also provides an internal DNS server. This DNS provides instances with basic name resolution services for Internet access. This is crucial for access to AWS service endpoints such as AWS CloudFormation and Amazon Simple Storage Service (Amazon S3) during the bootstrapping process when you launch the Quick Start.

Amazon-provided DNS server settings will be assigned to instances launched into the VPC based on a DHCP options set. DHCP options sets are used within an Amazon VPC to define scope options, such as the domain name or the name servers that should be handed to your instances via DHCP. Amazon-provided DNS is used only for public DNS resolution.

Since Amazon-provided DNS cannot be used to provide name resolution services for Active Directory, you'll need to ensure that domain-joined Windows instances have been configured to use Active Directory DNS.

As an alternative to statically assigning Active Directory DNS server settings on Windows instances, you have the option of specifying them using a custom DHCP options set. This will allow you to assign your Active Directory DNS suffix and DNS server IP addresses as the name servers within the Amazon VPC via DHCP.

Figure 5 shows the configuration of a custom DHCP options set, where the `domain-name-servers` field has been set to two IP addresses (the maximum is four) of domain controllers running Active Directory-integrated DNS in separate Availability Zones.

A screenshot of a Windows PowerShell window. The title bar reads "Windows PowerShell". The command entered is `(Get-EC2DhcpOption -DhcpOptionsId dopt-2d14d749 -Region us-west-2).DhcpConfigurations`. The output is a table with two columns: "Key" and "Values". The rows are: `domain-name` with value `{example.com}`; `domain-name-servers` with value `{AmazonProvidedDNS}`; `ntp-servers` with value `{10.0.0.10}`; `netbios-node-type` with value `{2}`; and `netbios-name-servers` with value `{10.0.32.10, 10.0.0.10}`. The prompt `C:\>` is visible at the bottom.

```
Windows PowerShell
C:\> (Get-EC2DhcpOption -DhcpOptionsId dopt-2d14d749 -Region us-west-2).DhcpConfigurations

Key                Values
----                -
domain-name        {example.com}
domain-name-servers {AmazonProvidedDNS}
ntp-servers        {10.0.0.10}
netbios-node-type  {2}
netbios-name-servers {10.0.32.10, 10.0.0.10}

C:\>
```

**Figure 5: PowerShell Output Showing DHCP Options Set Configuration**

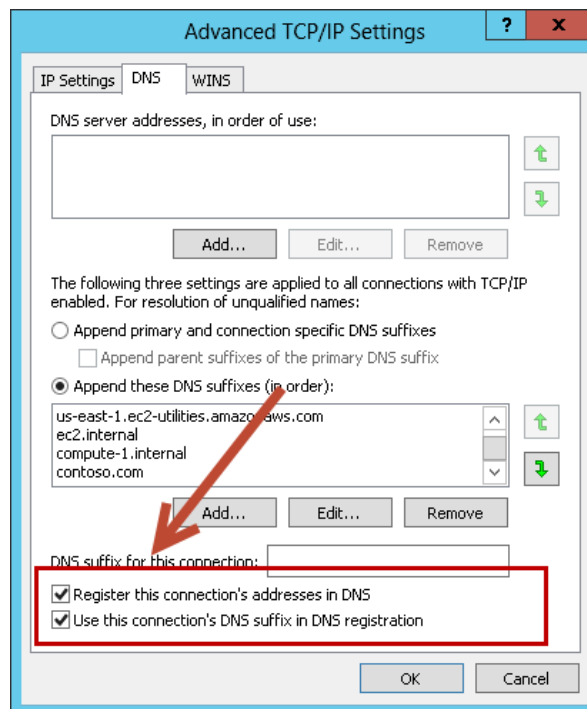
**Note** The IP addresses in the `domain-name-servers` field are always returned in the same order. If the first DNS server in the list fails, instances should fall back to the second IP and continue to resolve host names successfully. However, during normal operations, the first DNS server listed will always handle DNS requests. If you want to ensure that DNS queries are distributed evenly across multiple servers, you should consider statically configuring DNS server settings on your instances.

For details on creating a custom DHCP options set and associating it with your Amazon VPC, see [Working with DHCP Options Sets](#) in the *Amazon VPC User Guide*.

**Note** For [scenario 3](#), the AWS CloudFormation template configures the DHCP options set with the Active Directory domain controllers as the name servers, as recommended by the [AWS Directory Service documentation](#). This means that instances that need to join the domain will automatically be able to join, without requiring any changes.

## DNS Settings on Windows Server Instances

To make sure that domain-joined Windows instances will automatically register host (A) and reverse lookup (PTR) records with Active Directory-integrated DNS, set the properties of the network connection as shown in Figure 6.



**Figure 6: Advanced TCP/IP Settings on a Domain-Joined Windows Instance**

The default configuration for a network connection is set to automatically register the connections address in DNS. In other words, as shown in Figure 6, the **Register this connection's address in DNS** option is selected for you automatically. This takes care of host (A) record dynamic registration. However, if you do not also select the second option, **Use this connection's DNS suffix in DNS registration**, dynamic registration of PTR records will not take place.

If you have a small number of instances in the Amazon VPC, you may choose to configure the network connection manually. For larger fleets, you can push this setting out to all your Windows instances by using Active Directory Group Policy. For step-by-step instructions, see [IPv4 and IPv6 Advanced DNS Tab](#) in the Microsoft TechNet Library.

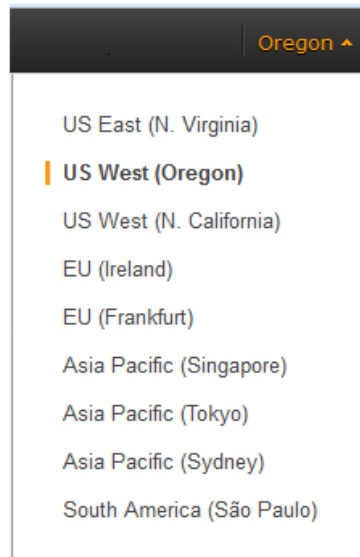
## Deployment Steps

Follow the step-by-step instructions in this section to set up your AWS account, launch the templates, and customize your deployment.

### Step 1. Prepare Your AWS Account

Before you deploy the Quick Start, make sure that your AWS account is set up properly by following these steps.

1. If you don't already have an AWS account, create one at <http://aws.amazon.com> by following the on-screen instructions. Part of the sign-up process involves receiving a phone call and entering a PIN using the phone keypad.
2. Use the region selector in the navigation bar to choose the Amazon EC2 region where you want to deploy AD DS on AWS.



**Figure 7: Choosing an Amazon EC2 Region**

Consider choosing a region closest to your data center or corporate network to reduce network latency between systems running on AWS and the systems and users on your corporate network.

**Important** This Quick Start uses the **m4.xlarge** instance type for the Active Directory and Remote Desktop Gateway instances, and NAT gateways for outbound Internet access. At the time of this writing, some of these features aren't available in China (Beijing), South America (São Paulo), or Asia Pacific (Seoul).

Also, if you're deploying [scenario 3](#), note that AWS Directory Service is available only in the regions listed on the [AWS Regions and Endpoints](#) page in the AWS documentation. We recommend that you check service availability before you choose a region. Otherwise, deployment will fail.

3. Create a [key pair](#) in your preferred region. To do this, in the navigation pane of the Amazon EC2 console, choose **Key Pairs**, **Create Key Pair**, type a name, and then choose **Create**.

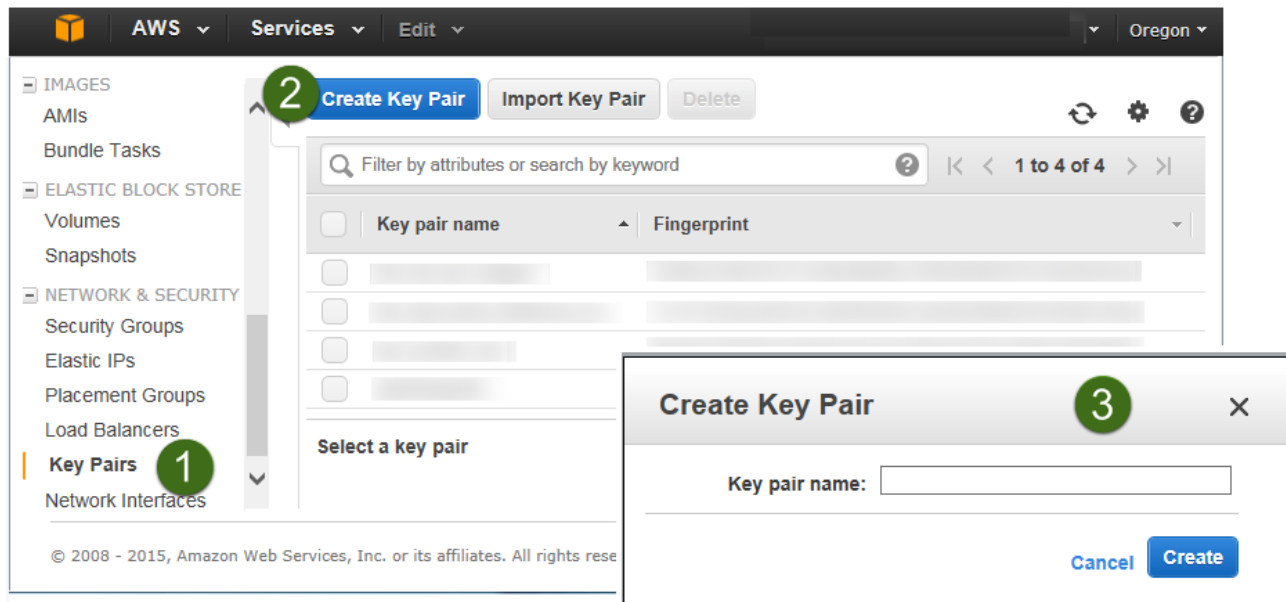


Figure 8: Creating a Key Pair

Amazon EC2 uses public-key cryptography to encrypt and decrypt login information. To be able to log in to your instances, you must create a key pair. With Windows instances, we use the key pair to obtain the administrator password via the Amazon EC2 console and then log in using Remote Desktop Protocol (RDP) as explained in the [step-by-step instructions](#) in the *Amazon Elastic Compute Cloud User Guide*.

4. If necessary, [request a service limit increase](#) for the Amazon EC2 **m4.xlarge** instance type. To do this, in the AWS Support Center, choose **Create Case, Service Limit Increase, EC2 instances**, and then complete the fields in the limit increase form. The current default limit is 20 instances.

You might need to request an increase if you already have an existing deployment that uses this instance type, and you think you might exceed the default limit with this reference deployment. It might take a few days for the new service limit to become effective. For more information, see [Amazon EC2 Service Limits](#) in the AWS documentation.

The screenshot shows the AWS Support Center interface for creating a case. The left sidebar has a 'Create Case' button marked with a green circle '1'. The main content area is titled 'Create Case' and shows the 'Service Limit Increase' option selected under 'Regarding\*', marked with a green circle '2'. The 'Limit Type' is set to 'EC2 Instances', marked with a green circle '3'. Below this, there is a 'Request 1' section with the following fields: 'Region\*' (US West (Oregon)), 'Primary Instance Type\*' (c3.8xlarge), 'Limit\*' (Instance Limit), and 'New limit value\*' (25). This section is marked with a green circle '4'. There is also an 'Add another request' button at the bottom.

Figure 9: Requesting a Service Limit Increase



## Step 2. Launch the Quick Start

In this section, we've provided general instructions for deploying the templates in the AWS CloudFormation console, followed by links and parameter tables for each [scenario](#).

1. Choose one of the following options to deploy the AWS CloudFormation template into your AWS account. For help choosing an option, see the discussion of [deployment scenarios](#) earlier in this guide.

<a href="#">Scenario 1</a>	<a href="#">Scenario 2</a>	<a href="#">Scenario 3</a>
<b>Deploy and manage your own AD DS installation on AWS</b>	<b>Extend your on-premises AD DS to AWS</b>	<b>Deploy AD DS with AWS Directory Service on AWS</b>
<a href="#">Launch for new VPC</a>	<a href="#">Launch for new VPC</a>	<a href="#">Launch for new VPC</a>
<a href="#">Launch for existing VPC</a>	<a href="#">Launch for existing VPC</a>	<a href="#">Launch for existing VPC</a>

The template is launched in the US West (Oregon) Region by default. You can change the region by using the region selector in the navigation bar.

Each deployment takes approximately one hour.

**Note** You are responsible for the cost of the AWS services used while running this Quick Start reference deployment. There is no additional cost for using this Quick Start. For cost estimates, please use the [AWS Simple Monthly Calculator](#), and see the pricing pages for each AWS service you will be using in this Quick Start for full details.

2. On the **Select Template** page, keep the default setting for the template URL, and then choose **Next**.
3. On the **Specify Details** page, review the parameters for the template. Provide values for the parameters that require input. For all other parameters, review the default settings and customize them as necessary. When you finish reviewing and customizing the parameters, choose **Next**.

**Note** You can also download the templates and edit them to create your own parameters based on your specific deployment scenario.

In the following tables, parameters are listed and described separately for [scenario 1](#), [scenario 2](#), and [scenario 3](#).

**Note** The two templates provided for each scenario share most, but not all, of the same parameters. For example, the template for an existing VPC also prompts you for the VPC and private subnet IDs in your existing VPC environment.

- **Scenario 1: Parameters for deploying and managing your own AD DS**

[View the template for new VPC](#)

[View the template for existing VPC](#)

### Network Configuration:

Parameter label (name)	Default	Description
<b>Availability Zones</b> (AvailabilityZones)	<i>Requires input</i>	The list of Availability Zones to use for the subnets in the VPC. The Quick Start uses two Availability Zones from your list and preserves the logical order you specify.
<b>VPC CIDR</b> (VPCCIDR)	10.0.0.0/16	CIDR block for the Amazon VPC.
<b>Private Subnet 1 CIDR</b> (PrivateSubnet1CIDR)	10.0.0.0/19	CIDR block for the private subnet located in Availability Zone 1.
<b>Private Subnet 2 CIDR</b> (PrivateSubnet2CIDR)	10.0.32.0/19	CIDR block for the private subnet located in Availability Zone 2.
<b>Public Subnet 1 CIDR</b> (PublicSubnet1CIDR)	10.0.128.0/20	CIDR block for the public subnet located in Availability Zone 1.
<b>Public Subnet 2 CIDR</b> (PublicSubnet2CIDR)	10.0.144.0/20	CIDR block for the public subnet located in Availability Zone 2.
<b>Allowed Remote Desktop Gateway External Access CIDR</b> (RDGWCIDR)	0.0.0.0/0	Allowed CIDR block for external access to the Remote Desktop Gateway instances.

**Amazon EC2 Configuration:**

Parameter label (name)	Default	Description
<b>Key Pair Name</b> (KeyPairName)	<i>Requires input</i>	Public/private key pair, which allows you to connect securely to your instance after it launches. When you created an AWS account, this is the key pair you created in your preferred region.
<b>NAT Instance Type</b> (NATInstanceType)	t2.small	EC2 instance type for the NAT instances. NAT instances are used only if the region doesn't support NAT gateways.
<b>Domain Controller 1 Instance Type</b> (ADServer1InstanceType)	m4.xlarge	EC2 instance type for the first Active Directory instance.
<b>Domain Controller 1 NetBIOS Name</b> (ADServer1NetBIOSName)	DC1	NetBIOS name of the first Active Directory server. This can be up to 15 characters long.
<b>Domain Controller 1 Private IP Address</b> (ADServer1PrivateIP)	10.0.0.10	Fixed private IP for the first Active Directory server located in Availability Zone 1.
<b>Domain Controller 2 Instance Type</b> (ADServer2InstanceType)	m4.xlarge	EC2 instance type for the second Active Directory instance.
<b>Domain Controller 2 NetBIOS Name</b> (ADServer2NetBIOSName)	DC2	NetBIOS name of the second Active Directory server. This can be up to 15 characters long.
<b>Domain Controller 2 Private IP Address</b> (ADServer2PrivateIP)	10.0.32.10	Fixed private IP for the second Active Directory server located in Availability Zone 2.
<b>Remote Desktop Gateway 1 Instance Type</b> (RDGW1InstanceType)	t2.large	EC2 instance type for the first Remote Desktop Gateway instance.
<b>Remote Desktop Gateway 1 NetBIOS Name</b> (RDGW1NetBIOSName)	RDGW1	NetBIOS name of the first Remote Desktop Gateway. This can be up to 15 characters long.
<b>Remote Desktop Gateway 1 Private IP Address</b> (RDGW1PrivateIP)	10.0.128.11	Fixed private IP for the first Remote Desktop Gateway located in Availability Zone 1.
<b>Remote Desktop Gateway 2 Instance Type</b> (RDGW2InstanceType)	t2.large	EC2 instance type for the second Remote Desktop Gateway instance.

Parameter label (name)	Default	Description
<b>Remote Desktop Gateway 2 NetBIOS Name</b> (RDGW2NetBIOSName)	RDGW2	NetBIOS name of the second Remote Desktop Gateway. This can be up to 15 characters long.
<b>Remote Desktop Gateway 2 Private IP Address</b> (RDGW2PrivateIP)	10.0.144.11	Fixed private IP for the second Remote Desktop Gateway located in Availability Zone 2.

### Microsoft Active Directory Configuration:

Parameter label (name)	Default	Description
<b>Domain DNS Name</b> (DomainDNSName)	example.com	Fully qualified domain name (FQDN) of the forest root domain.
<b>Domain NetBIOS Name</b> (DomainNetBIOSName)	example	NetBIOS name of the domain for users of earlier versions of Windows. This can be up to 15 characters long.
<b>Restore Mode Password</b> (RestoreModePassword)	<i>Requires input</i>	Password for a separate administrator account when the domain controller is in restore mode. This must be a <a href="#">complex password</a> that's at least 8 characters long.
<b>Domain Admin User Name</b> (DomainAdminUser)	StackAdmin	User name for the account that is added as domain administrator. This is separate from the default administrator account.
<b>Domain Admin Password</b> (DomainAdminPassword)	<i>Requires input</i>	Password for the domain administrator user. This must be a <a href="#">complex password</a> that's at least 8 characters long.

- **Scenario 2: Parameters for extending your on-premises AD DS to AWS**

[View the template for new VPC](#)

[View the template for existing VPC](#)

**Note** The default CIDR ranges in this template are provided as examples to help you get started and can be modified to meet your specific requirements. Keep in mind that the provided CIDR blocks may overlap with your on-premises networks. If this is the case, you'll need use unique CIDR ranges to successfully deploy a VPN connection.

## Network Configuration:

Parameter label (name)	Default	Description
<b>Availability Zones</b> (AvailabilityZones)	<i>Requires input</i>	The list of Availability Zones to use for the subnets in the VPC. The Quick Start uses two Availability Zones from your list and preserves the logical order you specify.
<b>VPC CIDR</b> (VPCCIDR)	10.0.0.0/16	CIDR block for the Amazon VPC.
<b>Private Subnet 1 CIDR</b> (PrivateSubnet1CIDR)	10.0.0.0/19	CIDR block for the private subnet located in Availability Zone 1.
<b>Private Subnet 2 CIDR</b> (PrivateSubnet2CIDR)	10.0.32.0/19	CIDR block for the private subnet located in Availability Zone 2.
<b>Public Subnet 1 CIDR</b> (PublicSubnet1CIDR)	10.0.128.0/20	CIDR block for the public subnet located in Availability Zone 1.
<b>Public Subnet 2 CIDR</b> (PublicSubnet2CIDR)	10.0.144.0/20	CIDR block for the public subnet located in Availability Zone 2.
<b>Allowed Remote Desktop Gateway External Access CIDR</b> (RDGWCIDR)	0.0.0.0/0	Allowed CIDR block for external access to the Remote Desktop Gateway instances.

## Amazon EC2 Configuration:

Parameter label (name)	Default	Description
<b>Key Pair Name</b> (KeyPairName)	<i>Requires input</i>	Public/private key pair, which allows you to connect securely to your instance after it launches. When you created an AWS account, this is the key pair you created in your preferred region.
<b>NAT Instance Type</b> (NATInstanceType)	t2.small	EC2 instance type for the NAT instances. NAT instances are used only if the region doesn't support NAT gateways.
<b>Domain Controller 1 Instance Type</b> (ADServer1InstanceType)	m4.xlarge	EC2 instance type for the first Active Directory instance.
<b>Domain Controller 1 NetBIOS Name</b> (ADServer1NetBIOSName)	DC1	NetBIOS name of the first Active Directory server. This can be up to 15 characters long.
<b>Domain Controller 1 Private IP Address</b> (ADServer1PrivateIp)	10.0.0.10	Fixed private IP for the first Active Directory server located in Availability Zone 1.

Parameter label (name)	Default	Description
<b>Domain Controller 2 Instance Type</b> (ADServer2InstanceType)	m4.xlarge	EC2 instance type for the second Active Directory instance.
<b>Domain Controller 2 NetBIOS Name</b> (ADServer2NetBIOSName)	DC2	NetBIOS name of the second Active Directory server. This can be up to 15 characters long.
<b>Domain Controller 2 Private IP Address</b> (ADServer2PrivateIp)	10.0.32.10	Fixed private IP for the second Active Directory server located in Availability Zone 2.
<b>Remote Desktop Gateway 1 Instance Type</b> (RDGW1InstanceType)	t2.large	EC2 instance type for the first Remote Desktop Gateway.
<b>Remote Desktop Gateway 1 NetBIOS Name</b> (RDGW1NetBIOSName)	RDGW1	NetBIOS name of the first Remote Desktop Gateway. This can be up to 15 characters long.
<b>Remote Desktop Gateway 1 Private IP Address</b> (RDGW1PrivateIP)	10.0.128.11	Fixed private IP for the first Remote Desktop Gateway located in Availability Zone 1.
<b>Remote Desktop Gateway 2 Instance Type</b> (RDGW2InstanceType)	t2.large	EC2 instance type for the second Remote Desktop Gateway.
<b>Remote Desktop Gateway 2 NetBIOS Name</b> (RDGW2NetBIOSName)	RDGW2	NetBIOS name of the second Remote Desktop Gateway. This can be up to 15 characters long.
<b>Remote Desktop Gateway 2 Private IP Address</b> (RDGW2PrivateIP)	10.0.144.11	Fixed private IP for the second Remote Desktop Gateway located in Availability Zone 1.

### Microsoft Remote Desktop Gateway Configuration:

Parameter label (name)	Default	Description
<b>AdminUser</b>	StackAdmin	User name for the new local administrator account.
<b>AdminPassword</b>	<i>Requires input</i>	Password for the administrative account. This must be a <a href="#">complex password</a> that's at least 8 characters long.
<b>Domain DNS Name</b> (DomainDNSName)	example.com	Fully qualified domain name (FQDN) of the forest root domain.

- **Scenario 3: Parameters for deploying AD DS with AWS Directory Service**

[View the template for new VPC](#)

[View the template for existing VPC](#)

### Network Configuration:

Parameter label (name)	Default	Description
<b>Availability Zones</b> (AvailabilityZones)	<i>Requires input</i>	The list of Availability Zones to use for the subnets in the VPC. The Quick Start uses two Availability Zones from your list and preserves the logical order you specify.
<b>VPC CIDR</b> (VPCCIDR)	10.0.0.0/16	CIDR block for the Amazon VPC.
<b>Private Subnet 1 CIDR</b> (PrivateSubnet1CIDR)	10.0.0.0/19	CIDR block for the private subnet located in Availability Zone 1.
<b>Private Subnet 2 CIDR</b> (PrivateSubnet2CIDR)	10.0.32.0/19	CIDR block for the private subnet located in Availability Zone 2.
<b>Public Subnet 1 CIDR</b> (PublicSubnet1CIDR)	10.0.128.0/20	CIDR block for the public subnet located in Availability Zone 1.
<b>Public Subnet 2 CIDR</b> (PublicSubnet2CIDR)	10.0.144.0/20	CIDR block for the public subnet located in Availability Zone 2.
<b>Allowed Remote Desktop Gateway External Access CIDR</b> (RDGWCIDR)	0.0.0.0/0	Allowed CIDR block for external access to the Remote Desktop Gateway instances.

### Amazon EC2 Configuration:

Parameter label (name)	Default	Description
<b>Key Pair Name</b> (KeyPairName)	<i>Requires input</i>	Public/private key pair, which allows you to connect securely to your instance after it launches. When you created an AWS account, this is the key pair you created in your preferred region.
<b>NAT Instance Type</b> (NATInstanceType)	t2.small	EC2 instance type for the NAT instances. NAT instances are used only if the region doesn't support NAT gateways.
<b>Remote Desktop Gateway 1 Instance Type</b> (RDGW1InstanceType)	t2.large	EC2 instance type for the first Remote Desktop Gateway.

Parameter label (name)	Default	Description
<b>Remote Desktop Gateway 1 NetBIOS Name</b> (RDGW1NetBIOSName)	RDGW1	NetBIOS name of the first Remote Desktop Gateway. This can be up to 15 characters long.
<b>Remote Desktop Gateway 1 Private IP Address</b> (RDGW1PrivateIP)	10.0.128.11	Fixed private IP for the first Remote Desktop Gateway located in Availability Zone 1.
<b>Remote Desktop Gateway 2 Instance Type</b> (RDGW2InstanceType)	t2.large	EC2 instance type for the second Remote Desktop Gateway.
<b>Remote Desktop Gateway 2 NetBIOS Name</b> (RDGW2NetBIOSName)	RDGW2	NetBIOS name of the second Remote Desktop Gateway. This can be up to 15 characters long.
<b>Remote Desktop Gateway 2 Private IP Address</b> (RDGW2PrivateIP)	10.0.144.11	Fixed private IP for the second Remote Desktop Gateway located in Availability Zone 1.

### Microsoft Active Directory Configuration:

Parameter label (name)	Default	Description
<b>Domain DNS Name</b> (DomainDNSName)	example.com	Fully qualified domain name (FQDN) of the forest root domain.
<b>Domain NetBIOS Name</b> (DomainNetBIOSName)	example	NetBIOS name of the domain for users of earlier versions of Windows. This can be up to 15 characters long.
<b>Domain Admin Password</b> (DomainAdminPassword)	<i>Requires input</i>	Password for the domain administrator user. This must be a <a href="#">complex password</a> that's at least 8 characters long.

- On the **Options** page, you can [specify tags](#) (key-value pairs) for resources in your stack and [set advanced options](#). When you're done, choose **Next**.
- On the **Review** page, review and confirm the template settings. Under **Capabilities**, select the check box to acknowledge that the template will create IAM resources.
- Choose **Create** to deploy the stack.
- Monitor the status of the stack. When the status is **CREATE\_COMPLETE**, the AD DS cluster is ready.



## Step 3. Post-Deployment Tasks (Scenario 2 Only)

If you're extending your on-premises AD DS to the AWS cloud ([scenario 2](#)), you'll need to perform the following tasks manually, after the stack has been successfully created:

1. Connect your on-premises network to the Amazon VPC using AWS Direct Connect or a VPN connection.
2. Add domain controllers to the AWS cloud to provide a reliable, low-latency network connection for resources in AWS that need access to your AD DS.
3. Configure your on-premises Active Directory Sites and Services to include sites and subnets that represent the Availability Zones within your Amazon VPC.
4. Promote the Windows Server instances in the private subnet 1 and private subnet 2 to domain controllers in your Active Directory domain.
5. Ensure that instances can resolve names via AD DNS by using one of these methods:
  - Statically assign AD DNS servers on Windows instances.  
—or—
  - Set the `domain-name-servers` field in a new DHCP options set in your Amazon VPC to include your AWS-based domain controllers hosting Active Directory DNS.

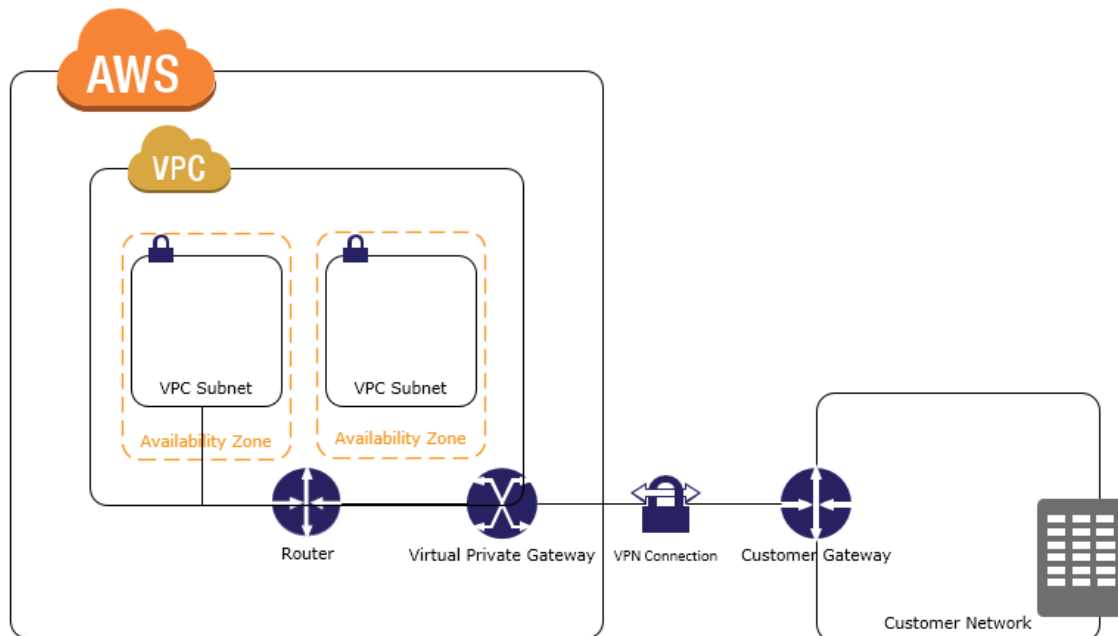
The following sections provide more information about these post-deployment tasks.

### Connecting Your On-Premises Network to Amazon VPC

By default, instances that you launch into a virtual private cloud can't communicate with your own network. To extend your existing AD DS into the AWS cloud, you'll need to extend your on-premises network to the Amazon VPC. We'll discuss two ways to do this: by using IPsec Virtual Private Network (VPN) tunnels or by using AWS Direct Connect.

#### *Using IPsec VPN Tunnels*

The most common scenario for extending your on-premises network to your Amazon VPC is through IPsec VPN tunnels. Within the Amazon VPC, you can create a virtual private gateway that acts as a VPN concentrator on the Amazon side of the VPN tunnel. A customer gateway is the anchor on your side of that connection. The customer gateway can be a physical device or a software appliance.

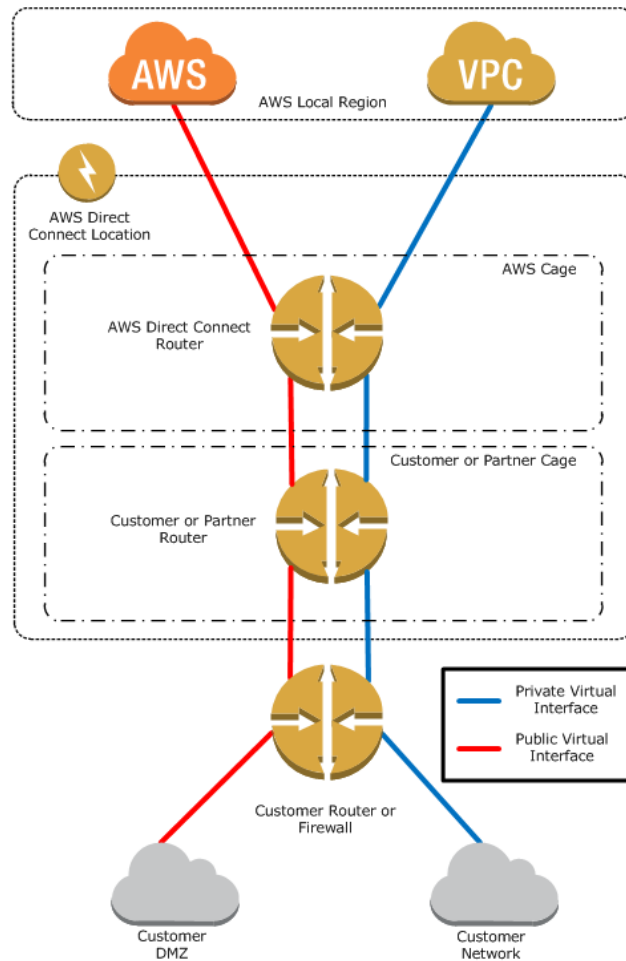


**Figure 10: Single VPN Connection from Your On-Premises Network to Your Amazon VPC**

Multiple VPN configuration options are available, including the ability to use multiple on-premises customer gateways and configuring redundant VPN connections to provide failover. For details, see [VPN Configuration Examples](#) in the *Amazon VPC User's Guide*. Details about which hardware or software appliances you can use are available in the [Customer Gateway devices we've tested](#) and [Requirements for your customer gateway](#) sections of the *Amazon VPC Network Administrator Guide*.

### *Using AWS Direct Connect*

AWS Direct Connect links your internal network to an AWS Direct Connect location over a standard 1 gigabit or 10 gigabit Ethernet fiber-optic cable. One end of the cable is connected to your router, the other to an AWS Direct Connect router. With this connection in place, you can create virtual interfaces directly to the AWS cloud (for example, to Amazon EC2, to Amazon S3, and to Amazon VPC), bypassing Internet service providers in your network path.



**Figure 11: How AWS Direct Connect Interfaces with Your Network**

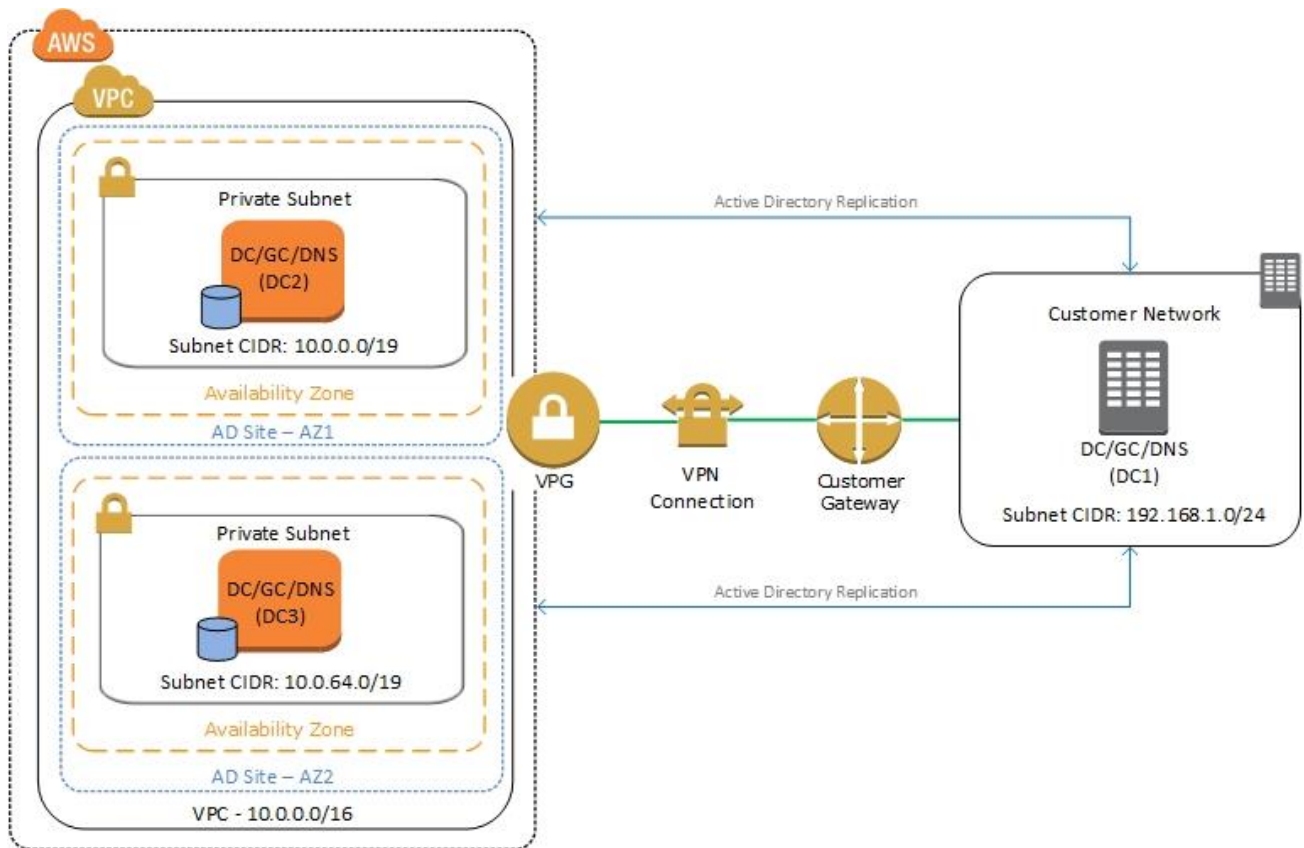
When you choose AWS Direct Connect to extend your on-premises network to the cloud, you should consider configuring two dedicated connections for maximum redundancy. There are different configuration choices available when you provision two dedicated connections, including active/active (BGP multipath), and active/passive (failover).

In a failover configuration, only one connection link handles traffic. If that link becomes unavailable, the standby connection link becomes active. We recommend that you configure both connection links as active, because this will help ensure that network traffic is load-balanced across both connections. In an active configuration, if one connection link becomes unavailable, all traffic is routed through the other link. For implementation details, see [Getting Started](#) in the *AWS Direct Connect User Guide*.

## Deploying Additional Domain Controllers in the AWS Cloud

Although you can use AWS Direct Connect or a VPN connection to provide access to on-premises resources from the Amazon VPC, we recommend that you also add domain controllers to the AWS cloud. Additional domain controllers provide a reliable, low-latency network connection for resources in AWS that need access to your AD DS. They can also maintain availability for AD DS in the AWS cloud in the event of an on-premises infrastructure outage.

In the architecture shown in Figure 12, a single Active Directory forest has been extended from an on-premises deployment into an Amazon VPC using a VPN connection. Within the Amazon VPC, additional domain controllers configured as global catalog and DNS servers are deployed in the existing Active Directory forest.



**Figure 12: Single AD Forest with a Domain Controller on Premises and in an Amazon VPC**

In this type of environment, the customer network will already be defined in Active Directory Sites and Services. For example, there will already be a site definition that corresponds to the on-premises network, along with a subnet definition for the 192.168.1.0/24 network. The next step is to configure Active Directory Sites and Services to support the network components located in the Amazon VPC.

## Configuring Active Directory Sites and Services

Additional Active Directory sites should be created to reference the Availability Zones in AWS. The 10.0.0.0/19 and 10.0.64.0/19 CIDR blocks used by the Amazon VPC subnets should be added to Active Directory Sites and Services. The subnets can then be associated with the AD DS site definition for each AWS Availability Zone. Additional subnets for web, application, and database tiers in the Amazon VPC can be mapped to each AWS site object. Both the on-premises site and the site in the AWS cloud can be mapped to a site link, which can be configured to replicate at custom intervals or during a specific time of day, if needed.

By properly configuring Active Directory Sites and Services, you can help ensure that the AD DS queries and authentication requests that originate from the Amazon VPC are serviced by a local domain controller in the same AWS Availability Zone. This configuration reduces network latency and minimizes traffic that may otherwise need to travel across the VPN back to the on-premises infrastructure.

## Configuring DNS Resolution

After you've created an Amazon VPC and established connectivity to your on-premises network by using AWS Direct Connect or a VPN connection, your next step is to launch Windows instances to act as domain controllers. In order to join the on-premises Active Directory domain and promote your Windows instances to domain controllers, you'll need to ensure that DNS resolution is configured appropriately.

As discussed previously, by default, instances launched into the Amazon VPC will be assigned an Amazon-provided DNS server, which will not provide DNS resolution for your on-premises infrastructure. To address this, you can do one of two things:

- Manually assign DNS server settings on the Windows instances. This static DNS setting would initially point to the on-premises Active Directory DNS server. After promoting the instance to a domain controller, you could modify the setting to use a cloud-based

Active Directory DNS server IP address to prevent subsequent DNS queries from traversing the link back to the on-premises environment.

—or—

- Initially configure the Amazon VPC DHCP options set to assign your on-premises Active Directory DNS server IP address to your instances launched into the Amazon VPC. After the Windows instances have been joined to the domain and promoted to domain controllers, you can create a new DHCP options set to assign the IP address of the Active Directory DNS server instances running in AWS.

## Troubleshooting

When you deploy the Quick Start, if you encounter a **CREATE\_FAILED** error instead of the **CREATE\_COMPLETE** status message, we recommend that you relaunch the template with **Rollback on failure** set to **No**. (This setting is under **Advanced** in the AWS CloudFormation console, **Options** page.) With this setting, the stack's state will be retained and the instance will be left running, so you can troubleshoot the issue. (You'll want to look at the log files in `%ProgramFiles%\Amazon\EC2ConfigService` and `C:\cfn\log`.)

**Important** When you set **Rollback on failure** to **No**, you'll continue to incur AWS charges for this stack. Please make sure to delete the stack when you've finished troubleshooting.

The following table lists specific **CREATE\_FAILED** error messages you might encounter.

<b>CREATE_FAILED</b> <b>error message</b>	<b>Possible cause</b>	<b>What to do</b>
<b>API: ec2: RunInstances</b> <b>Not authorized for</b> <b>images: ami-ID</b>	The template is referencing an AMI that has expired	We refresh AMIs on a regular basis, but our schedule isn't always synchronized with AWS AMI updates. If you get this error message, notify us, and we'll update the AMI IDs in the template.  You can also download the template and update the mappings in <code>AWSWinRegionMap</code> with the latest AMI ID for your region.

<b>CREATE_FAILED error message</b>	<b>Possible cause</b>	<b>What to do</b>
<b>We currently do not have sufficient m4.large capacity in the AZ you requested</b>	One of the instances requires a larger instance type	Switch to an instance type that supports higher capacity, or complete the <a href="#">request form</a> in the AWS Support Center to increase the Amazon EC2 limit for the instance type or region. Limit increases are tied to the region they were requested for.
<b>Instance ID did not stabilize</b>	You have exceeded your IOPS for the region	Request a limit increase by completing the <a href="#">request form</a> in the AWS Support Center.
<b>System Administrator password must contain at least 8 characters</b>	The master password contains \$ or other special characters	Change the password for the <b>RestoreModePassword</b> or <b>DomainAdminPassword</b> parameter and then relaunch the Quick Start.  You must use a <a href="#">complex password</a> that is at least 8 characters long, consisting of uppercase and lowercase letters and numbers. Avoid using special characters such as @ or \$.

For additional information, see [Troubleshooting AWS CloudFormation](#) on the AWS website.

## Security

AWS provides a set of building blocks, including the Amazon EC2 and Amazon VPC services, that you can use to provision infrastructure for your applications. In this model, some security capabilities such as physical security are the responsibility of AWS and are highlighted in the [AWS security whitepaper](#). Other capabilities, such as controlling access to applications, are the responsibility of the application developer and the tools provided in the Microsoft platform.

If you have followed the automated deployment options in this guide, the necessary security groups are configured for you by the provided AWS CloudFormation templates and are listed here for your reference.

Security group	Associated with	Inbound source	Port(s)
<b>DomainControllerSG1</b>	DC1	VPCCIDR	TCP5985, TCP53, UDP53, TCP80
		DomainMemberSG	UDP123, TCP135, UDP138, TCP445, UDP445, TCP464, UDP464, TCP49152-65535, UDP49152-65535, TCP389, UDP389, TCP636, TCP3268, TCP3269, TCP88, UDP88, UDP67, UDP2535, TCP9389
		PrivateSubnet2CIDR (subnet where the second DC is deployed)	UDP123, TCP135, UDP137, UDP138, TCP445, UDP445, TCP464, UDP464, TCP49152-65535, UDP49152-65535, TCP389, UDP389, TCP636, TCP3268, TCP3269, TCP88, UDP88, UDP67, UDP2535, UDP5355, UDP137, TCP139, TCP5722, TCP9389
		PublicSubnet1CIDR (subnet where the Remote Desktop Gateway is deployed in Availability Zone 1)	TCP3389, (ICMP -1)
		PublicSubnet2CIDR (subnet where the Remote Desktop Gateway is deployed in Availability Zone 2)	TCP3389, (ICMP -1)
<b>DomainControllerSG2</b>	DC2	VPCCIDR	TCP5985, TCP53, UDP53, TCP80
		DomainMemberSG	UDP123, TCP135, UDP138, TCP445, UDP445, TCP464, UDP464, TCP49152-65535, UDP49152-65535, TCP389, UDP389, TCP636, TCP3268, TCP3269, TCP88, UDP88, UDP67, UDP2535, TCP9389
		PrivateSubnet1CIDR (subnet where the first DC is deployed)	UDP123, TCP135, UPD137, UDP138, TCP445, UDP445, TCP464, UDP464, TCP49152-65535, UDP49152-65535, TCP389, UDP389, TCP636, TCP3268, TCP3269, TCP88, UDP88, UDP67, UDP2535, UDP5355, UDP137, TCP139, TCP5722, TCP9389
		PublicSubnet1CIDR (subnet where the Remote Desktop Gateway is deployed in Availability Zone 1)	TCP3389, (ICMP -1)



Security group	Associated with	Inbound source	Port(s)
		PublicSubnet2CIDR (subnet where the Remote Desktop Gateway is deployed in Availability Zone 2)	TCP3389, (ICMP -1)
<b>DomainMemberSG</b>	RDGW1, RDGW2	PrivateSubnet1CIDR (subnet where the primary DC is deployed)	TCP5985, TCP53, UDP53, TCP49152-65535, UDP49152-65535
		PrivateSubnet2CIDR (subnet where the secondary DC is deployed)	TCP5985, TCP53, UDP53, TCP49152-65535, UDP49152-65535
		PublicSubnet1CIDR (subnet where the Remote Desktop Gateway is deployed in Availability Zone 1)	TCP3389
		PublicSubnet2CIDR (subnet where the Remote Desktop Gateway is deployed in Availability Zone 2)	TCP3389
<b>RDGWSecurityGroup</b>	RDGW1, RDGW2	0.0.0.0/0 *	TCP3389

**\* Important** RDP should never be opened up to the entire Internet, not even temporarily or for testing purposes. For more information, see this [Amazon security bulletin](#). Always restrict ports and source traffic to the minimum necessary to support the functionality of the application. For more about securing Remote Desktop Gateway, see the [Securing the Microsoft Platform on Amazon Web Services](#) whitepaper.

# Additional Resources

## AWS services

- AWS CloudFormation  
<http://aws.amazon.com/documentation/cloudformation/>
- Amazon EC2 user guide for Windows  
<http://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/>
- Amazon VPC
  - User guide  
<http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/>
  - Basic scenarios  
[http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC\\_Scenarios.html](http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Scenarios.html)
  - Network administrator guide  
<http://docs.aws.amazon.com/AmazonVPC/latest/NetworkAdminGuide/>
- NAT Gateway  
<http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-nat-gateway.html>
- AWS Direct Connect  
<http://aws.amazon.com/documentation/directconnect/>
- AWS Directory Service  
<http://aws.amazon.com/documentation/directory-service/>

## Active Directory Domain Services

- Active Directory Domain Services  
<https://technet.microsoft.com/en-us/library/dd448614.aspx>
- Active Directory Sites and Services  
<https://technet.microsoft.com/library/cc730868.aspx>

## Deploying Microsoft software on AWS

- Microsoft on AWS  
<http://aws.amazon.com/microsoft/>

- Securing the Microsoft platform on AWS  
[http://media.amazonwebservices.com/AWS\\_Microsoft\\_Platform\\_Security.pdf](http://media.amazonwebservices.com/AWS_Microsoft_Platform_Security.pdf)

### Quick Start reference deployments

- AWS Quick Start home page  
<https://aws.amazon.com/quickstart/>
- Quick Start deployment guides  
<https://aws.amazon.com/documentation/quickstart/>
- Building a Modular and Scalable Virtual Network Architecture with Amazon VPC  
<http://docs.aws.amazon.com/quickstart/latest/vpc/>
- Microsoft Remote Desktop Gateway on AWS  
<http://docs.aws.amazon.com/quickstart/latest/rd-gateway/>

## Send Us Feedback

We welcome your questions and comments. Please post your feedback on the [AWS Quick Start Discussion Forum](#).

You can visit our [GitHub repository](#) to download the templates and scripts for this Quick Start, and to share your customizations with others.

# Document Revisions

Date	Change	In sections
<b>July 2016</b>	For all three scenarios, added option to deploy Quick Start in an existing VPC. Updated the templates to use NAT gateways and an updated VPC configuration.	<a href="#">Deployment Scenarios</a> <a href="#">Deployment Steps</a> Template updates
<b>April 2016</b>	Added a new scenario that uses AWS Directory Service for Microsoft AD to provision and manage AD DS. Also: Replaced NAT instances with the NAT Gateway service; upgraded to Windows Server 2012 R2 for AD domain and forest functionality; updated templates with parameter groups and labels to simplify usage.	<a href="#">Scenario 3</a> Additional updates throughout document to reflect new functionality Template updates
<b>September 2015</b>	In the sample templates, changed the default type for Active Directory and RD Gateway instances from <b>m3.xlarge</b> to <b>m4.xlarge</b> for better performance and price.	<a href="#">Deployment Steps</a> (template customization tables)
<b>March 2015</b>	Optimized the Amazon VPC subnet design implemented by the Quick Start to support expansion and to reduce complexity.	<a href="#">Amazon VPC recommendations</a> Architecture diagram updates
<b>November 2014</b>	In the sample templates, changed the default type for <b>NATInstanceType</b> to <b>t2.small</b> to support the EU (Frankfurt) region.	<a href="#">Deployment Steps</a> (template customization tables)
<b>March 2014</b>	Initial publication	—

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

### **Notices**

This document is provided for informational purposes only. It represents AWS's current product offerings and practices as of the date of issue of this document, which are subject to change without notice. Customers are responsible for making their own independent assessment of the information in this document and any use of AWS's products or services, each of which is provided "as is" without warranty of any kind, whether express or implied. This document does not create any warranties, representations, contractual commitments, conditions or assurances from AWS, its affiliates, suppliers or licensors. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

The software included with this guide is licensed under the Apache License, Version 2.0 (the "License"). You may not use this file except in compliance with the License. A copy of the License is located at <http://aws.amazon.com/apache2.0/> or in the "license" file accompanying this file. This code is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.