Getting Started with AWS

Getting Started with AWS

Table of Contents

Getting Started with AWS

Amazon Web Services (AWS) provides on-demand computing resources and services in the cloud, with pay-as-you-go pricing. For example, you can run a server on AWS that you can log on to, configure, secure, and run just as you would a server that's sitting in front of you. For more information, see What is Cloud Computing?

Using AWS resources instead of your own is like purchasing electricity from a power company instead of running your own generator, and it provides many of the same benefits: capacity exactly matches your need, you pay only for what you use, economies of scale result in lower costs, and the service is provided by a vendor experienced in running large-scale networks.

What Can I Do with AWS?

You can use AWS to make it easier to build and manage your websites and applications. The following are some common uses for AWS:

- Store public or private data.
- Host a static website. These websites use client-side technologies (such as HTML, CSS, and JavaScript) to display content that doesn't change frequently. A static website doesn't require server-side technologies (such as PHP and ASP.NET).
- Host a dynamic website, or web app. These websites include classic three-tier applications, with web, application, and database tiers.
- Support students or online training programs.
- Process business and scientific data.
- Handle peak loads.

For more information, see AWS Solutions.

How Do I Get Started?

The following documentation provides an overview of AWS and list of tutorials that demonstrate the common uses for AWS:

- AWS Overview (p. 3)

- AWS Tutorials (p. 21)

How Do I Access AWS?

AWS provides several ways to create and manage resources.

AWS Management Console
A web interface. To get started, see the Getting Started with the AWS Management Console.

AWS Command Line Interface (AWS CLI)
Commands for a broad set of AWS products. To get started, see AWS Command Line Interface User Guide.

Command Line Tools
Commands for individual AWS products. For more information, see Command Line Tools.

AWS Software Development Kits (SDK)
APIs that are specific to your programming language or platform. For more information, see SDKs.

Query APIs
Low-level APIs that you access using HTTP requests. For more information, see the API documentation for each service.

Pricing

AWS can offer significant cost savings compared to the equivalent on-premises infrastructure. You can use the AWS Simple Monthly Calculator to estimate what it would cost to use AWS.

Note that if you created your AWS account within the last 12 months, you are eligible for the AWS Free Tier.

For more information about managing your costs, see Pricing and the Economics Center.

AWS Overview

AWS offers a broad set of services that help you move faster, lower your costs, and scale your applications. The following documentation provides a high-level overview of the concepts that you should understand before you get started with AWS.

Contents

Regions and Availability Zones

Amazon has data centers in different areas of the world (for example, North America, Europe, and Asia). Correspondingly, AWS products are available to use in different *regions*. By placing resources in separate regions, you can design your website or app to be closer to specific customers or to meet legal or other requirements. Note that prices for AWS usage vary by region.

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 placing resources in separate Availability Zones, you can protect your website or app from the failure of a single location.

AWS resources can be tied to a region or tied to an Availability Zone. Not every region or Availability Zone supports every AWS resource. When you view your resources, you'll only see the resources tied to the region you've specified. This is because regions are isolated from each other, and we don't replicate resources across regions automatically.

The following table lists the AWS regions.

Name	Region Code
Asia Pacific (Seoul)	`ap-northeast-2`
Asia Pacific (Singapore)	`ap-southeast-1`
Asia Pacific (Sydney)	`ap-southeast-2`
Asia Pacific (Tokyo)	`ap-northeast-1`

Name	Region Code
EU (Frankfurt)	`eu-central-1`
EU (Ireland)	`eu-west-1`
South America (São Paulo)	`sa-east-1`
US East (N. Virginia)	`us-east-1`
US West (N. California)	`us-west-1`
US West (Oregon)	`us-west-2`

Security

AWS provides a secure global infrastructure, plus a range of features that you can use to secure your data in the cloud. The following are highlights:

- Physical access to AWS data centers is strictly controlled, monitored, and audited.
- Access to the AWS network is strictly controlled, monitored, and audited.
- You can manage the security credentials that enable users to access your AWS account using AWS Identity and Access Management (IAM). You can create fine-grained permissions to AWS resources and apply them to users or groups of users.
- You can apply ACL-type permissions on your data and can also use encryption of data at rest.
- You can set up a virtual private cloud (VPC), which is a virtual network that is logically isolated from other virtual networks in the AWS cloud. You can control whether the network is directly routable to the Internet.
- You control and configure the operating system on your virtual server.
- You can set up a security group, which acts as a virtual firewall to control the inbound and outbound traffic for your virtual servers.
- You can specify a key pair when you launch your virtual server, which is used to encrypt your login information. When you log in to your virtual server, you must present the private key of the key pair to decrypt the login information.

For more information, see the following resources:

- AWS Security Center
- AWS Overview of Security Processes
- AWS Security Best Practices

AWS Product Categories

AWS offers a broad set of services. To learn more about each AWS product category, see the following:

- Compute and Networking Services (p. 5)
- Storage and Content Delivery Services (p. 9)
- Security and Identity Services (p. 12)
- Database Services (p. 13)
- Analytics Services (p. 16)

Compute and Networking Services for AWS

AWS provides a variety of computing and networking services to meet the needs of your applications. You can provision virtual servers, set up a firewall, configure Internet access, allocate and route IP addresses, and scale your infrastructure to meet increasing demand.

You can use the compute and networking services with the storage, database, and application services to provide a complete solution for computing, query processing, and storage across a wide range of applications.

Contents

Key Services

The following are the key compute and networking services:

Amazon EC2
  Provides virtual servers in the AWS cloud.
Amazon VPC
  Provides an isolated virtual network for your virtual servers.
Elastic Load Balancing
  Distributes network traffic across your set of virtual servers.
Auto Scaling
  Automatically scales your set of virtual servers based on changes in demand.
Amazon Route 53
  Routes traffic to your domain name to a resource, such as a virtual server or a load balancer.
AWS Lambda
  Runs your code on virtual servers from Amazon EC2 in response to events.
Amazon ECS
  Provides Docker containers on virtual servers from Amazon EC2.

Key Concepts

The following are concepts that you should understand before using the compute and networking services.

Instances and AMIs

Amazon Elastic Compute Cloud (Amazon EC2) provides resizeable computing capacity—literally, servers in Amazon's data centers—that you use to build and host your software systems. 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 an *instance*, which is a copy of the AMI running as a virtual server on a host computer in Amazon's data center. You can launch multiple instances from an AMI, as shown in the following figure.
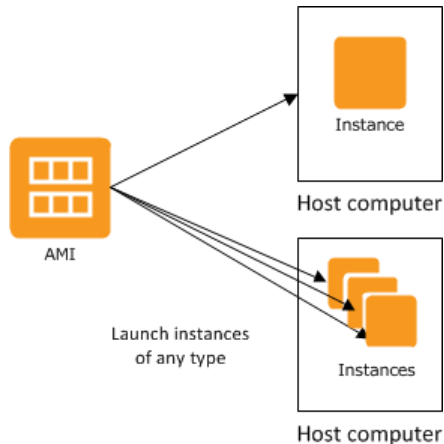
When you launch an instance, you select an *instance type*, which determines the hardware capabilities (such as memory, CPU, and storage) of the host computer for the instance. You can access your instance using its assigned public DNS name or public IP address. The public DNS names for instances are as follows:

The US East (N. Virginia) region

```
ec2-public_ip.compute-1.amazonaws.com
```

Other regions

```
ec2-public_ip.region_code.compute.amazonaws.com
```

Your 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.

You start from an existing AMI that most closely meets your needs, log on to the instance, and then customize the instance with additional software and settings. You can save this customized configuration as a new AMI, which you can then use to launch new instances whenever you need them.

VPCs and Subnets

A *virtual private cloud* (VPC) is a virtual network dedicated to your AWS account. It is logically isolated from other virtual networks in the AWS cloud, providing security and robust networking functionality for your compute resources. A VPC closely resembles a traditional network that you'd operate in your own data center, with the benefits of using the scalable infrastructure of AWS.

A *subnet* is a segment of a VPC's IP address range that you can launch instances into. Subnets enable you to group instances based on your security and operational needs. To enable instances in a subnet to reach the Internet and AWS services, you must add an Internet gateway to the VPC and a route table with a route to the Internet to the subnet.

We recommend that you launch your EC2 instances into a VPC. Note that if you created your AWS account after 2013-12-04, you have a default VPC and you must launch EC2 instances into a default or a nondefault VPC.

Security Groups

A *security group* acts as a virtual firewall for your instance to control inbound and outbound traffic. You can specify one or more security groups when you launch your instance. When you create a security

group, you add *rules* that control the inbound traffic that's allowed, and a separate set of rules that control the outbound traffic. All other traffic is discarded. You can modify the rules for a security group at any time and the new rules are automatically enforced.
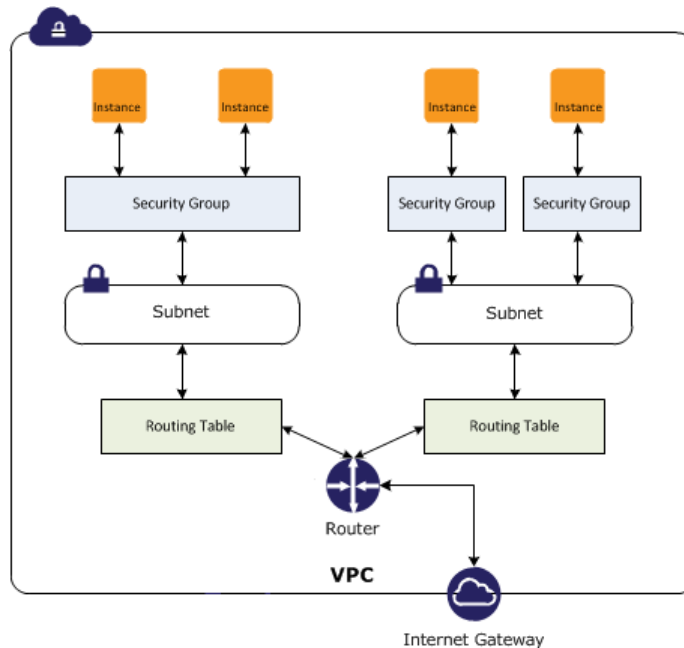
Amazon Route 53 Hosted Zones

Amazon Route 53 is a highly available and scalable cloud Domain Name System (DNS) web service. It is designed as an extremely reliable and cost-effective way to route visitors to websites by translating domain names (such as `www.example.com`) into the numeric IP addresses (such as `192.0.2.1`) that computers use to connect to each other. AWS assigns URLs to your AWS resources, such as your EC2 instances. However, you might want a URL that is easy for your users to remember. For example, you can map your domain name to your AWS resource. If you don't have a domain name, you can search for available domains and register them using Amazon Route 53. If you have an existing domain name, you can transfer it to Amazon Route 53.

Amazon Route 53 enables you to organize your DNS records using *hosted zones*. When you create a hosted zone, you receive four name servers to help ensure a high level of availability.

Auto Scaling Groups

Auto Scaling supports groups of virtual servers, an *Auto Scaling group* that can grow or shrink on demand.
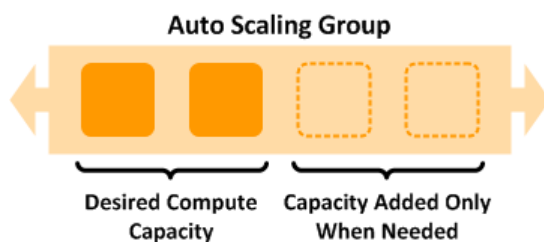
Load Balancer

A *load balancer* distributes traffic to multiple instances. You can achieve even higher levels of fault tolerance by using your load balancer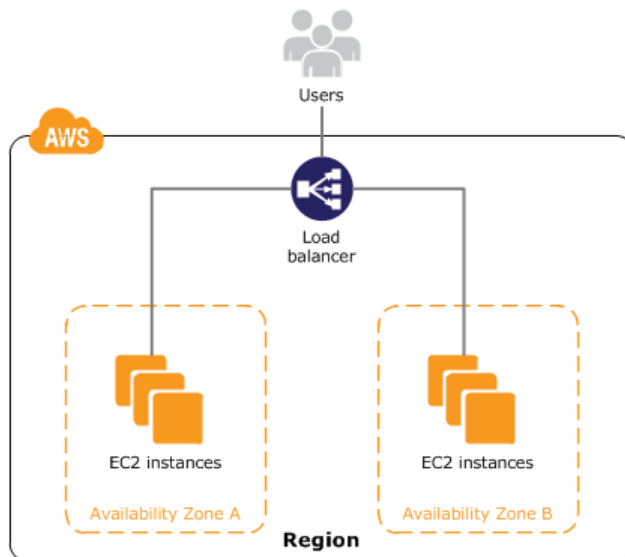 with instances in multiple Availability Zones. As instances are launched and terminated, the load balancer automatically directs traffic to the running instances. Elastic Load Balancing also performs health checks on each instance. If an instance is not responding, the load balancer can automatically redirect traffic to the healthy instances.

Architecture

The following diagram shows an example architecture for your compute and networking services. There are EC2 instances in public and private subnets. Access to the instances in the public subnets over protocols like SSH or RDP is controlled by one or more security groups. Security groups also control whether the instances can talk to each other. The Auto Scaling group maintains a fleet of EC2 instances that can scale to handle the current load. This Auto Scaling group spans multiple Availability Zones to protect against the potential failure of a single Availability Zone. The load balancer distributes traffic evenly among the EC2 instances in the Auto Scaling group. When the Auto Scaling group launches or terminates instances based on load, the load balancer automatically adjusts accordingly. Amazon Route 53 provides secure and reliable routing of your domain name to your infrastructure hosted on AWS.
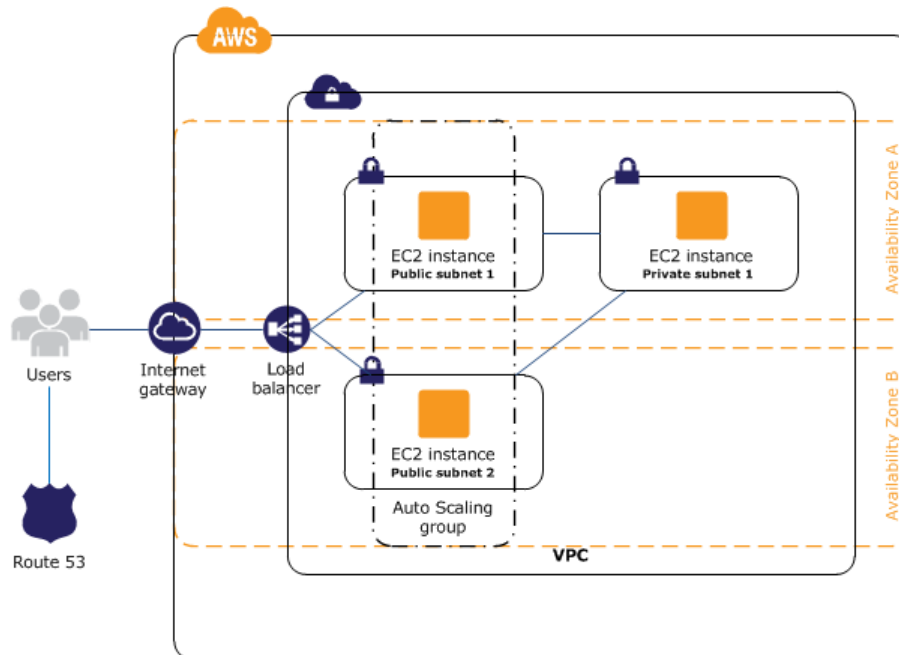
Documentation

For more information, see the following documentation:

- Amazon EC2
- Amazon ECS
- Amazon Route 53
- Amazon VPC
- Auto Scaling
- AWS Lambda
- Elastic Load Balancing

Storage and Content Delivery Services for AWS

AWS provides a variety of services to meet your storage needs.

Contents

Key Services

The following are the key storage and content delivery services:

Amazon S3
　　Scalable storage in the AWS cloud.

CloudFront
    A global content delivery network (CDN).
Amazon EBS
    Network attached storage volumes for your virtual servers.
Amazon Glacier
    Low-cost archival storage.

Key Concepts

The following are concepts that you should understand before using the storage and content delivery services.

Buckets and Objects

Amazon Simple Storage Service (Amazon S3) is an Internet-scale data storage service that handles vast amounts of data and large numbers of concurrent accesses. You store your data in containers called *buckets*. You can control who has permissions for each bucket. A bucket can contain any number of objects of any type, such as HTML pages, source code files, image files, and encrypted data. The name of each bucket must be globally unique.

You can group objects using *folders*. These folders are included in the key name for an object. For example, `mysite/html/default.html`. Each object can be accessed through its unique URL:

```
protocol://domain/bucket_name/object_key
```

protocol
    The protocol is either `http` or `https`.
domain
    The domain is `s3.amazonaws.com` for buckets created in US Standard or `s3-region_code.amazonaws.com` for buckets created in a region.
bucket_name
    The name of the bucket.
object_key
    The full key name for the object.

For example, `https://s3.amazonaws.com/mysite/html/default.html` (US Standard) or `https://s3-us-west-2.amazonaws.com/my-bucket/image.png` (US West (Oregon)).

CloudFront Distributions

Use Amazon CloudFront to create a content delivery network (CDN) that makes your website content available from data centers around the world, called *edge locations*. You store your content on an *origin server*, such as an Amazon S3 bucket or an HTTP server running on an EC2 instance. You create a CloudFront *distribution*, associate the distribution with the origin server, and then use a CloudFront URL to access your content:

```
http://distribution_id.cloudfront.net/file.ext
```

Alternatively, you can associate your own domain name with your CloudFront distribution.

When a user accesses an object that's part of a CloudFront distribution, CloudFront checks whether the object is already in a cache that's near the user. If it is, CloudFront serves the content from the cache; otherwise, CloudFront copies the requested content from the origin server to the cache.

Instance Store Volumes

Amazon EC2 provides instance store volumes for many types of EC2 instances. The disks for these volumes are physically attached to the same host computer as the instance, so access to these volumes is very fast. There is no additional charge to use the instance store volumes provided by the instance type, but you can incur data transfer changes for data transferred to and from the instance.

An instance store volume is usable only from a single instance and only during its lifetime. For instance types that support multiple instance store volumes, you can stripe your data across multiple volumes. The data on an instance store volume is erased when the instance terminates. If an application uses instance store volumes for persistent data, it must periodically replicate the data or copy it to durable storage.

Amazon EBS Volumes

Amazon Elastic Block Store (Amazon EBS) provides network-attached storage (NAS) for use with your EC2 instances. After you create, attach, and mount an Amazon EBS volume to your instance, you can use it just as you would use a physical hard drive on your computer. Each volume can be attached to only one EC2 instance, but you can detach an Amazon EBS volume from one EC2 instance and attach it to another. You can attach multiple Amazon EBS volumes to an EC2 instance, and you can also stripe your data across multiple volumes.

You can back up the data on your Amazon EBS volumes by creating snapshots, which are stored in Amazon S3. You can create a new Amazon EBS volume from a snapshot and then attach it to an EC2 instance.

Vaults and Archives

Amazon Glacier provides low-cost, durable storage that is suitable for data archiving and backup. You create a *vault*, which is a container for archives. An *archive* is any object that you store in the vault, such as a photo, video, or document. Each archive has a unique ID and an optional description. While you can create and delete a vault using the AWS Management Console, you must upload, download, and delete archives programmatically. Note that these operations typically take 3 to 5 hours to complete.

Usage Scenarios

The following table summarizes the common usage scenarios for the AWS storage options. For more information, see AWS Storage Options.

Storage Option	Usage
Amazon S3	Use for a wide range of scenarios, from backing up your data, to storing your images and videos (to be accessed directly or through a CDN), to hosting static websites.
Amazon EBS	Use for data that changes frequently and must persist. For example, use EBS volumes as the primary storage for a database or file system, or for applications that require access to raw block-level storage.
Instance store volumes	Use instance store volumes for temporary storage of data that changes frequently, such as buffers, caches, or scratch data, or data that is replicated across a fleet of instances.

Storage Option	Usage
	If your data must persist beyond the lifetime of the EC2 instance, use Amazon EBS volumes instead.
Amazon CloudFront	Use CloudFront edge locations to improve the speed of your website. This is especially important if your website displays large media files, such as high-resolution images, audio, or video.
AWS Import/Export	Use AWS Import/Export to transfer data to or from AWS (Amazon S3 buckets, Amazon EBS snapshots, or Amazon Glacier vaults), using portable storage devices. This is a good option if it would be too costly or slow (more than a week) to transfer your data to AWS over the Internet.
AWS Storage Gateway	Use AWS Storage Gateway to provide a seamless and secure connection between an on-premises software appliance and Amazon S3. This is useful for corporate file sharing, enabling existing on-premises backup applications to store primary backups in Amazon S3, and data mirroring.
Amazon Glacier	Use Amazon Glacier when cost is paramount, you need the data infrequently, and you can wait several hours for the data to be retrieved. If you need fast or frequent access to your data, use Amazon S3 instead.

Documentation

For more information, see the following documentation:

- Amazon CloudFront
- Amazon EBS
- Amazon Glacier
- Amazon S3
- AWS Import/Export
- AWS Storage Gateway

Security and Identity Services for AWS

AWS provides security and identity services that help you secure your data and systems in the cloud.

Key Services

The following are the key security and identity services:

AWS Identity and Access Management
> Manage user access to AWS resources through policies.

AWS Directory Service
> Manage user access to AWS through your existing Microsoft Active Directory, or a directory you create in the AWS cloud.

Key Concepts

The following are concepts that you should understand before using the security and identity services.

User Permissions

AWS Identity and Access Management (IAM) enables you to manage users and user permissions in AWS. You can control which users can access which AWS resources.

Directory Services

AWS Directory Service enables you to grant access to AWS to directory users and groups. You can create a new directory in the cloud, or connect to an existing on-premises directory, such as Microsoft Active Directory.

Documentation

For more information, see the following documentation:

- AWS Directory Service
- AWS Identity and Access Management

Database Services for AWS

AWS provides fully-managed relational and NoSQL database services, as well as in-memory caching as a service and a petabyte-scale data-warehouse solution.

Contents

Key Services

The following are the key database services:

Amazon RDS
> Provides managed relational databases.

Amazon Redshift
> A fast, fully-managed, petabyte-scale data warehouse.

Amazon DynamoDB
> Provides managed NoSQL databases.

Amazon ElastiCache
    An in-memory caching service.

Key Concepts

The following are concepts that you should understand before using the database services.

Relational Database

A relational database is organized into tables that are related to each other by key values. It is the traditional type of database.

If your application primarily indexes and queries data and doesn't require joins or complex transactions, consider a NoSQL database instead. If you have large binary files (audio, video, and image), consider storing the files in Amazon Simple Storage Service (Amazon S3) and storing the metadata for the files in your database.

NoSQL Database

A NoSQL database offers schema flexibility (for example, to provide JSON document model support), fast read and write performance, virtually limitless scaling, and high availability. NoSQL database tables are schemaless and can be used to store JSON-style documents or key-value pairs. This functionality makes NoSQL databases ideal for managing structured or unstructured data.

If your application requires joins or complex transactions, consider a relational database instead. If you have large binary files (audio, video, and image), consider storing the files in Amazon S3 and storing the metadata for the files in your database.

DB Instance

A DB instance is the basic building block of Amazon Relational Database Service (Amazon RDS); it is an isolated database environment in the cloud. A DB instance can contain multiple databases.

When you launch a DB instance, you select a database engine (MySQL, PostgreSQL, Oracle, or Microsoft SQL Server) and a DB instance class, which determines the compute and memory capabilities for the DB instance. You also specify a security group for the DB instance. The firewall for the DB instance prevents any access to its databases other than what you've granted through the rules for the security group. You can launch your DB instance in a virtual private cloud (VPC) for additional network access control. You can also use the security features of your DB engine the same way that you'd use them on your local network.

In a Multi-AZ deployment, Amazon RDS automatically provisions and maintains a synchronous *standby replica* in a different Availability Zone. The primary DB instance is synchronously replicated across Availability Zones to a standby replica to provide data redundancy, eliminate I/O freezes, and minimize latency spikes during system backups. In the event of a planned or unplanned outage of your DB instance, Amazon RDS automatically switches to a standby replica.

Amazon RDS can use replication functionality built into the PostgreSQL or MySQL DB engine to create a special type of DB instance called a *read replica* from a source DB instance.

Amazon RDS provides two different methods for backing up and restoring your DB instances: automated backups and user-initiated backups known as *DB snapshots*.

Usage Scenarios

The following table summarizes the common usage scenarios for the AWS database options. For more information, see Running Databases on AWS.

Database Option	Usage
Amazon RDS	Provides a fully-managed relational database that scales to large datasets. It is easy to scale the database storage and compute resources and provide read replicas. You have a choice of database engines: MySQL, PostgreSQL, Oracle, or Microsoft SQL Server. Most software designed for use with these databases should work unmodified with Amazon RDS. If you need a specific relational database that isn't supported by Amazon RDS, host the database on Amazon EC2 instead.
Amazon Redshift	Provides a fast, fully-managed, petabyte-scale data warehouse that makes it easy and cost-effective to analyze a vast amount of data. If you need to perform online transaction processing, use Amazon RDS instead.
Amazon DynamoDB	Provides a fully-managed NoSQL database with fast performance at a low cost. Common use cases include mobile apps, gaming, digital ad serving, live events, metadata storage for Amazon S3 objects, e-commerce shopping carts, and web session management. If you have large objects (for example, 10 GB or larger in size), store those objects in Amazon S3 and maintain the metadata in Amazon DynamoDB.
Amazon ElastiCache	Provides a fast, fully managed, in-memory cache in the cloud. You have a choice of caching engines: Memcached and Redis. Common use cases include improving performance by caching the results of I/O-intensive queries, managing web session data, and caching dynamically-generated web pages. If you need data persistence, use DynamoDB instead.
Hosted on Amazon EC2	Enables you to manage the software, compute resources, and storage resources for your database with complete control. For best performance, select the right EC2 instance type and EBS volume type for your scenario. You can also increase the number of EBS volumes and use striping to increase performance. If you are running a database engine that's supported by Amazon RDS, consider the benefits offered by using a fully-managed RDS database instead.

Documentation

For more information, see the following documentation:

- Amazon DynamoDB
- Amazon ElastiCache
- Amazon RDS
- Amazon Redshift

Analytics Services for AWS

Today, the amount of data that companies need to analyze is vast. AWS provides analytics tools that can scale to very large data stores efficiently and cost-effectively.

Key Services

Amazon EMR (Amazon EMR) uses Hadoop, an open source framework, to manage and process data. Hadoop uses the MapReduce engine to distribute processing using a *cluster*.

Amazon EMR makes it easier to install, configure, and manage Hadoop. You identify the data source, specify the number and type of EC2 instances for the cluster and what software should be on them, and provide a MapReduce program or run interactive queries. Amazon EMR manages the computing resources and runs your MapReduce program or provides tools like Hive or Pig for queries.

AWS Data Pipeline makes it easy for you to regularly move and process data. You create a *pipeline*, which defines the input data source, the compute resources (EMR clusters or EC2 instances) to perform the processing, any conditions that must be met before performing any processing, and the output data location (such as Amazon S3, Amazon Redshift, Amazon RDS, and Amazon DynamoDB).

Amazon Kinesis enables real-time processing of streaming data at a massive scale. You can send data from Amazon Kinesis to a data warehouse, such as Amazon Simple Storage Service (Amazon S3) or Amazon Redshift, or to an Amazon EMR cluster.

Amazon ML makes is easy for developers to use machine learning technology to obtain predictions for their applications using simple APIs. Amazon ML finds patterns in your existing data, creates machine learning models, and then uses those models to process new data and generate predictions.

Documentation

For more information, see the following documentation:

- Amazon EMR
- Amazon Kinesis
- Amazon Machine Learning
- AWS Data Pipeline

Application Services for AWS

AWS provides a variety of managed services to use with your applications.

Contents

Key Services

The following are the key application services:

Amazon AppStream
Host your streaming application in the AWS cloud and stream the input and output to your users' devices.

Amazon CloudSearch
> Add search to your website.

Amazon Elastic Transcoder
> Convert digital media into the formats required by your users' devices.

Amazon SES
> Send email from the cloud.

Amazon SNS
> Send or receive notifications from the cloud.

Amazon SQS
> Enable components in your application to store data in a queue to be retrieved other components.

Amazon SWF
> Coordinate tasks across the components of your application.

Key Concepts

The following are concepts that you should understand before using the app services.

Message Queues

A *queue* contains messages and enables processes to coordinate their work in an asynchronous manner. Processes write, read, and delete messages from the queue, and perform work based on the information contained in the retrieved messages. You can design your application in a modular way, so that instead of performing tasks in serial and waiting until each task completes, tasks are performed independently and communicate with each other when there is work to be done.

Amazon Simple Queue Service (Amazon SQS) ensures that each message is delivered at least once, supports multiple processes reading and writing from the same queue, and enables you to control who can write to and read from the queue. Each queue must have a unique name and receives a unique URL. Messages are small, so you'll often use Amazon Simple Storage Service (Amazon S3) to store data and include a pointer to the data in your message.

Notifications

A *notification* lets a recipient know when an event has occurred. Unlike a message queue, the information is delivered to the recipient automatically, so the recipient doesn't need to check for messages and retrieve them.

Amazon Simple Notification Service (Amazon SNS) coordinates and manages the delivery of messages to recipients. A *publisher* produces messages and sends them to a *topic*, which defines the message protocol (such as email, SMS, or Amazon SQS) and the recipient list. A *consumer* subscribes to the topic and receives the messages sent to the topic. When you create a topic, you can control which publishers can send messages to the topic and which subscribers can register for notifications.

Workflows

A *workflow engine* coordinates work across distributed components. A *workflow* is a set of activities together with the logic that coordinates the activities. Amazon Simple Workflow Service (Amazon SWF) serves as a central hub for the workflow and also maintains the state of each workflow execution. You create and register *activity workers*, which complete the work, and a *decider*, which provides the coordination logic. Then you start one or more executions of your workflow.

Email

Amazon Simple Email Service (Amazon SES) provides an easy and cost-effective way for you to send a high volume of email. Amazon SES handles outbound email only. It also collects metrics for

you about the messages that were delivered, bounced, rejected, and marked as spam, and provides access to your metrics in real time.

Search

Amazon CloudSearch makes it easy for you to add search capability for your content to your website. First, create a *search domain*, which defines the data to search, controls how Amazon CloudSearch indexes your data, and allocates instances to index the data and process requests. Next, upload the data for search to your domain. Amazon CloudSearch analyzes your data and generates a *search index*, which provides the results for search requests. Now you can submit search requests to the search endpoint for your domain to quickly locate content that meets the specified search criteria.

Streaming Applications

Amazon AppStream deploys your application on AWS infrastructure and streams input and output between your application and a variety of devices, such as personal computers and mobile devices. The processing occurs in the cloud, so the client application on the device can be small in size and require minimal computing power.

Transcoding

Transcoding is the process of converting a video from one format to another. This enables the video to play back on a variety of mobile devices, web browsers, and televisions. Amazon Elastic Transcoder simplifies common transcoding tasks. You create a job, which defines the transcoding operation for a specific video stored in Amazon S3. Elastic Transcoder completes the job, providing status updates during the transcoding operation, and storing the output video in Amazon S3.

Documentation

For more information, see the following documentation:

- Amazon AppStream
- Amazon CloudSearch
- Amazon Elastic Transcoder
- Amazon SES
- Amazon SNS
- Amazon SQS
- Amazon SWF

Management Tools for AWS

AWS provides a variety of tools that help you manage your AWS resources.

Contents

Key Tools

The following are the key management tools:

Amazon CloudWatch
> Monitor resources and applications.

AWS CloudFormation
> Provision your AWS resources using templates.

AWS CloudTrail
> Track the usage history for your AWS resources by logging AWS API calls.

AWS Config
> View the current and previous configuration of your AWS resources, and monitor changes to your AWS resources.

AWS OpsWorks
> Configure and manage the environment for your application, whether in the AWS cloud or your own data center.

AWS Service Catalog
> Distribute servers, databases, websites, and applications to users using AWS resources.

Key Concepts

The following are concepts that you should understand before using the management services.

CloudWatch Metrics

CloudWatch monitors your AWS resources and provides you with data known as *metrics*. You can also configure CloudWatch to send alerts. For example, you can receive an email when your AWS bill reaches a certain amount, or configure Auto Scaling to add or remove EC2 instances from your Auto Scaling group as demand changes.

CloudTrail Logging

CloudTrail captures AWS API calls made for your AWS account by the AWS Management Console, AWS SDKs, command line tools, and AWS deployment and management services, and delivers log files to Amazon S3 so that you can get a history of the calls. The logs contain the identify the users and accounts that made the calls, the source IP addresses the calls were made from, and when the calls occurred.

Elastic Beanstalk Environments

Elastic Beanstalk enables you to easily move your app from its development environment to AWS infrastructure in minutes. You provide basic configuration information, and Elastic Beanstalk launches an *environment*, which runs a single version of your app, provisions the AWS resources required to support your app, such as EC2 instances, and then deploys your app. After your app is deployed, you can update your environment and the changes are applied automatically.

If you know which AWS resources you want to use and how to configure them, you might prefer to use AWS CloudFormation instead.

AWS CloudFormation Templates and Stacks

AWS CloudFormation enables you to provision the AWS resources that you need predictably and repeatedly. First, you create a *template*, a JSON file that defines the required AWS resources, such as EC2 instances and Amazon EBS volumes. You can start from an existing template, or create one from scratch. You can save your template locally or in Amazon S3. Then you create a *stack*, which manages your resources as a single unit, and AWS CloudFormation provisions the resources described in your template and configures them to work together. If stack creation fails, AWS CloudFormation deletes any resources that were created. If stack creation succeeds, you can begin to use your resources.

If needed, you can edit your template and update the stack based on the modified template. AWS CloudFormation updates the resources as specified.

AWS OpsWorks Stacks and Apps

AWS OpsWorks enables you to provision the AWS resources that you need and deploy your apps to those resources. It depends on Chef recipes to handle tasks such as installing software and configuring resources. You create a *stack* to manage your AWS resources as a single unit. An *app* represents the code that you want to run on an application server and contains the information required to deploy the code. The code itself is stored in a repository such as Amazon S3.

Documentation

For more information, see the following documentation:

- Amazon CloudWatch
- AWS CloudFormation
- AWS CloudTrail
- AWS Config
- AWS Elastic Beanstalk
- AWS OpsWorks
- AWS Service Catalog

AWS Tutorials

The following tutorials are designed to get you started creating and using resources on AWS.

Tutorials

For video tutorials, see Getting Started with AWS and AWS Instructional Videos and Labs.

Run a Virtual Server

Launch a virtual server using Amazon EC2 for complete control of your AWS computing resources. For more information, see Amazon EC2.

Task	Tutorial
Launch and connect to a Linux instance	Getting Started with Amazon EC2 Linux Instances
Launch and connect to a Windows instance	Getting Started with Amazon EC2 Windows Instances

Store Files

You can use AWS to store files that you'll access or share on a regular basis, or files that you need to archive and access infrequently.

For more information, see Backup and Recovery.

Task	Tutorial
Use Amazon S3 to store data and retrieve it quickly	Getting Started with Amazon Simple Storage Service
Use Amazon Glacier to store archival data at a low cost	Getting Started with Amazon Glacier

Share Digital Media

You can use AWS to store or share digital media. For more information, see Digital Media.

Deploy a Website

You can use AWS to host your static website. You can also use AWS deployment services to quickly set up a dynamic website.

Task	Tutorial
Deploy a static website using Amazon S3	Getting Started with AWS: Hosting a Static Website
Deploy a web app using AWS Elastic Beanstalk	Getting Started with AWS: Deploying a Web App

Host a Website (Linux)

You can use AWS to host your website on Linux virtual servers. For more information, see Websites & Website Hosting and Application Hosting.

Task	Tutorial
Host a WordPress Blog	Hosting a WordPress Blog with Amazon EC2
Host Drupal using scaling and load balancing	Getting Started with AWS: Hosting a Web App for Linux

Host a Website (Windows)

You can use AWS to host your website on Windows virtual servers. For more information, see Websites & Website Hosting and Application Hosting.

Task	Tutorial
Host a WordPress Blog	Deploying a WordPress Blog on Your Amazon EC2 Windows Instance
Host a .NET app using scaling and load balancing	Getting Started with AWS: Hosting a .NET Web App

Run a Database

You can use AWS to run your database. For more information, see Running Databases on AWS.

Task	Tutorial
Use Amazon DynamoDB, a fully-managed NoSQL database service	Getting Started
Use Amazon RDS to set up, operate, and scale a relational database in the cloud	Getting Started
Use Amazon Redshift, a fast, fully-managed, petabyte-scale data warehouse solution	Amazon Redshift Getting Started

Analyze Your Data

You can use AWS to manage and analyze your business or scientific data. For more information, see Big Data on Amazon Web Services.

Task	Tutorial
Use Hive with Amazon EMR to process sample data in Amazon S3	Getting Started: Analyzing Big Data with Amazon EMR

Related Resources

The following table lists some of the AWS resources that you'll find useful as you work with AWS.

Resource	Description
AWS Products & Services	Information about the products and services that AWS offers.
AWS Documentation	Official documentation for each AWS product, including service introductions, service features, and API reference.
AWS Discussion Forums	Community-based forums for discussing technical questions about Amazon Web Services.
Contact Us	A central contact point for account questions such as billing, events, and abuse. For technical questions, use the forums.
AWS Support Center	The hub for creating and managing your AWS Support cases. Also includes links to other helpful resources, such as forums, technical FAQs, service health status, and AWS Trusted Advisor.
AWS Support	The home page for AWS Support, a one-on-one, fast-response support channel to help you build and run applications in the cloud.
AWS Architecture Center	Provides the necessary guidance and best practices to build highly scalable and reliable applications in the AWS cloud. These resources help you understand the AWS platform, its services and features. They also provide architectural guidance for design and implementation of systems that run on the AWS infrastructure.
AWS Security Center	Provides information about security features and resources.
AWS Economics Center	Provides access to information, tools, and resources to compare the costs of Amazon Web Services with IT infrastructure alternatives.

Resource	Description
AWS Technical Whitepapers	Provides technical whitepapers that cover topics such as architecture, security, and economics. These whitepapers have been written by the Amazon team, customers, and solution providers.
AWS Blogs	Provides blog posts that cover new services and updates to existing services.
AWS Podcast	Provides podcasts that cover new services, existing services, and tips.