

## **FOR FREE MATERIALS JOIN HERE**

JOIN IN TELEGRAM: [CLICK HERE](#)

FOLLOW IN INSTAGRAM: [CLICK HERE](#)

JOIN IN FACEBOOK: [CLICK HERE](#)

FOLLOW IN TWITTER: [CLICK HERE](#)

**Solutions Architect Associate exam**

<https://www.demystify.cloud/>

Sourav Banerjee

# **AWS Certified Solutions Architect Associate**

## Index

### Compute

Amazon EC2 .....	3
AWS Batch .....	5
AWS Elastic Beanstalk .....	7
AWS Lambda .....	10
AWS SAR .....	12
AWS Fargate .....	13

### Containers

Amazon EKS.....	14
Amazon ECS .....	16
Amazon ECR .....	18

### Storage

Amazon S3.....	19
AWS Backup .....	21
Amazon EBS .....	22
Amazon EFS .....	24
Amazon FSx for Windows File Server .....	26
Amazon FSx for Lustre .....	27
Amazon S3 Glacier .....	28
AWS Snowball .....	29
AWS Storage Gateway .....	30

### Database

Amazon Aurora .....	33
Amazon DocumentDB .....	35
Amazon DynamoDB .....	36
Amazon ElastiCache .....	39
Amazon Keyspaces .....	41
Amazon Neptune .....	42
Amazon RDS .....	43
Amazon Redshift .....	46

### Security, Identity, & Compliance

AWS IAM .....	48
Amazon Cognito .....	50
AWS Directory Service .....	52
Amazon RAM .....	54
AWS Secrets Managers.....	55
AWS Security Hub .....	57

### Cryptography & PKI

AWS KMS .....	58
AWS Certificate Manager ...	60

### Management & Governance

AWS Auto Scaling .....	62
AWS CloudFormation .....	63

AWS CloudTrail .....	65
Amazon CloudWatch .....	67
AWS Config .....	69
AWS License Manager .....	71
AWS Organizations .....	72
AWS Systems Manager .....	74

### Developer Tools

AWS CodeBuild .....	76
AWS CodeCommit .....	77
AWS CodeDeploy .....	78
AWS X-Ray .....	79

### Migration & Transfer

AWS DMS .....	80
---------------	----

### Networking & Content Delivery

Amazon API Gateway .....	81
AWS Cloud Map .....	83
Amazon CloudFront .....	85
AWS PrivateLink .....	87
AWS Transit Gateway .....	88
AWS Direct Connect .....	90
AWS ELB .....	92
Amazon Route 53 .....	94
Amazon VPC .....	96

### Front End Web & Mobile

AWS AppSync .....	100
-------------------	-----

### Application Integration

Amazon EventBridge .....	102
Amazon SNS .....	104
Amazon SQS .....	106
AWS Step Functions .....	107
Amazon SWF .....	109

### Billing & Cost Management

AWS Cost Explorer .....	111
AWS Budgets .....	113
AWS Cost & Usage Reports .....	115
Reserved Instance Reporting .....	116

### Monitoring

AWS Personal Health Dashboard .....	117
--	-----

### AWS Management Console

AWS Management Console .....	118
---------------------------------	-----

# AWS EC2

## What is AWS EC2?

- EC2 stands for Elastic Compute Cloud.
- Amazon EC2 is the virtual machine in the Cloud Environment.
- Amazon EC2 provides scalable capacity. Instances can scale up and down automatically based on the traffic.
- You do not have to invest in the hardware.
- You can launch as many servers as you want and you will have complete control over the servers and can manage security, networking, and storage.

## Instance Type:

- Instance type is providing a range of instance types for various use cases.
- The instance is the processor and memory of your EC2 instance.

## EBS Volume:

- EBS Stands for Elastic Block Storage.
- It is the block-level storage that is assigned to your single EC2 Instance.
- It persists independently from running EC2.
  - o Types of EBS Storage
    - General Purpose (SSD)
    - Provisioned IOPS (SSD)
    - Throughput Optimized Hard Disk Drive
    - Cold Hard Disk Drive
    - Magnetic

**Instance Store:** Instance store is the ephemeral block-level storage for the EC2 instance.

- Instance stores can be used for faster processing and temporary storage of the application.

**AMI:** AMI Stands for **Amazon Machine Image**.

- AMI decides the OS, installs dependencies, libraries, data of your EC2 instances.
- Multiple instances with the same configuration can be launched using a single AMI.

**Security Group:** A Security group acts as a virtual firewall for your EC2 Instances.

- It decides the type of port and kind of traffic to allow.
- Security groups are active at the instance level whereas Network ACLs are active at the subnet level.
- Security Groups can only allow but can't deny the rules.
- The Security group is considered stateful.
- By default, in the outbound rule all traffic is allowed and needs to define the inbound rules.

**Key Pair:** A key pair, consisting of a private key and a public key, is a set of security credentials that you can use to prove your identity while connecting to an instance.

- Amazon EC2 instances use two keys, one is the public key which is attached to your EC2 instance.
- Another is the private key which is with you. You can get access to the EC2 instance only if these keys get matched.
- Keep the private key in a secure place.

**Tags:** Tag is a key-value name you assign to your AWS Resources.

- Tags are the identifier of the resource.
- Resources can be organized well using the tags.

**Pricing:**

- You will get different pricing options such as On-Demand, Savings Plan, Reserved Instances, and Spot Instances.

# AWS Batch

## What is AWS Batch?

AWS Batch allows developers, scientists, and engineers to run thousands of computing jobs in the AWS platform. It is a managed service that dynamically maintains the optimal compute resources like CPU, Memory based on the volume of submitted jobs. The User just has to focus on the applications (like shell scripts, Linux codes or java programs).

It executes workloads on **EC2 (including Spot instances) and AWS Fargate**.

## Components:

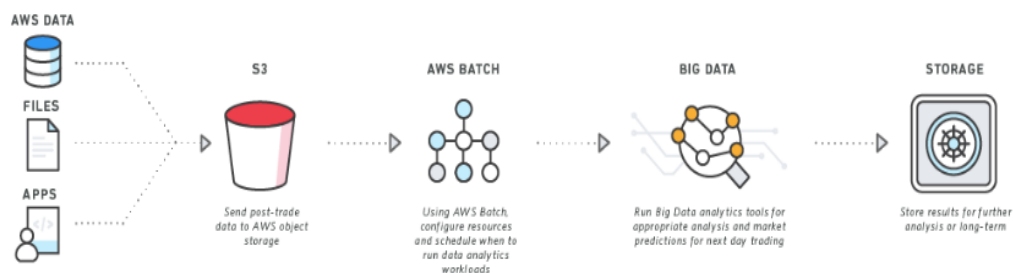
- **Jobs** - are the fundamental applications running on Amazon EC2 machines in containerised form.
- **Job Definitions** – define how the job is meant to be run. Like the associated IAM role, vCPU requirement, and container properties.
- **Job Queues** – Jobs reside in the Job queue where they wait till they are scheduled.
- **Compute Environments** – Each job queue is linked to a computing environment which in itself contains the EC2 instance to run containerized applications.
- There are two types of environments: **Managed** where the user gives min and max vCPU, EC2 instance type and AWS runs it on your behalf and **Unmanaged** where you have your own ECS agent.
- **Scheduler** – maintains the execution of jobs submitted to the queue as time and dependencies.

## Best Practices:

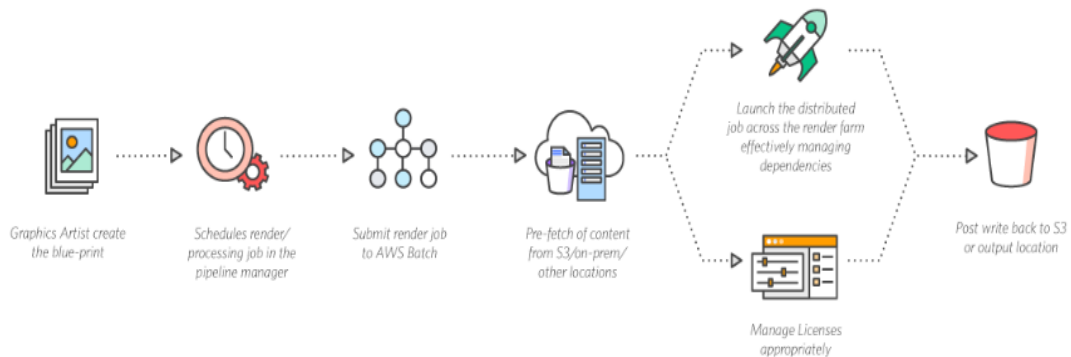
- Use Fargate if you want to run the application without getting into EC2 infrastructure details. Let the AWS batch manage it.
- Use EC2 if your work scale is very large and you want to get into machine specifications like memory, CPU, GPU.
- Jobs running on Fargate are faster on startup as there is no time lag in scale-out operation, unlike EC2 where launching new instances may take time.

## Use Cases:

- **Stock markets and Trading** – The trading business involves daily processing of large scale data and loading them into a Data warehouse for analytics. So that your predictions and decisions are quick enough to make a business grow on a regular basis.



- Media houses and the Entertainment industry – Here a large amount of data in the forms of audio, video and photos are being processed daily to cater to their customers. These application workloads can be moved to containers on AWS Batch.



### Pricing:

- There is no charge for AWS Batch rather you pay for the resources like EC2 and Fargate you use.

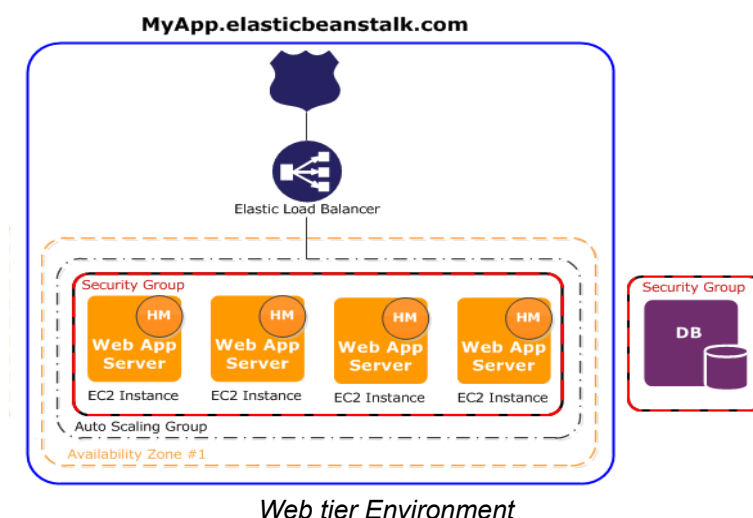
# AWS Beanstalk

## What is Amazon Elastic Beanstalk?

- Beanstalk is a compute service for deploying and scaling applications developed in many popular languages.
- Developers can focus on writing code and don't need to worry about the underlying infrastructure required to run the application.
- AWS Elastic Beanstalk is the best way to deploy your application in the fastest and simplest way.
- It provides the user interface/dashboard to monitor your application.
- It gives you the flexibility to choose AWS resources such as Amazon EC2 Instance along with the pricing options which suit your application needs.

AWS Elastic Beanstalk supports two types of Environment:

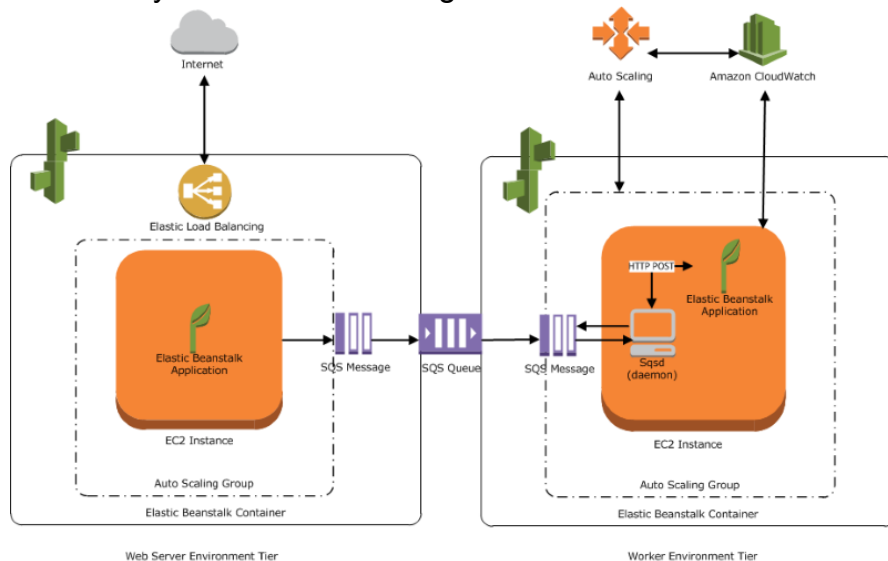
- **Web Tier Environment**
  - This application hosted on the Web Server Environment handles the HTTP and HTTPS requests from the users.
  - Beanstalk Environment: When an environment is launched, Beanstalk automatically assigns various resources to run the application successfully.
  - Elastic Load Balancer: Request is received from the user via Route53 which forwards the request to ELB. Then ELB distributes the request among various EC2 Instances of the Autoscaling group.
  - Auto Scaling Group: Auto Scaling will automatically add or remove EC2 Instance based on the load in the application.
  - Host Manager: Software components inside every EC2 Instance which is responsible for the following:
    - Log files generation
    - Monitoring
    - Events in Instance





- **Worker Environment**

- A worker is a background process that helps applications for handling heavy resource and time-intensive operations.
- It is responsible for database clean up, report generation that helps to remain up and running.
- In the Worker Environment, Beanstalk installs a Daemon on each EC2 Instance in the Auto Scaling Group.
- Daemon pulls requests from the SQS queue and executes the task based on the message received.
- After execution, SQS will delete the message, and in case of failure, it will retry to send the message.



## Platform Supported

- .Net (on Linux or Windows)
- Docker
- GlassFish
- Go
- Java
- Node.js
- Python
- Ruby
- Tomcat

## Deployment Models:

**All at Once:** Deployment will start taking place in all the instances at the same time. It means all your EC2 Instances will be out of service for a short time. Your application will be completely down for the same duration.

**Rolling** – Deploy the new version in batches; unlike all at once, one group of instances will run the old version of the application. That means there will not be complete downtime just like all at once.

**Rolling with additional batch** - Deploy the new version in batches. But before that, provision an additional group of instances to compensate for the updating one.

**Immutable** – Deploy the new version to a separate group of instances, and the update will be immutable.

**Traffic splitting** – Deploy the new version to a separate group of instances and split the incoming traffic between the older and the new ones.

**Pricing:**

- Amazon will not charge you for AWS Elastic Beanstalk.
- Instead, you will be paying for the resources such as EC2 Instance, ELB and Auto Scaling group where your application is hosted.

# AWS Lambda

## What is AWS Lambda?

- AWS Lambda is a **serverless** compute service through which you can run your code without provisioning any Servers.
- It only runs your code when needed and also scales automatically when the request count increases.
- AWS Lambda follows the Pay per use principle – it means there is no charge when your code is not running.
- Lambda allows you to run your code for any application or backend service with zero administration.
- Lambda can run code in response to the events. Example – update in DynamoDB Table or change in S3 bucket.
- You can even run your code in response to HTTP requests using Amazon API Gateway.

## What is Serverless computing?

- Serverless computing is a method of providing backend services on a pay per use basis.
- Serverless/Cloud vendor allows you to write and deploy code without worrying about the underlying infrastructure.
- Servers are still there, but you are not managing them, and the vendor will charge you based on usage.

## When do you use Lambda?

- When using AWS Lambda, you are only responsible for your code.
- AWS Lambda manages the memory, CPU, Network, and other resources.
- It means you cannot log in to the compute instances or customize the operating system.
- If you want to manage your own compute resources, you can use other compute services such as EC2, Elastic Beanstalk.
- There will be a level of abstraction which means you cannot log in to the server or customize the runtime.

## How does Lambda work?



## **Lambda Functions**

- A function is a block of code in Lambda.
- You upload your application/code in the form of single or multiple functions.
- You can upload a zip file, or you can upload a file from the S3 bucket as well.
- After deploying the Lambda function, Lambda automatically monitors functions on your behalf, reporting metrics through Amazon CloudWatch.

## **Lambda Layers**

- A Lambda layer is a container/archive which contains additional code such as libraries, dependencies, or custom runtimes.
- AWS Lambda allows five layers in a function.
- Layers are immutable.
- A new version will be added if you publish a new layer.
- Layers are by default private but can be shared and made public explicitly.

## **Lambda Event**

- Lambda Event is an entity that invokes the lambda function.
- Lambda supports synchronous invocation of Lambda Functions.
- Lambda supports the following sources as an event:
  - AWS DynamoDB
  - AWS SQS
  - AWS SNS
  - CloudWatch Event
  - API Gateway
  - AWS IoT
  - Kinesis
  - CloudWatch Logs

## **Language Supported in AWS Lambda**

- NodeJS
- Go
- Java
- Python
- Ruby

## **Lambda@Edge**

- It is the feature of Amazon CloudFront which allows you to run your code closer to the location of Users of your application.
- It improves performance and reduces latency.
- Just like lambda, you don't have to manage and provision the infrastructure around the world.
- Lambda@Edge runs your code in response to the event created by the CDN.

## **Pricing:**

- Charges will be calculated based on the number of requests for the function executed in a particular duration.
- Duration will be counted on a per 100-millisecond basis.
- Lambda Free tier usage includes 1 million free requests per month.
- It also comes with 400,000 GB-Seconds of compute time per month.

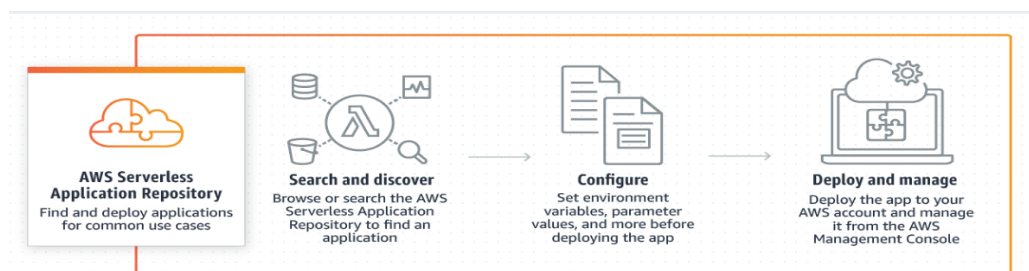
# AWS Serverless Application Repository

## What is AWS Serverless Application Repository?

It is a managed repository for serverless applications. It is used by organizations and independent developers to store and share reusable applications.

### Features:

- AWS Serverless Application Repository has applications for **Alexa Skills, chatbots, IoT, real-time** media processing from many publishers.
- All the applications provided by AWS come under MIT open source license while publicly available applications by other users come under Open Source Initiative (OSI).
- All applications published on Serverless Application Repository are carefully examined by AWS for the correct set of permissions so that the customer knows which application can be accessed.
- AWS **CodePipeline** can be used to link GitHub with Serverless Application Repository.
- Before publishing, describe the application using AWS SAM, package it using CLI, and publish via CLI or SDK or console.
- Applications can be shared within all accounts of AWS Organizations. Users cannot share applications across other Organizations.
- AWS Serverless Application Repository is **integrated with AWS Lambda**. The application can be downloaded and with API Gateway it can trigger the Lambda function. See below diagram -



### Use Case:

- Used for various AWS Alexa skills and integration with IoT devices.
- Used for chatbots that remove inappropriate messages, images from channels.
- Used in Twitter leadership boards.

### Pricing:

- There is no charge for this service itself but you pay for the resources used in the application

# AWS Fargate

## What is AWS Fargate?

AWS Fargate is a serverless compute service that is used for containers by Amazon Elastic Container Service (ECS) and Amazon Elastic Kubernetes Service (EKS).

- It eliminates the tasks required to provision, configure, or scale groups of virtual machines like Amazon EC2 to run containers.
- It packages the application in containers, by just specifying the CPU and memory requirements with IAM policies. Fargate task does not share its underlying kernel, memory resources, CPU resources, or elastic network interface (ENI) with another task.
- It does not support all the task definition parameters that are available in Amazon ECS tasks. Only a few are valid for Fargate tasks with some limitations.
- Kubernetes can be integrated with AWS Fargate by using controllers. These controllers are responsible for scheduling native Kubernetes pods onto Fargate.
- Security groups for pods in EKS can not be used when pods running on Fargate.
- The following storage types are supported for Fargate tasks:
  - Amazon EFS volumes for persistent storage
  - Ephemeral storage for nonpersistent storage

## Benefits:

- Fargate allows users to focus on building and operating the applications rather than focusing on securing, scaling, patching, and managing servers.
- Fargate automatically scales the compute environment that matches the resource requirements for the container.
- Fargate provides built-in integrations with other AWS services like Amazon CloudWatch Container Insights.

## Price details:

- Charges are applied for the amount of vCPU and memory consumed by the containerized applications.
- Fargate's Savings Plans provide savings of up to 50% in exchange for one or three-year long term commitment.
- Additional charges will be applied if containers are used with other AWS services.

## **Amazon Elastic Kubernetes Service(EKS)**

### **What is Amazon Elastic Kubernetes Service(EKS)?**

Amazon Elastic Kubernetes Service (Amazon EKS) is a service that enables users to manage Kubernetes applications in the AWS cloud or on-premises.

Any standard Kubernetes application can be migrated to EKS without altering the code.

The EKS cluster consists of two components:

- Amazon EKS control plane
- Amazon EKS nodes

The Amazon EKS control plane consists of nodes that run the Kubernetes software, such as etcd and the Kubernetes API server. Amazon EKS runs its own Kubernetes control plane without sharing control plane infrastructure across other clusters or AWS accounts.

To ensure high availability, Amazon EKS runs Kubernetes control plane instances across multiple Availability Zones. It automatically replaces unhealthy control plane instances and provides automated upgrades and patches for the new control planes.

The two methods for creating a new Kubernetes cluster with nodes in Amazon EKS:

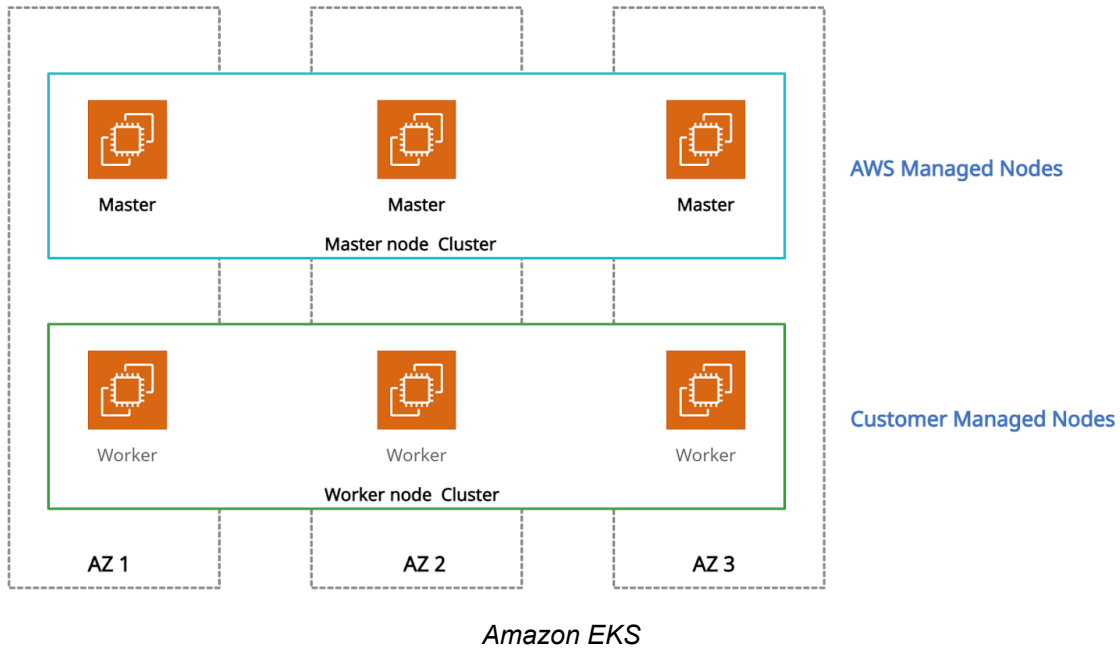
- eksctl – A command-line utility that consists of kubectl for creating and managing Kubernetes clusters on Amazon EKS.
- AWS Management Console and AWS CLI

There are methods that Amazon EKS cluster uses to schedule pods using single or combined node groups:

- Self-managed nodes - consist of one or more Amazon EC2 instances that are deployed in an Amazon EC2 Auto Scaling group
- Amazon EKS Managed node groups - helps to automate the provisioning and lifecycle management of nodes.
- AWS Fargate - run Kubernetes pods on AWS Fargate

Amazon Elastic Kubernetes Service is integrated with many AWS services for unique capabilities:

- Images - Amazon ECR for container images
- Load distribution - AWS ELB (Elastic Load Balancing)
- Authentication - AWS IAM
- Isolation - Amazon VPC



### Use Cases:

- Using Amazon EKS, Kubernetes clusters and applications can be managed across hybrid environments.
- EKS with kubeflow can model machine learning workflows using the latest EC2 GPU-powered instances.
- Users can execute batch workloads on the EKS cluster using the Kubernetes Jobs API across AWS compute services such as Amazon EC2, Fargate, and Spot Instances.

### Price details:

- \$0.10 per hour is charged for each Amazon EKS cluster created.
- Using EKS with EC2 - Charged for AWS resources (e.g. EC2 instances or EBS volumes).
- Using EKS with AWS Fargate - Charged for CPU and memory resources starting from the time to download the container image until the Amazon EKS pod terminates.



# Amazon Elastic Container Service

## What is Amazon ECS?

Amazon Elastic Container Service (Amazon ECS) is a regional container orchestration service like Docker that allows to execute, stop, and manage containers on a cluster.

A container is a standard unit of software development that combines code, its dependencies, and system libraries so that the application runs smoothly from one environment to another.

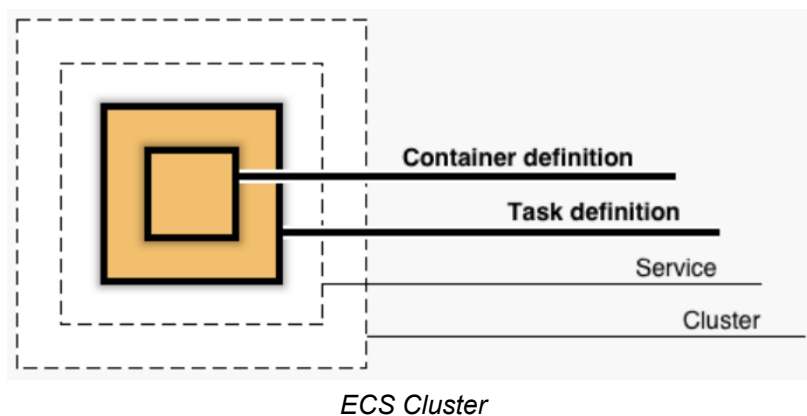
Images are created from a Dockerfile (text format), which specifies all of the components that are included in the container. These images are then stored in a registry from where they can then be downloaded and executed on the cluster.

All the containers are defined in a task definition that runs a single task or tasks within a service. The task definitions (JSON format) defines which container images should run across the clusters. A service is a configuration that helps to run and maintain several tasks simultaneously in a cluster.

ECS cluster is a combination of tasks or services that can be executed on EC2 Instances or AWS Fargate, a serverless compute for containers. When using Amazon ECS for the first time, a default cluster is created.

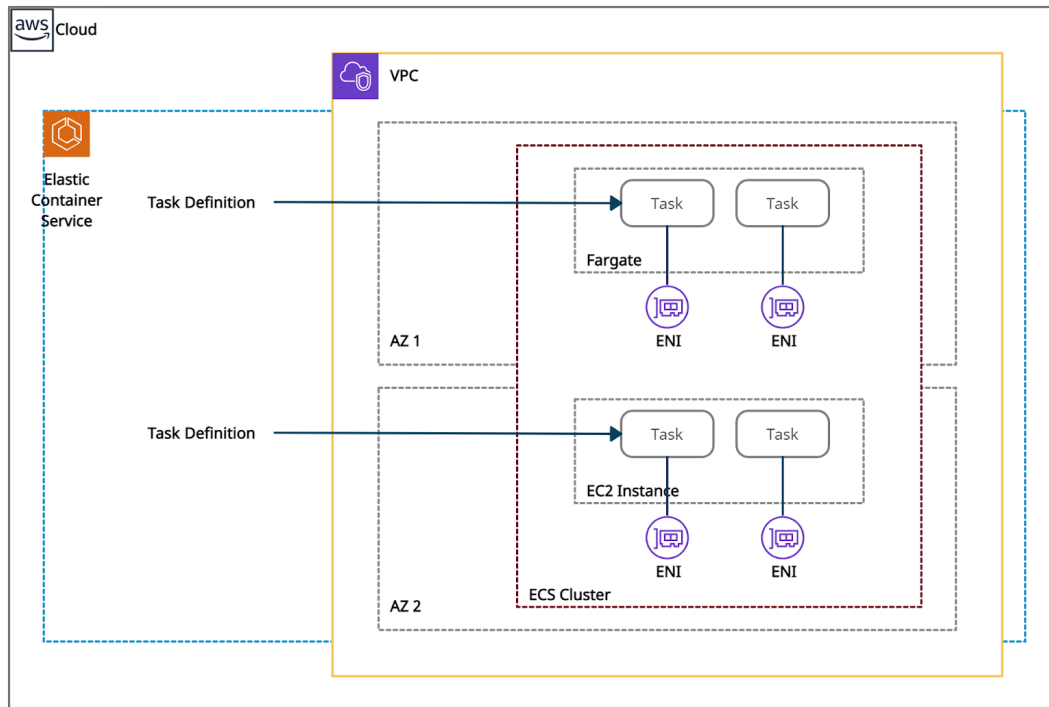
The container agent runs on each instance within an Amazon ECS cluster. It sends data on the resource's current running tasks and resource utilization to Amazon ECS. It starts and stops the tasks whenever it receives a request from Amazon ECS.

A task is the representation of a task definition. The number of tasks to run on your cluster is specified after the task definition is created within Amazon ECS. The task scheduler is responsible for attaching tasks within your cluster based on the task definitions.



Application Load Balancers offer some attractive features:

- It enables containers to use dynamic host port mapping. For that, multiple tasks from the same service are allowed per container instance.
- It supports path-based routing and priority rules due to which multiple services can use the same listener port on a single Application Load Balancer.



*Amazon Elastic Container Service*

Amazon ECS can be integrated with:

- AWS Identity and Access Management
- Amazon EC2 Auto Scaling
- Elastic Load Balancing
- Amazon Elastic Container Registry
- AWS CloudFormation

It decreases time consumption by eliminating user tasks to install, operate, and scale cluster management infrastructure. With API calls, Docker-enabled applications can be launched and stopped.

It powers other services such as Amazon SageMaker, AWS Batch, Amazon Lex. It also integrates with AWS App Mesh, to provide rich observability, controls traffic and security features to the applications.

### Use Cases:

The two main use cases in Amazon ECS are:

- Microservices - They are built by the architectural method that decomposes or decouples complex applications into smaller and independent services.
- Batch Jobs - Docker containers are best suited for batch job workloads. Batch jobs are short-lived packages processed under Docker image. So they can be deployed anywhere, such as in an Amazon ECS task.

### Pricing details:

- Amazon ECS provides two charge models:
  - Fargate Launch Type Model - pay for the amount of vCPU and memory resources.
  - EC2 Launch Type Model - pay for the AWS resources created to store and run the application.

# Amazon Elastic Container Registry

## What is Amazon Elastic Container Registry?

Amazon Elastic Container Registry (ECR) is a managed service that allows users to store, manage, share, and deploy container images and artifacts. It is mainly integrated with Amazon Elastic Container Service (ECS), for simplifying the production workflow.

### Features:

- It stores both the containers which are created, and any container software bought through AWS Marketplace.
- It is integrated with Amazon Elastic Container Service (ECS), Amazon Elastic Kubernetes Service (EKS), and AWS Lambda, and AWS Fargate for easy deployments.
- AWS Identity and Access Management (IAM) enables resource-level control of each repository within ECR.
- It supports public and private container image repositories. It allows sharing container applications privately within the organization or publicly for anyone to download.
- A separate portal called Amazon ECR Public Gallery, helps to access all public repositories hosted on Amazon ECR Public.
- It stores the container images in Amazon S3 because S3 provides 99.999999999% (11 9's) of data durability.
- It allows cross-region and cross-account replication of the data for high availability applications.
- Encryption can be done via HTTPS while transferring container images. Images are also encrypted at rest using Amazon S3 server-side encryption or by using customer keys managed by AWS KMS.
- It is integrated with continuous integration and continuous delivery and also with third-party developer tools.
- Lifecycle policies are used to manage the lifecycle of the images.

### Pricing details:

- Using AWS Free Tier, new customers get 500 MB-month of storage for one year for private repositories and 50 GB-month of free storage for public repositories.
- Without Sign-up, 500 GB of data can be transferred to the internet for free from a public repository each month.
- By signing-up to an AWS account, or authenticating to ECR with an existing AWS Account, 5 TB of data can be transferred to the internet for free from a public repository each month.

# Amazon S3

## What is Amazon S3?

S3 stands for Simple Storage Service.

Amazon S3 is object storage that allows us to store any kind of data in the bucket. It provides availability in multiple AZs, durability, security, and performance at a very low cost.

Any type of customer can use it to store and protect any amount of data for use cases, like static and dynamic websites, data analytics, and backup.

## Basics of S3?

- It is object-based storage.
- Files are stored in Buckets.
- The bucket is a kind of folder.
- Folders can be from 0 to 5 TB.
- S3 bucket names must be unique globally.
- When you upload a file in S3, you will receive an HTTP 200 code if the upload was successful.
- S3 offers Strong consistency for PUTs of new objects, overwrites or delete of current object and List operations.
- By Default, all the Objects in the bucket are private.

## Properties of Amazon S3.

- **Versioning:** This allows you to keep multiple versions of Objects in the same bucket.
- **Static Website Hosting:** S3 can be used to host a Static Website, which does not require any server-side Technology.
- **Encryption:** Encrypt Object at rest with Amazon S3 Managed keys (SSE-S3), or Amazon KMS Managed Keys (SS3-KMS).
- **Objects Lock:** Block Version deletion of the object for a defined period. Object lock can be enabled during the bucket creation only.
- **Transfer Acceleration:** Transfer Acceleration takes advantage of Amazon CloudFront's globally distributed edge locations and enables the fast, easy, and secure transfer of files.

## Permissions & Management.

- **Access Control List: ACLs** used to grant read/write permission to another AWS Account.
- **Bucket Policy:** It uses JSON based access policy advance permission to your S3 Resources.
- **CORS:** CORS stands for Cross-Origin Resource Sharing. It allows cross-origin access to your S3 Resources.

## Charges:

You will be charged based on multiple factors:

- Storage
- Requests
- Storage Management Pricing (Life Cycle Policies)
- Transfer Acceleration
- Data Transfer Pricing

### Miscellaneous Topic

- **Access Point:** By creating Access Point, you can make S3 accessible over the internet.
- **Life Cycle:** By Configuring Lifecycle, you can make a transition of objects to different storage classes.
- **Replication:** This feature will allow you to replicate data between buckets within the same or different region.

### Storage Class/Pricing model of S3

- S3 Standard
- S3 Standard-IA (Infrequent Access)
- S3 Intelligent Tiering (No need to mentioned Life Cycle Policy)
- S3 One Zone-IA (Kept in a Single Zone)
- S3 Glacier (For Archiving Purpose)
- S3 Glacier Deep Archive (For Archiving Purpose)

Storage class	Suitable for	Durability	Availability	Availability Zones	Min. storage days
S3 Standard	accessed data frequently	100%	99.99%	>= 3	None
S3 Standard-IA	accessed data infrequently	100%	99.90%	>= 3	30 days
S3 Intelligent-Tiering	Storage for unknown access patterns	100%	99.90%	>= 3	30 days
S3 One Zone-IA	Non-critical data	100%	99.50%	1	30 days
S3 Glacier	For long term Data Archival. e.g., 3 years – 5 years	100%	99.99%	>= 3	90 days
S3 Glacier Deep Archive	For long term Data Archival. e.g., 3 years – 5 years	100%	99.99%	>= 3	180 days
RRS (Reduced Redundancy Storage)	Frequently accessed for non-critical data but not recommended	99%	99.99%	>= 3	NA

# AWS Backup

## What is AWS Backup?

AWS Backup is a secure service that automates and governs data backup (protection) in the AWS cloud and on-premises.

### Features:

- It offers a backup console, backup APIs, and the AWS Command Line Interface (AWS CLI) to manage backups across the AWS resources like instances and databases.
- It offers backup functionalities based on policies, tags, and resources.
- It provides scheduled backup plans (policies) to automate backup of AWS resources across AWS accounts and regions.
- It offers incremental backup to minimize storage costs. The first backup backs up a full copy of the data and then only the successive incremental backup changes.
- It provides backup retention plans to retain and expire backups automatically. Automated backup retention also helps to minimize storage costs for backup.
- It provides a dashboard in the AWS Backup console to monitor backup and restore activities.
- It offers an enhanced solution by providing separate encryption keys for encrypting multiple AWS resources.
- It provides lifecycle policies configured to transition backups from Amazon EFS to cold storage automatically.
- It is tightly integrated with Amazon EC2 to schedule backup jobs and the storage (EBS) layer. It also simplifies recovery by restoring whole EC2 instances from a single point.
- It supports cross-account backup and restores either manually or automatically within the AWS organizations.
- It allows backups and restores to different regions, especially during any disaster, to reduce downtime and maintain business continuity.
- It integrates with Amazon CloudWatch, AWS CloudTrail, and Amazon SNS to monitor, audit API activities and notifications.

### Use cases:

- It can use AWS Storage Gateway volumes for hybrid storage backup. AWS Storage Gateway volumes are secure and compatible with Amazon EBS, which helps restore volumes to on-premises or the AWS environment.

### Price details:

- AWS charges monthly based on the amount of backup storage used and the amount of backup data restored.

## AWS EBS - Elastic Block Store

### What is AWS EBS?

Amazon Elastic Block Store (AWS EBS) is a persistent block-level storage (volume) service designed to be used with Amazon EC2 instances. **EBS is AZ specific** & automatically replicated within its AZ to protect from component failure, offering high availability and durability.

### Types of EBS:

<b>SSD-backed volumes</b> (Solid State Drive)	<b>Optimized for transactional workloads</b> (small and frequent I/O) - IOPS	
<b>Types SSD</b>	<b>General Purpose SSD- gp2</b> (1 GiB — 16 TiB)  IOPS : 3000 to 20000 Max / Volume	Boot volumes Development /Test Low-latency Apps Virtual Desktops
	<b>Provisioned IOPS SSD (io1)</b> low-latency or high-throughput Consistent IOPS (16,000+ IOPS ) Transactional workloads	MongoDB / NoSQL MySQL / RDS Latency Critical Apps
<b>HDD-backed volumes:</b> (Magnetic Drive)	<b>Low-Cost throughput-intensive workloads</b> (Not Suitable for Low Latency(IOPS) -- i.e. booting)	
<b>Types HDD</b>	<b>Throughput Optimized HDD (st1)</b> Low Cost - Frequently accessed, throughput-intensive & Large-Sequential O/I -- 500 MB/s	Stream Processing Big Data Processing Data Warehouse
	<b>Cold HDD (sc1)</b> Lowest Cost - less frequently accessed data Throughput : 250 MiB/s	Colder Data requires fewer scans per day.

### Features:

- High Performance (Provides single-digit-millisecond latency for high-performance)
- Highly Scalable (Scale to petabytes)
- Offers high availability (guaranteed 99.999% by Amazon) & Durability
- Offers seamless encryption of data at rest through Amazon Key Management Service (KMS).
- Automate Backups through **data lifecycle policies** using EBS Snapshots to S3 Storage.
- EBS detached from an EC2 instance and attached to another one quickly.

### Key Points to Remember:

- **Backup/Migration:** To move a volume across AZs, you first need to take a snapshot.
- **Provisioned capacity:** capacity needs to be provisioned in advanced (GBs & IOPS)
- You can increase the capacity of the drive over time.
- It can be detached from an EC2 instance and attached to another one quickly.
- It's locked to **Single Availability Zone (AZ)**
- The default volume type is General Purpose SSD (gp2)
- EBS Volume can be mounted parallelly using RAID Settings:
  - RAID 0 (increase performance)
  - RAID 1 (increase fault tolerance)
- It's a network drive (i.e. not a physical drive).
- Unencrypted volume can be encrypted using an encrypted snapshot
- Snapshot of the encrypted volume is encrypted by default.
- When you share an encrypted snapshot, you must also share the customer-managed CMK used to encrypt the snapshot.

### Pricing:

- You will get billed for all the provisioned capacity & snapshots on S3 Storage + Sharing Cost between AZs/Regions

### EBS vs Instance Store

#### Instance Store (ephemeral storage) :

- It is ideal for temporary block-level storage like buffers, caches, temporary content
- Data on an instance store volume persists only during the life of the associated instance. (As it is volatile storage - lose data if stop the instance/instance crash)
- **Physically attached to ec2 instance** - hence, the **lowest possible latency**.
- **Massive IOPS - High performance**
- Instance store backed Instances can be of maximum 10GiB volume size
- Instance store volume cannot be attached to an instance, once the Instance is up and running.
- Instance store volume can be used as root volume.
- You cannot create a snapshot of an instance store volume.

#### EBS :

- Persistent Storage.
- Reliable & Durable Storage.
- EBS volume can be detached from one instance and attached to another instance.
- EBS boots faster than instance stores.



# AWS EFS - Elastic File Storage

## What is AWS EFS?

Amazon Elastic File System (Amazon EFS) provides a scalable, fully managed elastic distributed file system based on NFS. It is persistent file storage & can be easily scaled up to petabytes.

It is designed to share parallelly with thousands of EC2 instances to provide better throughput and IOPS. It is a regional service automatically replicated across multiple AZ's to provide High Availability and durability.

## Types of EFS Storage Classes:

Standard Storage	For frequently accessed files.
Infrequent Access Storage ( EFS-IA )	For files not accessed every day Cost-Optimized (costs only \$0.025/GB-month) Use EFS Lifecycle Management to move the file to EFS IA

## EFS Access Modes :

### 1) Performance Modes:

- General Purpose: low latency at the cost of lower throughput.
- Max I/O: high throughput at the cost of higher latency.

### 2) Throughput Modes :

- Bursting (default): throughput grows as the file system grows
- Provisioned: specify throughput in advance. (fixed capacity)

## Features:

- Fully Managed and Scalable, Durable, Distributed File System (NFSv4)
- Highly Available & Consistent low latencies. (EFS is based on SSD volumes)
- POSIX Compliant (NFS) Distributed File System.
- EC2 instances can access EFS across AZs, regions, VPCs & on-premises through AWS Direct Connect or AWS VPN.
- Provides EFS Lifecycle Management for the better price-performance ratio
- It can be integrated with AWS Datasync for moving data between on-premise to AWS EFS
- Supported Automatic/Schedule Backups of EFS (AWS Backups)
- It can be integrated with CloudWatch & CloudTrail for monitoring and tracking.
- EFS supports encryption at transit(TLS) and rest both. (AWS Key Management Service (KMS))
- Different Access Modes: Performance and Throughput for the better cost-performance tradeoff.
- EFS is more expensive than EBS.
- Once your file system is created, you cannot change the performance mode
- Not suitable for boot volume & highly transactional data (SQL/NoSQLdatabases)
- Read-after-write consistency for data access.
- Integrated with IAM for access rights & security.

**Use Cases:** (Sharing Files Across instances/containers)

- Mission critical business applications
- Microservice based Applications
- Container storage
- Web serving and content management
- Media and entertainment file storage
- Database Backups
- Analytics and Machine Learning

**Best Practices:**

- Monitor using cloudWatch and track API using CloudTrails
- Leverage IAM services for access rights and security
- Test before fully migrating mission critical workload for performance and throughput.
- Separate out your latency-sensitive workloads. Storing these workloads on separate volumes ensures dedicated I/O and burst capabilities.

**Pricing:**

- Pay for what you have used based on Access Mode/Storage Type + Backup Storage.

# Amazon FSx for Windows File Server

## What is Amazon FSx for Windows File Server?

- Amazon FSx for Windows File Server is an FSx solution that offers a scalable and shared file storage system on the Microsoft Windows server.
- Using the Server Message Block (SMB) protocol with Amazon FSx Can access file storage systems from multiple windows servers.
- It offers to choose from HDD and SSD storage, offers high throughput, and IOPS with sub-millisecond latencies for Windows workloads.
- Using SMB protocol, Amazon FSx can connect file systems to Amazon EC2, Amazon ECS, Amazon WorkSpaces, Amazon AppStream 2.0 instances, and on-premises servers using AWS Direct Connect or AWS VPN.
- It provides high availability (Multi-AZ deployments) with an active and standby file server in separate AZs.
- It automatically and synchronously replicates data in the standby Availability Zone (AZ) to manage failover.
- Using AWS DataSync with Amazon FSx helps to migrate self-managed file systems to Windows storage systems.
- It offers identity-based authentication using Microsoft Active Directory (AD).
- It automatically encrypts data at rest with the help of AWS Key Management Service (AWS KMS). It uses SMB Kerberos session keys to encrypt data in transit.

## Use cases:

- Large organizations which require shared access to multiple data sets between multiple users can use Amazon FSx for Windows File Server.
- Using Windows file storage, users can easily migrate self-managed applications to AWS using AWS DataSync.
- It helps execute business-critical Microsoft SQL Server database workloads easily and automatically handles SQL Server Failover and data replication.
- Using Amazon FSx for Windows File Server, users can easily process media workloads with low latencies and high throughput.
- It enables users to execute high intensive analytics workloads, including business intelligence and data analytics applications.

## Price details:

- Charges are applied monthly based on the storage and throughput capacity used for the file system's file system and backups.
- The cost of storage and throughput depends on the deployment type, either single-AZ or multi-AZ.

# Amazon FSx for Lustre

## What is Amazon FSx for Lustre?

- Amazon FSx for Lustre is an FSx solution that offers scalable storage for the Lustre system (parallel and high-performance file storage system).
- It supports fast processing workloads like custom electronic design automation (EDA) and high-performance computing (HPC).
- It provides shared file storage with hundreds of gigabytes of throughput, sub-millisecond latencies, and millions of IOPS.
- It offers a choice between SSD and HDD for storage.
- It integrates with Amazon S3 to process data concurrently using parallel data-transfer techniques.
- It stores datasets in S3 as files instead of objects and automatically updates with the latest data to run the workload.
- It offers to select unreplicated file systems for shorter-term data processing.
- It can be used with existing Linux-based applications without any changes.
- It offers network access control using POSIX permissions or Amazon VPC Security Groups.
- It easily provides data-at-rest and in-transit encryption.
- AWS Backup can also be used to backup Lustre file systems.
- It integrates with SageMaker to process machine learning workloads.

## Use cases:

- The workloads which require shared file storage and multiple compute instances use Amazon FSx for Lustre for high throughput and low latency.
- It is also applicable in media and big data workloads to process a large amount of data.

## Price details:

- Charges are applied monthly in GB based on the storage capacity used for the file system.
- Backups are stored incrementally, which helps in storage cost savings.

# Amazon S3 Glacier

## What is Amazon S3 Glacier?

Amazon S3 Glacier is a web service with vaults that offer long-term data archiving and data backup.

It is the cheapest S3 storage class and offers 99.999999999% of data durability.

It helps to retain unlimited data like photos, videos, documents as TAR or ZIP file, data lakes, analytics, IoT, machine learning, and compliance data.

S3-Standard, S3 Standard-IA, and S3 Glacier storage classes, objects, or data are automatically stored across availability zones in a specific region.

S3 Glacier provides the following data retrieval options:

- Expedited retrievals -
  - It retrieves data in 1-5 minutes.
- Standard retrievals -
  - It retrieves data between 3-5 hours.
- Bulk retrievals -
  - It retrieves data between 5-12 hours.

## Features:

- It integrates with AWS IAM to allow vaults to grant permissions to the users.
- It integrates with AWS CloudTrail to log and monitor API call activities for auditing.
- A vault is a place for storing archives with certain functionalities like to create, delete, lock, list, retrieve, tag, and configure.
- Vaults can be set with access policies for additional security by the users.
- Amazon S3 Glacier jobs are the select queries that execute to retrieve archived data.
- It uses Amazon SNS to notify when the jobs complete.
- It uses 'S3 Glacier Select' to query specific archive objects or bytes for analytics instead of complete archives.
- S3 Glacier Select operates on uncompressed comma-separated values (CSV format) and output results to Amazon S3.
- Amazon S3 Glacier Select uses SQL queries using SELECT, FROM, and WHERE.
- It offers only SSE-KMS and SSE-S3 encryption.
- Amazon S3 Glacier does not provide real-time data retrieval of the archives.

## Use Cases:

- It helps to store and archive media data that can increase up to the petabyte level.
- Organizations that generate, analyze, and archive large data can make use of Amazon S3 Glacier and S3 Glacier Deep Archive storage classes.
- Amazon S3 Glacier replaces tape libraries for storage because it does not require high upfront cost and maintenance.

## Price details:

- Free Usage Tier - Users can retrieve with standard retrieval up to 10 GB of archive data per month for free.
- Data transfer out from S3 Glacier in the same region is free.

# AWS Snowball

## What is AWS Snowball?

- AWS Snowball is a storage device used to transfer a large amount of data ranging from 50TB - 80TB between Amazon Simple Storage Service and onsite data storage location at high speed.
- These devices are protected by the AWS Key Management Service to protect data in transit securely.
- AWS Snow Family Management Console helps to manage data transfer jobs using job management API.
- The Snowball client and the Amazon S3 Adapter for Snowball are used to perform data transfers on the Snowball device locally.
- If data transfers involve large files and multiple jobs, you might separate the data into several smaller data segments. Troubleshooting a large transfer can be more complicated than a small transfer. Parallelization helps to transfer data with Snowball at a faster rate.
- AWS Snowball is integrated with other AWS services such as AWS CloudTrail to capture all API calls as events and with Amazon Simple Notification Service (Amazon SNS) to notify about data transfer.
- AWS Snowball Edge is a type of Snowball device that can transport data at speeds faster than the internet and can do local processing and edge-computing workloads between the local environment and the AWS Cloud.
- Using Snowball Edge devices, one can execute EC2 AMIs and deploy AWS Lambda code on the devices to perform processing and analysis with the applications.

There are two other categories of the AWS Snow family:

- Snowball Edge Compute Optimized - provide block storage, object storage, and 40 vCPUs
- Snowball Edge Storage Optimized - provides block storage, object storage, and 52 vCPUs, and an optional GPU for high processing use cases.

## Use cases:

- AWS Snowball helps to transfer or receive large amounts of data with clients or partners regularly.
- AWS Snowball collects large data and performs analysis to overcome failure and improve safety, efficiency and productivity.

## Pricing details:

Charges are applied based on the following components:

- Service Fee per Job - region-specific.
- Per Day Fee to keep snowfall onsite - region-specific
- Data Transfer fee for Amazon S3:
  - Data transfer IN - free.
  - Data transfer OUT - region-specific.
- Shipping Costs - standard carrier rates.

# AWS Storage Gateway

## What is the AWS Storage Gateway?

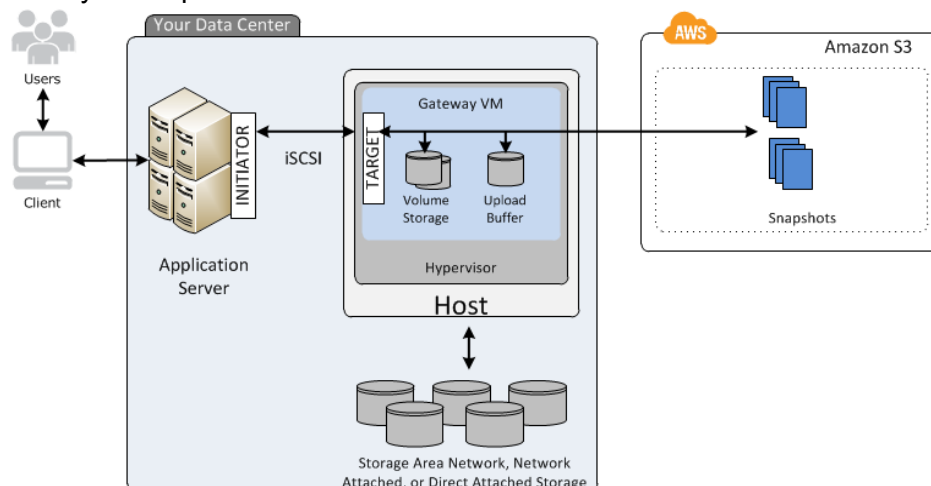
AWS Storage Gateway is a **hybrid cloud storage service** that allows your on-premise storage & IT infrastructure to seamlessly integrate with AWS Cloud Storage Services. It Can be AWS Provided Hardware or Compatible Virtual Machine.

## Purpose of Using AWS Storage Gateway(hybrid Cloud Storage) :

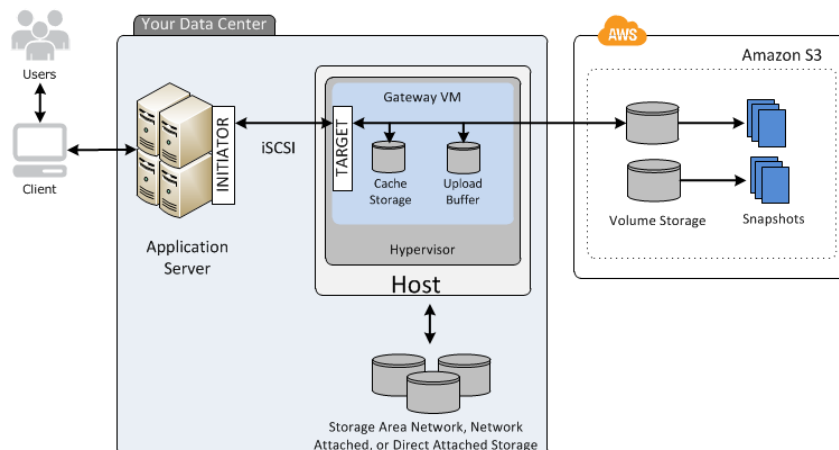
- To Fulfill Licencing Requirements.
- To Achieve Data-Compliance Requirements.
- To Reduce Storage & Management Cost.
- For Easy and Effective Application Storage-Lifecycle & Backup Automation.
- For Hybrid Cloud & Easy Cloud Migration.

## Volume Gateway (iSCSI)

- To Access Virtual Block-Level Storage Stored on-premise
- It can be asynchronously backed up and stored as a snapshot on AWS S3 for high reliability & durability.
  - **Storage Volume Gateway:** All Applications Data Stored on-premise and the only backup is stored on AWS S3.

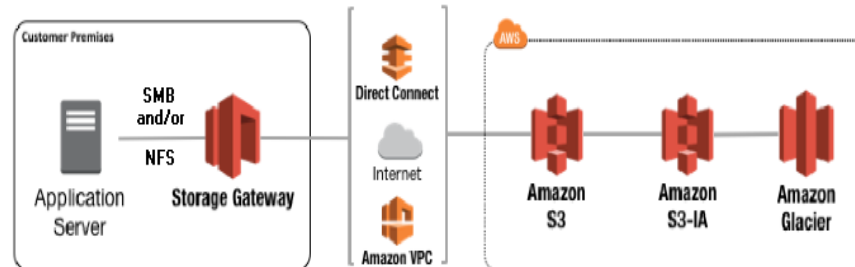


- **Cache Volume Gateway:** Only Hot Data / Cached data is Stored on-premise and all other application data is stored on AWS S3.



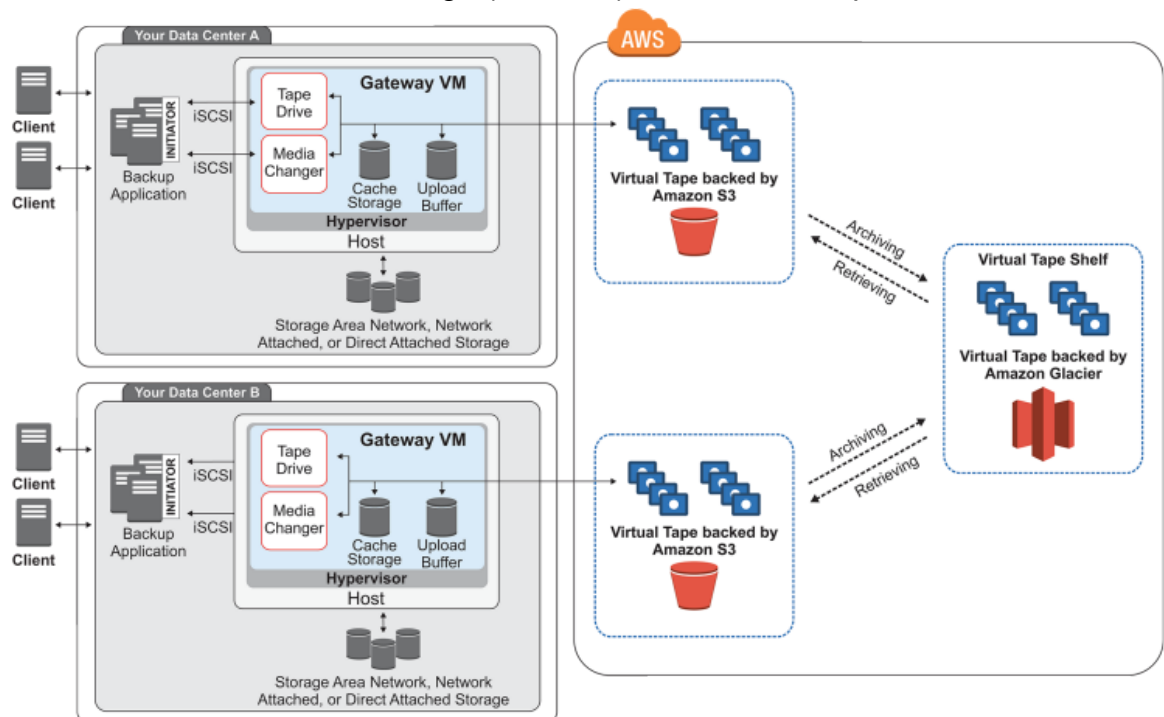
## File Gateway (NFSv4 / SMB)

- To Access Object-based Storage ( AWS S3 Service )
- Supports NFS Mount Point for accessing S3 Storage to the local system as Virtual Local File System
- Leverage the benefits of AWS S3 Storage Service



## Tape Gateway (VTL)

- It is virtual local tape storage.
- It uses the Virtual Tape Library(VTL) by iSCSI protocol.
- It is cost-effective archive storage (AWS S3) for cloud backup.



## Features of AWS Storage Gateway

- Cost-Effective Storage Management
- To achieve Low Latency on-premise.
- Greater Control over Data still take advantage of the cloud (Hybrid Cloud)
- Compatible and Compliance
- To meets license requirement
- Supports both hardware and software gateway
- Easy on-premise to Cloud Migrations
- Standard Protocol for storage access like NFS/SMB/iSCSI



**Use Cases:**

- Cost-Effective Backups and Disaster Recovery Management
- Migration to/from Cloud
- Managed Cache: Integration of Local(on-premises) Storage to Cloud Storage (Hybrid Cloud)
- To Achieve Low Latency by storing data on-premise and still leverage cloud benefits

**Pricing :**

- Charges are applied on what you use with the AWS Storage Gateway and based on the type and amount of storage you use.

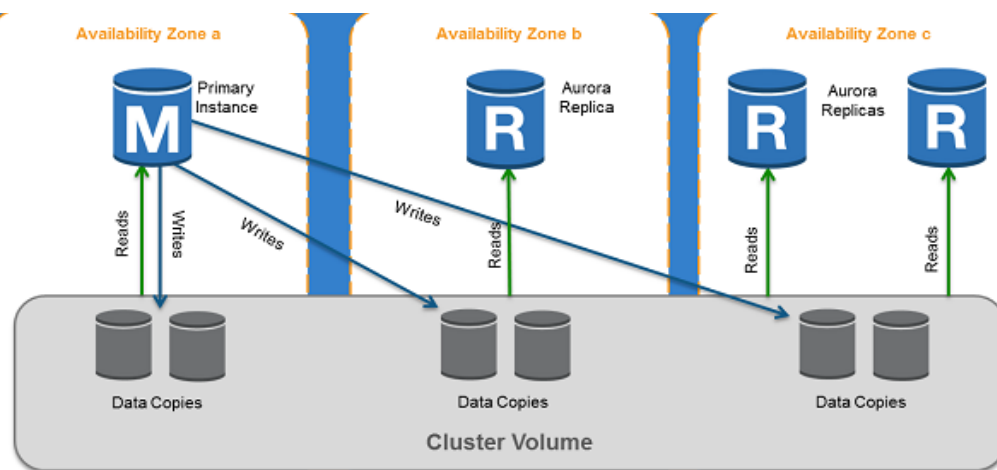
# Amazon Aurora

## What is Amazon Aurora?

Aurora is the fully managed RDS services offered by AWS. **It's only compatible with PostgreSQL/MySQL.** As per AWS, Aurora provides 5 times throughput to traditional MySQL and 3 times throughput to PostgreSQL.

## Features:

- It is only supported by regions which have minimum 3 availability zones.
- High availability of 99.99%. Data in Aurora is kept as 2 copies in each AZ with a minimum 3 AZ's making a total of 6 copies.
- It can have up to 15 Read replicas (RDS has only 5).
- It can scale up to 128 TB per database instance.
- Aurora DB cluster comprises two instances:
  - Primary DB instance – It supports both read/write operations and one primary DB instance is always present in the DB cluster.
  - Aurora Replica – It supports only read operation. Aurora automatically fails over to its replica in less time in case a primary DB instance is not available.



- Read replicas fetch the same result as the primary instance with a lag of not more than 100 ms.
- Data is highly secure as it resides in VPC. Encryption at rest is done through AWS KMS and encryption in transit is done by SSL.
- **Aurora Global Database** - helps to span in multiple AWS regions for low latency access across the globe. This can also be utilised as backup in case the whole region has gone over an outage or disaster.
- **Aurora Multi master** – is a new feature only compatible only with **MySQL edition**. It gives the ability to scale out write operations over multiple AZ. So there is no single point of failure in the cluster and applications can perform both read/write at any node.
- **Aurora Serverless** - gives you the flexibility to scale in and out on the basis of database load. The user has to only specify the minimum (2 GB of RAM), maximum (488 GB of RAM) capacity units. This feature of Aurora is highly beneficial if the user has intermittent or unpredictable workload. It is available for both MySQL and PostgreSQL.

- **Fault tolerance and Self-Healing feature-** In Aurora, each set of data is replicated in six copies over 3 AZ. So that it can handle the loss up to 2 copies without impacting the write feature and up to 3 copies without impacting the read feature. Aurora storage is also self-healing which means disks are continuously scanned for errors and repaired.

#### **Best practices:**

- If the user is not sure about the workload of the database then prefer Aurora Serverless. If you have a team of developers and testers who hit the database only during particular hours of day and it remains minimal during night, again prefer Aurora Serverless.
- If write operations and DDL are crucial requirements, choose Multi-master Aurora for MySQL. In this manner, all writer nodes are equally functional and failure one doesn't impact the other.
- Aurora Global database is best for industries in finance, gaming as one single DB instance provides a global footprint. The application enjoys low latency read operation on such databases.

#### **Pricing:**

- There are no up-front fees.
- On-demand instances are costlier than reserved instances. There is no additional fee for backup if the retention period is less than a day.
- Data transfer between Aurora DB instance and EC2 in the same AZ is free.
- All data transfer IN to Database is free of charge.
- Data transfer OUT of Database through the internet is chargeable if it exceeds 1 GB/month.

# Amazon DocumentDB

## What is Amazon DocumentDB?

DocumentDB is a fully managed document database service by AWS which supports MongoDB workloads. It is highly recommended for storing, querying, and indexing JSON Data.

## Features:

- It is compatible with MongoDB versions 3.6 and 4.0.
- All on-premise MongoDB or EC2 hosted MongoDB databases can be migrated to DocumentDB by using DMS (Database Migration Service).
- All database patching is automated in a stipulated time interval.
- DocumentDB storage scales automatically in increments of 10GB and maximum up to 64TB.
- Provides up to **15 Read replicas** with single-digit millisecond latency.
- All database instances are highly secure as they reside in VPCs which only allow a given set of users to access through security group permissions.
- It supports **role-based access control (RBAC)**.
- Minimum **6 read copies of data is created in 3 availability zones making it fault-tolerant**.
- **Self-healing** – Data blocks and disks are continuously scanned and repaired automatically.
- All cluster snapshots are user-initiated and stored in S3 till explicitly deleted.

## Best Practices:

- It reserves 1/3<sup>rd</sup> RAM for its services, so choose your instance type with enough RAM so that performance and throughput are not impacted.
- Setup Cloudwatch alerts to notify users when the database is reaching its maximum capacity.

## Use Case:

- Highly beneficial for workloads that have flexible schemas.
- It removes the overhead of keeping two databases for operation and reporting. Store the operational data and send them parallel to BI systems for reporting without having two environments.

## Pricing:

- Pricing is based on the instance hours, I/O requests, and backup storage.

# Amazon DynamoDB

## What is DynamoDB?

- AWS DynamoDB is a Key-value and DocumentDB database by Amazon.
- It delivers a single Digit millisecond Latency.
- It can handle 20 million requests per second and 10 trillion requests a day.
- It is a Serverless Service; it means no servers to manage.
- It maintains the performance by managing the data traffic of tables over multiple servers.

## Features:

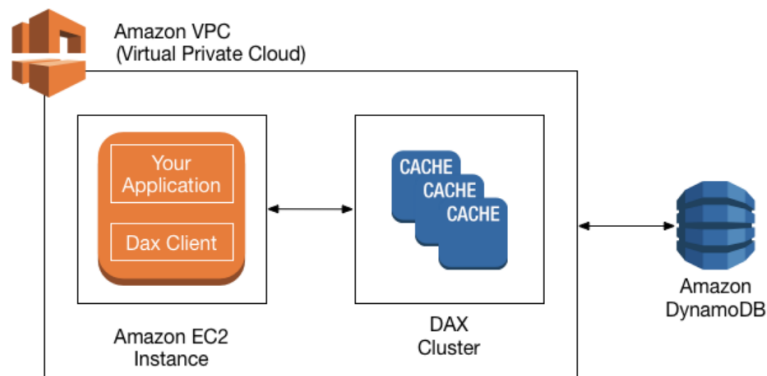
- It can create the Table for your application and can handle the rest.
- No-SQL database provides fast and predictable performance.
- It is designed for automatic scaling and can be replicated across regions.
- It can also create Dynamic Tables, which means it can store any number of multi-valued attributes.
- **Primary Key** – Uniquely identifies each item in the table, such as Student\_ID in Student Table, Employee\_ID in employees Table.
- **Partition Key** – Primary key with one attribute
- **Partition Key and Sort Key** – Primary Key with two attributes.  
It is used when we do not have any attribute in the table, which will identify the item in the table.
- **Indexes**
  - A database index is an entity in the database that will improve data retrieval speed in any table.
- **Secondary Index**
  - A secondary index is a way to efficiently access records in a database utilizing some information other than the usual primary key.
  - We can create one or more secondary Indexes in the table.
  - Secondary Indexes is of two types:
    - **Global Secondary Indexes:** An Index that can have different partitions and sort keys from the table.
    - **Local Secondary Indexes:** An index with the same partition key as the table but a different sort of key.

## DynamoDB Accelerator

- Amazon DynamoDB Accelerator (DAX) is a fully managed in-memory cache engine designed for Amazon DynamoDB.
- It can deliver up to 10 times performance improvement and can handle around 20 million requests per second.
- The performance can improve from milliseconds to microseconds.
- DAX is for the workloads that are read-intensive, not write-intensive. DAX is not for strongly consistent reads.
- We don't have to think about the hardware provisioning, pathing, and replication over multiple instances.
- It also supports encryption but does not support transport layer security.
- Types of Scaling in DAX:
  - **Horizontal Scaling:** It means adding read replicas to the cluster. We can add or remove up to 10 read replicas.
  - **Vertical Scaling:** It means changing the node type.

## DynamoDB Scan

- The Scan operation returns more than one item in the table.
- Scan operations are slower than query operations and process sequentially.
- One Scan operation can fetch up to 1 MB of data.
- The application can request for parallel scan operation by providing **Segment** and **TotalSegments** parameters for large tables.
- While accessing the data in the table, Scan operation uses eventually consistent reads.



## DynamoDB Queries

- The DynamoDB queries in DynamoDB are based on the value of the primary keys.
- In query operation, partition key attribute and single value to that attribute are mandatory.
- The query returns the result based on the partition key value. In addition to that, you can also provide a sort key to get a more refined result.
- Query operation returns a result set. It will give an empty result if no matching result is found.
- One Query operation can fetch up to 1 MB of data.

## DynamoDB Streams

- It captures the sequence of item-level modification in the table.
- The Streams information is stored up to 24 hrs.
- Any application can access these stream records.
- The information in the DynamoDB Streams is in near real-time.
- DynamoDB and DynamoDB streams have two different endpoints.
- To work with tables and indexes, you must access DynamoDB Endpoint.
- To work with Stream records, your application must access DynamoDB Streams Endpoints.
- There are multiple ways to access the Streams.
- The most common way is to use AWS Lambda.
- A second common approach is to use a standalone application that uses the Kinesis Client Library (KCL) with Streams Kinesis Adapter.

## DynamoDB Transactions

- DynamoDB Transactions have ACID (atomicity, consistency, isolation, and durability) Property within a single account across one or more tables.

- You can use transactions where the application needs to execute multiple operations (insert, update, delete) as a part of single logic.
- Transactions have the properties of DynamoDB, such as scalability and performance.
- Each transaction can store 4 MB of data and can store up to 25 unique items.
- The use case where we can implement DynamoDB Transactions:
  - o Financial transaction processing
  - o Gaming applications
  - o Processing a high volume of orders
  - o Processing financial transactions

### Consistency Model:

**Eventual Consistent Reads:** If you read the data from the recent write operation, you will get stale data.

**Strongly Consistent Writes:** If you read the data from the recent write operation, you will get updated data. But it might not be available instantly.

### Throughput Model:

**Read Capacity Unit (RCU):** For an item, it represents a single strongly consistent read or double eventual consistent reads in a second, and the size of an item can be up to 4KB. If you need to read an item that is larger than 4KB, you need to add an additional read capacity unit.

**Write Capacity Unit (WCU):** It represents one write per second for an item. And the size of an item can be up to 1KB. If you need to write a larger than 1KB item, you need to add an additional write capacity unit.

**Provisioned:** We Need to mention the throughput in advance. This model is for Predictable workloads. We have to define a range for Reading and Write Capacity Units.

**On-Demand:** We need not mention the throughput in Advance. This Model is for non-predictable workloads. We do not need to define a range for Reading and Write Capacity Units.

### Charges:

- DynamoDB charges as per the disk space you consume.
- Charges for data transfer out.
- Charges for provisioned throughput.
- Charges for Reserved Capacity Unit.

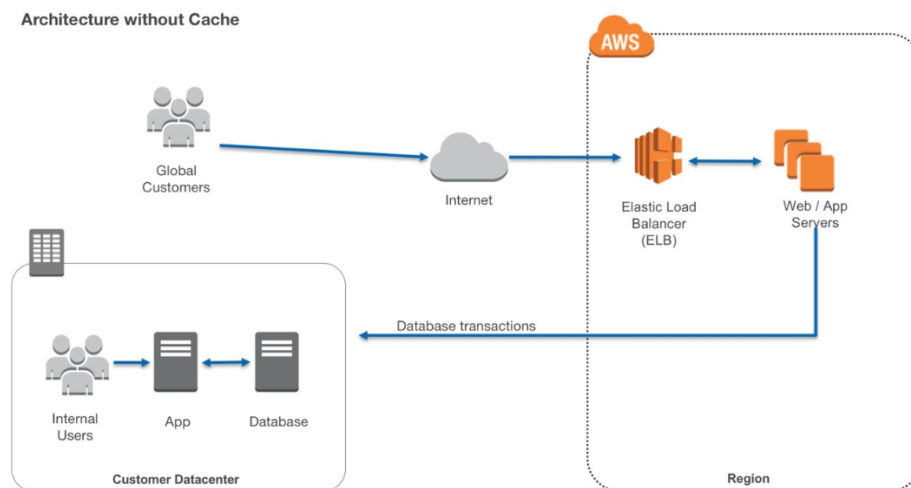
# Amazon ElastiCache

## What is Amazon ElastiCache?

ElastiCache is a fully managed in-memory data store. It significantly improves latency and performance for all read-heavy application workloads. In-memory caches are faster than disk-based databases. It works with both **Redis** and **Memcached** protocol based engines.

### Features:

- It is high availability as even the data center is under maintenance or outage; the data is still retrieved from Cache.
- Unlike databases, data is retrieved in a key-value pair fashion.
- Data is stored in nodes which is a unit of network-attached RAM. Each node has its own Redis or Memcached protocol running. Automatic replacement of failed nodes is configured.



- **Memcached** features –
  - Data is volatile.
  - Supports only simple data-type.
  - Supports multi-threading.
  - Scaling can be done by adding or removing nodes.
  - Nodes can span in different Availability Zones.
  - Multi-AZ failover is not supported.
- **Redis** features –
  - Data is non-volatile.
  - Supports complex Data types like strings, hashes, and geospatial-indexes.
  - Doesn't support multi-threading.
  - Scaling can be done by adding shards and not nodes. A shard is a collection of primary nodes and read-replicas.
  - Multi-AZ is possible by placing a read replica in another AZ.
  - In case of failover, can be switched to read replica in another AZ

### Best practices:

- Storing web sessions. In web applications running behind a load balancer, use Redis so if one the server is lost, data can still be retrieved.



- Caching Database results. Use Memcached in front of any RDS where repetitive queries are being fired to improve latency and performance.
- Live Polling and gaming dashboards. Store frequently accessed data in Memcached to fetch results quickly.
- Combination of RDS and ElastiCache can be utilized to improve architecture on the backend.

**Pricing:**

- Available only for on-demand and reserved nodes.
- Charged for per node hour.
- Partial node hours will be charged as full node hours.
- No charge for data exchange between ElastiCache and EC2 within the same AZ. <https://aws.amazon.com/elasticache/pricing/>

# Amazon Keyspaces

## What is Amazon Keyspaces (for Apache Cassandra)?

Keyspaces is an Apache Cassandra compatible database in AWS. It is fully managed by AWS, highly available, and scalable. Management of servers, patching is done by Amazon. It scales based on incoming traffic with virtually unlimited storage and throughput.

### Features:

- Keyspaces is compatible with Cassandra Query Language (CQL). So your application can be easily migrated from on-premise to cloud.
- Two operation modes are available as below
  1. **The On-Demand capacity mode** is used when the user is not certain about the incoming load. So throughput and scaling are managed by Keyspaces itself. It's costly and you pay only for the resources you use.
  2. **The Provisioned capacity mode** is used when you have predictable application traffic. A user just needs to provide many max read/write per second in advance while configuring the database. It's less costly.
- There is no upper limit for throughput and storage.
- Keyspaces is integrated with Cloudwatch to measure the performance of the database with incoming traffic.
- Data is replicated across 3 Availability Zones for high durability.
- Point-in-Time-recovery (PITR) is there to recover data lost due to accidental deletes. The data can be recovered up to any second till 35 days.

### Use Cases:

- Build Applications using open source Cassandra APIs and drivers. Users can use Java, Python, .NET, Ruby, Perl.
- Highly recommended for applications that demand a low latency platform like trading.
- Use cloud trail to check the DDL operations. It gives brief information on who accessed, when, what services were used and a response returned from AWS. Some hackers creeping into the database firewall can be detected here.

### Pricing:

- Users only pay for the read and write throughput, storage, and networking resources.

# Amazon Neptune

## What is Amazon Neptune?

Amazon Neptune is a graph database service used as a web service to build and run applications that require connected datasets.

The graph database engine helps to store billions of connections and provides milliseconds latency for querying them.

It offers a choice from graph models and languages for querying data.

- Property Graph (PG) model with Apache TinkerPop Gremlin graph traversal language,
- W3C standard Resource Description Framework (RDF) model with SPARQL Query Language.

It is highly available across three AZs and automatically fails over any of the 15 low latency read replicas.

It provides fault-tolerant storage by replicating two copies of data across three availability zones.

It provides continuous backup to Amazon S3 and point-in-time recovery from storage failures.

It automatically scales storage capacity and provides encryption at rest and in transit.

# Amazon RDS

## What is Amazon RDS?

RDS (Relational Database System) in AWS makes it easy to operate, manage, and scale in the cloud.

It provides scalable capacity with a cost-efficient pricing option and automates manual administrative tasks such as patching, backup setup, and hardware provisioning.

## Engines supported by RDS are given below:

### MySQL

- It is the most popular open-source DB in the world.
- Amazon RDS makes it easy to provision the DB in AWS Environment without worrying about the physical infrastructure.
- In this way, you can focus on application development rather than Infra. Management.

### MS SQL

- MS-SQL is a database developed by Microsoft.
- Amazon allows you to provision the DB Instance with provisioned IOPS or Standard Storage.

### MariaDB

- MariaDB is also an open-source DB developed by MySQL developers.
- Amazon RDS makes it easy to provision the DB in AWS Environment without worrying about the physical infrastructure.

### PostgreSQL

- Nowadays, PostgreSQL has become the preferred open-source relational DB. Many enterprises now have started using PostgreSQL powered database engines.

### Oracle

- Amazon RDS also provides a fully managed commercial database engine like Oracle.
- Amazon RDS makes it easy to provision the DB in AWS Environment without worrying about the physical infrastructure.
- You can run Oracle DB Engine with two different licensing models – “License Included” and “Bring-Your-Own-License (BYOL).”

### Amazon Aurora

- It is the relational database engine developed by AWS only.
- It is a MySQL and PostgreSQL-compatible DB engine.
- Amazon claims that it is five times faster than the standard MySQL DB engine and around three times faster than the PostgreSQL engine.
- The cost of the aurora is also less than the other DB Engines.
- In Amazon Aurora, you can create up to 15 read replicas instead of 5 in other databases.

### DB Instance Class

DB Instance Class Type	Example	Use Case
Standard	db.m6g, db.m5, db.m4, db.m3, db.m1	These deliver balanced compute, memory, and networking for a broad range of general-purpose workloads.
Burstable Performance	db.t3, db.t2	Burstable performance instances are designed to provide a baseline level of CPU performance with the ability to burst to a higher level when required by your workload.
Memory Optimized	db.z1d, db.x1e, db.x1, db.6g, db.r5, db.r4, db.r3	designed to deliver fast performance for workloads that process large data sets in memory

### Multi AZ Deployment

- Enabling multi-AZ deployment creates a Replica (Copy) of the database in different availability zones in the same Region.
- Multi-AZ synchronously replicates the data to the standby instance in different AZ.
- Each AZ runs on physically different and independent infrastructure and is designed for high reliability.
- Multi-AZ deployment is for Disaster recovery not for performance enhancement.

### Read Replicas

- Read Replicas allow you to create one or more read-only copies of your database in the same or different regions.
- **Read Replica is mostly for performance enhancement. We can now use Read-Replica with Multi-AZ as a Part of DR (disaster recovery) as well.**
- A Read Replica in another region can be used as a standby database in event of regional failure/outage. It can also be promoted to the Production database.

Multi-AZ Deployment	Read Replica
Synchronous Replication	Asynchronous Replication
Highly Durable	Highly scalable
Spans two availability Zone within a region	Can be within an Availability Zone, cross-AZ or cross-region as well
Automatic failover to the standby database	Can be manually promoted to stand-alone Database
Used for Disaster Recovery	Used to enhance the performance

## Storage Type

- **General Purpose (SSD):** General Purpose storage is suitable for database workloads that provide a baseline of 3 IOPS/GiB and the ability to burst to 3,000 IOPS.
- **Provisioned IOPS (SSD):** Provisioned IOPS storage is suitable for I/O-intensive database workloads. I/O range is from 1,000 to 30,000 IOPS.

## Monitoring

- By default, enhanced monitoring is disabled.
- Enabling enhanced monitoring incurs extra charges.
- Enhanced monitoring is not available in the AWS GovCloud(US) Region.
- Enhanced monitoring is not available for the instance class db.m1.small.
- Enhanced monitoring metrics include IOPS, Latency, Throughput, Queue Depth.
- Enhanced monitoring gathers information from an agent installed in DB Instance.

## Backups & Restore

- The default backup retention period for automatic backup is 7 days if you use the console, for CLI and RDS API it's 1 day.
- Automatic backup can be retained for up to 35 days.
- The minimum Automatic backup retention period is 0 days, which will disable the automatic backup for the instance.
- 100 Manual snapshots are allowed in a single region.

## Charges:

You will be charged based on multiple factors:

- Active RDS Instances
- Storage
- Requests
- Backup Storage
- Enhanced monitoring
- Transfer Acceleration
- Data Transfer for cross-region replication

# Amazon Redshift

## What is Amazon Redshift?

Amazon redshift is a fast and powerful, fully managed, petabyte-scale data warehouse service in the cloud. This service is highly scalable to a petabyte or more for \$1000 per terabyte per year, less than a tenth of most other data warehousing solutions.

Redshift can be configured as follows:

- **Single node** (160 GB)
- **Multi-Node**
  - Leader Node (manages client connections and receives queries)
  - Compute Node (store data and perform queries and computations). Up to 128 compute nodes.

## Features:

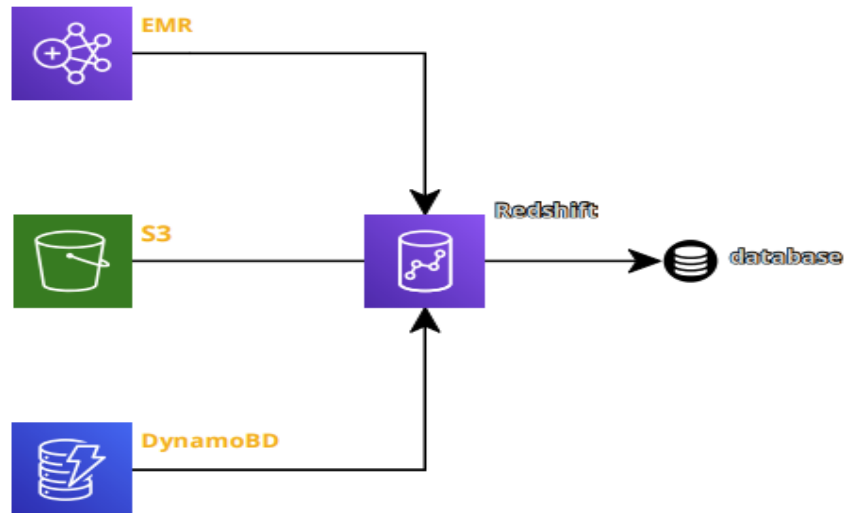
- It employs multiple compression techniques and can often achieve significant compression relative to traditional relational databases.
- It doesn't require indexes or materialized views, so uses less space than traditional database systems.
- Massively parallel processing (**MPP**): Amazon redshift automatically distributes data and query load across all nodes. Amazon redshift makes it easy to add nodes to your data warehouse and maintain fast query performance as data grows in future.
- Enabled by default with a 1-day retention period.
- Maximum retention period is 35 days.
- Redshift always maintain at least three copies of your data (the original and replica on the compute nodes and a backup in Amazon S3)
- Redshift can also asynchronously replicate your snapshots to S3 in another region for disaster recovery.
- It is only available in 1 AZ but can store snapshots to new AZs in the event of an outage.

## Security Considerations

- Data encrypted in transit using SSL.
- Encrypted at rest using AES-256 encryption.
- By default, RedShift takes care of key management.
  - Manager your own keys through HSM
  - AWS key Management service.

## Use cases

- If we want to copy data from EMR, S3, and DynamoDB to power a custom Business intelligence tool. Using a third-party library, we can connect and query redshift for results.



### Pricing:

- Compute Node Hours - total number of hours you run across your all compute nodes for the billing period.
- You are billed for 1 unit per node per hour, so a 3-node data warehouse cluster running persistently for an entire month would incur 2160 instance hours.
- You will not be charged for leader node hours, only compute nodes will incur charges.



# AWS IAM

## What is Identity Access and Management?

- IAM stands for Identity and Access Management.
- AWS IAM may be a service that helps you control access to AWS resources securely.
- You use IAM to regulate who is allowed and have permissions to AWS Resources.
- You can manage and use resources in your AWS account without having to share your password or access key.
- It enables you to manage access to AWS services and resources securely.
- We can attach Policies to AWS users, groups, and roles.

**Principal:** An Entity that will make a call for action or operation on an AWS Resource. Users, Groups, Roles all are AWS Principal. AWS Account Root user is the first principal.

## IAM User & Root User

- **Root User** - When you first create an AWS account, you begin with an email (Username) and Password with complete access to all AWS services and resources in the account. This is the AWS account, root user.
- **IAM User** - A user that you create in AWS.
  - o It represents the person or service who interacts with AWS.
  - o IAM users' primary purpose is to give people the ability to sign in to AWS individually without sharing the password with others.
  - o Access permissions will be depending on the policies which are assigned to the IAM User.

## IAM Group

- A group is a collection of IAM users.
- You can assign specific permission to a group and add the users to that group.
- For example, you could have a group called DB Admins and give that type of permission that Database administrators typically need.

## IAM Role

- IAM Role is like a user with policies attached to it that decides what an identity can or cannot do.
- It will not have any credentials/Password attached to it.
- A Role can be assigned to a federated user who signed in from an external Identity Provider.
- IAM users can temporarily assume a role and get different permission for the task.

## IAM Policies

- It decides what level of access an Identity or AWS Resource will possess.
- A Policy is an object associated with identity and defines their level of access to a certain resource.
- These policies are evaluated when an IAM principal (user or role) makes a request.

- Policies are JSON based documents.
- Permissions inside policies decide if the request is allowed or denied.
  - **Resource-Based Policies:** These JSON based policy documents attached to a resource such as Amazon S3 Bucket.
  - These policies grant permission to perform an action on that resource and define under what condition it will apply.
  - These policies are the inline policies, not managed resource-based policies.
  - IAM supports only one type of resource-based policy called trust policy, and this policy is attached to a role.
  - **Identity-Based Policies:** These policies have complete control over the identity that it can perform on which resource and under which condition.
  - **Managed policies:** Managed policies can attach to the multiple users, groups, and roles in the AWS Account.
    - **AWS managed policies:** These policies are created and managed by AWS.
    - **Customer managed policies:** These policies are created and managed by you. It provides more precise control than AWS Managed policies.
  - **Inline policies:** Inline policies are the policies that can directly be attached to any individual user, group, or role. It maintains a one-to-one relationship between the policy and the identity.

#### **IAM Security Best Practices:**

- Grant least possible access rights.
- Enable multi-factor authentication (MFA).
- Monitor activity in your AWS account using CloudTrail.
- Use policy conditions for extra security.
- Create a strong password policy for your users.
- Remove unnecessary credentials.

#### **Pricing:**

- Amazon provides IAM Service at no additional charge.
- You will be charged for the services used by your account holders.

# Amazon Cognito

## What is Amazon Cognito?

Amazon Cognito is a service used for authentication, authorization, and user management for web or mobile applications.

Amazon Cognito enables users to sign in through social identity providers such as Google, Facebook, and Amazon, and through enterprise identity providers such as Microsoft Active Directory via SAML.

Amazon Cognito authorizes a unique identifier for each user and acts as an OpenID token provider trusted by AWS Security Token Service (STS) to access temporary, limited-permission AWS credentials.

The two main components of Amazon Cognito are

**User pools** are user repositories (where user profile details are kept) that provide sign-up and sign-in options for your app users. User pools provides

- sign-up and sign-in services through a built-in customizable web UI.
- user directory and user profiles.
- security features such as multi-factor authentication (MFA), checks for compromised credentials, account takeover protection, and phone and email verification.
- helps in customized workflows and user migration through AWS Lambda triggers.

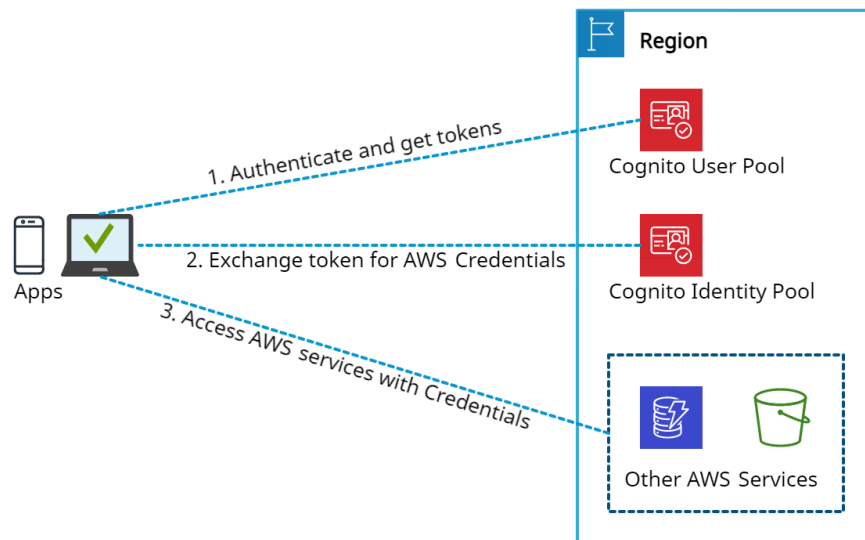
**Identity pools** provide temporary AWS credentials to the users so that they could access other AWS resources without re-entering their credentials. Identity pools support the following identity providers

- Amazon Cognito user pools.
- Third-party sign-in facility.
- OpenID Connect (OIDC) providers.
- SAML identity providers.
- Developer authenticated identities.

Amazon Cognito is capable enough to allow usage of user pools and identity pools separately or together.

## Amazon Cognito Federated Identities

- It is a service that provides limited temporary security credentials to mobile devices and other untrusted environments.
- It helps to create a unique identity for the user over the lifetime of an application.



### *Amazon Cognito*

#### **Features:**

- Advanced security features of Amazon Cognito provide risk-based authentication and protection from the use of compromised credentials.
- To add user sign-up and sign-in pages to your apps, Android, iOS, and JavaScript SDKs for Amazon Cognito can be used.
- Cognito User Pools provide a user directory that scales to millions of users.
- Amazon Cognito uses famous identity management standards like OAuth 2.0, OpenID Connect, and SAML 2.0.
- Users' identities can be verified using SMS or a Time-based One-time Password (TOTP) generator, like Google Authenticator.

#### **Pricing Details:** (pay only for what you use)

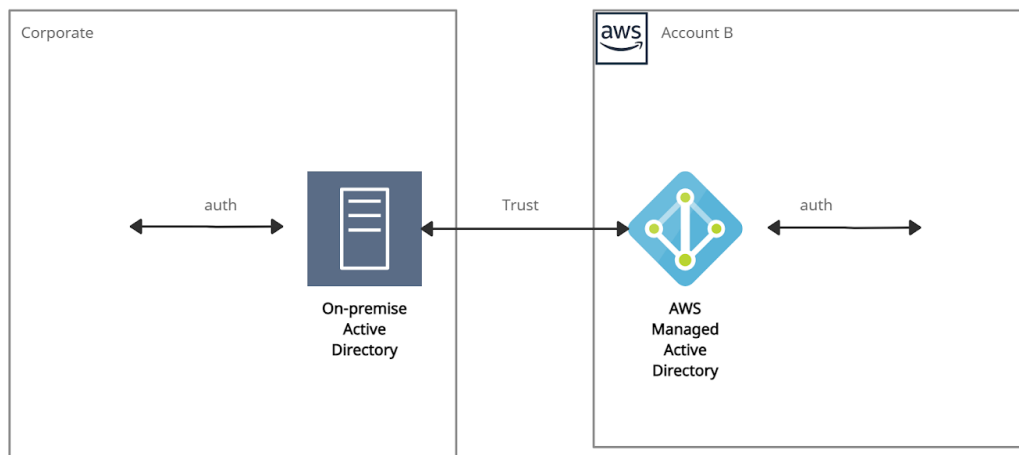
- Amazon Cognito is mainly charged for identity management and data synchronization.
- There are volume-based pricing tiers above the free tier for users who sign in directly with their credentials from a User Pool or with social identity providers such as Apple, Google, Facebook, and Amazon.

# AWS Directory Service

## What is AWS Directory Service?

AWS Directory Service, also known as AWS Managed Microsoft Active Directory (AD), enables multiple ways to use Microsoft Active Directory (AD) with other AWS services.

- Trust relationships can be set up from on-premises Active Directories into the AWS cloud to extend authentication.
- It runs on a Windows Server, can perform schema extensions, and works with SharePoint, Microsoft SQL Server, and .Net apps.
- The directory remains available for use during the patching (updating) process for AWS Managed Microsoft AD.
- Using AWS Managed Microsoft AD, it becomes easy to migrate AD-dependent applications and Windows workloads to AWS.
- A trust relationship can be created between AWS Managed Microsoft AD and existing on-premises Microsoft Active using single sign-on (SSO).



*AWS Managed AD*

AWS Directory Service provides the following directory types to choose from

- Simple AD
- Amazon Cognito
- AD Connector

### Simple AD:

- It is an inexpensive Active Directory-compatible service driven by SAMBA 4.
- It is an isolated or self-supporting AD directory type.
- It can be used when there is a need for less than 5000 users.
- It cannot be joined with on-premise AD.
- It is not compatible with RDS SQL Server.
- It provides some features like
  - Applying Kerberos-based SSO,
  - Assigning Group policies,
  - Managing user accounts and group memberships,
  - Helping in joining a Linux domain or Windows-based EC2 instances.
- It does not support the following functionalities.
  - Multi-factor authentication (MFA),

- Trust relationships,
- DNS dynamic update,
- Schema extensions,
- Communication over LDAPS,
- PowerShell AD cmdlets.

### **Amazon Cognito:**

- It is a user directory type that provides sign-up and sign-in for the application using Amazon Cognito User Pools.
- It can create customized fields and store that data in the user directory.
- It helps to federate users from a SAML IdP with Amazon Cognito user pools and provide standard authentication tokens after they authenticate with a SAML IdP (identities from external identity providers).

### **AD Connector:**

- It is like a gateway used for redirecting directory requests to the on-premise Active Directory.
- For this, there must be an existing AD, and VPC must be connected to the on-premise network via VPN or Direct Connect.
- It is compatible with Amazon WorkSpaces, Amazon WorkDocs, Amazon QuickSight, Amazon Chime, Amazon Connect, Amazon WorkMail, and Amazon EC2.
- It is also not compatible with RDS SQL Server.
- It supports multi-factor authentication (MFA) via existing RADIUS-based MFA infrastructure.

### **Use cases:**

- It provides a Sign In option to AWS Cloud Services with AD Credentials.
- It provides Directory Services to AD-Aware Workloads.
- It enables a single-sign-on (SSO) feature to Office 365 and other Cloud applications.
- It helps to extend On-Premises AD to the AWS Cloud by using AD trusts.

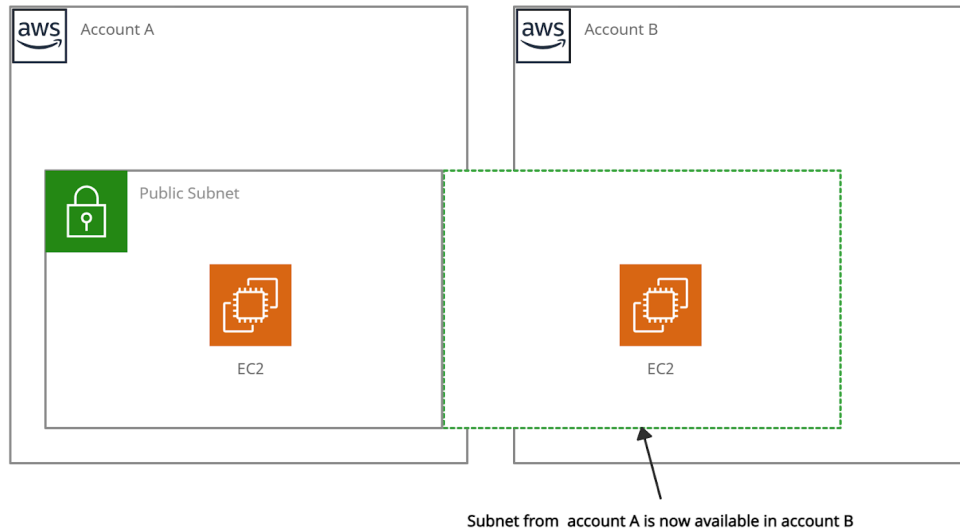
### **Pricing:**

- Prices vary by region for the directory service.
- Hourly charges are applied for each additional account to which a directory is shared.
- Charges are applied per GB for the data transferred “out” to other AWS Regions where the directory is deployed.

# AWS Resource Access Manager

## What is AWS Resource Access Manager?

AWS Resource Access Manager (RAM) is a service that permits users to share their resources across AWS accounts or within their AWS Organization.



*AWS Resource Access Manager*

## Resources that can be integrated with AWS RAM are:

- AWS App Mesh
- Amazon Aurora
- AWS Certificate Manager Private Certificate Authority
- AWS CodeBuild
- EC2 Image Builder
- AWS Glue
- AWS License Manager
- AWS Network Firewall
- AWS Outposts
- AWS Resource Groups

## Benefits:

- The resource sharing feature of AWS RAM reduces customers' need to create duplicate resources in each of their accounts.
- It controls the consumption of shared resources using existing policies and permissions.
- It can be integrated with Amazon CloudWatch and AWS CloudTrail to provide detailed visibility into shared resources and accounts.
- Access control policies in AWS Identity & Access Management (IAM) and Service Control Policies in AWS Organizations provide security and governance controls to AWS Resource Access Manager (RAM).

## Price details:

- The charges only differ based on the resource type. No charges are applied for creating resource shares and sharing your resources across accounts.

# AWS Secrets Manager

## What is AWS Secrets Manager?

AWS Secrets Manager is a service that replaces secret credentials in the code like passwords, with an API call to retrieve the secret programmatically. The service provides a feature to rotate, manage, and retrieve database passwords, OAuth tokens, API keys, and other secret credentials. It ensures in-transit encryption of the secret between AWS and the system to retrieve the secret.

Secrets Manager can easily rotate credentials for AWS databases without any additional programming. Though rotating the secrets for other databases or services requires Lambda function to instruct how Secrets Manager interacts with the database or service.

## Accessing Secrets Manager:

- AWS Management Console
  - It stores binary data in secret.
- AWS Command Line Tools
  - AWS Command Line Interface
  - AWS Tools for Windows PowerShell
- AWS SDKs
- Secrets Manager HTTPS Query API

Secret rotation is available for the following Databases:

- MySQL on Amazon RDS
- PostgreSQL on Amazon RDS
- Oracle on Amazon RDS
- MariaDB on Amazon RDS
- Amazon DocumentDB
- Amazon Redshift
- Microsoft SQL Server on Amazon RDS
- Amazon Aurora on Amazon RDS

## Features:

- It provides security and compliance facilities by rotating secrets safely without the need for code deployment.
- With Secrets Manager, IAM policies and resource-based policies can assign specific permissions for developers to retrieve secrets and passwords used in the development environment or the production environment.
- Secrets can be secured with encryption keys managed by AWS Key Management Service (KMS).
- It integrates with AWS CloudTrail and AWS CloudWatch to log and monitor services for centralized auditing.

## Use cases:

- Store sensitive information as part of the encrypted secret value, either in the SecretString or SecretBinary field.
- Use a Secrets Manager open-source client component to cache secrets and update them only when there is a need for rotation.



- When an API request quota exceeds, the Secrets Manager throttles the request and returns a 'ThrottlingException' error. To resolve this, retry the requests.
- It integrates with AWS Config and facilitates tracking of changes in Secrets Manager.

**Price details:**

- There are no upfront costs or long-term contracts.
- Charges are based on the total number of secrets stored and API calls made.
- AWS charges at the current AWS KMS rate if the customer master keys(CMK) are created using AWS KMS.

# AWS Security Hub

## What is AWS Security Hub?

AWS Security Hub is a service that provides an extensive view of the security aspects of AWS and helps to protect the environment against security industry standards and best practices.

It provides an option to aggregate, organize, and prioritize the security alerts, or findings from multiple AWS services, such as Amazon GuardDuty, Amazon Inspector, Amazon Macie, AWS IAM Access Analyzer, AWS Firewall Manager, and also from AWS Partner solutions.

It helps the Payment Card Industry Data Security Standard (PCI DSS) and the Center for Internet Security (CIS) AWS Foundations Benchmark with a set of security configuration best practices for AWS. If any problem occurs, AWS Security Hub recommends remediation steps.

Enabling (or disabling) AWS Security Hub can be quickly done through,

- AWS Management Console
- AWS CLI
- By using Infrastructure-as-Code tools -- Terraform

If AWS architecture is divided across multiple regions, it needs to enable Security Hub within each region.

The most powerful aspect of using Security Hub is the continuous automated compliance checks using CIS AWS Foundations Benchmark.

The CIS AWS Foundations Benchmark consists of 43 best practice checks (such as “Ensure IAM password policy requires at least one uppercase letter” and “Ensure IAM password policy requires at least one number”).

## Benefits:

- It collects data using a standard findings format and reduces the need for time-consuming data conversion efforts.
- Integrated dashboards are provided to show the current security and compliance status.

## Price details:

- Charges applied for usage of other services that Security Hub interacts with, such as AWS Config items, but not for AWS Config rules that are enabled by Security Hub security standards.
- Using the Master account's Security Hub, the monthly cost includes the costs associated with all of the member accounts.
- Using a Member account's Security Hub, the monthly cost is only for the member account.
- Charges are applied only for the current Region, not for all Regions in which Security Hub is enabled.

# AWS Key Management Service

## What is AWS Key Management Service?

AWS Key Management Service (AWS KMS) is a secured service to create and control the encryption keys. It is integrated with other AWS services such as Amazon EBS, Amazon S3 to provide data at rest security with encryption keys. KMS is a global service but keys are regional which means you can't send keys outside the region in which they are created.

**Customer master keys (CMKs):** The CMK contains metadata, such as key ID, creation date, description, key state, and key material to encrypt and decrypt data. AWS KMS supports symmetric and asymmetric CMKs:

- Symmetric CMK constitutes a 256-bit key that is used for encryption and decryption.
- An asymmetric CMK resembles an RSA key pair that is used for encryption and decryption or signing and verification (but not both), or an elliptic curve (ECC) key pair that is used for signing and verification.
- Both symmetric CMKs and the private keys of asymmetric CMKs never leave AWS KMS unencrypted.

## Customer managed CMKs:

- Customer-managed CMKs are CMKs that are created, owned, and managed by user full control.
- Customer-managed CMKs are visible on the Customer-managed keys page of the AWS KMS Management Console.
- Customer-managed CMKs can be used in cryptographic operations.

## AWS managed CMKs:

- AWS managed CMKs are CMKs that are created, managed, and used on the user's behalf by an AWS service that is integrated with AWS KMS.
- AWS managed CMKs are visible on the AWS managed keys page of the AWS KMS Management Console.
- It can not be used in cryptographic operations.

**Envelope encryption** is the method of encrypting plain text data with a data key and then encrypting the data key under another key.

Envelope encryption offers several benefits:

- Protecting data keys.
- Encrypting the same data under multiple master keys.
- Combining the strengths of multiple algorithms.

## Features:

- The automatic rotation of master keys generated in AWS KMS once per year is done without the need to re-encrypt previously encrypted data.
- Using AWS CloudTrail, each request to AWS KMS is recorded in a log file that is delivered to the specified Amazon S3 bucket. Log information includes details of the user, time, date, API action, and the key used.
- This service automatically scales as the encryption grows.
- For the high availability of data and keys, KMS stores multiple copies of an encrypted version of keys.

**Benefits:**

- Key Management Service (KMS) with Server-side Encryption in S3.
- Manage encryption for AWS services.

**Price details:**

- Provides a free tier of 20,000 requests/month across all regions where the service is available.
- Each customer master key (CMK) that you create in AWS Key Management Service (KMS) costs \$1 per month until deleted.
- Creation and storage of AWS-managed CMKs are not charged as they are created on the user's behalf by AWS.
- Customer-managed CMKs are scheduled for deletion but it will incur charges if deletion is canceled during the waiting period.

# AWS Certificate Manager (ACM)

## What is AWS Certificate Manager?

AWS Certificate Manager is a service that allows a user to provide, manage, renew and deploy public and private Secure Sockets Layer/Transport Layer Security (SSL/TLS) X.509 certificates.

The certificates can be integrated with AWS services either by issuing them directly with ACM or importing third-party certificates into the ACM management system.

## SSL Server Certificates:

- HTTPS transactions require server certificates X.509 that bind the public key in the certificate to provide authenticity.
- The certificates are signed by a certificate authority (CA) and contain the server's name, the validity period, the public key, the signature algorithm, and more.

## The different types of SSL certificates are:

- Extended Validation Certificates (EV SSL) - most expensive SSL certificate type
- Organization Validated Certificates (OV SSL) - validate a business' creditably
- Domain Validated Certificates (DV SSL) - provide minimal encryption
- Wildcard SSL Certificate - secures base domain and subdomains
- Multi-Domain SSL Certificate (MDC) - secure up to hundreds of domain and subdomains
- Unified Communications Certificate (UCC) - single certificate secures multiple domain names.

## Ways to deploy managed X.509 certificates:

1. AWS Certificate Manager (ACM) - useful for large customers who need a secure web presence.
  - ACM certificates are deployed using Amazon API Gateway, Elastic Load Balancing, Amazon CloudFront.
2. ACM Private CA - useful for large customers building a public key infrastructure (PKI) inside the AWS cloud and intended for private use within an organization.
  - It helps create a certificate authority (CA) hierarchy and issue certificates to authenticate users, computers, applications, services, servers, and other devices.
  - Private certificates by Private CA for applications provide variable certificate lifetimes or resource names.

## ACM certificates are supported by the following services:

- Elastic Load Balancing
- Amazon CloudFront
- AWS Elastic Beanstalk
- Amazon API Gateway
- AWS Nitro Enclaves (an Amazon EC2 feature)
- AWS CloudFormation

**Benefits:**

- It automates the creation and renewal of private certificates for on-premises and AWS resources.
- It provides an easy process to create certificates. Just submit a CSR to a Certificate Authority, or upload and install the certificate once received.
- SSL/TLS provides data-in-transit security, and SSL/TLS certificates authorize the identity of sites and connections between browsers and applications.

**Price details:**

- The certificates created by AWS Certificate Manager for using ACM-integrated services are free.
- With AWS Certificate Manager Private Certificate Authority, monthly charges are applied for the operation of the private CA and the private certificates issued.

# AWS Auto Scaling

## What is AWS Auto Scaling?

- AWS Auto Scaling keeps on monitoring your Application and automatically adjusts the capacity required for steady and predictable performance.
- By using auto scaling it's very easy to set up the scaling of the application automatically with no manual intervention.
- It allows you to create scaling plans for the resources like EC2 Instances, Amazon EC2 tasks, Amazon DynamoDB, Amazon Aurora Read Replicas.
- It balances Performance Optimization and cost.

## Terminologies related to AWS Autoscaling Groups:

### Launch Configuration vs Launch Template

- o EC2 Auto Scaling uses two types of instance configuration templates: launch configurations and launch templates.
- o We recommend that you use launch templates to make sure that you're getting the latest features from Amazon EC2.
- o For example, you must use launch templates to use Dedicated Hosts, which enable you to bring your eligible software licenses from vendors, including Microsoft, and use them on EC2.
- o If you intend to use a launch configuration with EC2 Auto Scaling, be aware that not all Auto Scaling group features are available.
- o If you want to launch on-demand and spot both instances you have to choose a launch template.

### Auto Scaling Lifecycle Hooks:

- The Lifecycle hook will pause your EC2 instance.
- The paused instances will remain in the wait state until the action is completed.
- The Wait state will remain active till the timeout period ends.

## Monitoring:

- **Health Check:** Keep on checking the health of the instance and remove the unhealthy instance out of Target Group.
- **CloudWatch Events:** AutoScaling can submit events to Cloudwatch for any type of action to perform in the autoscaling group such as a launch or terminate an instance.
- **CloudWatch Metrics:** It shows you the statistics of whether your application is performing as expected.
- **Notification Service:** Autoscaling can send a notification to your email if the autoscaling group launches or the instance gets terminated.

## Charges:

- AWS will not charge you additionally for the Autoscaling Group.
- You will be paying for the AWS Resources that you will use.

# AWS CloudFormation

## What is AWS CloudFormation?

AWS CloudFormation is a service that collects AWS and third-party resources and manages them throughout their life cycles, by launching them together as a stack.

A template is used to create, update, and delete an entire stack as a single unit, without managing resources individually.

It provides the capability to reuse the template to set the resources easily and repeatedly.

It can be integrated with AWS IAM for security. It can be integrated with CloudTrail to capture API calls as events.

**Templates** - A JSON or YAML formatted text file used for building AWS resources.

**Stack** - It is a single unit of resources.

**Change sets** - It allows checking how any change to a resource might impact the running resources.

**Stacks** can be created using the AWS CloudFormation console and AWS Command Line Interface (CLI).

**Stack updates:** First the changes are submitted and compared with the current state of the stack and only the changed resources get updated.

There are two methods for updating stacks:

- **Direct update** - when there is a need to quickly deploy the updates.
- **Creating and executing change sets** - they are JSON files, providing a preview option for the changes to be applied.

**StackSets** are responsible for safely provisioning, updating, or deleting stacks.

**Nested Stacks** are stacks created within another stack by using the `AWS::CloudFormation::Stack` resource.

When there is a need for common resources in the template, Nested stacks can be used by declaring the same components instead of creating the components multiple times. The main stack is termed as parent stack and other belonging stacks are termed as child stack, which can be implemented by using ref variable '`!Ref`'.

**AWS CloudFormation Registry** helps to provision third-party application resources alongside AWS resources. Examples of third-party resources are incident management, version control tools.

## Price details:

- AWS does not charge for using AWS CloudFormation, charges are applied for the services that the CloudFormation template comprises.
- AWS CloudFormations supports the following namespaces: `AWS::*`, `Alexa::*`, and `Custom::*`. If anything else is used except these namespaces, charges are applied per handler operation.
- Free tier - 1000 handler operations per month per account
- Handler operation - \$0.0009 per handler operation



### **Example: CloudFormation template for creating EC2 instance**

EC2Instance:

Type: AWS::EC2::Instance

Properties:

ImageId: 1234xyz

KeyName: aws-keypair

InstanceType: t2.micro

SecurityGroups:

- !Ref EC2SecurityGroup

BlockDeviceMappings:

- DeviceName: /dev/sda1

Ebs:

VolumeSize: 50

# AWS CloudTrail

## What is AWS CloudTrail?

AWS CloudTrail is defined as a global service that permits users to enable operational and risk auditing of the AWS account.

It allows users to view, search, download, archive, analyze, and respond to account activity across the AWS infrastructure.

It records actions as an event taken by a user, role, or an AWS service in the AWS Management Console, AWS Command Line Interface, and AWS SDKs and APIs.

AWS CloudTrail mainly integrates with:

- Amazon S3 can be used to retrieve log files.
- Amazon SNS can be used to notify about log file delivery to the bucket with Amazon Simple Queue Service (SQS).
- Amazon CloudWatch for monitoring and AWS Identity and Access Management (IAM) for security.

CloudTrail events of the past 90 days recorded by CloudTrail can be viewed in the CloudTrail console and can be downloaded in CSV or JSON file.

Trail log files can be aggregated from multiple accounts to a single bucket and can be shared between accounts.

AWS CloudTrail Insights enables AWS users to identify and respond to unusual activities of API calls by analyzing CloudTrail management events.

There are three types of CloudTrail events:

- Management events or control plane operations
  - Example - Amazon EC2 *CreateSubnet* API operations and *CreateDefaultVpc* API operations
- Data events
  - Example - S3 Bucket *GetObject*, *DeleteObject*, and *PutObject* API operations
- CloudTrail Insights events (unusual activity events)
  - Example - Amazon S3 *deleteBucket* API, Amazon EC2 *AuthorizeSecurityGroupIngress* API

## Example of CloudTrail log file:

### IAM log file -

The below example shows that the IAM user Rohit used the AWS Management Console to call the *AddUserToGroup* action to add Nayan to the administrator group.

```
{
  "Records": [
    {
      "eventVersion": "1.0",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "PR_ID",
        "arn": "arn:aws:iam::210123456789:user/Rohit",
        "accountId": "210123456789",
        "accessKeyId": "KEY_ID",
        "userName": "Rohit"
      }
    }
  ],
}
```

```

"eventTime": "2021-01-24T21:18:50Z",
"eventSource": "iam.amazonaws.com",
"eventName": "CreateUser",
"awsRegion": "ap-south-2",
"sourceIPAddress": "176.1.0.1",
"userAgent": "aws-cli/1.3.2 Python/2.7.5 Windows/7",
"requestParameters": {"userName": "Nayan"},
"responseElements": {"user": {
  "createDate": "Jan 24, 2021 9:18:50 PM",
  "userName": "Nayan",
  "arn": "arn:aws:iam::128x:user/Nayan",
  "path": "/",
  "userId": "12xyz"
}}
}}

```

CloudWatch monitors and manages the activity of AWS services and resources, reporting on their health and performance. Whereas, CloudTrail resembles logs of all actions performed inside the AWS environment.

#### **Price details:**

- Charges are applied based on the usage of Amazon S3.
- Charges are applied based on the number of events analyzed in the region.
- The first copy of Management events within a region is free, but charges are applied for additional copies of management events at \$2.00 per 100,000 events.
- Data events are charged at \$0.10 per 100,000 events.
- CloudTrail Insights events provide visibility into unusual activity and are charged at \$0.35 per 100,000 write management events analyzed.

# Amazon CloudWatch

## What is Amazon CloudWatch?

Amazon CloudWatch is a service that helps to monitor and manage services by providing data and actionable insights for AWS applications and infrastructure resources.

It monitors AWS resources such as Amazon RDS DB instances, Amazon EC2 instances, Amazon DynamoDB tables, and, as well as any log files generated by the applications.

Amazon CloudWatch can be accessed by the following methods:

- Amazon CloudWatch console
- AWS CLI
- CloudWatch API
- AWS SDKs

Amazon CloudWatch is used together with the following services:

- Amazon Simple Notification Service (Amazon SNS)
- Amazon EC2 Auto Scaling
- AWS CloudTrail
- AWS Identity and Access Management (IAM)

It collects monitoring data in the form of logs, metrics, and events from AWS resources, applications, and services that run on AWS and on-premises servers. Some metrics are displayed on the home page of the CloudWatch console. Additional custom dashboards to display metrics can be created by the user.

Alarms can be created using CloudWatch Alarms that monitor metrics and send notifications or make automatic changes to the resources based on actions whenever a threshold is breached.

CloudWatch console provides Cross-account functionality which provides cross-account visibility to the dashboards, alarms, metrics, and dashboards without Sign-in and Sign-out of different accounts. This functionality becomes more useful if the accounts are managed by AWS Organizations.

CloudWatch Container Insights are used to collect and summarize metrics and logs from containerized applications. These Insights are available for Amazon ECS, Amazon EKS, and Kubernetes platforms on Amazon EC2.

CloudWatch Lambda Insights are used to collect and summarize system-level metrics including CPU time, memory, disk, and network for serverless applications running on AWS Lambda.

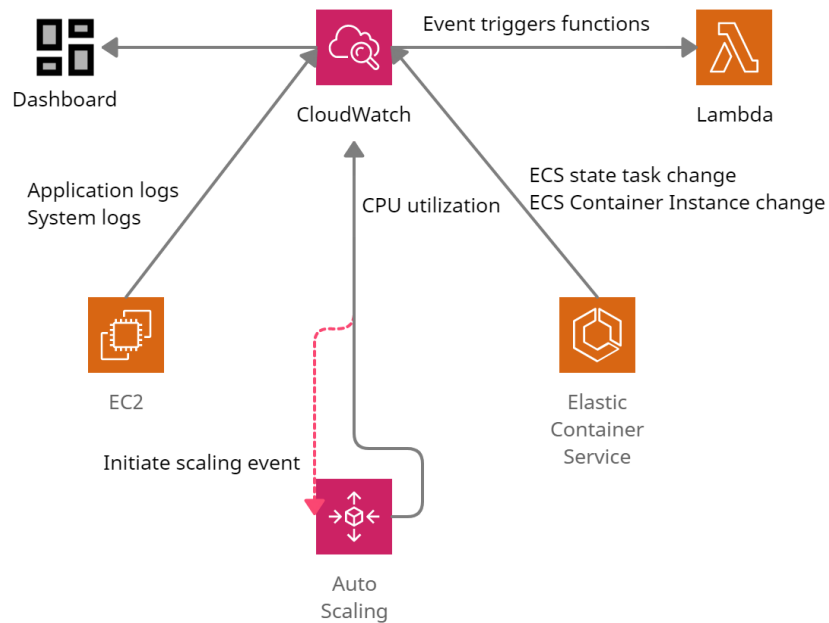
CloudWatch agent is installed on the EC2 instance to provide the following features:

- It collects system-level metrics from Amazon EC2 instances or on-premises servers across operating systems.
- It collects custom metrics from the applications using the StatsD and collectd protocols.

StatsD - supported on both Linux servers and Windows Server

collectd - supported only on Linux servers.

- The metrics from the CloudWatch agent can be collected and stored in CloudWatch just like any other CloudWatch metrics.
- The default namespace for the CloudWatch agent metrics is CWAgent, and can be changed while configuring the agent.



*Amazon CloudWatch in action*

# AWS Config

## What is AWS Config?

AWS Config is a service that continuously monitors and evaluates the configurations of the AWS resources (services).

It helps to view configuration changes performed over a specific period of time using AWS Config console and AWS CLI.

It evaluates AWS resource configurations based on specific settings and creates a snapshot of the configurations to provide a complete inventory of resources in the account.

It retrieves previous configurations of resources and generates notifications whenever a resource is created, modified, or deleted.

It uses Config rules to evaluate configuration settings of the AWS resources. AWS Config also checks any condition violation in the rules. There can be 150 AWS Config rules per region.

- Managed Rules
- Custom Rules

It is integrated with AWS IAM, to create permission policies attached to the IAM role, Amazon S3 buckets, and Amazon Simple Notification Service (Amazon SNS) topics.

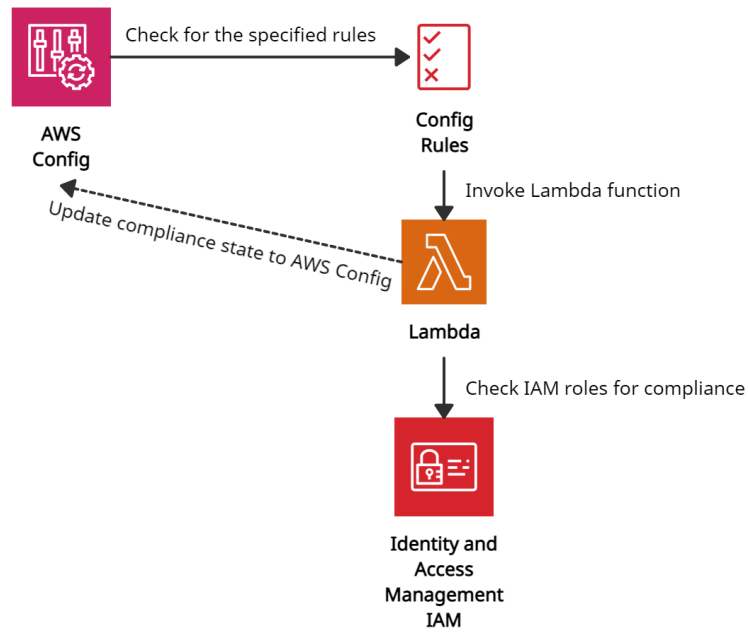
It is also integrated with AWS CloudTrail, which provides a record of user actions or an AWS Service by capturing all API calls as events in AWS Config.

AWS Config provides an aggregator (a resource) to collect AWS Config configuration and compliance data from:

- Multiple accounts and multiple regions.
- Single account and multiple regions.
- An organization in AWS Organizations
- The Accounts in the organization which have AWS Config enabled.

## Use Cases:

- It enables the user to code custom rules in AWS Lambda that define the best guidelines for resource configurations. Users can also automate the assessment of the resource configuration changes to ensure compliance and self-governance across your AWS infrastructure.
- Data from AWS Config allows users to continuously monitor the configurations for potential security weaknesses. After any security alert, Config allows the user to review the configuration history and understand the risk factor.



*AWS Config in action*

**Price details:**

- Charges are applied based on the total number of configuration items recorded at the rate of \$0.003 per configuration item recorded per AWS Region in the AWS account.
- For Config rules, charges are applied based on the number of AWS Config rules evaluated.
- Additional charges are applied if AWS Config integrates with other AWS Services at a standard rate.

# AWS License Manager

## What is AWS License Manager?

- AWS License Manager is a service that manages software licenses in AWS and on-premises environments from vendors such as Microsoft, SAP, Oracle, and IBM.
- It supports Bring-Your-Own-License (BYOL) feature which means that users can manage their existing licenses for third-party workloads (Microsoft Windows Server, SQL Server) to AWS.
- It enables administrators to create customized licensing rules that help to prevent licensing violations (using more licenses than the agreement).
- The rules operate by stopping the instance from launching or by notifying administrators about the infringement (violation of a law).
- Administrators use rule-based controls on the consumption of licenses, to set limits on new and existing cloud deployments.
- **Hard limit** - does not allow the launch of non-compliant instances
- **Soft limit** - allow the launch of non-compliant instance but sends an alert to the administrators
- It provides control and visibility of all the licenses to the administrators with the help of the AWS License Manager dashboard.
- It allows administrators to specify Dedicated Host management preferences for allocation and capacity utilization.
- AWS License Manager's managed entitlements provide built-in controls to software vendors (ISVs) and administrators so that they can assign licenses to approved users and workloads.
- AWS Systems Manager can manage licenses on physical or virtual servers hosted outside of AWS using AWS License Manager.
- AWS Systems Manager helps to discover software running on existing EC2 instances and then rules can be attached and validated in EC2 instances allowing the licenses to be tracked using the License Manager's dashboard.
- AWS Organizations along with AWS License Manager helps to allow cross-account disclosure of computing resources in the organization by using service-linked roles and enabling trusted access between License Manager and Organizations.

AWS License Manager is integrated with the following services:

- AWS Marketplace
- Amazon EC2
- Amazon RDS
- AWS Systems Manager
- AWS Identity and Access Management (IAM)
- AWS Organizations
- AWS CloudFormation
- AWS X-Ray

## Price details:

- Charges are applied at normal AWS rates only for the AWS resources integrated with AWS License Manager.



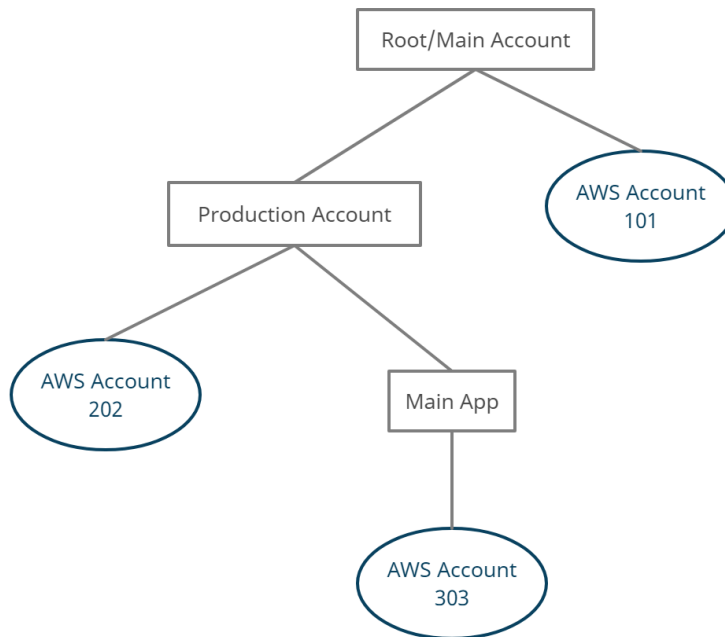
# AWS Organizations

## What are AWS Organizations?

AWS Organizations is a global service that enables users to consolidate and manage multiple AWS accounts into an organization.

It includes account management and combined billing capabilities that help to meet the budgetary, and security needs of the business better.

- The main account is the management account – it cannot be changed.
- Other accounts are member accounts that can only be part of a single organization.



*AWS Organizations flow*

AWS Organizations can be accessed in the following ways:

- AWS Management Console
- AWS Command Line Tools
  - AWS Command Line Interface (AWS CLI)
  - AWS Tools for Windows PowerShell.
- AWS SDKs
- AWS Organizations HTTPS Query API

## Features:

- AWS Organizations provides security boundaries using multiple member accounts.
- It makes it easy to share critical common resources across the accounts.
- It organizes accounts into organizational units (OUs), which are groups of accounts that serve specified applications.
- Service Control Policies (SCPs) can be created to provide governance boundaries for the OUs. SCPs ensure that users in the accounts only perform actions that meet security requirements.

- Cost allocation tags can be used in individual AWS accounts to categorize and track the AWS costs.
- It integrates with the following services:
  - AWS CloudTrail - Manages auditing and logs all events from accounts.
  - AWS Backup - Monitor backup requirements.
  - AWS Control Tower - to establish cross-account security audits and view policies applied across accounts.
  - Amazon GuardDuty - Managed security services, such as detecting threats.
  - AWS Resource Access Manager (RAM) - Can reduce resource duplication by sharing critical resources within the organization.
- Steps to be followed for migrating a member account:
  - Remove the member account from the old Organization.
  - Send an invitation to the member account from the new Organization.
  - Accept the invitation to the new Organization from the member account.

**Price details:**

- AWS Organizations is free. Charges are applied to the usage of other AWS resources.
- The management account is responsible for paying charges of all resources used by the accounts in the organization.
- AWS Organizations provides consolidated billing that combines the usage of resources from all accounts, and AWS allocates each member account a portion of the overall volume discount based on the account's usage.

# AWS Systems Manager

## What is AWS Systems manager?

AWS Systems Manager is a service which helps users to manage EC2 and on-premises systems at scale. It not only detects the insights about the state of the infrastructure but also easily detects problems as well.

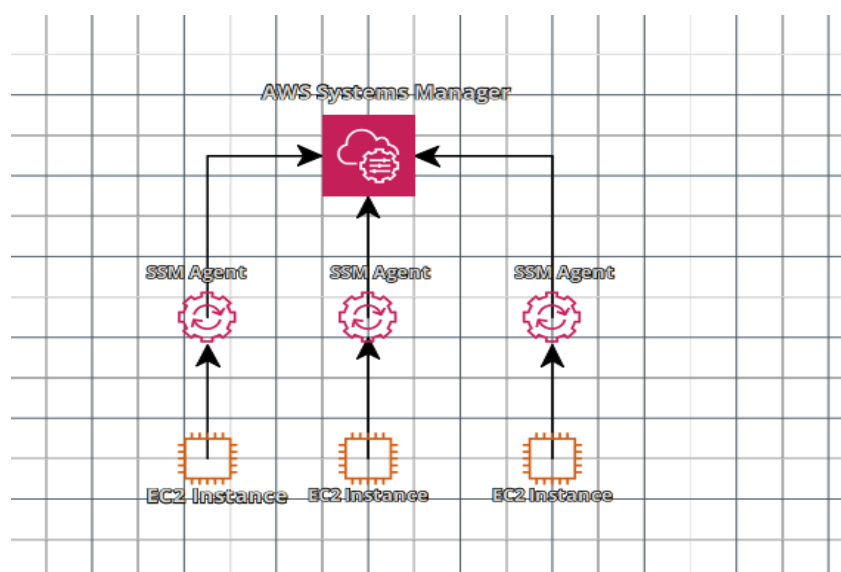
Additionally, we can patch automation for enhanced compliance. This AWS service works for both Windows and Linux operating systems.

## Features:

- Easily integrated with CloudWatch metrics/dashboards and AWS Config.
- It helps to discover and audit the software installed.
- Compliance management
- We can group more than 100 resource types into applications, business units, and environments.
- It helps to view instance information such as operating system patch levels, install software and see the compliance with the desired state.
- Act associate and configurations with resources and find out the discrepancies.
- Distribute multiple software versions safely across the instances.
- Increase the security area by running a command or maintaining scripts.
- Patch your instances of schedule to keep them compliant.
- Helps managers to automate workflows.
- It helps to reduce errors by securely applying configurable parameters into centralized service.

## How does the System Manager work?

Firstly, User needs to install the SSM agent on the system they control. If an instance can't be controlled with SSM, it's probably an issue with the SSM agent. Also, we need to make sure all the EC2 instances have a proper IAM role to allow SSM actions.



**Pricing:**

- App Config:
  - Get Configuration API Calls: \$0.2 per 1M Get Configuration calls
  - Configurations Received: \$0.0008 per configuration received
- Parameter Store:
  - Standard: No additional charge.
  - Advanced: \$0.05 per advance parameter per month.
- Change Manager:
  - Number of change requests: \$0.296 per change request.
  - Get, described, Update, and GetoptsSummary API requests: \$0.039 per 1000 requests.

# AWS CodeBuild

## What is AWS CodeBuild?

Amazon codeBuild is a fully managed continuous integration service that helps developers to build and run test code rapidly. This service also provides test artifacts with great efficiency.

This gives an advantage to developers/DevOps engineers not to wait in long build queues, scaling, configuring, and maintaining build service. AWS CodeBuild runs continuously and avoids the wait time for concurrent jobs. Additionally, users pay only for the build time they use. In other words, the ideal time won't be counted in billing time for the user.

## Features:

1. Easy to Set up – Amazon CodeBuild is easy to set up for the developers. Either they can build their code build environment or use one of the preconfigured environments.
2. It works with existing tools such as Jenkins plugin, GIT, etc.
3. It can scale automatically with several concurrent builds.
4. Automated Build: Developers need to configure builds once and whenever there is a code change, CodeBuild service will automatically run and generate the test results.
5. Pay as you go which means developers are only charged for the time it takes to complete the build, idle time won't be considered in billing.

## Pricing:

Pricing will be computed based on the build minutes. The first 100 minutes of the build are free and then the rest will be charged based on each instance type usage.

# AWS CodeCommit

## What Is AWS CodeCommit?

AWS CodeCommit is a version control service hosted by Amazon Web Services that users can use privately to store and manage assets (such as documents, source code, and binary files) in the cloud.

In other words, AWS provides the service which allows you to just store the code or any assets without worrying about the version control like other version control tools such as Bitbucket, GitHub, etc. AWS manages everything on its own and takes full responsibility for scaling its infrastructure.

## Features:

- Ensures High secure code (encrypted) with any type of code.
- Collaborative work (users can access the same piece of code with different IAM users and different security groups).
- Easy scalability.
- Easy to integrate with third-party groups.

## Pricing:

AWS CodeCommit provides free tier term for the first 5 active users for the below configurations:

First 5 active user receives:

- Unlimited repositories
- 50 GB storage
- 10K Git requests/month

Additional active user beyond the first 5 users:

- Unlimited repositories
- 10 GB storage
- 2K Git requests/month

There will be additional charges for additional storage or an increase in GIT requests if increased.

# AWS CodeDeploy

## What is AWS CodeDeploy?

AWS CodeDeploy is a deployment service that automates application deployments to Amazon EC2 instances, on-premises instances, serverless Lambda functions, or Amazon ECS services.

Below type of deployments can be done using AWS CodeDeploy service:

- Code, Serverless Lambda Functions.
- Web & Configuration Files
- Executables and Packages.
- Scripts and Multimedia.

Following are components that concerns AWS CodeDeploy Service:

- Compute Platform
- Deployment Types & Groups.
- IAM & Service Roles
- Applications.

## How does AWS CodeDeploy work?

It is divided into 3 parts. There might be multiple versions available for your application. First, the developer has to finalize the application revision which needs to be deployed. Then deployment configuration needs to be finalized. This is an app specification file(YML extension) that contains information such as source/destination location etc. and the last part is, deploy the appropriate revision to cloud location which is called deployment group.

## Features:

- Help to release new features rapidly.
- It supports avoiding downtime during application deployment by maximizing application availability and handles all application complexity.
- It allows easy launch and tracking of application status.

## Pricing:

- Free code deployment to Amazon EC2 or AWS Lambda.
- \$0.02 charges per on-premises instance deployment.

# AWS X-Ray

## What is AWS X-Ray?

AWS X-Ray helps to analyze and debug production systems in distributed environments. AWS X-Ray is a method used to profile and monitor applications, mostly built using microservice architecture.

Using X-Ray service features, it's very easy for a user to troubleshoot and find the root cause of slowness of the system, including other performance-related errors.

## X-Ray components:

- Daemon: a unique application that collects related segment data.
- Segments: provides resource names, requests information.
- Subsegments: This provides granular details about downstream calls.
- Service Graph: Visual representation of micro-service response or failure.
- Traces: Collects all segments created from a single request.
- Sampling: Algorithm which decides which request needs to be traced.
- Filter Expressions: easier to deep dive to understand the particular path.

## Features:

- Supports all language (Go, NodeJS, Ruby, Java, Python, ASP.NET, PHP)
- It supports AWS service integration with Lambda, API Gateway, App Mesh, CloudTrail, CloudWatch, AWS Config, EB, ELB, SNS, SQS, EC2, ECS, Fargate.
- Helps in improving application performance.
- Easy to discover application issues with insights provided by X-Ray.
- It helps in tracking user requests as they travel through the entire application.

## Pricing:

- Traces recorded/retrieved/scanned will cost \$.50 per 1 million requests beyond the free tier.



# AWS Database Migration Service

## What is AWS Database Migration Service?

AWS Database Migration Service is a cloud service used to migrate relational databases from on-premises, Amazon EC2, or Amazon RDS to AWS securely. It does not stop the running application while performing the migration of databases, resulting in downtime minimization.

It performs homogeneous as well as heterogeneous migrations between different database platforms.

MySQL - MySQL (homogeneous migration)

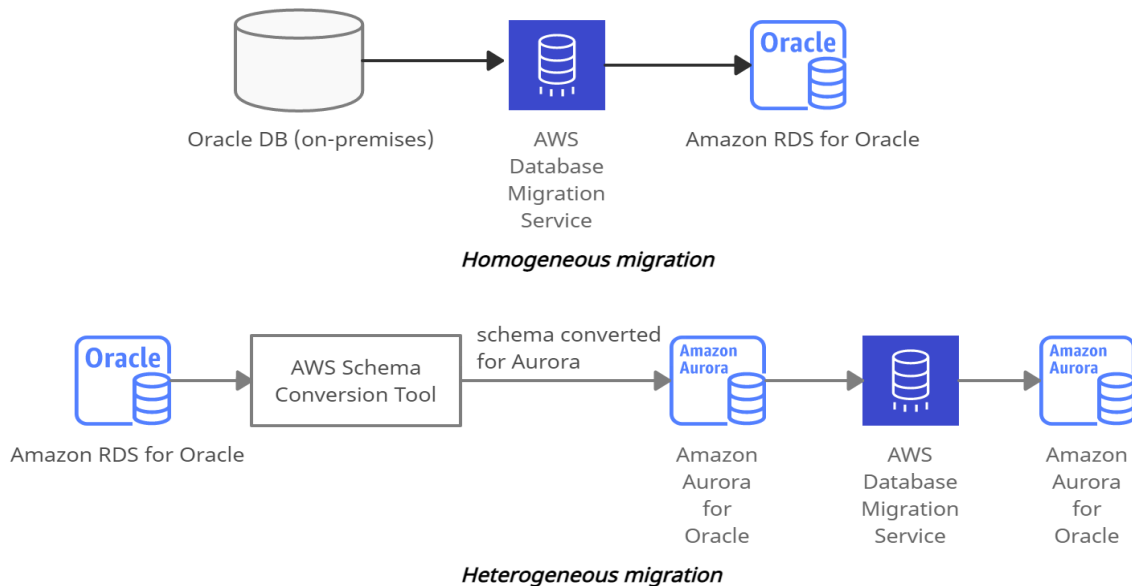
MySQL - Amazon Aurora (heterogeneous migration)

AWS DMS supports the following data sources and targets engines for migration:

- **Sources:** Oracle, Microsoft SQL Server, PostgreSQL, Db2 LUW, SAP, MySQL, MariaDB, MongoDB, and Amazon Aurora.
- **Targets:** Oracle, Microsoft SQL Server, PostgreSQL, SAP ASE, MySQL, Amazon Redshift, Amazon S3, and Amazon DynamoDB.

It performs all the management steps required during the migration, such as monitoring, scaling, error handling, network connectivity, replicating during failure, and software patching.

AWS DMS with AWS Schema Conversion Tool (AWS SCT) helps to perform heterogeneous migration.



*AWS Database Migration Service*

# Amazon API Gateway

## What is Amazon API Gateway?

Amazon API Gateway is a service which creates, publishes, maintains, monitors and secures APIs at any scale.

- It helps to create Synchronous microservices with Load Balancers and forms the app-facing part of the AWS serverless infrastructure with AWS Lambda.
- It handles the tasks involved in processing concurrent API calls.
- It combines with Amazon EC2, AWS Lambda or any web application (public or private endpoints) to work as back-end services.

API Gateway creates RESTful APIs that:

- Are HTTP-based.
- Enable stateless and client-server communication.
- Create standard HTTP methods such as GET, POST, PUT, PATCH and DELETE.

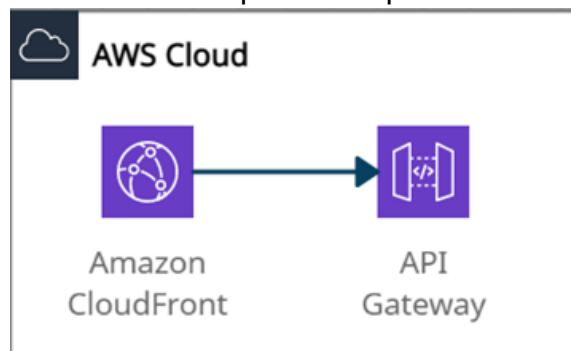
API Gateway creates WebSocket APIs that:

- Follow WebSocket protocol and enable stateful, full-duplex communication between client and server.
- Route incoming messages to the destination based on message content.

Endpoint Types for API Gateway:

### Edge-optimized endpoint:

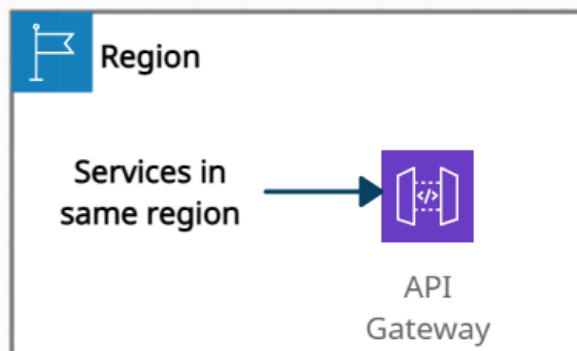
- It signifies reduced latency for requests all around the world.
- CloudFront is also used as the public endpoint.



*Edge-optimized endpoint*

### Regional endpoint:

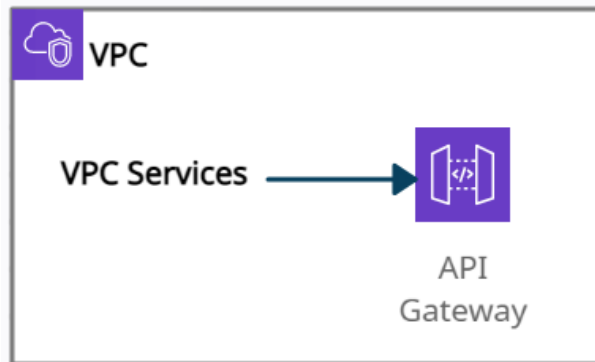
- It signifies reduced latency for requests that originate in the same region. It can also configure the CDN and protect WAF.



*Regional endpoint*

**Private endpoint:**

- It securely exposes the REST APIs to other services only within the VPC.



*Private endpoint*

**API Gateway - Securities:**

- Resource-based policies
- IAM Permissions
- Lambda Authorizer (formerly Custom Authorizers)
- Cognito user pools

**Features:**

- It helps to create stateful (WebSocket) and stateless (HTTP and REST) APIs.
- It integrates with CloudTrail for logging and monitoring API usage and API changes.
- It integrates with CloudWatch metrics to monitor REST API execution and WebSocket API execution.
- It integrates with AWS WAF to protect APIs against common web exploits.
- It integrates with AWS X-Ray for understanding and triaging performance latencies.

**Price details:**

- You pay for API Caching as it is not eligible for the AWS Free Tier.
- API requests are not charged for authorization and authentication failures.
- Method calls which consist of API keys are not charged if API keys are missing or invalid.
- API Gateway-throttled and plan-throttled requests are not charged if the request rate exceeds the predefined limits.

# AWS Cloud Map

## What is AWS Cloud Map?

AWS Cloud Map is a service that keeps track of application components, location dependencies, attributes and health status, and also allows dynamic scaling and responsiveness of the application.

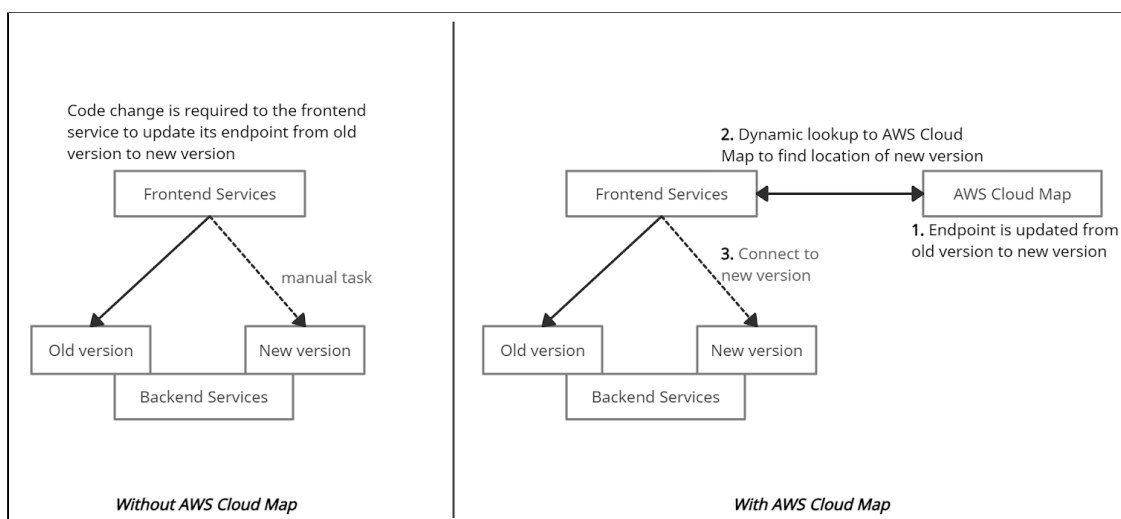
It uses AWS Cloud Map DiscoverInstances API calls, DNS queries in a VPC, or public DNS queries to locate resources in the backend.

A service is a template that is used by AWS Cloud Map when an application adds another resource, such as database servers.

When the application needs to connect to a resource, AWS Cloud Map calls API DiscoverInstances and specifies the namespace and service that are associated with the resource. AWS Cloud Map only returns healthy instances if health checking is specified while creating the service.

When an application adds a resource, a service instance is created by calling the AWS Cloud Map *RegisterInstance* API action. The service instance helps to locate the resource, by using DNS or using the AWS Cloud Map DiscoverInstances API action.

It can be used to register and locate any cloud resources, such as Amazon EC2 instances, Amazon Simple Queue Service (Amazon SQS) queues, Amazon DynamoDB tables, Amazon S3 buckets, or APIs deployed on top of Amazon API Gateway, among others.



AWS Cloud Map

## Features:

- It decreases time consumption as it restricts the user to manage all the resource names and their locations manually within the application code.
- It is strongly integrated with Amazon Elastic Container Service (Amazon ECS).
- It constantly checks the health of the resources and allows users to choose whether to use Amazon Route 53 health checks or a third-party health checker.

- It provides a registry for the application services defined by namespaces and restricts developers to store, track, and update resource name and location information within the application code.

**Price details:**

- Extra charges related to Amazon Route 53 DNS and health check usage.
- Service registry charge - \$0.10 per registered resource per month.
- Lookup requests charge - \$1.00 per million discovery API calls.

# Amazon CloudFront

## What is Amazon CloudFront?

Amazon CloudFront is a content delivery network (CDN) service that securely delivers any kind of data to customers worldwide with low latency, low network and high transfer speeds.

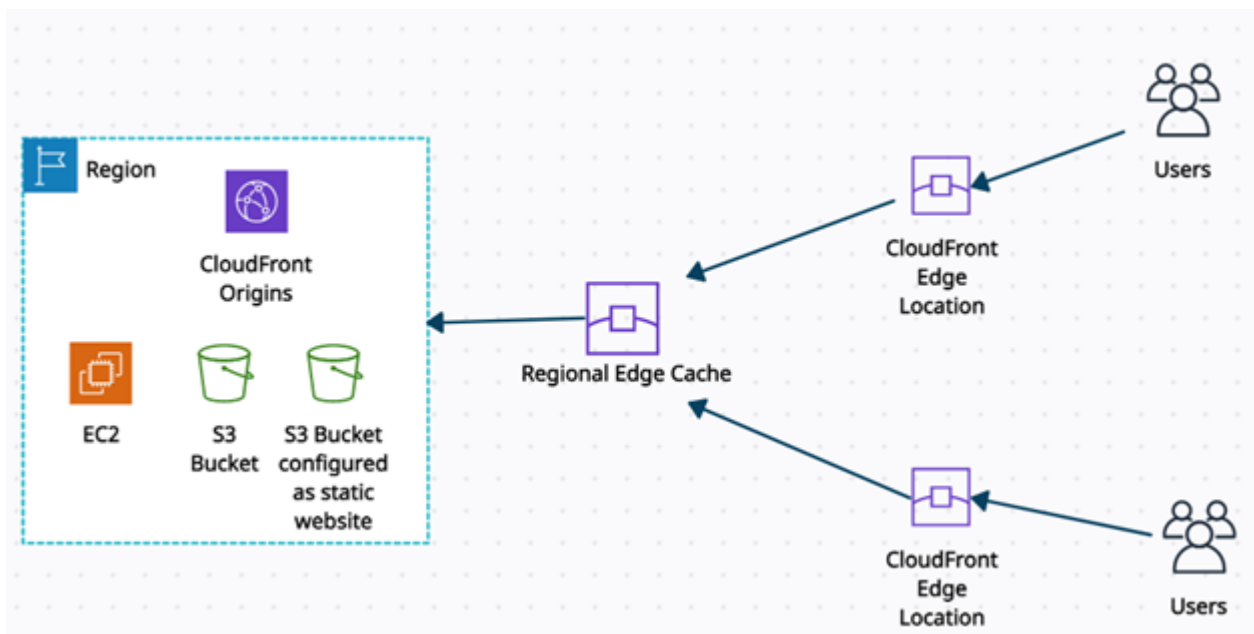
It uses edge locations (a network of small data centers) to cache copies of the data for the lowest latency. If the data is not present at edge locations, the request is sent to the source server, and data gets transferred from there.

It is integrated with AWS services such as

- Amazon S3,
- Amazon EC2,
- Elastic Load Balancing,
- Amazon Route 53,
- AWS Elemental Media Services.

The AWS origins from where CloudFront gets its traffic or requests are:

- Amazon S3
- Amazon EC2
- Elastic Load Balancing
- Customized HTTP origin



*CloudFront Overview*

It provides the programmable and secure edge CDN computing feature through AWS Lambda@Edge.

- It provides operations such as dynamic origin load-balancing, custom bot-management computationally, or building serverless origins.
- It has a built-in security feature to protect data from side-channel attacks such as Spectre and Meltdown.
- Field-level encryption with HTTPS - data remains encrypted throughout starting from the upload of sensitive data.

- AWS Shield Standard - against DDoS attacks.
- AWS Shield Standard + AWS WAF + Amazon Route53 - against more complex attacks than DDoS.

### **Amazon CloudFront Access Controls:**

Signed URLs:

- Use this to restrict access to individual files.

Signed Cookies:

- Use this to provide access to multiple restricted files.
- Use this if the user does not want to change current URLs.

Geo Restriction:

- Use this to restrict access to the data based on the geographic location of the website viewers.

Origin Access Identity (OAI):

- Outside access is restricted using signed URLs and signed cookies, but what if someone tries to access objects using Amazon S3 URL, bypassing CloudFront signed URL and signed cookies. To restrict that, OAI is used.
- Use OAI as a special CloudFront user, and associate it with your Cloudfront distribution to secure Amazon S3 content.

### **Pricing Details:**

- You pay for:
  - Data Transfer Out to Internet / Origin
  - A number of HTTP/HTTPS Requests.
  - Each custom SSL certificate associated with CloudFront distributions
  - Field-level encryption requests.
  - Execution of Lambda@Edge
- You do not pay for:
  - Data transfer between AWS regions and CloudFront.
  - AWS ACM SSL/TLS certificates and Shared CloudFront certificates.

# AWS PrivateLink

## What is AWS PrivateLink?

AWS PrivateLink is a network service used to connect to AWS services hosted by other AWS accounts (referred to as endpoint services) or AWS Marketplace. Whenever an interface VPC endpoint (interface endpoint) is created for service in the VPC, an Elastic Network Interface (ENI) in the required subnet with a private IP address is also created that serves as an entry point for traffic destined to the service.

### Interface endpoints

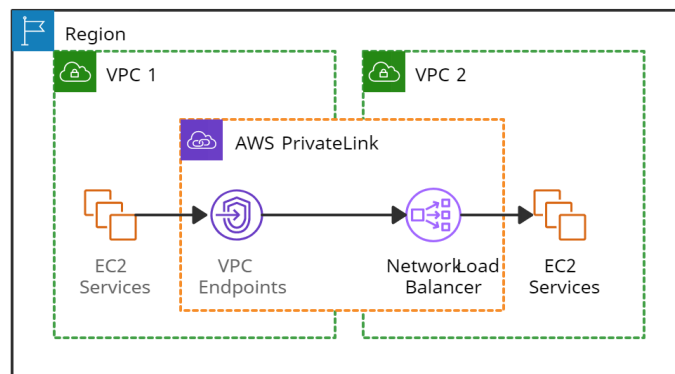
- It serves as an entry point for traffic destined to an AWS service or a VPC endpoint service.

### Gateway endpoints

- It is a gateway in the route-table that routes traffic only to Amazon S3 and DynamoDB.

PrivateLink is used for scenarios where the source VPC acts as a service provider, and the destination VPC acts as a service consumer. So service consumers use an interface endpoint to access the services running in the service provider.

But Direct Connect is something different. It only creates a connection between an interface endpoint and your on-premises data center. It can be used with AWS PrivateLink.



*AWS PrivateLink*

## Features:

- It is integrated with AWS Marketplace services so that the services can be directly attached to the endpoint.
- It provides security by not allowing the public internet and reducing the exposure to threats, such as brute force and DDoS attacks.
- It helps to connect services across different accounts and Amazon VPCs without any firewall rules, VPC peering connection, or managing VPC Classless Inter-Domain Routing (CIDRs).
- It helps to migrate on-premise applications to the AWS cloud more securely. Services can be securely accessible from the cloud and on-premises via AWS Direct Connect and AWS VPN.

## Pricing details:

- PrivateLink is charged based on the use of endpoints.

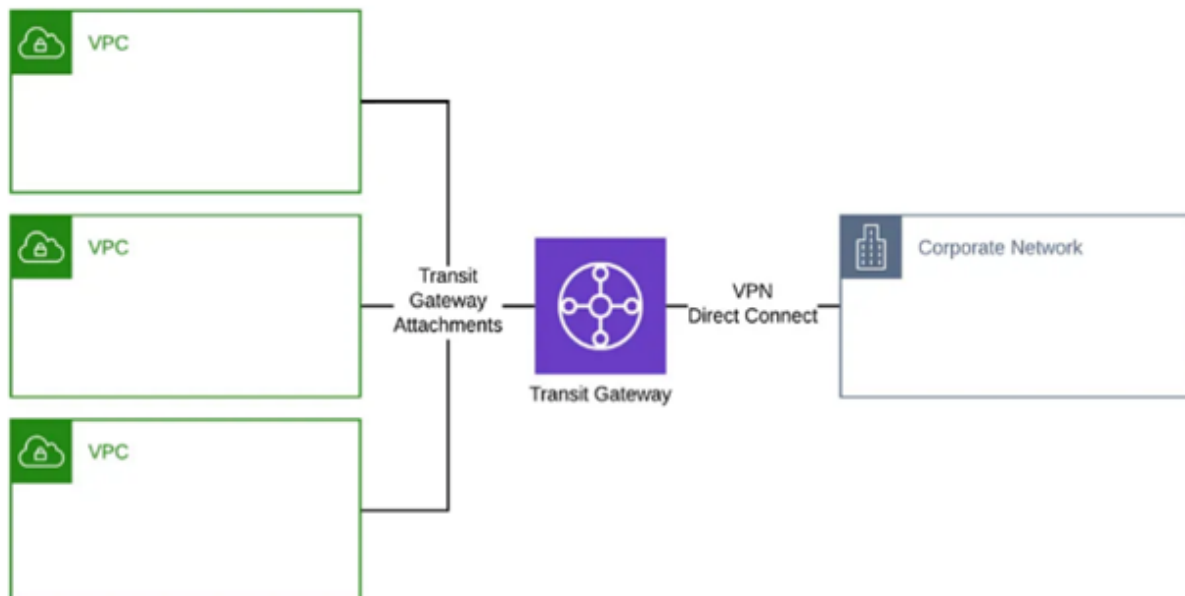


# AWS Transit Gateway

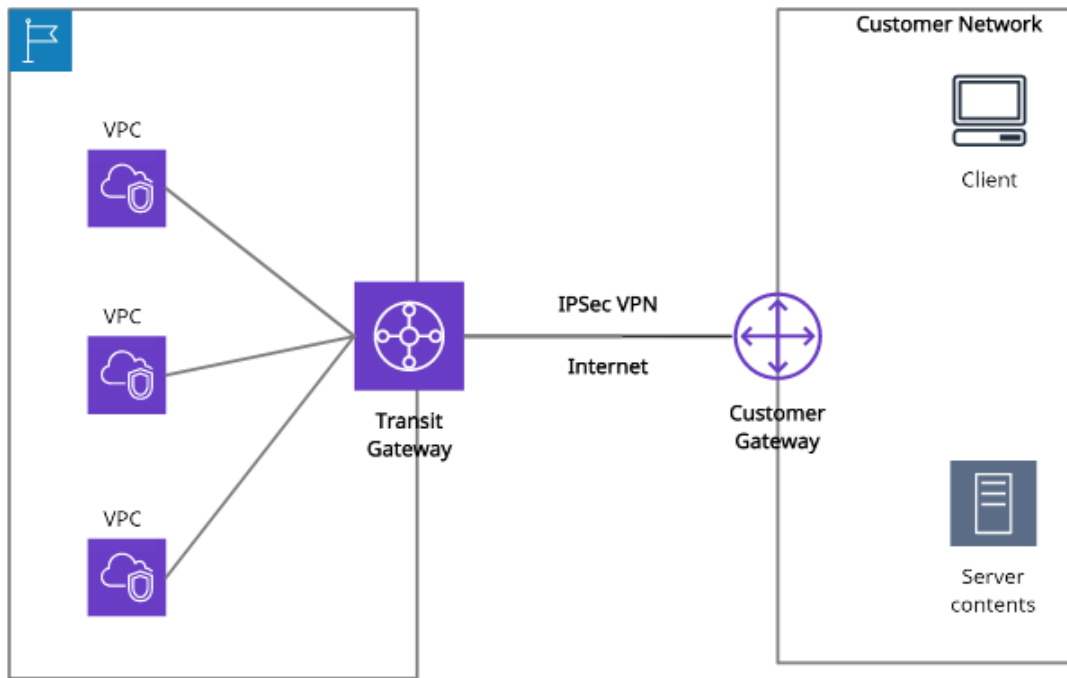
## What is AWS Transit Gateway?

AWS Transit Gateway is a network hub used to interconnect multiple VPCs. It can be used to attach all hybrid connectivity by controlling your organization's entire AWS routing configuration in one place.

- It can be more than one per region but can not be peered within a single region.
- It helps to solve the problem of complex VPC peering connections.
- It can be connected with an AWS Direct Connect gateway from a different AWS account.
- Resource Access Manager (RAM) cannot integrate AWS Transit Gateway with Direct Connect gateway.
- To implement redundancy, Transit Gateway also allows multi-user gateway connections.
- Transit Gateway VPN attachment is a feature to create an IPsec VPN connection between your remote network and the Transit Gateway.
- Transit Gateway Network Manager is used to manage and monitor networking resources and connections to remote branch locations.
- It reduces the complexity of maintaining VPN connections with hundreds of VPCs, which become very useful for large enterprises.
- It supports attaching Amazon VPCs with IPv6 CIDRs.



*AWS Transit Gateway*



*AWS Transit Gateway + VPN*

#### Transit Gateway vs. VPC peering:

Transit Gateway	VPC peering
<p>It has an hourly charge per attachment in addition to the data transfer fees.</p> <p>Multicast traffic can be routed between VPC attachments to a Transit Gateway.</p> <p>It provides Maximum bandwidth (burst) of 50 Gbps per Availability Zone per VPC connection.</p> <p>Security groups feature does not currently work with Transit Gateway.</p>	<p>It does not charge for data transfer.</p> <p>Multicast traffic cannot be routed to peering connections.</p> <p>It provides no aggregate bandwidth.</p> <p>Security groups feature works with intra-Region VPC peering.</p>

Transit Gateway can be created using the following ways

- AWS CLI
- AWS Management Console
- AWS CloudFormation

#### Price details:

- Users will be charged for your AWS Transit Gateway on an hourly basis.

# AWS Direct Connect

## What is AWS Direct Connect?

AWS Direct Connect is a cloud service that helps to establish a dedicated connection from an on-premises network to one or more VPCs in the same region.

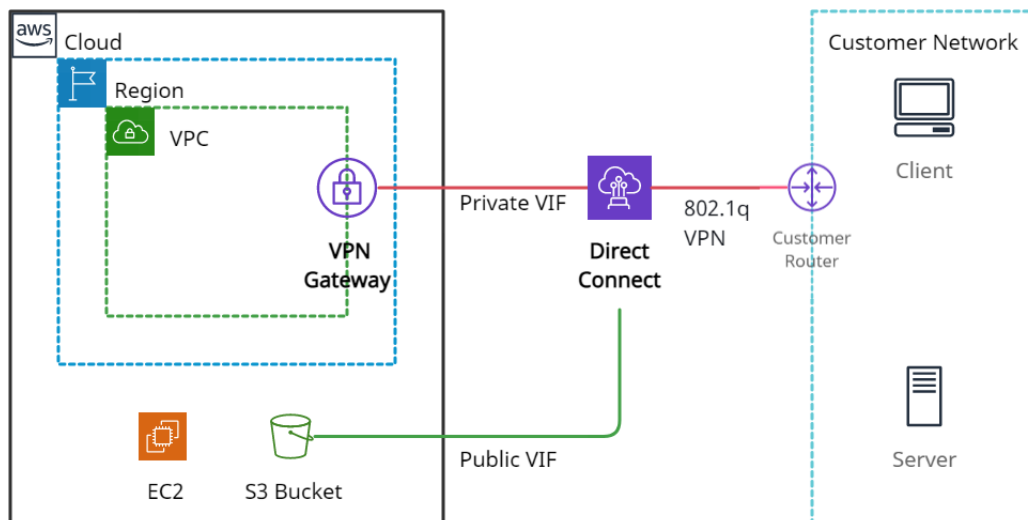
Private VIF with AWS Direct Connect helps to transfer business-critical data from the data-center, office or colocation environment into AWS, bypassing your Internet service provider and removing network traffic.

Private virtual interface: It helps to connect an Amazon VPC using private IP addresses.

Public virtual interface: It helps to connect AWS services located in any AWS region (except China) from your on-premises data center using public IP addresses.

## Methods of connecting to a VPC:

- AWS Managed VPN.
- AWS Direct Connect.
- AWS Direct Connect plus a VPN.
- AWS VPN CloudHub.
- Transit VPC.
- VPC Peering.
- AWS PrivateLink.
- VPC Endpoints.

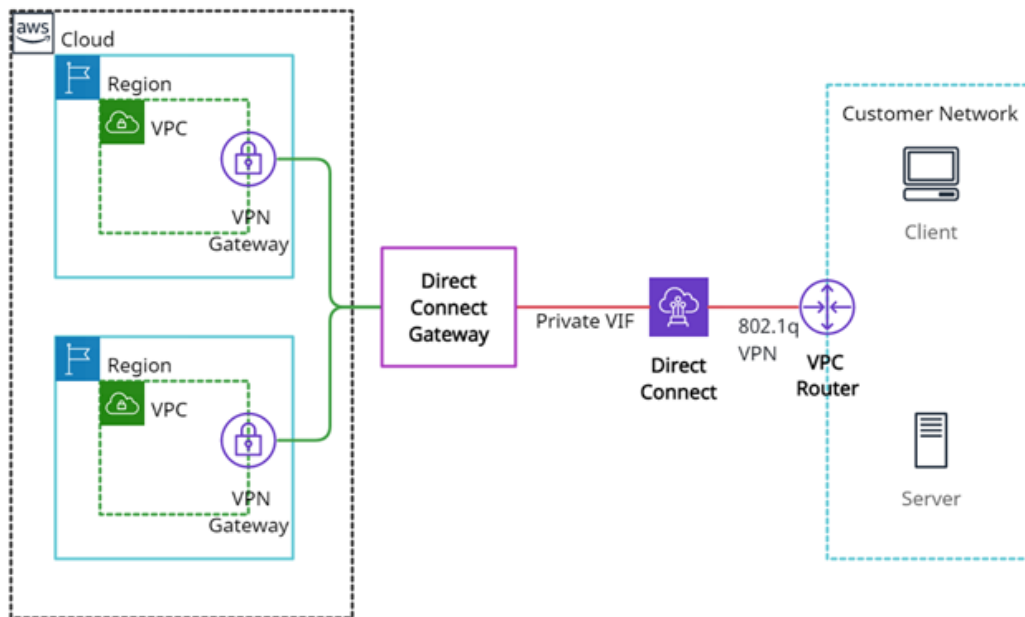


Amazon Direct Connect

## Direct Connect gateway:

It is a globally available service used to connect multiple Amazon VPCs across different regions or AWS accounts. It can be integrated with either of the following gateways:

- Transit gateway - it is a network hub used to connect multiple VPCs to an on-premise network in the same region.
- Virtual private gateway - It is a distributed edge routing function on the edges of VPC.



*Amazon Direct Connect gateway*

### Features:

- AWS Management Console helps to configure AWS Direct Connect service quickly and easily.
- It helps to choose the dedicated connection providing a more consistent network experience over Internet-based connections.
- It works with all AWS services that are accessible over the Internet.
- It helps to scale by using 1Gbps and 10 Gbps connections based on the capacity needed.

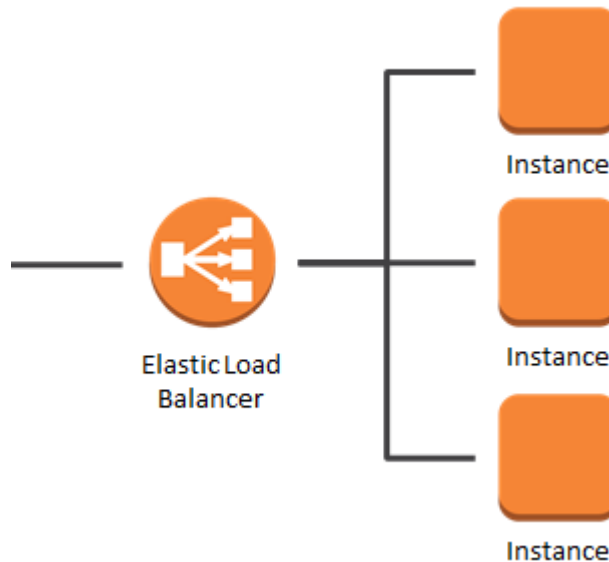
### Price details:

- Pay only for what you use. There is no minimum fee.
- Charges for Dedicated Connection port hours are consistent across all AWS Direct Connect locations globally except Japan.
- Data Transfer OUT charges are dependent on the source AWS Region.

# AWS Elastic Load Balancer

## What is AWS Elastic Load Balancer?

- ELB Stands for Elastic Load Balancer.
- It distributes the incoming traffic to multiple targets such as Instances, Containers, Lambda Functions, IP Addresses etc.
- It spans in single or multiple availability zones.
- It provides high availability, scaling and security for the application.



## Types of Elastic Load Balancer

### Application Load Balancer

- o It is best suited for load balancing of the web applications and websites.
- o It routes traffic to targets within Amazon VPC based on the content of the request.

### Network Load Balancer

- o It is mostly for the application which has ultra-high performance.
- o This load balancer also acts as a single point of contact for the clients.
- o This Load Balancer distributes the incoming traffic to the multiple targets.
- o The listener checks the connection request from the clients using the protocol and ports we specify.
- o It supports TCP, UDP and TLS protocol.

### Gateway Load Balancer (Newly Introduced)

- It is like other load balancers but it is for third-party appliances.
- This provides load balancing and auto scaling for the fleet of third-party appliances.
- It is used for security, network analytics and similar use cases.

### Classic Load Balancer

- It operates at request and connection level.
- It is for the EC2 Instance build in the old Classic Network.
- It is an old generation Load Balancer.
- AWS recommends to use Application or Network Load Balancer instead.

## Listeners

- A listener is a process that checks for connection requests, using the protocol and port that you configured.
- You can add HTTP, HTTPS or both.

## Target Group

- It is the destination of the ELB.
- Different target groups can be created for different types of requests.
- For example, one target group i.e., a fleet of instances will be handling the general request and other target groups will handle the other type of request such as micro services.
- Currently, three types of target supported by ELB: Instance, IP and Lambda Functions.

## Health Check

- Health checks will be checking the health of Targets regularly and if any target is unhealthy then traffic will not be sent to that Target.
- We can define the number of consecutive health checks failure then only the Load Balancer will not send the traffic to those Targets.
- e.g., If 4 EC2 are registered as Target behind Application Load Balancer and if one of the EC2 Instance is not healthy then Load Balancer will not send the traffic to that EC2 Instance

## Use Cases:

- **Web Application Deployed in Multiple Servers:** If a web Application/Website is deployed in multiple EC2 Instances then we can distribute the traffic between the Application Load Balancers.
- **Building a Hybrid Cloud:** Elastic Load Balancing offers the ability to load balance across AWS and on-premises resources, using a single load balancer. You can achieve this by registering all of your resources to the same target group and associating the target group with a load balancer.
- **Migrating to AWS:** ELB supports the load balancing capabilities critical for you to migrate to AWS. ELB is well positioned to load balance both traditional as well as cloud native applications with auto scaling capabilities that eliminate the guess work in capacity planning.

## Charges:

- Charges will be based on each hour or partial hour that the ELB is running.
- Charges will also depend on the LCU (Load Balancer Units)

## Amazon Route 53

### What is Amazon Route 53?

Route53 is a managed DNS (Domain Name System) service where DNS is a collection of rules and records intended to help clients/users understand how to reach any server by its domain name.

**Route 53 hosted zone** is a collection of records for a specified domain that can be managed together. There are two types of zones:

- Public host zone – It determines how traffic is routed on the Internet.
- Private hosted zone – It determines how traffic is routed within VPC.

### Route 53 TTL (seconds):

- It is the amount of time for which a DNS resolver creates a cache information about the records and reduces the query latency.
- Default TTL does not exist for any record type but always specifies a TTL of 60 seconds or less so that clients/users can respond quickly to changes in health status.

### Route53 CNAME vs. Alias

CNAME	Alias
It points a hostname to any other hostname. (app.mything.com -> abc.anything.com) It works only for the non-root domains. (abcxyz.maindomain.com) Route 53 charges for CNAME queries. It points to any DNS record that is hosted anywhere.	It points a hostname to an AWS Resource. (app.mything.com -> abc.amazonaws.com) It works for the root domain and non-root domain. (maindomain.com) Route 53 doesn't charge for Alias queries. It points to an ELB, CloudFront distribution, Elastic Beanstalk environment, S3 bucket as a static website, or another record in the same hosted zone.

The most common records supported in Route 53 are:

- A: hostname to IPv4
- AAAA: hostname to IPv6
- CNAME: hostname to hostname
- Alias: hostname to AWS resource.

Other supported records are:

- CAA (certification authority authorization)
- MX (mail exchange record)
- NAPTR (name authority pointer record)
- NS (name server record)
- PTR (pointer record)
- SOA (start of authority record)
- SPF (sender policy framework)
- SRV (service locator)
- TXT (text record)

## **Route 53 Routing Policies:**

### **Simple:**

- It is used when there is a need to redirect traffic to a single resource.
- It does not support health checks.

### **Weighted:**

- It is similar to simple, but you can specify a weight associated with resources.
- It supports health checks.

### **Failover:**

- If the primary resource is down (based on health checks), it will route to a secondary destination.
- It supports health checks.

### **Geo-location:**

- It routes traffic to the closest geographic location you are in.

### **Geo-proximity:**

- It routes traffic based on the location of resources to the closest region within a geographic area.

### **Latency based:**

- It routes traffic to the destination that has the least latency.

### **Multi-value answer:**

- It distributes DNS responses across multiple IP addresses.
- If a web server becomes unavailable after a resolver cache a response, a user can try up to eight other IP addresses from the response to reduce downtime.

## **Use cases:**

- When users try to register a domain with Route 53, it becomes the trustworthy DNS server for that domain and creates a public hosted zone.
- Users can have their domain registered in one AWS account and the hosted zone in another AWS account.
- For private hosted zones, the following VPC settings must be 'true':
  - enableDnsHostname.
  - enableDnsSupport.
- Health checks can be pointed at:
  - Endpoints (can be IP addresses or domain names.)
  - Status of other health checks.
  - Status of a CloudWatch alarm.
- Route53 as a Registrar: A domain name registrar is an organization that manages the reservation of Internet domain names.
- Domain Registrar != DNS

## **Price details:**

- There are no contracts or any down payments for using Amazon Route 53.
- Route 53 charges annually for each domain name registered via Route 53.
- Different rates are applied for Standard Queries, Latency Based Routing Queries, Geo DNS and Geo Proximity Queries.



# AWS VPC

## What is AWS VPC?

Amazon Virtual Private Cloud (VPC) is a service that allows users to create a virtual dedicated network for resources.

## Security Groups:

### Default Security Groups:-

Inbound rule - Allows all inbound traffic

Outbound rule - Allows all outbound traffic

### Custom Security Groups:- (by default)

Inbound rule - Allows no inbound traffic

Outbound rule - Allows all outbound traffic

## Network ACLs (access control list):

### Default Network ACL:-

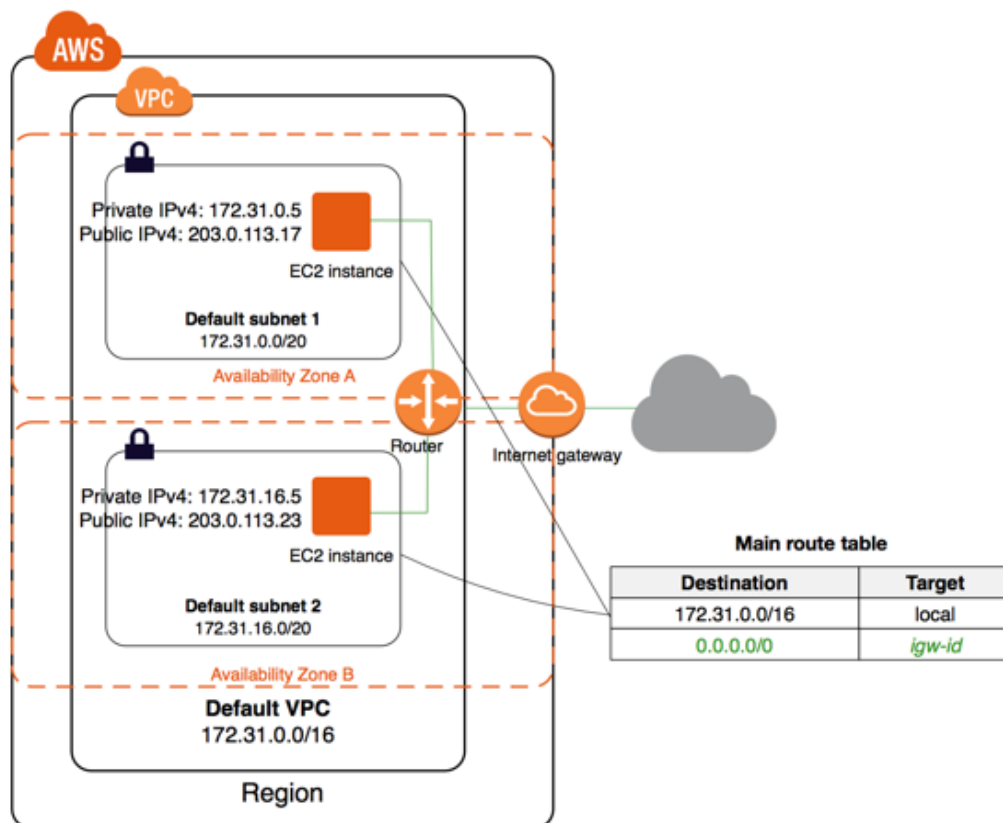
Inbound rule - Allows all inbound traffic

Outbound rule - Allows all outbound traffic

### Custom Network ACL:- (by default)

Inbound rule - Denies all inbound traffic

Outbound rule - Denies all outbound traffic



VPC Architecture

## Components of VPC:

### Subnets

- The subnet is a core component of the VPC.
- Resources will reside inside the Subnet only.
- Subnets are the logical division of the IP Address.
- One Subnet should not overlap another subnet.
- A subnet can be private or public.
- Resources in **Public Subnet** will have internet access.
- Resources in the **Private Subnet** will not have internet access.
- If private subnet resources want internet accessibility then we will need a NAT gateway or NAT instance in a public subnet.

### Route Tables

- Route tables will decide where the network traffic will be directed.
- One Subnet can connect to one route table at a time.
- But one Route table can connect to multiple subnets.
- If the route table is connected to the Internet Gateway and that route table is associated with the subnet, then that subnet will be considered as a Public Subnet.
- The private subnet is not associated with the route table which is connected to the Internet gateway.

### NAT Devices

- NAT stands for Network Address Translation.
- It allows resources in the Private subnet to connect to the internet if required.

#### NAT Instance

- It is an EC2 Instance.
- It will be deployed in the Public Subnet.
- NAT Instance allows you to initiate IPv4 Outbound traffic to the internet.
- It will not allow the instance to receive inbound traffic from the internet.

#### NAT Gateway

- Nat Gateway is Managed by AWS.
- NAT will be using the elastic IP address.
- You will be charged for NAT gateway on a per hour basis and data processing rates.
- NAT is not for IPv6 traffic.
- NAT gateway allows you to initiate IPv4 Outbound traffic to the internet.
- It will not allow the instance to receive inbound traffic from the internet.

### DHCP Options Set:

- DHCP stands for Dynamic Host Configuration Protocol.
- It is the standard for passing the various configuration information to hosts over the TCP/IP Network.
- DHCP contains information such as domain name, domain name server.
- All this information will be contained in Configuration parameters.
- DHCP will be created automatically while creating VPC.

## **PrivateLink**

- PrivateLink is a technology that will allow you to access services privately without internet connectivity and it will use the private IP Addresses.

## **Endpoints**

- It allows you to create connections between your VPC and supported AWS services.
- The endpoints are powered by PrivateLink.
- The traffic will not leave the AWS network.
- It means endpoints will not require Internet Gateway, Virtual Private Gateway, NAT components.
- The public IP address is not required for communication.
- Communication will be established between the VPC and other services with high availability.

## **Types of Endpoints**

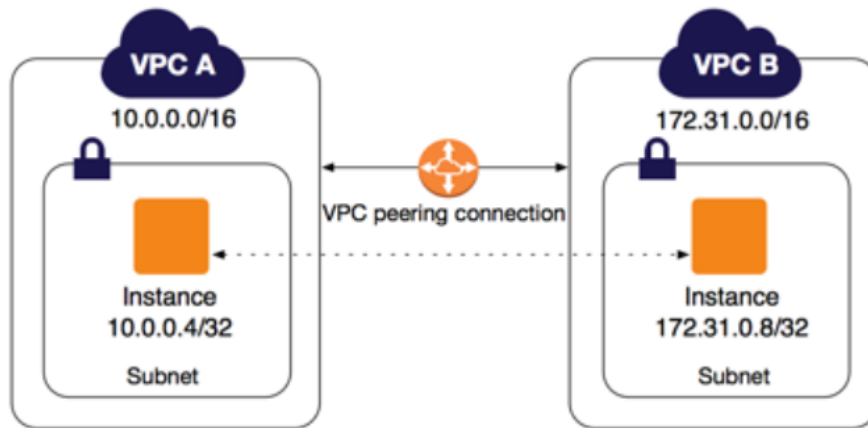
- **Interface Endpoints**
  - It is an entry point for traffic interception.
  - It will route the traffic to the service that you configure.
  - It will use an ENI with a private IP address.
  - For Example: it will allow instances to connect to Amazon Kinesis through interface endpoint.
- **Gateway Load balancer Endpoints**
  - It is an entry point for traffic interception.
  - It will route the traffic to the service that you configure.
  - It will use load balancers to route the traffic.
  - For Example Security Inspection.
- **Gateway Endpoints**
  - It is a gateway that you defined in Route Table as a Target.
  - And the destination will be the supported AWS Services.
  - Amazon S3, DynamoDB supports Gateway Endpoint.

## **Egress Only Internet Gateway**

- An egress-only internet gateway is designed only for IPv6 address communications.
- It is a highly available, horizontally scaled component which will allow outbound only rule for IPv6 traffic.
- It will not allow inbound connection to your EC2 Instances.

## **VPC Peering:**

- VPC peering establishes a connection between two VPCs.
- EC2 Instances in both the VPC can communicate with each other as if they are in the same network.
- Peering connections can be established between VPCs in the same region, VPCs in a different region or VPCs in another AWS Account as well.



## VPN

- Virtual Private Network (VPN) establish secure connections between multiple networks i.e., on-premise network, client space, AWS Cloud, and all the network acts
- VPN provides a high-available, elastic, and managed solution to protect your network traffic.

### AWS Site-to-Site VPN

- o AWS Site-to-Site VPN creates encrypted tunnels between your network and your Amazon Virtual Private Clouds or AWS Transit Gateways.

### AWS Client VPN

- o AWS Client VPN connects your users to AWS or on-premises resources using a VPN software client.

## Use Cases:

- Host a simple public-facing website.
- Host multi-tier web applications.
- Used for disaster recovery as well.

## Pricing:

- No additional charges for creating a custom VPC.
- NAT does not come under the free tier limit you will get charged per hour basis.
- NAT Gateway data processing charge and data transfer charges will be separate.
- You will get charged per hour basis for traffic mirroring.

# AWS AppSync

## What is AWS AppSync?

AWS AppSync simplifies the process of developing an application by providing us to create flexible, secure, extensible, and real-time APIs. It can be called “**The Facilitator**” because it connects the client applications (mobile apps, web apps, IOT services, etc.) to AWS services (DynamoDB, AWS Aurora, etc.).

### AppSync = “The Facilitator”

Within AWS AppSync, there are GraphQL schema and Resolvers that help secure access and combine data from databases, API, and other backend systems.

GraphQL Schema: This is the unique structure that AWS AppSync uses to layout the data format before putting it into a database schema.

Resolvers: This resolves the data types, which the user creates in the Graph schema to put and receive from the data source.

### AppSync Benefits:

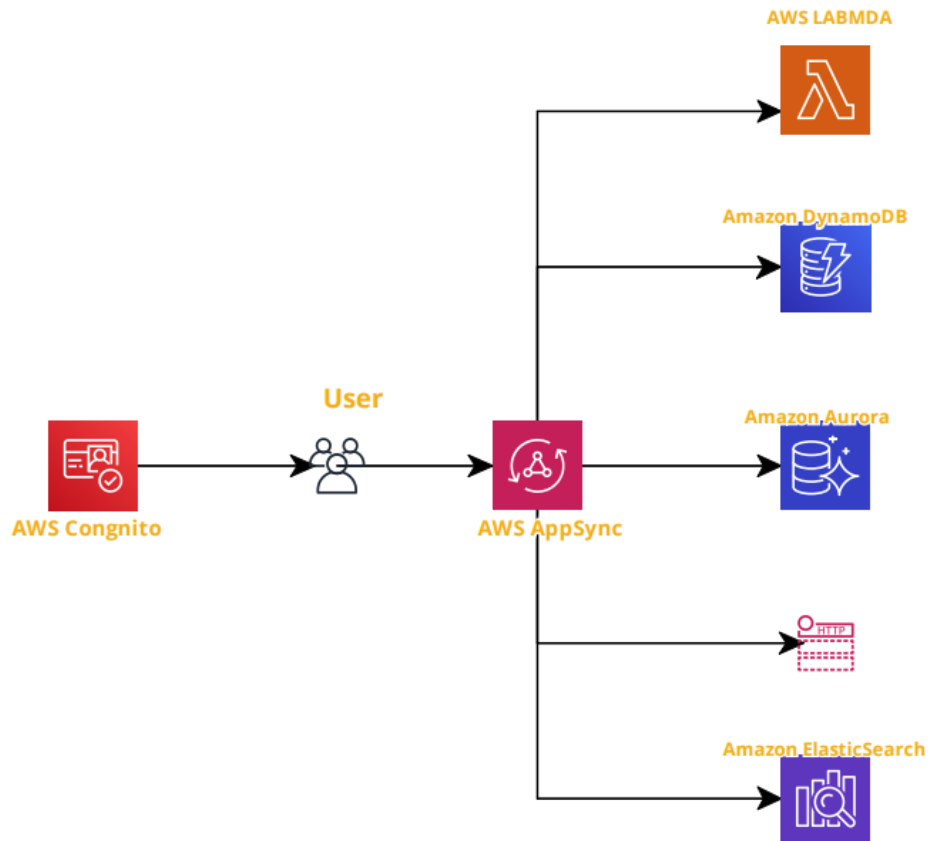
- Fast setup – great scalability as needed.
- Real-time subscriptions and offline access.
- Unified secured access.
- Provision of caching capabilities for performance improvements.
- Bandwidth optimization.
- Conflict resolution in the cloud.

### Use Cases:

There are many use cases where AWS AppSync can play a vital role. Some of them are below:

- Banking Alerts.
- Chat Applications.
- Financial transactions.
- Shared Whiteboards.
- Document Collaboration
- Multiplayer games.

The below diagram shows how AWS AppSync helps to build a real-time blog application. AWS Cognito provides secure authentication. The user can create all the CRUD operations, posts, comments, etc., using the facilitator (AWS AppSync) and store/receive to Amazon backend servers such as DynamoDB, AWS Aurora, AWS Lambda, etc.



### Pricing:

- Query and Data Modification Operations
  - \$4.00 per million operations.
- Real-time Updates
  - \$2.00 per million updates.
  - \$0.08 per million minutes of connection to the AWS AppSync service

# Amazon EventBridge

## What is Amazon EventBridge?

A serverless event bus service for Software-as-a-Service (SAAS) and AWS services.

In simple words, Amazon EventBridge provides an easy solution to integrate SAAS, custom-build applications with more than 17+ AWS services with the delivery of real-time data from different event sources. Users can easily set up the routing rules to determine the target web-service, and multiple target locations (such as AWS Lambda or AWS SNS) can be selected at once.

It is a fully managed service that takes care of event ingestion, delivery, security, authorization, error handling, and required infrastructure management tasks to set up and run a highly scalable serverless event bus. EventBridge was formerly called Amazon CloudWatch Events, and it uses the same CloudWatch Event API.

## Key Concepts:-

### Event Buses:

An event bus receives events. When a user creates a rule, which will be associated with a specific event bus, the rule matches only to the event received by the event bus. Each user's account has one default event bus, which receives events from AWS services. We can also create our custom event buses.

### Events:

An event indicates a change in the environment. By creating rules, you can have AWS services that act automatically when changes occur in other AWS services, in SaaS applications, or user's custom applications.

### Schema Registry:

A Schema Registry is a container for schemas. Schemas are available for the events for all AWS services on Amazon EventBridge. Users can always create or update their schemas or automatically infer schemas from events running on event buses. Each schema will have multiple versions. Users can use the latest schema or select earlier **versions**.

### Rules:

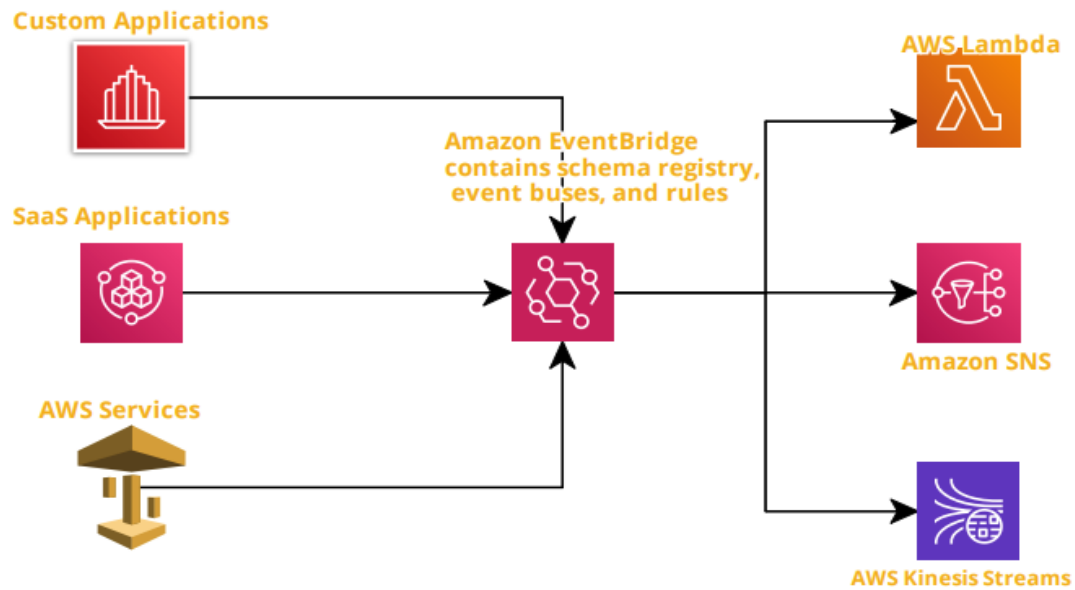
A rule matches incoming events and routes them to targets for processing. A single rule can route an event (JSON format) to multiple targets. All pointed targets will be processed in parallel and no particular order.

### Targets:

A target processes events and receives events in JSON format. A rule's target must be in the same region as a rule.

## Features:

- Fully managed, pay-as-you-go.
- Native integration with SaaS providers.
- 90+ AWS services as sources.
- 17 AWS services as targets.
- \$1 per million events put into the bus.
- No additional cost for delivery.
- Multiple target locations for delivery.
- Easy to scale and manage.



*Structure of the Amazon EventBridge service.*

As shown above, this service receives input from different sources (such as custom apps, SaaS applications, and AWS services). Amazon EventBridge contains an event source for a SaaS application responsible for authentication and security of the source. EventBridge has a schema registry, event buses (default, custom, and partner), and rules for the target services.

### Pricing:

- There are no additional charges for rules or event delivery.
- The users only pay for events published to your event bus, events ingested for Schema Discovery, and Event Replay.
  - *Custom events*: Charge \$1.00 per million requests.
  - *Third-party events* (SaaS): Charge \$1.00 per million requests.
  - *Cross-account events*: \$1.00 per million.



# AWS SNS (Simple Notification Service)

## What is AWS SNS?

Amazon Simple Notification Service (Amazon SNS) is a web service that makes it easy to set up, operate, and send notifications from the cloud.

It provides developers with a highly scalable, flexible, and cost-effective approach to publish messages from an application and deliver them to subscribers or other applications. It provides push notifications directly to mobile devices and delivers notifications by SMS text messages, email to Amazon Simple Queue Service (SQS), or any HTTP client.

It allows developers to group multiple recipients using topics.

It consists of **topics** and **subscribers**.

A topic is an access point for allowing recipients to get identical copies for the same notification. One topic can support deliveries to multiple end-points – for example - we can group together to android, IOS, and SMS text messages.

Two types of topics can be defined in the AWS SNS service.

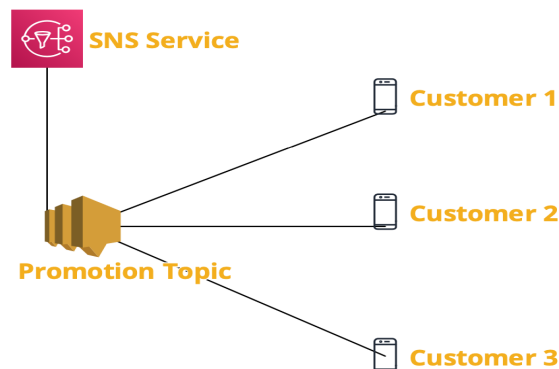
1. Standard topic is used when incoming messages are not in order. In other words, messages can be delivered as they are received.
2. FIFO topic is designed to maintain order of the messages between the applications, especially when the events are critical. Duplication will be avoided in this case.

## Features:

- Instantaneous, push-based delivery.
- Simple API and easy integration with AWS services.
- Flexible message delivery over multiple message protocols.
- Cost-effective – as pay as pay-as-you-go model.
- Fully managed and durable with automatic scalability.

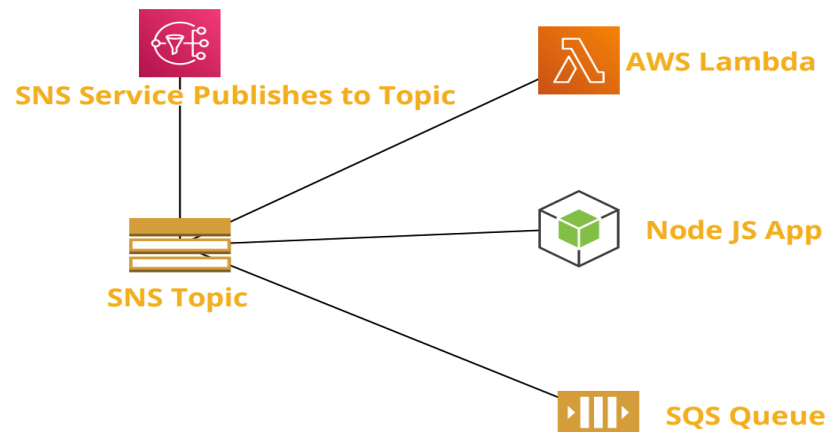
## Use cases:

- SNS application to person: below use cases show SNS service publishes messages to topic, sending messages to each customer's cell phone. This is an example of an AWS application to personal service.



- SNS Application to Application: In this type of service, where SNS topic would interact with different AWS services such as AWS Lambda, Node JS app, and SQS services. For example, AWS S3 service has only configuration

with AWS SNS service, which will be responsible for sending identical messages to other AWS services.



### Pricing:

- **Standard Topics:** First 1 million Amazon SNS requests per month are free. There will be a cost associated with \$0.50 per 1 million requests.
- **FIFO Topics:** Amazon SNS FIFO topic pricing is based on the number of published messages, the number of subscribed messages, and their respective amount of payload data.

# Amazon Simple Queue Service (SQS)

## What is Amazon Simple Queue Service (SQS)?

Amazon Simple Queue Service (SQS) is a serverless service used to decouple (loose couple) serverless applications and components.

The queue represents a temporary repository between the producer and consumer of messages.

It can scale up to 1-10000 messages per second.

The default retention period of messages is four days and can be extended to fourteen days.

SQS messages get automatically deleted after being consumed by the consumers.

SQS messages have a fixed size of 256KB.

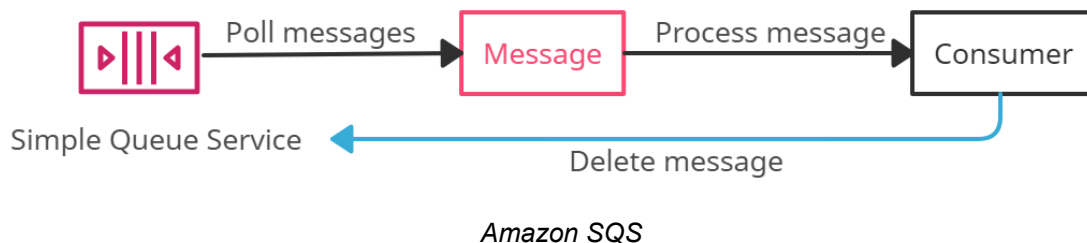
There are two SQS Queue types:

### Standard Queue -

- The unlimited number of transactions per second.
- Messages get delivered in any order.
- Messages can be sent twice or multiple times.

### FIFO Queue -

- 300 messages per second.
- Support batches of 10 messages per operation, results in 3000 messages per second.
- Messages get consumed only once.



**Delay Queue** is a queue that allows users to postpone/delay the delivery of messages to a queue for a specific number of seconds.

Messages can be delayed for 0 seconds (default) -15 (maximum) minutes.

**Dead-Letter Queue** is a queue for those messages that are not consumed successfully. It is used to handle message failure.

Visibility Timeout is the amount of time during which SQS prevents other consumers from receiving (poll) and processing the messages.

- Default visibility timeout - 30 seconds
- Minimum visibility timeout - 0 seconds
- Maximum visibility timeout - 12 hours

# AWS Step Functions

## What are step functions?

Step functions allow developers to offload application orchestration into fully managed AWS services. This means you can just modularize your code to “Steps” and let AWS worry about handling partial failure cases, retries, or error handling scenarios.

## Types of step functions:

1. Standard workflow: Standard workflow can be used for long-running, durable, and auditable workflows.
2. Express Workflow: Express workflow is designed for high volume, and event processing workloads.

## Features:

- Allow to create workflow which follows a fixed or dynamic sequence.
- Inbuilt “Retry” and error handling functionality.
- Support integration with AWS native Lambda, SNS, ECS, AWS Fargate, etc.
- Support GUI audit workflow process, input/output, etc., well.
- GUI provides support to analyze the running process and detect the failures immediately.
- High availability, High scalability and low cost.
- Manages the states of the application during workflow execution.
- Step function is based on the concepts of tasks and state machines.
  - o Tasks can be defined by using an activity or an AWS Lambda function.
  - o State machines can express an algorithm that contains relations, input/output.

## Best Practices:

- Set time-outs in state machine definitions, which help in better task response when something goes wrong in getting a response from an activity.

Example:

```
"ActivityState": {  
  "Type": "Task",  
  "Resource":  
    "arn:aws:states:us-east-1:123456789012:activity:abc",  
  "TimeoutSeconds": 900,  
  "HeartbeatSeconds": 40,  
  "Next": "State2"  
}
```

- Always provide the Amazon S3 arn (amazon resource name) instead of large payloads to the state machine when passing input to Lambda function.

Example:

```
{  
  "Data": "arn:aws:s3:::MyBucket/data.json"  
}
```

- Handle errors in state machines while invoking AWS lambda functions.

Example:

```
"Retry": [ {  
  "ErrorEquals": [ "Lambda.CreditServiceException" ]  
}
```

```
"IntervalSeconds": 2,  
"MaxAttempts": 3,  
"BackoffRate": 2  
}]
```

- It has a hard quota of 25K entries during execution history. To avoid this for long-running executions, implement a pattern using the AWS lambda function.

It supports below AWS services:

- Lambda
- AWS Batch
- DynamoDB
- ECS/Fargate
- SNS
- SQS
- SageMaker
- EMR

#### **Pricing:**

- With Step Functions Express Workflows, you pay only for what you use. You are charged based on the number of requests for your workflow and its duration.
  - \$0.025 per 1,000 state transitions (For Standard workflows)
  - \$1.00 per 1M requests (For Express workflows)

## **Amazon Simple Workflow Service(SWF)**

### **What is SWF?**

Amazon simple workflow service (Amazon SWF) is a web service that provides generic solutions for distributed program workflows. The primary concepts of Amazon SWF are to implement scheduling, concurrency, and dependencies. Service also responsible to take care of message flow, locking, and state management-related work.

Amazon SWF provides simple API calls that can be executed from code written in any language and run on EC2 instances, or any of the machines located in any part of the work and can be accessed via the internet.

### **SWF Actors**

- Workflow starters: Any application that can trigger the workflow which could be your eCommerce website or a mobile app.
- Deciders: Control the flow of activity tasks in workflow execution. Based on component behavior, deciders confirm the next steps. It may also help in applying conditional and concurrent processes if required.
- Activity Workers: Carry out the activity tasks.

### **Features:**

- Logical separation of each component.
- Workflow retention for up to 12 months.
- Task orient API structure which can be invoked programmatically or manually.
- Amazon SWF ensures that a task is assigned only once and is never duplicated.
- Keep track of all the tasks and events in an application.
- Routing and Queuing of tasks.
- Workflows can have child workflows.
- Retry handling and auditing/logging.
- Fits naturally with immutable infrastructure.

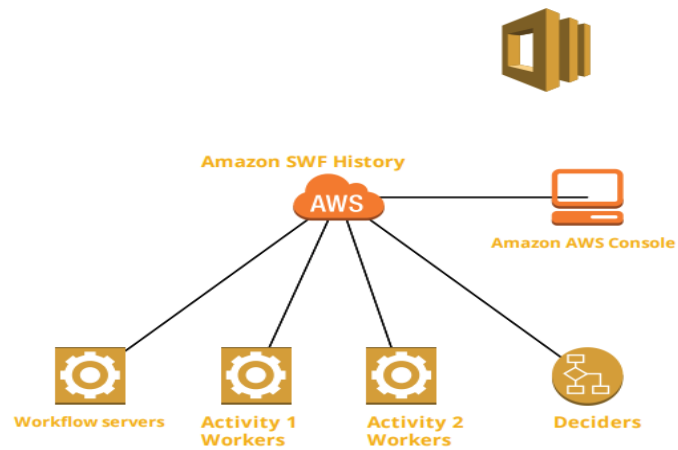
### **Use Cases:**

- Video Encoding/Media processing
- Data Center Migration
- Product catalogs with Human workers
- Data warehouse processing.

### **Data warehouse process:**

The below diagram shows a real-time example of data warehouse processing.

At the very first step, the AWS console works as starters/triggers of the workflow, then action will be picked by the deciders, and activity workers will perform their responsibilities once the decider performs the action.



### Pricing:

Workflow executions: \$0.0001 per workflow execution above the free tier.

Open and retained workflows:

- \$0.000005 per 24-hour period that a workflow is retained or open.

# AWS Cost Explorer

## What is AWS Cost Explorer?

AWS Cost Explorer is a UI-tool that enables users to analyze the costs and usage with the help of a graph, the Cost Explorer cost and usage reports, and/or the Cost Explorer RI report. It can be accessed from the Billing and Cost Management console.

It provides default reports for analysis with some filters and constraints to create the reports. Analysis using Cost Explorer can be saved as a bookmark, CSV file download, or save them as a report.

The default reports provided by Cost Explorer are:

- **Cost and usage reports:**

It provides the following data for understanding the costs:-

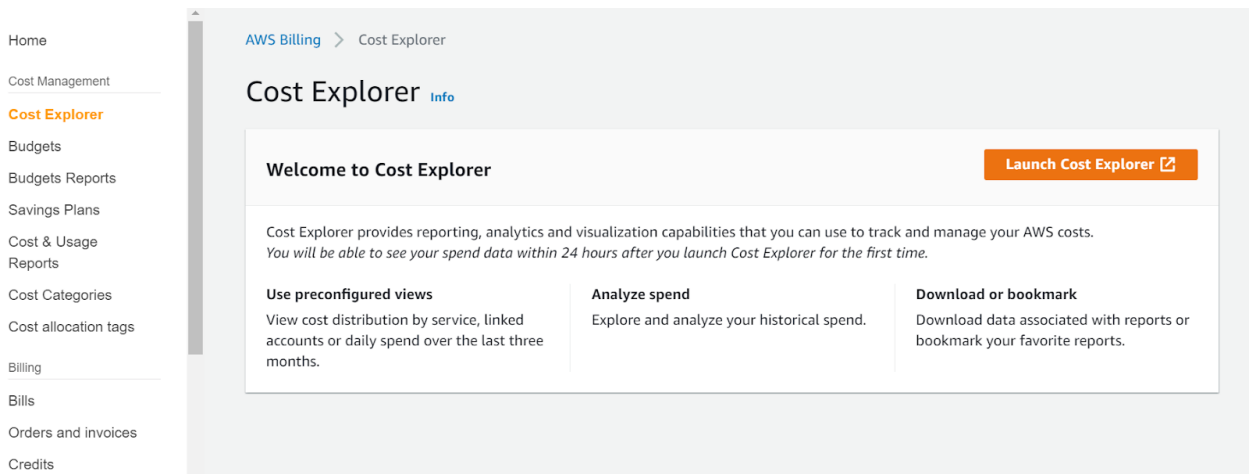
- AWS Marketplace
- Daily costs
- Monthly costs by linked account
- Monthly costs by service
- Monthly EC2 running hours costs and usage

- **Reserved Instance reports:**

It provides the following reports for understanding the reservations:-

- **RI utilization reports:** It gives information about how much costs are saved or overspent by using Reserved Instances (RIs).
- **RI coverage reports:** It gives information about how many hours are saved or overspent by using Reserved Instances (RIs).
- The first time that the user signs up for Cost Explorer, it directs through the main parts of the console. It prepares the data regarding costs & usage and displays up to 12 months of historical data (might be less if less used), current month data, and then calculates the forecast data for the next 12 months.
- It uses the same set of data that is used to generate the AWS Cost and Usage Reports and the billing reports.
- It provides a custom time period to view the data at a monthly or daily interval.
- It provides a feature of Savings Plans which provides savings of up to 72% on the AWS compute usage.
- It provides a way to access the data programmatically using the Cost Explorer API.





*AWS Cost Explorer*

### Price details:

- Analysis of costs and usage using the Cost Explorer can be viewed free of charge.
- The cost of using AWS Cost Explorer API is \$0.01 per API request.

# AWS Budgets

## What is AWS Budgets?

AWS Budgets enables the customer to set custom budgets to track cost and usage from the simplest to the complex use cases.

- AWS Budgets can be used to set reservation utilization or coverage targets allowing you to get alerts by email or SNS notification when the metrics reach the threshold.
- Reservation alerts feature is provided to Amazon EC2, Amazon RDS, Amazon Redshift, Amazon ElastiCache, and Elasticsearch.
- The Budgets can be filtered based on specific dimensions such as Service, Linked Account, Tags, Availability Zone, API Operation, and Purchase Option (i.e., “Reserved”) and be notified using SNS.
- AWS Budgets can be accessed from the AWS Management Console’s service links and within the AWS Billing Console. Budgets API or CLI (command-line interface) can also be used to create, edit, delete, and view up to 20,000 budgets per payer account.
- AWS Budgets can be integrated with other AWS services such as AWS Cost Explorer, AWS Chatbot, Amazon Chime room, and AWS Service Catalog.
- AWS Budgets can now be created monthly, quarterly, or annual budgets for the AWS resource usage or the AWS costs.

The following types of budgets can be created using AWS Budgets:

- Cost budgets
- Usage budgets
- RI utilization budgets
- RI coverage budgets
- Savings Plans utilization budgets
- Savings Plans coverage budgets

## Best Practices:

- Users can set up to five alerts for each budget. But the most important are:
  - Alerts when current monthly costs exceed the budgeted amount.
  - Alerts when current monthly costs exceed 80% of the budgeted amount.
  - Alerts when forecasted monthly costs exceed the budgeted amount.
- When creating budgets using Budgets API, a separate IAM user should be made for allowing access or IAM role for each user, if multiple users need access to Budgets API.
- If using consolidated billing in an organization is handled by a master account, IAM policies can control access to budgets by member accounts. Member account owners can create their budgets but cannot change or edit budgets of Master accounts.
- Two of the related managed policies are provided for budget actions. One policy allows a user to pass a role to the budgets service, and the other allows budgets to execute the action.

- Budget actions are not effective enough to control costs with Auto Scaling groups.

**Price details:**

- Monitoring the budgets and receiving notifications are free of charge.
- Each subsequent action-enabled budget will experience a \$0.10 daily cost after the free quota ends.

# AWS Cost & Usage Report

## What is AWS Cost & Usage Report?

AWS Cost & Usage Report (AWS CUR) allows users to access the detailed set of AWS cost and usage data available, including metadata about AWS resources, pricing, Reserved Instances, and Savings Plans.

## AWS Cost & Usage Report is a part of AWS Cost Explorer.

- AWS Cost and Usage Reports functions:
  - It sends report files to your Amazon S3 bucket.
  - It updates reports up to three times a day.
  - It creates, retrieves, and deletes reports using the AWS CUR API Reference.
- There is a feature of Data Dictionary that lists the columns added in the report to easily analyze cost and usage in detail.
- For viewing, reports can be downloaded from the Amazon S3 console, for analyzing the report Amazon Athena can be used, or upload the report into Amazon Redshift or Amazon QuickSight.
- Users with IAM permissions or IAM roles can access and view the reports.
- If a member account in an organization owns or creates a Cost and Usage Report, then it can have access only to billing data for the time it has been a member of the Organization.
- If the master account of an AWS Organization wants to block access to the member accounts to set-up a Cost and Usage Report, Service Control Policy (SCP) can be used.

## Reserved Instance Reporting

### What is Reserved Instance Reporting?

Reserved Instance Reporting or Reserved Instance Utilization and Coverage reports are available in AWS Cost Explorer. It is used to check how much Reserved Instance (RIs) is used or overspent by AWS resources (Amazon EC2, Amazon Redshift, Amazon RDS, Amazon Elasticsearch Service, and Amazon ElastiCache) in a specific period.

AWS Cost Explorer provides recommendations for Reserved Instance (RI) purchases based on past usage and enhances opportunities for savings as compared to On-Demand usage.

To access information from the Reserved Instance Utilization report, one must enable Amazon's Cost Explorer.

Reserved Instance Utilization and Coverage report both can be exported to both PDF and CSV formats.

### RI utilization reports:

- Reserved Instance Reporting displays the total number of RI hours used by the account and helps to understand and monitor combined utilization across all of the RIs and services.
- AWS calculates the total savings by subtracting the costs of unused reservations from the reservations savings.

### RI coverage reports:

- Reserved Instance Reporting displays the percentage of instance hours used by the account and helps to understand and monitor the combined coverage across all of your RIs.
- The RI coverage reports use the Cost Explorer filters instead of the RI Utilization filters to analyze the purchasing accounts, instance types, and more.

### Reserved Instance Utilization and Coverage reports (both):

- Target utilization (threshold utilization) of RI utilization reports and Target coverage of RI coverage reports can be viewed as a dotted line in the chart and in the table below the chart as a colored status bar.
  - Red status bar - RIs with no hours used.
  - Yellow status bar - Under the utilization target.
  - Green status bar - Reached utilization target.
  - Gray status bar - instances not using reservations.
- RI reports make use of a combined filter of RI-specific and Cost Explorer.
- Daily RI Utilization chart - displays RI utilization for the previous three months daily.
- Monthly RI Utilization chart - displays your RI utilization for the previous 12 months monthly.

### Price details:

There is a cost of \$0.01 USD per request to retrieve the recommendation data in AWS.

# AWS Personal Health Dashboard

## What is AWS Personal Health Dashboard?

AWS Personal Health Dashboard is powered by the AWS Health API that provides alerts and remediation measures to diagnose and resolve issues related to AWS resources and infrastructure.

AWS Health provides continuous visibility into performance and the availability of the AWS services.

AWS offers two dashboards: AWS Service Health Dashboard and the Personal Health Dashboard.

- AWS Service Health Dashboard provides access to the current status and a complete health check of all services in all regions.
- The Personal Health Dashboard provides notification of any service interruptions that may affect the resources within the AWS account.

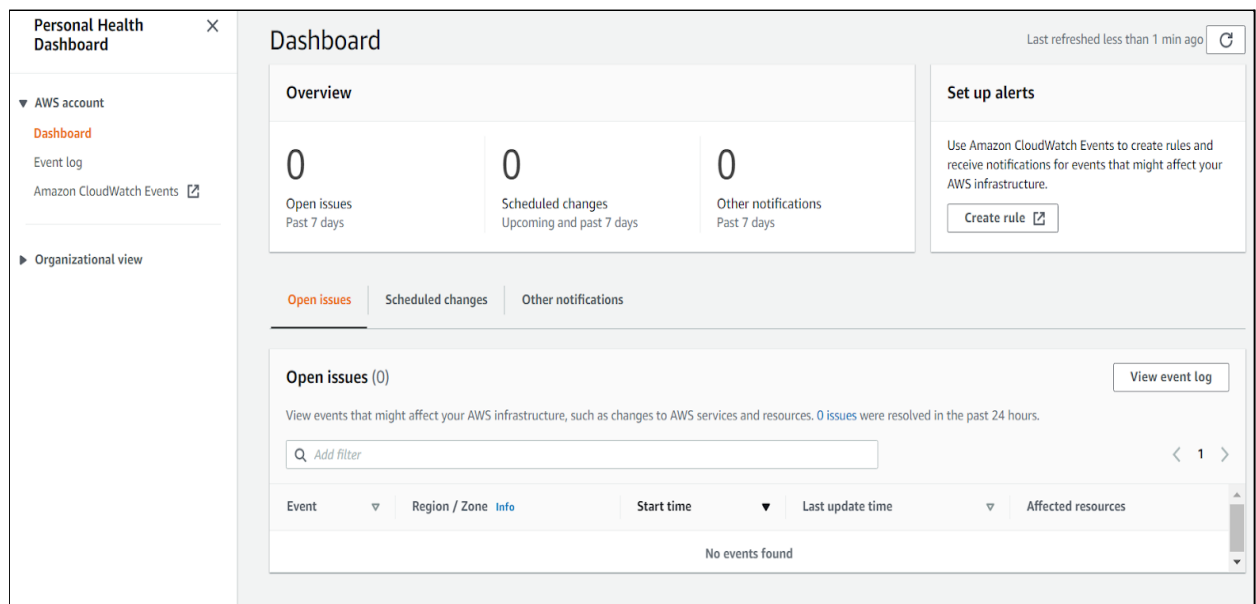
At the Personal Health Dashboard, there are three categories:

Open issues - shows issues of the last seven days.

Scheduled changes - shows items of any upcoming changes.

Other notifications.

AWS Personal Health Dashboard integrates with Amazon CloudWatch Events to create custom rules and specify targets such as AWS Lambda functions to enable remediation actions.



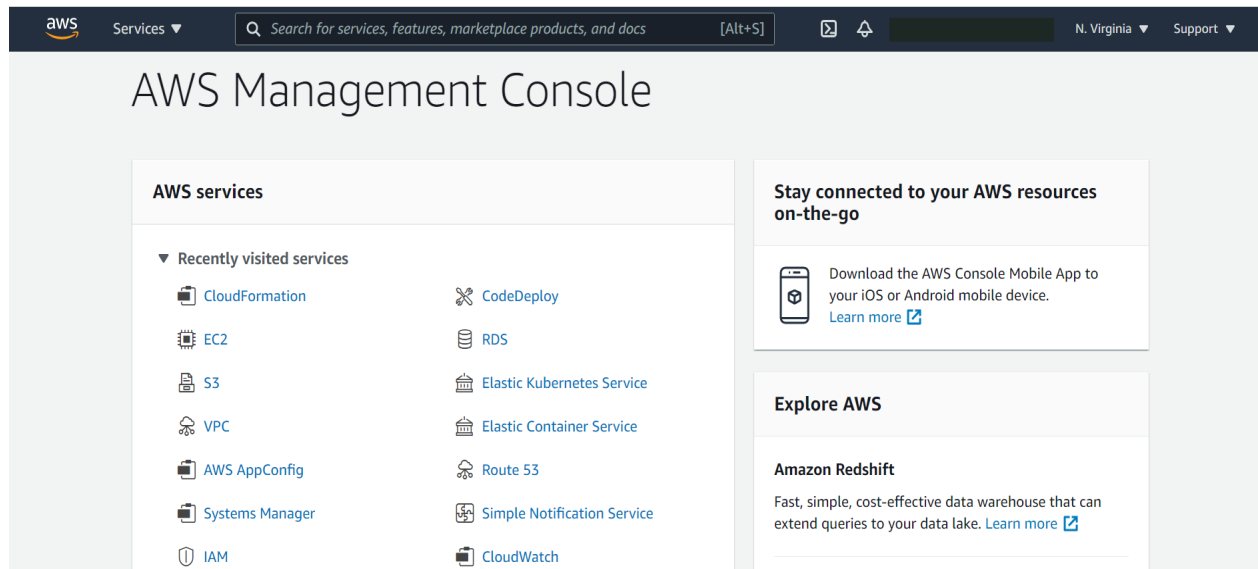
*AWS Personal Health Dashboard*

# AWS Management Console

## What is AWS Management Console?

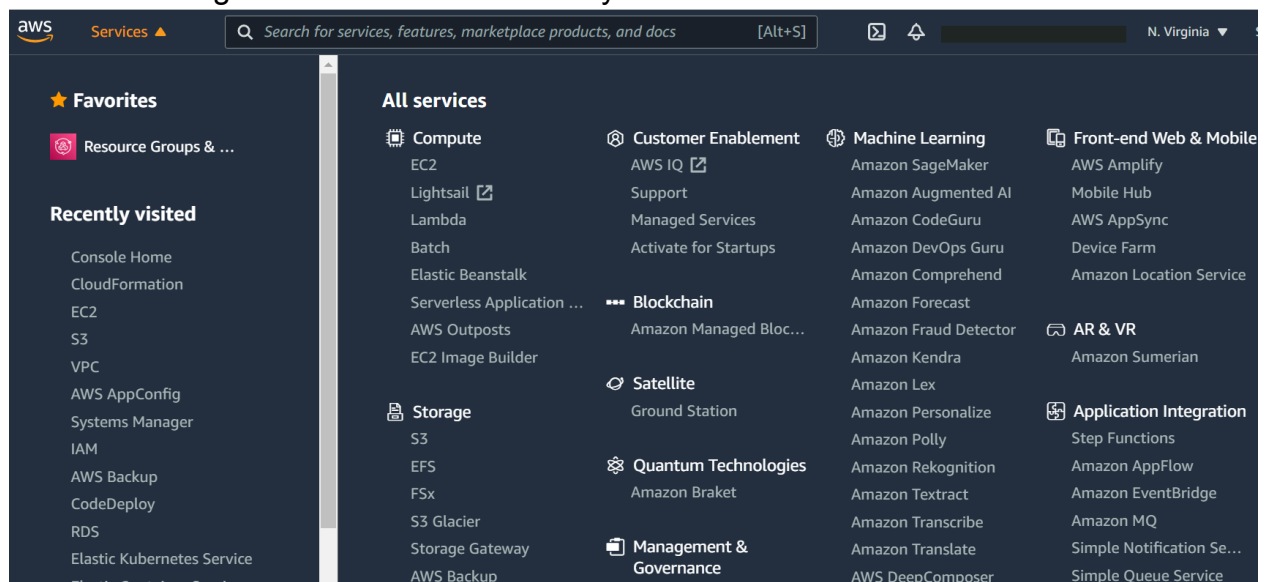
AWS Management Console is a web application that consists of many service consoles for managing Amazon Web Services.

It can be visible at the time a user first signs in. It provides access to other service consoles and a user interface for exploring AWS.



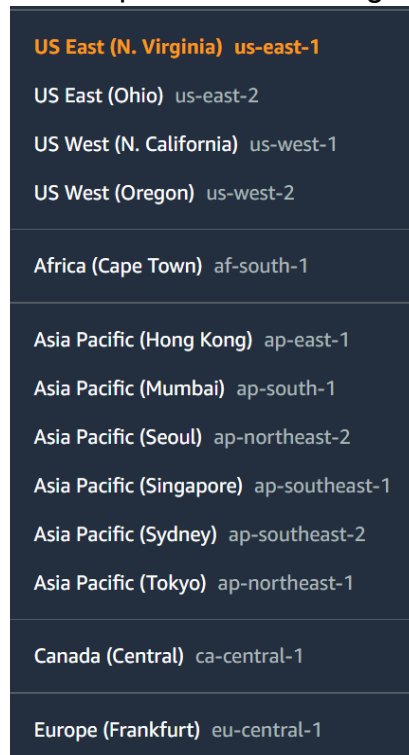
*AWS Management Console*

AWS Management Console provides a Services option on the navigation bar that allows choosing services from the Recently visited list or the All services list.



*AWS Services Console*

On the navigation bar, there is an option to select Regions from.

A screenshot of the AWS Regions dropdown menu. The menu is dark blue with white text. It lists various AWS regions, with 'US East (N. Virginia) us-east-1' highlighted in orange. The regions are grouped into sections: US East, US West, Africa, Asia Pacific, Canada, and Europe.

US East (N. Virginia)	us-east-1
US East (Ohio)	us-east-2
US West (N. California)	us-west-1
US West (Oregon)	us-west-2
Africa (Cape Town)	af-south-1
Asia Pacific (Hong Kong)	ap-east-1
Asia Pacific (Mumbai)	ap-south-1
Asia Pacific (Seoul)	ap-northeast-2
Asia Pacific (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Asia Pacific (Tokyo)	ap-northeast-1
Canada (Central)	ca-central-1
Europe (Frankfurt)	eu-central-1

*AWS Regions*

On the navigation bar, there is a Search box to search any AWS services by entering all or part of the name of the service.

The Console is also available as an app for Android and iOS with maximized Horizontal and vertical space and larger buttons for a better touch experience.