

Are you Ready for AWS Certified Developer - Associate exam?  
Self-assess yourself with "[Whizlabs](#)  
[FREE TEST](#)"



## AWS Certified Developer - Associate (DVA-C02) WhizCard

Quick Bytes for you before the exam!

*The information provided in WhizCards is for educational purposes only; created in our efforts to help aspirants prepare for the AWS Certified Developer - Associate certification exam. Though references have been taken from AWS documentation, it's not intended as a substitute for the official docs. The document can be reused, reproduced, and printed in any form; ensure that appropriate sources are credited and required permissions are received.*



Service Names	Page No.
<b>Analytics</b>	
1. Amazon Athena	5
2. Amazon OpenSearch Service	6
3. AWS Kinesis Data Streams	7
4. Amazon Kinesis Data Firehose	8
5. Amazon Kinesis Data Analytics	9
<b>Application Integration</b>	
6. AWS Step Functions	12
7. AWS EventBridge	13
8. AWS SNS	14
9. AWS SQS	15
10. AWS AppSync	16
<b>Compute</b>	
11. AWS EC2	18
12. Amazon EC2 Auto Scaling	19
13. AWS Elastic Beanstalk	20
14. AWS Lambda	21
15. AWS Serverless Application Model	22

Service Names	Page No.
<b>Containers</b>	
16. Amazon Elastic Container Registry	25
17. Amazon Elastic Container Service	26
18. Amazon Elastic Kubernetes Service	27
19. AWS Fargate	28
20. AWS Copilot	29
<b>Databases</b>	
21. Amazon Aurora	32
22. Amazon DynamoDB	33
23. Amazon ElastiCache	34
24. Amazon RDS	35
<b>Security, Identity, and Compliance</b>	
25. Amazon IAM	37
26. Amazon Cognito	38
27. AWS Certificate Manager	39
28. AWS Key Management Service	40
29. AWS Secrets Manager	41
30. AWS STS	42
31. AWS Private Certificate Authority	44

# Index

Service Names	Page No.
32. AWS WAF	46
<b>Front-End Web and Mobile</b>	
33. Amazon API Gateway	49
<b>Management and Governance</b>	
34. Amazon CloudWatch	51
35. Amazon CloudWatch Logs	52
36. AWS AppConfig	54
37. AWS CloudFormation	56
38. AWS CloudTrail	57
39. AWS Systems Manager	58
<b>Networking and Content Delivery</b>	
40. Amazon VPC	60
41. Amazon CloudFront	61
42. Amazon Route 53	62
43. Elastic Load Balancing	63

Service Names	Page No.
<b>Storage</b>	
44. Amazon S3	65
45. Amazon Elastic Block Storage	66
46. Amazon Elastic File System	68
47. Amazon S3 Glacier	69
<b>Developer Tools</b>	
48. AWS CodeCommit	71
49. AWS CodeBuild	72
50. AWS CodeDeploy	73
51. AWS CodePipeline	74
52. AWS Amplify	76
53. AWS CloudShell	78
54. AWS Cloud9	80
55. AWS Code Artifact	81
56. AWS CodeStar	83
57. Amazon Code Whisperer	85
58. AWS X-Ray	87



# Analytics

# Amazon Athena

## What is Amazon Athena?

Amazon Athena is an interactive serverless service used to analyze data directly in Amazon Simple Storage Service using standard SQL ad-hoc queries.



## Pricing Details:



- ❑ Charges are applied based on the amount of data scanned by each query at standard S3 rates for storage, requests, and data transfer.
- ❑ Canceled queries are charged based on the amount of data scanned.
- ❑ No charges are applied for Data Definition Language (DDL) statements.
- ❑ Charges are applied for canceled queries also based on the amount of data scanned.
- ❑ Additional costs can be reduced if data gets compressed, partitioned, or converted into a columnar format.

## Functions of Athena:



- ❑ It helps to analyze different kinds of data (unstructured, semi-structured, and structured) stored in Amazon S3.
- ❑ Using Athena, ad-hoc queries can be executed using ANSI SQL without actually loading the data into Athena.
- ❑ It can be integrated with Amazon Quick Sight for data visualization and helps to generate reports with business intelligence tools.
- ❑ It helps to connect SQL clients with a JDBC or an ODBC driver.
- ❑ It executes multiple queries in parallel, so no need to worry about compute resources.
- ❑ It supports various standard data formats, such as CSV, JSON, ORC, Avro, and Parquet.

# Amazon OpenSearch Service








OpenSearch Service is a free and open-source search engine for all types of data like textual, numerical, geospatial, structured, and unstructured.



## What is Amazon OpenSearch Service?

Amazon OpenSearch Service is a managed service that allows users to deploy, manage, and scale OpenSearch clusters in the AWS Cloud. It provides direct access to the OpenSearch APIs.

### Amazon OpenSearch Service can be integrated with following services:

- Amazon CloudWatch 
- Amazon CloudTrail 
- Amazon Kinesis 
- Amazon S3 
- AWS IAM 
- AWS Lambda 
- Amazon DynamoDB 

- ✓ Amazon OpenSearch Service with Kibana (visualization) & Logstash (log ingestion) provides an enhanced search experience for the applications and websites to find relevant data quickly.
- ✓ Amazon OpenSearch Service launches the OpenSearch cluster's resources and detects the failed OpenSearch nodes and replaces them.
- ✓ The OpenSearch Service cluster can be scaled with a few clicks in the console.

### Pricing Details:



- Charges are applied for each hour of use of EC2 instances and storage volumes attached to the instances.
- Amazon OpenSearch Service does not charge for data transfer between availability zones.

# Amazon Kinesis Data Streams

## What are Amazon Kinesis Data Streams?

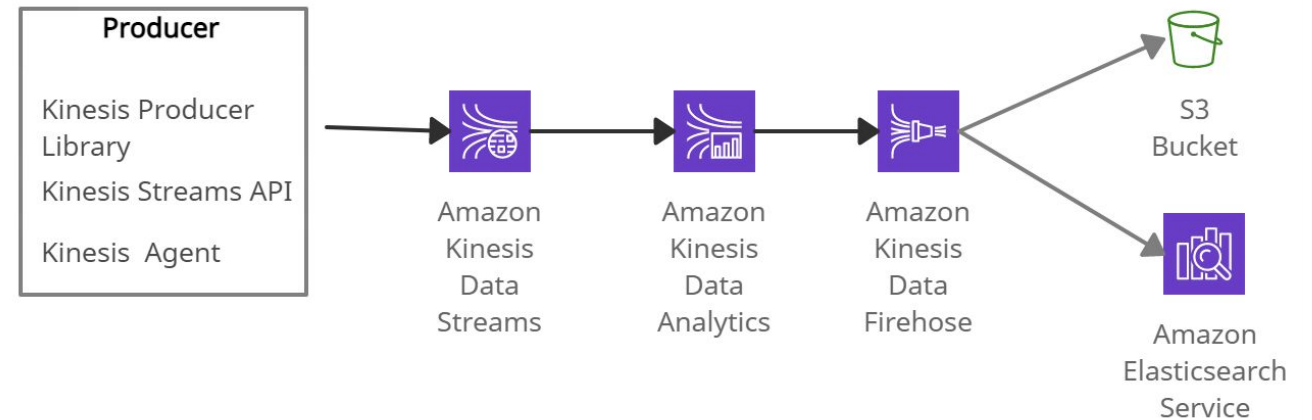
Amazon Kinesis Data Streams (KDS) is a scalable real-time data streaming service. It captures gigabytes of data from sources like website clickstreams, events streams (database and location-tracking), and social media feeds.



### Amazon Kinesis Data Streams

**Amazon Kinesis** is a service used to collect, process, and analyze real-time streaming data. It can be an alternative to Apache Kafka.

- ❑ Kinesis family consists of Kinesis Data Streams, Kinesis Data Analytics, Kinesis Data Firehose, and Kinesis Video Streams.
- ❑ The Real-time data can be fetched from Producers that are Kinesis Streams API, Kinesis Producer Library (KPL), and Kinesis Agent.
- ❑ It allows building custom applications known as Kinesis Data Streams applications (Consumers), which reads data from a data stream as data records.



### Amazon Kinesis Data Streams

Data Streams are divided into Shards / Partitions whose data retention is 1 day (by default) and can be extended to 7 days

Each shard provides a capacity of 1MB per second input data and 2MB per second output data.

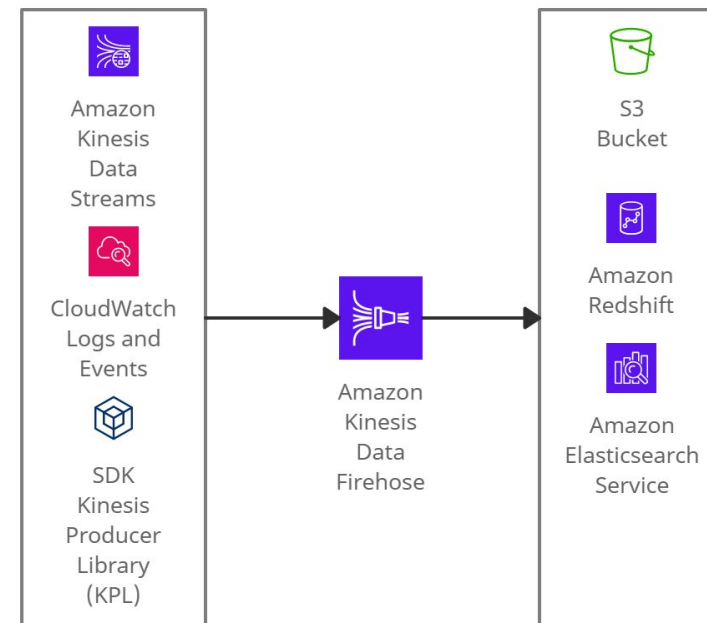
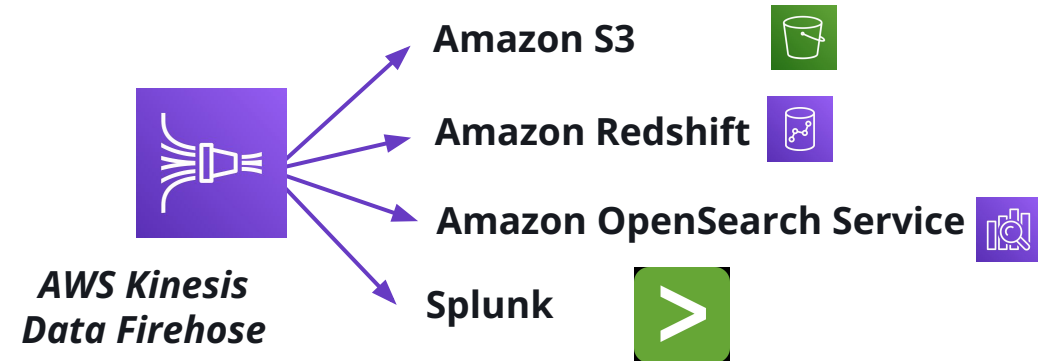
# Amazon Kinesis Data Firehose

## What is Amazon Kinesis Data Firehose?

Amazon Kinesis Data Firehose is a serverless service used to capture, transform, and load streaming data into data stores and analytics services.

- ❖ It synchronously replicates data across three AZs while delivering them to the destinations.
- ❖ It allows real-time analysis with existing business intelligence tools and helps to transform, batch, compress and encrypt the data before delivering it.
- ❖ It creates a Kinesis Data Firehose delivery stream to send data. Each delivery stream keeps data records for one day.
- ❖ It has 60 seconds minimum latency or a minimum of 32 MB of data transfer at a time.
- ❖ Kinesis Data Streams, CloudWatch events can be considered as the source(s) to Kinesis Data Firehose.

*It delivers streaming data to the following services:*



**AWS Kinesis Data Firehose**



# Amazon Kinesis Data Analytics

## What is Amazon Kinesis Data Analytics?

Amazon Kinesis Data Analytics is a cloud-native offering within the AWS ecosystem, designed to simplify the processing and analysis of real-time streaming data. It is an integral component of the broader Amazon Kinesis family, which is tailored to streamline operations involving streaming data.

### Limitations:

**Data Retention:** Data retention in Kinesis Data Analytics is generally limited. You may need to store your data in another AWS service (e.g., Amazon S3) if you require long-term storage of streaming data.

**Throughput:** There are limits on the maximum throughput that Kinesis Data Analytics can handle. If you need to process extremely high volumes of streaming data, you may need to consider partitioning your data streams and scaling your application accordingly.

**Resource Allocation:** AWS manages the underlying infrastructure for Kinesis Data Analytics, but you may have limited control over the resource allocation. This means that you might not be able to fine-tune the resources allocated to your application.

### Features

**Real-time Data Processing:** Kinesis Data Analytics can ingest and process data streams in real-time, making it well-suited for applications that require immediate insights and responses to streaming data, such as IoT (Internet of Things) applications, clickstream analysis, and more.

**SQL-Based Programming:** You can write SQL queries to transform, filter, aggregate, and analyze streaming data without the need for low-level coding. It may not support very complex SQL queries or advanced analytical functions found in traditional databases.

**Integration with Other AWS Services:** Kinesis Data Analytics can easily integrate with other AWS services like Kinesis Data Streams (for data ingestion), Lambda (for serverless computing), and various data storage and analytics tools like Amazon S3, Amazon Redshift, and more.

**Real-time Analytics Applications:** You can use Kinesis Data Analytics to build real-time analytics applications, perform anomaly detection, generate alerts based on streaming data patterns, and even create real-time dashboards to visualize your insights.

**Scalability:** Kinesis Data Analytics is designed to scale automatically based on the volume of data you're processing, ensuring that your analytics application can handle growing workloads without manual intervention.

## Amazon Kinesis Data Analytics

### Best Practices

**Use Appropriate Windowing:** Utilize windowing functions effectively to define the scope of your data analysis. Choose tumbling or sliding windows based on your specific use case and requirements.

**Partition Your Data:** Distribute your data across partitions in a way that allows for parallel processing for better scalability and performance.

**Use Schema Inference or Schema Registry:** If your data streams have evolving schemas, consider using schema inference or a schema registry to manage schema changes without interrupting your data processing.

Configure appropriate TTL (Time-to-Live) settings for your in-application data.

### Pricing

Kinesis Data Streams adopts a straightforward pay-as-you-go pricing model, ensuring that there are no initial expenses or minimum charges. You are billed solely based on the resources you consume. Kinesis Data Streams provides two capacity modes—on-demand and provisioned—each with its own billing choices to cater to different use cases.

### Use Cases:

**Real-time Anomaly Detection: Use Case: Monitor IoT sensor data for anomalies.**

In an IoT application, you can ingest sensor data from devices in real time using Amazon Kinesis Data Streams. Kinesis Data Analytics can then analyze this data using machine learning algorithms to detect anomalies or unusual patterns, such as a sudden temperature spike or a drop in device activity. When an anomaly is detected, you can trigger alerts or take automated actions to address the issue.

**Clickstream Analysis: Analyze user behavior on a website or mobile app.**

When users interact with your website or app, their actions generate clickstream data. You can use Amazon Kinesis Data Analytics to process and analyze this data in real time.



# Application Integration

# AWS Step Functions

## What is AWS Step Functions?

AWS Step Functions is a serverless orchestration service that converts an application's workflow into a series of steps by combining AWS Lambda functions and other AWS services.



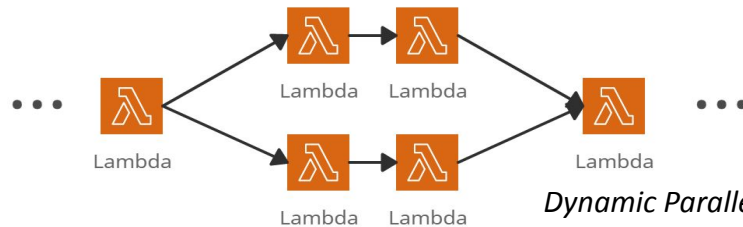
### Standard Workflows

- It executes once in a workflow execution for up to one year.
- They are ideal for long-running and auditable workflows.

### Express Workflows

- It executes at-least-once in a workflow execution for up to five minutes.
- They are ideal for high-processing workloads, such as streaming data processing and IoT data ingestion.

**Executions** are the instances where workflow runs to perform tasks.



*Dynamic Parallelism using AWS Step Functions*

- AWS Step Functions resembles state machines and tasks. Each step in a workflow is a state. The output of one step signifies an input to the next results in functions orchestration.
- It helps to execute each step in an order defined by the business logic of the application.
- It provides some built-in functionalities like sequencing, error handling, timeout handling, and removing a significant operational overhead from the team.
- It can control other AWS services, like AWS Lambda (to perform tasks), processing machine learning models, AWS Glue (to create an extract, transform, and load (ETL) workflows), and automated workflows that require human approval.
- It provides multiple automation features like routine deployments, upgrades, installations, migrations, patch management, infrastructure selection, and data synchronization



*Functions Orchestration using AWS Step Functions*

# Amazon EventBridge

## What is Amazon EventBridge?

Amazon EventBridge is a serverless event bus service that connects applications with data from multiple sources.



### ***Amazon EventBridge integrates with the following services:***

- AWS CloudTrail
- AWS CloudFormation
- AWS Config
- AWS Identity and Access Management (IAM)
- AWS Kinesis Data Streams
- AWS Lambda

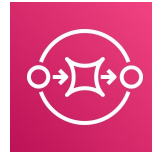
## Functions of Amazon EventBridge:

- An event bus is an entity that receives events, and rules get attached to that event bus that matches the events received.
- It helps to build loosely coupled and distributed event-driven architectures.
- It connects applications and delivers the events without the need to write custom code.
- It delivers a stream of real-time data from SaaS applications or other AWS services and routes that data to different targets such as Amazon EC2 instances, Amazon ECS tasks, AWS CodeBuild projects, etc.
- It sets up routing rules that determine the targets to build application architectures that react according to the data sources.
- The EventBridge schema registry stores a collection of event structures (schemas) and allows users to download code for those schemas in the IDE representing events as objects in the code.

# Amazon SNS

## What is Amazon SNS?

Amazon Simple Notification Service (Amazon SNS) is a serverless notification service that offers message delivery from publishers to subscribers.









- ✓ It creates asynchronous communication between publishers and subscribers by sending messages to a 'topic.'
- ✓ It supports application-to-application subscribers that include Amazon SQS and other AWS services and Application-to-person subscribers that include Mobile SMS, Email, etc.

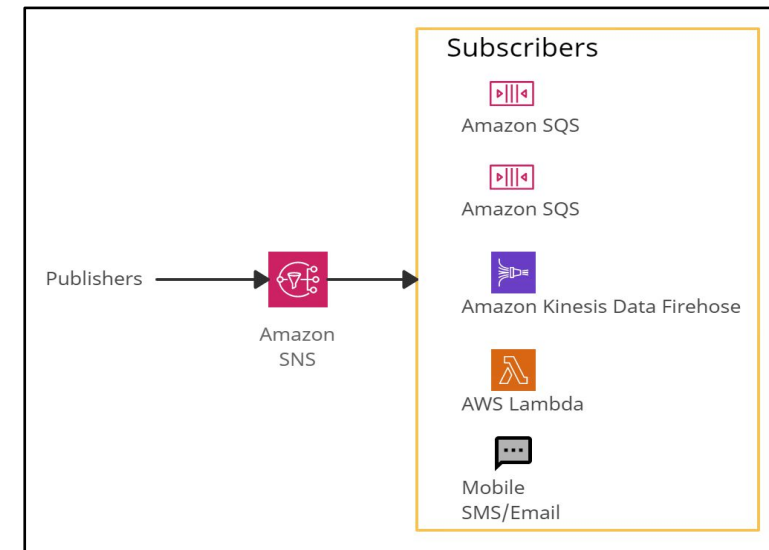
- The producer sends one message to one SNS topic.
- Multiple receivers (subscribers) listen for the notification of messages.
- All the subscribers will receive all the messages.

### Example:

1 message, 1 topic, 10 subscribers so that a single message will be notified to 10 different subscribers.

## SNS helps to publish messages to many subscriber endpoints:

- Amazon SQS Queues 
- AWS Lambda Functions 
- Email 
- Amazon Kinesis Data Firehose 
- Mobile push 
- SMS 



Amazon SNS

# Amazon SQS

## What are 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:

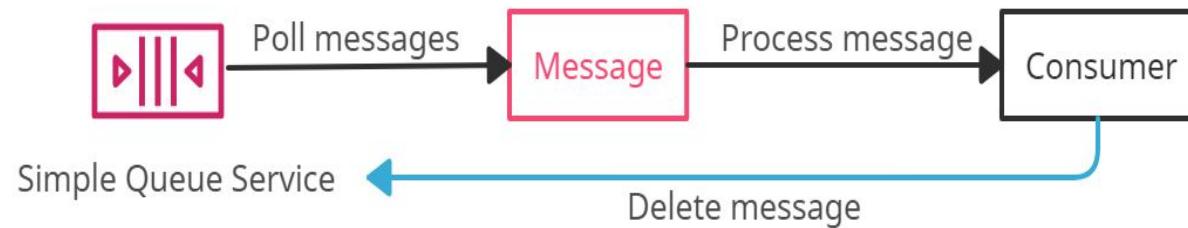
### Standard Queue -

- ❖ 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.

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



**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.

**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 AppSync

## What is AWS AppSync?

AWS AppSync is a serverless service used to build GraphQL API with real-time data synchronization and offline programming features.

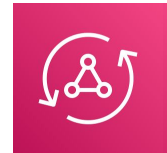
- It replaces the functionality of Cognito Sync by providing offline data synchronization.
- It improves performance by providing data caches, provides subscriptions to support real-time updates, and provides client-side data stores to keep off-line clients in sync.
- It offers certain advantages over GraphQL, such as enhanced coding style and seamless integration with modern tools and frameworks like iOS and Android
- AppSync interface provides a live GraphQL API feature that allows users to test and iterate on GraphQL schemas and data sources quickly.
- Along with AppSync, AWS provides an Amplify Framework that helps build mobile and web applications using GraphQL APIs.

GraphQL is a data language built to allow apps to fetch data from servers.

## The different data sources supported by AppSync are:

- Amazon DynamoDB tables
- RDS Databases
- Amazon Elasticsearch
- AWS Lambda Functions
- Third Party HTTP Endpoints

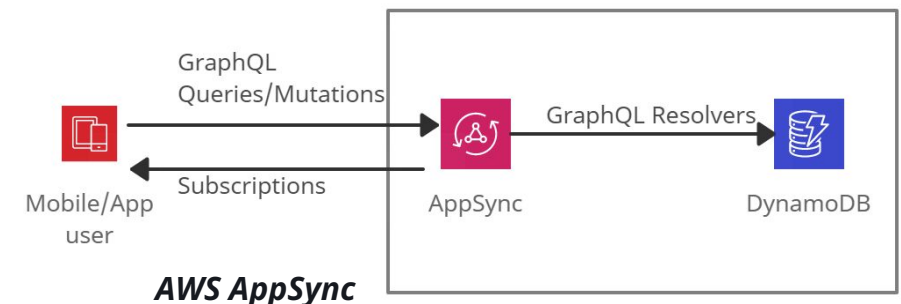
**AWS AppSync**



**Queries:** For fetching data from the API

**Mutations:** For changing data via API

**Subscriptions:** The connections for streaming data from API





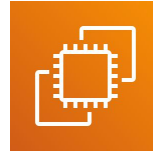


# Compute

# Amazon EC2

## What is Amazon EC2?

Amazon Elastic Compute Cloud (Amazon EC2) is a service that provides secure and scalable compute capacity in the AWS cloud. It falls under the category of Infrastructure as a Service (IAAS).



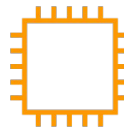
**It provides the different type of instances based on the pricing models:**

### **On-Demand Instances**

- ✓ Useful for short-term needs, unpredictable workloads.
- ✓ No advance payment, no prior commitment.

### **Spot Instances**

- ✓ No advance payment, no prior commitment.
- ✓ Useful for cost-sensitive compute workloads.



### **Reserved Instances**

- ✓ Useful for long-running workloads and predictable usage.
- ✓ Offer to choose from No upfront, Partial upfront, or All upfront.

### **Dedicated Instances**

- ✓ Instances run on hardware dedicated to a single user.
- ✓ Other customers can not share the hardware.

### **Dedicated Hosts**

- ✓ A whole physical server with an EC2 instance allocates to an organization.

It provides different compute platforms and instance types based on price, CPU, operating system, storage, and networking, and each instance type consists of one or more instance sizes. Eg., t2.micro, t4g.nano, m4.large, r5a.large, etc.

It provides pre-configured templates that package the operating system and other software for the instances. This template is called Amazon Machine Images (AMIs).

It helps to login into the instances using key-pairs, in which AWS manages the public key, and the user operates the private key.

It also provides firewall-like security by specifying IP ranges, type, protocols (TCP), port range (22, 25, 443) using security groups.

It provides temporary storage volumes known as instance store volumes, which are deleted if the instance gets stopped, hibernated, or terminated. It also offers non-temporary or persistent volumes known as Amazon EBS volumes.

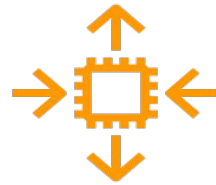
It enables users to write scripts under the option 'User data,' used at the instances' launch.

It offers to choose from three IP addresses, which are Public IP address (Changes when the instance is stopped or refreshed), Private IP address (retained even if the model is stopped), Elastic IP address (static public IP address).

# Amazon EC2 Auto Scaling

## What is Amazon EC2 Auto Scaling?

Amazon EC2 Auto Scaling is a region-specific service used to maintain application availability and enables users to automatically add or remove EC2 instances according to the compute workloads.



- ❖ The Auto Scaling group is a collection of the minimum number of EC2 used for high availability.
- ❖ It enables users to use Amazon EC2 Auto Scaling features such as fault tolerance, health check, scaling policies, and cost management.
- ❖ The scaling of the Auto Scaling group depends on the size of the desired capacity. It is not necessary to keep DesiredCapacity and MaxSize equal.

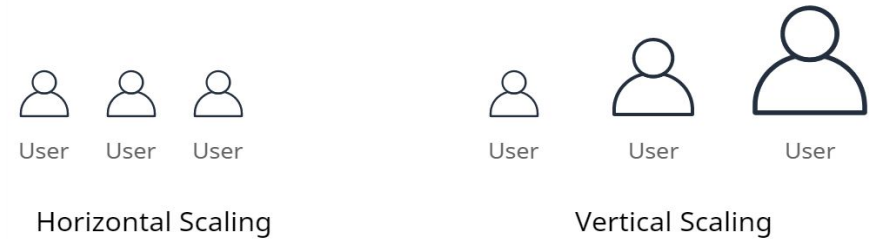
**E.g.,**  
**DesiredCapacity: '2' - There will be total 2 EC2 instances**  
**MinSize: '1'**  
**MaxSize: '2'**

- ❖ EC2 Auto Scaling supports automatic Horizontal Scaling (increases or decreases the number of EC2 instances) rather than Vertical Scaling (increases or decreases EC2 instances like large, small, medium).

## Launch Template

- A **launch template** is similar to launch configuration with extra features as below
- It launches both Spot and On-Demand instances.
- It specifies multiple instance types
- It specifies multiple launch templates.

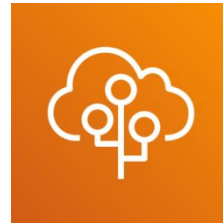
It scales across multiple Availability Zones within the same AWS region.



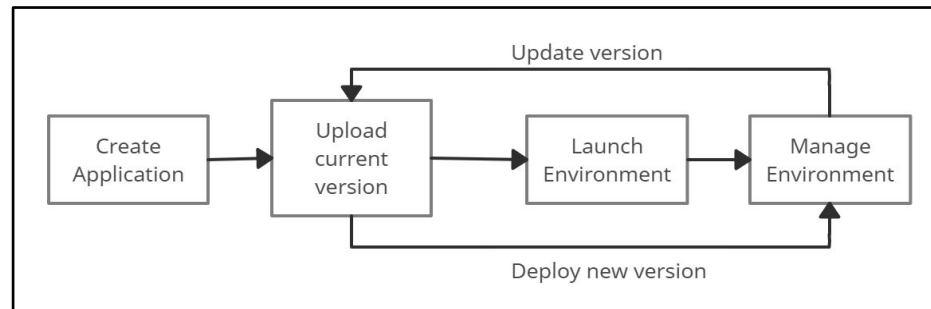
# AWS Elastic Beanstalk

## What is AWS Elastic Beanstalk?

AWS Elastic Beanstalk is a service used to quickly deploy, scale, and manage applications in the AWS Cloud with automatic infrastructure management.



- It falls under the category of Platform as a Service (PaaS)
- It is also defined as a developer-centric view of deploying an application on AWS. The only responsibility of the developer is to write, and Elastic Beanstalk handles code and the infrastructure
- An Elastic Beanstalk application comprises components, including environments, versions, platforms, and environment configurations.



The workflow of Elastic Beanstalk

- Elastic Beanstalk console offers users to perform deployment and management tasks such as changing the size of Amazon EC2 instances, monitoring (metrics, events), and environment status.

- It supports web applications coded in popular languages and frameworks such as Java, .NET, Node.js, PHP, Ruby, Python, Go, and Docker.
- It uses Elastic Load Balancing and Auto Scaling to scale the application based on its specific needs automatically.

It provides multiple deployment policies such as:

- All at once, Rolling
- Rolling with an additional batch
- Immutable
- Traffic splitting

### AWS CloudFormation vs. AWS Elastic Beanstalk

AWS CloudFormation	AWS Elastic Beanstalk
It deploys infrastructure using YAML/JSON template files.	It deploys applications on EC2.
It can deploy Elastic Beanstalk environments.	It cannot deploy CloudFormation templates.

# AWS Lambda

## What is AWS Lambda?

AWS Lambda is a serverless computing service that allows users to run code as functions without provisioning or managing servers.



It helps to run the code on highly-available infrastructure and performs administrative tasks like server maintenance, logging, capacity provisioning, and automatic scaling and code monitoring.

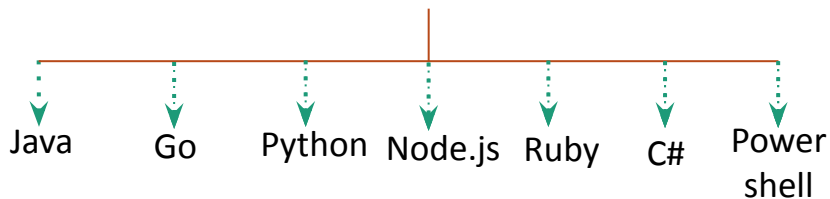
Using AWS Lambda, one can build serverless applications composed of Lambda functions triggered by events and can be automatically deployed using AWS CodePipeline and AWS CodeBuild.

Amazon EC2	Amazon Lambda
They are termed virtual servers in the AWS cloud.	They are termed virtual functions.
It is limited to instance types (RAM and CPU).	Limited by time (less execution time of 300 seconds).
It runs continuously.	It runs on demand.
Scaling computing resources is manual.	It has automated scaling.

✓ The memory allocated to AWS Lambda for computing is 128MB (minimum) to 3008MB (maximum). Additional memory can be requested in an increment of 64MB between 128MB - 3008MB.

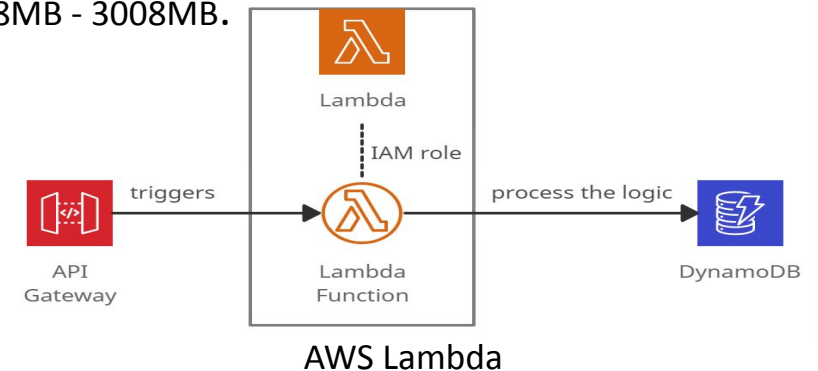
✓ The default execution time for AWS Lambda is 3 seconds, and the maximum is 15 minutes (900 seconds).

Lambda Functions supports the following languages:



## Pricing details: 💰

Charges are applied based on the number of requests for the functions and the time taken to execute the code



## What is AWS Serverless Application Model?

AWS Serverless Application Model (AWS SAM) is an open-source framework used for constructing serverless applications on AWS. It offers a streamlined method for defining the structure of serverless applications, encompassing AWS Lambda functions, Amazon API Gateway APIs, Amazon DynamoDB tables, and various other AWS resources, through a template-driven approach.

### CLI Commands

#### Init:

- ``sam init``: Initializes a new AWS SAM project, creating a basic project structure with sample code and a SAM template.

#### Build:

- ``sam build``: Prepares your AWS SAM application for deployment by installing dependencies and creating deployment packages.

#### Features:

**Simplified Serverless Application Definition:** AWS SAM provides a simplified, declarative syntax for defining serverless applications.

**Built-In Resource Types:** AWS SAM extends AWS CloudFormation with resource types specifically designed for serverless applications, but it may not cover all possible AWS resources or configurations.

**Local Development and Testing:** AWS SAM CLI allows for local development and testing of serverless applications.

**Auto-Scaling and Event-Driven:** Serverless applications built with AWS SAM can automatically scale in response to incoming events or requests.

**Integration with AWS Services:** AWS SAM integrates seamlessly with various AWS services, including AWS Lambda, Amazon API Gateway, Amazon DynamoDB, Amazon S3, Amazon SNS, and others.

**Event Sources and Triggers:** AWS SAM allows you to define event sources or triggers for Lambda functions, such as AWS API Gateway HTTP endpoints, S3 object uploads, DynamoDB streams, and more.

## Limitations:

**Limited Language Support:** AWS SAM has better support for certain programming languages like Python, Node.js, and Java compared to others.

**Vendor Lock-In:** AWS SAM is tightly integrated with AWS services, which may make it less portable to other cloud providers.

**Advanced Deployment Strategies:** If your application requires advanced deployment strategies like canary deployments or blue-green deployments, AWS SAM may not provide built-in support for these.

## Best Practices:

**Use the Latest AWS SAM Version:** Keep your AWS SAM CLI and templates up to date to take advantage of new features and improvements.

**Separate Your Functions:** Break down your application into smaller functions that perform specific tasks. This promotes reusability and maintainability.

**Use Layers for Code Reuse:** Store shared code (e.g., libraries or custom runtimes) in AWS Lambda layers to avoid duplication and reduce the size of your deployment packages.

**Leverage Environment Variables:** Store configuration values like API keys or database connection strings in environment variables rather than hardcoding them in your functions.

## Use Case:

- ❑ **Web Applications and APIs:** You can use AWS SAM to build web applications and APIs that scale automatically based on demand.
- ❑ **Data Processing and ETL Pipelines:** AWS SAM is suitable for building data processing and ETL (Extract, Transform, Load) pipelines that process large amounts of data without the need for server provisioning.
- ❑ **IoT Backend Services:** For Internet of Things (IoT) applications, AWS SAM can be used to develop backend services that handle device telemetry, data storage, and real-time processing.

## Pricing:

There are no extra costs associated with the usage of AWS SAM. You are billed only for the resources you utilize, and this payment structure does not entail any mandatory upfront commitments or minimum fees.



# Containers








# Amazon Elastic Container Registry

## What is Amazon Elastic Container Registry?

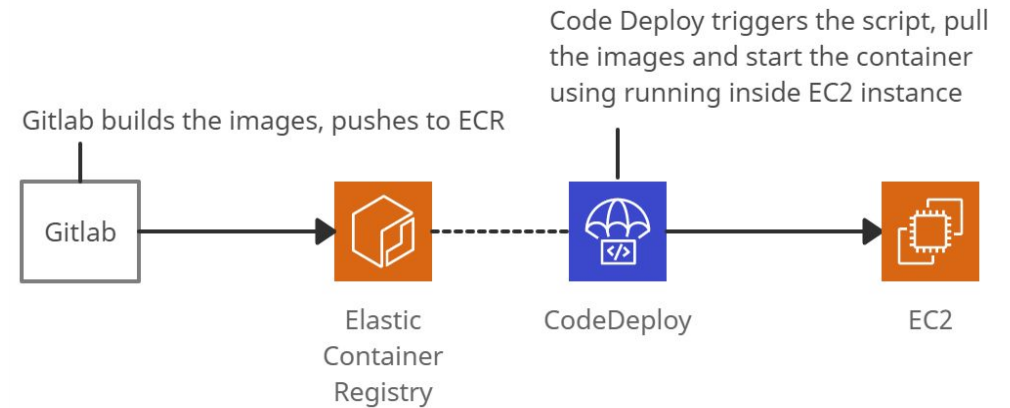
Amazon Elastic Container Registry (ECR) is a managed service that allows users to store, manage, share, and deploy container images and other artifacts.



- It stores both the containers which are created, and any container software bought through AWS Marketplace.
- It is integrated with the following services:

- Amazon Elastic Container Service(ECS) 
- Amazon Elastic Kubernetes Service(EKS) 
- AWS Lambda 
- Docker CLI 
- AWS Fargate for easy deployments 

- ▶ AWS Identity and Access Management (IAM) enables resource-level control of each repository within ECR.
- ▶ Amazon Elastic Container Registry (ECR) supports public and private container image repositories. It allows sharing container applications privately within the organization or publicly for anyone to download.
- ▶ Images are encrypted at rest using Amazon S3 server-side encryption or using customer keys managed by AWS Key Management System (KMS).
- ▶ Amazon Elastic Container Registry (ECR) is integrated with continuous integration, continuous delivery, and third-party developer tools.
- ▶ Image scanning allows identifying vulnerabilities in the container images. It ensures that only scanned images are pushed to the repository



Amazon ECR example

# Amazon Elastic Container Service

## What is Amazon Elastic Container Service?

Amazon Elastic Container Service is a regional and docker-supported service that allows users to manage and scale containers on a cluster.



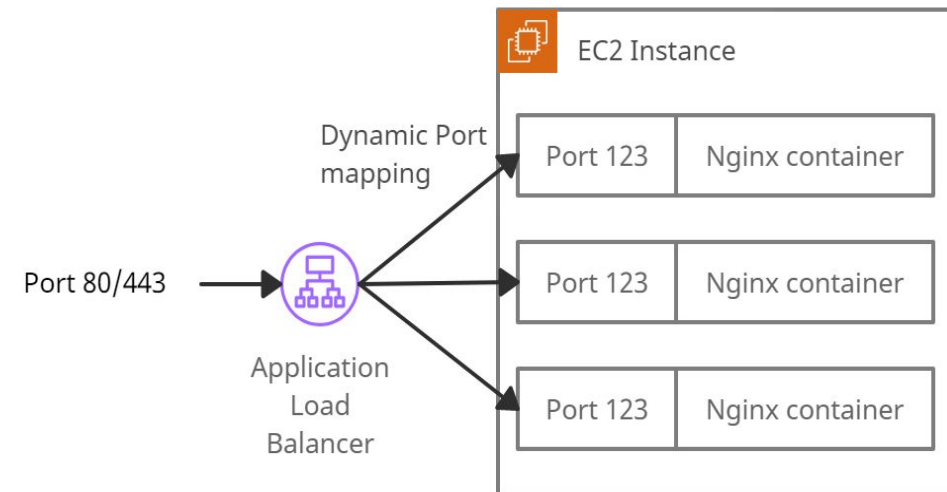
## Two main use cases of Amazon ECS are:



Microservices are built by the architectural method that decouples complex applications into smaller and independent services

*Batch Jobs* - Batch jobs are short-lived packages that can be executed using containers.

- ECS cluster is a combination of tasks or services executed on EC2 Instances or AWS Fargate.
- It offers to scale ECS clusters using Autoscaling based on CPU usage and other Autoscaling rules.
- Using Application Load Balancer, ECS enables dynamic port mapping and path-based routing.
- It provides Multi-AZ features for the ECS clusters.



Amazon ECS with Application Load Balancer

## Amazon Elastic Kubernetes Service(EKS)

### What is Amazon Elastic Kubernetes Service?

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



**Amazon EKS**

Using Amazon EKS, Kubernetes clusters and applications can be managed across hybrid environments 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.
- 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.

- 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.
- 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/managing Kubernetes clusters on Amazon EKS.
  - AWS Management Console and AWS CLI

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

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

# AWS Fargate

## What is AWS Fargate?

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

In the AWS Management Console, ECS clusters containing Fargate and EC2 tasks are displayed separately.

- »» It executes each task of Amazon ECS or pods of Amazon EKS in its kernel as an isolated computing environment and improves security.
- »» 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 automatically scales the compute environment that matches the resource requirements for the container.

Security groups for pods in EKS cannot be used when pods are running on Fargate.

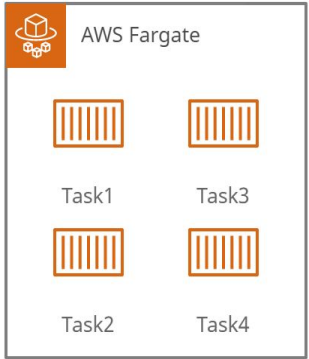


Storage Types supported for Fargate Tasks



Amazon EFS volume for persistent storage

Ephemeral Storage for non-persistent storage



**Difference between EC2 instance and AWS Fargate**

## AWS Copilot

### What is AWS Copilot?

AWS Copilot is used to simplify the process of building, deploying, and managing containerized applications on AWS infrastructure.

It is designed to streamline the development workflow for container-based applications, making it easier for developers to work with container services like AWS Fargate and Amazon Elastic Container Service (ECS).

### Features:

- ❑ **Application and Service Management:** AWS Copilot allows developers to define and manage applications and services using a simple and intuitive YAML-based configuration file.
- ❑ **Deployment Automation:** It automates the process of building and deploying containerized applications, taking care of tasks like container image creation, pushing images to Amazon Elastic Container Registry (ECR), creating ECS or Fargate tasks, and setting up load balancers and networking.
- ❑ **Local Development:** AWS Copilot provides tools for local development and testing of containerized applications. Developers can use it to run their containers locally on their development machines for testing and debugging purposes.
- ❑ **Continuous Integration and Continuous Deployment (CI/CD) Integration:** It integrates seamlessly with popular CI/CD tools like AWS CodePipeline and AWS CodeBuild, allowing for automated CI/CD pipelines for containerized applications.

### Important CLI Commands:

**copilot init:** This command initializes a new AWS Copilot application in your project directory. It prompts you to select an application name and a service name.

**copilot env init:** Use this command to create a new environment (e.g., development, staging, production) for your application.

**copilot app:** This command provides information about your AWS Copilot application, including its name, repository URL, and deployment status.

**copilot service:** You can use this command to manage your AWS Copilot services. For example, you can create a new service, view information about existing services, and deploy services using this command.

**copilot logs:** This command allows you to access and view logs from your deployed services. You can stream logs from a specific service or environment to your terminal.

## AWS Copilot

### Use Case:

**Microservices Deployment:** Building a microservices-based application with multiple services, each serving a specific function (e.g., user authentication, product catalog, order processing).

**Development and Staging Environments:** Create isolated development and staging environments for your application to test new features and changes before deploying them to production.

**Serverless Backend for Mobile Apps:** You are building a mobile app and need a serverless backend to handle user authentication, data storage, and API requests.

### Limitations

**AWS Region Availability:** Not all regions support all Copilot features and services.

**Service Complexity:** While AWS Copilot simplifies many aspects of containerized application deployment, it may not cover all possible use cases or complex application architectures.

**Lack of Support for All AWS Services:** If you need to integrate other AWS services extensively (e.g., AWS RDS for databases), you'll need to configure those separately.

**Service Size and Complexity:** AWS Copilot is well-suited for small to medium-sized applications. Very large and complex applications may require more granular control and customization than Copilot provides.

### Pricing

Amazon distributes AWS Copilot under an Open-Source license without any charges.

Customers are solely responsible for paying for the resources they generate through the CLI.



# Databases

# Amazon Aurora



## What is Aurora?

Amazon Aurora is a MySQL and PostgreSQL-compatible, fully managed relational database engine built to enhance traditional enterprise databases' performance and availability.

- Is a part of the fully managed Amazon Relational Database Service (Amazon RDS).

### Features include:

- RDS Management Console
- CLI commands and API operations for patching Backup
- Recovery
- Database Setup
- Failure Detection and repair

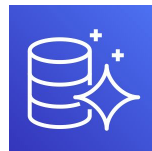
### Performance



5x greater than



MySQL on RDS



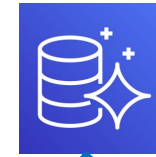
3x greater than



PostgreSQL on RDS

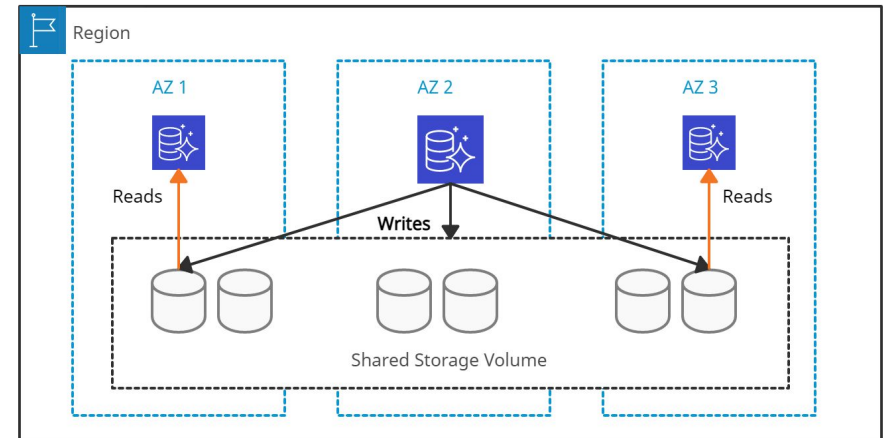
- Amazon Aurora replicates 2 copies of data in each availability zone (minimum of 3 AZ). So a total of 6 copies per region.

### Data Replication : 2 Types



- Aurora replica (in-region)** It can provide 15 read replicas.
- MySQL Read Replica (cross-region)** It can provide 5 read replicas.

Amazon Aurora Cross-Region read replicas help to improve disaster recovery and provide fast reads in regions closer to the application users.

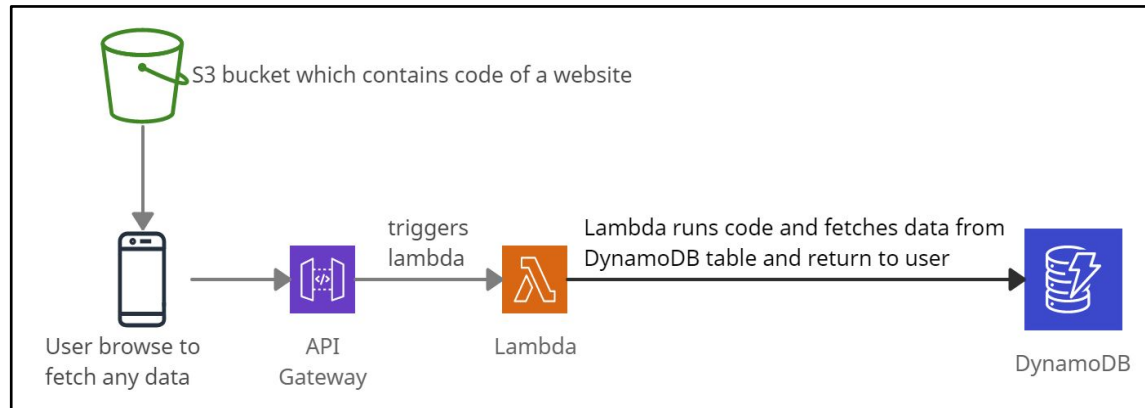
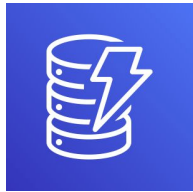




# Amazon DynamoDB

## What is Amazon DynamoDB?

Amazon DynamoDB is a serverless NoSQL database service that provides fast and predictable performance with single-digit millisecond latency.



*Amazon DynamoDB example*

- ▶ It provides a push button scaling feature, signifying that DB can scale without any downtime.
- ▶ It is a multi-region cloud service that supports key-value and document data structure.
- ▶ It provides high availability and data durability by replicating data synchronously on solid-state disks (SSDs) across 3 AZs in a region.
- ▶ It helps to store session states and supports ACID transactions for business-critical application
- ▶ It provides the on-demand backup capability of the tables for long-term retention and enables point-in-time recovery from accidental write or delete operations.
- ▶ Amazon DynamoDB Accelerator (DAX) is a highly available in-memory cache service that provides data from DynamoDB tables. DAX is not used for strongly consistent reads and write-intensive workloads.
- ▶ It supports Cross-Region Replication using DynamoDB Global Tables. Global Tables helps to deploy a multi-region database and provide automatic multi-master replication to AWS regions.

# Amazon ElastiCache

## What is Amazon ElastiCache?

ElastiCache is a web service used to manage and run in-memory data stores Redis and Memcached in the cloud.

- »» It is best suited for Online Analytical Processing (OLAP) transaction workloads and for storing session states.
- »» It has in-memory caching features to provide sub-millisecond latency for read-heavy application workloads.

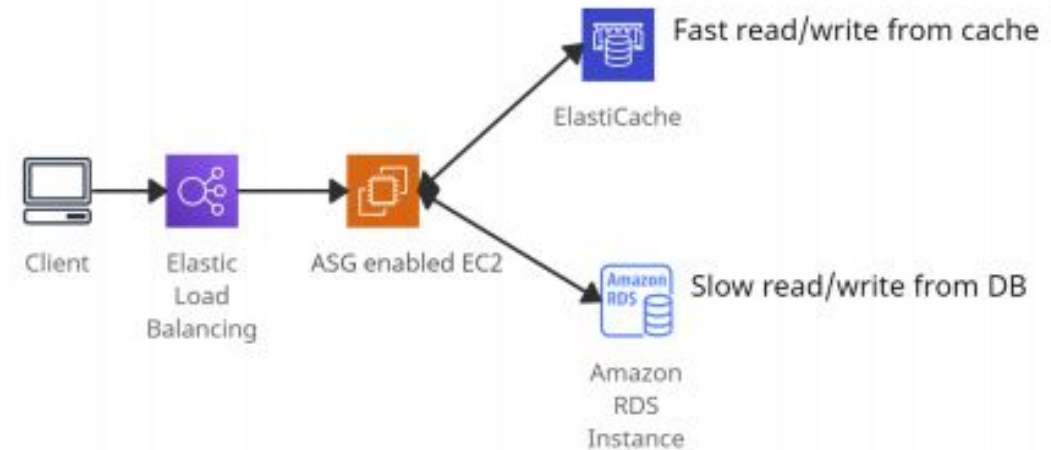
### Amazon ElastiCache for Redis:

- ❖ It is useful for gaming applications, geospatial services, caching, session stores, and replication.
- ❖ Data is persistent.
- ❖ It is not multi-threaded.
- ❖ It supports Multi-AZ using read replicas.



### Amazon ElastiCache for Memcached:

- ❖ It is useful for building applications that require caching layers.
- ❖ Data is not persistent.
- ❖ It supports multi-threading.
- ❖ It does not support Multi-AZ failover.
- ❖ It does not support snapshots.

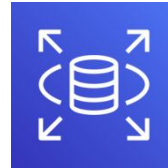


**Amazon ElastiCache**

# Amazon RDS

## What is Amazon RDS?

Amazon Relational Database Service (Amazon RDS) is a service used to build and operate relational databases in the AWS Cloud



RDS provides read replicas of reading replicas and can also read replicas as a standby DB like Multi-AZ.

Read replicas feature is not available for SQL Server.

- It is best suited for structured data and Online Transaction Processing (OLTP) types of database workloads such as InnoDB.

*It supports the following database engines:*

- SQL Server
- PostgreSQL
- Amazon Aurora
- MySQL
- MariaDB
- Oracle

- ✓ If there is a need for unsupported RDS database engines, DB can be deployed on EC2 instances.

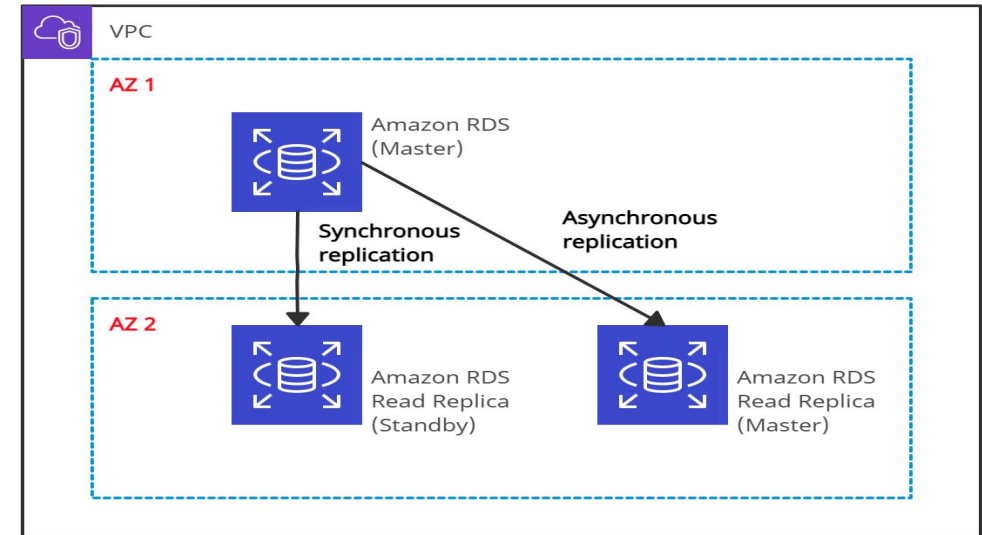
*The following tasks need to be taken care of manually.*

### Encryption and Security

### Updates and Backups

### Disaster Recovery

- AWS KMS provides encryption at rest for RDS instances, DB snapshots, DB instance storage, and Read Replicas. The existing database cannot be encrypted.
- Amazon RDS only scales up for compute and storage, with no option for decreasing allocated storage
- It provides Multi-AZ and Read Replicas features for high availability, disaster recovery, and scaling.
  - **Multi-AZ Deployments** - Synchronous replication
  - **Read Replicas** - Asynchronous replication.



Amazon RDS



# Security, Identity, and Compliance

## Amazon Identity and Access Management (IAM)

### What is Amazon IAM ?

AWS Identity and Access Management is a free service used to define permissions and manage users to access multi-account AWS services.



### *Amazon Identity and Access Management*

#### Amazon Identity and Access Management allows:

- ❖ users to analyze access and provide MFA (Multi-factor authentication) to protect the AWS environment.
- ❖ managing IAM users, IAM roles, and federated users.

#### **IAM Policies**

Policies are documents written in JSON (key-value pairs) used to define permissions.

#### **IAM Users**

User can be a person or service.

#### **IAM Groups**

Groups are collections of users, and policies are attached to them. It is used to assign permissions to users.

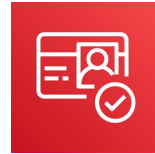
#### **IAM Roles**

IAM users or AWS services can assume a role to obtain temporary security credentials to make AWS API calls.

# 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 allows customers to sign in through social identity providers such as Google, Facebook, and Amazon, and through enterprise identity providers such as Microsoft Active Directory via

CAMM

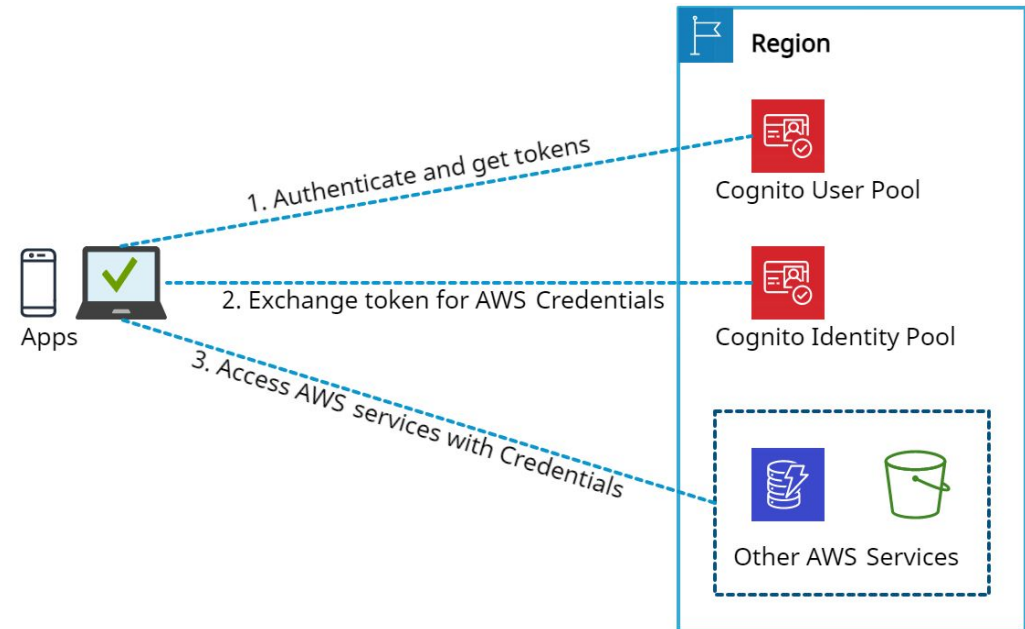
**The two main components of Amazon Cognito are as follows:**



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

**Identity pools** are user repositories of an account, which provide temporary and limited-permission AWS credentials to the users so that they can access other AWS resources without re-entering their credentials.

- Amazon Cognito User Pools is a standards-based Identity Provider and supports OAuth 2.0, SAML 2.0, and OpenID Connect. Amazon Cognito identity pools are useful for both authenticated and unauthenticated identities.
- Amazon Cognito is capable enough to allow usage of user pools and identity pools separately or together

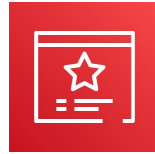




## AWS Certificate Manager

### What is AWS Certificate Manager?

AWS Certificate Manager is a service that allows a user to protect AWS applications by storing, renewing, and deploying public and private SSL/TLS X.509 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.
- It centrally manages the certificate lifecycle and helps to automate certificate renewals.
- SSL/TLS certificates provide data-in-transit security and authorize the identity of sites and connections between browsers and applications.
- 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 private CA operation and the private certificates issued.

### *The types of SSL certificates are:*

#### **Extended Validation Certificates (EV SSL)**

Most expensive SSL certificate type

#### **Organization Validated Certificates (OV SSL)**

Validates a business' creditably.

#### **Domain Validated Certificates (DV SSL)**

Provides 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:*

#### **AWS Certificate Manager (ACM)**

Useful for customers who need a secure and public web presence.

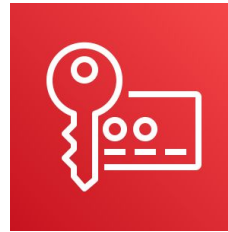
#### **ACM Private CA**

Useful for customers that are intended for private use within an organization.

# AWS Key Management Service

## What is AWS Key Management Service?

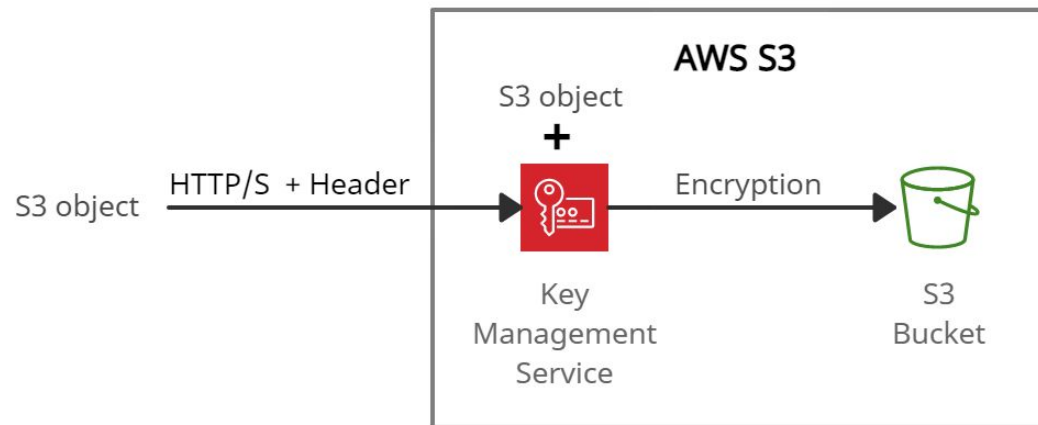
AWS Key Management Service is a global service that creates, stores, and manages encryption keys.



## AWS Key Management Service

- ❑ Provides data security at rest using encryption keys and provides access control for encryption, decryption, and re-encryption.
- ❑ Offers SDKs for different languages to add digital signature capability in the application code.
- ❑ AWS KMS produces new cryptographic data for the KMS key once a year, when automatic key rotation is turned on for a KMS key.
- ❑ AWS KMS preserves all previous iterations of the cryptographic information so that you can decrypt any data that has been encrypted using that KMS key. Until the KMS key is deleted, AWS KMS does not remove any rotated key material.

### Encryption using AWS KMS



### Customer Managed CMKs:

The CMKs created, managed, and used by users are termed as Customer managed CMKs and support cryptographic operations.

### AWS Managed CMKs:

The CMKs created, managed, and used by AWS services on the user's behalf are termed AWS-managed CMKs.



## AWS Secrets Manager

### What is AWS Secrets Manager?

AWS Secrets Manager is a service that prevents secret credentials from being hardcoded in the source code.



### AWS Secrets Manager

#### AWS Secrets Manager:

- Ensures in-transit encryption of the secret between AWS and the system to retrieve the secret.
- Rotates credentials for AWS services using the Lambda function that instructs Secrets Manager to interact with the service or database.
- Stores the encrypted secret value in SecretString or SecretBinary field.
- Uses open-source client components to cache secrets and updates them when there is a need for rotation.

#### Secrets Manager can be accessed using the following ways:

- AWS Management Console
  - AWS Command Line Tools
  - AWS SDKs
  - HTTPS Query API
- 
- It provides security and compliance facilities by rotating secrets safely without the need for code deployment.
  - It integrates with AWS CloudTrail and AWS CloudWatch to log and monitor services for centralized auditing.
  - It integrates with AWS Config and facilitates tracking of changes in Secrets Manager.

#### Secret rotation is supported with the below Databases:

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

### What is AWS STS?

AWS STS stands for "AWS Security Token Service." It is a web service provided by Amazon Web Services (AWS) that enables you to request temporary, limited-privilege credentials for accessing AWS resources. STS is a crucial component of AWS identity and access management (IAM) services and is designed to enhance the security of your AWS environment.

#### Features:

- ❑ **Temporary Credentials:** AWS STS issues short-lived security credentials, which typically consist of an Access Key ID, a Secret Access Key, and a Session Token. These credentials are valid for a specified duration, usually ranging from a few minutes to a few hours.
- ❑ **Identity Federation:** STS allows you to integrate with identity providers, such as AWS Identity and Access Management (IAM), Active Directory, or SAML-based identity providers. This enables you to grant temporary AWS access to users and services from external identity sources.
- ❑ **AssumeRole:** One of the most commonly used operations in AWS STS is "AssumeRole." It lets you grant permission to an IAM user or an AWS service to assume a role temporarily. This role can have different permissions and access controls compared to the original entity requesting access.
- ❑ **Cross-Account Access:** AWS STS facilitates cross-account access, enabling you to allow one AWS account to access resources in another AWS account securely.
- ❑ **Fine-Grained Permissions:** By using STS, you can implement fine-grained permissions, restricting what actions and resources a user or application can access during their temporary session.
- ❑ **Temporary Session Tokens:** STS issues session tokens alongside temporary credentials, allowing users or applications to access AWS services by presenting these tokens instead of long-term credentials. This reduces the risk associated with credential exposure.
- ❑ **Multi-Factor Authentication (MFA):** AWS STS can be used in conjunction with MFA to provide an extra layer of security when requesting temporary credentials.
- ❑ **AWS CLI and SDK Integration:** AWS CLI (Command Line Interface) and AWS SDKs (Software Development Kits) have built-in support for AWS STS, making it easy to request and use temporary credentials programmatically.

## AWS STS

### Best Practices:

- Imagine you have two AWS accounts: Account A and Account B. You want to allow an IAM user in Account A to access an S3 bucket in Account B without sharing long-term credentials. You can use AWS STS to accomplish this.
- You have a web application running on an Amazon EC2 instance that needs to access an Amazon S3 bucket securely. Instead of storing long-term credentials on the EC2 instance, you can use AWS STS to grant temporary access to the S3 bucket.

### Pricing:

- AWS STS itself does not have any additional charges. However, if you use it with other AWS services, you will be charged for the other services. For example, if you use STS to grant permissions to an application to write data to an Amazon S3 bucket, you'll be charged for the S3 usage.

### When to use AWS STS?

Use AWS STS when you need to enhance security, delegate permissions, or provide temporary, controlled access to AWS resources for users, applications, or services in a flexible and granular manner. It helps you follow security best practices and reduce the reliance on long-lived credentials, improving overall security posture in your AWS environment.

### Several API operations in AWS STS:

AWS STS provides several API operations that allow you to manage temporary security credentials and perform various identity and access management tasks. Here are some of the key AWS STS API operations:

**AssumeRole**

**AssumeRoleWithSAML**

**AssumeRoleWithWebIdentity**

**GetSessionToken**

**DecodeAuthorizationMessage**

**GetCallerIdentity**

### When not to use AWS STS?

AWS STS provides additional security but may incur additional operational costs so if your requirement is simple and straightforward within a single AWS account, IAM users and roles can suffice. Also, you can avoid AWS STS if you looking for long-term access to AWS resources.

## What is AWS Private CA?

AWS Private CA (Certificate Authority) is a service provided by Amazon Web Services (AWS) that allows you to create and manage a private certificate authority infrastructure within your AWS environment.

This service enables you to issue, revoke, and manage digital certificates for various use cases, such as securing communication between your applications and services, encrypting data in transit, and authenticating users and devices.

### Limitations

**Regional Scope:** AWS Private CA is region-specific. This means that a private CA you create in one AWS region is not directly accessible or usable in another AWS region.

**Certificate Chain Trust:** Certificates issued by an AWS Private CA are only trusted within your organization or the specific AWS environment where the CA is created.

**Domain Validation:** AWS Private CA may not provide domain validation or extended validation certificates that are typically required for SSL/TLS certificates used in publicly accessible websites.

**Lack of Browser Trust:** Certificates issued by AWS Private CA may not be trusted by web browsers and other publicly available software by default, as they are not part of the publicly trusted certificate authority list.

### Features

- **Private Certificate Authority:** AWS Private CA allows you to create a private CA that is not publicly trusted. This means the certificates issued by your private CA are only trusted within your organization or within the AWS environment.
- **Custom Root CA:** You can create your own root certificate authority, giving you full control over the certificate hierarchy and trust chain.
- **Flexible Certificate Management:** AWS Private CA provides a range of APIs and tools for managing certificates, including issuing, revoking, and renewing them.
- **Integration with AWS Services:** AWS Private CA can be seamlessly integrated with various AWS services like Amazon EC2, AWS Elastic Load Balancer, AWS IoT, and more.
- **Enhanced Security:** You can set policies to control certificate issuance and revocation, ensuring that only authorized entities can request and use certificates.
- **Certificate Revocation:** In case of compromised or no longer needed certificates, you can easily revoke them to enhance security.

### Use Cases

1. **Securing Internal APIs and Microservices:** You have a microservices architecture with multiple APIs communicating with each other internally. You want to secure these communication channels to prevent eavesdropping and unauthorized access.
2. **Device Authentication for IoT:** You are building an IoT (Internet of Things) application, and you need a way to authenticate and secure communication between IoT devices and your cloud infrastructure.
3. **Internal Email Encryption:** Your organization needs to ensure the confidentiality and integrity of internal email communications.

### Pricing

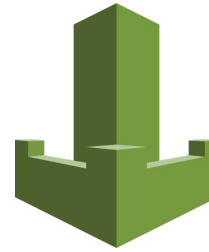
Your AWS account incurs monthly fees for each private CA created, starting from the creation date. Additionally, you'll be charged for every certificate you issue. This charge applies to certificates whether you export them from ACM, create them through the AWS Private CA API, or use the AWS Private CA CLI.

There are no charges for a private CA once it's deleted. However, if you decide to restore a previously deleted private CA, you'll be billed for the period between deletion and restoration.

## What is AWS WAF?

AWS WAF is a web application firewall that helps protect web applications from common web exploits and attacks.

It acts as a protective shield for your web applications, helping you safeguard them from threats like SQL injection, cross-site scripting (XSS), and other malicious activities.



### Features:

- ❑ **Web Traffic Filtering:** AWS WAF allows you to filter and inspect web traffic coming to your applications. You can set up rules to allow, block, or monitor traffic based on various criteria, such as IP addresses, HTTP headers, request methods, and query strings.
- ❑ **Protection Against Common Attacks:** It protects a wide range of common web attacks, including SQL injection, XSS, and cross-site request forgery (CSRF).
- ❑ **Custom Rules:** You can create custom rules to address specific security requirements and business logic.
- ❑ **AWS WAF Managed Rules for AWS Organizations:** This feature allows you to centrally manage and deploy WAF rules across multiple AWS accounts within an AWS Organization.

### Pricing:

AWS WAF costs depend on the quantity of web access control lists (web ACLs) you establish, the number of rules incorporated into each web ACL, and the volume of web requests you receive.

No prior commitments are required. It's important to note that AWS WAF charges are separate from Amazon CloudFront pricing, AWS Cognito pricing, Application Load Balancer (ALB) pricing, Amazon API Gateway pricing, and AWS AppSync pricing.

**Best Practices:**

- ❖ Combine AWS WAF with other AWS services such as AWS Shield (for DDoS protection) and Amazon CloudFront (for content delivery) to create a robust, multi-layered security strategy.
- ❖ If you're using AWS Managed Rule Sets, ensure that you keep them up to date. AWS regularly updates these rule sets to protect against emerging threats.
- ❖ Enable logging for AWS WAF to capture detailed information about web requests and potential threats. Use Amazon CloudWatch or a SIEM solution to monitor and analyze these logs.
- ❖ Implement rate-limiting rules to protect APIs from abuse and DDoS attacks. Set appropriate rate limits based on expected traffic patterns.
- ❖ Tailor your web access control lists (web ACLs) to the specific needs of your application.
- ❖ Periodically review your AWS WAF rules to make adjustments based on changing application requirements and emerging threats.



# Front-End Web and Mobile



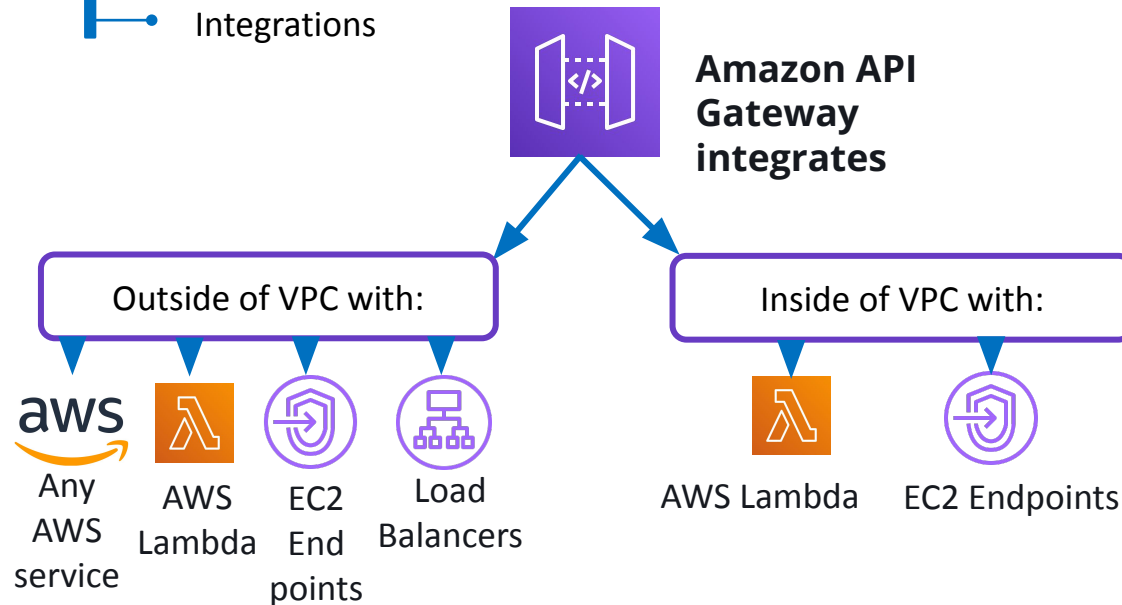
# Amazon API Gateway

## What is Amazon API Gateway?

Amazon API Gateway is a service that maintains and secures APIs at any scale. It is categorized as a serverless service of AWS.

### API Gateway consists of:

- Stages
- Resources
- Methods
- Integrations

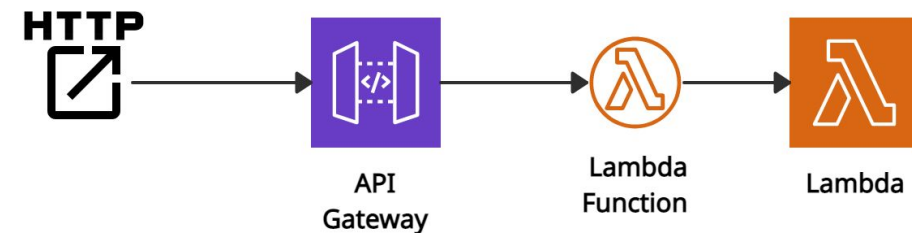


### Amazon API Gateway:

- ✓ Acts as a front door for real-world applications to access data, business logic from the back-end services, such as code running on AWS Lambda, or any web application.
- ✓ Handles the processing of hundreds of thousands of existing API calls, including authorization, access control, different environments (dev, test, production), and API version management.
- ✓ Helps to create web APIs that route HTTP requests to Lambda functions

### Example:

When a request is sent through a browser or HTTP client to the public endpoint, API Gateway API broadcasts the request and sends it to the Lambda function. The Function calls the Lambda API to get the required data and returns it to the API.



*AWS Lambda + API Gateway = No need to manage infrastructure*

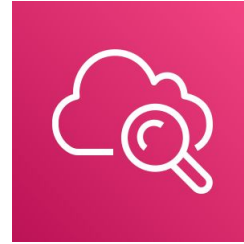


# Management and Governance

## Amazon CloudWatch

### What is Amazon CloudWatch?

Amazon CloudWatch is a service that monitors based on multiple metrics of AWS and on-premises resources.



### *Amazon CloudWatch*

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

- ❑ Collects and correlates monitoring data in logs, metrics, and events from AWS resources, applications, and services that run on AWS and on-premises servers.
- ❑ Offers dashboards and creates graphs to visualize cloud resources.
- ❑ Visualizes logs to address issues and improve performance by performing queries.

- ❑ Alarms can be created using CloudWatch Alarms that monitors metrics and send notifications.
- ❑ CloudWatch Agent or API can be used to monitor hybrid cloud architectures.
- ❑ CloudWatch Container Insights and Lambda Insights both provide dashboards to summarize the performance and errors for a selected time window.

Amazon CloudWatch is used alongside the following applications:

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

## Amazon CloudWatch Logs

### What is Amazon CloudWatch Logs?

Amazon CloudWatch Logs is a service provided by Amazon Web Services (AWS) that enables you to monitor, store, and access log data from various AWS resources and applications. It is designed to help you centralize and gain insights from logs generated by your AWS resources, applications, and services in a scalable and cost-effective manner.

#### Use Cases:

- **Application Debugging:** Developers want to troubleshoot and debug issues in a microservices-based application.
- **Cost Monitoring for EC2 Instances:** An organization wants to track and control costs associated with their Amazon EC2 instances.
- **Security and Compliance Auditing:** A company needs to monitor and audit user activities across its AWS environment to ensure compliance with security policies.

#### → Features

- ❖ **Log Collection:** CloudWatch Logs allows you to collect log data from a wide range of AWS resources and services, including Amazon EC2 instances, Lambda functions, AWS CloudTrail, AWS Elastic Beanstalk, and custom applications running on AWS or on-premises.
- ❖ **Log Storage:** It provides a secure and durable repository for your log data.
- ❖ **Real-time Monitoring:** You can set up CloudWatch Alarms to monitor log data in real time and trigger notifications or automated actions when specific log events or patterns are detected.
- ❖ **Log Queries:** CloudWatch Logs Insights allows you to run ad-hoc queries on your log data to extract valuable information and troubleshoot issues. You can use a simple query language to filter and analyze logs.
- ❖ **Log Retention:** You can define retention policies for your log data, specifying how long you want to retain logs before they are automatically archived or deleted. This helps in cost management and compliance with data retention policies.
- ❖ **Log Streams:** Within a log group, log data is organized into log streams, which represent individual sources of log data. This organization makes it easy to distinguish between different sources of log data.

## Amazon CloudWatch Logs

### Pricing

Amazon CloudWatch operates on a pay-as-you-go model, meaning there are no initial obligations or minimum charges. You are billed based on your actual usage, with charges calculated and billed at the conclusion of each month.

**CloudWatch Logs offer two distinct tiers namely Free Tier and Paid Tier.**

### Best Practices:

- **Log Structure:** Design your log messages with a consistent and meaningful structure. Use JSON or key-value pairs to include relevant information such as timestamps, log levels, and context.
- **Log Events Filtering:** Use log event filters to extract valuable information from log data. You can create metric filters or use CloudWatch Logs Insights for more advanced log queries.
- **Use CloudWatch Metrics:** Integrate CloudWatch Logs with CloudWatch Metrics to gain insights into log data trends and visualize log-related metrics in CloudWatch Dashboards.

### → Limitations

- ❖ **Query Execution Time:** When using CloudWatch Logs Insights to query log data, there is a maximum query execution time limit. Complex queries or queries over a large dataset may time out.
- ❖ **Data Exports:** While you can export log data to Amazon S3 or other destinations, there are limitations on the frequency of exports and the destinations you can use.
- ❖ **Data Structure:** CloudWatch Logs is designed primarily for unstructured log data. If you have structured data, you may need to parse it using CloudWatch Logs Insights to make it more accessible.
- ❖ **API Limitations:** The CloudWatch Logs API has rate limits on the number of requests you can make, and these limits vary based on your AWS account and region.
- ❖ **Log Streams:** Each log stream within a log group must have a unique name, which can make it challenging to manage log streams for resources that generate a large number of logs.
- ❖ **Log Data Size:** There are limits on the size of individual log events and log batches, which may require you to segment and compress log data if it exceeds these limits.

### What is AWS AppConfig?

AWS AppConfig is a service that helps you manage and deploy application configurations. It enables you to create, manage, and deploy application configurations in a controlled and organized manner, making it easier to maintain and update your applications while ensuring consistency and compliance.

#### Limitations

**Resource Limitations:** AWS AppConfig might have resource limitations such as a maximum number of applications, environments, configurations, and deployments you can manage.

**Custom Rollback:** While you can define deployment strategies and control how configurations are deployed, rollback mechanisms are not fully automated.

**Data Size Limits:** There are limits on the size of configuration data you can store in AWS AppConfig. Be aware of these limits when managing large or complex configurations.

#### Features

- ❖ **Configuration Management:** AWS AppConfig allows you to store and manage configuration data separately from your application code.
- ❖ **Versioning:** You can create and manage different versions of your configurations.
- ❖ **Deployment Strategy:** AWS AppConfig supports various deployment strategies, including rolling deployments, allowing you to control how configuration changes are rolled out to your application instances.
- ❖ **Validation:** You can use AWS AppConfig to validate configurations before deploying them.
- ❖ **Integration with Other AWS Services:** You can integrate AWS AppConfig with other AWS services like AWS Systems Manager and AWS CloudFormation to automate application configurations.
- ❖ **Customizable Workflow:** You can set up your own workflow for configuration deployment, and you have the flexibility to define rules for when and how configurations are applied.

#### Pricing

AWS AppConfig pricing is typically based on the number of API requests you make to the service and the number of configuration items you store.

## How is AppConfig different from AWS Beanstalk?

Feature	AWS AppConfig	AWS Elastic Beanstalk
<b>Purpose</b>	Configuration management and deployment	Full-stack application deployment and management
<b>Deployment Focus</b>	Configuration settings	Application code and infrastructure
<b>Management</b>	Configuration-centric	Application-centric
<b>Deployment Complexity</b>	Relatively straightforward	Abstracts infrastructure complexity
<b>Control</b>	Greater control over configurations	Limited control over underlying infrastructure
<b>Integration</b>	Often used with other AWS services for dynamic updates	Standalone PaaS platform

## Best Practices

- Avoid Hardcoding:** Never hardcode configuration values directly into the code. Instead, use environment variables, configuration files, or external services.
- Sensitive Data Protection:** Encrypt sensitive data like API keys, database passwords, and other secrets. Tools like AWS Secrets Manager or HashiCorp Vault can manage and rotate secrets.
- Use AWS Parameter Store or Secrets Manager:** For sensitive or secret data within your configurations, consider using AWS Systems Manager Parameter Store or AWS Secrets Manager.
- Configuration Change Auditing:** Set up logging and auditing for configuration changes. AWS CloudTrail can be used to track and log configuration changes made through AWS AppConfig.

## Use Case

**Localization and Language Settings:** Manage localization and language settings for a multi-language application. AWS AppConfig can be used to store and control language-specific settings, such as text translations, date formats, and currency symbols. This makes it easy to update language configurations as your application expands to new markets.

**Email Notification Templates:** Manage email notification templates for your application's automated notifications, such as registration confirmations, password resets, or order confirmations. AWS AppConfig can store the email templates, allowing you to update the email content and formatting without code changes.



# AWS CloudFormation

## What is AWS CloudFormation?

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



**AWS CloudFormation**

### Stacks:

- ❑ **Stacks** can be created using the AWS CloudFormation console and AWS Command Line Interface (CLI).
- ❑ **Nested Stacks** are stacks created within another stack by using the 'AWS::CloudFormation::Stack' resource attribute.
- ❑ 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 does not charge for using AWS CloudFormation, and charges are applied for the CloudFormation template services.



### Template:

- ❑ A **template** is used to create, update, and delete an entire stack as a single unit without managing resources individually.
- ❑ CloudFormation provides the capability to reuse the template to set the resources easily and repeatedly.

### 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 a service that gets enabled when the AWS account is created and is used to enable compliance and auditing of the AWS account.



```
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"
  }}
}]}
```

- ✓ It offers to view, analyze, and respond to activity across the AWS infrastructure.
- ✓ It records actions as an event by an IAM user, role, or an AWS service.
- ✓ CloudTrail records can download Cloud Trail events in JSON or CSV file.
- ✓ **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.
- ✓ **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.

## AWS Systems Manager

### What is AWS Systems Manager?

AWS Systems Manager (SSM) is a service that allows users to centralize or group operational data using multiple services and automate operations across AWS infrastructure.

- ✓ It simplifies maintenance and identifies issues in the resources that may impact the applications.
- ✓ It displays the operational data, system and application configurations, software installations, and other details on a single dashboard known as AWS Systems Manager Explorer.
- ✓ It manages secrets and configuration data and separates them from code using a centralized store known as Parameter Store.
- ✓ It helps to communicate with the Systems Manager agent installed on AWS servers and in an on-premises environment. Agents are installed to manage resources on servers using different operating systems.



It helps to manage servers without actually logging into the server using a web console known as Session Manager.



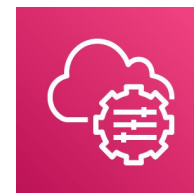
It helps to automate repetitive operations and management tasks using predefined playbooks.



It connects with Jira Service Desk and ServiceNow to allow ITSM platform users to manage AWS resources.



Systems Manager Distributor helps to distribute software packages on hosts along with versioning.





# Networking and Content Delivery

## Amazon VPC

### What is Amazon VPC?

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

**Private subnet** - A subnet that does not have internet access is termed a private subnet.

**Public subnet** - A subnet that has internet access is termed a public subnet.

**VPN only subnet** - A subnet that does not have internet access but has access to the virtual private gateway for a VPN connection is termed a VPN-only subnet.

- ❑ It includes many components such as Internet gateways, VPN tools, CIDR, Subnets, Route tables, VPC endpoint, NAT instances, Bastion servers, Peering Connection, and others.
- ❑ It spans across multiple Availability Zones (AZs) within a region.
- ❑ The first four IP and last one IP addresses are reserved per subnet.
- ❑ It creates a public subnet for web servers that uses internet access and a private subnet for backend systems, such as databases or application servers.
- ❑ It can monitor resources using Amazon CloudWatch and Auto Scaling Groups.

- ❖ Every EC2 instance is launched within a default VPC with equal security and control like normal Amazon VPC. Default VPC has no private subnet.
- ❖ It uses Security Groups and NACL (Network Access Control Lists) for multi-layer security.
- ❖ Security Groups (stateful) provide instance-level security, whereas NACLs (stateless) provide subnet-level security.
- ❖ VPC sharing is a component that allows subnets to share with other AWS accounts within the same AWS Organization.

# 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 makes use of Edge locations (worldwide network of data centers) to deliver the content faster.
- Without edge locations, it retrieves data from an origin such as an Amazon S3 bucket, a Media Package channel, or an HTTP server.

### CloudFront provides some security features such as:

- ❖ **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 Route 53** - Against more complex attacks than DDoS.

CloudFront is integrated with AWS Services such as:

- Amazon S3
- Amazon EC2
- Elastic Load Balancing
- Amazon Route 53
- AWS Essential Media Services

## 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):

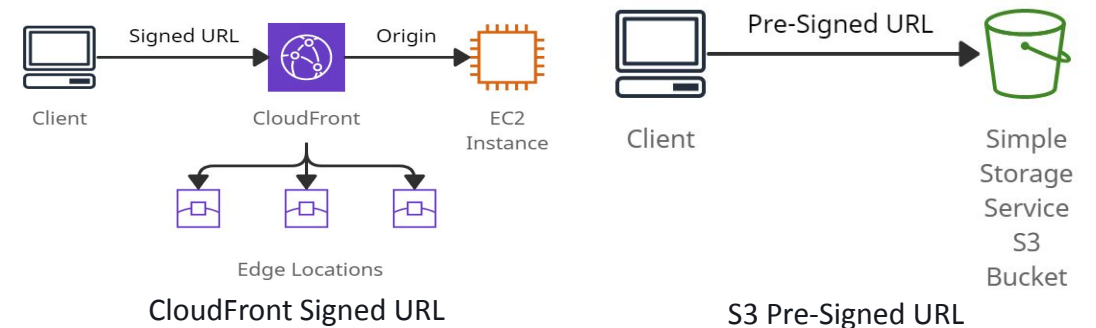
- Outside access is restricted using signed URLs and signed cookies, but what if someone tries to access objects using Amazon S3 URL, bypassing CloudFront signed URL and signed cookies. To restrict that, OAI is used.
- Use OAI as a special CloudFront user and associate it with your CloudFront distribution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

### CloudFront Signed URL:

- It allows access to a path, no matter what is the origin
- It can be filtered by IP, path, date, expiration
- It leverages caching features

### S3 Pre-Signed URL:

- It issues a request as the person who pre-signed the URL.



# Amazon Route 53

## What is Route 53?

Route 53 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.



- 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

- **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 Hosted Zone** - Determines how traffic is routed on the Internet.
  - **Private Hosted Zone** - Determines how traffic is routed within VPC.

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

## 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.

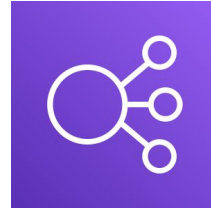


Amazon Route 53

## Elastic Load Balancing (ELB)

### What is Elastic Load Balancing?

Elastic Load Balancing is a managed service that allows traffic to get distributed across EC2 instances, containers, and virtual appliances as target groups.



*Elastic Load Balancing*

### Elastic Load Balancer types are as follows:

#### Application Load Balancer:

- Routes HTTP and HTTPS traffic at layer 7.
- Offers path-based routing, host-based routing, query-string, parameter-based routing, and source IP address-based routing.

#### Network Load Balancer:

- Routes TCP, UDP, and TLS traffic at layer 4.
- Suitable for high-performance and low latency applications.

#### Gateway Load Balancer:

- Suitable for third-party networking appliances.
- It simplifies tasks to manage, scale, and deploy virtual appliances

- ELB integrates with every AWS service throughout the applications.
- It is tightly integrated with Amazon EC2, Amazon ECS/EKS.
- ELB integrates with Amazon VPC and AWS WAF to offer extra security features to the applications.
- It helps monitor the servers' health and performance in real-time using Amazon CloudWatch metrics and request tracing.
- ELB can be placed based on the following aspects:
  - Internet-facing ELB:
    - Load Balancers have public IPs.
  - Internal only ELB:
    - Load Balancers have private IPs.
- ELB offers the functionality of Sticky sessions. It is a process to route requests to the same target from the same client.



# Storage



## Amazon Simple Storage Service (S3)

### What is Amazon Simple Storage Service?

Amazon S3 is a simple service used to provide key-based object storage across multiple availability zones (AZs) in a specific region.

- S3 is a global service with region-specific buckets.
- It is also termed a static website hosting service.
- It provides 99.99999999% (11 9's) of content durability.
- S3 offers strong read-after-write consistency for any object.
- Objects (files) are stored in a region-specific container known as Bucket.
- Objects that are stored can range from 0 bytes - 5TB.

- It provides 'Multipart upload' features that upload objects in parts, suitable for 100 MB or larger objects.
- It offers to choose 'Versioning' features to retain multiple versions of objects, must enable versioning at both source and destination.
- Amazon S3 Transfer Acceleration allows fast and secure transfer of objects over long distances with minimum latency using Amazon CloudFront's Edge Locations.
- Amazon S3 uses access control lists (ACL) to control access to the objects and buckets.
- Amazon S3 provides Cross-Account access to the objects and buckets by assuming a role with specified privileges.



*Amazon S3*

Amazon S3 uses the following ways for security:

#### User-based security

- IAM policies

#### Resource-Based

- Bucket Policies
- Bucket Access Control List (ACL)
- Object Access Control List (ACL)

Amazon S3 provides the following storage classes used to maintain the integrity of the objects:

- S3 Standard** - offers frequent data access.
- S3 Intelligent-Tiering** - automatically transfer data to other cost-effective access tiers.
- S3 Standard-IA** - offers immediate and infrequent data access.
- S3 One Zone-IA** - infrequent data access.
- S3 Glacier** - long-term archive data, cheap data retrieval.
- S3 Glacier Deep Archive** - used for long-term retention.

Amazon S3 offers to choose from the following ways to replicate objects:

- Cross-Region Replication - used to replicate objects in different AWS Regions.
- Same Region Replication - used to replicate objects in the same AWS Region.

## Amazon Elastic Block Store

### What is Amazon Elastic Block Store?

Amazon Elastic Block Store is a service that provides the block-level storage drive to store persistent data.



- ❖ Multiple EBS volumes can be attached to a single EC2 instance in the same availability zone.
- ❖ A single EBS volume can not be attached to multiple EC2 instances.
- ❖ Amazon EBS Multi-Attach is a feature used to attach a single Provisioned IOPS SSD (io1 or io2) volume to multiple instances in the same Availability Zone.
- ❖ EBS volumes persist independently after getting attached to an instance, which means the data will not be erased even if it terminates.
- ❖ By default, the root EBS volume gets terminated when the instance is terminated.

By default, the non-root EBS volume does not get affected when the instance is terminated.

Amazon EBS can be attached and detached to an instance and can be reattached to other EC2 instances.

Amazon EBS easily scales up to petabytes of data storage.

Amazon EBS volumes are best suited for database servers with high reads and write and throughput-intensive workloads with continuous reads and write.

Amazon EBS uses AWS KMS service with AES-256 algorithm to support encryption.

Amazon EBS offers point-in-time snapshots for volumes to migrate to other AZs or regions.

EBS snapshots are region-specific and are incrementally stored in Amazon S3.

## Amazon Elastic Block Store

***EBS volumes types are as follows:***

### **SSD (Solid-state drives)**

*General Purpose SSD:*

- Useful for low-latency applications, development, and test environments.
- Supports volume size from 1 GiB to 16 TiB.
- Allows 16,000 as maximum IOPS per volume.
- Allows 1000 MiB/s as maximum throughput per volume.

### **Provisioned IOPS SSD:**

- Useful for I/O-intensive database workloads and provide sub-millisecond latency.
- Supports volume size from 4 GiB to 64 TiB.
- Allows 256,000 as maximum IOPS per volume.
- Allows 4,000 MiB/s as maximum throughput per volume.
- The multi-Attach feature is supported for io1 and io2

### **HDD (Hard disk drives)**

*Throughput Optimized HDD:*

- Useful for Big data and Log processing workloads.
- Supports volume size from 125 GiB to 16 TiB.
- Allows 500 as maximum IOPS per volume.
- Allows 500 MiB/s as maximum throughput per volume.

### **Cold HDD:**

- Useful for infrequently accessed data and lowest cost workloads.
- Supports volume size from 125 GiB to 16 TiB.
- Allows 250 as maximum IOPS per volume.
- Allows 250 MiB/s as maximum throughput per volume.

## Amazon Elastic File System (EFS)

### What is Amazon Elastic File System?

Amazon Elastic File System is a managed service used to create and scale file storage systems for AWS and on-premises resources.



**Amazon EFS**

It offers the following storage classes for file storage:

- EFS Standard storage class
- EFS Infrequent Access storage class - can store less frequently accessed files.

It offers the following modes to ease the file storage system:

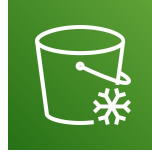
- Performance modes -
  - General Purpose performance mode: Useful for low-latency workloads.
  - Max I/O mode: High throughput workloads.
- Throughput modes -
  - Bursting Throughput mode: Throughput increases based on the file system storage.
  - Provisioned Throughput mode: Throughput changes are independent of the file system storage.
- It provides EFS lifecycle management policies based on the number of days ranges from 7-90 days to automatically move files from Standard storage class to EFS IA storage class.

- It spans multiple availability zones and regions.
- It uses EFS Mount Target to share a file system with multiple availability zones and VPCs.
- It is best suited for Linux-based workloads and applications.
- Multiple instances can access it at the same time leads to high throughput and low latency IOPS.
- It automatically scales storage capacity up to petabyte.
- It supports file locking and strong data consistency.
- It offers data encryption at rest and in-transit using AWS KMS and TLS, respectively.
- It uses POSIX permissions to control access to files and directories.

# 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.

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.

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

A vault is a place for storing archives with a unique address.

Amazon S3 Glacier jobs are the select queries that execute to retrieve archived data. It uses Amazon SNS to notify when the jobs complete.

Amazon S3 Glacier does not provide real-time data retrieval of the archives.

Amazon S3 Glacier uses 'S3 Glacier Select' to query archive objects in uncompressed CSV format and store the output to the S3 bucket.

Amazon S3 Glacier Select uses common SQL statements like SELECT, FROM, and WHERE.

It offers only SSE-KMS and SSE-S3 encryption.

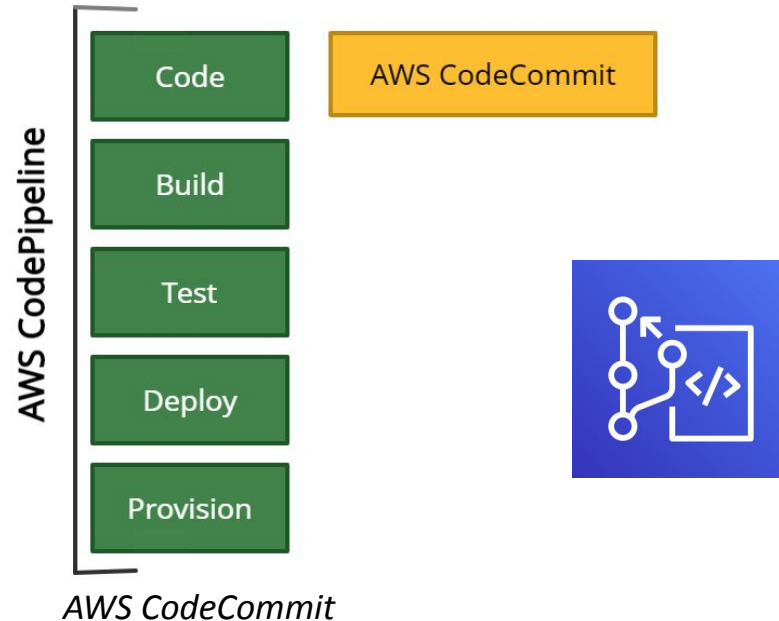


# Developer Tools

# AWS CodeCommit







## What is AWS CodeCommit?

AWS CodeCommit is a managed source control service used to store and manage private repositories in the AWS cloud, such as Git.



## Functions of AWS CodeCommit:

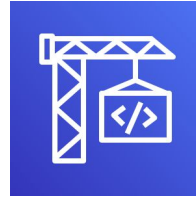


-  It works with existing Git-based repositories, tools, and commands in addition to AWS CLI commands and APIs.
-  It provides high availability, durability, and redundancy.
-  It eliminates the need to back up and scale the source control servers.
-  CodeCommit repositories support pull requests, version differencing, merge requests between branches, and notifications through emails about any code changes.
-  As compared to Amazon S3 versioning of individual files, AWS CodeCommit support tracking batched changes across multiple files.
-  It provides encryption at rest and in transit for the files in the repositories.

# AWS CodeBuild

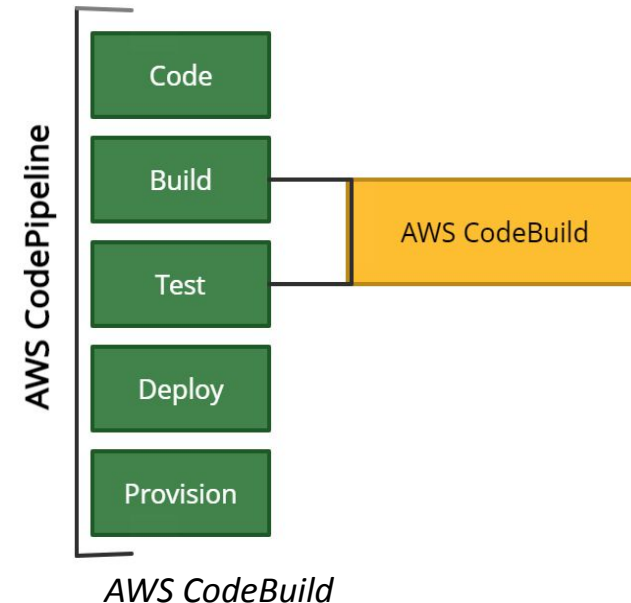
## What is AWS CodeBuild?

AWS CodeBuild is a continuous integration service in the cloud used to compile source code, run tests, and build packages for deployment.



- ❑ AWS Code Services family consists of AWS CodeBuild, AWS CodeCommit, AWS CodeDeploy, and AWS CodePipeline that provide complete and automated continuous integration and delivery (CI/CD).
- ❑ It provides prepackaged and customized build environments for many programming languages and tools.
- ❑ It scales automatically to process multiple separate builds concurrently.
- ❑ It can be used as a build or test stage of a pipeline in AWS CodePipeline.
- ❑ It requires VPC ID, VPC subnet IDs, and VPC security group IDs to access resources in a VPC to perform build or test.

- ❑ Charges are applied based on the amount of time taken by AWS CodeBuild to complete the build.
- ❑ The following ways are used to run CodeBuild:
  - AWS CodeBuild
  - AWS CodePipeline console
  - AWS Command Line Interface (AWS CLI)
  - AWS SDKs

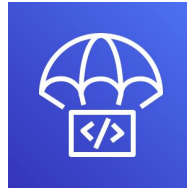




# AWS CodeDeploy

## What is AWS CodeDeploy?

AWS CodeDeploy is a service that helps to automate application deployments to a variety of compute services such as Amazon EC2, AWS Fargate, AWS ECS, and on-premises instances.



**AWS CodeDeploy**

*It provides the following deployment type to choose from:*

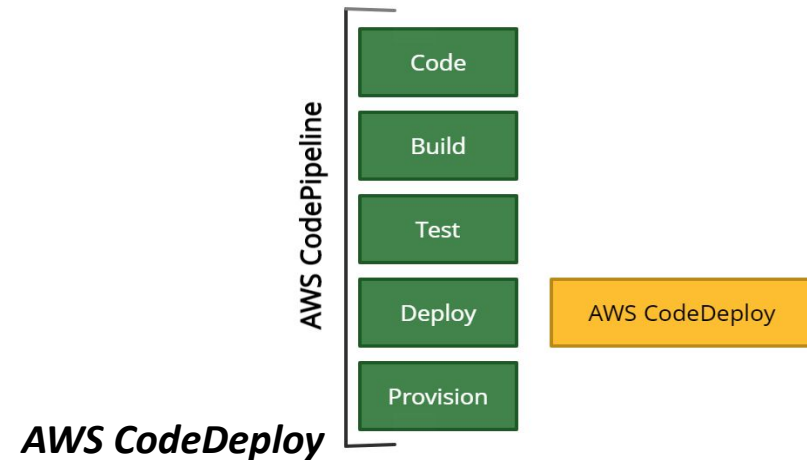
### In-place deployment:

- All the instances in the deployment group are stopped, updated with new revision and started again after the deployment is complete.
- Useful for EC2/On-premises compute platform.

### Blue/green deployment:

- The instances in the deployment group of the original environment are replaced by a new set of instances of the replacement environment.
- Using Elastic Load Balancer, traffic gets rerouted from the original environment to the replacement environment and instances of the original environment get terminated after the deployment is complete.
- Useful for EC2/On-Premises, AWS Lambda and Amazon ECS compute platform

- ❑ Using Amazon EKS, Kubernetes clusters and applications can be managed across hybrid environments without altering the code.
- ❑ It can fetch the content for deployment from Amazon S3 buckets, Bitbucket, or GitHub repositories.
- ❑ It can deploy different types of application content such as Code, Lambda functions, configuration files, scripts and even Multimedia files.
- ❑ It can scale with the infrastructure to deploy on multiple instances across development, test, and production environments.
- ❑ It can integrate with existing continuous delivery workflows such as AWS CodePipeline, GitHub, Jenkins.



**AWS CodeDeploy**

## AWS CodePipeline

### What is AWS CodePipeline?

AWS CodePipeline is a Continuous Integration(CI) and Continuous Delivery (CD) service by Amazon. It helps automate the build, test, and deployment phases of your software release process.

With AWS CodePipeline, you can create a workflow that automates the steps required to release your application, allowing you to deliver new features and updates more quickly and reliably.

### Limitations of AWS CodePipeline

**Complexity for Large Pipelines:** It is challenging to troubleshoot issues or understand the flow of your pipeline when dealing with more stages and intricate branching or parallelism.

**Limited Built-in Testing Capabilities:** While AWS CodePipeline integrates with various testing tools and services, it doesn't provide built-in testing capabilities.

**Limited Language and Framework Support:** If your preferred language or framework isn't supported by the available AWS services or third-party integrations, you may need to implement custom solutions.

### Features of AWS CodePipeline:

**Pipeline:** A pipeline in AWS CodePipeline is a series of stages and actions that define the steps your code must go through from source code to production deployment. Each stage can represent a different part of your CI/CD process, such as source code integration, building, testing, and deployment.

**Source Stage:** This is the first stage of a pipeline, where you specify the source code repository (e.g., AWS CodeCommit, GitHub, Amazon S3, etc.) that contains your application code. When changes are detected in the source repository, CodePipeline automatically triggers the pipeline.

**Build Stage:** In this stage, you can use AWS CodeBuild or another build tool to compile your source code, run tests, and generate deployable artifacts, such as executable files or container images.

**Test Stage:** You can integrate testing tools and frameworks in this stage to automatically test your application, ensuring that it meets the required quality standards. Common testing tools include AWS CodeBuild, AWS Device Farm, or third-party services.

**Deployment Stage:** This stage is responsible for deploying your application to various environments, such as development, testing, staging, and production. AWS CodePipeline supports deployment to different AWS services like AWS Elastic Beanstalk, AWS Lambda, Amazon ECS, or custom deployment targets.

**Approval Actions:** In some cases, you may want to introduce manual approval steps before promoting changes to production. AWS CodePipeline allows you to include approval actions, where designated individuals or teams can review and approve the changes before they proceed to the next stage.

**Notifications:** AWS CodePipeline can send notifications through Amazon SNS (Simple Notification Service) or other notification mechanisms to alert stakeholders about pipeline events and status changes.

**Integration with Other AWS Services:** AWS CodePipeline seamlessly integrates with various AWS services and tools, such as AWS CodeBuild, AWS CodeDeploy, AWS CodeCommit, AWS Elastic Beanstalk, AWS Lambda, and more, making it easy to build a comprehensive CI/CD pipeline in the AWS ecosystem.

## AWS CodePipeline

### Best Practices:

**Use Version Control:** Store your application code in a version control system (e.g., AWS CodeCommit, GitHub, GitLab, Bitbucket). This ensures code versioning, traceability, and collaboration among team members.

**Separate Environments:** Set up separate pipelines for different environments (e.g., development, testing, staging, production) and isolate them.

**Testing at Every Stage:** Incorporate automated testing at each stage of the pipeline. This includes unit tests, integration tests, and end-to-end tests. Failed tests should prevent the pipeline from progressing further.

**Parallelism and Concurrency:** Leverage parallelism to speed up your pipeline by running multiple actions concurrently, especially in the testing and deployment stages. However, be mindful of resource limitations and potential race conditions.

**Failure Handling:** Define what should happen when an action fails, whether it should be retried or escalated to a human for resolution.

### Use Case

**Web Application Deployment:** You have a web application hosted on AWS (e.g., AWS Elastic Beanstalk, Amazon S3 static website, or an EC2 instance), and you want to automate the deployment process.

**Serverless Application Deployment:** You're developing a serverless application using AWS Lambda, API Gateway, and other AWS services, and you want to automate the deployment process whenever changes are made to your code or infrastructure.

**Continuous Integration and Continuous Deployment for Containerized Applications:** You have a containerized application (e.g., Docker containers) and want to automate the building, testing, and deployment of containers to a container orchestration platform like Amazon ECS or Amazon EKS.

### Pricing

AWS CodePipeline has a flexible pay-as-you-go pricing model.

It costs \$1.00 per active pipeline per month, and there are no upfront fees. You get the first 30 days for free to encourage experimentation. An active pipeline is one that has been around for more than 30 days and had at least one code change go through in a month.

As part of the AWS Free Tier, you receive one free active pipeline monthly, which applies across all AWS regions.

**Note: Additional charges may apply for storing and accessing pipeline artifacts in Amazon S3, as well as for actions triggered by other AWS and third-party services integrated into your pipeline.**

# AWS Amplify

## What is AWS Amplify?

AWS Amplify is a set of tools and services provided by Amazon Web Services (AWS) that simplifies the process of building full-stack web and mobile applications. AWS Amplify is designed to accelerate development by providing a streamlined workflow for front-end and back-end development, as well as deployment and hosting.

### Limitations

- **Tight Integration with AWS Services:** While AWS Amplify is a great choice if you plan to use AWS services, it may not be the best option if you want to use non-AWS services extensively.
- **Limited Backend Customization:** AWS Amplify is designed to simplify backend development, but this simplicity comes at the cost of limiting your ability to customize the backend infrastructure extensively.
- **Limited Control over Resources:** While AWS Amplify abstracts away many operational tasks, it may not provide the level of control over resources that some applications require.
- **Complexity as the Project Grows:** AWS Amplify is excellent for getting started quickly, but as your application grows in complexity and scale, you may find it challenging to manage and optimize all aspects of your infrastructure.

### Pricing

AWS Amplify is not a separately priced service; it's a set of tools and services provided by AWS. Therefore, there is no specific cost associated with using AWS Amplify.

### Features

- ❖ **Authentication:** AWS Amplify provides pre-built authentication components and backend services for user sign-up, sign-in, and user management. It supports various authentication providers like Amazon Cognito, OAuth, and social identity providers (Facebook, Google, Apple ID).
- ❖ **API Integration:** AWS Amplify enables easy integration with various AWS services and APIs. Developers can create, manage, and interact with APIs, including GraphQL and REST, and perform operations like CRUD (Create, Read, Update, Delete) on data.
- ❖ **App Hosting:** Developers can deploy their web applications on AWS Amplify with features like continuous deployment, SSL certificates, and custom domains.
- ❖ **Offline Support:** AWS Amplify enables offline data access and synchronization for mobile and web applications, making it useful for scenarios where connectivity may be unreliable.
- ❖ **Analytics:** Built-in analytics tools help developers track user engagement and application performance.
- ❖ **Amplify CLI:** The AWS Amplify Command Line Interface (CLI) is a powerful tool for developers to configure and manage their Amplify applications, provision resources, and perform various tasks from the command line.

## AWS Amplify

### Best Practices

- ❖ **Use Version Control:** Always use version control (e.g., Git) to track changes to your AWS Amplify projects.
- ❖ **Modularize Your App:** Organize your application into smaller, manageable modules.
- ❖ **Infrastructure as Code:** Leverage the AWS Amplify CLI to define your infrastructure as code (IaC).
- ❖ **Environment Management:** Use environment variables and separate configurations for development, testing, and production environments.
- ❖ **Continuous Integration/Continuous Deployment (CI/CD):** Implement CI/CD pipelines to automate the building, testing, and deployment of your application.
- ❖ **Data Validation and Sanitization:** Always validate and sanitize user input to prevent security vulnerabilities like cross-site scripting (XSS) and SQL injection.

### Use Case

- ❖ **Personal Blog Website:** You want to create a personal blog website to publish articles and showcase your writing skills.
- ❖ **E-commerce Mobile App:** You're building an e-commerce mobile app that needs features like user registration, product catalog, and real-time updates for inventory changes.
- ❖ **Event Scheduling Web App:** You want to create a web application for scheduling and managing events for a local community organization.

## AWS CloudShell

### What is AWS CloudShell?

AWS CloudShell is a service provided by Amazon Web Services (AWS) that offers an interactive, browser-based shell environment that allows users to access a virtual machine (VM) with a pre-configured set of development tools and AWS CLI (Command Line Interface) pre-installed.

It's designed to make it easier for developers, system administrators, and other AWS users to manage their AWS resources and perform tasks without having to set up and configure their own development environment.

### Limitations of AWS CloudShell

**Limited Customization:** While AWS CloudShell comes with many pre-installed tools and utilities, it may not support all the software or configurations that users typically have in their local development environments.

**Resource Sharing:** AWS CloudShell environments are associated with individual AWS accounts and users. Sharing resources or collaborative development within AWS CloudShell may require additional setup and coordination.

### Features:

- ❑ **Pre-configured environment:** AWS CloudShell comes with a predefined set of commonly used tools and utilities for AWS development and management. This eliminates the need for users to install and configure these tools on their local machines.
- ❑ **AWS CLI integration:** AWS CloudShell includes the AWS CLI, which allows users to interact with AWS services and resources through the command line. Users can run AWS CLI commands directly within the CloudShell environment.
- ❑ **Persistent storage:** Each user is provided with a home directory in AWS CloudShell, which includes a persistent storage volume. This allows users to store files and scripts that can be accessed across sessions.
- ❑ **Secure environment:** AWS CloudShell is hosted within the AWS infrastructure and is integrated with AWS Identity and Access Management (IAM), ensuring security and access control. Users can access the environment using their AWS IAM credentials.
- ❑ **Browser-based access:** Users can access AWS CloudShell directly from the AWS Management Console, making it easy to switch between the console and the shell environment without the need for additional login credentials.



## AWS CloudShell

### Best Practices

- **Monitoring and Logging:** Set up CloudWatch Logs or other monitoring tools to capture logs and activities within AWS CloudShell. This helps in auditing user actions and troubleshooting issues.
- **Secure Data Handling:** Be cautious when handling sensitive data within CloudShell. Avoid storing confidential information, such as access keys, in scripts or files within your AWS CloudShell environment.
- **Resource Cleanup:** If you create AWS resources during your CloudShell session, remember to clean them up afterward to avoid incurring unnecessary costs.
- **Multiple Tabs:** Open multiple tabs within CloudShell to run multiple sessions concurrently.

### Use Case

**Quick Troubleshooting and Debugging:** You suspect an issue with one of your AWS resources, such as an EC2 instance or an S3 bucket, and need to troubleshoot it quickly.

**Managing AWS Resources from a Mobile Device:** You're on the go and need to perform routine AWS resource management tasks, such as stopping or starting EC2 instances or updating security groups.

**Script Automation and Integration:** You want to automate a repetitive task, such as regularly updating Route 53 DNS records based on changing IP addresses.

### Pricing

AWS CloudShell is provided at no extra cost. You are only charged for the other AWS resources used alongside AWS CloudShell for your application development. There is no minimum fee or obligatory upfront commitment. Data transfer costs are based on the regular AWS data transfer rates.

# AWS Cloud9

## What is AWS Cloud9?

AWS Cloud9 represents a cloud-hosted integrated development environment (IDE) offered by Amazon Web Services (AWS). It is designed to facilitate collaborative software development, making it easier for developers to write, debug, and deploy code in the cloud.

As AWS Cloud9 IDE is cloud-based it will let your code write, run, and debug within the browser itself.

It means no need to install any kind of IDE in your local machine.

### Limitations:

- The performance of your AWS Cloud9 environment is influenced by the size and configuration of the underlying EC2 instance, and you may need to pay for additional resources.
- While AWS Cloud9 does support customization through plugins, the extent of customization may not be as extensive as some desktop-based IDEs.

### Best Practices:

- **Resource Monitoring:** Keep an eye on resource usage, especially if you're using an EC2 instance for your AWS Cloud9 environment. Monitor CPU, memory, and storage to ensure you're not over-provisioning or running into performance issues.
- **Environment Cleanup:** When you're done with a development environment, terminate it to avoid incurring unnecessary charges. AWS CloudFormation can help automate environment creation and cleanup.

### Features

- ❖ **Cloud-Based IDE:** AWS Cloud9 is entirely cloud-based, which means you can access it from any device with an internet connection.
- ❖ **Code Collaboration:** AWS Cloud9 includes features for real-time collaboration among developers. Multiple team members can work on the same codebase simultaneously, making it easier to collaborate on projects.
- ❖ **Built-In Code Editor:** The IDE comes with a built-in code editor that supports popular programming languages such as Python, JavaScript, Java, and many others. It also provides code highlighting, autocompletion, and code formatting features.
- ❖ **Terminal Access:** Developers can access a fully functional terminal within the IDE, enabling them to run commands and manage their AWS resources directly from the same interface where they write code.
- ❖ **Integrated Debugger:** AWS Cloud9 includes debugging tools that help developers identify and fix issues in their code. This includes features like breakpoints, step-through debugging, and variable inspection.
- ❖ **Version Control Integration:** It supports integration with popular version control systems like Git, allowing developers to easily manage and track changes to their code.
- ❖ **Serverless Development:** AWS Cloud9 is well-suited for serverless application development. It includes AWS Lambda function support and can be used to build and test serverless applications.
- ❖ **Cloud Integration:** As part of the AWS ecosystem, AWS Cloud9 can seamlessly interact with other AWS services, making it easier to deploy and manage applications on AWS infrastructure.
- ❖ **Customization:** Developers can customize the IDE to suit their preferences by installing plugins and configuring settings.
- ❖ **Cost Management:** AWS Cloud9 offers cost-efficient pricing models, including a free tier with limited resources and pay-as-you-go pricing for additional resources.

### Pricing

AWS Cloud9 is free to use. You're only charged for specific resources you use, like EC2 instances or storage. Connecting to an existing Linux server via SSH is also free. No minimum fees or upfront commitments; you pay as you go for any additional AWS resources used within AWS Cloud9.



## AWS CodeArtifact

### What is AWS CodeArtifact?

AWS CodeArtifact is a fully managed comprehensive software artifact repository service. It is designed to help organizations store, manage, and share software artifacts such as libraries, packages, and dependencies.

AWS CodeArtifact can be used to improve the software development and deployment workflow, particularly for teams working with multiple programming languages and dependencies.

### How it is different from CodeCommit?

AWS CodeArtifact is a managed artifact repository for storing and managing software dependencies, while AWS CodeCommit is a version control service for hosting and managing source code repositories. AWS CodeArtifact focuses on artifacts, while CodeCommit focuses on code.

### Features:

- ❖ **Centralized Artifact Repository:** AWS CodeArtifact provides a centralized location for storing and managing software artifacts.
- ❖ **Support for Multiple Package Formats:** AWS CodeArtifact supports multiple package formats, including popular ones like npm (Node.js), Maven (Java), PyPI (Python), and others.
- ❖ **Security and Access Control:** AWS CodeArtifact integrates with AWS Identity and Access Management (IAM), allowing you to control who can access and publish artifacts.
- ❖ **Dependency Resolution:** AWS CodeArtifact can be used to resolve dependencies for your projects.
- ❖ **Integration with Popular Tools:** AWS CodeArtifact seamlessly integrates with popular build and deployment tools like AWS CodePipeline, AWS CodeBuild, and AWS CodeDeploy.

### Limitations:

**Package Size Limits:** There are size limits for individual packages stored in AWS CodeArtifact, and these limits may vary depending on the package format.

**Access Control Complexity:** While AWS CodeArtifact offers robust access control through IAM policies, setting up fine-grained access controls can be complex and may require careful configuration.

**No Support for Git Repositories:** AWS CodeArtifact primarily deals with artifact repositories and does not provide Git repository hosting. If you need Git repository hosting, you would typically use AWS CodeCommit or other Git hosting services.

## AWS CodeArtifact

### Best Practices:

- Create separate repositories within AWS CodeArtifact to organize your artifacts based on their purpose, project, or team.
- Follow a consistent versioning strategy for your artifacts. This ensures that you can easily track and manage changes, and it helps with dependency resolution in your projects.
- If you require high availability and disaster recovery, consider setting up cross-region replication for your artifacts.
- Integrate AWS CodeArtifact into your CI/CD pipeline to automate dependency resolution. This reduces the risk of using incompatible or outdated dependencies in your applications.
- Periodically review and clean up old or unused artifacts to manage storage costs and keep your repository organized.

### Use Case

**Multi-Region Artifact Distribution:** If you have a globally distributed development team or your application is deployed in multiple AWS regions, you can use AWS CodeArtifact to store and replicate artifacts across regions.

**Centralized Dependency Management:** With AWS CodeArtifact, the company sets up a centralized repository for all commonly used libraries and dependencies. The repository ensures that all teams access the same version of a library, promoting consistency across projects.

It also provides a single source of truth, making it easier to manage and update dependencies.

### Pricing

AWS CodeArtifact does not require any initial payments or obligations. Charges are incurred solely based on your usage, encompassing the size of stored artifacts, the volume of requests, and data egress from an AWS Region.

Moreover, AWS CodeArtifact offers a monthly free tier, granting complimentary usage of up to 2GB of storage and the initial 100,000 requests as part of the AWS Free Usage Tier.

## AWS CodeStar

### What is AWS CodeStar?

AWS CodeStar is a fully managed development service offered by Amazon Web Services (AWS) that aims to simplify the development and deployment of applications on AWS.

It provides a set of tools and services that help developers quickly build, test, and deploy applications on AWS cloud infrastructure.

### Limitations

**Language and Framework Support:** AWS CodeStar supports a variety of programming languages and frameworks, the selection may not cover all possible languages and frameworks used in software development.

**Limited Customization:** While AWS CodeStar is designed to simplify the development process, it might not provide the level of customization and flexibility that some complex projects or organizations require.

**Integration Dependencies:** AWS CodeStar is tightly integrated with other AWS services like CodeCommit, CodeBuild, and CodeDeploy. If you prefer or are required to use other third-party tools, you might face challenges in integrating them seamlessly with CodeStar.

### Features:

**Project Templates:** AWS CodeStar offers pre-configured project templates for various programming languages and application types. These templates provide a starting point for developers, saving them time on initial setup and configuration.

**Integrated Development Tools:** AWS CodeStar integrates with popular development tools such as AWS Cloud9, Visual Studio Code, and others, making it easier for developers to write code and collaborate on projects.

**Continuous Integration/Continuous Deployment (CI/CD):** Developers can automate the building, testing, and deployment of their applications, helping to maintain a reliable and efficient development workflow all these can be achieved using AWS CodePipeline.

# AWS CodeStar

## Best Practices:

- Before using AWS CodeStar, have a clear project plan in place, including goals, requirements, and a rough architecture.
- Select the AWS CodeStar project template that aligns with your project's technology stack and application type.
- Enforce a code review process within your team using tools like AWS CodeCommit. Code reviews help maintain code quality, identify bugs, and ensure that best practices are followed.
- Integrate AWS CloudWatch for monitoring and logging.
- Implement backup and disaster recovery strategies for your data and infrastructure.

## Pricing

AWS CodeStar incurs no additional fees.

You are exclusively charged for the AWS resources you allocate within your AWS CodeStar projects, such as Amazon EC2 instances, AWS Lambda executions, Amazon Elastic Block Store volumes, or Amazon S3 buckets.

There are no obligatory minimum fees or upfront commitments.

## Use Case

**Rapid Project Initialization & Deployment:** With AWS CodeStar, the startup can select a pre-configured project template (e.g., a Python web app using Flask deployed on AWS Elastic Beanstalk). CodeStar automatically provisions the necessary AWS services like AWS CodeCommit, AWS CodeBuild, AWS CodeDeploy, and AWS CodePipeline. The startup can then immediately start coding and see their changes deployed in real time.

**Standardizing Development Across Multiple Projects:** Using AWS CodeStar, the IT department can create custom project templates that align with the company's best practices and standards. Each team can then use these templates when starting a new project, ensuring a consistent development and deployment process across the enterprise.

**Note: On July 31, 2024, AWS will cease providing assistance for the creation and viewing of AWS CodeStar projects. Following this date, accessing the AWS CodeStar console or initiating new projects will no longer be feasible. Nonetheless, the AWS assets generated by AWS CodeStar, which encompass your source repositories, pipelines, and builds, will remain unaffected by this alteration and will persist in their normal functionality. It's important to note that AWS CodeStar Connections will remain unaffected by this discontinuation.**

# Amazon CodeWhisperer

## What is Amazon CodeWhisperer?

Amazon CodeWhisperer is a general-purpose, machine learning-powered code generator that provides developers with code recommendations in real time.

It is trained on billions of lines of code from public repositories and can generate code in a variety of programming languages, like Java, JavaScript, C++, and Python.

### Limitations:

- **Code quality:** The quality of the code generated by Amazon CodeWhisperer can vary depending on the complexity of the task and the quality of the training data.
- **Security vulnerabilities:** Amazon CodeWhisperer may generate code that contains security vulnerabilities.
- **Bias:** Amazon CodeWhisperer is trained on a massive dataset of code, which may contain biases.
- **Limited language support:** Amazon CodeWhisperer only supports a limited number of programming languages.
- **Limited IDE support:** Amazon CodeWhisperer is not compatible with all IDEs.

### Features:

- **Code generation:** Amazon CodeWhisperer can generate code for a variety of tasks, including completing code, writing new code, and refactoring code.
- **Security Scanning:** Amazon CodeWhisperer can scan Java, JavaScript, and Python projects to detect hard-to-find vulnerabilities.
- **Support for popular programming languages and IDEs:** Amazon CodeWhisperer supports multiple programming languages and IDEs, including Amazon SageMaker Studio, PyCharm, Visual Studio, JupyterLab, AWS Lambda console, and Cloud9.

### Best Practices

- By generating code suggestions, Amazon CodeWhisperer can help you understand how to use different language features and APIs.
- Review and test all generated code before using it in production.
- Scan all generated code for security vulnerabilities before using it in production.
- Use Amazon CodeWhisperer in conjunction with other tools. CodeWhisperer is a powerful tool, but it should not be used in isolation.

## Amazon CodeWhisperer

### Best Practices

- By generating code suggestions, Amazon CodeWhisperer can help you understand how to use different language features and APIs.
- Review and test all generated code before using it in production.
- Scan all generated code for security vulnerabilities before using it in production.
- Use Amazon CodeWhisperer in conjunction with other tools. CodeWhisperer is a powerful tool, but it should not be used in isolation.

### Use Cases:

Use Amazon CodeWhisperer to generate code for boilerplate tasks. Boilerplate tasks are repetitive tasks that are often required to write code, such as creating database tables, initializing variables, and writing error-handling code.

Use Amazon CodeWhisperer to learn new programming languages and APIs.

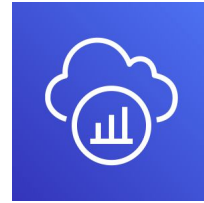
Use Amazon CodeWhisperer to write code for unfamiliar domains. If a developer is working on a project in a domain that they are not familiar with, CodeWhisperer can help them write code for that domain.

- ❖ **Individual Tier:** The Individual Tier is free for individual developers. It includes unlimited code suggestions, reference tracking, and up to 50 code scans per month.
- ❖ **Professional Tier:** The Professional Tier costs \$19 per user per month. It includes all of the features of the Individual Tier, plus enhanced security scanning, the ability to generate code for more complex tasks, and priority support.

# AWS X-Ray

## What is AWS X-Ray?

AWS X-Ray is a service that allows visual analysis or allows to trace microservices based applications.



- ✓ It provides end-to-end information about the request, response and calls made to other AWS resources by travelling through the application's underlying components consisting of multiple microservices.
- ✓ It creates a service graph by using trace data from the AWS resources.
  - The graph shows the information about front-end and backend services calls to process requests and continue the flow of data.
  - The graph helps to troubleshoot issues and improve the performance of the applications.

*It works with the following AWS services:*

- AWS EC2 (Applications deployed on Instances)
- AWS Elastic Load Balancer
- AWS Elastic BeanStalk
- AWS Lambda
- Amazon ECS (Elastic Container Service)
- Amazon API Gateway

The X-Ray SDKs are available for the following languages:

- Go
- Java
- Node.js
- Python
- Ruby
- .Net

