WHIZLABS

AWS Data Engineer Associate

Cheat Sheet

Quick Bytes for you before the exam!

The content in Cheat Sheet is designed for educational purposes to assist aspirants in preparing for the AWS Data Engineer Associate. Though references have been taken from Amazon Cloud 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.

Are you Ready for

AWS Data Engineer Associate?

Self-assess yourself with

Whizlabs FREE TEST

750+ Hands-on-Labs

Hands-on Labs - AWS, GCP, Azure (Whizlabs)

Cloud Sandbox environments

Cloud Sandbox - AWS, Azure, GCP & Power BI

Index

– Back to Index –

Amazon Athena

What is Amazon Athena?

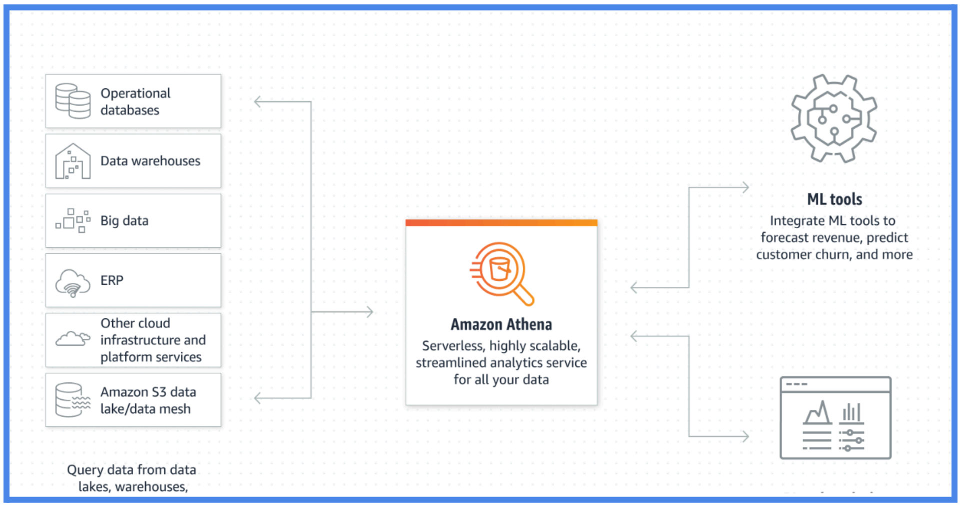
Amazon Athena is an interactive serverless service used to analyze data directly in Amazon Simple Storage Service using standard SQL ad-hoc queries.

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 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.


Operational databases
Data warehouses
Big data
ERP
Other cloud infrastructure and platform services
Amazon S3 data lake/data mesh
Query data from data lakes, warehouses,
Amazon Athena
Serverless, highly scalable, streamlined analytics service for all your data
ML tools
Integrate ML tools to forecast revenue, predict customer churn, and more

[Source: AWS Documentation]

Amazon EMR

What is Amazon EMR?

Amazon EMR (Elastic Map Reduce) is a service used to process and analyze large amounts of data in the cloud using Apache Hive, Hadoop, Apache Flink, Spark, etc.

- The main component of EMR is a cluster that collects Amazon EC2 instances (also known as nodes in EMR).
- It decouples the compute and storage layer by scaling independently and storing cluster data on Amazon S3.
- It also controls network access for the instances by configuring instance firewall settings.
- It offers basic functionalities for maintaining clusters such as monitoring, replacing failed instances, bug fixes, etc.
- It analyzes machine learning workloads using Apache Spark MLlib and TensorFlow, clickstream workloads using Apache Spark and Apache Hive, and real-time streaming workloads from Amazon Kinesis using Apache Flink.

It provides more than one compute instance or container to process the workloads and can be executed on the following AWS services:

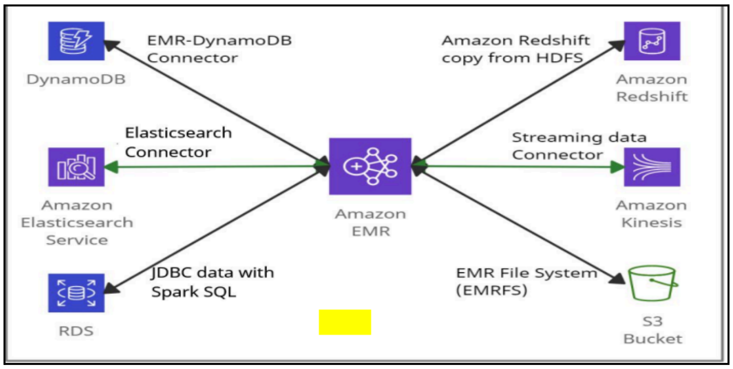
- Amazon EC2
- Amazon EKS
- AWS Outposts

Amazon EMR can be accessed in the following ways:

- EMR Console
- AWS Command Line Interface (AWS CLI)
- Software Development Kit (SDK)
- Web Service API

It offers basic functionalities for maintaining clusters such as

- Monitoring
- Replacing failed instances
- Bug fixes



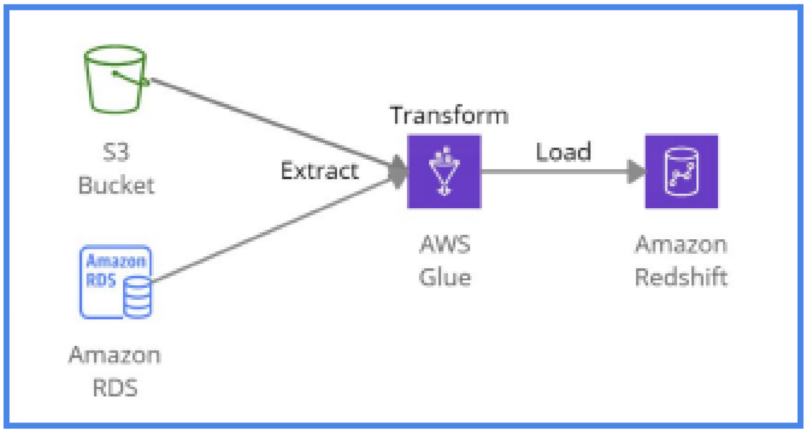
AWS Glue

What is AWS Glue?

AWS Glue is a serverless ETL (extract, transform, and load) service used to categorize data and move them between various data stores and streams.

AWS Glue works with the following services:

- Redshift - for data warehouses
- S3 - for data lakes
- RDS or EC2 instances - for data stores



Properties of AWS Glue:

- It supports data integration, preparing and combining data for analytics, machine learning, and other application development
- It has a central repository known as the AWS Glue Data Catalog that automatically generates Python or Scala code
- It processes semi-structured data using a simple 'dynamic' frame in the ETL scripts similar to an Apache Spark data frame that organizes data into rows and columns.
- It helps execute the Apache Spark environment's ETL jobs by discovering data and storing the associated metadata in the AWS Glue Data Catalog.
- AWS Glue and Spark can be used together by converting dynamic frames and Spark data frames to perform all kinds of analysis
- It allows organizations to work together and perform data integration tasks, like extraction, normalization, combining, loading, and running ETL workloads

AWS Glue DataBrew

Overview:

- AWS Glue DataBrew is a user-friendly tool designed for data analysts and scientists to streamline data cleaning and normalization for analytics and machine learning purposes.
- With access to a vast library of over 250 prebuilt transformations, users can automate various data preparation tasks effortlessly, eliminating the need for manual coding.
- The tool facilitates tasks like filtering anomalies, converting data to standardized formats, and rectifying invalid values, ensuring data is primed for analysis.
- Once data is prepared, it can be readily utilized for analytics and ML projects, saving time and effort.
- Users enjoy a pay-as-you-go model, meaning they only pay for the resources they consume without any upfront commitments.

Features:

- Utilize data profiling to assess data quality, identifying patterns and anomalies, and establishing connections directly from your data repositories such as data lakes, warehouses, and databases.
- Streamline data cleaning and normalization processes through an intuitive point-and-click visual interface, leveraging a diverse array of over 250 prebuilt transformations.
- Gain insights into your data's lineage by visually mapping its journey through various sources and transformation stages, aiding in understanding its origins and processing steps.
- Implement automated data preparation workflows, enabling the application of saved transformations to incoming data in real-time, enhancing efficiency and consistency in data handling.

AWS Lake Formation

A data lake is a secure repository that stores all the data in its original form and is used for analysis.

What is AWS Lake Formation?

AWS Lake Formation is a cloud service that is used to create, manage, and secure data lakes. It automates the complex manual steps required to create data lakes.

AWS Lake Formation integrates with:

- Amazon CloudWatch
- Amazon CloudTrail
- Amazon Glue: Both use the same Data Catalog
- Amazon Redshift Spectrum
- Amazon EMR
- AWS Key Management Service
- Amazon Athena: Athena's users can query the AWS Glue catalog which has Lake Formation permissions on them.

Lake Formation is pointed at the data sources, then crawls the sources and moves the data into the new Amazon S3 data lake.

It integrates with AWS Identity and Access Management (IAM) to provide fine-grained access to the data stored in data lakes using a simple grant/revoke process

Pricing Details:

Charges are applied based on the service integrations (AWS Glue, Amazon S3, Amazon EMR, Amazon Redshift) at a standard rate

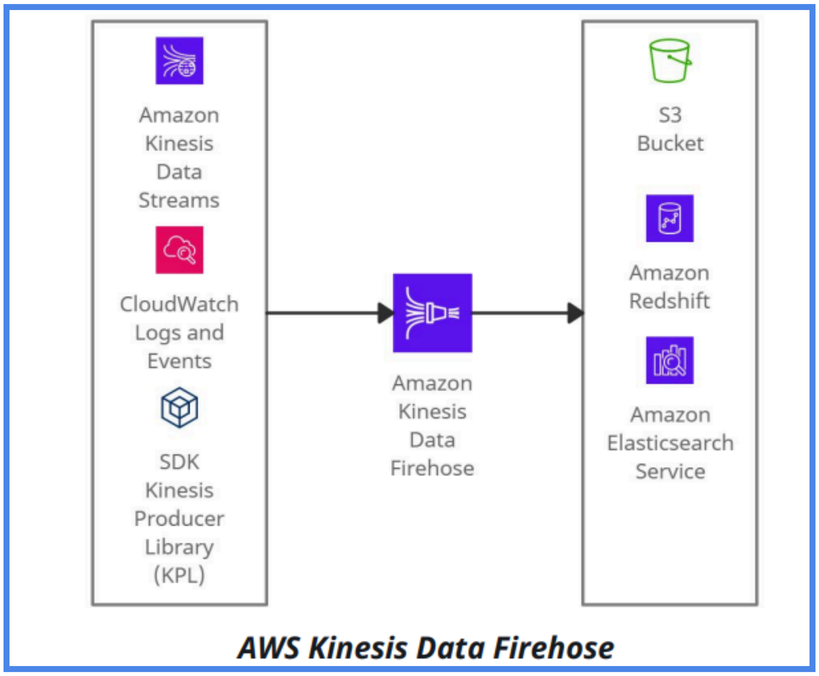
Amazon Data Firehose

What is Amazon Data Firehose?

Amazon 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 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.
- Data Streams and CloudWatch events can be considered as the source(s) to Data Firehose.

  It delivers streaming data to the following services:
  - Amazon S3
  - Amazon Redshift
  - Amazon Elasticsearch Service
  - Splunk
  - AWS Data Firehose


Amazon Kinesis Data Streams
CloudWatch Logs and Events
SDK Kinesis Producer Library (KPL)
Amazon Kinesis Data Firehose
S3 Bucket
Amazon Redshift
Amazon Elasticsearch Service

AWS Kinesis Data Firehose

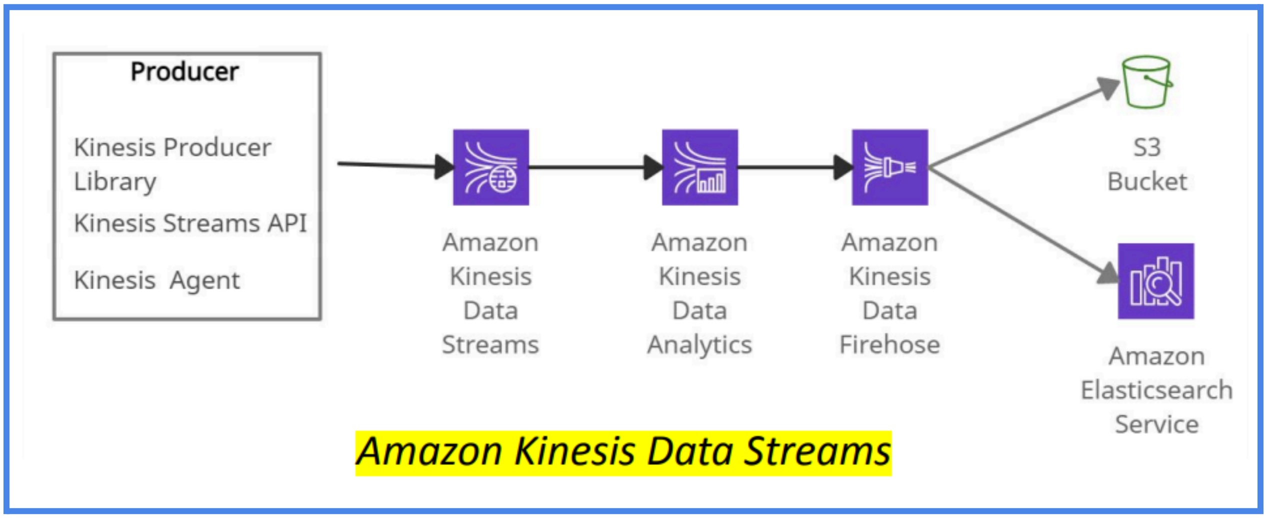
Amazon Kinesis Data Streams

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

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

- The 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 which 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.


Producer
Kinesis Producer Library
Kinesis Streams API
Kinesis Agent
Amazon Kinesis Data Streams
Amazon Kinesis Data Analytics
Amazon Kinesis Data Firehose
S3 Bucket
Amazon Elasticsearch Service

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 of input data and 2MB per second of output data.

Amazon Managed Service for Apache Flink

What is Amazon Managed Service for Apache Flink?

The Amazon Managed Service for Apache Flink stands as a holistic data streaming platform, adept at managing the intricacies involved in configuring and linking data origins and destinations, all while requiring minimal coding efforts. This service streamlines the process of continuous data processing, ensuring minimal latency to enable organizations to swiftly address real-time events as they unfold.

Features:

- Develop and execute Apache Flink applications without the hassle of infrastructure setup and resource management.
- Achieve high-speed data processing, handling gigabytes of data every second, while maintaining subsecond latencies.
- React to events in real-time by leveraging the capabilities of Apache Flink for rapid event processing.
- Ensure high availability and durability of applications through Multi-AZ deployments.
- Utilize APIs for seamless management of the application lifecycle, facilitating efficient deployment and maintenance processes.

Use Cases:

- Rapidly deliver streaming data to destinations like Amazon S3 and OpenSearch Service.
- Create real-time analytics apps for interactive querying and analysis.
- Continuously generate insights for time-sensitive use cases.
- Implement stateful processing for executing long-running computations and anomaly detection using historical data.

Pricing

- Apache Flink application pricing is based on Kinesis Processing Units (KPUs) covering compute and memory, with extra charges for orchestration, storage, and backups.
- Streaming mode scales KPUs automatically according to memory and compute needs, and allows manual KPU provisioning.
- Apache Flink Studio charges additional KPUs for interactive mode, alongside storage fees. Testing with production loads helps gauge accurate KPU usage.

Amazon Managed Streaming for Apache Kafka (Amazon MSK)

It helps to populate machine learning applications, analytical applications, and data lakes, and stream changes to and from databases using Apache Kafka APIs.

What is Amazon MSK?

Amazon MSK is a managed cluster service used to build and execute Apache Kafka applications for processing streaming data.

It provides multiple kinds of security for Apache Kafka clusters, including:

- AWS IAM for API Authorization
- Encryption at Rest
- Apache Kafka Access Control Lists (ACLs)

It easily configures applications by removing all the manual tasks used to configure.

The steps that Amazon MSK manages are:

- Replacing servers during failures
- Handling server patches and upgrades with no downtime
- Maintenance of Apache Kafka clusters
- Maintenance of Apache ZooKeeper
- Multi-AZ replication for Apache Kafka clusters
- Planning scaling events

Amazon MSK Integrates with:

- AWS Glue: To execute Apache Spark job on Amazon MSK cluster
- Amazon Kinesis Data Analytics: To execute the Apache Flink job on the Amazon MSK cluster
- Lambda Functions

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.

Amazon OpenSearch Service can be integrated with the following services:

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

What is Amazon OpenSearch Service?

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

Amazon OpenSearch Service with Kibana (visualization) & Logstash (log ingestion) provides an enhanced search experience for applications and websites to find relevant data quickly

Amazon OpenSearch Service launches the Elasticsearch cluster's resources detects the failed Elasticsearch nodes and replaces them.

The OpenSearch Service cluster can be scaled with a few clicks in the console.

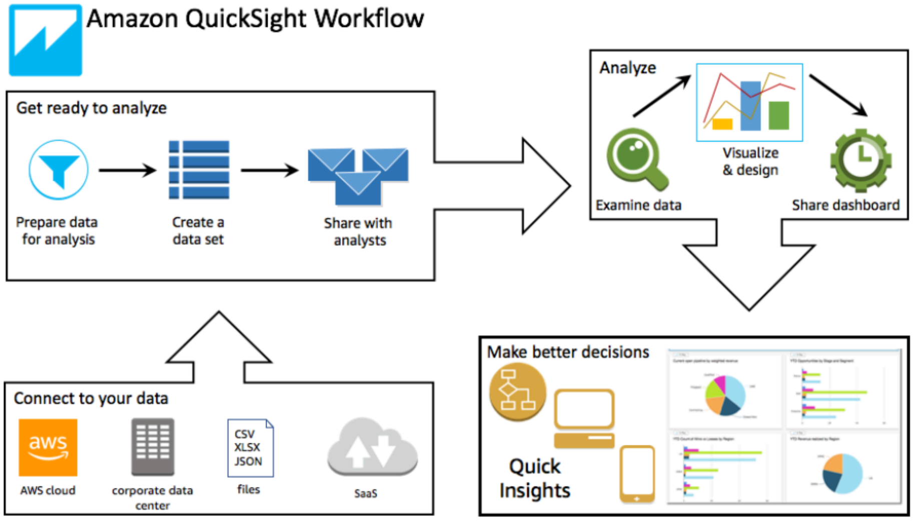
Pricing Details:

- 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 QuickSight

What is Amazon QuickSight?

- Amazon QuickSight: A scalable cloud-based BI service providing clear insights to collaborators worldwide.
- Connects to various data sources, consolidating them into single data dashboards.
- Fully managed with enterprise-grade security, global availability, and built-in redundancy.
- User management tools support scaling from 10 users to 10,000 without infrastructure deployment.
- Empowers decision-makers to explore and interpret data interactively.
- Securely accessible from any network device, including mobile devices.



Features:

- Automatically generate accurate forecasts.
- Automatically detect anomalies.
- Uncover latent trends.
- Take action based on critical business factors.
- Transform data into easily understandable narratives, such as headline tiles for your dashboard.

The platform offers enterprise-grade security with authentication for federated users and groups via IAM Identity Center, supporting single sign-on with SAML, OpenID Connect, and AWS Directory Service. It ensures fine-grained permissions for AWS data access, row-level security, and robust encryption for data at rest. Users can access both AWS and on-premises data within Amazon Virtual Private Cloud for enhanced security.

Benefits:

- Achieve a 74% cost reduction in BI solutions over three years, with up to a 300% increase in analytics usage.
- Enjoy no upfront licensing costs and minimal total cost of ownership (TCO).
- Enable collaborative analytics without application installation.
- Aggregate diverse data sources into single analyses and share them as dashboards.
- Manage dashboard features, permissions, and simplify database permissions management for viewers accessing shared content.

Amazon Q in QuickSight:

Amazon Q within QuickSight enhances business productivity by leveraging Generative BI capabilities to expedite decision-making. New dashboard authoring features empower analysts to swiftly build, discover, and share insights using natural language prompts. Amazon Q simplifies data comprehension with executive summaries, an improved context-aware Q&A experience, and customizable interactive data stories.

Pricing:

- QuickSight offers flexible pricing based on user roles, allowing selection of the model that aligns with business requirements.
- A low $3/month reader fee enables organization-wide access to interactive analytics and natural language capabilities.
- Choose between per-user pricing and capacity pricing based on business needs.

Links: https://docs.aws.amazon.com/quicksight/latest/user/welcome.html

Amazon QuickSight - Business Intelligence Tools

https://aws.amazon.com/quicksight/pricing/

Amazon AppFlow

What is Amazon AppFlow?

Amazon AppFlow is a service that enables seamless bidirectional data movement between various Software as a Service (SaaS) applications and AWS services with minimal effort. Users can easily set up automated data flows between these applications and AWS, scheduling them as needed or triggering them in response to specific business events.

With AppFlow, data preparation tasks such as transformations, partitioning, and aggregation are simplified, streamlining the process of getting data ready for analysis or use in machine learning.

Additionally, AppFlow automates the preparation and registration of data schemas with the AWS Glue Data Catalog, facilitating data discovery and sharing across AWS analytics and machine learning services.

Features:

- Amazon AppFlow: Managed integration service for seamless data transfer between Salesforce, SAP, Google Analytics, and Amazon Redshift.
- Scalable Data Transfer: Allows transfer at scale without user provision of system resources, ensuring efficiency.
- Automated Data Cataloging: Automates data cataloging for easier discovery and sharing across AWS analytics and machine learning services.
- Streamlined Data Preparation: Simplifies and automates tasks like transformations, partitioning, and aggregation for efficient data preparation.

Use Cases:

- Gain a holistic customer view: Integrate marketing, customer support, and sales data for a complete understanding of the customer journey.
- Enhance SaaS data: Utilize Amazon SageMaker Data Wrangler to import datasets for ML model training and operationalize data with reverse ETL.
- Automate event-driven workflows: Create application workflows triggered by data events from various sources, like generating Salesforce records from Marketo leads.
- Streamline Salesforce data analysis: Transfer opportunity records from Salesforce to Amazon Redshift for real-time dashboard updates and analysis.

Pricing:

You are charged for each successful data transfer operation initiated by a flow run, regardless of whether new data is found. Additionally, certain connectors like SAP offer the option to expedite transfers with extra concurrent processes, each of which incurs an additional charge.

- Price per flow run
- Maximum number of flow runs per AWS account per month - 10 Million

Amazon EventBridge

What is Amazon EventBridge?

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

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

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

Key Concepts

Event Buses

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

Events

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

Shema Registry

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

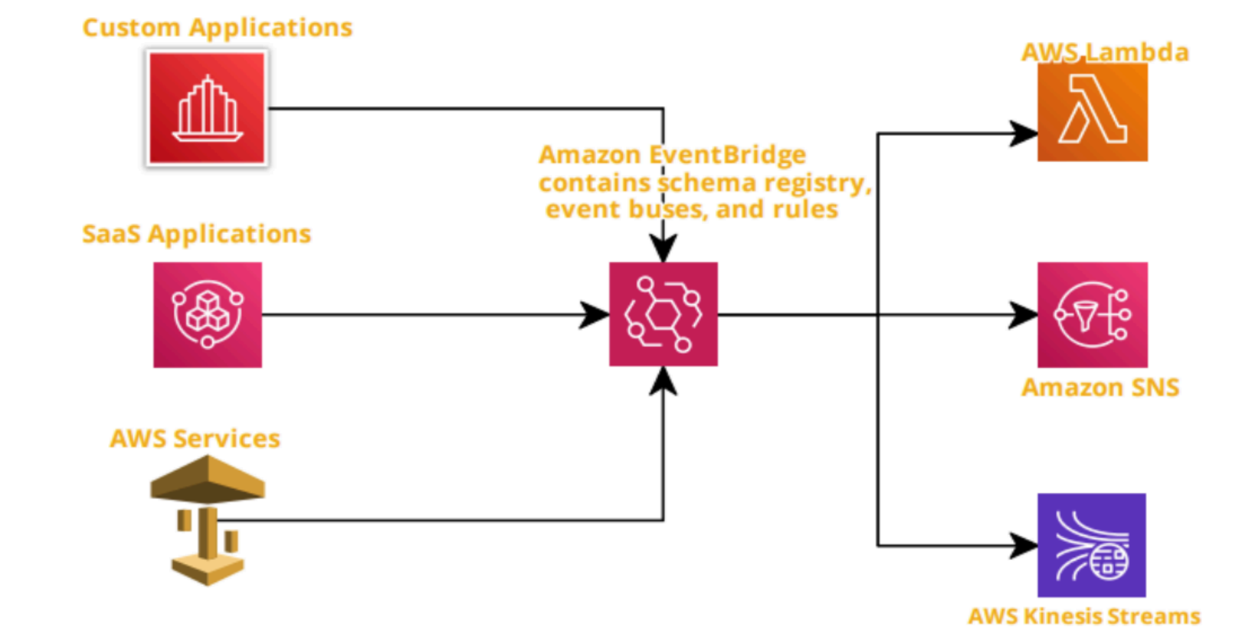
Rules

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

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

Features:

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



Structure of the Amazon EventBridge service.

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

Pricing

- There are no additional charges for rules or event delivery.
- The users only pay for events published to your event bus, events ingested for Schema Discovery, and Event Replay.

  Custom events: Charge $1.00 per million requests.
  Third-party events (SaaS): Charge $1.00 per million requests.
  Cross-account events: $1.00 per million.

AWS SNS (Simple Notification Service)

What is AWS SNS?

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

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

It allows developers to group multiple recipients using topics.
It consists of **topics and subscribers**.

A topic is an access point for allowing recipients to get identical copies for the same notification. One topic can support deliveries to multiple end-points – for example - we can group together to android, IOS, and SMS text messages.
Two types of topics can be defined in the AWS SNS service.

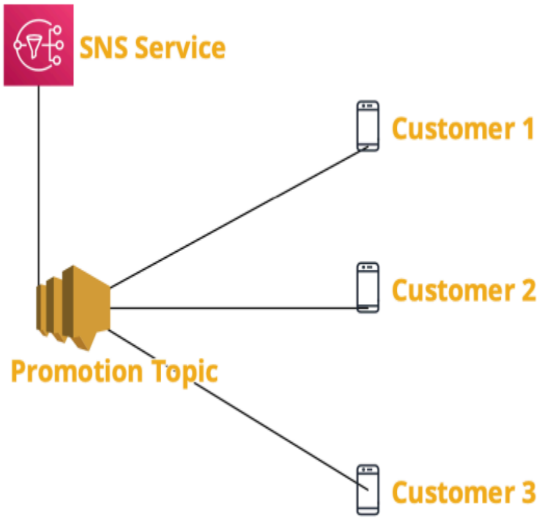
1. Standard topic is used when incoming messages are not in order. In other words, messages can be delivered as they are received.
2. FIFO topic is designed to maintain order of the messages between the applications, especially when the events are critical. Duplication will be avoided in this case.

Features

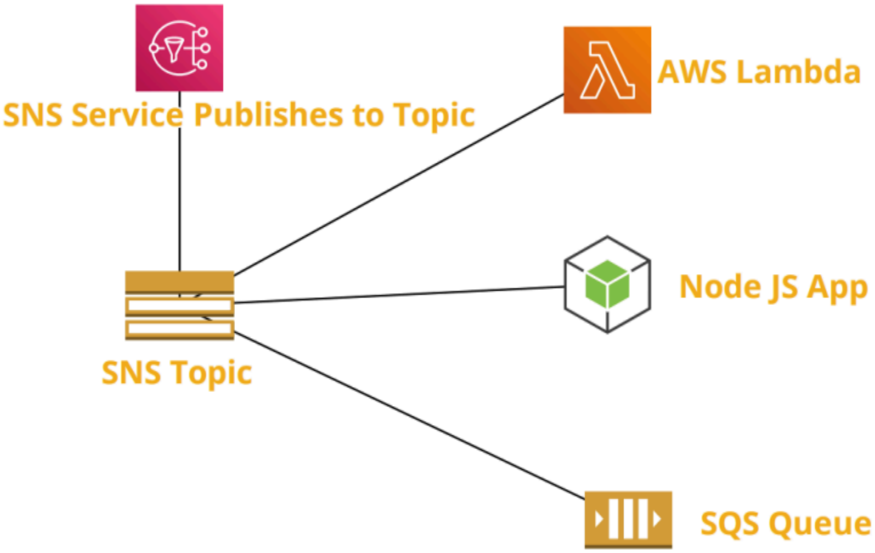
- Instantaneous, push-based delivery.
- Simple API and easy integration with AWS services.
- Flexible message delivery over multiple message protocols.
- Cost-effective – as pay as pay-as-you-go model.
- Fully managed and durable with automatic scalability.

Use cases

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



- SNS Application to Application: In this type of service, where SNS topic would interact with different AWS services such as AWS Lambda, Node JS app, and SQS services. For example, AWS S3 service has only configuration with AWS SNS service, which will be responsible for sending identical messages to other AWS services.



Pricing

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

Amazon Simple Queue Service (SQS)

What is Amazon Simple Queue Service (SQS)?

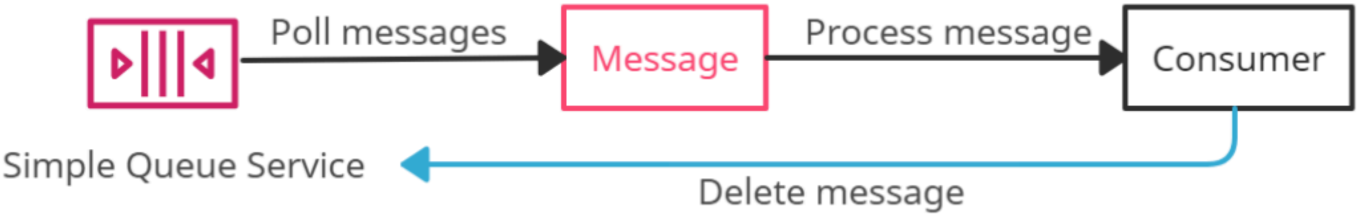
Amazon Simple Queue Service (SQS) is a serverless service used to decouple (loose couple) serverless applications and components.
The queue represents a temporary repository between the producer and consumer of messages.
It can scale up to 1-10000 messages per second.
The default retention period of messages is four days and can be extended to fourteen days.
SQS messages get automatically deleted after being consumed by the consumers. SQS messages have a fixed size of 256KB.
There are two SQS Queue types:

Standard Queue -

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

FIFO Queue -

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



Amazon SQS

Delay Queue is a queue that allows users to postpone/delay the delivery of messages to a queue for a specific number of seconds. Messages can be delayed for 0 seconds (default) -15 (maximum) minutes.

Dead-Letter Queue is a queue for those messages that are not consumed successfully. It is used to handle message failure. Visibility Timeout is the amount of time during which SQS prevents other consumers from receiving (poll) and processing the messages.

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

AWS Step Functions

What are Step functions?

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

Types of step functions:

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

Features:

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

Best Practices:

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

  Example:

  ```
  "ActivityState": {
  "Type": "Task",
  "Resource":
  "arn:aws:states:us-east-1:123456789012:activity:abc",
  "TimeoutSeconds": 900,
  "HeartbeatSeconds": 40,
  "Next": "State2" }
  ```
- Always provide the Amazon S3 arn (amazon resource name) instead of large payloads to the state machine when passing input to Lambda function.

  Example:

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

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

```
        "Retry": [ {
                "ErrorEquals": [ "Lambda.CreditServiceException"]
                "IntervalSeconds": 2,
                "MaxAttempts": 3,
                "BackoffRate": 2
               } ]
```

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

It supports below AWS services:

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

Pricing:

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

AWS Budgets

What is AWS Budgets?

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

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

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

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

Best Practices:

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

● Two of the related managed policies are provided for budget actions. One policy allows a user to pass a role to the budgets service, and the other allows budgets to execute the action.
● Budget actions are not effective enough to control costs with Auto Scaling groups.

Price details:

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

AWS Cost Explorer

What is AWS Cost Explorer?

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

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

The default reports provided by Cost Explorer are:

- ### Cost and Usage Report
- It provides the following data for understanding the costs:-

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

- ### Reserved Instance Reports:

  It provides the following reports for understanding the reservations:-

    - RI utilization reports: It gives information about how much costs are saved or overspent by using Reserved Instances (RIs)
    - RI Coverage Reports: It gives information about how many hours are saved or overspent by using Reserved Instances (RIs).

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

AWS Batch

What is AWS Batch?

AWS Batch allows developers, scientists, and engineers to run thousands of computing jobs in the AWS platform. It is a managed service that dynamically maintains the optimal compute resources like CPU, Memory based on the volume of submitted jobs. The User just has to focus on the applications (like shell scripts, Linux codes or java programs).
It executes workloads on EC2 (including Spot instances) and AWS Fargate.

Components:

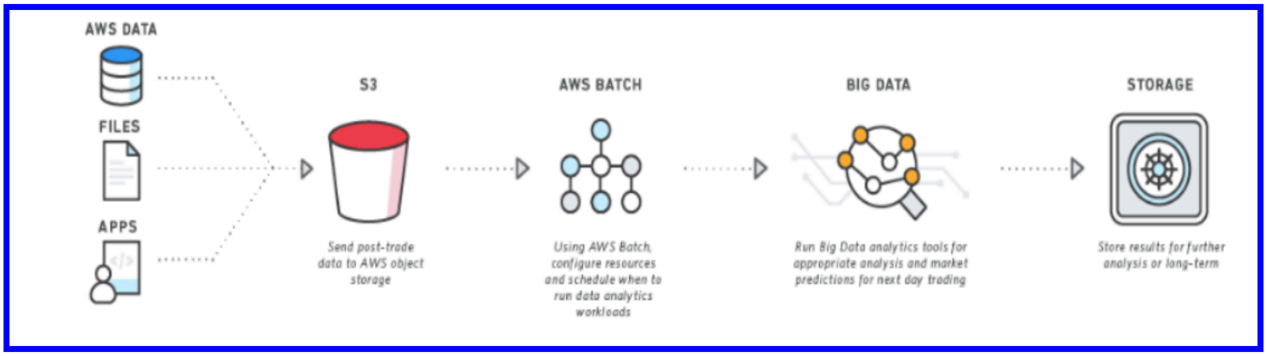
- Jobs - are the fundamental applications running on Amazon EC2 machines in containerised form
- Job Definitions – define how the job is meant to be run. Like the associated IAM role, vCPU requirement, and container properties.
- Job Queues – Jobs reside in the Job queue where they wait till they are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

Best Practices:

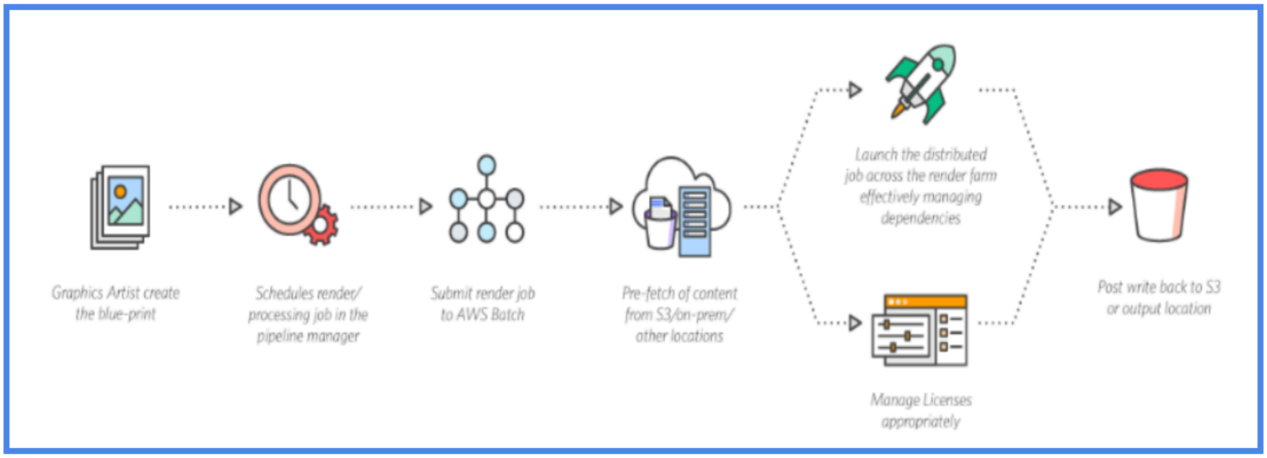
- Use Fargate if you want to run the application without getting into EC2 infrastructure details. Let the AWS batch manage it.
- Use EC2 if your work scale is very large and you want to get into machine specifications like memory, CPU, GPU.
- Jobs running on Fargate are faster on startup as there is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

Use Cases:

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


AWS DATA
FILES
APPS
S3
Send post-trade data to AWS object storage
AWS BATCH
Using AWS Batch, configure resources and schedule when to run data analytics workloads
BIG DATA
Run Big Data analytics tools for appropriate analysis and market predictions for next day trading
STORAGE
Store results for further analysis or long-term

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


Graphics Artist create the blue-print
Schedules render/ processing job in the pipeline manager
Submit render job to AWS Batch
Pre-fetch of content from S3/on-prem/ other locations
Launch the distributed job across the render farm effectively managing dependencies
Manage Licenses appropriately
Post write back to S3 or output location

Pricing:

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

AWS EC2

What is AWS EC2?

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

Instance Type:

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

EBS Volume:

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

Instance Store:

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

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

AMI:

AMI Stands for Amazon Machine Image.

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

Security Group:

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

- It decides the type of port and kind of traffic to allow.

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

Key Pair:

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

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

Tags:

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

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

Pricing:

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

AWS Lambda

What is AWS Lambda?

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

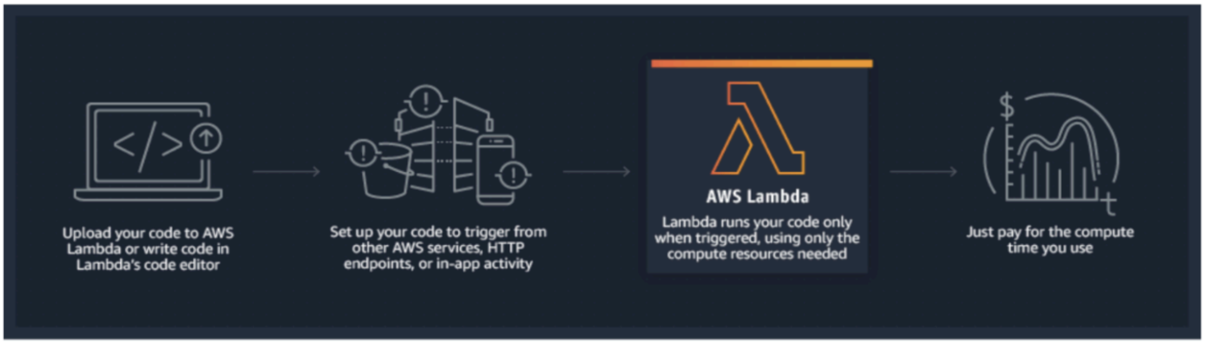
What is Serverless computing?

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

When do you use Lambda?

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

How does Lambda work?


Upload your code to AWS Lambda or write code in Lambda's code editor
Set up your code to trigger from other AWS services, HTTP endpoints, or in-app activity
AWS Lambda
Lambda runs your code only when triggered, using only the compute resources needed
Just pay for the compute time you use

Lambda Functions

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

Lambda Layers

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

Lambda Event

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

o AWS IoT
o Kinesis
o CloudWatch Logs

Language Supported in AWS Lambda

- NodeJS
- Go
- Java
- Python
- Ruby

Lambda@Edge

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

Pricing:

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

AWS Serverless Application Model (SAM)

What is AWS Serverless Application Model (SAM)?

AWS Serverless Application Model (SAM) comprises AWS SAM templates and the AWS SAM Command Line Interface (CLI). SAM templates offer a concise syntax tailored for defining Infrastructure as Code (IaC) for serverless applications.
As an extension of AWS CloudFormation, SAM templates are deployed directly to AWS CloudFormation, leveraging its robust IaC capabilities on AWS. The AWS SAM CLI is a developer-centric tool that empowers users to efficiently create, develop, and deploy serverless applications.
Notable features of AWS SAM include SAM Accelerate, which enhances local development and cloud testing speed, and SAM CLI integrations that expand SAM functionality to other tools like AWS CDK and Terraform.

Features:

- Accelerate Serverless Development: AWS SAM CLI speeds up the development process for serverless applications, facilitating rapid progress from concept to deployment.
- Simplified Development Process: Develop, debug, and deploy serverless applications smoothly using straightforward AWS SAM CLI commands, streamlining the workflow.
- Seamless Infrastructure Management: Easily define and oversee infrastructure components using concise AWS SAM templates, ensuring effortless management.
- Cloud-Centric Debugging and Testing: Perform real-time debugging and testing directly in the cloud with AWS SAM Accelerate, enabling efficient troubleshooting and problem resolution.

Use Cases:

- Build and Deploy Serverless Apps: Utilize AWS SAM CLI commands like sam build and sam deploy to prepare and deploy serverless applications on AWS.
- Sync to Cloud: Employ sam sync to watch for local changes and swiftly deploy them to the AWS Cloud for development and testing purposes.
- CI/CD Pipelines: Create or adjust pipelines for your CI/CD system using sam pipeline to streamline deployment processes.
- Terraform Integration: Debug and test Terraform projects locally using AWS SAM CLI for enhanced development efficiency.

– Back to Index –

Amazon Elastic Container Registry

What is Amazon Elastic Container Registry?

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

Features:

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

Pricing details:

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

Amazon Elastic Container Service

What is Amazon ECS?

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

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

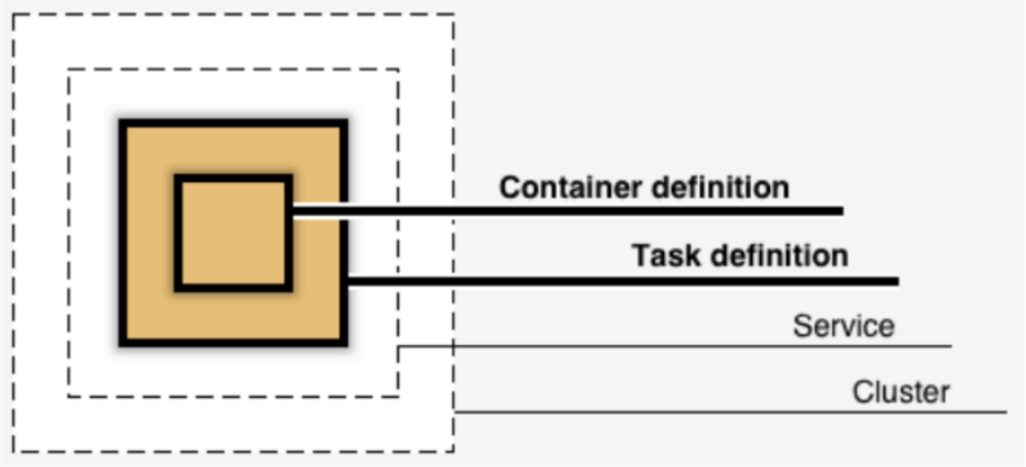
Images are created from a Dockerfile (text format), which specifies all of the components that are included in the container. These images are then stored in a registry from where they can then be downloaded and executed on the cluster.

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

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

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

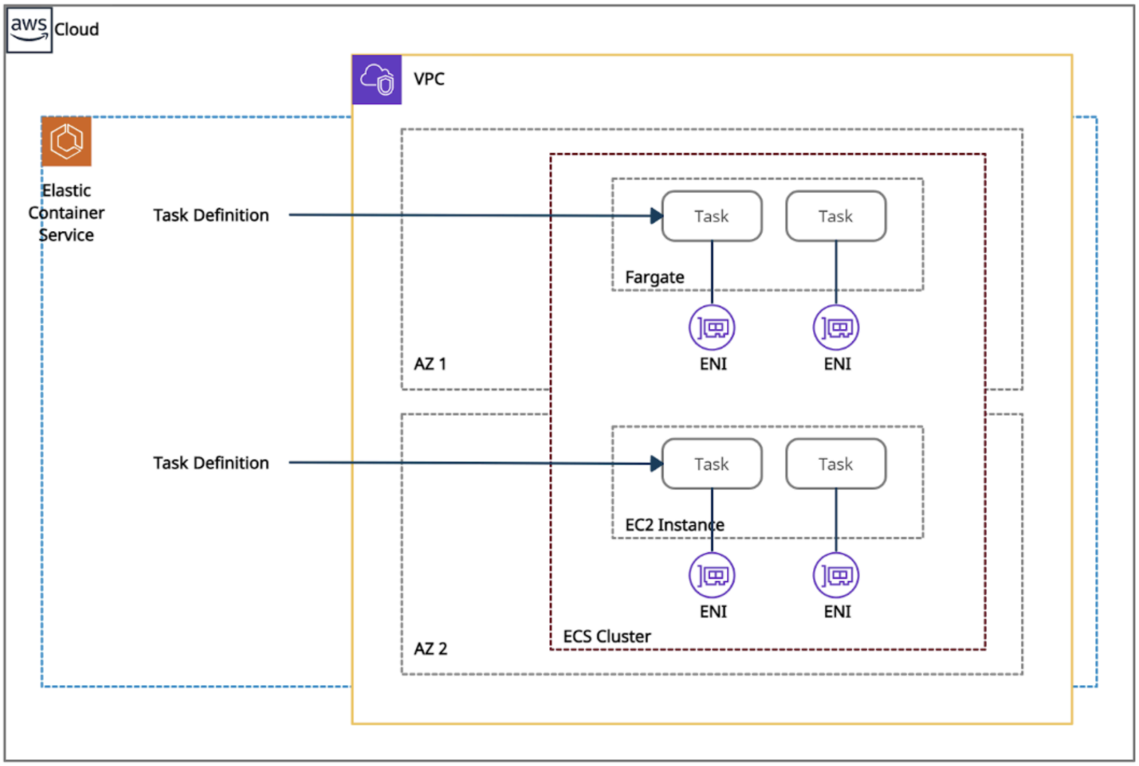
The number of tasks to run on your cluster is specified after the task definition is created within Amazon ECS. The task scheduler is responsible for attaching tasks within your cluster based on the task definitions.



ECS Cluster

Application Load Balancers offer some attractive features:

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



Amazon Elastic Container Service

Amazon ECS can be integrated with:

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

  - ➢ It decreases time consumption by eliminating user tasks to install, operate, and scale cluster management infrastructure.With API calls, Docker-enabled applications can be launched and stopped.
  - ➢ It powers other services such as Amazon SageMaker, AWS Batch, Amazon Lex. It also integrates with AWS App Mesh, to provide rich observability, controls traffic and security features to the applications.

Use Cases:

The two main use cases in Amazon ECS are:

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

.

Pricing details:

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

Amazon Elastic Kubernetes Service(EKS)

What is Amazon Elastic Kubernetes Service(EKS)?

Amazon Elastic Kubernetes Service (Amazon EKS) is a service that enables users to manage Kubernetes applications in the AWS cloud or on-premises. Any standard Kubernetes application can be migrated to EKS without altering the code.

The EKS cluster consists of two components:

- Amazon EKS control plane
- Amazon EKS nodes

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

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

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

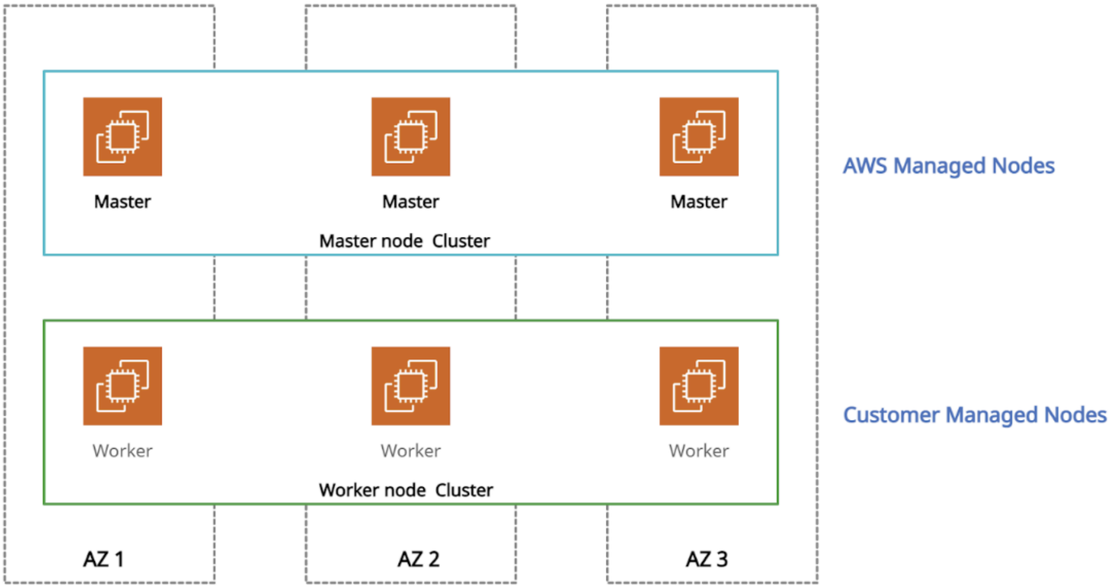
- eksctl – A command-line utility that consists of kubectl for creating and managing Kubernetes clusters on Amazon EKS.
- AWS Management Console and AWS CLI

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

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

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

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



Amazon EKS

Use Cases:

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

Price details:

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

Amazon DocumentDB(with MongoDB compatibility)

What is Amazon DocumentDB?

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

Features:

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

Best Practices:

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

Use Case:

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

Pricing:

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

Amazon DynamoDB

What is Amazon DynamoDB?

AWS DynamoDB, offered by Amazon, is a versatile database solution supporting both key-value and DocumentDB structures. It boasts an impressive feature set, including single-digit millisecond latency, the capability to manage a staggering 20 million requests per second, and handling up to 10 trillion requests daily. As a serverless service, DynamoDB eliminates the need for manual server management, providing a hassle-free experience for users. Furthermore, it ensures consistent performance by efficiently distributing data traffic across multiple servers, optimizing resource utilization and enhancing scalability.

Features:

It can create the Table for your application and can handle the rest.

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

Pricing:

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

Amazon Keyspaces (for Apache Cassandra)

What is Amazon Keyspaces (for Apache Cassandra)?

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

Features:

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

Use Cases:

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

Pricing:

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

Amazon MemoryDB for Redis

What is Amazon MemoryDB for Redis?

Amazon MemoryDB for Redis is a high-performance database solution designed for rapid read and write operations, scalability, and robust security. It ensures 99.99% availability and offers near-instantaneous recovery capabilities without compromising data integrity. As a Redis-compatible service, MemoryDB provides durable, in-memory storage for optimal performance.

Features:

- Achieve massive scalability, supporting hundreds of millions of requests per second and over one hundred terabytes of storage per cluster.
- Ensure durability of data through multi-AZ transaction logs, providing 99.9% availability and rapid recovery in case of failures without compromising data integrity.
- Enhance data security with encryption mechanisms for data at rest and in transit, private endpoints for restricted access, and multiple authentication methods including IAM authentication.
- Expedite application development by leveraging Redis data structures, utilizing a comprehensive open-source API, and seamlessly integrating with various AWS services.

Use Cases:

- Expedite app development using Redis open source's versatile data structures, speeding up time-to-market.
- Retrieve customer data efficiently for personalized experiences, inventory tracking, and profile management, with fast read and write speeds.
- Fuel online gaming apps with scalable player data stores, session histories, and leaderboards, ensuring real-time updates and low latency.
- Support high-concurrency streaming feeds for media platforms, handling millions of daily requests for interactive features.

Pricing:

- Instance Usage Time: Duration of on-demand instance usage per node, billed per hour.
- Data Output: Volume of transmitted or written data to a cluster, billed per gigabyte.
- Snapshot Storage: Space used by automated and manual snapshots, billed per gigabyte-month.

Amazon Neptune

What is Amazon Neptune?

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

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

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

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

Query Language.

- It is highly available across three AZs and automatically fails over any of the 15 low latency read replicas.
- It provides fault-tolerant storage by replicating two copies of data across three availability zones.
- It provides continuous backup to Amazon S3 and point-in-time recovery from storage failures.
- It automatically scales storage capacity and provides encryption at rest and in transit.

Amazon RDS

What is Amazon RDS?

RDS (Relational Database System) in AWS makes it easy to operate, manage, and scale in the cloud. It provides scalable capacity with a cost-efficient pricing option and automates manual administrative tasks such as patching, backup setup, and hardware provisioning.

Engines supported by RDS are given below:

MySQL

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

MS SQL

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

MariaDB

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

PostgreSQL

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

Oracle

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

Amazon Aurora

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

DB Instance Class

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

Multi AZ Deployment

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

Read Replicas

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

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

Storage Type

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

Monitoring

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

Backups & Restore

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

Charges:

You will be charged based on multiple factors:

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

Amazon Redshift

What is Amazon Redshift?

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

Redshift can be configured as follows:

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

Features:

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

Security Considerations

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

Use cases

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

AWS Command Line Interface (AWS CLI)

What is AWS Command Line Interface (AWS CLI)?

The AWS Command Line Interface (AWS CLI) serves as a comprehensive solution for overseeing AWS services. It simplifies management by providing a single tool for controlling various AWS offerings directly from the command line, enabling automation through scripting.

Features:

- **Simplified Installation Process:** AWS CLI v2 introduces improved installation methods, offering pre-built binaries for Windows, Linux, and macOS. Users no longer need to install Python separately, eliminating concerns about Python versions, virtual environments, or conflicting packages.
- **Enhanced Installation Packages:** Windows users can benefit from an MSI installer, while macOS users can utilize a .pkg installer, ensuring straightforward installation procedures tailored to each operating system.
- **Expanded Configuration Options:** Users can now access new configuration options, including the AWS IAM Identity Center, which replaces AWS SSO, offering more flexibility in managing identity and access.
- **Streamlined Configuration Options:** AWS CLI v2 introduces new mechanisms for configuring credentials, including the addition of the aws configure import command. This command enables users to import credentials directly from .csv files generated in the AWS Console, simplifying the setup process.
- **Interactive Capabilities:** AWS CLI v2 incorporates various interactive features, enhancing user experience and facilitating smoother interactions with AWS services.
    - Enhanced User Interaction: AWS CLI v2 prioritizes user interaction by introducing new features tailored for interactive command-line usage.
    - Server-Side Auto-Completion: Users benefit from server-side auto-completion, which suggests available commands and options as they type, improving command accuracy and efficiency.
    - CLI Auto-Prompt Command: The --cli-auto-prompt command enhances usability by automatically prompting users for input when executing commands, streamlining the interaction process.
    - Wizards: AWS CLI v2 introduces wizards to guide users through complex tasks step-by-step, ensuring smooth execution and minimizing errors during interactive sessions.

AWS-shell

AWS-shell supercharges the AWS Command Line Interface (AWS CLI) for both beginners and veterans.

Features:

- Advanced Auto-Completion: AWS-shell anticipates what you're typing and suggests options, making it faster and easier to enter commands accurately.
- Interactive Documentation: No more searching for manuals! AWS-shell provides real-time explanations for commands as you type, ensuring you use them correctly.
- OS Command Integration: Use familiar commands like "cat" or "ls" directly within AWS-shell, without switching between windows. You can even combine AWS and system commands for powerful workflows.
- Command History Export: AWS-shell keeps track of the commands you've used and lets you export them to a file, making it easy to share or reference them late

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.

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.

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.

AWS Cloud Development Kit (AWS CDK)

What is AWS Cloud Development Kit (AWS CDK)?

The AWS Cloud Development Kit (AWS CDK) is a tool that expedites cloud development by enabling developers to use popular programming languages to define and model their applications' resources in the cloud.

Features:

- Utilize commonly used programming languages to specify application resources and enhance the development process.
- Simplify AWS adoption by leveraging preconfigured constructs with established defaults for setting up cloud resources.
- Develop reusable components tailored to organization-specific needs for security, compliance, and governance.
- Seamlessly create applications, write runtime code, and define resources within your integrated development environment (IDE) without switching tools.
- Reuse constructs encapsulating configuration details and logic for AWS services to define infrastructure efficiently.
- Customize, share, and reuse constructs within your organization or community for faster development and best practices adherence.
- Provision infrastructure through AWS CloudFormation using CFN Resources for complete coverage and updates.
- Model application infrastructure using familiar programming languages like TypeScript, Python, Java, .NET, and Go, and utilize existing IDEs and workflow patterns.
- Deploy both infrastructure and runtime code together by referencing assets in the same project.
- Interact with CDK applications, synthesize CFN templates, visualize stack differences, confirm security changes, and deploy multiple stacks across environments using the AWS CDK CLI.

Use Cases:

- Enhance both infrastructure and business logic by leveraging AWS CDK as the primary framework for defining cloud infrastructure as code.

- Expedite the development process by utilizing AWS CDK to provision common infrastructure patterns quickly, facilitating efficient migration of complex backend infrastructure.
- Integrate seamlessly with CI/CD pipelines to automate the provisioning of AWS services, enabling faster delivery and deployment cycles.
- Utilize Construct Hub to explore and utilize AWS CDK constructs developed by the community, enabling the programmable creation of new microservices.
- Accelerate development and deployment transitions by leveraging tools optimized for the cloud, such as TypeScript, Python, Java, .NET, and Go (in Developer Preview) with AWS CDK's support.

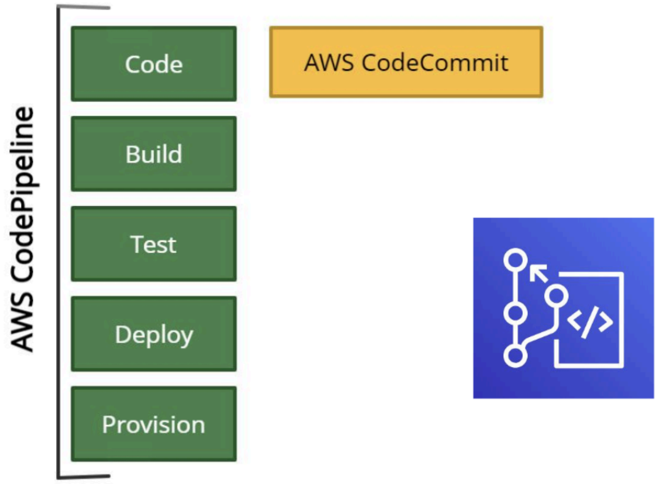
AWS CodeCommit

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.

Features:

- It works with existing Git-based repositories, tools, and commands in addition to AWS CLI commands and APIs.
- CodeCommit repositories support pull requests, version differencing, merge requests between branches, and notifications through emails about any code changes.
- AWS CodeCommit 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.
- It provides high availability, durability, and redundancy.
- It eliminates the need to back up and scale the source control servers.



AWS CodeCommit

Use Cases:

- AWS CodeCommit offers high availability, scalability, and durability for Git repositories.
- AWS CodeCommit provides built-in security features such as encryption, access control, and integration with AWS Identity and Access Management (IAM).
- It enables teams to collaborate effectively on codebases regardless of their geographical locations.
- It integrates seamlessly with other AWS services such as AWS CodePipeline and AWS CodeBuild to automate the CI/CD process.

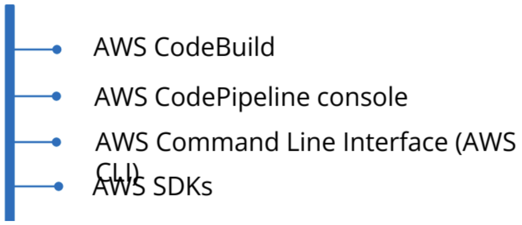
AWS CodeBuild

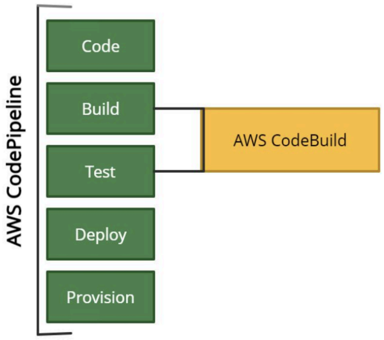
What is AWS CodeBuild?

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

Features:

- 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.





AWS CodeBuild

Use Cases:

- AWS services like AWS Lambda, Amazon S3,
Amazon ECR, and AWS CodeArtifact, enabling developers to deploy applications to AWS cloud services easily.
- It optimizes build performance by automatically provisioning and scaling build resources based on workload demands.
- It offers pre-configured build environments with popular programming languages, runtime versions, and build tools pre-installed.

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.

Features:

● 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

Use Cases:

In-place deployment:

● All the instances in the deployment group are stopped, updated with new revision and started again after the deployment is complete.

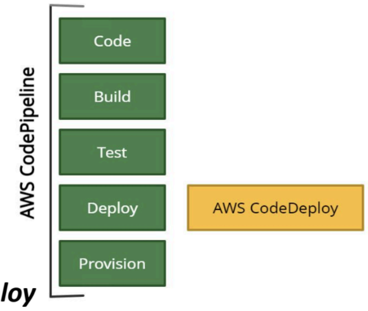
● Useful for EC2/On-premises compute platform.

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.



AWS CodeDeploy

AWS CodePipeline

What is AWS CodePipeline?

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

Features:

- **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 represent a different part of your CI/CD process, they are as follow:
- **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.

– Back to Index –

Use Cases:

- **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.

Amazon API Gateway Machine Learning

Overview:

How Amazon API Gateway is used to train Machine learning model?
Imagine you have a trained model that can make predictions based on data. Here's how AWS can help you make it accessible to applications:

- Client Interaction: Amazon API Gateway acts as a central control point, like a receptionist in a building. It receives requests from user applications.
- Backend Protection: API Gateway keeps the actual model (running on Amazon SageMaker Studio) safe behind a secure wall, just like a secure server room.
- Model Inference: When a request arrives, AWS Lambda (a super efficient worker) steps in. It takes the data from the request and prepares it for the model.
- Response Handling: Lambda sends the data to your trained model in SageMaker Studio. The model then analyzes the data and makes a prediction.
- Client Response: The prediction result goes back to Lambda, which formats it nicely.
- Amazon API Gateway: Finally, API Gateway delivers the prediction from your model back to the application that requested it.

Use Cases:

- **Model Selection:** The tutorial focuses on deploying a binary classification XGBoost model designed for auto insurance fraud detection. This model has been trained using a synthetically generated dataset comprising details on insurance claims and customers, along with a fraud indicator column.
- **Dataset Description:** The dataset used for training contains extracted features related to claims and customers, alongside a label indicating whether each claim is fraudulent or not.
- **Inference Task:** The deployed model's primary task is to predict the probability of a given insurance claim being fraudulent in real-time. This inference process enables quick decision-making regarding the legitimacy of insurance claims.
- **Deployment Objective:** The main goal of the tutorial is to deploy the trained model to a real-time inference endpoint, allowing external applications to interact with it.
- **REST API Exposure:** After deploying the model, the next step involves exposing the inference endpoint through a REST API. This API facilitates the submission of external data payloads for inference, enabling the detection of potential fraud in insurance claims.
- **Role of Machine Learning Engineer:** As the machine learning engineer, your role is to oversee the deployment process and ensure that the model performs accurately in real-time inference scenarios.

Amazon SageMaker

What is Amazon SageMaker?

Amazon SageMaker is a cloud service that allows developers to prepare, build, train, deploy and manage machine learning models.

Features:

- It provides a secure and scalable environment to deploy a model using SageMaker Studio or the SageMaker console.
- It has pre-installed machine learning algorithms to optimize and deliver 10X performance.
- It scales up to petabytes level to train models and manages all the underlying infrastructure.
- Amazon SageMaker notebook instances are created using Jupyter notebooks to write code to train and validate the models.
- Amazon SageMaker gets billed in seconds based on the amount of time required to build, train, and deploy machine learning models.

Use Cases:

Data Extraction & Analysis:
Automate the extraction, processing, and analysis of documents to enhance investigation accuracy and expedite decision-making.

Fraud Detection:
Utilize automation to swiftly detect suspicious transactions, enabling prompt customer alerts and mitigating potential financial losses.

Churn Prediction:
Predict customer churn probability, identify potential abandoners, and implement proactive measures like targeted promotions to enhance retention rates.

Personalized Recommendations:
Enhance customer satisfaction and business growth by providing tailored and unique experiences to customers through personalized recommendations.

– Back to Index –

AWS CloudFormation

What is AWS CloudFormation?

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

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

It provides the capability to reuse the template to set the resources easily and repeatedly. It can be integrated with AWS IAM for security.

It can be integrated with CloudTail to capture API calls as events.

Templates -
A JSON or YAML formatted text file used for building AWS resources.
Stack -
It is a single unit of resources.
Change sets -
It allows checking how any change to a resource might impact the running resources.
Stacks can be created using the AWS CloudFormation console and AWS Command Line Interface (CLI).
Stack updates:
First the changes are submitted and compared with the current state of the stack and only the changed resources get updated.
There are two methods for updating stacks:

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

StackSets are responsible for safely provisioning, updating, or deleting stacks.
Nested Stacks are stacks created within another stack by using the AWS::CloudFormation::Stack resource.
When there is a need for common resources in the template, Nested stacks can be used by declaring the same components instead of creating the components multiple times. The main stack is termed as parent stack and other belonging stacks are termed as child stack, which can be implemented by using ref variable '! Ref'.

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

Price details:

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

Example:

CloudFormation template for creating EC2 instance

```
EC2Instance:
Type: AWS::EC2::Instance
Properties:
ImageId: 1234xyz
KeyName: aws-keypair
InstanceType: t2.micro
SecurityGroups: - !Ref EC2SecurityGroup
BlockDeviceMappings: -
DeviceName: /dev/sda1
Ebs: VolumeSize: 50
```

AWS CloudTrail

What is AWS CloudTrail?

AWS CloudTrail is defined as a global service that permits users to enable operational and risk auditing of the AWS account.
It allows users to view, search, download, archive, analyze, and respond to account activity across the AWS infrastructure.
It records actions as an event taken by a user, role, or an AWS service in the AWS Management Console, AWS Command Line Interface, and AWS SDKs and APIs.

AWS CloudTrail mainly integrates with:

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

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

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

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

There are three types of CloudTrail events:

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

Example of CloudTrail log file:

IAM log file -

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

```
{"Records": [{
 "eventVersion": "1.0", "userIdentity": {
"type": "IAMUser", "principalId":
"PR_ID", "arn": "arn:aws:iam::210123456789:user/Rohit",
"accountId": "210123456789",
"accessKeyId": "KEY_ID",
"userName": "Rohit"
"eventTime": "2021-01-24T21:18:50Z",
"eventSource": "iam.amazonaws.com",
eventName": "CreateUser",
"awsRegion": "ap-south-2",
"sourceIPAddress": "176.1.0.1",
"userAgent": "aws-cli/1.3.2 Python/2.7.5 Windows/7",
"requestParameters": {"userName": "Nayan"},
"responseElements": {"user": {
"createDate": "Jan 24, 2021 9:18:50 PM",
"userName": "Nayan",
 "arn": "arn:aws:iam::128x:user/Nayan",
"path": "/", "userId": "12xyz"
}}
}]}
```

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

Price details:

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

Amazon CloudWatch

What is Amazon CloudWatch?

Amazon CloudWatch is a service that helps to monitor and manage services by providing data and actionable insights for AWS applications and infrastructure resources.
It monitors AWS resources such as Amazon RDS DB instances, Amazon EC2 instances, Amazon DynamoDB tables, and, as well as any log files generated by the applications.

Amazon CloudWatch can be accessed by the following methods:

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

Amazon CloudWatch is used together with the following services:

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

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

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

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

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

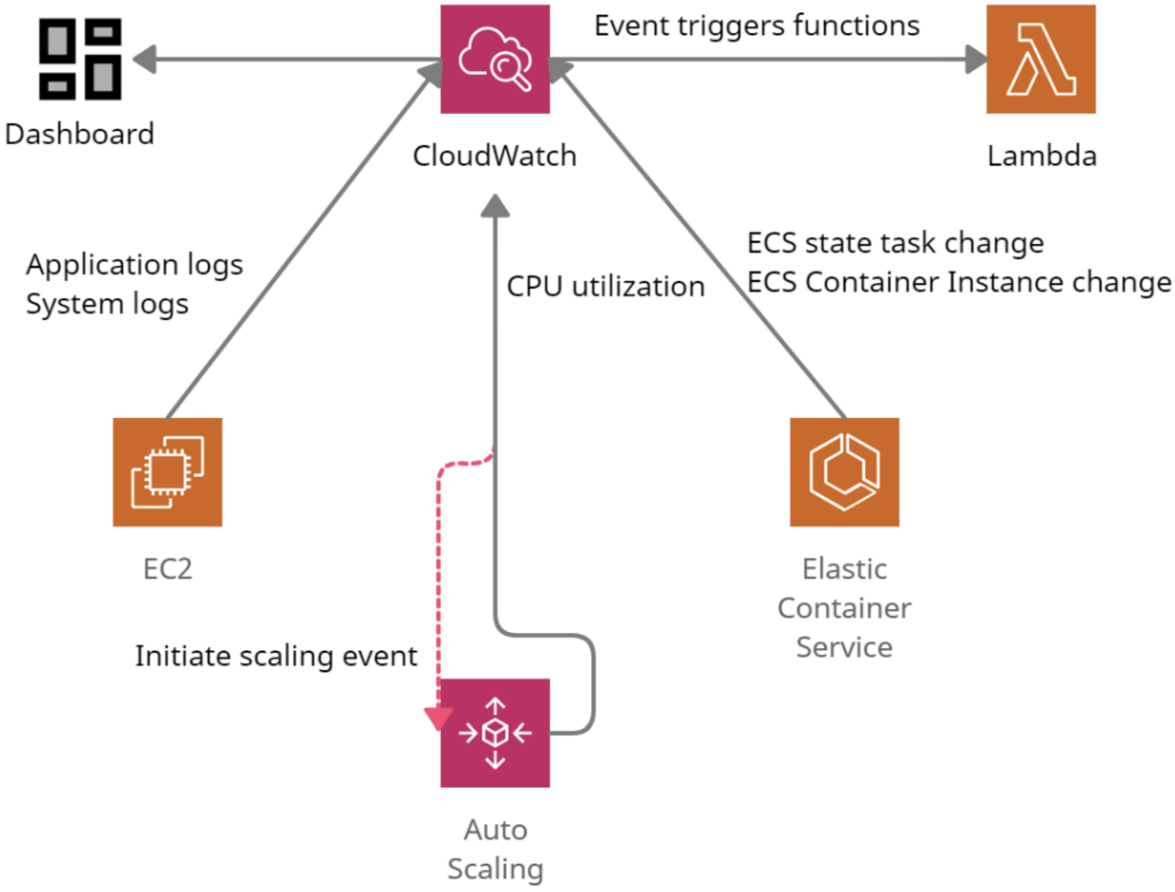
CloudWatch Lambda Insights are used to collect and summarize system-level metrics including CPU time, memory, disk, and network for serverless applications running on AWS Lambda.

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

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

StatsD - supported on both Linux servers and Windows Server
collectd - supported only on Linux servers.

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



Amazon CloudWatch in action

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.

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.

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.

AWS Config

What is AWS Config?

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

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

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

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

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

- Managed Rules
- Custom Rules

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

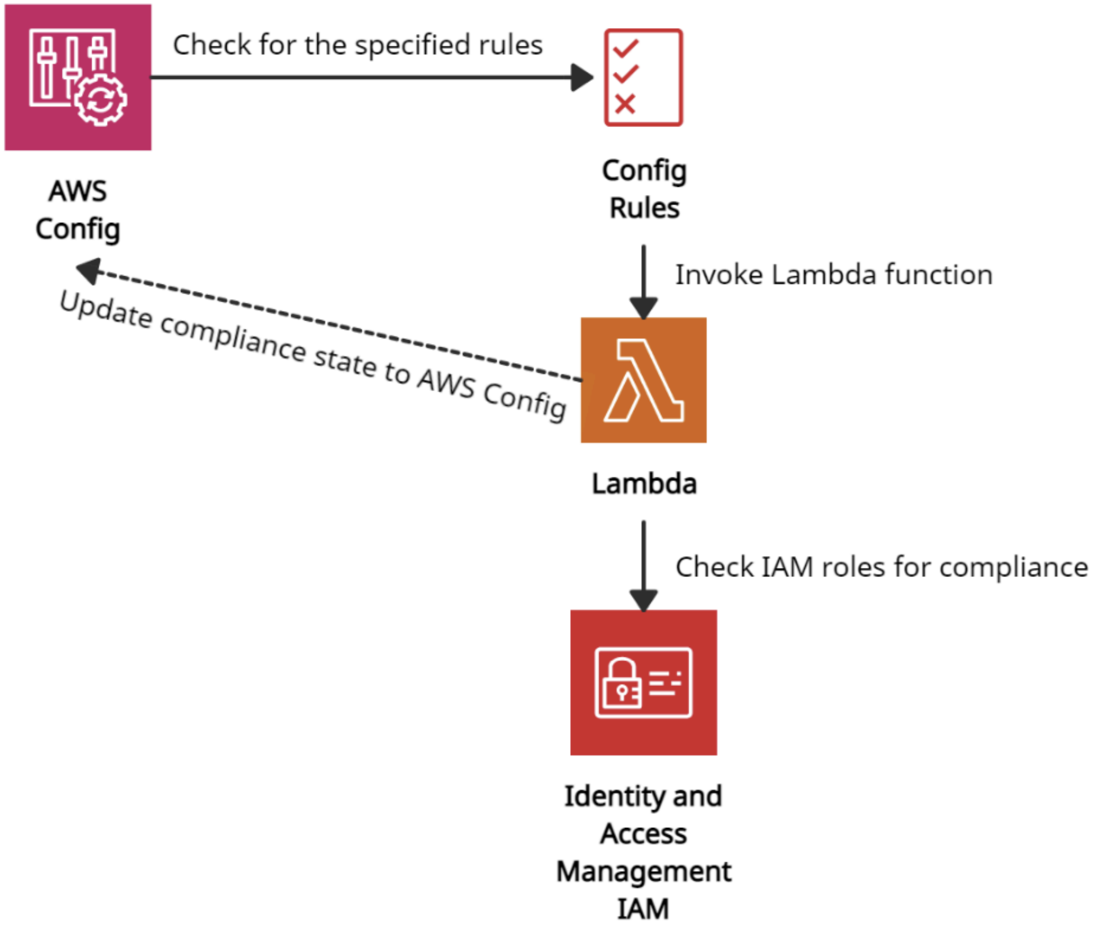
It is also integrated with AWS CloudTrail, which provides a record of user actions or an AWS Service by capturing all API calls as events in AWS Config.

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

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

Use Cases:

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


Check for the specified rules
AWS Config
Config Rules
Invoke Lambda function
Update compliance state to AWS Config
Lambda
Check IAM roles for compliance
Identity and Access Management IAM

AWS Config in action

Price details:

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

Amazon Managed Grafana

What is Amazon Managed Grafana?

Amazon Managed Grafana provides a fully managed solution for Grafana, an analytics platform widely used for querying, visualizing, and setting alerts on metrics, logs, and traces. It offers scalable and secure visualization of operational data, including metrics, logs, and traces.

Features:

- Construct environments tailored to your needs without the hassle of manual provisioning or maintenance.

Develop, bundle, and roll out workspaces effortlessly, ensuring they're configured and managed for optimal performance.

- Operational Data:
- Employ operational data visualization tools to interpret and connect data from various origins.

Analyze and cross-reference operational data from diverse AWS accounts and regions, enabling comprehensive insights.

- Integrate with AWS Security:
- Seamlessly merge with AWS security solutions to align with organizational security protocols and regulatory standards.

Enhance data protection and compliance by integrating with AWS security services.

- Migrate Grafana:
- Transition smoothly from your existing Grafana setup, eliminating the need for a fresh start.
- Transfer data, configurations, and setups seamlessly from your self-managed Grafana environment to Amazon Managed Grafana.

Use Cases:

Unified Observability: Combine metrics, logs, and traces for comprehensive analysis through unified dashboards.
Resource Exploration: Monitor container performance across diverse environments like Amazon EKS, ECS, and self-managed Kubernetes.
Collaborative Troubleshooting: Facilitate real-time dashboard editing and sharing to enhance teamwork in resolving operational issues.
IoT Device Monitoring: Utilize Grafana's adaptable tools to efficiently monitor and analyze data from IoT and edge devices.

AWS Systems Manager

What is AWS Systems manager?

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

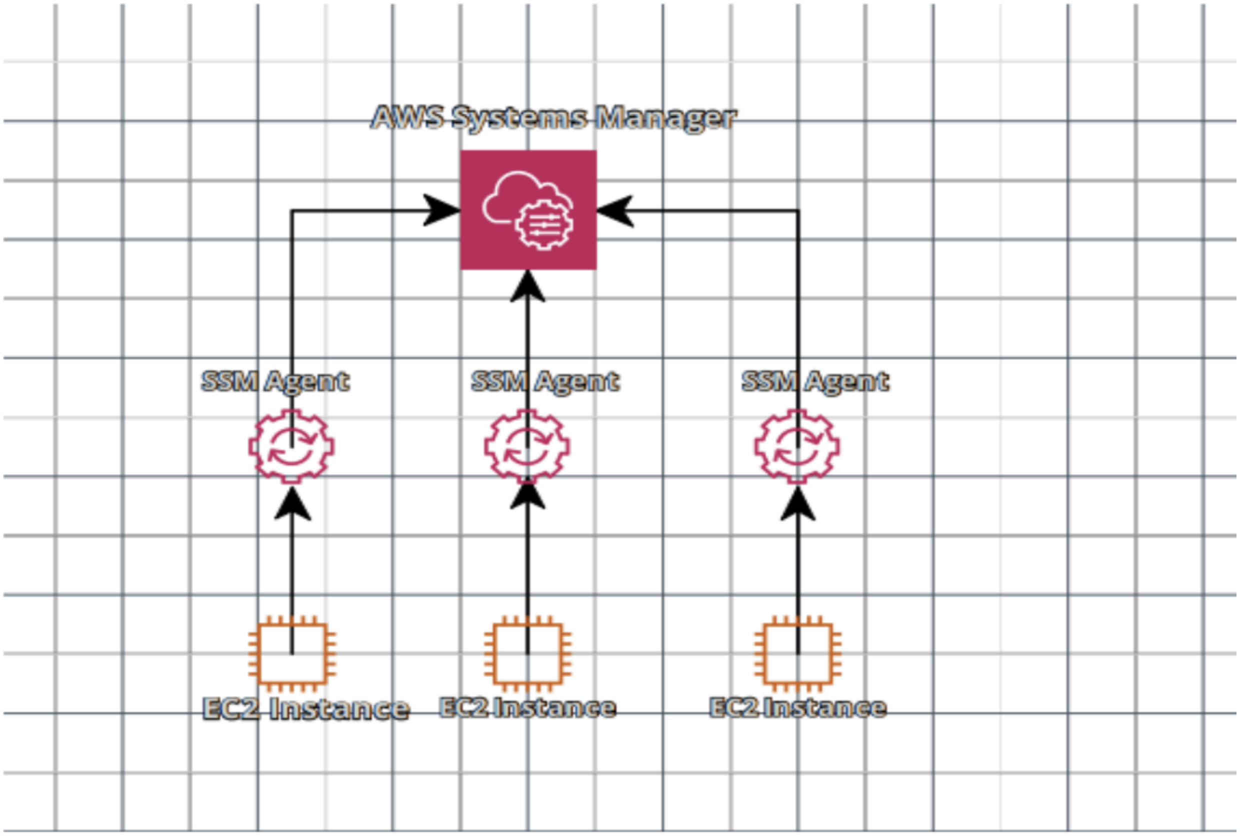
Additionally, we can patch automation for enhanced compliance. This AWS service works for both Windows and Linux operating systems.

Features:

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

How does the System Manager work?

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



Pricing:

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

AWS Application Discovery Service

What is AWS Application Discovery Service?

Amazon Web Services Application Discovery Service (Application Discovery Service) helps you plan application migration projects. It automatically identifies servers, virtual machines (VMs), and network dependencies in your on-premises data centers.

Features:

- Agentless discovery using Amazon Web Services Application Discovery Service Agentless Collector (Agentless Collector), which doesn't require you to install an agent on each host.
- Agent-based discovery using the Amazon Web Services Application Discovery Agent (Application Discovery Agent) collects a richer set of data than agentless discovery, which you install on one or more hosts in your data center.
- Amazon Web Services Partner Network (APN) solutions integrate with Application Discovery Service, enabling you to import details of your on-premises environment directly into Amazon Web Services Migration Hub (Migration Hub) without using Agentless Collector or Application Discovery Agent.

Use cases:

- Discover on-premises server and database inventory
- Map network communication patterns
- Mobilize for migration

Pricing:

You can use the AWS Application Discovery Service to discover your on-premises servers and plan your migrations at no charge.

You only pay for the AWS resources (e.g., Amazon S3, Amazon Athena, or Amazon Kinesis Firehose) that are provisioned to store your on-premises data. You only pay for what you use, as you use it; there are no minimum fees and no upfront commitments.

AWS Application Migration Service

What is AWS Application Discovery Service?

Amazon Web Services Application Discovery Service (Application Discovery Service) helps you plan application migration projects. It automatically identifies servers, virtual machines (VMs), and network dependencies in your on-premises data centers.

Features:

- Agentless discovery using Amazon Web Services Application Discovery Service Agentless Collector (Agentless Collector), which doesn't require you to install an agent on each host.
- Agent-based discovery using the Amazon Web Services Application Discovery Agent (Application Discovery Agent) collects a richer set of data than agentless discovery, which you install on one or more hosts in your data center.
- Amazon Web Services Partner Network (APN) solutions integrate with Application Discovery Service, enabling you to import details of your on-premises environment directly into Amazon Web Services Migration Hub (Migration Hub) without using Agentless Collector or Application Discovery Agent.

Use cases:

- Discover on-premises server and database inventory
- Map network communication patterns
- Mobilize for migration

Pricing:

You can use the AWS Application Discovery Service to discover your on-premises servers and plan your migrations at no charge.

You only pay for the AWS resources (e.g., Amazon S3, Amazon Athena, or Amazon Kinesis Firehose) that are provisioned to store your on-premises data. You only pay for what you use, as you use it; there are no minimum fees and no upfront commitments.

AWS Database Migration Service

What is AWS Database Migration Service?

AWS Database Migration Service is a cloud service used to migrate relational databases from on-premises, Amazon EC2, or Amazon RDS to AWS securely.

It does not stop the running application while performing the migration of databases, resulting in downtime minimization.
It performs homogeneous as well as heterogeneous migrations between different database platforms.
MySQL - MySQL (homogeneous migration)
MySQL - Amazon Aurora (heterogeneous migration)

AWS DMS supports the following data sources and targets engines for migration:
● **Sources**: Oracle, Microsoft SQL Server, PostgreSQL, Db2 LUW, SAP, MySQL, MariaDB, MongoDB, and Amazon Aurora.
● **Targets:** Oracle, Microsoft SQL Server, PostgreSQL, SAP ASE, MySQL, Amazon Redshift, Amazon S3, and Amazon DynamoDB.

It performs all the management steps required during the migration, such as monitoring, scaling, error handling, network connectivity, replicating during failure, and software patching.
AWS DMS with AWS Schema Conversion Tool (AWS SCT) helps to perform heterogeneous migration.

AWS DataSync

What is AWS DataSync?

AWS DataSync is a secure, reliable, managed migration Service that automates the movement of data online between storage systems. AWS DataSync provides the capability to move data between AWS storage, On-premises File Systems, Edge locations, and other Cloud Storage services like Azure. AWS DataSync helps you simplify your migration planning and reduce costs associated with the data transfer.

Features

- Data movement workloads using AWS DataSync support migration scheduling, bandwidth throttling, task filtering, and logging.
- AWS DataSync provides enhanced performance using compression, and parallel transfers for transferring data at speed.
- AWS DataSync supports In-Flight encryption using TLS and encryption at rest.
- AWS DataSync provides capabilities for Data Integrity Verification ensuring that all data is transferred successfully.
- AWS DataSync integrates with AWS Management tools like CloudWatch, CloudTrail, and EventBridge.
- With DataSync, you only pay for the data you transfer without any minimum cost.
- AWS DataSync can copy data to and from Amazon S3 buckets, Amazon EFS file systems, and all Amazon FSx file system types.
- AWS DataSync supports Internet, VPN, and Direct Connect to transfer data between On-premises data centers, Cloud environments & AWS

Use cases

- Application Data Migration residing on On-premises storage systems like Windows Server, NAS file systems, Object storage to AWS.
- Archival of On-premises storage data to AWS to free capacity & reduce costs for continuously investing in storage infrastructure.
- Continuous replication of data present On-premises or on existing Cloud platforms for Data Protection and Disaster Recovery

Best Practices

- In General, when planning a Data Migration, migration tools need to be evaluated, check for available bandwidth for online migration, and understand the source & destination migration data sources.
- For using DataSync to transfer data from On-premises storage to AWS, an Agent needs to be deployed and activated at On-premises locations. Use the Agent's local console as a tool for accessing various configurations

  - System resources
  - Network connectivity
  - Getting Agent activation key
  - View Agent ID & AWS region where the agent is activated
- A common pattern that can be used as a best practice is to use a combination of AWS ● DataSync & AWS Storage Gateway. DataSync can be used to archive On-premises data to ● ● ● AWS while Storage Gateway can be used to access commonly used data at On-premises.
- DataSync effectively manages the transfer of data between different storage devices without you having to write migration scripts or keep track of data that is transferred
- AWS DataSync can also be triggered using a Lambda function in case a migration schedule is not defined
- Data transfers between AWS services like S3 -> S3 or S3 -> EFS do not require the DataSync Agent. It is used only for data transfers from On-premises to AWS
- You pay 1.25 Cents per GigaByte of data transferred.

AWS Schema Conversion Tool (AWS SCT)

Overview:

When it comes to migrating databases, AWS offers two convenient methods for seamlessly converting your database schema:

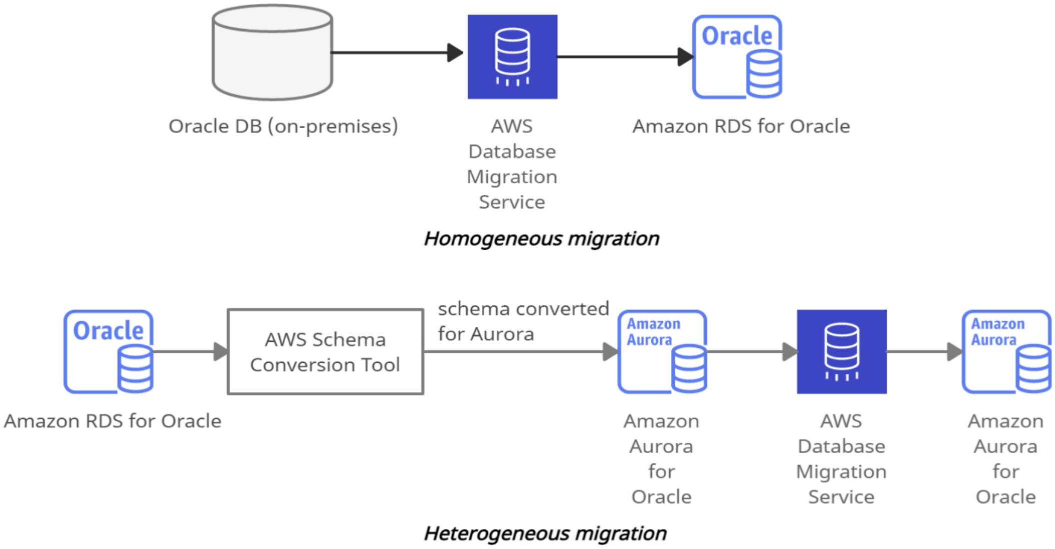
Fully Managed DMS Schema Conversion: This integrated solution available on the AWS DMS console streamlines the process. Simply specify your old database and the new one, and it automatically translates the schema and most objects like views and functions. For any complex elements that can't be converted automatically, clear instructions are provided for manual handling.

For Enhanced Control, Try the AWS Schema Conversion Tool (SCT): Download this tool to your local device for added flexibility. While it performs the same fundamental conversion as DMS, SCT goes a step further by analyzing your application code. It can rectify embedded SQL statements to function in the new environment and even optimize legacy functions for cloud compatibility. Additionally, if you're transitioning from another data warehouse to Amazon Redshift, SCT comes equipped with built-in data migration agents to manage that aspect seamlessly.

In summary, AWS offers versatile options for schema conversion during database migrations, catering to both beginners and experienced users. You can opt for the user-friendly managed service or utilize the downloadable tool for greater control and customization.

How it works?

The combination of AWS DMS and AWS SCT facilitates heterogeneous migration processes effectively.



AWS Database Migration Service

Features:

- Make your database migrations hassle-free with automated schema analysis, recommendation generation, and scalable conversion capabilities.
- Achieve seamless compatibility across a wide array of widely-used databases and analytics platforms, serving as both the origin and destination engines. These include Oracle, SQL Server, PostgreSQL, and MySQL.
- Witness significant decreases in manual efforts and resource consumption, leading to potential time savings of several weeks or even months.

Use Cases:

- Replicate a database schema from a source to a destination.
- Transform a database or data warehouse schema to align with the target environment.
- Evaluate the complexity of converting a database schema.
- Assess potential limitations for running on Amazon RDS by analyzing a database.
- Determine the feasibility of downgrading a license by examining a database.
- Adapt embedded SQL code within an application for compatibility with the target database.
- Transfer data warehouse data to Amazon Redshift seamlessly.

AWS Snow Family

What is AWS Snow Family?

Easily transfer large amounts of data to AWS using specialized Snow devices. These devices are available for lease and are optimized for seamless petabyte-scale data migration. Engineered with robustness and security in mind, Snow devices undergo thorough field testing to ensure they can withstand harsh conditions while preserving data integrity. Offering versatility, they come in different configurations suitable for environments with space or weight constraints, providing both portability and adaptable networking capabilities. Whether you're handling data at the edge or moving substantial volumes to AWS, Snow devices offer a dependable solution for your data transfer needs.

Features:

- **On-board computing:** Snow Family devices support Amazon EC2 instances, AWS IoT Greengrass functions, and Kubernetes deployments for edge data processing.
- **End-to-end tracking:** Each device features E-Ink shipping labels for easy tracking and label updates via Amazon SNS and text messages.
- **Simple management and monitoring:** AWS OpsHub GUI simplifies device setup and management, facilitating rapid deployment of edge workloads and data migration.
- **Encryption:** Data transferred to Snow Family devices is encrypted with 256-bit keys managed by AWS KMS, ensuring secure transit without storing keys on the device.
- **Secure erasure:** After data migration, AWS performs NIST-compliant software erasure for secure data disposal.
- **NFS endpoint:** Snow Family devices support NFS mount points for seamless integration with on-premises servers and file-based applications.
- **Anti-tamper & Tamper-evident:** Equipped with a TPM, Snow devices undergo post-use inspection to maintain integrity and data confidentiality.

AWS Snow Family service models:

AWS Snowcone: Your compact data companion, perfect for tight spaces or demanding conditions. Choose between lightning-fast SSD or capacious HDD options to tailor your storage to your needs.

AWS Snowball: The ultimate data transporter, with "Compute Optimized" for on-device processing and "Storage Optimized" for moving massive data volumes effortlessly. Built tough and secure, Snowball is your trusted ally for any data challenge.

AWS Transfer Family

What is AWS Transfer Family?

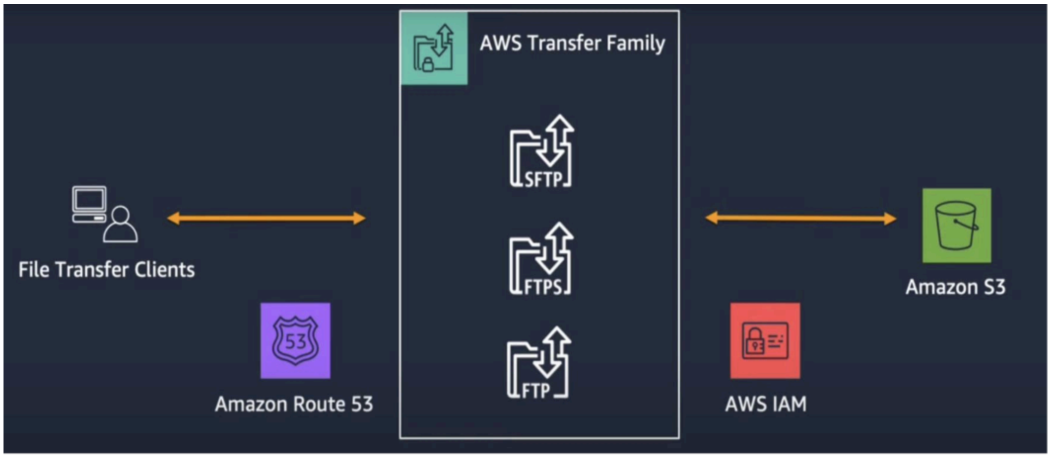
AWS Transfer Family is a fully managed & secure service that enables transfer of files using SFTP, FTPS & FTP. The destination storage services to which files are transferred are S3, and EFS. It helps you to seamlessly migrate File Transfer workloads to AWS without having any impact on existing application integrations or configuration.

Features

- AWS Transfer Family provides a fully managed endpoint for transferring files into and out of S3, EFS.
- The Secure File Transport Protocol (SFTP) is a file transfer provided over SSH.
- File Transfer Protocol over SSL (FTPS is an FTP over a TLS-encrypted channel.
- Plain File Transfer Protocol (FTP) does not require a secure channel for transferring files.
- AWS Transfer Family exhibits high availability across the globe.
- AWS Transfer Family provides compliance with regulations within your Region.
- Using a pay-as-you-use model, the AWS Transfer Family service becomes cost-effective and is simple to use.
- AWS Transfer Family has the ability to use custom Identity Providers using AWS API Gateway & Lambda.

Use cases

- IAM Roles are used to grant access to S3 buckets for file transfer clients in a secure way.
- Users can use Route 53 to migrate an existing File Transfer hostname for use in AWS.
- SFTP & FTPS protocols can be set up to be accessible from the public internet while FTP is limited for access from inside a VPC using VPC endpoints.



AWS Documentation

Amazon CloudFront

What is Amazon CloudFront?

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

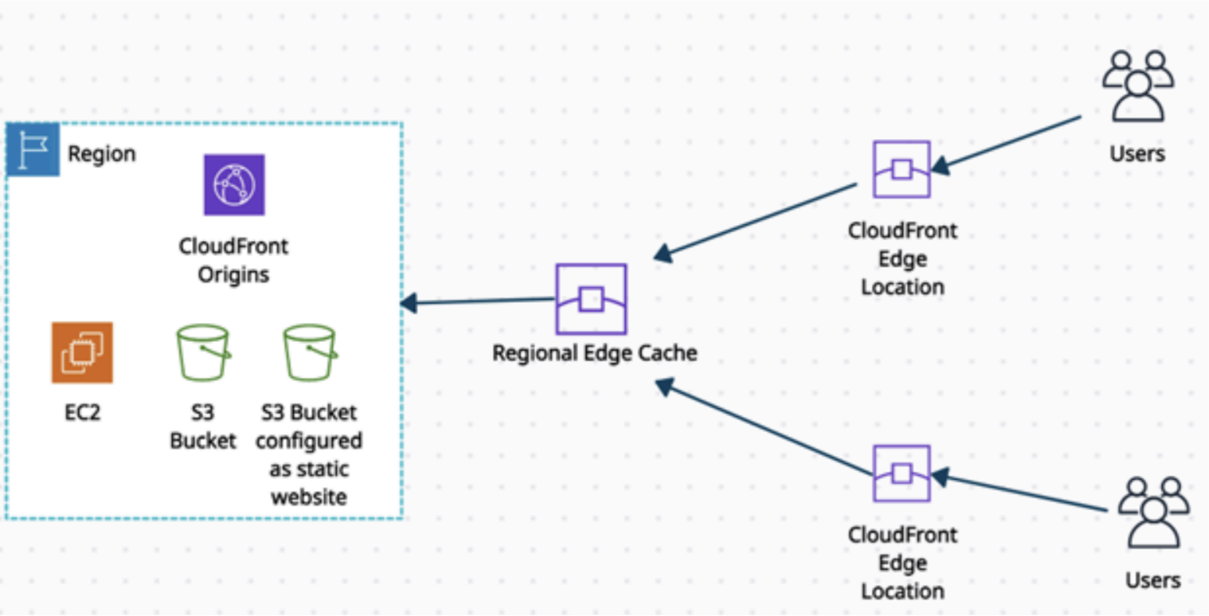
It uses edge locations (a network of small data centers) to cache copies of the data for the lowest latency. If the data is not present at edge locations, the request is sent to the source server, and data gets transferred from there.

It is integrated with AWS services such as

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

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

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



CloudFront Overview

It provides the programmable and secure edge CDN computing feature through

AWS Lambda@Edge.

- It provides operations such as dynamic origin load-balancing, custom bot-management computationally, or building serverless origins.
- It has a built-in security feature to protect data from side-channel attacks such as Spectre and Meltdown.
- Field-level encryption with HTTPS - data remains encrypted throughout starting from the upload of sensitive data. --Back to Index-- 85
- AWS Shield Standard - against DDoS attacks.
- AWS Shield Standard + AWS WAF + Amazon Route53 - against more complex attacks than DDoS.

Amazon CloudFront Access Controls:

Signed URLs:

- Use this to restrict access to individual files.

Signed Cookies:

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

Geo Restriction:

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

Origin Access Identity (OAI):

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

Pricing Details:

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

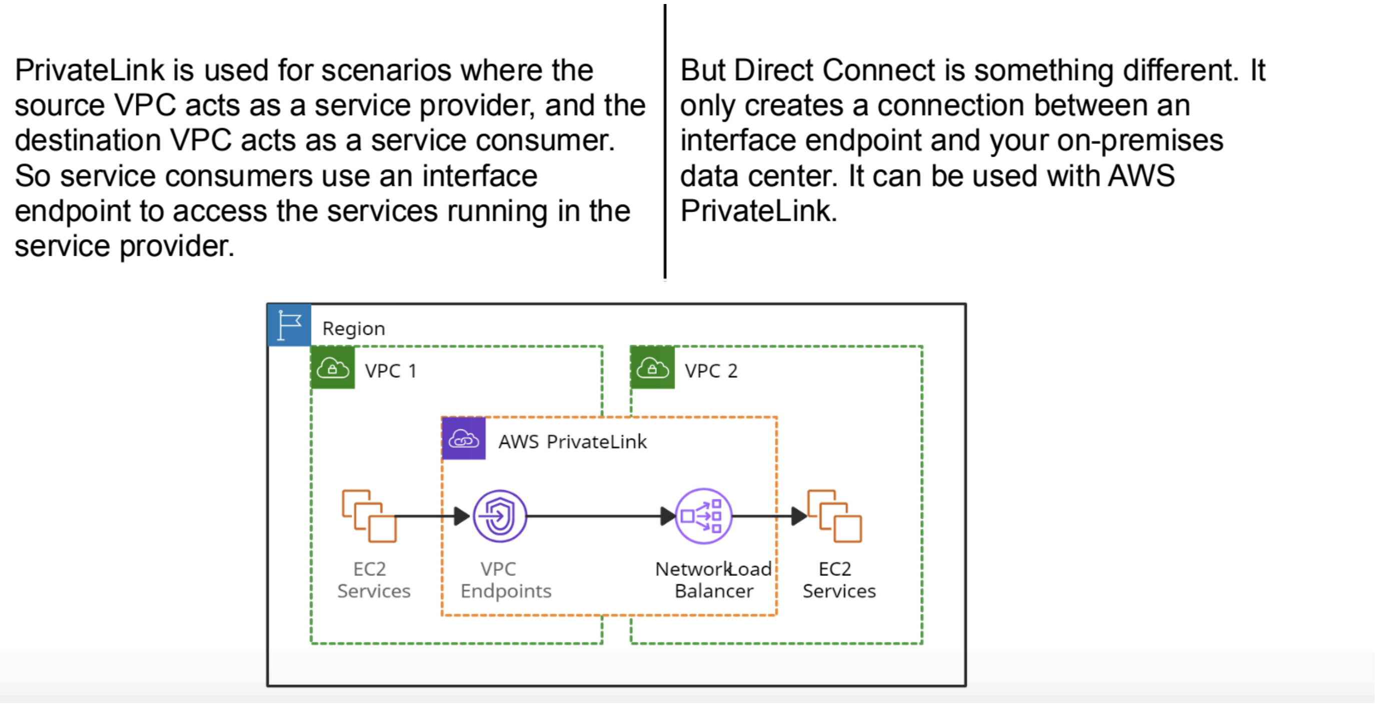
AWS PrivateLink

What is AWS PrivateLink?

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

Interface endpoints

- It serves as an entry point for traffic destined to an AWS service or a VPC endpoint service. Gateway endpoints
- It is a gateway in the route-table that routes traffic only to Amazon S3 and DynamoDB.



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

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

Features:

- It is integrated with AWS Marketplace services so that the services can be directly attached to the endpoint.

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

Pricing details:

- PrivateLink is charged based on the use of endpoints.

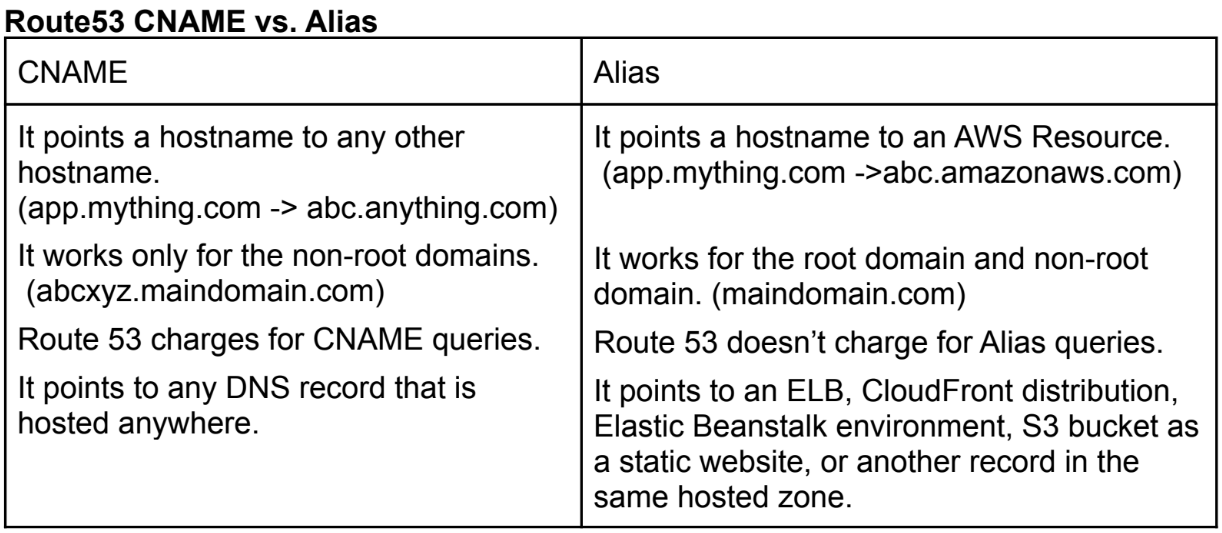
Amazon Route 53

What is Amazon Route 53?

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

Route 53 hosted zone is a collection of records for a specified domain that can be managed together.
There are two types of zones:
● Public host zone – It determines how traffic is routed on the Internet.
● Private hosted zone – It determines how traffic is routed within VPC.

Route 53 TTL (seconds):
● It is the amount of time for which a DNS resolver creates a cache information about the records and reduces the query latency.
● Default TTL does not exist for any record type but always specifies a TTL of 60 seconds or less so that clients/users can respond quickly to changes in health status.



Route53 CNAME vs. Alias

CNAME	Alias
It points a hostname to any other hostname. (app.mything.com -> abc.anything.com)	It 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)
Route 53 charges for CNAME queries.	Route 53 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.

The most common records supported in Route 53 are:

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

Other supported records are:

- CAA (certification authority authorization)
- MX (mail exchange record)

- NAPTR (name authority pointer record)
- NS (name server record)
- PTR (pointer record)
- SOA (start of authority record)
- SPF (sender policy framework)
- SRV (service locator)
- TXT (text record)

Route 53 Routing Policies:

Simple:

- It is used when there is a need to redirect traffic to a single resource.
- It does not support health checks. Weighted:
- It is similar to simple, but you can specify a weight associated with resources.
- It supports health checks.

Failover:

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

Geo-location:

- It routes traffic to the closest geographic location you are in. Geo-proximity:
- It routes traffic based on the location of resources to the closest region within a geographic area.

Latency based:

- It routes traffic to the destination that has the least latency. Multi-value answer:
- It distributes DNS responses across multiple IP addresses.
- If a web server becomes unavailable after a resolver cache a response, a user can try up to eight other IP addresses from the response to reduce downtime.

Use cases:

- When users try to register a domain with Route 53, it becomes the trustworthy DNS server for that domain and creates a public hosted zone.
- Users can have their domain registered in one AWS account and the hosted zone in another AWS account.
- For private hosted zones, the following VPC settings must be ‘true’:
  - enableDnsHostname.
  - enableDnsSupport.
- Health checks can be pointed at:
  - Endpoints (can be IP addresses or domain names.)

  - Status of other health checks.
  - Status of a CloudWatch alarm.
- Route53 as a Registrar: A domain name registrar is an organization that manages the reservation of Internet domain names.
- Domain Registrar != DNS

Price details:

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

AWS VPC

What is AWS VPC?

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

Security Groups:

Default Security Groups:-

Inbound rule - Allows all inbound traffic

Outbound rule - Allows all outbound traffic

Custom Security Groups:-

(by default) Inbound rule - Allows no inbound traffic

Outbound rule - Allows all outbound traffic

Network ACLs (access control list):

Default Network ACL:-

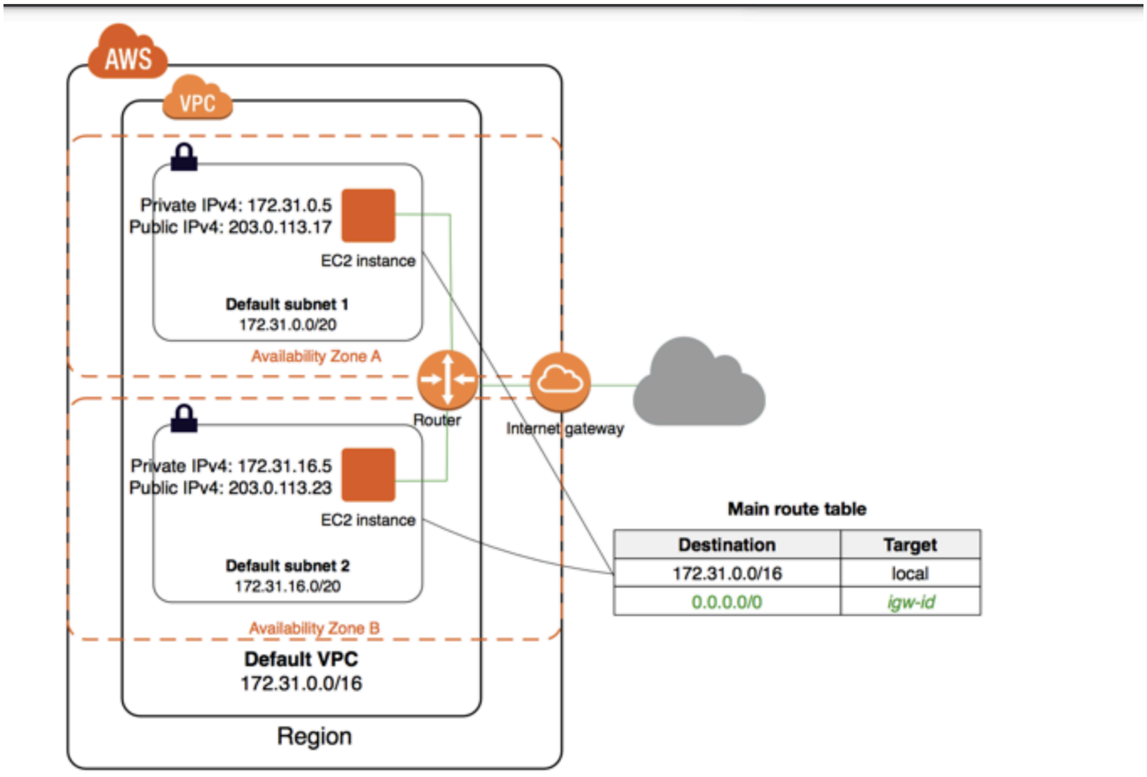
Inbound rule - Allows all inbound traffic

Outbound rule - Allows all outbound traffic

Custom Network ACL:-

(by default) Inbound rule - Denies all inbound traffic

Outbound rule - Denies all outbound traffic



Main route table

Destination	Target
172.31.0.0/16	local
0.0.0.0/0	*igw-id*

VPC Architecture

Components of VPC:

Subnets

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

Route Tables

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

NAT Devices

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

NAT Instance

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

NAT Gateway

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

DHCP Options Set:

- DHCP stands for Dynamic Host Configuration Protocol.

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

PrivateLink

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

Endpoints

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

Types of Endpoints

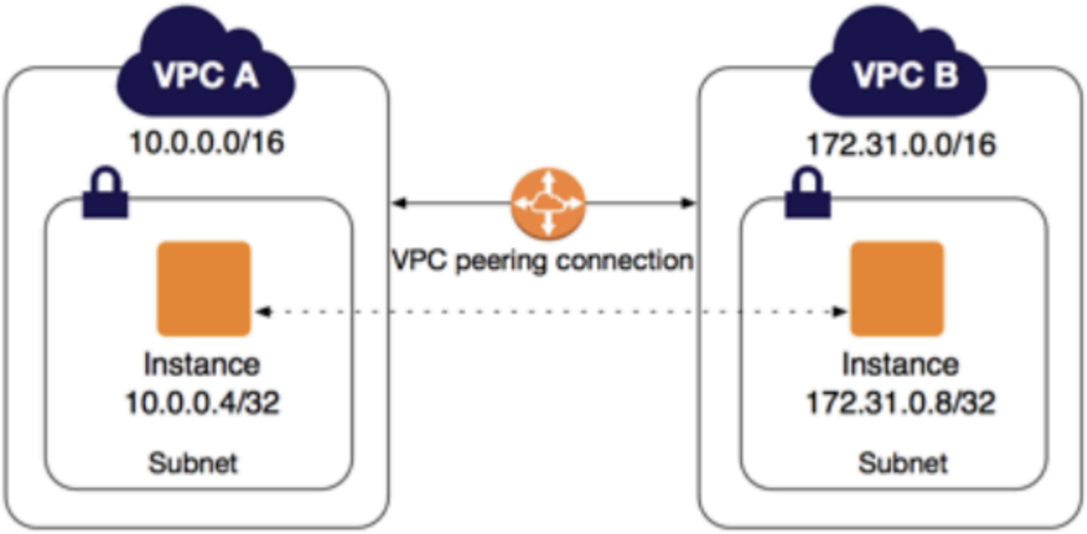
- **Interface Endpoints**
  - o It is an entry point for traffic interception.
  - o It will route the traffic to the service that you configure.
  - o It will use an ENI with a private IP address.
  - o For Example: it will allow instances to connect to Amazon Kinesis through interface endpoint.

- **Gateway Load balancer Endpoints**
  - o It is an entry point for traffic interception.
  - o It will route the traffic to the service that you configure.
  - o It will use load balancers to route the traffic.
  - o For Example Security Inspection.

- **Gateway Endpoints**
  - o It is a gateway that you defined in Route Table as a Target.
  - o And the destination will be the supported AWS Services.
  - o Amazon S3, DynamoDB supports Gateway Endpoint.
- **Egress Only Internet Gateway**
- An egress-only internet gateway is designed only for IPv6 address communications.
- It is a highly available, horizontally scaled component which will allow outbound only rule for IPv6 traffic.
- It will not allow inbound connection to your EC2 Instances.

VPC Peering:

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



VPN

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

AWS Site-to-Site VPN

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

AWS Client VPN

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

Use Cases:

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

Pricing:

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

AWS IAM

What is Identity Access and Management?

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

Principal:

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

IAM User & Root User

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

IAM Group

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

IAM Role

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

IAM Policies

- It decides what level of access an Identity or AWS Resource will possess.

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

IAM Security Best Practices:

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

Pricing:

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

AWS Key Management Service

What is AWS Key Management Service?

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

Customer master keys (CMKs):

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

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

Customer managed CMKs:

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

AWS managed CMKs:

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

Envelope encryption

is the method of encrypting plain text data with a data key and then encrypting the data key under another key. Envelope encryption offers several benefits:

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

Features:

- The automatic rotation of master keys generated in AWS KMS once per year is done without the need to re-encrypt previously encrypted data.

● Using AWS CloudTrail, each request to AWS KMS is recorded in a log file that is delivered to the specified Amazon S3 bucket. Log information includes details of the user, time, date, API action, and the key used.
● This service automatically scales as the encryption grows.
● For the high availability of data and keys, KMS stores multiple copies of an encrypted version of keys.

Benefits:

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

Price details:

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

Amazon Macie

What is Amazon Macie?

Amazon Macie is a data security solution that employs machine learning algorithms and pattern recognition techniques to identify and safeguard sensitive data. By leveraging machine learning and pattern matching capabilities, Amazon Macie not only detects sensitive data but also offers insights into potential data security threats. Additionally, it facilitates automated measures to mitigate these risks, enhancing overall data protection.

Features:

- Implement automated processes for detecting sensitive data on a large scale.
- Obtain cost-effective insights into the presence of sensitive data within your Amazon S3 storage.
- Evaluate the security posture and access permissions of your Amazon S3 bucket inventory.
- Minimize response time by receiving actionable reports on identified sensitive data within Amazon S3.

Use Cases:

- Enhance your data security stance by identifying and addressing sensitive data scattered throughout your S3 environment through automated risk mitigation measures.
- Ensure compliance with regulatory requirements by regularly scanning data to ensure sensitive information is identified and safeguarded.
- Safeguard sensitive data throughout the migration process by verifying its protection status during data transfer.
- Improve oversight of critical business data by implementing automated monitoring mechanisms to continuously track sensitive data stored in S3 buckets.

Pricing:

- Amazon Macie uses ML and pattern matching to detect sensitive data and automate protection and tracks objects for automated anomaly detection.
- Pricing depends on S3 bucket count, data volume inspected, and extent of sensitive data examination.
- It continuously tracks S3 bucket count for inventory and monitoring.
- Monitors data quantity for automated and targeted sensitive data discovery.

AWS Secrets Manager

What is AWS Secrets Manager?

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

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

Accessing Secrets Manager:

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

Secret rotation is available for the following Databases:

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

Features:

- It provides security and compliance facilities by rotating secrets safely without the need for code deployment.
- With Secrets Manager, IAM policies and resource-based policies can assign specific permissions for developers to retrieve secrets and passwords used in the development environment or the production environment.

● Secrets can be secured with encryption keys managed by AWS Key Management Service (KMS).
● It integrates with AWS CloudTrail and AWS CloudWatch to log and monitor services for centralized auditing.

Use cases:

● Store sensitive information as part of the encrypted secret value, either in the SecretString or SecretBinary field.
● Use a Secrets Manager open-source client component to cache secrets and update them only when there is a need for rotation.
● When an API request quota exceeds, the Secrets Manager throttles the request and returns a 'ThrottlingException' error. To resolve this, retry the requests.
● It integrates with AWS Config and facilitates tracking of changes in Secrets Manager.

Price details:

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

AWS Shield

What is AWS Shield?

AWS Shield is a fully managed service safeguarding AWS-hosted applications from DDoS attacks. For enhanced protection, AWS Shield Advanced offers customized defense mechanisms, utilizing exabyte-scale detection across AWS infrastructure. It automatically detects and mitigates advanced network-level DDoS events.

Tailor your application's defense against DDoS threats by integrating with the Shield Response Team (SRT) protocol or AWS Web Application Firewall (WAF), allowing you to adapt strategies to your application's needs.

Additionally, AWS Shield offers visibility, insights, and cost-effective solutions for mitigating DDoS events impacting your AWS resources, aiding in effective threat management and response.

Features:

- Automatically filter out malicious traffic at designated layers to shield applications and APIs from SYN floods, UDP floods, or similar reflection attacks.
- Implement measures to reduce application downtime and latency by employing inline mitigations like deterministic packet filtering and priority-based traffic shaping to thwart basic network-layer attacks effectively.
- Leverage AWS Shield's capabilities to monitor and safeguard up to 1,000 resource types, enabling automatic detection, mitigation, or protection for each resource type per AWS account.

AWS Shield is a fully managed service designed to defend against distributed denial of service (DDoS) attacks targeting applications hosted on AWS. It offers proactive detection and automatic mitigation measures to reduce application downtime and latency. With AWS Shield, users can benefit from DDoS protection without the need to involve AWS Support. The service is available in two tiers: Standard and Advanced.

AWS Shield Standard:

- Automatically defends against common network and transport layer DDoS attacks on AWS-hosted applications.
- Provides comprehensive availability protection in conjunction with Amazon CloudFront and Amazon Route 53 against known infrastructure attacks.
- Offers static threshold DDoS protection for AWS services and utilizes network flow monitoring for real-time threat detection.
- Employs automated mitigation techniques like deterministic packet filtering and priority-based traffic shaping.

AWS Shield Advanced:

- Delivers enhanced protection against attacks targeting applications across various AWS resources.
- Provides additional detection and mitigation capabilities for large and sophisticated DDoS attacks.
- Offers near real-time visibility into attacks and integrates seamlessly with AWS Web Application Firewall (WAF) for heightened security.
- Grants access to the AWS Shield Response Team (SRT) for round-the-clock support and manual mitigation of complex attacks.
- Features tailored detection based on application traffic patterns and health-based detection for quicker response.
- Provides automatic application layer DDoS mitigation and proactive engagement from the SRT during DDoS events.
- Offers DDoS cost protection and enables resource grouping for customized mitigation strategies.
- Ensures complete visibility into DDoS attacks with detailed diagnostics and notifications through Amazon CloudWatch and AWS WAF.

Pricing:

- AWS Shield Standard provides free protection against common network and transport layer DDoS attacks for all AWS users.
- AWS Shield Advanced, a paid service, offers enhanced protection for internet-facing applications on EC2, ELB, CloudFront, Global Accelerator, and Route 53.
- While Shield Advanced is available to all customers, access to the AWS Shield Response Team necessitates Enterprise or Business Support levels of AWS Premium Support. It involves a 1-year subscription commitment with a monthly fee and usage fee based on data transfer out from CloudFront, ELB, EC2, and Global Accelerator.

AWS WAF

What is AWS WAF?

AWS WAF stands for Amazon Web Services Web Application Firewall. It is a managed service provided by AWS that helps protect web applications from common web exploits that could affect application availability, compromise security, or consume excessive resources.
AWS WAF provides an additional layer of security for your web applications, helping to protect them from common web vulnerabilities and attacks such as SQL injection, cross-site scripting (XSS), and distributed denial-of-service (DDoS) attacks.

Features:

- 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.

– Back to Index –

AWS Backup

What is AWS Backup?

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

Features:

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

Use cases:

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

Price details:

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

AWS EBS - Elastic Block Store

What is AWS EBS?

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

Types of EBS:

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

Features:

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

- EBS detached from an EC2 instance and attached to another one quickly.

Key Points to Remember:

- **Backup/Migration:** To move a volume across AZs, you first need to take a snapshot.
- **Provisioned capacity:** capacity needs to be provisioned in advanced (GBs & IOPS)
- You can increase the capacity of the drive over time.
- It can be detached from an EC2 instance and attached to another one quickly.
- It's locked to **Single Availability Zone (AZ)**
- The default volume type is General Purpose SSD (gp2)
- EBS Volume can be mounted parallely using RAID Settings:

○ RAID 0 (increase performance)

○ RAID 1 (increase fault tolerance)

- It's a network drive (i.e. not a physical drive).
- Unencrypted volume can be encrypted using an encrypted snapshot
- Snapshot of the encrypted volume is encrypted by default.
- When you share an encrypted snapshot, you must also share the customer-managed CMK used to encrypt the snapshot.

Pricing:

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

EBS vs Instance Store

Instance Store (ephemeral storage) :

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

EBS :

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

AWS EFS - Elastic File Storage

What is AWS EFS?

Amazon Elastic File System (Amazon EFS) provides a scalable, fully managed elastic distributed file system based on NFS. It is persistent file storage & can be easily scaled up to petabytes. It is designed to share parallelly with thousands of EC2 instances to provide better throughput and IOPS.

It is a regional service automatically replicated across multiple AZ's to provide High Availability and durability.

Types of EFS Storage Classes:

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

EFS Access Modes :

1) Performance Modes:

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

2) Throughput Modes :

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

Features:

- Fully Managed and Scalable, Durable, Distributed File System (NFSv4)
- Highly Available & Consistent low latencies. (EFS is based on SSD volumes)
- POSIX Compliant (NFS) Distributed File System.
- EC2 instances can access EFS across AZs, regions, VPCs & on-premises through AWS Direct Connect or AWS VPN.
- Provides EFS Lifecycle Management for the better price-performance ratio
- It can be integrated with AWS Datasync for moving data between on-premise to AWS EFS
- Supported Automatic/Schedule Backups of EFS (AWS Backups)
- It can be integrated with CloudWatch & CloudTrail for monitoring and tracking.
- EFS supports encryption at transit(TLS) and rest both. (AWS Key Management Service (KMS))
- Different Access Modes: Performance and Throughput for the better cost-performance tradeoff.
- EFS is more expensive than EBS.

- Once your file system is created, you cannot change the performance mode
- Not suitable for boot volume & highly transactional data (SQL/NoSQLdatabases)
- Read-after-write consistency for data access.
- Integrated with IAM for access rights & security.

Use Cases: (Sharing Files Across instances/containers)

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

Best Practices:

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

Pricing:

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

– Back to Index –

Amazon S3

What is Amazon S3?

S3 stands for Simple Storage Service. Amazon S3 is object storage that allows us to store any kind of data in the bucket. It provides availability in multiple AZs, durability, security, and performance at a very low cost. Any type of customer can use it to store and protect any amount of data for use cases, like static and dynamic websites, data analytics, and backup.

Basics of S3?

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

Properties of Amazon S3.

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

Permissions & Management.

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

Charges:

You will be charged based on multiple factors:

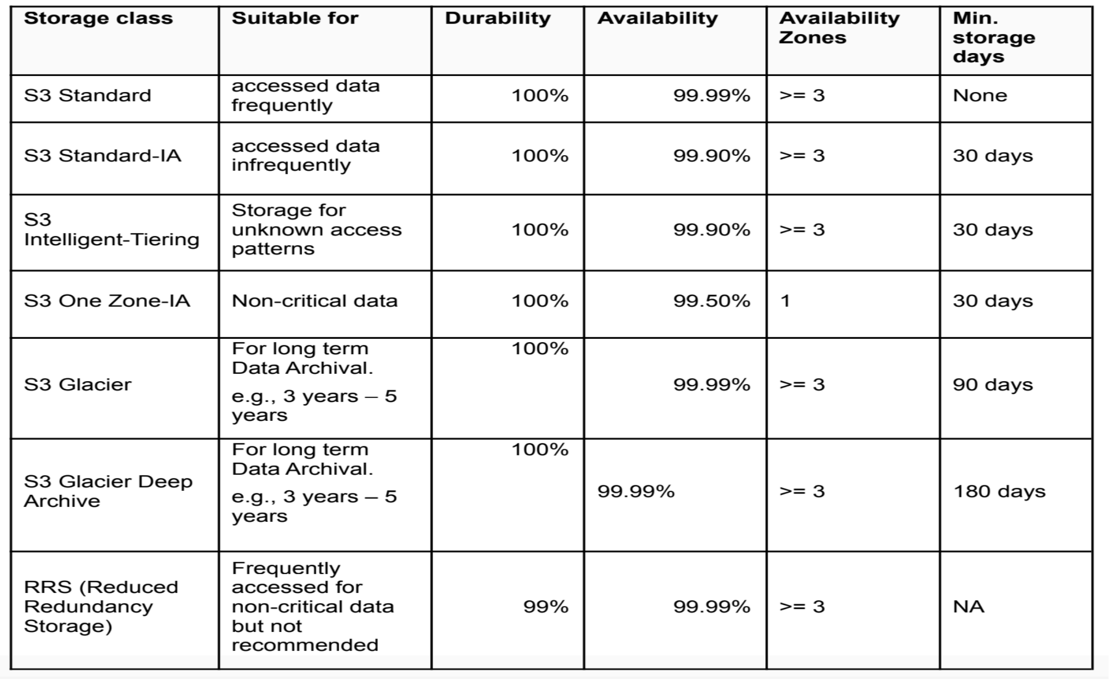
- Storage
- Requests
- Storage Management Pricing (Life Cycle Policies)
- Transfer Acceleration
- Data Transfer Pricing

Miscellaneous Topic

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

Storage Class/Pricing model of S3

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



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

Amazon S3 Glacier

What is Amazon S3 Glacier?

Amazon S3 Glacier is a web service with vaults that offer long-term data archiving and data backup. It is the cheapest S3 storage class and offers 99.999999999% of data durability. It helps to retain unlimited data like photos, videos, documents as TAR or ZIP file, data lakes, analytics, IoT, machine learning, and compliance data. S3-Standard, S3 Standard-IA, and S3 Glacier storage classes, objects, or data are automatically stored across availability zones in a specific region.

S3 Glacier provides the following data retrieval options:

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

Features:

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

Use Cases:

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

Price details:

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