

Amazon Direct Connect gateway

Features:

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

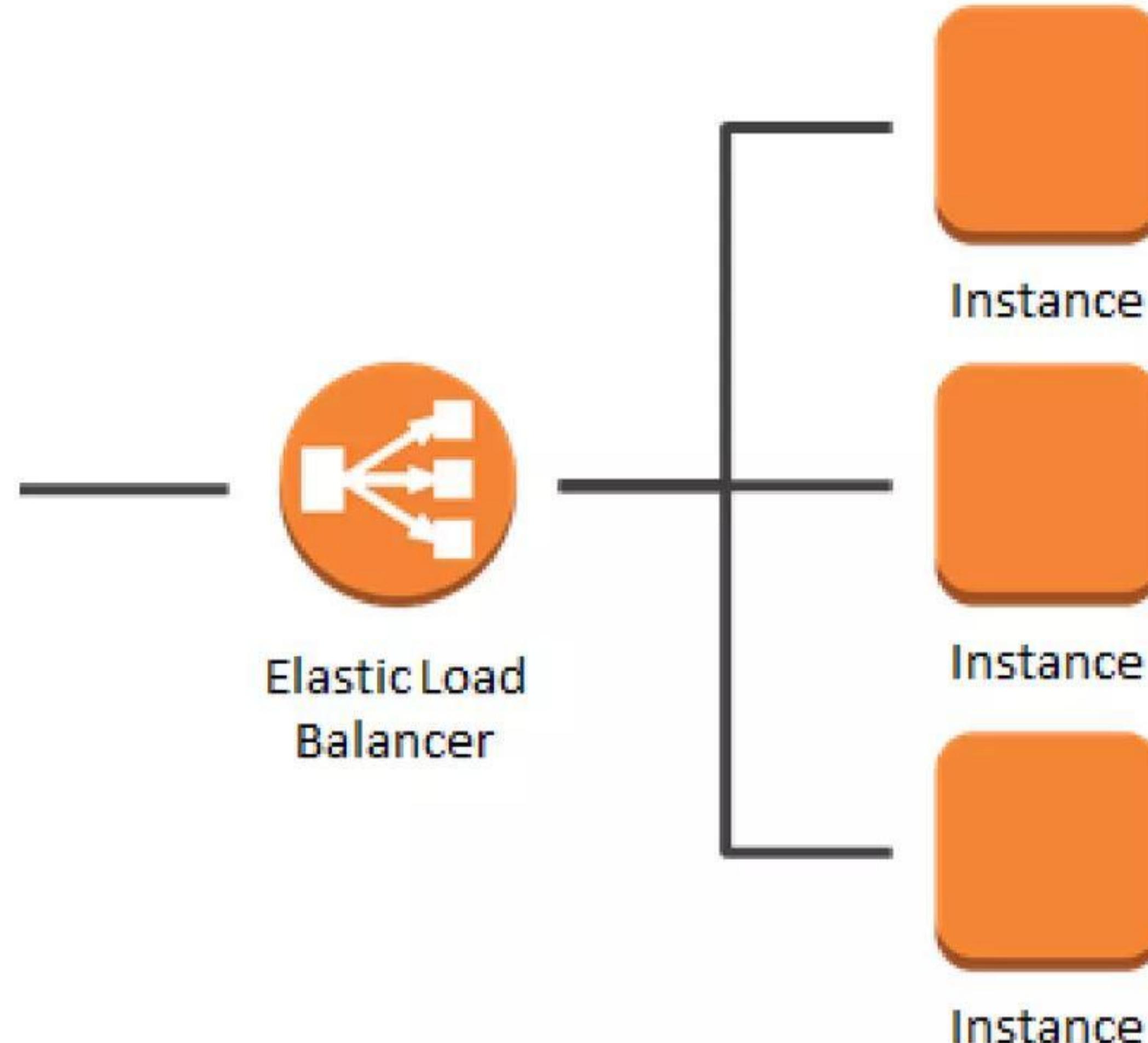
Price details:

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

AWS Elastic Load Balancer

What is AWS Elastic Load Balancer?

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



Types of Elastic Load Balancer

Application Load Balancer

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

Network Load Balancer

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

Gateway Load Balancer (Newly Introduced)

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

Classic Load Balancer

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

Listeners

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

Target Group

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

Health Check

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

Use Cases:

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

Charges:

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

Amazon Route 53

What is Amazon Route 53?

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

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

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

Route 53 TTL (seconds):

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

Route53 CNAME vs. Alias

CNAME	Alias
It points a hostname to any other hostname. (app.mything.com -> abc.anything.com)	It 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:-

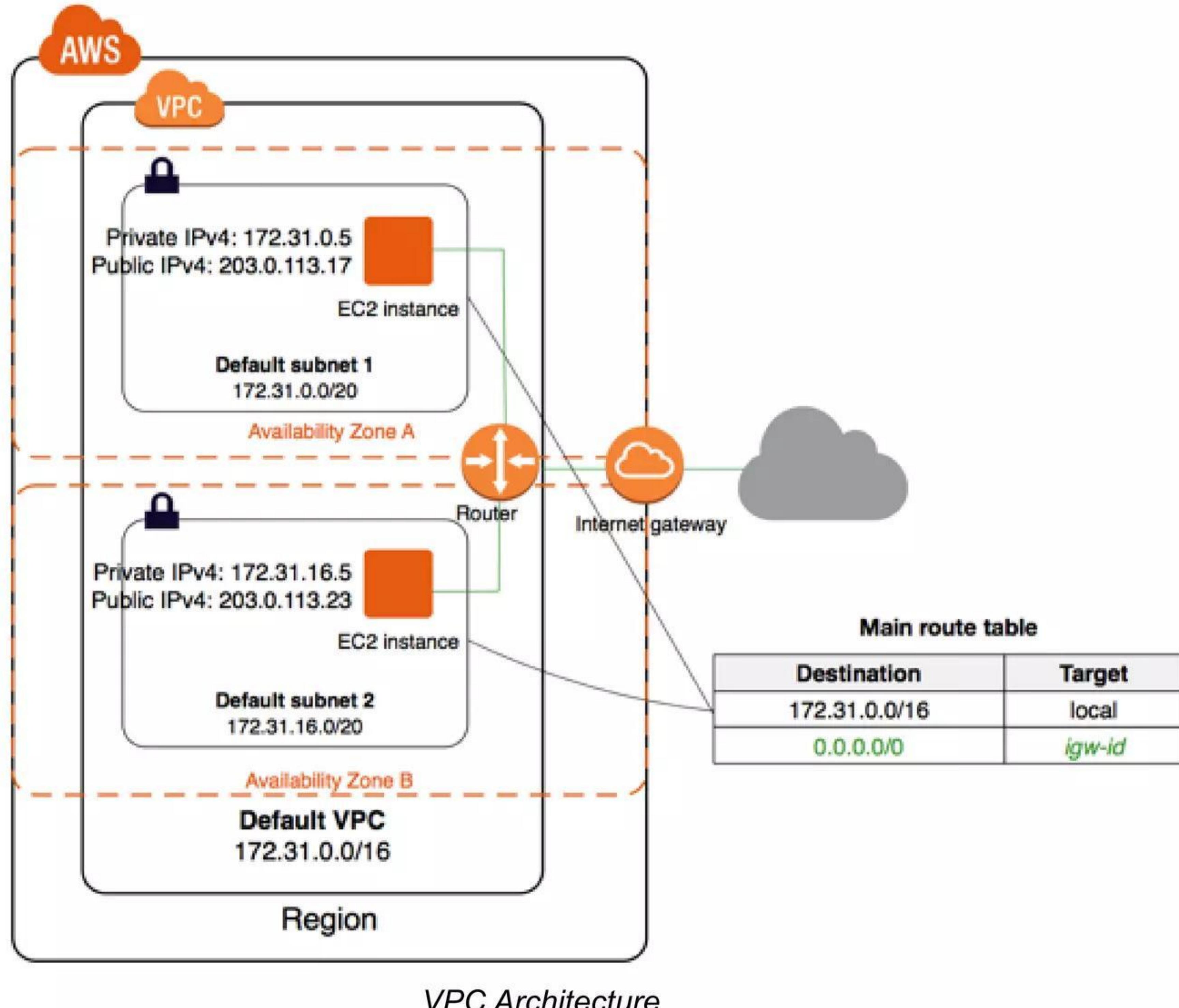
Inbound rule - Allows all inbound traffic

Outbound rule - Allows all outbound traffic

Custom Network ACL:- (by default)

Inbound rule - Denies all inbound traffic

Outbound rule - Denies all outbound traffic



Components of VPC:

Subnets

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

Route Tables

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

NAT Devices

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

NAT Instance

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

NAT Gateway

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

DHCP Options Set:

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

PrivateLink

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

Endpoints

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

Types of Endpoints

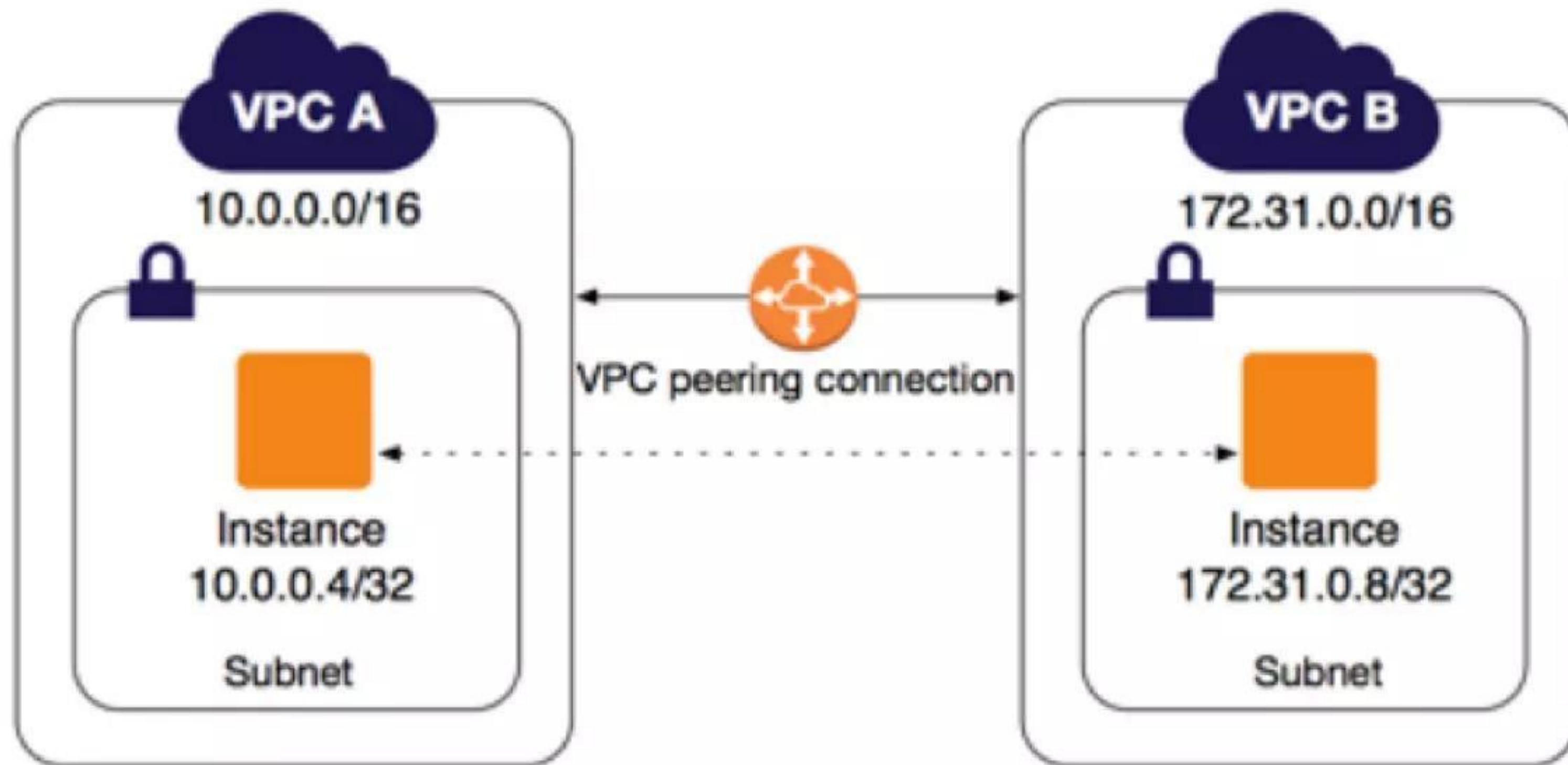
- **Interface Endpoints**
 - It is an entry point for traffic interception.
 - It will route the traffic to the service that you configure.
 - It will use an ENI with a private IP address.
 - For Example: it will allow instances to connect to Amazon Kinesis through interface endpoint.
- **Gateway Load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

Egress Only Internet Gateway

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

VPC Peering:

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



VPN

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

AWS Site-to-Site VPN

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

AWS Client VPN

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

Use Cases:

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

Pricing:

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

AWS AppSync

What is AWS AppSync?

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

AppSync = “The Facilitator”

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

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

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

AppSync Benefits:

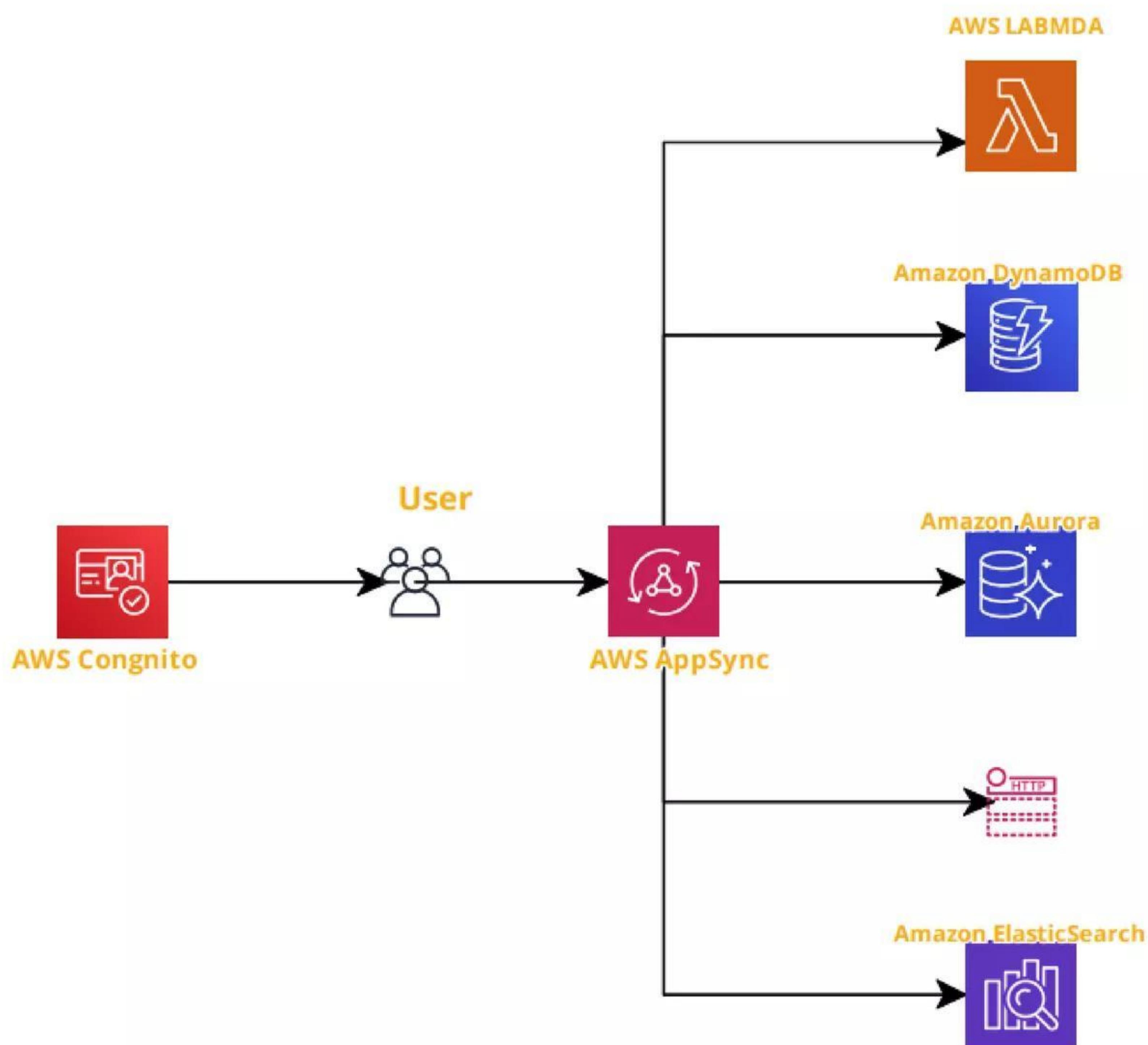
- Fast setup – great scalability as needed.
- Real-time subscriptions and offline access.
- Unified secured access.
- Provision of caching capabilities for performance improvements.
- Bandwidth optimization.
- Conflict resolution in the cloud.

Use Cases:

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

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

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



Pricing:

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

Amazon EventBridge

What is Amazon EventBridge?

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

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

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

Key Concepts:-

Event Buses:

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

Events:

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

Schema Registry:

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

Rules:

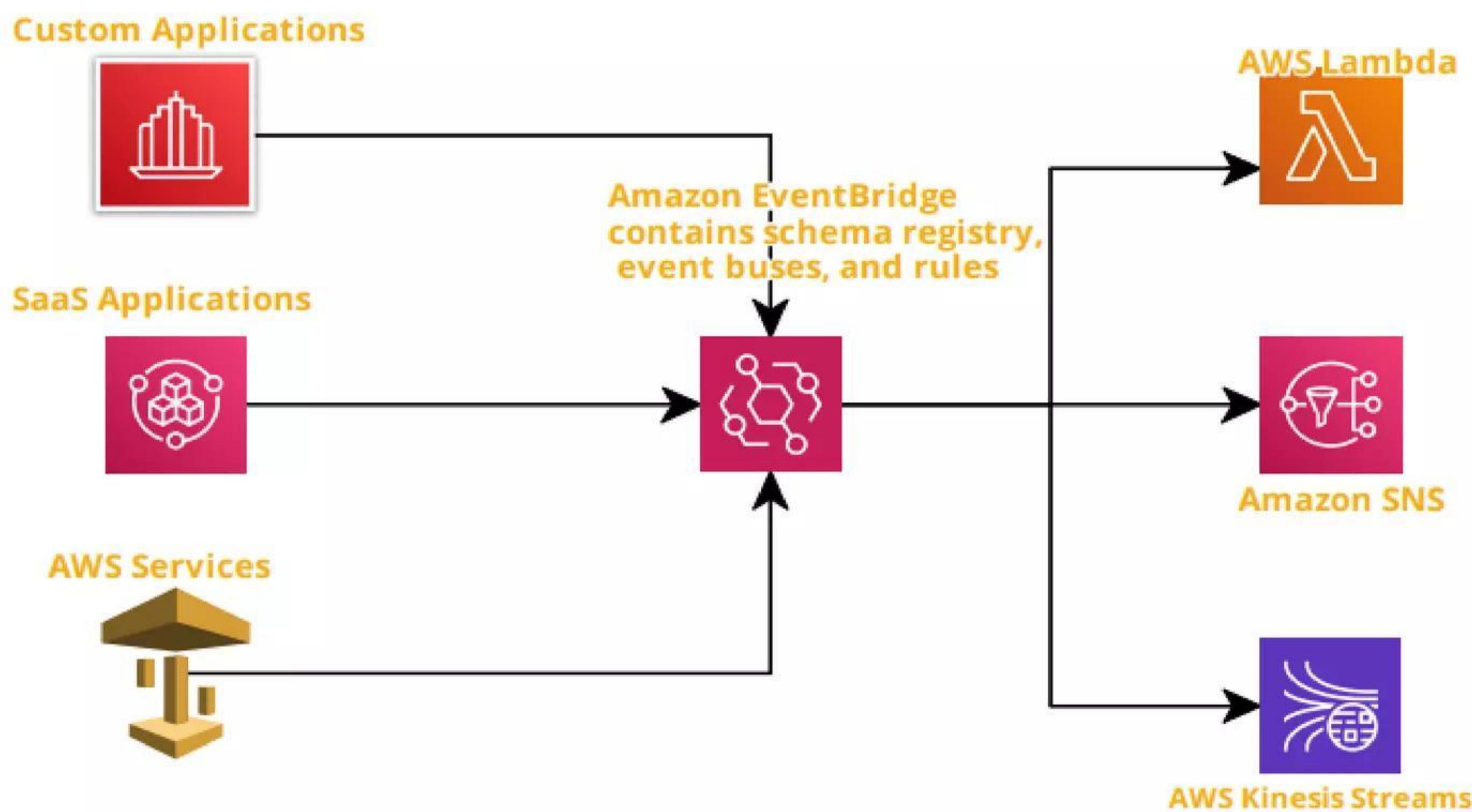
A rule matches incoming events and routes them to targets for processing. A single rule can route an event (JSON format) to multiple targets. All pointed targets will be processed in parallel and no particular order.

Targets:

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

Features:

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



Structure of the Amazon EventBridge service.

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

Pricing:

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

AWS SNS (Simple Notification Service)

What is AWS SNS?

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

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

It allows developers to group multiple recipients using topics.

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

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

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

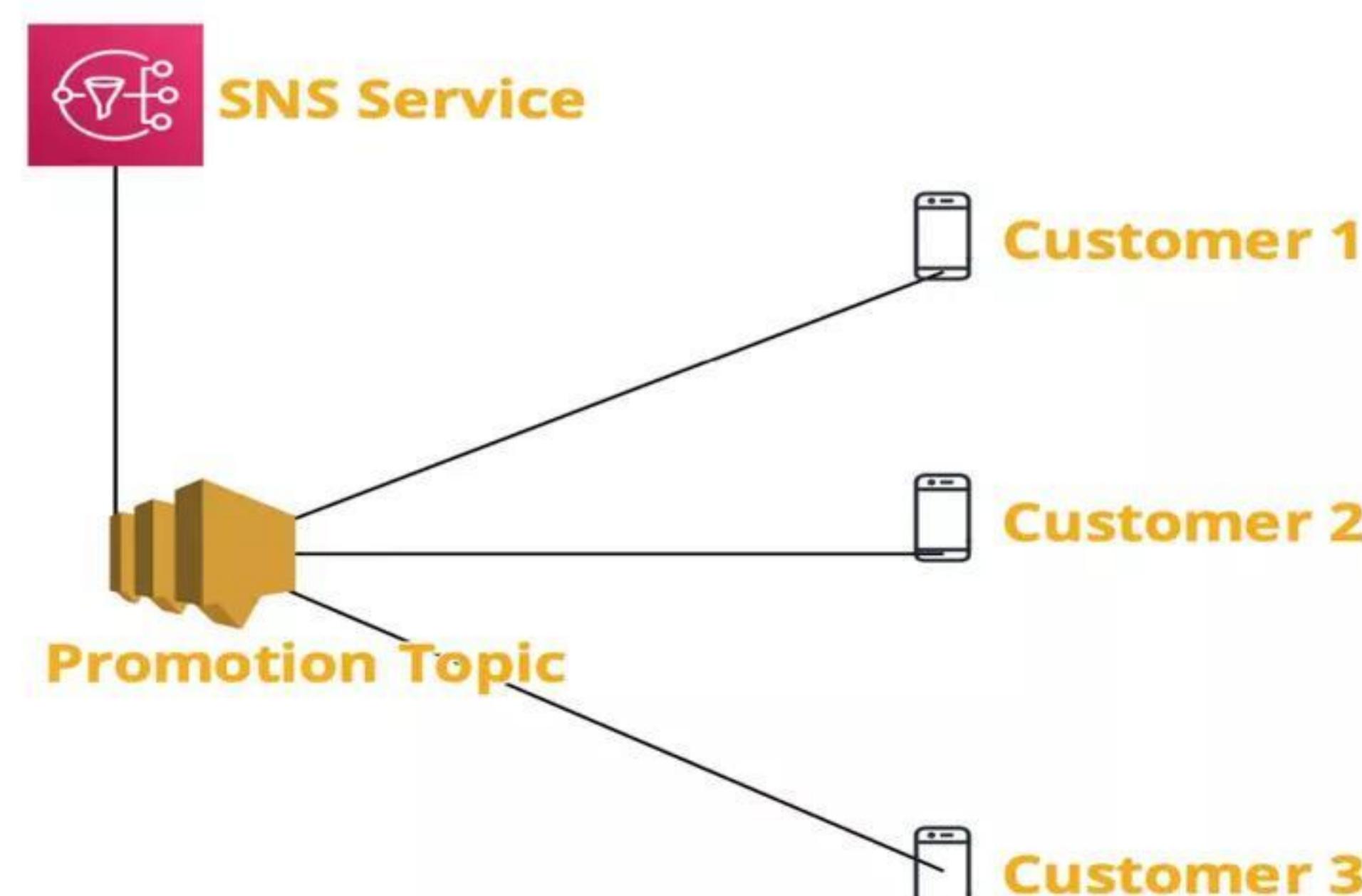
1. Standard topic is used when incoming messages are not in order. In other words, messages can be delivered as they are received.
2. FIFO topic is designed to maintain order of the messages between the applications, especially when the events are critical. Duplication will be avoided in this case.

Features:

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

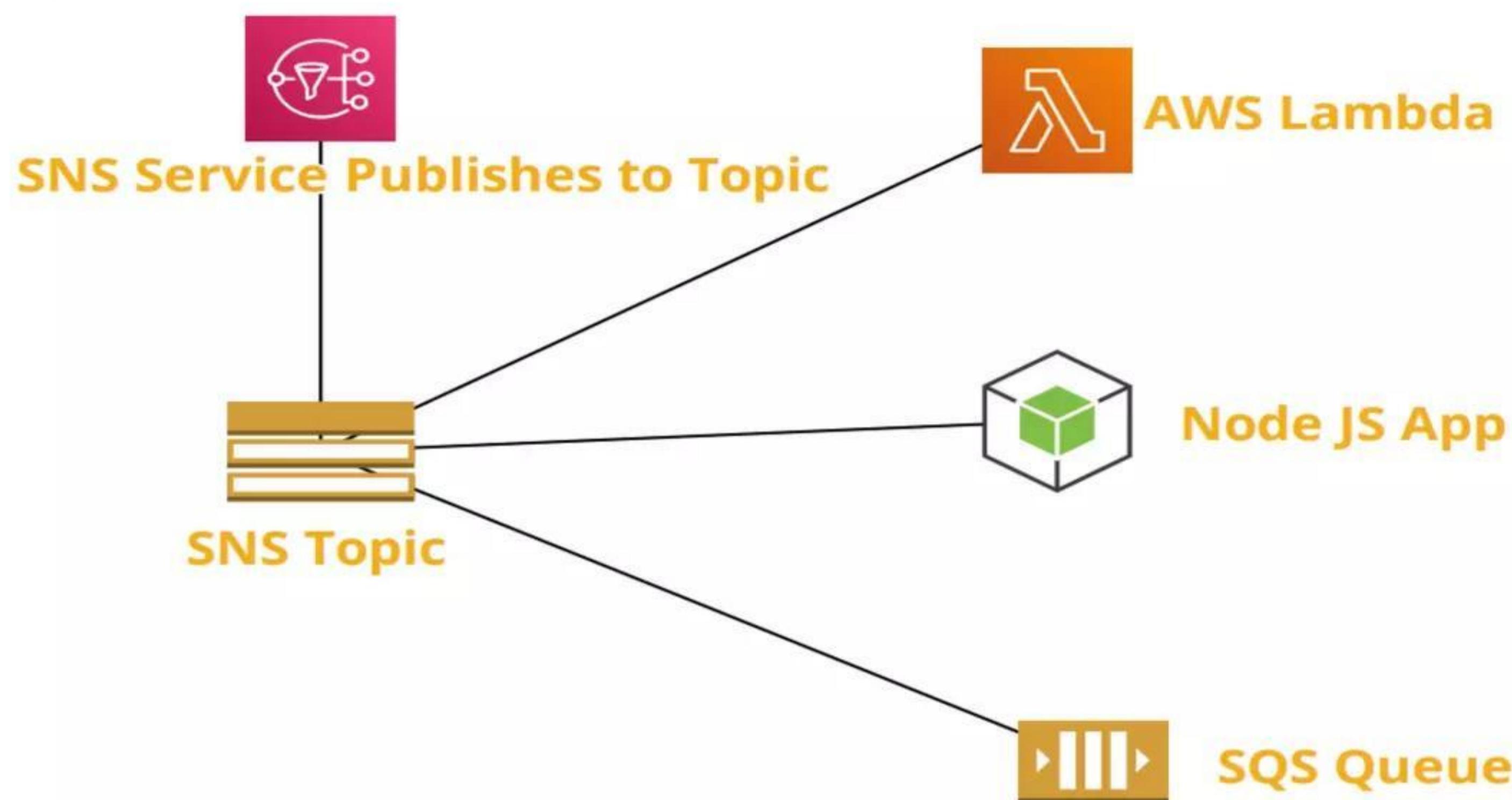
Use cases:

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



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

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



Pricing:

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

Amazon Simple Queue Service (SQS)

What is Amazon Simple Queue Service (SQS)?

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

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

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

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

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

SQS messages have a fixed size of 256KB.

There are two SQS Queue types:

Standard Queue -

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

FIFO Queue -

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



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

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

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

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

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

AWS Step Functions

What are step functions?

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

Types of step functions:

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

Features:

- Allow to create workflow which follows a fixed or dynamic sequence.
- Inbuilt "Retry" and error handling functionality.
- Support integration with AWS native Lambda, SNS, ECS, AWS Fargate, etc.
- Support GUI audit workflow process, input/output, etc., well.
- GUI provides support to analyze the running process and detect the failures immediately.
- High availability, High scalability and low cost.
- Manages the states of the application during workflow execution.
- Step function is based on the concepts of tasks and state machines.
 - 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 Standard workflows)
 - \$1.00 per 1M requests (For Express workflows)

Amazon Simple Workflow Service(SWF)

What is SWF?

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

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

SWF Actors

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

Features:

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

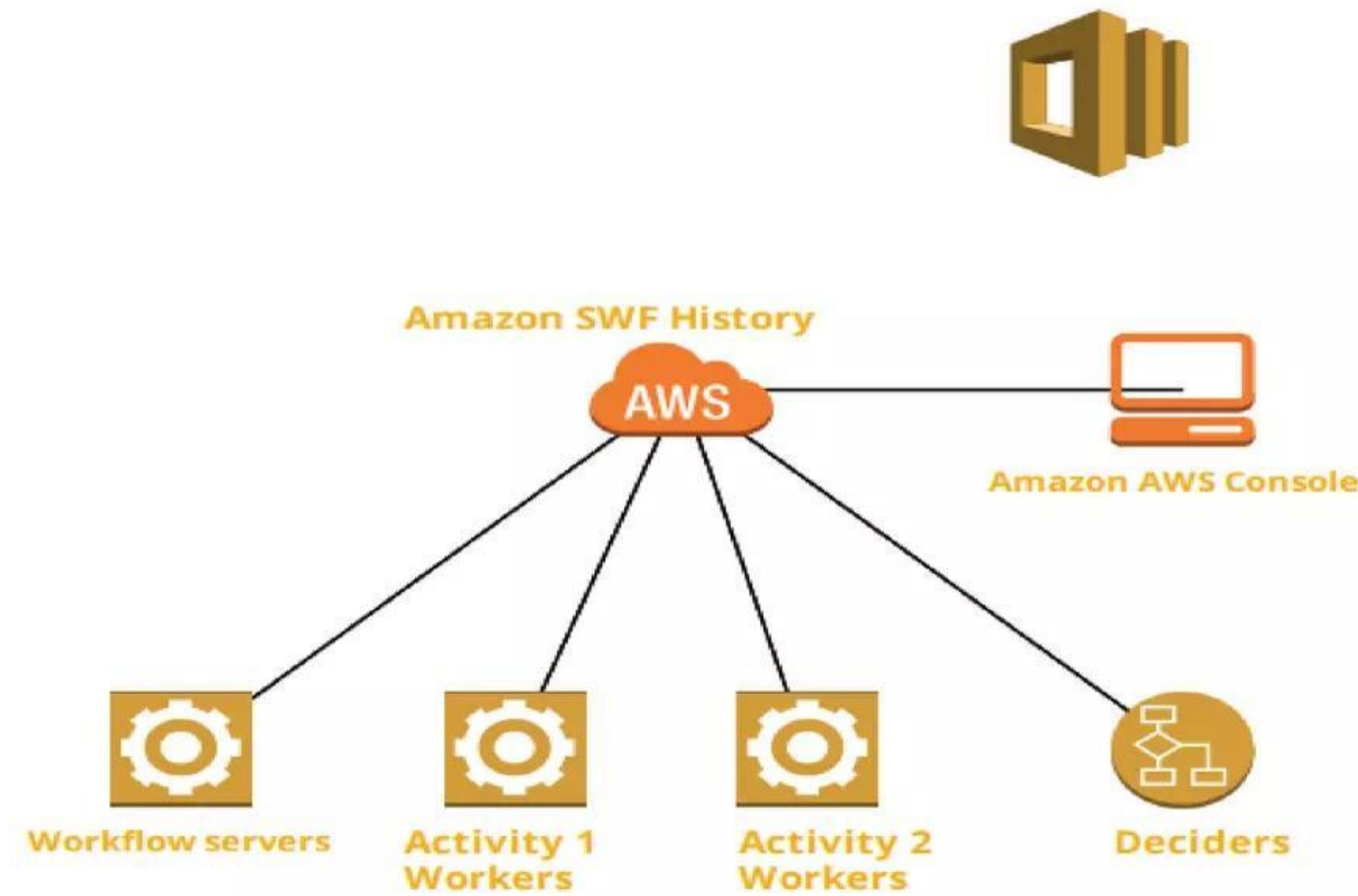
Use Cases:

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

Data warehouse process:

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

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



Pricing:

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

Open and retained workflows:

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

AWS Cost Explorer

What is AWS Cost Explorer?

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

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

The default reports provided by Cost Explorer are:

- **Cost and usage reports:**

It provides the following data for understanding the costs:-

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

- **Reserved Instance reports:**

It provides the following reports for understanding the reservations:-

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

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

The screenshot shows the AWS Cost Explorer interface. On the left, there's a navigation sidebar with links like Home, Cost Management, Cost Explorer (which is highlighted in orange), Budgets, Budgets Reports, Savings Plans, Cost & Usage Reports, Cost Categories, Cost allocation tags, Billing, Bills, Orders and invoices, and Credits. The main content area has a header "Cost Explorer" with an "Info" link. Below it is a "Welcome to Cost Explorer" section. A large orange button on the right says "Launch Cost Explorer". Underneath, there are three sections: "Use preconfigured views" (describing cost distribution by service over three months), "Analyze spend" (describing historical spend analysis), and "Download or bookmark" (describing data download and bookmarking). A note at the top states: "Cost Explorer provides reporting, analytics and visualization capabilities that you can use to track and manage your AWS costs. You will be able to see your spend data within 24 hours after you launch Cost Explorer for the first time."

AWS Cost Explorer

Price details:

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

AWS Budgets

What is AWS Budgets?

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

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

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

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

Best Practices:

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

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

Price details:

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

AWS Cost & Usage Report

What is AWS Cost & Usage Report?

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

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

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

Reserved Instance Reporting

What is Reserved Instance Reporting?

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

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

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

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

RI utilization reports:

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

RI coverage reports:

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

Reserved Instance Utilization and Coverage reports (both):

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

Price details:

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

AWS Personal Health Dashboard

What is AWS Personal Health Dashboard?

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

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

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

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

At the Personal Health Dashboard, there are three categories:

Open issues - shows issues of the last seven days.

Scheduled changes - shows items of any upcoming changes.

Other notifications.

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

The screenshot shows the AWS Personal Health Dashboard interface. On the left, a sidebar lists 'AWS account' (Dashboard, Event log, Amazon CloudWatch Events), 'Organizational view', and other options. The main dashboard has a header 'Dashboard' and a timestamp 'Last refreshed less than 1 min ago'. It features an 'Overview' section with three large numbers: 0 Open issues (Past 7 days), 0 Scheduled changes (Upcoming and past 7 days), and 0 Other notifications (Past 7 days). To the right is a 'Set up alerts' section with a 'Create rule' button. Below the overview is a table for 'Open issues (0)'. The table includes columns for 'Event', 'Region / Zone', 'Info', 'Start time', 'Last update time', and 'Affected resources'. A search bar at the top of the table says 'Add filter'. At the bottom of the table, it says 'No events found'. There are also navigation arrows and a 'View event log' button.

AWS Personal Health Dashboard

AWS Management Console

What is AWS Management Console?

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

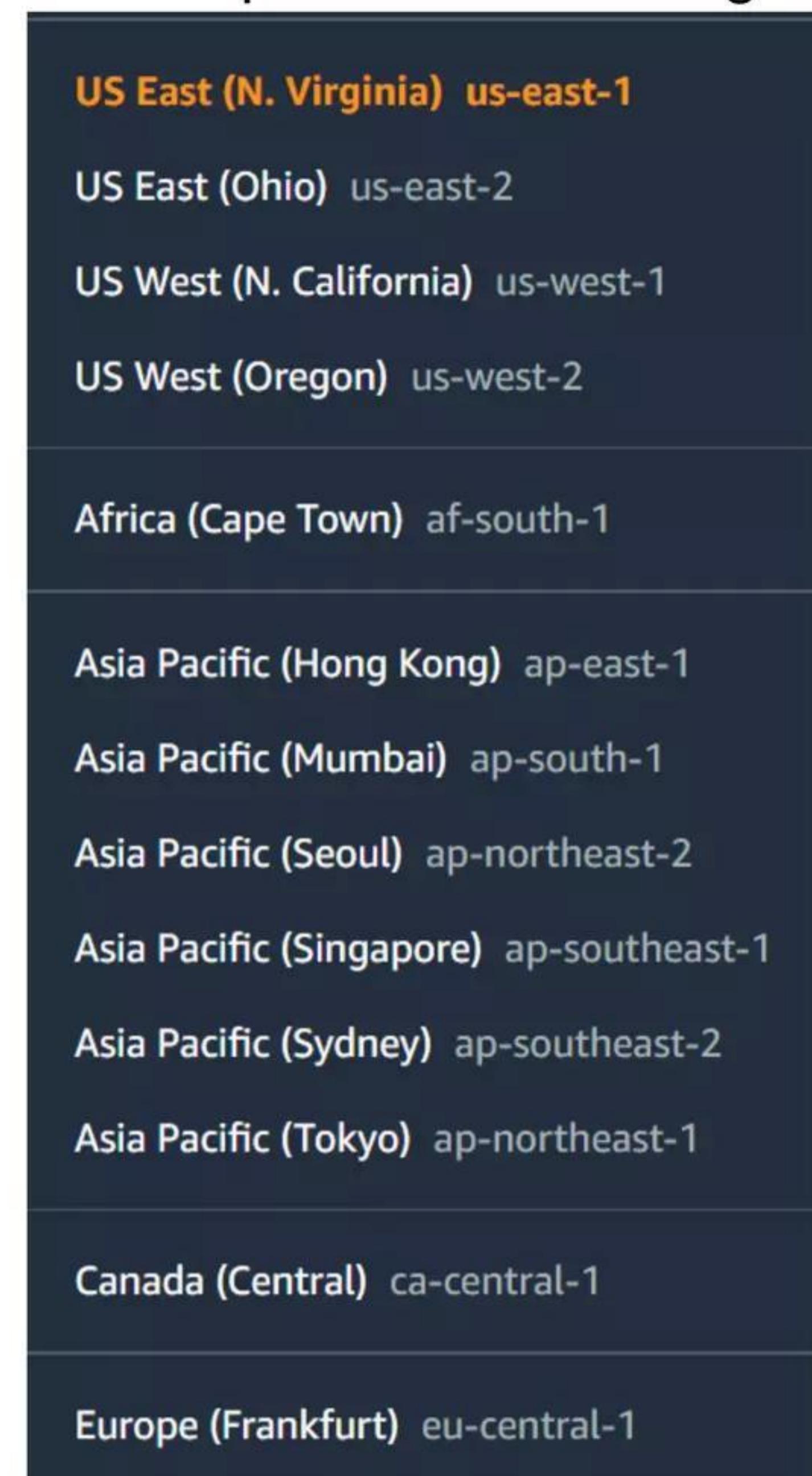
It can be visible at the time a user first signs in. It provides access to other service consoles and a user interface for exploring AWS.

The screenshot shows the AWS Management Console homepage. At the top, there's a navigation bar with the AWS logo, a "Services" dropdown, a search bar containing "Search for services, features, marketplace products, and docs", a keyboard shortcut "[Alt+S]", and account information for "N. Virginia" and "Support". The main title "AWS Management Console" is centered above two columns. The left column, titled "AWS services", lists "Recently visited services" including CloudFormation, EC2, S3, VPC, AWS AppConfig, Systems Manager, and IAM. The right column, titled "Stay connected to your AWS resources on-the-go", promotes the AWS Console Mobile App for iOS and Android, with a "Learn more" link. Below these are sections for "Explore AWS" and "Amazon Redshift", each with a "Learn more" link. A central banner at the bottom reads "AWS Management Console".

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

The screenshot shows the AWS Services Console, which is a subset of the Management Console. It features a dark-themed sidebar on the left with "Favorites" (Resource Groups & ...), "Recently visited" (Console Home, CloudFormation, EC2, S3, VPC, AWS AppConfig, Systems Manager, IAM, AWS Backup, CodeDeploy, RDS, Elastic Kubernetes Service, Elastic Container Service), and a "Services" dropdown. The main area is titled "All services" and is organized into several categories: Compute (EC2, Lightsail, Lambda, Batch, Elastic Beanstalk, Serverless Application ...), Storage (S3, EFS, FSx, S3 Glacier, Storage Gateway, AWS Backup), Customer Enablement (AWS IQ, Support, Managed Services, Activate for Startups), Blockchain (Amazon Managed Blockchain), Satellite (Ground Station), Quantum Technologies (Amazon Braket), Management & Governance, Machine Learning (Amazon SageMaker, Amazon Augmented AI, Amazon CodeGuru, Amazon DevOps Guru, Amazon Comprehend, Amazon Forecast, Amazon Fraud Detector, Amazon Kendra, Amazon Lex, Amazon Personalize, Amazon Polly, Amazon Rekognition, Amazon Textract, Amazon Transcribe, Amazon Translate, AWS DeepComposer), Front-end Web & Mobile (AWS Amplify, Mobile Hub, AWS AppSync, Device Farm, Amazon Location Service, Amazon Forecast, Amazon Fraud Detector, Amazon Kendra, Amazon Lex, Amazon Personalize, Amazon Polly, Amazon Rekognition, Amazon Textract, Amazon Transcribe, Amazon Translate, AWS DeepComposer), AR & VR (Amazon Sumerian), Application Integration (Step Functions, Amazon AppFlow, Amazon EventBridge, Amazon MQ, Simple Notification Service, Simple Queue Service), and others like Amazon Forecast, Amazon Fraud Detector, Amazon Kendra, Amazon Lex, Amazon Personalize, Amazon Polly, Amazon Rekognition, Amazon Textract, Amazon Transcribe, Amazon Translate, AWS DeepComposer, Step Functions, Amazon AppFlow, Amazon EventBridge, Amazon MQ, Simple Notification Service, and Simple Queue Service.

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



AWS Regions

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

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