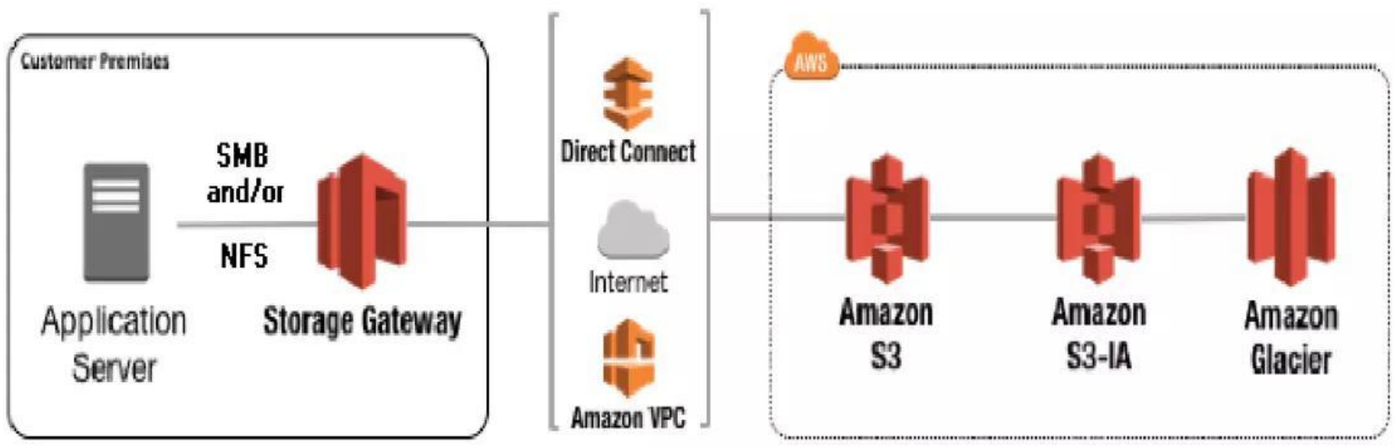
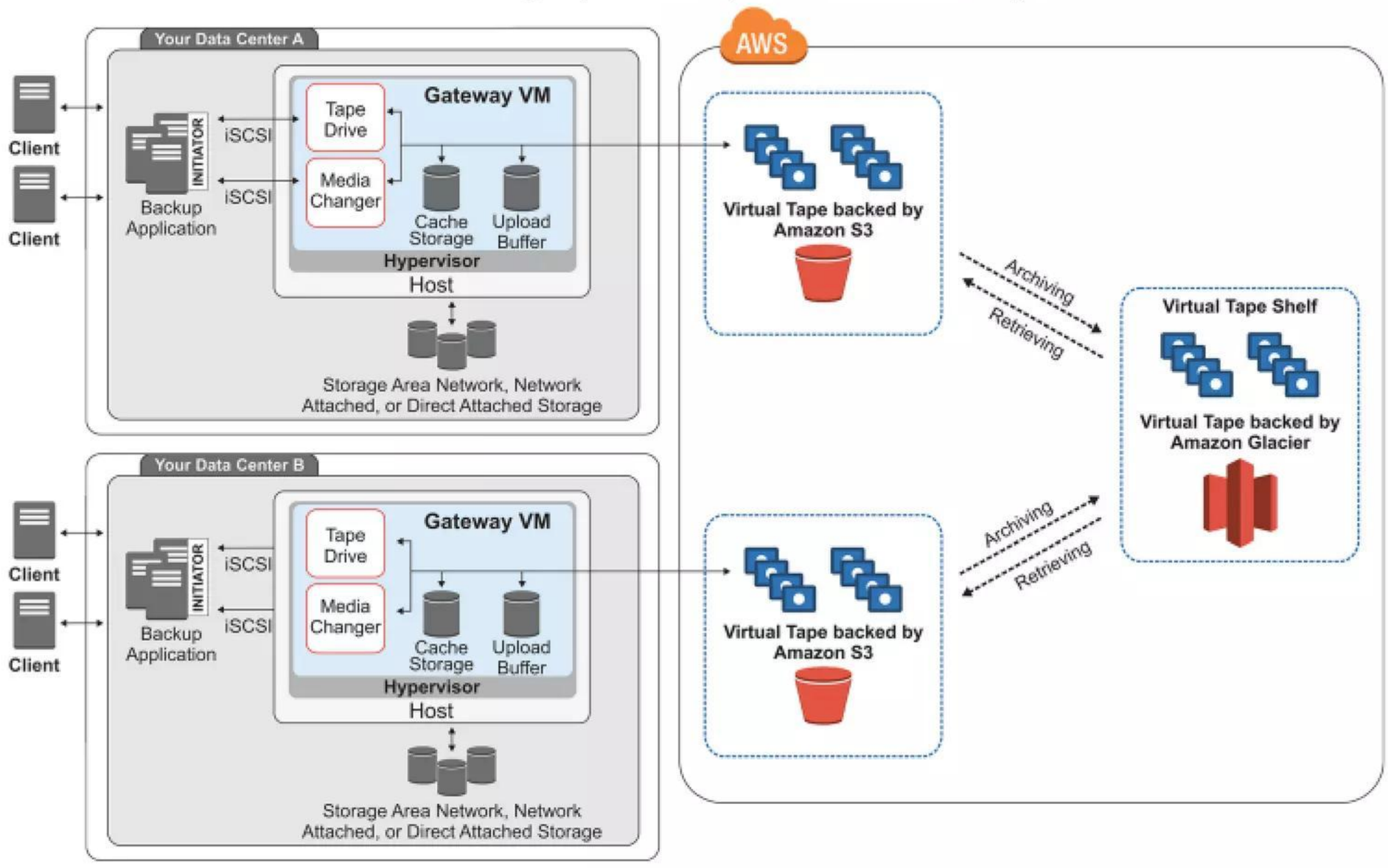
File Gateway (NFSv4 / SMB)

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



Tape Gateway (VTL)

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



Features of AWS Storage Gateway

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

Use Cases:

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

Pricing :

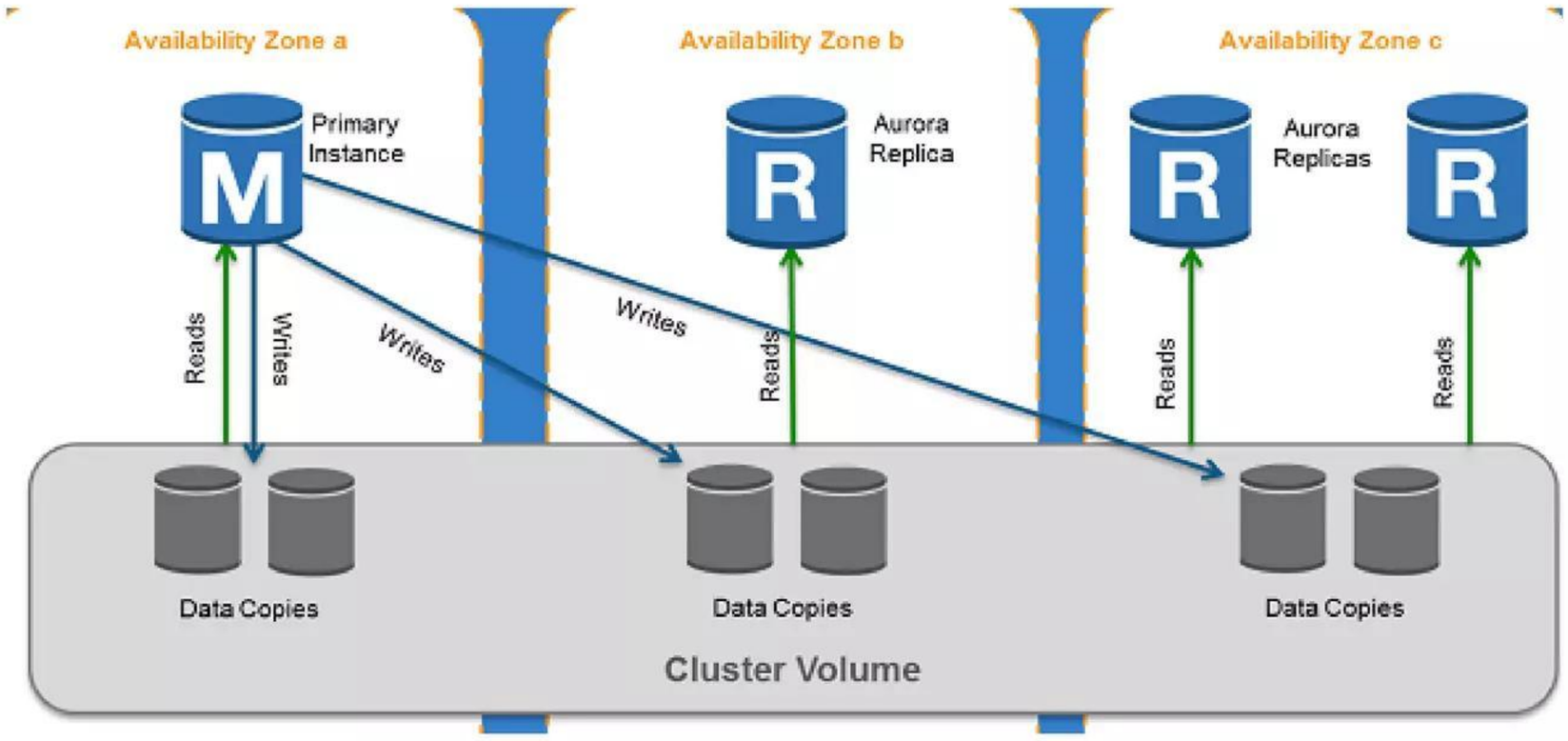
- Charges are applied on what you use with the AWS Storage Gateway and based on the type and amount of storage you use.

Amazon Aurora

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

Features:

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



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

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

Best practices:

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

Pricing:

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

Amazon DocumentDB

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

Features:

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

Best Practices:

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

Use Case:

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

Pricing:

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

Amazon DynamoDB

What is DynamoDB?

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

Features:

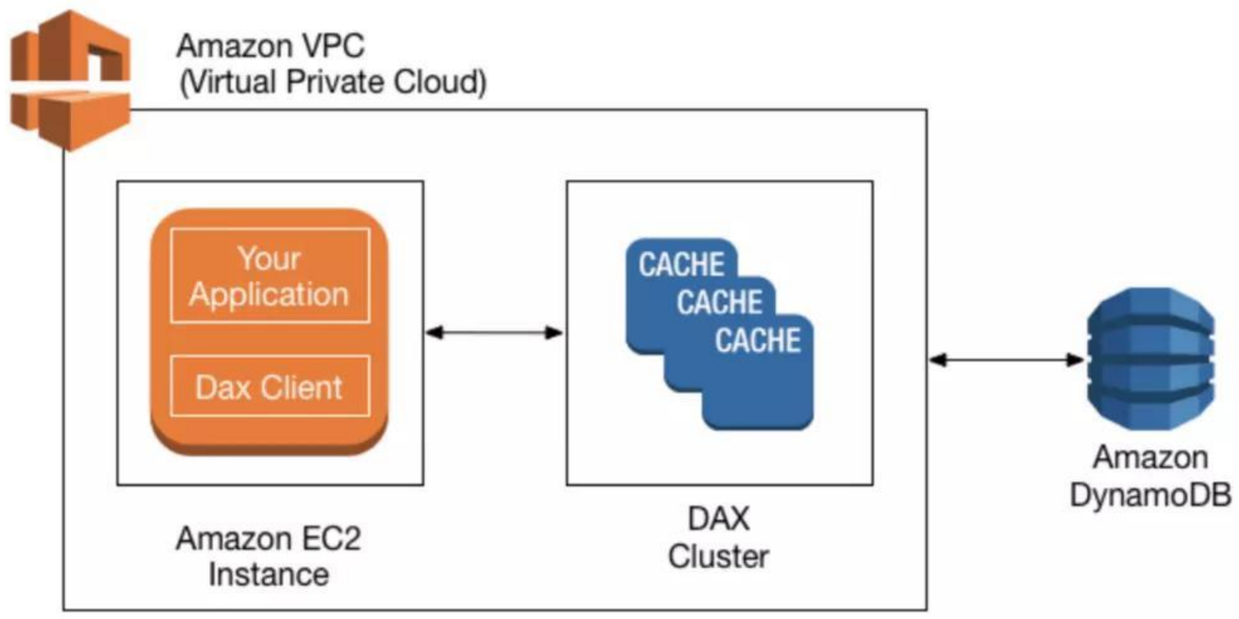
- It can create the Table for your application and can handle the rest.
- No-SQL database provides fast and predictable performance.
- It is designed for automatic scaling and can be replicated across regions.
- It can also create Dynamic Tables, which means it can store any number of multi-valued attributes.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

DynamoDB Accelerator

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

DynamoDB Scan

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



DynamoDB Queries

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

DynamoDB Streams

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

DynamoDB Transactions

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

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

Consistency Model:

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

Throughput Model:

Read Capacity Unit (RCU): For an item, it represents a single strongly consistent read or double eventual consistent reads in a second, and the size of an item can be up to 4KB. If you need to read an item that is larger than 4KB, you need to add an additional read capacity unit.
Write Capacity Unit (WCU): It represents one write per second for an item. And the size of an item can be up to 1KB. If you need to write a larger than 1KB item, you need to add an additional write capacity unit.
Provisioned: We Need to mention the throughput in advance. This model is for Predictable workloads. We have to define a range for Reading and Write Capacity Units.
On-Demand: We need not mention the throughput in Advance. This Model is for non-predictable workloads. We do not need to define a range for Reading and Write Capacity Units.

Charges:

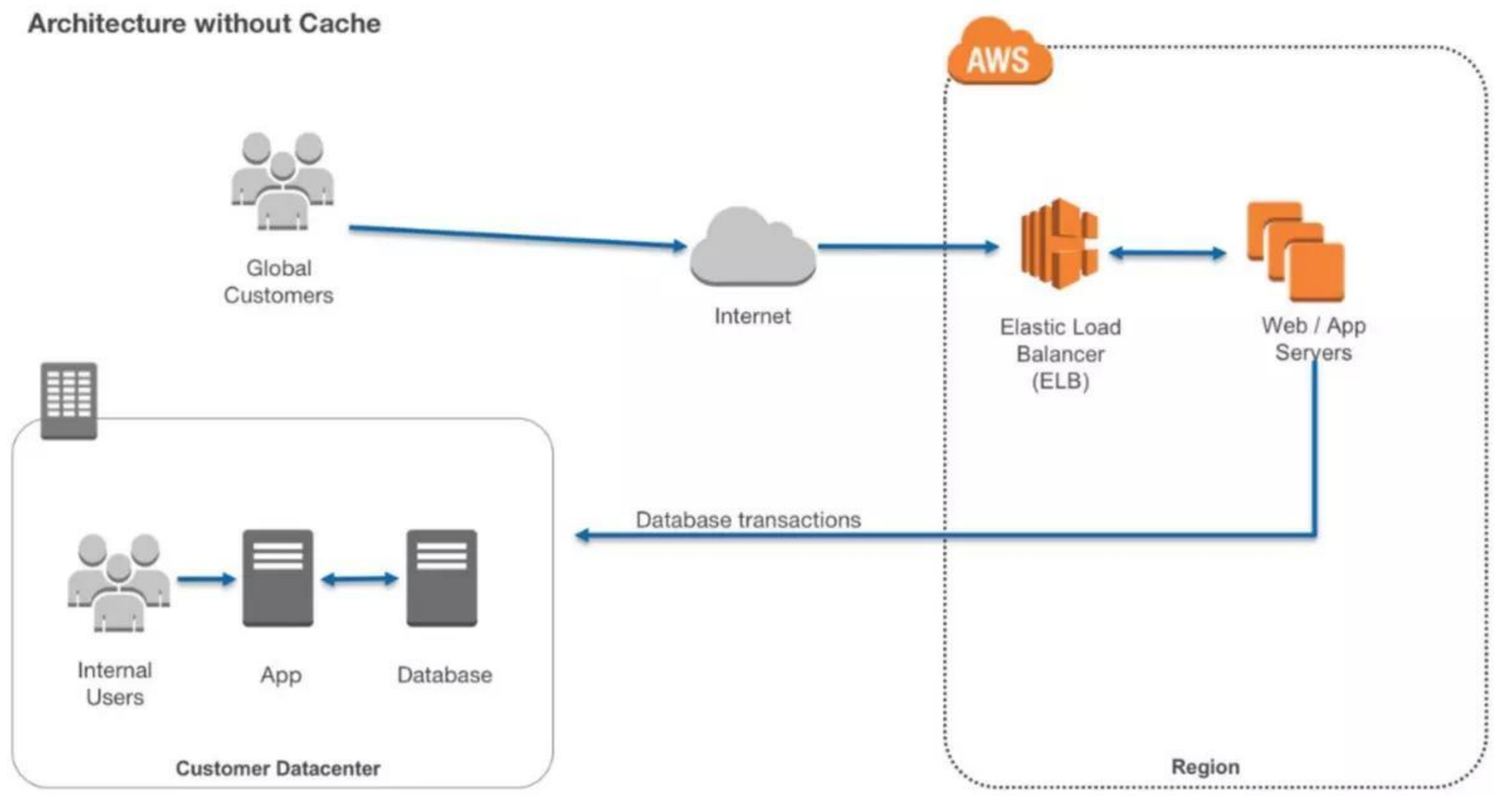
- DynamoDB charges as per the disk space you consume.
- Charges for data transfer out.
- Charges for provisioned throughput.
- Charges for Reserved Capacity Unit.

Amazon ElastiCache

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

Features:

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



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

Best practices:

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

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

Pricing:

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

Amazon Keyspaces

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

Features:

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

Use Cases:

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

Pricing:

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

Amazon Neptune

What is Amazon Neptune?

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

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

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

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

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

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

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

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

Amazon RDS

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

Engines supported by RDS are given below:

MySQL

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

MS SQL

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

MariaDB

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

PostgreSQL

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

Oracle

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

Amazon Aurora

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

DB Instance Class

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

Multi AZ Deployment

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

Read Replicas

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

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

Storage Type

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

Monitoring

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

Backups & Restore

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

Charges:

You will be charged based on multiple factors:

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

Amazon Redshift

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

Redshift can be configured as follows:

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

Features:

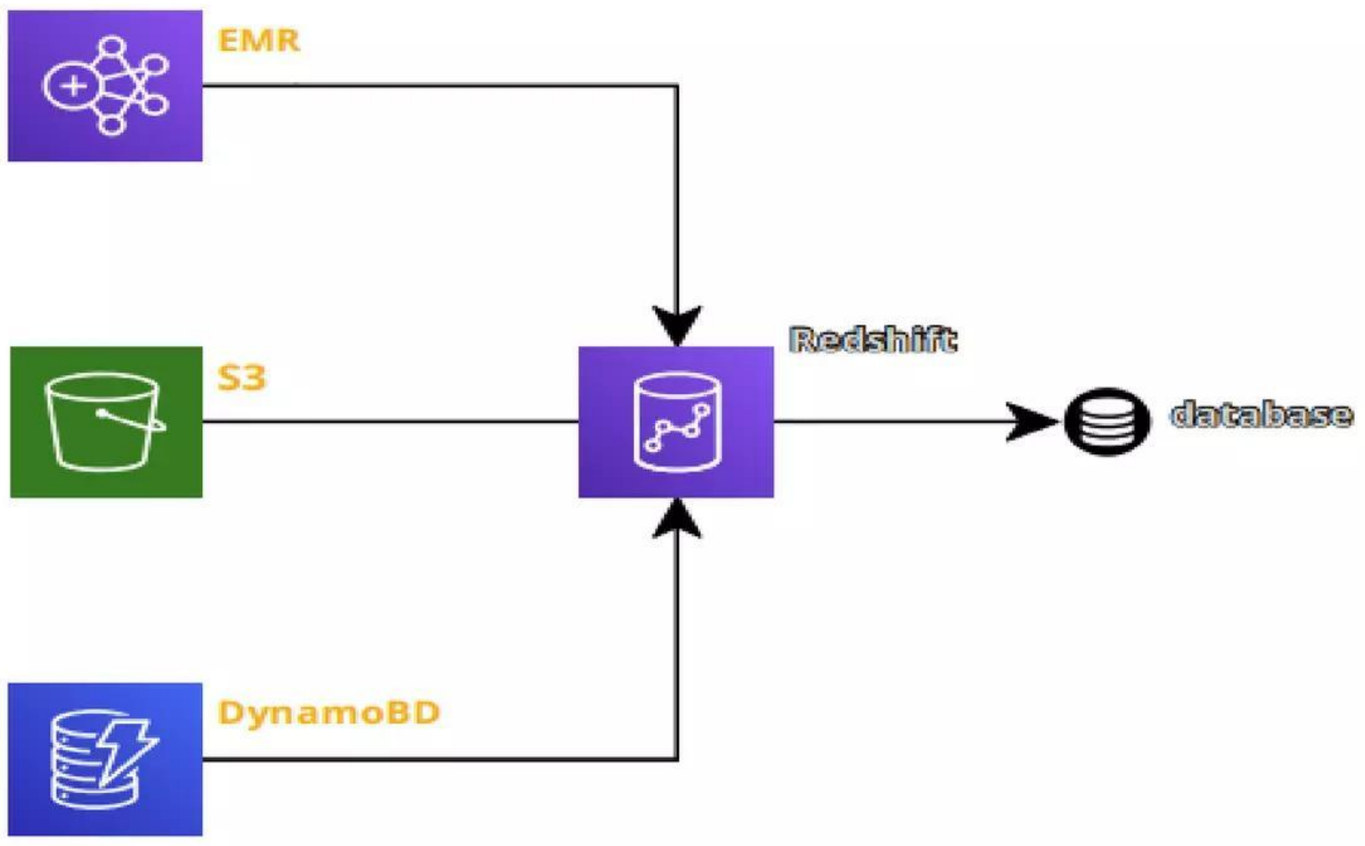
- It employs multiple compression techniques and can often achieve significant compression relative to traditional relational databases.
- It doesn't require indexes or materialized views, so uses less space than traditional database systems.
- Massively parallel processing (**MPP**): Amazon redshift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

Security Considerations

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

Use cases

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



Pricing:

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

AWS IAM

What is Identity Access and Management?

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

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

IAM User & Root User

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

IAM Group

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

IAM Role

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

IAM Policies

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

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

IAM Security Best Practices:

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

Pricing:

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

Amazon Cognito

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

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

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

The two main components of Amazon Cognito are

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

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

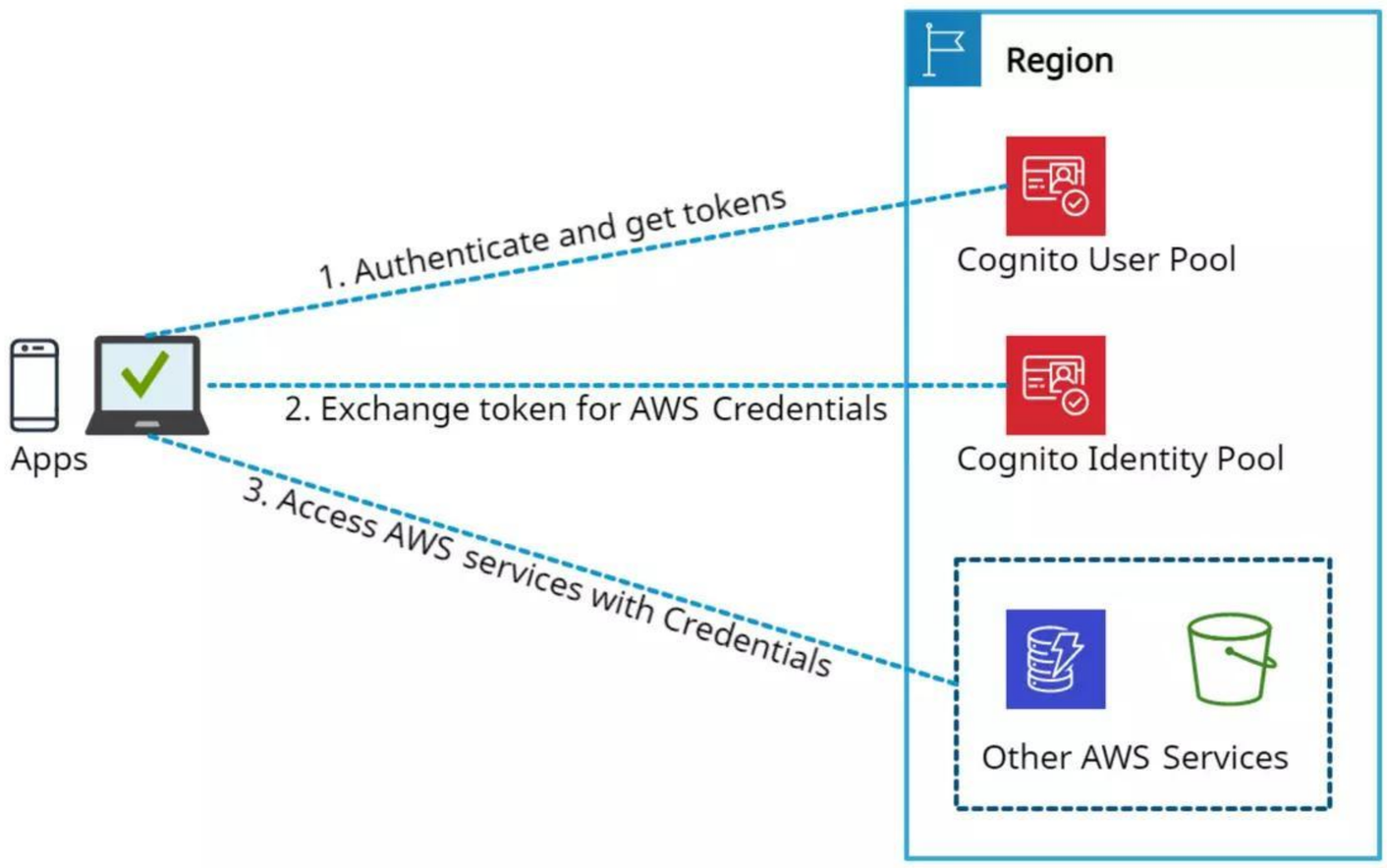
Identity pools provide temporary AWS credentials to the users so that they could access other AWS resources without re-entering their credentials. Identity pools support the following identity providers

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

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

Amazon Cognito Federated Identities

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



Amazon Cognito

Features:

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

Pricing Details: (pay only for what you use)

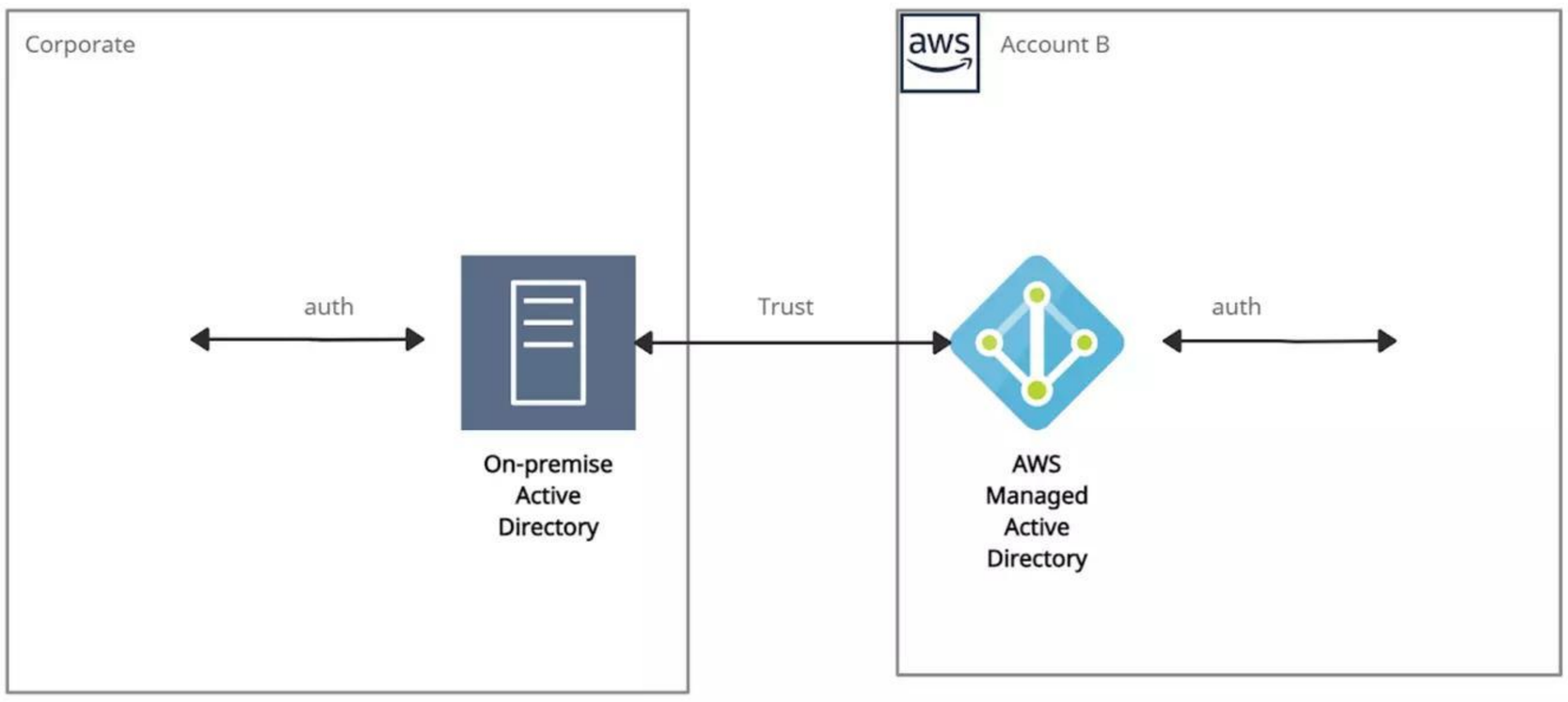
- Amazon Cognito is mainly charged for identity management and data synchronization.
- There are volume-based pricing tiers above the free tier for users who sign in directly with their credentials from a User Pool or with social identity providers such as Apple, Google, Facebook, and Amazon.

AWS Directory Service

What is AWS Directory Service?

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

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



AWS Managed AD

AWS Directory Service provides the following directory types to choose from

- Simple AD
- Amazon Cognito
- AD Connector

Simple AD:

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

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

Amazon Cognito:

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

AD Connector:

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

Use cases:

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

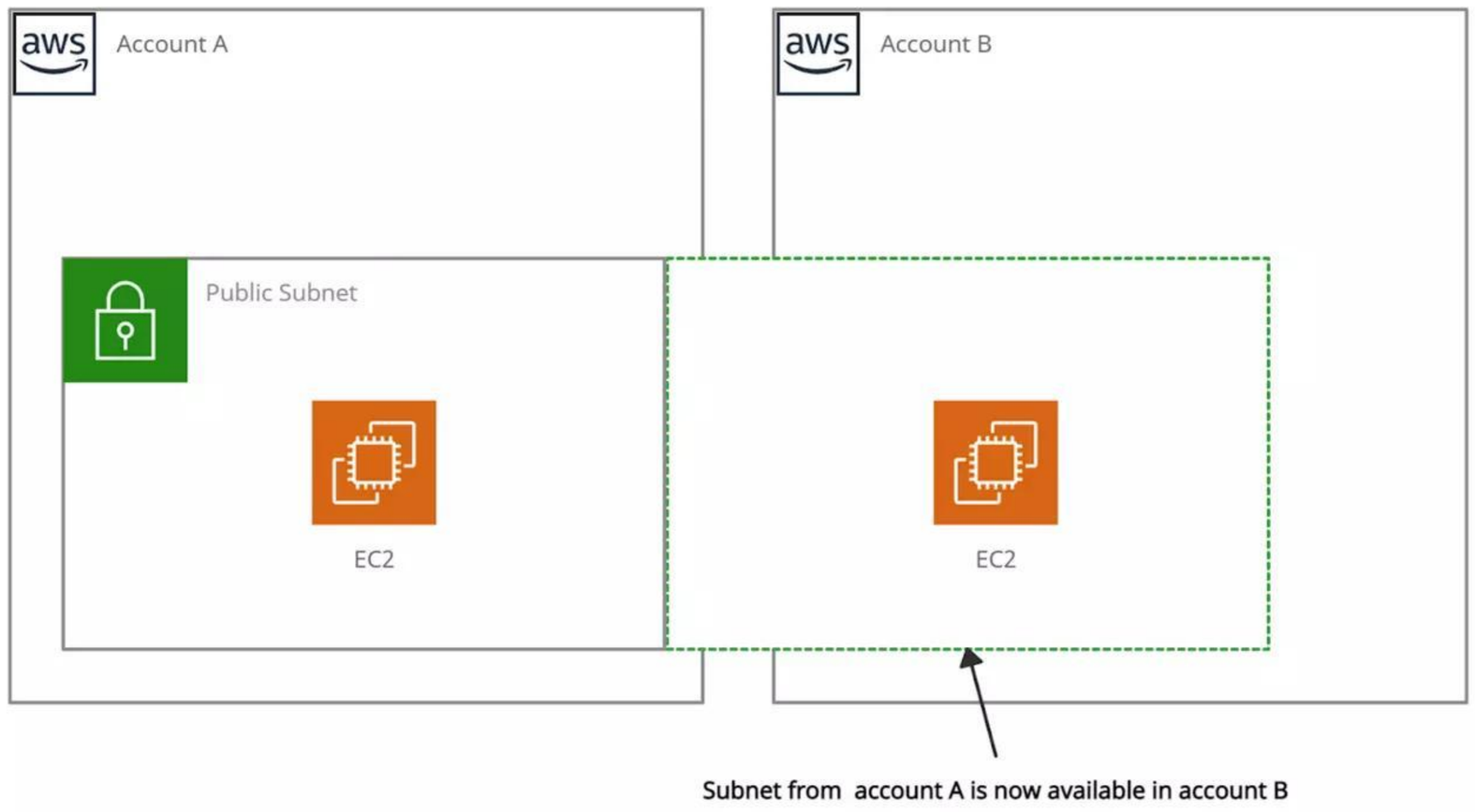
Pricing:

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

AWS Resource Access Manager

What is AWS Resource Access Manager?

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



AWS Resource Access Manager

Resources that can be integrated with AWS RAM are:

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

Benefits:

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

Price details:

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

AWS Secrets Manager

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

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

Accessing Secrets Manager:

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

Secret rotation is available for the following Databases:

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

Features:

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

Use cases:

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

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

Price details:

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

AWS Security Hub

What is AWS Security Hub?

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

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

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

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

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

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

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

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

Benefits:

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

Price details:

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

AWS Key Management Service

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

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

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

Customer managed CMKs:

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

AWS managed CMKs:

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

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

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

Features:

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

Benefits:

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

Price details:

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

AWS Certificate Manager (ACM)

What is AWS Certificate Manager?

AWS Certificate Manager is a service that allows a user to provide, manage, renew and deploy public and private Secure Sockets Layer/Transport Layer Security (SSL/TLS) X.509 certificates.
The certificates can be integrated with AWS services either by issuing them directly with ACM or importing third-party certificates into the ACM management system.

SSL Server Certificates:

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

The different types of SSL certificates are:

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

Ways to deploy managed X.509 certificates:

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

ACM certificates are supported by the following services:

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