AWS NOTES

--- Sheikh Navez

What is CLOUD?

"The cloud" refers to servers that are accessed over the Internet, the software and databases that run on those servers. Cloud servers are located in data centers all over the world. By using cloud computing, users and companies do not have to manage physical servers themselves or run software applications on their own machines.

The cloud enables users to access the same files and applications from almost any device, because the computing and storage takes place on servers in a data center, instead of locally on the user device.

What is AWS?

AWS stands for **Amazon Web Services**, which is a cloud computing platform that provides a variety of services to **individuals, businesses, and organizations.** AWS offers a wide range of services, including **computing power, storage solutions, database management, content delivery, and more.** It is a highly scalable, flexible, and cost-effective platform that allows users to access computing resources on-demand and **pay only for what they use.**

Why AWS?

There are several reasons why AWS is a popular choice for businesses and organizations looking to leverage cloud computing technologies:

1. Scalability: which allows users to scale up or down their computing resources as needed. This makes it easy for businesses to handle sudden spikes in traffic or workload without having to invest in additional hardware or infrastructure.

2. Flexibility: which offers a wide range of services and tools that can be customized to meet specific business needs. This includes everything from storage solutions and database management to machine learning and analytics.

3. Reliability: which is designed to provide high availability and uptime for its users. This means that businesses can rely on AWS to keep their applications and data accessible and secure at all times.

4. Security: which has implemented a comprehensive security framework that includes multiple layers of security controls and regular security audits. This makes it a trusted platform for businesses that need to protect sensitive data.

5. Cost-effective: which offers a pay-as-you-go pricing model, which means that businesses only pay for the computing resources they use. This can be a more cost-effective option than investing in and maintaining their own hardware and infrastructure.

Overall, AWS offers businesses and organizations a powerful, flexible, and cost-effective platform for running their applications and managing their data in the cloud.

Features of AWS?

Amazon Web Services (AWS) is a cloud computing platform that offers a wide range of services and features. Some of the key features of AWS include:

1. Elastic Compute Cloud (EC2): EC2 allows users to create virtual machines, or instances, in the cloud. Users can choose from a variety of instance types with different computing, memory, and storage capabilities, and can quickly scale up or down as needed.

2. Simple Storage Service (S3): S3 is a scalable, durable, and secure object storage service. It allows users to store and retrieve data from anywhere on the web, with built-in redundancy and availability features to ensure data is always accessible.

3. Relational Database Service (RDS): RDS allows users to easily deploy and manage relational databases in the cloud. Users can choose from several database engines, including MySQL, PostgreSQL, and Oracle, and can easily scale their database instances as needed.

4. Identity and Access Management (IAM): IAM allows users to manage access to AWS resources and services. It provides granular control over user permissions and allows users to set up multi-factor authentication and other security features.

5. Elastic Load Balancing (ELB): ELB distributes incoming traffic across multiple instances to improve availability and scalability. Users can choose

from several load balancing options, including Application Load Balancer, Network Load Balancer, and Classic Load Balancer.

6. Auto Scaling: Auto Scaling allows users to automatically adjust the number of EC2 instances based on demand. It helps ensure that users have enough resources to handle peak demand and can save money by reducing resources during periods of low demand.

7. Virtual Private Cloud (VPC): VPC allows users to create a private, isolated section of the AWS cloud. Users can control their network settings and configure a custom virtual network topology.

8. AWS Lambda: AWS Lambda is a serverless computing service that allows users to run code without provisioning or managing servers. Users can write their code in several programming languages and run it in response to events, such as changes to data in an S3 bucket or updates to a database.

These are just a few of the many features offered by AWS. The platform includes many other services and features, including analytics, machine learning, security, and more.

Benefits of AWS?

There are numerous benefits of using AWS (Amazon Web Services). Here are some of the key advantages:

1. Scalability: AWS allows users to easily scale up or down their computing resources as needed, which helps businesses handle sudden spikes in traffic or workload without investing in additional hardware or infrastructure.

2. Flexibility: AWS offers a wide range of services and tools that can be customized to meet specific business needs. This includes everything from storage solutions and database management to machine learning and analytics.

3. Reliability: AWS is designed to provide high availability and uptime for its users. This means that businesses can rely on AWS to keep their applications and data accessible and secure at all times.

4. Security: AWS has implemented a comprehensive security framework that includes multiple layers of security controls and regular security audits. This makes it a trusted platform for businesses that need to protect sensitive data.

5. Cost-effective: AWS offers a pay-as-you-go pricing model, which means that businesses only pay for the computing resources they use. This can be a more cost-effective option than investing in and maintaining their own hardware and infrastructure.

6. Global Infrastructure: AWS has a global infrastructure with data centers located in different regions around the world, which allows businesses to easily deploy their applications closer to their users for better performance.

7. Integration: AWS integrates with a wide range of third-party tools and services, making it easier for businesses to build, deploy, and manage their applications and services.

Overall, AWS provides businesses with a powerful, flexible, and cost-effective platform for running their applications and managing their data in the cloud.

What are all the Services AWS provides?

AWS offers a wide range of cloud-based services across multiple categories. Here's a brief overview of some of the most popular services:

1. Compute Services:

- Amazon Elastic Compute Cloud (EC2)
- Amazon Elastic Container Service (ECS)
- Amazon Elastic Kubernetes Service (EKS)
- AWS Lambda
- AWS Batch

2. Storage and Content Delivery Services:

- Amazon Simple Storage Service (S3)
- Amazon Elastic Block Store (EBS)
- Amazon Elastic File System (EFS)
- Amazon Glacier
- Amazon CloudFront

3. Database Services:

- Amazon Relational Database Service (RDS)
- Amazon Neptune

- Amazon DynamoDB - Amazon DocumentDB

- Amazon ElastiCache

4. Networking Services:

- Amazon Virtual Private Cloud (VPC) - AWS Global Accelerator

- AWS Direct Connect - Amazon Elastic Load Balancing (ELB)

- Amazon Route 53

5. Security, Identity, and Compliance Services:

- AWS Identity and Access Management (IAM) - Amazon Inspector

- AWS Key Management Service (KMS) - Amazon CloudWatch

- AWS Certificate Manager

6. Management and Governance Services:

- AWS CloudFormation - AWS Config

- Amazon CloudWatch - AWS Trusted Advisor

- AWS Systems Manager

7. Analytics Services:

- Amazon Kinesis - Amazon Elasticsearch Service

- Amazon Athena - Amazon EMR - Amazon QuickSight

8. Machine Learning Services:

- Amazon SageMaker - Amazon Transcribe

- Amazon Comprehend - Amazon Polly

- Amazon Recognition

These are just some of the many services that AWS offers. Each of these services is designed to help businesses and organizations leverage the power of the cloud to improve their operations, reduce costs, and increase their agility.

What is Cloud Computing?

Cloud computing refers to the delivery of computing services over the internet, including storage, processing power, and applications. Instead of running software and storing data on local servers or personal computers, cloud computing allows users to access these resources remotely from a network of servers hosted on the internet.

Cloud computing providers offer a range of services, including Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS).

IaaS provides access to computing infrastructure such as virtual machines, storage, and networking resources,

While, **PaaS provides a platform for developers to build, test, and deploy applications**.

SaaS allows users to access and use software applications hosted in the cloud.

Features of Cloud Computing?

Some of the key features of cloud computing includes:

1. On-demand self-service: Cloud computing allows users to provision computing resources, such as processing power and storage, on-demand without requiring human interaction with service providers.

2. Broad network access: Cloud computing resources are accessible via the internet, which allows users to access them from anywhere and with any device.

3. Resource pooling: Cloud computing resources are pooled together to serve multiple customers simultaneously, with each customer receiving a portion of the overall pool of resources.

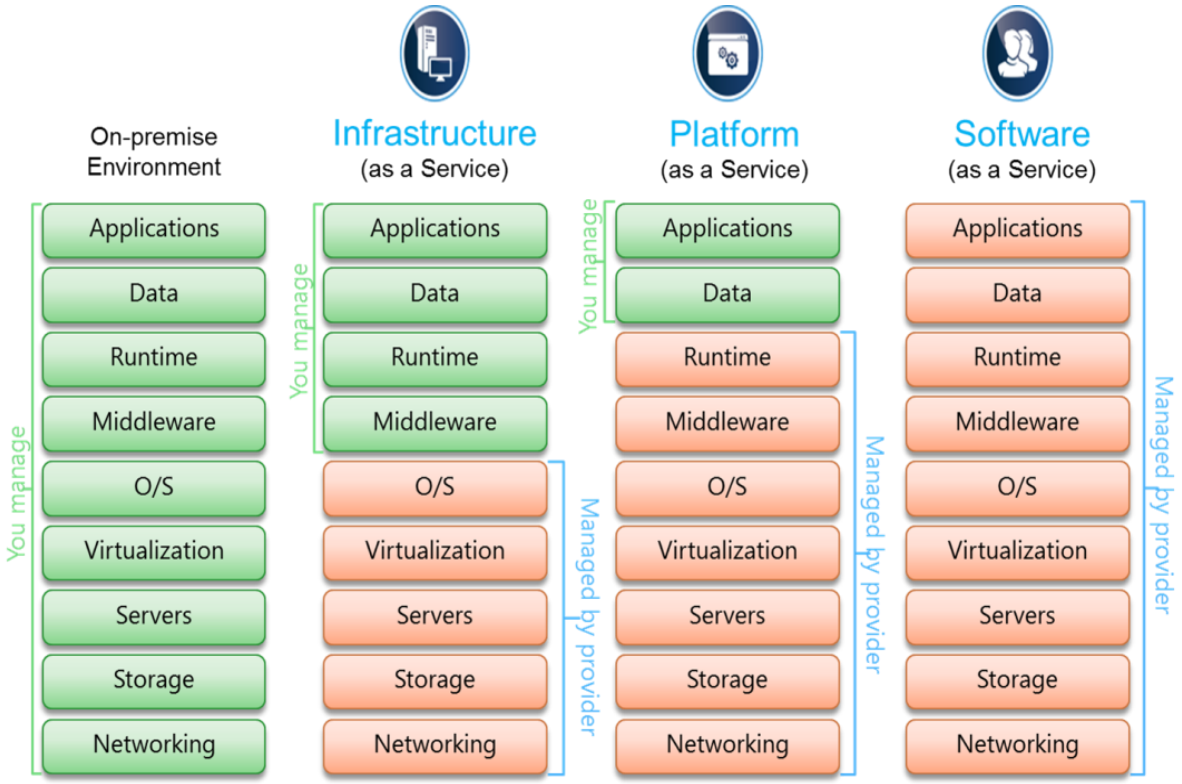
4. Rapid elasticity: Cloud computing resources can be quickly scaled up or down to meet changing demands, without requiring significant changes to the underlying infrastructure.

5. Measured service: Cloud computing providers typically charge users based on the amount of resources they use, allowing users to pay only for what they consume.

6. Multi-tenancy: Cloud computing resources are often shared among multiple customers, allowing providers to achieve economies of scale and reduce costs.

7. Service-level agreements (SLAs): Cloud computing providers typically offer SLAs that specify the level of service that users can expect, including uptime guarantees, response times, and other metrics.

8. Data security: Cloud computing providers typically offer robust security measures to protect customer data, including encryption, firewalls, intrusion detection and prevention, and other security measures.



Difference between IaaS, PasS, SaaS?

The main differences between these models are as follows:

1. **IaaS:** Infrastructure as a Service provides users with access to computing resources such as virtual machines, storage, and networking. With IaaS, users have full control over their computing environment, including operating systems, applications, and data. Users are responsible for managing and maintaining their own infrastructure, including installing software updates and patches.

2. PaaS: Platform as a Service provides users with a platform for building, testing, and deploying applications. With PaaS, users can focus on developing their applications, while the cloud provider handles the underlying infrastructure, such as operating systems, middleware, and runtime environments. This allows developers to be more productive and focus on developing high-quality applications without worrying about the underlying infrastructure.

3. SaaS: Software as a Service provides users with access to software applications that are hosted in the cloud. With SaaS, users can access and use software applications without having to install or maintain any software on their own devices. The cloud provider is responsible for managing and maintaining the underlying infrastructure, including hardware, software, and security.

In short, IaaS provides access to computing infrastructure, PaaS provides a platform for building and deploying applications, and SaaS provides access to software applications.

How many types of Clouds are there?

There are several types of clouds, each with its unique characteristics and functions. Here are the three primary types of clouds:

1. Public Cloud: A public cloud is a service provided by a third-party provider accessible to anyone who has an internet connection. The infrastructure is shared among various users and can offer services like Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS).

2. Private Cloud: A private cloud is a cloud infrastructure that is only accessible by a single organization or company. The infrastructure can be managed by the organization itself or by a third-party provider. It ideals with

organizations that have sensitive data or workloads and require a higher level of security.

3. Hybrid Cloud: A hybrid cloud is a combination of public and private cloud infrastructures. The organization can use a public cloud for non-sensitive data and a private cloud for sensitive data. It ideals with organizations that have varying workloads and require a flexible and scalable infrastructure.

What is Deployment model in cloud computing?

In cloud computing, a deployment model refers to the specific arrangement or configuration of cloud resources and services that are utilized to meet the needs of an organization or application. **Deployment models define how cloud computing resources are provisioned, managed, and accessed by users.**

There are generally four main deployment models in cloud computing:

1. Public Cloud: In a public cloud, the cloud infrastructure is owned and operated by a cloud service provider (such as Amazon Web Services, Microsoft Azure, or Google Cloud Platform), and resources are shared among multiple organizations or users. The services and infrastructure are delivered over the internet, and users can access and utilize them on a pay-as-you-go basis. Public clouds are typically highly scalable and offer a wide range of services, making them suitable for most applications.

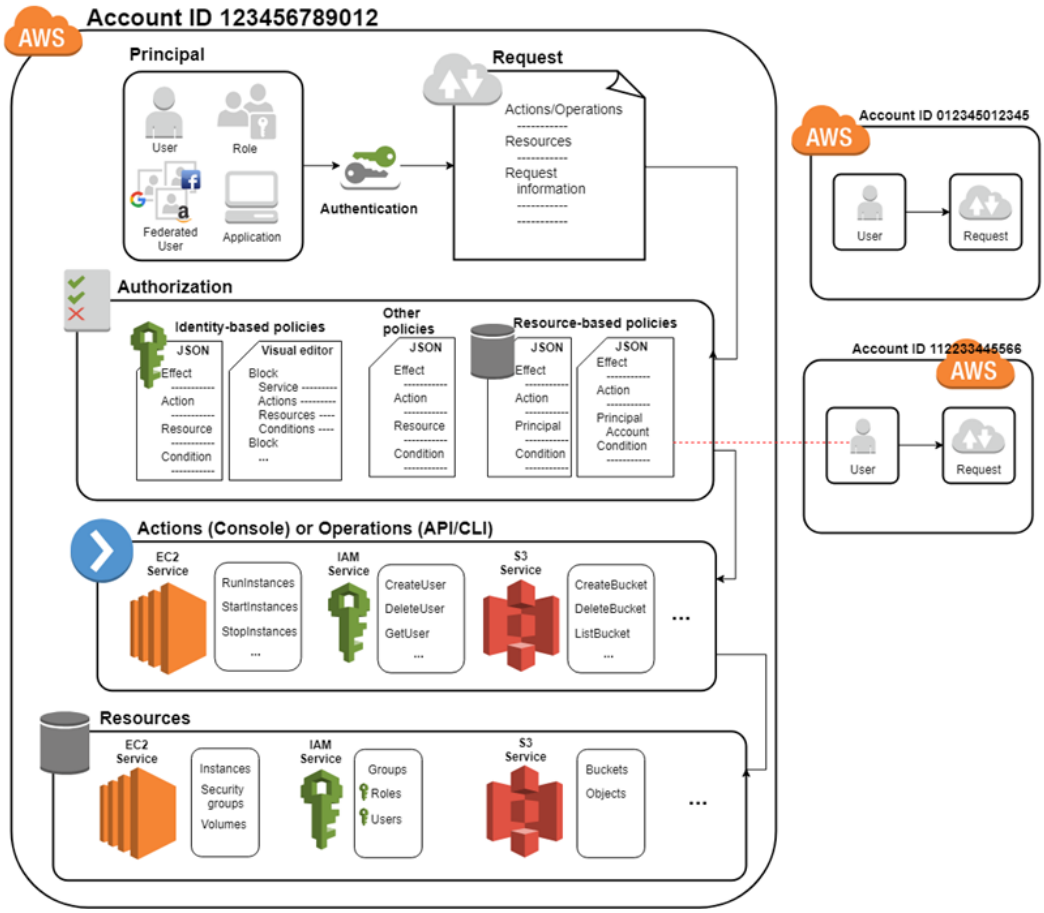
2. Private Cloud: A private cloud is dedicated to a single organization and is either managed internally or by a third-party provider. It offers the same benefits as a public cloud, such as scalability and flexibility, but is deployed within the organization's own data center or on-premises infrastructure. Private clouds are often preferred by organizations with specific security, compliance, or performance requirements that cannot be met in a public cloud environment.

3. Hybrid Cloud: A hybrid cloud combines both public and private cloud resources, allowing organizations to take advantage of the benefits of both models. It enables the seamless movement of data and applications between the public and private cloud environments, providing greater flexibility and scalability. Hybrid cloud deployments are useful for organizations that want to maintain control over sensitive data or have varying workload demands.

4. Multi-Cloud (hybrid): A multi-cloud deployment involves utilizing services and resources from multiple cloud service providers. Organizations can distribute their workloads across different clouds, selecting the most suitable provider for each specific application or task. Multi-cloud deployments offer greater redundancy, avoid vendor lock-in, and allow organizations to leverage the unique features and capabilities of different cloud platforms.

It's important to note that these deployment models are not mutually exclusive, and organizations can adopt a combination of models to meet their specific requirements. The choice of deployment model depends on factors such as security needs, compliance regulations, scalability requirements, budget considerations, and overall IT strategy.

IAM



What is IAM?

IAM provides the infrastructure necessary to control authentication and authorization for your AWS account.

IAM is a web service that helps you securely control access to AWS resources. With IAM, you can centrally manage permissions that control which AWS resources users can access. You use IAM to control who is authenticated (signed in) and authorized (has permissions) to use resources.

[**authentication** is the process of verifying who someone is, whereas **authorization** is the process of verifying what specific applications, files, and data a user has access to.]

What are the components of IAM?

1. Users: IAM allows you to create individual users and manage their credentials, such as passwords, access keys, and multi-factor authentication

devices. Users can be grouped and assigned specific permissions based on their roles and responsibilities.

2. Groups: You can organize users into groups to simplify the management of permissions. By assigning permissions to groups, you can grant or deny access to multiple users at once. Users within a group inherit the permissions assigned to the group.

3. Roles: IAM roles define a set of permissions that determine what actions can be performed on AWS resources. Unlike users, roles are not associated with specific individuals, but rather with AWS services or resources. Roles are used for granting permissions to services, applications, or AWS resources themselves.

4. Policies: IAM policies are JSON documents that define permissions. Policies can be attached to users, groups, or roles. They define the actions that are allowed or denied, the resources those actions can be performed on, and any conditions that must be met.

5. Access Keys: IAM allows users to generate access keys, which consist of an access key ID and a secret access key. Access keys are used to programmatically access AWS resources through APIs, SDKs, and command-line tools.

6. Multi-factor Authentication (MFA): IAM supports the use of MFA to provide an additional layer of security. MFA requires users to provide a second form of authentication, such as a time-based, one-time password generated by a virtual or hardware MFA device.

7. Identity Providers (IdPs): IAM allows you to integrate with external identity providers, such as Microsoft Active Directory or Facebook, through the use of identity federation. This enables users to sign in to AWS using their existing credentials from the identity provider.

These components work together to help us manage access and permissions to your AWS resources in a secure and controlled manner.

Categories of IAM

IAM classified into four major categories: **authentication, authorization, user management, central user repository.**

What is Centralized user repository

It is recommended to use centralized user repositories such as AWS Directory Services (Active Directory/Simple AD), Okta, Auth0, PingIdentity, OneLogin, Azure Active Directory, IBM Cloud Identity, etc**. ideally integrated to AWS IAM Identity Center (succesor to AWS Single Sign-On) to provide temporary credentials to users accessing AWS.**

It is also important that the repository is integrated with the human resources management system to propagate employee terminations (either through an identity management system or directly into the centralized authentication repository).

Features of IAM?

1) Shared access to your AWS account: You can grant other people permission to administer and use resources in your AWS account without having to share your password or access key.

2) Granular permissions: You can grant different permissions to different people for different resources.

3) Secure access to AWS resources for applications that run on EC2: You can use IAM features to securely provide credentials for applications that run on EC2 instances. These credentials provide permissions for your application to access other AWS resources. Examples include S3 buckets and DynamoDB tables.

4) Multi-factor authentication (MFA): You can add two-factor authentication to your account and to individual users for extra security.

5) Identity federation: You can allow users who already have passwords elsewhere—for example, in your corporate network or with an internet identity provider—**to get temporary access to your AWS account.**

6) Identity information for assurance: If you use AWS CloudTrail, you receive log records that include information about those who made requests for resources in your account. That information is based on IAM identities.

7) PCI DSS Compliance: IAM supports the processing, storage, and transmission of credit card data by a merchant or service provider, and has been validated as being compliant with Payment Card Industry (PCI) Data Security Standard (DSS). For more information about PCI DSS, including how to request a copy of the AWS PCI Compliance Package

When do I use IAM?

1. When you are performing different job functions
2. When you are authorized to access AWS resources
3. When you sign-in as an IAM user
4. When you assume an IAM role
5. When you create policies and permissions

Authentication is how you sign in to AWS using your identity credentials. You must be *authenticated*(signed in to AWS) as the AWS account root user, as an IAM user, or by assuming an IAM role.You can sign in to AWS as a federated identity by using credentials provided through an identity source.

What is Programmatic access?

Programmatic access is which gives us to manage aws resources from our Development environment and manage by writing code in JSON format.

Or

Programmatic access allows you to invoke actions on your AWS resources either through an application that you write or through a third-party tool. You use an access key ID and a secret access key to sign-in your requests for authorization to AWS.

What is Availability Zones?

Availability Zones are distinct locations within an AWS Region that are engineered to be isolated from failures in other Availability Zones. They provide inexpensive, low-latency network connectivity to other AZ in the same AWS Region.

Regions

Each Region is designed to be isolated from the other Regions. This achieves the greatest possible fault tolerance and stability.

When you view your resources, you see only the resources that are tied to the Region that you specified. This is because Regions are isolated from each other, and we don't automatically replicate resources across Regions.

- Each Region is a separate geographic area.
- Availability Zones are multiple, isolated locations within each Region.
- Local Zones provide you the ability to place resources, such as compute and storage, in multiple locations closer to your end users.
- AWS Outposts brings native AWS services, infrastructure, and operating models to virtually any data centre, co-location space, or on-premises facility.
- Wavelength Zones allow developers to build applications that deliver ultra-low latencies to 5G devices and end users. Wavelength deploys standard AWS compute and storage services to the edge of telecommunication carriers' 5G networks.

AWS operates state-of-the-art, highly available data centres. Although rare, failures can occur that affect the availability of instances that are in the same location. If you host all of your instances in a single location that is affected by a failure, none of your instances would be available.

How are AWS Local Zones different than Availability Zones?

Local Zones are designed to bring the core services needed for the latency sensitive portions of your workload closer to end-users,
Whereas, **Availability Zones** provide access to the full array of AWS services.

What is a policy?

Policies are JSON documents in AWS that let you specify who has access to AWS resources, and what actions they can perform on those resources. You can attach a policy to an identity or resource to define their permissions. AWS evaluates these policies when the IAM principal makes a request. Permissions in the policies determine whether the request is allowed or denied.

What is the maximum length of AWS policy name? (Optional)

Maximum length of a policy is 6144 charachters. A list of tags that you want to attach to the new IAM customer managed policy. Each tag consists of a key name and an associated value.

What are the different types of policies in AWS?

AWS supports six types of policies: **identity-based policies, resource-based policies, permissions boundaries, Organizations SCPs (service control policy), ACLs, and session policies.**

What is Identity based policies?

Identity-based policies are attached to an IAM user, group, or role. These policies let you specify what that identity can do (its permissions).

What is an inline policy in AWS?

An inline policy is a policy created for a single IAM identity (a user, group, or role). Inline policies maintain a strict one-to-one relationship between a policy and an identity. They are deleted when you delete the identity.

What is the difference between inline and customer managed policy?

A customer managed policy is a standalone policy that you administer in your own AWS account. An inline policy is embedded in an IAM identity (a user, group, or role).

Why we use an inline policy in AWS?

When you use an inline policy, the permissions in the policy cannot be inadvertently attached to the wrong identity. When you use the AWS management Console to delete that identity, the policies embedded in the identity are deleted as well because they are part of the principal entity.

What is a federated user?

Federated identity allows authorized users to access multiple applications and domains using a single set of credentials. It links a user's identity across

multiple identity management systems so they can access different applications securely and efficiently.

In short: federated user can login to different accounts from a single account

How to create federated user?

1. Create a role: aws-iam-role
2. Attach a permissions policy to the role: role-policy or
   Create an inline permissions policy for the role:inline-policy
3. (Optional) Set the permission-boundaries for the role: role-boundaries

What is ARN in aws?

Amazon Resource Names (ARNs) uniquely identify AWS resources. **ARN provide a consistent and structured way to identify and reference aws resources across different services and regions**

arn:partition:service:region:account-id:resource-id

What are the 5 categories of multifactor authentication? (Optional)

The five main authentication factor categories are knowledge factors, possession factors, inherence factors, location factors, and behavior factors.

Three Main Types of MFA Authentication Methods? (Optional)

- Things you know (knowledge), such as a password or PIN.
- Things you have (possession), such as a badge or smartphone.
- Things you are (inherence), such as a biometric like fingerprints or voice recognition

Note: You can register upto 8 MFA devices of any combination of the currently supported MFA types with your AWS acc. Root and I am user.

How many types of IAM roles are there?

There are five types of roles that you can create:

1. User Roles: These roles are associated with IAM users within your AWS account. IAM user roles allow you to grant specific permissions to users temporarily by assuming a role. Users can switch their own permissions to

assume a role, which grants them the permissions assigned to that role for the duration of the session.

2. Service Roles: These roles are used by AWS services to interact with other AWS services on your behalf. When you configure an AWS service to perform actions or access resources in your account, you can assign an IAM service role to the service. The service then assumes that role and uses its permissions to perform actions or access resources.

3. Cross-Account Roles: Cross-account roles enable you to grant permissions to users in another AWS account. You can create a role in one AWS account and grant another AWS account access to assume that role. This allows users in the second account to access resources in the first account without having to create IAM users in both accounts. Cross-account roles are commonly used in scenarios where multiple AWS accounts need to collaborate or share resources securely.

4. Service-linked Role **5. Web-Identity Role.**

How many phases are there in IAM? (Optional)

Successful IAM projects take a gradual, methodical approach to the implementation and are broken down into five main phases: Analysis, Architecture, Implementation, Testing, and transition to Support.

How many policies can an IAM role have? (Optional)

You can attach up to 20 managed policies to IAM roles and users.

What is role?

A role is an IAM identity that you can create in your account that has specific permissions. An IAM role has some similarities to an IAM user. **Roles and users** are both AWS identities with permission policies that determine what the identity can and cannot do in AWS.

However, instead of being uniquely associated with one person, a role can be assumed by anyone who needs it. A role does not have standard long-term credentials such as a password or access keys associated with it. Instead, when you assume a role, it provides you with temporary security credentials for your role session.

You can assume a role by calling an AWS CLI or API operation or by using a custom URL.

What are tags?

A tag is a *key-value pair* applied to a resource to hold metadata about that resource. Each tag is a label consisting of a key and an optional value. Not all services and resource types currently support tags

Tags that a user creates and applies to AWS resources using the AWS CLI, API, or the AWS Management Console are known as *user-defined* tags. Several AWS services, such as AWS CloudFormation, Elastic Beanstalk, and Auto Scaling, automatically assign tags to resources that they create and manage.

Why Use Tags on AWS?

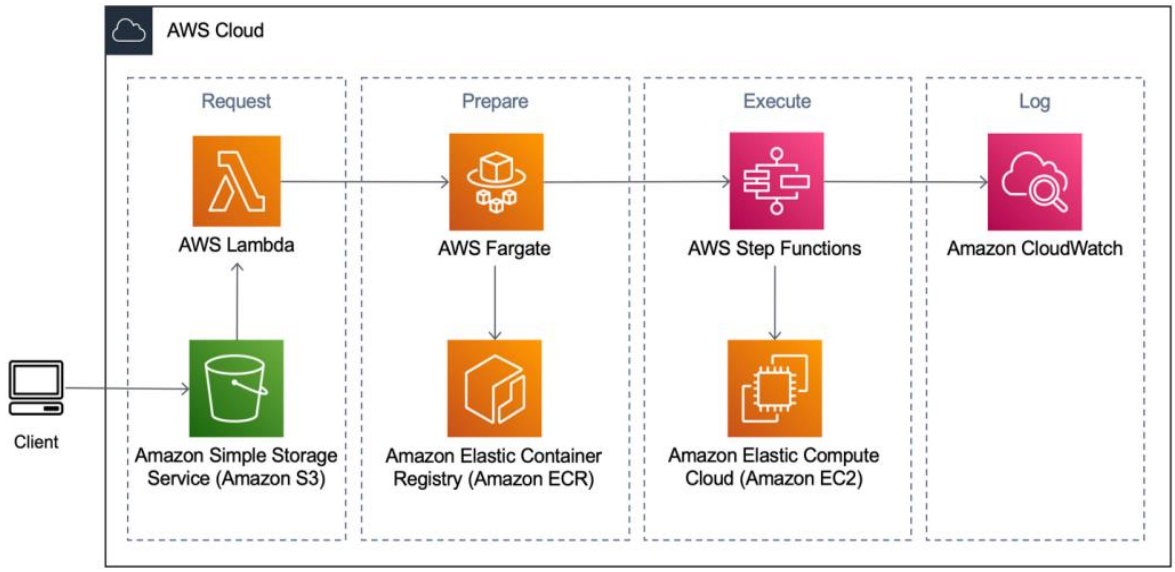
Tags are metadata that can be assigned to resources on AWS. Each tag is a label consisting of a user-defined key and optional value. Tags provide users with information and context about specific cloud resource

What is Permission boundaries

A permission boundary is an advanced feature for using a managed policy to set the maximum permissions that an identity-based policy can grant to an IAM entity. An entity's permission boundary allows it to perform only the actions that are allowed by both its identity-based policies and its permissions boundaries.

Note: A permissions boundary controls the maximum permissions that a role can have. Permissions boundaries are an advanced AWS feature.

S3



What is S3

S3 is an object storage service that offers **industry-leading scalability, data availability, security and performance**. Customers of all sizes and industries can use S3 to store and protect any amount of data for a range of use cases, such as data lakes, websites, mobile applications, backup and restore, archive, enterprise applications, IoT devices, and big data analytics. S3 provides management features so that you can optimize, organize, and configure access to your data to meet your specific business, organizational, and compliance requirements.

What are the Features of S3

1. **Storage classes**
2. **Storage management :**
   S3 life cycle, object lock, s3 replication, batch operation.
3. **Access management and security:**
   S3 public block access, IAM, bucket policies, access points, ACLs, object ownership, access analyser.
4. **Data processing:**
   Object lamda, event notification,
5. **Storage logging and monitoring:**
   - Automated monitoring tools: cloudwatch metrics for s3, cloudtrails

- Manual monitoring tools: server access logging, trust advisor.

6. **Analytics and insights**
7. **Strong consistency**

imp

What is Versioning in S3?

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use the S3 Versioning feature to preserve, retrieve, and restore every version of every object stored in your buckets. With versioning you can recover more easily from both unintended user actions and application failures. After versioning is enabled for a bucket, if S3 receives multiple write requests for the same object simultaneously, it stores all those objects. Versioning-enabled buckets can help you recover objects from accidental deletion or overwrite.

imp

S3 storage classes?

The S3 storage classes include (7):

1. **S3 Standard** for frequently accessed data,

2. **Intelligent-Tiering** for automatic cost savings for data with unknown or changing access patterns,

3. **Standard-Infrequent Access (S3 Standard-IA) and**

4. **One Zone-Infrequent Access (S3 One Zone-IA)** for less frequently accessed data,

5. **Glacier** Instant Retrieval for archive data that needs immediate access,

6. **Glacier Flexible Retrieval (formerly S3 Glacier)** for rarely accessed long-term data that does not require immediate access,

7. **Glacier Deep Archive (S3 Glacier Deep Archive)** for long-term archive and digital preservation with retrieval in hours at the lowest cost storage in the cloud.

If you have data residency requirements that can't be met by an existing AWS Region, you can use the S3 Outposts storage class to store your S3 data on premises. Amazon S3 also offers capabilities to manage your data throughout its lifecycle. Once an S3 Lifecycle policy is set, your data will automatically transfer to a different storage class without any changes to your application.

1. **S3 Standard (S3 Standard)**

S3 Standard offers high durability, availability, and performance object storage for frequently accessed data. Because it delivers low latency and high throughput, S3 Standard is appropriate for a wide variety of use cases, including cloud applications, dynamic websites, content distribution, mobile and gaming applications, and big data analytics. S3 Storage Classes can be configured at the object level and a single bucket can contain objects stored across S3 Standard, S3 Intelligent-Tiering, S3 Standard-IA, and S3 One Zone-IA. You can also use S3 Lifecycle policies to automatically transition objects between storage classes without any application changes.

Key Features:

- Low latency and high throughput performance
- Designed for durability of 99.999999999% of objects across multiple Availability Zones
- Resilient against events that impact an entire Availability Zone
- Designed for 99.99% availability over a given year
- Backed with the Amazon S3 Service Level Agreement for availability
- Supports SSL for data in transit and encryption of data at rest
- S3 Lifecycle management for automatic migration of objects to other S3 Storage Classes unknown or changing access.

2. **Amazon S3 Intelligent-Tiering (S3 Intelligent-Tiering)**

S3 Intelligent-Tiering is the first cloud storage that automatically reduces your storage costs on a granular object level by automatically moving data to the most cost-effective access tier based on access frequency, without

performance impact, retrieval fees, or operational overhead. S3 Intelligent-Tiering delivers milliseconds latency and high throughput performance for frequently, infrequently, and rarely accessed data in the Frequent, Infrequent, and Archive Instant Access tiers. You can use S3 Intelligent-Tiering as the default storage class for virtually any workload, especially data lakes, data analytics, new applications, and user-generated content.

S3 Intelligent-Tiering automatically stores objects in three access tiers: one tier that is optimized for frequent access, a 40% lower-cost tier that is optimized for infrequent access, and a 68% lower-cost tier optimized for rarely accessed data. S3 Intelligent-Tiering monitors access patterns and moves objects that have not been accessed for 30 consecutive days to the Infrequent Access tier and after 90 days of no access to the Archive Instant Access tier. For data that does not require immediate retrieval, you can set up S3 Intelligent-Tiering to monitor and automatically move objects that aren't accessed for 180 days or more to the Deep Archive Access tier to realize up to 95% in storage cost savings.

There are no retrieval charges in S3 Intelligent-Tiering. If an object in the Infrequent or Archive Instant Access tier is accessed later, it's automatically moved back to the Frequent Access tier. If the object you're retrieving is stored in the optional Deep Archive tiers, before you can retrieve the object, you must first restore a copy using RestoreObject.

Key Features:

- Frequent, Infrequent, and Archive Instant Access tiers have the same low-latency and high-throughput performance of S3 Standard
- The Infrequent Access tier saves up to 40% on storage costs
- The Archive Instant Access tier saves up to 68% on storage costs
- Opt-in asynchronous archive capabilities for objects that become rarely accessed
- Deep Archive Access tier has the same performance as Glacier Deep Archive and saves up to 95% for rarely accessed objects
- Designed for durability of 99.999999999% of objects across multiple Availability Zones and for 99.9% availability over a given year

- Backed with the Amazon S3 Service Level Agreement for availability
- Small monthly monitoring and auto tiering charge
- No operational overhead, no lifecycle charges, no retrieval charges, and no minimum storage duration
- Objects smaller than 128KB can be stored in S3 Intelligent-Tiering but will always be charged at the Frequent Access tier rates, and are not charged the monitoring and automation charge.

Infrequent access

3. Amazon S3 Standard-Infrequent Access (S3 Standard-IA)

S3 Standard-IA is for data that is accessed less frequently, but requires rapid access when needed. S3 Standard-IA offers the high durability, high throughput, and low latency of S3 Standard, with a low per GB storage price and per GB retrieval charge. This combination of low cost and high performance make S3 Standard-IA ideal for long-term storage, backups, and as a data store for disaster recovery files. S3 Storage Classes can be configured at the object level and a single bucket can contain objects stored across S3 Standard, S3 Intelligent-Tiering, S3 Standard-IA, and S3 One Zone-IA. You can also use S3 Lifecycle policies to automatically transition objects between storage classes without any application changes.

Key Features:

- Same low latency and high throughput performance of S3 Standard
- Designed for durability of 99.999999999% of objects across multiple Availability Zones
- Resilient against events that impact an entire Availability Zone
- Data is resilient in the event of one entire Availability Zone destruction
- Designed for 99.9% availability over a given year
- Backed with the Amazon S3 Service Level Agreement for availability
- Supports SSL for data in transit and encryption of data at rest

- S3 Lifecycle management for automatic migration of objects to other S3 Storage Classes
  Amazon S3 One Zone-Infrequent Access (S3 One Zone-IA)

  S3 One Zone-IA is for data that is accessed less frequently, but requires rapid access when needed. Unlike other S3 Storage Classes which store data in a minimum of three Availability Zones (AZs), S3 One Zone-IA stores data in a single AZ and costs 20% less than S3 Standard-IA. S3 One Zone-IA is ideal for customers who want a lower-cost option for infrequently accessed data but do not require the availability and resilience of S3 Standard or S3 Standard-IA. It's a good choice for storing secondary backup copies of on-premises data or easily re-creatable data. You can also use it as cost-effective storage for data that is replicated from another AWS Region using S3 Cross-Region Replication.

  S3 One Zone-IA offers the same high durability†, high throughput, and low latency of S3 Standard, with a low per GB storage price and per GB retrieval charge. S3 Storage Classes can be configured at the object level, and a single bucket can contain objects stored across S3 Standard, S3 Intelligent-Tiering, S3 Standard-IA, and S3 One Zone-IA. You can also use S3 Lifecycle policies to automatically transition objects between storage classes without any application changes.

  Key Features:

- Same low latency and high throughput performance of S3 Standard
- Designed for durability of 99.999999999% of objects in a single Availability Zone†
- Designed for 99.5% availability over a given year
- Backed with the Amazon S3 Service Level Agreement for availability
- Supports SSL for data in transit and encryption of data at rest
- S3 Lifecycle management for automatic migration of objects to other S3 Storage Classes
  † Because S3 One Zone-IA stores data in a single AWS Availability Zone, data stored in this storage class will be lost in the event of Availability Zone destruction.**Archive**

The Amazon S3 Glacier storage classes

are purpose-built for data archiving, and are designed to provide you with the highest performance, the most retrieval flexibility, and the lowest cost archive storage in the cloud. You can choose from three archive storage classes optimized for different access patterns and storage duration. For archive data that needs immediate access, such as medical images, news media assets, or genomics data, choose the S3 Glacier Instant Retrieval storage class, an archive storage class that delivers the lowest cost storage with milliseconds retrieval. For archive data that does not require immediate access but needs the flexibility to retrieve large sets of data at no cost, such as backup or disaster recovery use cases, choose S3 Glacier Flexible Retrieval (formerly S3 Glacier), with retrieval in minutes or free bulk retrievals in 5—12 hours. To save even more on long-lived archive storage such as compliance archives and digital media preservation, choose S3 Glacier Deep Archive, the lowest cost storage in the cloud with data retrieval from 12—48 hours.
Amazon S3 Glacier Instant Retrieval

Amazon S3 Glacier Instant Retrieval is an archive storage class that delivers the lowest-cost storage for long-lived data that is rarely accessed and requires retrieval in milliseconds. With S3 Glacier Instant Retrieval, you can save up to 68% on storage costs compared to using the S3 Standard-Infrequent Access (S3 Standard-IA) storage class, when your data is accessed once per quarter. S3 Glacier Instant Retrieval delivers the fastest access to archive storage, with the same throughput and milliseconds access as the S3 Standard and S3 Standard-IA storage classes. S3 Glacier Instant Retrieval is ideal for archive data that needs immediate access, such as medical images, news media assets, or user-generated content archives. You can upload objects directly to S3 Glacier Instant Retrieval, or use S3 Lifecycle policies to transfer data from the S3 storage classes

Key Features:

- Data retrieval in milliseconds with the same performance as S3 Standard
- Designed for durability of 99.999999999% of objects across multiple Availability Zones
- Data is resilient in the event of the destruction of one entire Availability Zone

- Designed for 99.9% data availability in a given year
- 128 KB minimum object size
- Backed with the Amazon S3 Service Level Agreement for availability
- S3 PUT API for direct uploads to S3 Glacier Instant Retrieval, and S3 Lifecycle management for automatic migration of objects

Amazon S3 Glacier Flexible Retrieval (Formerly S3 Glacier)

S3 Glacier Flexible Retrieval delivers low-cost storage, up to 10% lower cost (than S3 Glacier Instant Retrieval), for archive data that is accessed 1—2 times per year and is retrieved asynchronously. For archive data that does not require immediate access but needs the flexibility to retrieve large sets of data at no cost, such as backup or disaster recovery use cases, S3 Glacier Flexible Retrieval (formerly S3 Glacier) is the ideal storage class. S3 Glacier Flexible Retrieval delivers the most flexible retrieval options that balance cost with access times ranging from minutes to hours and with free bulk retrievals. It is an ideal solution for backup, disaster recovery, offsite data storage needs, and for when some data occasionally need to be retrieved in minutes, and you don't want to worry about costs. S3 Glacier Flexible Retrieval is designed for 99.999999999% (11 9s) of data durability and 99.99% availability by redundantly storing data across multiple physically separated AWS Availability Zones in a given year. For more information, visit the Amazon S3 Glacier storage classes page »

Key Features:

- Designed for durability of 99.999999999% of objects across multiple Availability Zones
- Data is resilient in the event of one entire Availability Zone destruction
- Supports SSL for data in transit and encryption of data at rest
- Ideal for backup and disaster recovery use cases when large sets of data occasionally need to be retrieved in minutes, without concern for costs
- Configurable retrieval times, from minutes to hours, with free bulk retrievals
- S3 PUT API for direct uploads to S3 Glacier Flexible Retrieval, and S3 Lifecycle management for automatic migration of objects

Amazon S3 Glacier Deep Archive

S3 Glacier Deep Archive

is Amazon S3's lowest-cost storage class and supports long-term retention and digital preservation for data that may be accessed once or twice in a year. It is designed for customers—particularly those in highly-regulated industries, such as financial services, healthcare, and public sectors—that retain data sets for 7—10 years or longer to meet regulatory compliance requirements. S3 Glacier Deep Archive can also be used for backup and disaster recovery use cases, and is a cost-effective and easy-to-manage alternative to magnetic tape systems, whether they are on-premises libraries or off-premises services. S3 Glacier Deep Archive complements Amazon S3 Glacier, which is ideal for archives where data is regularly retrieved and some of the data may be needed in minutes. All objects stored in S3 Glacier Deep Archive are replicated and stored across at least three geographically-dispersed Availability Zones, protected by 99.999999999% of durability, and can be restored within 12 hours. For more information, visit the Amazon S3 Glacier storage classes page »

Key Features:

- Designed for durability of 99.999999999% of objects across multiple Availability Zones
- Lowest cost storage class designed for long-term retention of data that will be retained for 7-10 years
- Ideal alternative to magnetic tape libraries
- Retrieval time within 12 hours
- S3 PUT API for direct uploads to S3 Glacier Deep Archive, and S3 Lifecycle management for automatic migration of objects

S3 on Outposts

S3 Outposts

Amazon S3 on Outposts delivers object storage to your on-premises AWS Outposts environment. Using the S3 APIs and features available in AWS Regions today, S3 on Outposts makes it easy to store and retrieve data on your

Outpost, as well as secure the data, control access, tag, and report on it. S3 on Outposts provides a single Amazon S3 storage class, named 'OUTPOSTS', which uses the S3 APIs, and is designed to durably and redundantly store data across multiple devices and servers on your Outposts. The S3 Outposts storage class is ideal for workloads with local data residency requirements, and to satisfy demanding performance needs by keeping data close to on-premises applications.

Key Features:

- S3 Object compatibility and bucket management through the S3 SDK
- Designed to durably and redundantly store data on your Outposts
- Encryption using SSE-S3 and SSE-C
- Authentication and authorization using IAM, and S3 Access Points
- Transfer data to AWS Regions using AWS DataSync
- S3 Lifecycle expiration actions

What is Replication in S3?

Replication enables automatic, asynchronous copying of objects across S3 buckets. **Buckets that are configured for object replication can be owned by the same AWS account or by different accounts**. You can replicate objects to a single destination bucket or to multiple destination buckets. The destination buckets can be in different AWS Regions or within the same Region as the source bucket.

What is objects in S3?

An object is a file or any metadata that describes that file. To store an object in S3, you create a bucket and then upload the object to the bucket. When the object is in the bucket, you can open it, download it, and move it. When you no longer need an object or a bucket, you can clean up your resources.

What is grantee user in s3?

A grantee can be an AWS account or one of the predefined S3 groups. You grant permission to an AWS account using the email address or the canonical user ID. However, if you provide an email address in your grant request, S3 finds the canonical user ID for that account and adds it to the ACL.

What is static web hosting in AWS?

Static websites deliver HTML, JavaScript, images, video and other files to your website visitors. Static websites are very low cost, provide high-levels of reliability, require almost no IT administration, and scale to handle enterprise-level traffic with no additional work.

S3 storage management :
Object lock , life cycle, replication, batch operation

What is API

(**Application Program Interface**) allows software programs to communicate making them more functional.

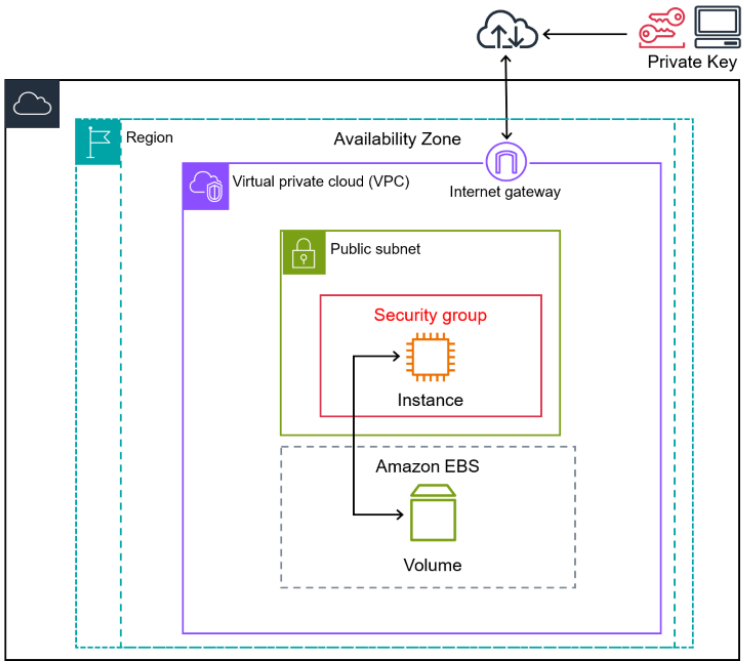
What is API Gateway

An API gateway accepts and processes concurrent API calls, which happen when APIs submit requests to a server. A user creates, manages and maintains APIs with the API Gateway.

Or

API Gateway is a fully managed service that makes it easy for developers to create, publish, maintain, monitor and secure APIs at any scale. APIs act as the “front door ” for the applications to access data, business logic, or functionally from your backend services.

EC2



What is EC2?

Amazon Elastic Compute Cloud (EC2) provides scalable computing capacity in the AWS Cloud. Using EC2 eliminates your need to invest in hardware up front, so you can develop and deploy applications faster. You can use EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage. EC2 enables you to scale up or down to handle changes in requirements or spikes in popularity, reducing your need to forecast traffic.

Features of EC2?

1) Virtual computing environments, known as ***instances.***
2) Preconfigured templates for your instances, known as ***Amazon Machine Images (AMIs)***, that package the bits you need for your server (including the operating system and additional software).
3) Various configurations of CPU, memory, storage, and networking capacity for your instances, known as ***instance types.***

4) Secure login information for your instances using ***key pairs*** *which* contains public key and private key. (AWS stores the public key, and you store the private key in a secure place).

5) Storage volumes for temporary data that is deleted when you stop, hibernate, or terminate your instance, known as ***instance store volumes.***

6) Persistent storage volumes for your data using Elastic Block Store (EBS), known as ***Amazon EBS volumes.***

7) Multiple physical locations for your resources, such as instances and EBS volumes, known as ***Regions* and *Availability Zones.***

8) A firewall that enables you to specify the protocols, ports, and source IP ranges that can reach your instances using ***security groups.***

9) Static IPv4 addresses for dynamic cloud computing, known as ***Elastic IP addresses.***

10) Metadata, known as ***tags,*** that you can create and assign to your EC2 resources.

11) Virtual networks you can create that are logically isolated from the rest of the AWS Cloud, and that you can optionally connect to your own network, known as ***virtual private clouds* (VPCs)**

What is Hypervisor?

A hypervisor is software that you can use to run multiple virtual machines on a single physical machine. Every virtual machine has its own operating system and applications. The hypervisor allocates the underlying physical computing resources such as CPU and memory to individual virtual machines as required.

Available zones:

Code	Name	Opt-in Status
us-east-2	US East (Ohio)	Not required
us-east-1	US East (N. Virginia)	Not required
us-west-1	US West (N. California)	Not required
us-west-2	US West (Oregon)	Not required
af-south-1	Africa (Cape Town)	Required
ap-east-1	Asia Pacific (Hong Kong)	Required
ap-south-2	Asia Pacific (Hyderabad)	Required
ap-southeast-3	Asia Pacific (Jakarta)	Required
ap-southeast-4	Asia Pacific (Melbourne)	Required
ap-south-1	Asia Pacific (Mumbai)	Not required
ap-northeast-3	Asia Pacific (Osaka)	Not required
ap-northeast-2	Asia Pacific (Seoul)	Not required
ap-southeast-1	Asia Pacific (Singapore)	Not required
ap-southeast-2	Asia Pacific (Sydney)	Not required
ap-northeast-1	Asia Pacific (Tokyo)	Not required
ca-central-1	Canada (Central)	Not required
eu-central-1	Europe (Frankfurt)	Not required
eu-west-1	Europe (Ireland)	Not required
eu-west-2	Europe (London)	Not required
eu-south-1	Europe (Milan)	Required
eu-west-3	Europe (Paris)	Not required
eu-south-2	Europe (Spain)	Required
eu-north-1	Europe (Stockholm)	Not required
eu-central-2	Europe (Zurich)	Required
me-south-1	Middle East (Bahrain)	Required
me-central-1	Middle East (UAE)	Required

What is the difference between IPv4 and IPv6?

IPv4 is composed of 32-bit address length and is the fourth version of the Internet Protocol (IP). IPv6 is composed of 128-bit address length and is the latest updated version of the Internet Protocol (IP).

What is region?

Regions are large and widely dispersed into separate geographic locations..

Or

EC2 is hosted in multiple locations world-wide. These locations are composed of AWS Regions, Availability Zones, Local Zones, AWS Outposts, and Wavelength Zones. Each Region is a separate geographic area. Availability Zones are multiple, isolated locations within each Region.

Availability Zones?

Each Region has multiple, isolated locations known as *Availability Zones*. Availability Zones are distinct locations within an AWS Region that are engineered to be isolated from failures in other Availability Zones

Local Zone?

A Local Zone is an extension of an Region in geographic proximity to your users. Local Zones have their own connections to the internet and support AWS Direct Connect, so that resources created in a Local Zone can serve local users with low-latency communications.

Wavelength Zone?

A Wavelength Zone is an isolated zone in the carrier location where the Wavelength infrastructure is deployed. Wavelength Zones are tied to a Region. A Wavelength Zone is a logical extension of a Region, and is managed by the control plane in the Region.

AWS Outpost?

AWS Outposts is a fully managed service that extends AWS infrastructure, services, APIs, and tools to customer premises. By providing local access to AWS managed infrastructure, AWS Outposts enables customers to build and run applications on premises using the same programming interfaces as in AWS Regions, while using local compute and storage resources for lower latency and local data processing needs.

An Outpost is a pool of AWS compute and storage capacity deployed at a customer site. AWS operates, monitors, and manages this capacity as part of an AWS Region. You can create subnets on your Outpost and specify them when you create AWS resources. Instances in Outpost subnets communicate with other instances in the AWS Region using private IP addresses, all within the same VPC.

What is AMI?

An *Amazon Machine Image (AMI)* is a template that contains a software configuration (for example, an operating system, an application server, and applications). From an AMI, you launch an ***instance,*** which is a copy of the

AMI running as a virtual server in the cloud. You can launch multiple instances of an AMI.

What is Snapshot?

A snapshot is a base feature for creating backups of your EBS volumes. A snapshot takes a copy of the EBS volume and places it in S3, where it is stored redundantly in multiple Availability Zones. The initial snapshot is a full copy of the volume; ongoing snapshots store incremental block-level changes only.

What is template?

A template is similar to a launch configuration, in that it specifies instance configuration information. It includes :

- the ID of Amazon Machine Image (AMI ID),
- the instance type,
- a key pair,
- security groups and
- other parameters used to launch instances.

What is Instance?

An instance is a virtual server in the cloud. Its configuration at launch is a copy of the AMI that you specified when you launched the instance.

You can launch different types of instances from a single AMI. An *instance type* essentially determines the hardware of the host computer used for your instance. Each instance type offers different compute and memory capabilities. Select an instance type based on the amount of memory and computing power that you need for the application or software that you plan to run on the instance.

Storage for your instance: The root device for your instance contains the image used to boot the instance. The root device is either an Elastic Block Store (Amazon EBS) volume or an instance store volume.

Your instance may include local storage volumes, known as instance store volumes, which you can configure at launch time with block device mapping.

What is Block device mapping?

A *block device* is a storage device that moves data in sequences of bytes or bits (blocks). These devices support random access and generally use buffered I/O. **Examples:** hard disks, CD-ROM drives, and flash drives.

A block device can be physically attached to a computer or accessed remotely as if it were physically attached to the computer.

EC2 supports two types of block devices:

1. Instance store volumes (virtual devices whose underlying hardware is physically attached to the host computer for the instance).
2. EBS volumes (remote storage devices).

What is EBS volume?

EBS volume is a durable, block-level storage device that you can attach to your instances. After you attach a volume to an instance, you can use it as you would use a physical hard drive. EBS volumes are flexible

You can use EBS volumes as primary storage for data that requires frequent updates, such as the system drive for an instance or storage for a database application. You can also use them for throughput-intensive applications that perform continuous disk scans.

Amazon provides the following EBS volume types:

- General Purpose SSD (gp2 and gp3),
- Provisioned IOPS SSD (io1 and io2),
- Throughput Optimized HDD (st1), Cold HDD (sc1), and
- Magnetic (standard).

You can attach multiple EBS volumes to a single instance. The volume and instance must be in the same Availability Zone.

What are the AWS EC2 Instance Types?

When you launch an instance, the *instance type* that you specify determines the hardware of the host computer used for your instance. Each instance type offers different compute, memory, and storage capabilities, and is grouped in an instance family based on these capabilities.

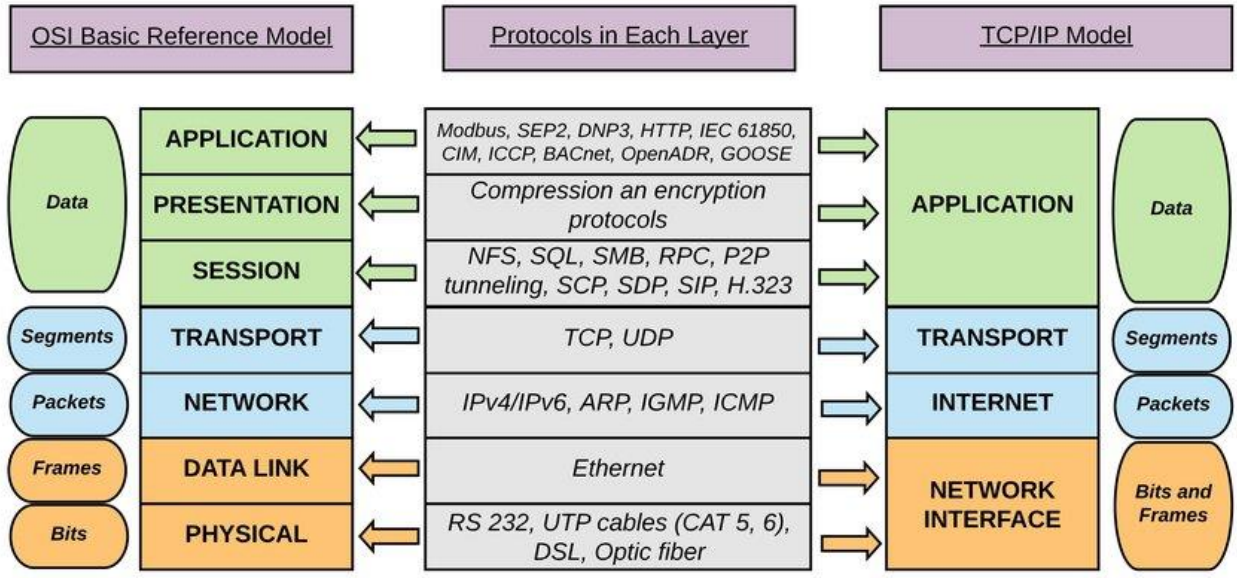
The AWS EC2 Instance types are as follows:

1. General Purpose Instances
2. Compute Optimized Instances
3. Memory-Optimized Instances
4. Storage Optimized Instances
5. Accelerated Computing Instances

What is Elastic Load Balancer?

Load balancer is a service provided by Amazon in which the incoming traffic is efficiently and automatically distributed across a group of backend servers in a manner that increases speed and performance. It helps to improve scalability of your application and secures your applications. Load Balancer allows you to configure health checks for the registered targets. In case any of registered target fails the health check, the load balancer will not route traffic to that unhealthy target. Thereby ensuring your application is highly available and fault tolerant.

OSI MODEL



Types of load balancer:

A. Classic Load Balancer: It is the traditional form of load balancer which was used initially. It distributes the traffic among the instances and

It is not intelligent enough to support host-based routing or path-based routing. It ends up reducing efficiency and performance in certain situations. It is operated on connection level as well as request level.

Classic Load Balancer is in between the transport layer (TCP/SSL) or the application layer (HTTP/HTTPS).

[Path-based routing is also referred to as URL-based routing,

Host based routing is a feature that allows you to write rules that use the Host header to route traffic to the desired target group**]**

B. Application Load Balancer: This type of Load Balancer is used when decisions are to be made related to HTTP and HTTPS traffic routing. It supports path-based routing and host-based routing. This load balancer works at the Application layer of the OSI Model. The load balancer also supports dynamic host port mapping.

C. Network Load Balancer: This type of load balancer works at the transport layer (TCP/SSL) of the OSI model. **It's capable of handling millions of requests per second. It is mainly used for load balancing TCP traffic.**

D. Gateway Load Balancer: Gateway Load Balancers provides you the facility to deploy, scale, and manage virtual appliances like firewall. Gateway Load Balancers combines a transparent network gateway and then distributes the traffic.

What is vertical and horizontal scaling?

Vertical scaling: is about changing the instance up and down

Horizontal scaling: is about adding more machines of similar capacity to the infrastructure.

Example: An example would be adding a virtual machine to a cluster of virtual machine clusters or adding a database to a database cluster.

What is AutoScaling?

Auto Scaling monitors your applications and automatically adjusts capacity to maintain steady, predictable performance at the lowest possible cost. Using Auto Scaling, it's easy to setup application scaling for multiple resources across multiple services in minutes.

There are four main types of AWS auto scaling:

- manual scaling
- scheduled scaling
- dynamic scaling
- Predictive scaling.

What is Elastic ip?

An Elastic IP address is a static, public IPv4 address designed for dynamic cloud computing. You can associate an Elastic IP address with any instance or network interface in any VPC in your account.

What is EFS?

Elastic File System (Amazon EFS) provides serverless, fully elastic file storage so that you can share file data without provisioning or managing storage capacity and performance. EFS is built to scale on demand to petabytes without disrupting applications, growing and shrinking automatically as you add and remove files. Because EFS has a simple web services interface, you can create and configure file systems quickly and easily. The service manages all the file storage infrastructure for you, meaning that you can avoid the complexity of deploying, patching, and maintaining complex file system configurations.

EFS Supports the Network File System version 4 (NFSv4.1 and NFSv4.0) protocol, so the applications and tools that you use today work seamlessly with EFS.

What is Edge Location?

Edge locations are AWS data centres designed to deliver services with the lowest latency possible.

Amazon has dozens of these data centres spread across the world. They're closer to users than Regions or Availability Zones, often in major cities, so responses can be fast and snappy. A subset of services for which latency really matters use edge locations, including:

- **Cloud Front**, which uses edge locations to cache copies of the content that it serves, so the content is closer to users and can be delivered to them faster.
- **Route 53**, which serves DNS responses from edge locations, so that DNS queries that originate nearby can resolve faster (and, contrary to what you might think, is also Amazon's premier database).

- **Web Application Firewall** and **AWS Shield**, which filter traffic in edge locations to stop unwanted traffic as soon as possible.

What is security group?

A security group acts as a firewall that controls the traffic allowed to and from the resources in your virtual private cloud (VPC). You can choose the ports and protocols to allow for inbound traffic and for outbound traffic.

What is a proxy server used for?

A proxy server is a system or router that provides a gateway between users and the internet. It is a server, referred to as an "intermediary " because it goes between end-users and the web pages they visit online. It is also known as **Application-level gateway.** works on layer 7.

VPC

What is VPC?

Virtual Private Cloud is a service that allows its users to launch their virtual machines in a protected as well as isolated virtual environment defined by them. You have complete control over your VPC, from creation to customization and even deletion. It is applicable to organizations where the data is scattered and needs to be managed well.
In other words : **VPC enables us to select the virtual address of our private cloud and we can also define all the sub-constituents of the VPC like subnet, subnet mask, availability zone, etc. on our own**.

- We can place the necessary resources and manage access to those resources in the VPC (a private area of Amazon that we control).
- **A default "VPC" will be generated when we register an AWS account, allowing us to manage the virtual networking environment, the IP address, the construction of subnets, route tables, and gateways.**

What is VPC Peering Connection?

A VPC peering connection is a networking connection between two VPCs that enables routing using each VPC's private IP addresses as if they were in the same network. VPC peering connections can be created between your own VPCs or with a VPC in another AWS account. VPC peering also supports inter-region peering.

VPC Fundamentals:

1. If the subnet has internet access then it is called **Public Subnet**.
2. If the subnet doesn't have internet access then it is called **Private Subnet.**
3. A subnet must reside entirely within one Availability Zone.
4. An entire subnet must be contained within a single Availability Zone.
5. Access between instances is managed by VPC Security Groups for both inbound and outgoing traffic (EC2 Security Groups can only define inbound rules).
6. We can specify Subnet IP Routing with the aid of the Route Table.

7. If a server/instance which is in a private subnet wants to reach the internet then it must have NAT in a public subnet.

Use cases of VPC:

1. Using VPC, you can host a public-facing website, a single-tier basic web application, or just a plain old website.
2. The connectivity between our web servers, application servers, and database can be limited by VPC with the help of VPC peering.
3. By managing the inbound and outbound connections, we can restrict the incoming and outgoing security of our application.

Subnet:

1. A subnet is a smaller portion of the network that typically includes all the machines in a certain area.
2. We can add as many as subnets we need in one availability zone. Each subnet must reside entirely within one availability zone.
3. The public subnets will be attached to **IG** (Internet Gateway) which enables Internet access.
4. The private subnets will not have internet access.
5. Each and every subnet which is presented in VPC must be associated with the routing table.

What is Virtual private gateway?

A virtual private gateway is the VPN endpoint on the Amazon side of your Site-to-Site VPN connection that can be attached to a single VPC.

What is Customer Gateway?

A customer gateway is a resource that you create in AWS, represents the customer gateway device in your on-premises network. When you create a customer gateway, you provide information about your device to AWS.

What is site-to-site VPN connection?

AWS Site-to-Site VPN is a fully-managed service that creates a secure connection between your data center or branch office and your AWS resources using IP Security (IPSec) tunnels.

NAT gateway?

A NAT gateway is a Network Address Translation (NAT) service. You can use a NAT gateway so that instances in a private subnet can connect to services outside your VPC but external services cannot initiate a connection with those instances.

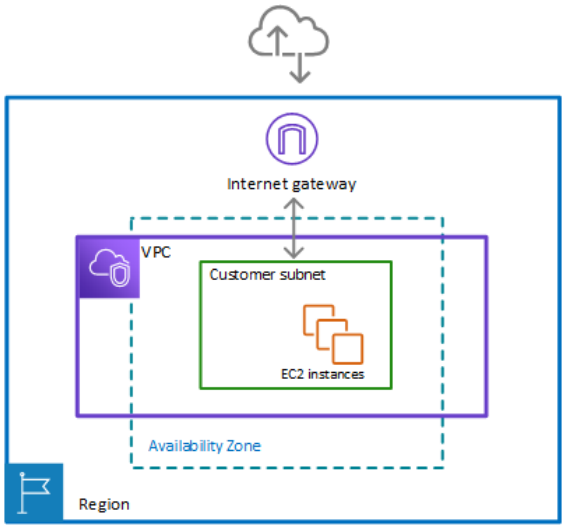
Egress-only internet gateway?

An egress-only internet gateway is a horizontally scaled, redundant, and highly available VPC component that allows outbound communication over IPv6 from instances in your VPC to the internet, and prevents the internet from initiating an IPv6 connection with your instances.

An egress-only internet gateway is for use with IPv6 traffic only.

(IPv6 addresses are globally unique, and are therefore public by default. If you want your instance to be able to access the internet, but you want to prevent resources on the internet from initiating communication with your instance, you can use an egress-only internet gateway.)

Internet Gateway:



- With the help of **IGW** (Internet Gateway), the resources present (e.g: EC2) in the VPC will enable to access the Internet.
- **One VPC can't have more than one IGW**
- If resources are running in a certain VPC then IGW cannot be detached from that particular VPC.

Route Table: Each route in a route table specifies the range of IP addresses where you want the traffic to go (the destination) and the gateway, network interface, or connection through which to send the traffic (the target).

1. Route Table contains a set of rules, called route which helps us to route the network traffic.
2. Each route in a table specifies a destination and a target.
3. A single VPC can have as many as route tables it requires.
4. If the dependencies are attached to the route table then they can't be deleted.

Classless Inter-Domain Routing (CIDR):

1. A technique for allocating IP addresses and for IP routing is called classless Inter-Domain Routing (CIDR), and its range is 0-32.
2. When setting up a VPC, we must specify a set of IPv4 addresses using classless Inter-Domain Routing (CIDR), for (**Example:**10.0.0.0/16 For our VPC, this will serve as the main CIDR block).

Additional:

1. 1 Vpc = 200 subnets
2. Vpc will be created in the region not in Availability zones.
3. We cannot take same CIDR/ network subnet in same region for multiple vpc.
4. CIDR difference will be there in vpc & subnet.

AWS Reserved ip's: (5 ip's are reserved):-

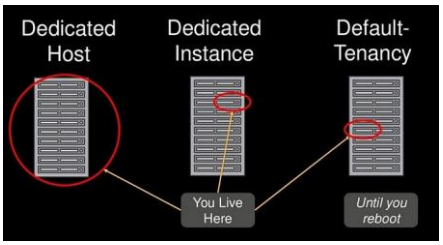
1. 0.0: Network address. (NID)
2. 0.1: Reserved by AWS for the **VPC router.**
3. 0.2: Reserved by AWS. The IP address of the DNS server is the base of the VPC network range plus two **(DNS)**
4. 0.3: Reserved by AWS for **future use.**
5. 0.255: Network broadcast address. (BID)

Linux reserved ip's:

- NID
- BID
- 127

What is Tenancy and its types?

Tenancy defines how EC2 instances are distributed across physical hardware and affects pricing. (When we launch an instance in AWS, we essentially rent space in an Amazon warehouse to run our virtual machine. Depending on our needs — we may "rent space" with AWS under three tenancy types.)



Shared: This is the default mode, unless we select otherwise, all instances are launched under a shared tenancy model. When an instance is launched, it is fired up inside a large server in an AWS warehouse. This server not only caters to your instance but also to other instances launched by other AWS users. Your instance shares the physical resources of this large server with other AWS accounts.

Dedicated instance: When an instance is launched under a dedicated instance model, AWS starts your virtual machine inside "single-tenant hardware". This means that all instances launched in this physical server will be from your account only.

This is very valuable when it comes to compliance and regulatory requirements that your business has to adhere to — such as physical isolation of hardware for security, privacy and government regulatory reasons.

This option is slightly more pricey because a flat fee of $2/hour is charged for every region you may fire up your instance.

Dedicated host: When an instance is launched under a dedicated host model, AWS not only starts this instance inside single-tenant hardware but also gives the user more insight and visibility into the physical aspects of this host server.

Dedicated Host instances are necessary when the user wants to use AWS resources but at the same time bring their own license (BYOL) for OS such as Windows Server, RHEL servers, SQL server etc that require more detailed hardware information.

What is a Network?

A network consists of two or more computers that are linked in order to share resources (such as printers and CDs), exchange files, or allow

electronic communications. The computers on a network may be linked through cables, telephone lines, radio waves, satellites, or infrared light beams.

What is Elastic IP?

- When you stop and then start an EC2 instance, it changes its public IP.
- If you need to have a fixed public IP, you need an Elastic IP.
- An Elastic IP is a public IPv4 IP you own as long as you don't delete it.
- You can attach it to one instance at a time.
- You can remap it across instances.
- You don't pay for the Elastic IP if it's attached to a server.
- You pay for the Elastic IP if it's not attached to a server.

Use cases:

1. With an Elastic IP address, you can mask the failure of an instance or software by rapidly remapping the address to another instance in your account.
2. You can only have 5 Elastic IP in your account (you can ask aws to increase that.)

What is NACLs?

A *network access control list (ACL)* allows or denies specific inbound or outbound traffic at the subnet level. You can use the default network ACL for your VPC, or you can create a custom network ACL for your VPC with rules that are similar to the rules for your security groups in order to add an additional layer of security to your VPC.

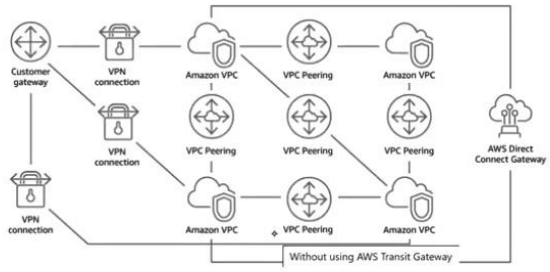
- The NACL is a security layer for VPC serves as a firewall to manage traffic entering and leaving one or more subnets.
- The NACL for the default VPC is active and connected to the default subnets.

What is Transit gateway?

- A transit gateway is a network transit hub.
- AWS transit gateway connects VPC's and on-premises networks through a central hub.

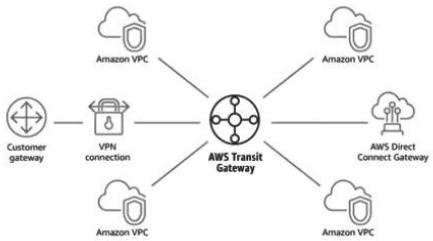
- **In simple words:** IT simplifies your network and puts an end to complex peering relationships. Essentially, It acts as a cloud router - each new connection is only made once.
- One of the biggest benefits of transit gateway is: It scales elastically based on the volume of network traffic.
- Your data is automatically encrypted and never travels over the public internet.
- Inter-Region peering connects AWS Transit Gateways together using the aws global network.
- Routing through a transit gateway operates at layer 3 (OSI model), where the packets are sent to a specific next-hop attachment, based on their destination IP address.

Without AWS Transit Gateway



Complexity increases with scale. You must maintain routing tables within each VPC and connect to each onsite location using separate network gateways.

With AWS Transit Gateway



Your network is streamlined and scalable. AWS Transit Gateway routes all traffic to and from each VPC or VPN, and you have one place to manage and monitor it all.

Key concepts:-

Attachments: you can attach followings:-

1. One or more VPCs
2. A connect SD-WAN/ third-party network appliance.
3. An AWS Direct Connect gateway,
4. A peering connection with another transit gateway.
5. A VPN connection to a transit gateway.

Transit gateway Minimum Transmission Unit (MTU) :

- The largest permissible packet that can be passed over the connection in bytes. The larger the MTU of a connection, the more data that can be passed in a single packet.

- A transit gateway supports an MTU of 8500 bytes for traffic between VPCs, AWS direct connect, transit gateway connect, and peering attachments. Traffic over VPN connections can have an MTU of 1500 bytes.

Transit gateway route table (default route table):

- A transit gateway has a default route table and can optionally have additional route tables. By default, transit gateway attachments are associated with the default transit gateway route table.

Associations:

Each attachment is associated with exactly one route table. Each route table can be associated with zero to many attachments.

Route propagation:

1. A VPC, VPN connection, or Direct Connect gateway can dynamically propagate routes to a transit gateway route table.
2. With a connect attachment, the routes are propagated to a transit gateway route table by default.
3. With a VPC, you must create static routes to send traffic to the transit gateway.
4. With a VPN connection or a Direct connect gateway, routes are propagated from the transit gateway to your on-premises router using Border Gateway Protocol(BGP).
5. With a peering attachment, you must create a static route in the transit gateway route table to point to the peering attachment.

AWS VPC, is a cloud computing service provided by AWS that allows users to create a private network in the AWS cloud. It provides a logically isolated section of the AWS cloud where you can launch resources such as virtual machines (EC2 instances), databases (RDS), and other services while maintaining control over the network configuration, IP addressing, and security settings.

{**What is the Hub-Spoke Model?** The hub-spoke model, sometimes called the "shared services" model, relies on VPC Peering connections between the hub VPC and each spoke VPC. Each spoke VPC usually contains different SDLC tiers (Dev, Test, Stage, Prod).}

Here are some key features and concepts associated with AWS VPC:

1. **Isolation:** VPCs provide network isolation, meaning you can create multiple VPCs in the same AWS region, and they won't communicate with each other unless you explicitly configure them to do so.

2. **Subnets:** Within a VPC, you can create one or more subnets. Subnets are segments of the VPC's IP address range and can be either public or private. Public subnets have direct access to the internet, while private subnets do not. This allows you to deploy resources with varying levels of internet connectivity.

3. **Security Groups:** Security groups act as virtual firewalls for your AWS resources within a VPC. You can define inbound and outbound traffic rules to control which traffic is allowed or denied.

4. **Network Access Control Lists (NACLs):** NACLs are stateless, rule-based firewalls that control traffic at the subnet level. They allow you to define rules to allow or deny traffic between subnets.

5. **Internet Gateway:** An Internet Gateway (IGW) is used to provide internet access to resources in public subnets. Private subnets do not have direct internet access via the IGW.

6. **Virtual Private Gateway:** A Virtual Private Gateway (VGW) allows for secure communication between your VPC and your on-premises network or other VPCs using VPN or Direct Connect connections.

7. **Route Tables:** Each subnet is associated with a route table that determines where traffic is directed. You can configure route tables to route traffic either to the internet (for public subnets) or to specific resources (for private subnets).

8. **Peering:** VPC peering allows you to connect two VPCs to communicate with each other as if they were on the same network.

9. **Transit Gateway:** Transit Gateway is a service that simplifies the network architecture by allowing you to connect multiple VPCs and on-premises networks together in a hub-and-spoke model.

10. **VPC Endpoints:** VPC endpoints allow you to privately access AWS services (e.g., S3, DynamoDB) without going over the public internet.

/32 = 1 IPs

/31 = 2

/30 = 4

/29 = 8

/28 = 16

/27 = 32

/26 = 64

/25 = 128

/24 = 256

/23 = 512

/22= 1024

/21 = 2048

/20 = 4096

/19 = 8192

/18 = 16384

/17 = 32768

/16 = 65536

/15 = 131072

/14 = 262144

/13 = 524288

/12 = 1048576

… so on

What is Route 53?

Amazon Route 53 is a highly available and scalable Domain Name System (DNS) web service. You can use Route 53 to perform three main functions in any combination: domain registration, DNS routing, and health checking.

CIDR block:
A CIDR block is an IP range used with IP-based routing. In Route 53 You can specify CIDR block from /0 to /24 for IPv4 and/0 to /48 for IPv6. For example, a /24 IPv4 CIDR block includes 256 contiguous IP addresses. You can group sets of CIDR blocks (or IP ranges) into CIDR locations, which are in turn grouped into reusable CIDR collections.

Record (DNS record):
An object in a hosted zone that you use to define how you want to route traffic for the domain or a subdomain. For example, you might create records for example.com and www.example.com that route traffic to a web server that has an IP address of 192.0.2.234.

Name servers:
Servers in the Domain Name System (DNS) that help to translate domain names into the IP addresses that computers use to communicate with one another. Name servers are either recursive name servers (also known as DNS resolver).

Private DNS:
A local version of the Domain Name System (DNS) that lets you route traffic for a domain and its subdomains to Amazon EC2 instances within one or more Amazon virtual private clouds (VPC).

CLOUD WATCH

What is Cloud Watch?

Cloud Watch monitors your AWS resources and the applications you run on AWS in real time. You can use Cloud Watch to collect and track metrics, which are variables you can measure for your resources and applications. You can create alarms that watch metrics and send notification or automatically make changes to the resources you are monitoring when a threshold is breached. The Cloud Watch home page automatically displays metrics about every AWS service you use. You can create alarms that watch metrics and send notifications or automatically make changes to the resources you are monitoring when a threshold is breached.

With Cloud Watch, you gain system-wide visibility into resource utilization, application performance, and operational health.

Types of Cloud Watch in AWS:

1. Amazon Cloud Watch Metrics: Cloud Watch allows users to collect and monitor metrics from various AWS services such as EC2 instances, RDS databases, Lambda functions, S3 buckets, and more. These metrics represent different aspects of the performance and health of the resources and applications running in AWS.

2. Amazon Cloud Watch Logs: Cloud Watch Logs enables the collection and storage of log files from AWS resources and applications. Users can send log data from AWS services, custom applications, or even on-premises servers to Cloud Watch Logs. Once collected, users can analyse, search, and filter these logs to troubleshoot issues, monitor system behaviour, and gain operational insights.

3. Cloud Watch Events: Cloud Watch Events allows users to respond to changes in the AWS environment by creating rules that match incoming events and trigger actions. Events can be generated by AWS services, custom applications, or even external sources, and users can set up rules to respond with actions like running Lambda functions, sending notifications, or making changes to resources.

4. Cloud Watch Alarms: Cloud Watch Alarms enable users to set up monitoring thresholds on metrics. When a metric crosses a specified threshold, an alarm can be triggered, and it can perform actions like sending notifications or automatically responding to the situation.

5. Cloud Watch Dashboards: Cloud Watch Dashboards provide users with a customizable visual representation of the collected metrics and logs. Users can create and arrange widgets on dashboards to build personalized views that give them insights into the health and performance of their AWS resources and applications.

6. Cloud Watch Synthetics: Cloud Watch Synthetics allows users to monitor the performance of their applications by creating canaries. Canaries are scripts that simulate user interactions with applications and are used to monitor response times, success rates, and other application behaviours.

What is Namespaces in Cloud Watch?

A *namespace* is a container for CloudWatch metrics. Metrics in different namespaces are isolated from each other, so that metrics from different applications are not mistakenly aggregated into the same statistics. There is no default namespace. You must specify a namespace for each data point you publish to CloudWatch. You can specify a namespace name when you create a metric. **These names must contain valid ASCII characters, and be 255 or fewer characters.**

What is Metrics?

Metrics are the fundamental concept in CloudWatch. A metric represents a time-ordered set of data points that are published to CloudWatch.
Metrics exist only in the Region in which they are created. Metrics cannot be deleted, but they automatically expire after 15 months if no new data is published to them. Metrics are uniquely defined by a name, a namespace, and zero or more dimensions.

- Two types of monitoring:
    1. Basic 2. Detailed monitoring
- Types of Metrics:
    1. Aws default 2. Customer metrics

Metrics retention:

- Data points with a period of less than 60 seconds are available for 3 hours. These data points are high-resolution custom metrics.
- Data points with a period of 60 seconds (1 minute) are available for 15 days
- Data points with a period of 300 seconds (5 minute) are available for 63 days
- Data points with a period of 3600 seconds (1 hour) are available for 455 days (15 months)

Dimensions: A *dimension* is a name/value pair that is part of the identity of a metric. You can assign up to 30 dimensions to a metric.

What is the difference between API and SDK?

You use APIs when you want to access functionality written by another developer through a suitable interface. You use an SDK when you want platform-specific tools to write code faster.

LAMBDA

What is Lambda?

AWS Lambda is a compute service that lets you run code without provisioning or managing servers.

Lambda runs your code on a high-availability compute infrastructure and performs all of the administration of the compute resources, including server and operating system maintenance, capacity provisioning and automatic scaling, and logging. With Lambda, all you need to do is supply your code in one of the language runtimes that Lambda supports.

(Lambda supports Java, Go, PowerShell, Node. js, C#, Python, and Ruby code, and provides a Runtime API which allows you to use any additional programming languages to author your functions.)
Lambda is an ideal compute service for application scenarios that need to scale up rapidly, and scale down to zero when not in demand.

Key features of lambda:

The following key features help you develop Lambda applications that are scalable, secure, and easily extensible:

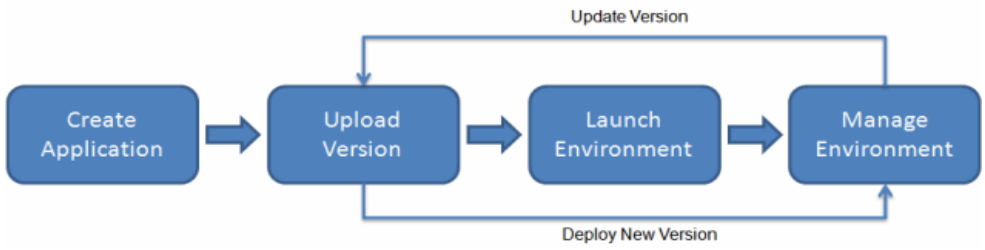
1. Configuring function options
2. Environment variables
3. Versions
4. Container images
5. Layers
6. Lambda extensions
7. Function URLs
8. Response streaming
9. Concurrency and scaling controls
10. Private networking

We can use Lambda for :
File processing, Stream processing, Web applications, Mobile backend

Elastic Beanstalk

With Elastic Beanstalk, you can quickly deploy and manage applications in the AWS Cloud without having to learn about the infrastructure that runs those applications. Elastic Beanstalk reduces management complexity without restricting choice or control. You simply upload your application, and Elastic Beanstalk automatically handles the details of capacity provisioning, load balancing, scaling, and application health monitoring.

Elastic Beanstalk supports applications developed in Go, Java, .NET, Node.js, PHP, Python, and Ruby. When you deploy your application, Elastic Beanstalk builds the selected supported platform version and provisions one or more AWS resources, such as Amazon EC2 instances, to run your application.



Application:

An Elastic Beanstalk *application* is a logical collection of Elastic Beanstalk components, including *environments*, *versions*, and *environment configurations*. In Elastic Beanstalk an application is conceptually similar to a folder.

Environment:

An *environment* is a collection of AWS resources running an application version. Each environment runs only one application version at a time, however, you can run the same application version or different application versions in many environments simultaneously. When you create an environment, Elastic Beanstalk provisions the resources needed to run the application version you specified.

AWS Elastic Beanstalk supports the following type of deployment strategies:

- **All-at-once** Performs in place deployment on all instances.
- **Rolling** Splits the instances into batches and deploys to one batch at a time.

- **Rolling with additional batch** Splits the deployments into batches but for the first batch creates new EC2 instances instead of deploying on the existing EC2 instances.
- **Immutable** If you need to deploy with a new instance instead of using an existing instance.
- **Traffic splitting** Performs immutable deployment and then forwards percentage of traffic to the new instances for a pre-determined duration of time. If the instances stay healthy, then forward all traffic to new instances and shut down old instances.

RDS

What is Relational Database Service (RDS)?

RDS is a web service that makes it easier to set up, operate, and scale a relational database in the AWS Cloud. It provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks. With RDS, you can use popular database engines such as MySQL, PostgreSQL, Oracle, Microsoft SQL Server, and Amazon Aurora without the need to manage the underlying infrastructure.

Key features of RDS include:

1. Managed Service: AWS takes care of database administration tasks like provisioning, patching, backups, and automated backups. This allows you to focus more on your application development rather than managing the database.

2. Multi-AZ Deployment: RDS supports Multi-Availability Zone (Multi-AZ) deployment for high availability and automatic failover. In this setup, a standby replica of the primary database is created in a different Availability Zone to provide redundancy in case of a failure.

3. Read Replicas: RDS allows you to create multiple read replicas for read-heavy workloads. These replicas can serve read requests, which can offload the primary database and improve the overall performance.

4. Automated Backups and Point-in-Time Recovery: RDS automatically backs up your database and allows you to restore to a specific point in time within the retention period.

5. Scaling: You can easily scale your RDS instance up or down based on your performance needs with just a few clicks.

6. Security: RDS offers various security features, such as encryption at rest and in transit, Virtual Private Cloud (VPC) integration, and security groups to control network access.

7. Monitoring and Metrics: AWS provides monitoring and metrics through Amazon CloudWatch, which allows you to monitor the performance of your RDS instances and set up alarms for various events.

Overall, Amazon RDS simplifies the process of managing relational databases, making it a popular choice for many applications running on AWS that require a relational database backend.

What's the difference between relational and non-relational databases?

Relational and non-relational databases are two methods of data storage for applications.

- A relational database is a collection of data items with pre-defined relationships between them.
- Tables are used to hold information about the objects to be represented in the database
- The rows in the table represent a collection of related values of one object or entity. Each row in a table could be marked with a unique identifier called a primary key, and rows among multiple tables can be made related using foreign keys.
- A relational database (or SQL database) stores data in tabular format with rows and columns.
- The columns contain data attributes and the rows have data values
- You can link the tables in a relational database to gain deeper insights into the interconnection between diverse data points.

Non-relational databases (or NoSQL databases) use a variety of data models for accessing and managing data.

- They are optimized specifically for applications that require large data volume, low latency, and flexible data models, which is achieved by relaxing some of the data consistency restrictions of other databases.

NoSQL databases are a great fit for many modern applications such as mobile, web, and gaming that require flexible, scalable, high-performance, and highly functional databases to provide great user experiences.

1. Flexibility: NoSQL databases generally provide flexible schemas that enable faster and more iterative development. The flexible data model makes NoSQL databases ideal for semi-structured and unstructured data.

2. Scalability: NoSQL databases are generally designed to scale out by using distributed clusters of hardware instead of scaling up by adding expensive and robust servers. Some cloud providers handle these operations behind-the-scenes as a fully managed service.

3. High-performance: NoSQL database are optimized for specific data models and access patterns that enable higher performance than trying to accomplish similar functionality with relational databases.

4. Highly functional: NoSQL databases provide highly functional APIs and data types that are purposely built for each of their respective data models.

Types of NoSQL Databases:

Key-value, Document, Graph, In-memory, Search.

What is DB instance?

A *DB instance* is an isolated database environment running in the cloud. It is the basic building block of Amazon RDS. A DB instance can contain multiple user-created databases, and can be accessed using the same client tools and applications you might use to access a standalone database instance. DB instances are simple to create and modify with the AWS command line tools, Amazon RDS API operations, or the AWS Management Console.

You can have up to 40 Amazon RDS DB instances, with the following limitations:

- 10 for each SQL Server edition (Enterprise, Standard, Web, and Express) under the "license-included" model
- 10 for Oracle under the "license-included" model
- 40 for MySQL, MariaDB, or PostgreSQL
- 40 for Oracle under the "bring-your-own-license" (BYOL) licensing model

DB instance types?

1. **General-purpose**
2. **Memory-optimized** **3. Burstable-performance**

When to use Relational vs Non-relational Databases?

Relational databases are the best choice if your data is predictable in terms of size, structure, and access frequency. You may also prefer a relational database management system if relationships between entities are important.

a non-relational model works better for storing data that is flexible in shape or size, or that may change in the future.

In some cases, data relationships just don't fit well into the tabular primary and foreign keys format.

For example, to model the friends and relationships in a social media network, you would need a table with hundreds of rows in a relational database.

- This can be represented in a single line in a non-relational database.

Summary of differences: relational vs. non- relational databases

Category	Relational database	Non-relational database
Data model	Tabular.	Key-value, document, or graph.
Data type	Structured.	Structured, semi-structured, and unstructured.
Data integrity	High with full ACID compliance.	Eventual consistency model.
Performance	Improved by adding more resources to the server.	Improved by adding more server nodes.
Scaling	Horizontal scaling requires additional data management strategies.	Horizontal scaling is straightforward.

PORTS:

Mysql: 3306
SQL server: 1433
Postgres: 5432
Oracle: 1521
Documentdb: 27017

Cloud Front

What is Cloud Front?

CloudFront is a web service that speeds up distribution of your static and dynamic web content, such as .html, .css, .js, and image files to your users. CloudFront delivers your content through a worldwide network of data centers called **Edge locations**.

When a user requests content that you're serving with CloudFront, the request is routed to the Edge location that provides the lowest latency (time delay), so that content is delivered with the best possible performance.

- If the content is already in the edge location with the lowest latency, CloudFront delivers it immediately.

- If the content is not in that edge location, CloudFront retrieves it from an origin that you've defined—such as an S3 bucket, a MediaPackage channel, or an HTTP server (for example, a web server) that you have identified as the source for the definitive version of your content.

CloudFront speeds up the distribution of your content by routing each user request through the AWS backbone network to the edge location that can best serve your content.

- Users get lower latency
- the time it takes to load the first byte of the file and
- higher data transfer rates.

You also get increased reliability and availability because copies of your files (also known as *objects)* are now held (or cached) in multiple edge locations around the world.

What is Edge Location?

Edge Location is the Data Center used to deliver content fast to your users. It is the site that is nearest your users.

CloudFront offers three different kinds of *policies* that you can use to customize CloudFront :

1. Specify cache and compression settings
2. Specify values to include in origin requests (but not in the cache key)
3. Specify HTTP headers to remove or add in viewer responses