**What is cloud?**

The cloud refers to a network of servers that are accessed over the internet to store, manage, and process data and applications, rather than on a local server or personal computer.

**Difference between public cloud and private cloud?**

Public Cloud: Services are provided over the internet by third-party providers like Amazon Web Services (AWS), Microsoft Azure, or Google Cloud Platform. Resources are shared among multiple users, and users pay only for the resources they consume.

Private Cloud: Services are maintained on a private network, either on-premises or hosted by a third-party provider. Resources are dedicated to a single organization, providing greater control and customization but often with higher costs.

**Why is public cloud so popular?**

Public clouds offer scalability, flexibility, and cost-effectiveness. They eliminate the need for organizations to invest in and maintain their own infrastructure, allowing them to quickly scale resources up or down as needed, and only pay for what they use. Additionally, public clouds provide access to advanced technologies and global infrastructure.

**How is AWS better than others?**

AWS offers a wide range of services, global infrastructure, and a strong ecosystem of partners and developers. It provides highly reliable, scalable, and cost-effective solutions for businesses of all sizes. Additionally, AWS continues to innovate with new services and features, maintaining its position as a leader in the cloud computing market.

**What is AWS?**

AWS stands for Amazon Web Services, which is a cloud computing platform that provides a wide range of cloud-based services to businesses and individuals also works with a pay as you go pricing model which means you only pay for how much of the services

**What Is An Instance?**

An EC2 instance is a virtual server that runs on the AWS cloud infrastructure. These instances are available in various configurations, with different combinations of CPU, memory, storage, and networking capacity.

**What is an instance type in AWS?**

**General Purpose Instances** **(e.g., t3, m5):** Suitable for a wide range of applications including web servers, development environments, and small to medium databases.

**Compute Optimized Instances (e.g., c5):** Designed for compute-bound applications that require high performance processing, such as batch processing, high-performance computing (HPC), and scientific modeling.

**Memory Optimized Instances (e.g., r5):** Ideal for memory-intensive applications such as in-memory databases, real-time big data analytics, and high-performance databases.

**Storage Optimized Instances (e.g., i3):** Optimized for applications that require high I/O performance and low-latency storage, such as NoSQL databases, data warehousing, and Elasticsearch.

Accelerated Computing Instances (e.g., p3, g4): Equipped with specialized hardware accelerators like GPUs and FPGAs, suitable for tasks such as machine learning, graphics rendering, and video encoding.

**Key Pair:**

An EC2 key pair consists of a public key and a private key. You use the public key to encrypt data and the private key to decrypt it. EC2 key pairs are used for secure SSH access to EC2 instances.

**What is the relation between the Availability Zone and Region?**

An AWS Availability Zone is a physical location where an Amazon data center is located. On the other hand, an AWS Region is a collection or group of Availability Zones or Data Centers.

**i) What Is EC2?**

Amazon Elastic Compute Cloud (EC2) is a web service provided by AWS that allows users to rent virtual servers (instances) on which they can run their own applications. EC2 instances provide scalable computing capacity in the cloud, making it easy to scale up or down as needed.

With EC2, users can launch instances of various types and sizes depending on their specific needs. EC2 instances can run a variety of operating systems, including Linux, Windows, and Unix, and can be customized to meet specific performance and security requirements.

EC2 instances are billed on a per-hour or per-second basis, depending on the pricing model selected by the user. EC2 offers several pricing options, including On-Demand instances, Reserved instances, and Spot instances.

## iii) EC2 Pricing Options

AWS offers several pricing models for its services, each designed to meet different needs and usage patterns. The following are some of the pricing models offered by AWS:

**On-Demand:** With this model, users pay for compute capacity by the hour or second, without any long-term commitments or upfront payments. This pricing model is best suited for applications with unpredictable workloads or short-term usage patterns.

**Reserved Instances:** With this model, users can make a one-time, upfront payment for a specific instance type and receive a discount on the hourly charge for that instance. This pricing model is best suited for applications with steady-state usage patterns or applications that require reserved capacity.

You commit to using specific instance types in a specific region for a one- or three-year term, which offers a significant discount compared to On-Demand pricing. It is suitable for predictable workloads.

**Spot Instances:** With this model, users can bid for unused Amazon EC2 capacity and receive a discount on the hourly charge for that capacity. This pricing model is best suited for applications with flexible start and end times or applications that can handle interruptions in their compute capacity.

**Dedicated Hosts:** With this model, users can rent an entire physical server that is dedicated to their use. This pricing model is best suited for applications with strict regulatory or compliance requirements.

The best pricing model for an application or workload on AWS depends on its usage patterns, workload characteristics, and budget. For example, if an application has unpredictable usage patterns, the On-Demand pricing model may be the best option, whereas if an application has a predictable usage pattern, the Reserved Instances or Savings Plans pricing models may be more cost-effective. It's important to carefully analyze your application's usage patterns and requirements to choose the best pricing model for your needs.

**iv) AMI [Amazon Machine Image]**

* An AMI stands for **Amazon Machine Images**.
* An AMI is a virtual image used to create a virtual machine within an EC2 instance.
* You can also create multiple instances using single AMI when you need instances with the same configuration.
* You can also create multiple instances using different AMI when you need instances with a different configuration.

It also provides a template for the root volume of an instance

**v) Security Groups:**

In Amazon Web Services (AWS), a security group is a virtual firewall that controls the inbound and outbound traffic for one or more Amazon Elastic Compute Cloud (EC2) instances. A security group acts as a set of rules that define which traffic is allowed to access the instances based on protocols, ports, and source or destination IP addresses

**What is a Static IP?**

A **Static IP address** is a fixed, unchanging address assigned to a device on a network. Unlike a dynamic IP address, which may change over time, a static IP remains constant.

**vi) Elastic Ip Address:**

In Amazon Web Services (AWS), an Elastic IP address (EIP) is a static, public IPv4 address that you can allocate to your AWS account and assign to your EC2 instances or other AWS resources. Elastic IP addresses are designed to provide a way to mask the failure of an instance or software by quickly remapping the address to another instance in your account.

The main difference between IPv4 and IPv6 is the address size of IP addresses. The IPv4 is a 32-bit address and it is in numeric values, whereas IPv6 is a 128-bit hexadecimal address. And it is in alpha numeric values. IPv6 provides a large address space, and it contains a simple header as compared to IPv4.

**\*) What is a load balancer in AWS?**

Load balancing is the method of distributing network traffic equally across a pool of resources that support an application

(OR)

Load balancing in AWS refers to the practice of distributing incoming network traffic across multiple servers or instances to improve application availability, scalability, and reliability.

AWS offers several load balancing options that can be used to distribute traffic across EC2 instances or containers, including:

**Elastic Load Balancing (ELB)** - A managed load balancer service that automatically scales up or down based on traffic demands.

**Application Load Balancer (ALB)** - A layer 7 load balancer that distributes incoming traffic based on application-level content, such as URL paths and HTTP headers.

**Network Load Balancer (NLB)** - A layer 4 load balancer that distributes incoming traffic based on transport-level protocols, such as TCP and UDP.

**2. Network Load Balancer (NLB):**

Think of a Network Load Balancer (NLB) as a dispatcher at a train station. Its job is to make sure trains (network traffic) go to the right platforms (target groups) based on certain characteristics like train type (protocol) or ticket (port number).

Example: At a train station, you have different types of trains (e.g., high-speed, local) arriving on various tracks. The NLB, like a dispatcher, ensures passengers (network requests) are directed to the right track (target group) for their specific type of train.

High-Speed Train (Port 80) Platform: Passengers with tickets for high-speed trains (HTTP requests) are sent to the high-speed train platform (target group) to catch their train.

Local Train (Port 443) Platform: Passengers with tickets for local trains (HTTPS requests) are directed to the local train platform (target group) for their ride.

So, NLB routes traffic based on characteristics like the type of train (protocol) and the ticket (port) to the appropriate platform (target group).

**3. Application Load Balancer (ALB):**

Now, think of an Application Load Balancer (ALB) as a restaurant host who manages seating in a restaurant. The host directs guests (web requests) to different tables (target groups) based on the type of food they want (content-based routing).

Example: You're running a restaurant with different dining areas (target groups), such as Italian, Mexican, and Japanese. The ALB host ensures that guests (web requests) are seated at the right table (target group) based on their food preference (content-based routing).

Italian Section: Guests who want Italian food (URLs containing "/italian") are seated in the Italian section (Italian target group).

Mexican Section: Guests craving Mexican food (URLs with "/mexican") are directed to the Mexican section (Mexican target group).

Japanese Section: Those in the mood for Japanese dishes (URLs with "/japanese") are placed in the Japanese section (Japanese target group).

In this way, ALB routes traffic based on the content of the request to the appropriate target group, just like a restaurant host seating guests based on their food preference.

To summarize, NLB directs traffic based on characteristics like the type of train (protocol) and port (ticket), while ALB routes traffic based on the content of the request, like seating guests in a restaurant based on their food preference.

**viii) Traget Groups**

When you create a load balancer, you must specify one or more target groups to receive traffic from the load balancer. The target group determines the routing algorithm that the load balancer uses to distribute traffic among the registered resources.

**ix) What is Launch Template?**

A launch template is similar to a [launch configuration](https://docs.aws.amazon.com/autoscaling/ec2/userguide/launch-configurations.html), in that it specifies instance configuration information. It includes the ID of the Amazon Machine Image (AMI), the instance type, a key pair, security groups, and other parameters used to launch EC2 instances. However, defining a launch template instead of a launch configuration allows you to have multiple versions of a launch template.

**What is Launch Configuration?**

A launch configuration is a set of instructions that specifies how to launch and configure instances in a particular AWS Auto Scaling group. It defines the AMI (Amazon Machine Image) to use, the instance type, and other launch parameters such as the security groups, key pairs, and block device mappings.

Once you create a launch configuration, you can associate it with an Auto Scaling group. This group can then automatically launch instances based on the specified launch configuration in response to changes in demand, such as increased traffic or higher usage.

**what is the use of launch configuration in aws?**

A launch configuration is a template that an EC2 Auto Scaling group uses to launch EC2 instances.

**What is the purpose of launch template in AWS?**

A launch template is similar to a launch configuration, in that it specifies instance configuration information. It includes the ID of the Amazon Machine Image (AMI), the instance type, a key pair, security groups, and other parameters used to launch EC2 instances.

**What is the difference between launch template and launch configuration in AWS**

launch configurations are used with Auto Scaling Groups. While launch templates are used when you launch an instance using the aws EC2 console, an AWS SDK, or a command line tool. Launch templates enable you to store the parameters (AMI, instance type, security groups, and key pairs etc.)

**What is Auto Scaling in AWS EC2?**

Amazon EC2 Auto Scaling helps you maintain application availability and lets you automatically add or remove EC2 instances using scaling policies that you define.

**What is AWS Lambda?**

**Ans.**It helps us in running our code without managing the servers. It runs our code in an environment that has high availability of servers, storage capacity, automatic scaling and other computing resources. It automatically scales the resources with the help of Lambda functions.

**What is AWS Elastic Beanstalk?**

**Ans.**When running our application on Elastic Beanstalk, we do not need to configure the infrastructure on our own. It automatically manages services like, load balancing, storage capacity, auto-scaling, etc.

**2) Storage :**

# **i) EBS[ Elastic Block Store**]

# **What is EBS?**

* EBS stands for **Elastic Block Store**.
* EC2 is a virtual server in a cloud while EBS is a virtual disk in a cloud.
* Amazon EBS allows you to create storage volumes and attach them to the EC2 instances.
* Once the storage volume is created, you can create a file system on the top of these volumes, and then you can run a database, store the files, applications or you can even use them as a block device in some other way.
* Amazon EBS volumes are placed in a specific availability zone, and they are automatically replicated to protect you from the failure of a single component.

**ii) Amazon S3 (Simple Storage Service)**

**What is AWS S3 Used for?**

S3 provides a simple web services interface that can be used to store and retrieve data

* S3 is a safe place to store the files.
* It is Object-based storage, i.e., you can store the images, word files, pdf files, etc.
* The files which are stored in S3 can be from 0 Bytes to 5 TB.
* It has unlimited storage means that you can store the data as much you want.
* Files are stored in Bucket. A bucket is like a folder available in S3 that stores the files.
* S3 is a universal namespace, i.e., the names must be unique globally. Bucket contains a DNS address. Therefore, the bucket must contain a unique name to generate a unique DNS address.

**S3 Storage Classes:**

**1) S3 Standard :** This is the default storage class and provides high durability, availability, and performance for frequently accessed data. It is suitable for storing data that requires real-time access and low latency. [ **In case if there is some data that need to be fetch right away**]

**2) S3 Standard-Infrequent Access (S3 Standard-IA):** This storage class is designed for data that is accessed less frequently but requires rapid access when needed. It provides a lower storage cost than S3 Standard but has a retrieval fee.

**3) S3 Glacier:** This storage class is designed for long-term data archival and provides the lowest storage cost among all S3 storage classes. It has a minimum storage duration of 90 days and a retrieval fee.

**4) S3 Intelligent-Tiering:** This storage class is designed to optimize costs by automatically moving data between two access tiers based on changing access patterns. It provides the same level of durability and availability as S3 Standard.

**5) S3 One Zone-Infrequent Access (S3 One Zone-IA**): This storage class is designed for infrequently accessed data that can be recreated if lost. It offers lower storage prices than S3 Standard-IA, but stores data in a single availability zone, making it less resilient to zone-level failures.

**6) S3 Glacier Deep Archive:** This storage class is designed for archiving data that is rarely accessed and has a retrieval time of 12 hours or more. It offers the lowest storage prices of all the storage classes in Amazon S3.

# **Versioning:**

**Versioning** is a means of keeping the multiple forms of an object in the same S3 bucket. Versioning can be used to retrieve, preserve and restore every version of an object in S3 bucket.

For example, bucket consists of two objects with the same key but with different version ID's such as photo.jpg (version ID is 11) and photo.jpg (version ID is 12).

**iii) Amazon S3 Glacier:**

Amazon S3 Glacier (S3 Glacier) is a secure and durable service for low-cost data archiving and long-term backup. With S3 Glacier, you can store your data cost effectively for months, years, or even decades.

**iv) What is Amazon Elastic File System?**

Amazon Elastic File System (Amazon EFS) provides a simple serverless, fully elastic file storage so that you can share file data without provisioning or managing storage capacity and performance.

Imagine you have a large collection of documents, pictures, and other digital files that you want to store and share with multiple people or programs. EFS provides a convenient and scalable way to do that.

**When to use AWS ELASTIC FILE SYSTEM?**

Amazon supports file object and block level storage

**1) for file storage we can use EFS**

**2) for object storage we can use AWS S3**

**3) for block level storage we can use AWS EBS**

Use S3 for storing objects like files, backups, and media.

Use EBS for attaching storage to individual EC2 instances.

Use EFS for shared file storage accessed by multiple instances.

**v) Storage Gateway:**

An AWS Storage Gateway is like a bridge between your on-premises (local) data and the cloud. It makes it easier to store, access, and manage your data in Amazon Web Services (AWS) from your own servers or data centers.

Imagine it as a translator or connector that allows your local data to communicate with AWS services like Amazon S3, EBS, or Glacier**.**

**3) Security Services :**

**i) Identity Access Management:[Authenticating Authorization Management]**

In AWS (Amazon Web Services), IAM stands for Identity and Access Management. It is a service that helps you manage access to AWS resources by creating and managing AWS users and groups, and by setting permissions for them.

With IAM, you can create individual users, each with their own set of credentials (such as a username and password), and define what actions they can perform on AWS resources. You can also organize these users into groups, and apply permissions to the groups instead of individual users.

# **What is a Role?**

In AWS IAM, a role is a set of permissions that define what actions an entity can perform on AWS resources. A role can be assigned to an AWS service or an AWS user, allowing them to assume the permissions of the role temporarily, without the need for the user or service to have their own credentials.

For example, if you have an EC2 instance that needs to access an S3 bucket, instead of creating an AWS user with the necessary permissions, you can create a role that grants access to the S3 bucket, and assign that role to the EC2 instance. When the instance needs to access the bucket, it assumes the role and is granted the necessary permissions.

**What is the use of CloudWatch and CloudTrail?**

CloudWatch is a monitoring service for AWS resources and applications. CloudTrail is a web service that records API activity in your AWS account. CloudWatch monitors applications and infrastructure performance in the AWS environment. CloudTrail monitors actions in the AWS environment.

**Cloud Front:**

Amazon CloudFront is a content delivery network (CDN) service that helps deliver content to users quickly, securely, and efficiently:

**How it works**

CloudFront uses a worldwide network of data centers called edge locations to deliver content. When a user requests content, CloudFront routes the request to the edge location with the lowest latency. If the content is already in that edge location, CloudFront delivers it immediately.

**What it can deliver**

CloudFront can deliver static and dynamic content, such as:

* HTML
* CSS
* JS
* Image files
* Videos
* Applications
* APIs

**4) Networks :**

**i) Amazon Virtual Private Cloud (VPC)**

VPC stands for Virtual Private Cloud, and it is a service provided by Amazon Web Services (AWS) that allows users to create a private, isolated virtual network in the AWS cloud.

With VPC, users can define a private IP address range, create subnets, and configure routing tables, network gateways, and security settings. Users can also choose to connect their VPC to their on-premises network or other VPCs in the same or different AWS regions.

**What is the use of subnet in VPC AWS?**

A subnet is a range of IP addresses in your VPC. You launch AWS resources, such as Amazon EC2 instances, into your subnets. You can connect a subnet to the internet, other VPCs, and your own data centers, and route traffic to and from your subnets using route tables.

**what is route table in vpc**

In Amazon Web Services (AWS) Virtual Private Cloud (VPC), a route table is a set of rules that determine how network traffic is directed within a VPC or between a VPC and other networks.

Every VPC has a default route table, which is created automatically when the VPC is created. You can also create additional custom route tables and associate them with specific subnets in the VPC.

**What is internet gateway in VPC?**

An internet gateway is a horizontally scaled, redundant, and highly available VPC component that allows communication between your VPC and the internet. It supports IPv4 and IPv6 traffic.

**What is NAT and internet 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.

AWS Direct Connect, and AWS Elastic Load Balancing

**Amazon Route 53:**

Route 53 is a web service that is a highly available and scalable Domain Name System (DNS.)

Let’s understand what is Amazon Route 53 in technical terms. AWS Route 53 lets developers and organizations route end users to their web applications in a very reliable and cost-effective manner. It is a Domain Name System (DNS) that translates domain names into IP addresses to direct traffic to your website. In simple terms, it converts World Wide Web addresses like www.example.com to IP addresses like 10.20.30.40

**What is the use of Route 53?**

Amazon Route 53 is a highly available and scalable Domain Name System (DNS) web service. Route 53 connects user requests to internet applications running on AWS or on-premises.

**What is Vpc Peering**

A VPC peering connection is a networking connection between two vpcs that enables routing each vpc private ip address as if they were in the same network

**1)What is VPC and its components (Public, private, NACL, Route tables, Internet gateway)**

**VPC (Virtual Private Cloud) and its Components:**

VPC is a virtual network dedicated to your AWS account. It allows you to create a logically isolated section of the AWS Cloud where you can launch resources.

Components:

Public Subnet: A subnet with a route to an internet gateway, making resources in this subnet directly accessible from the internet.

Private Subnet: A subnet without a direct route to the internet. Resources in this subnet can access the internet through a Network Address Translation (NAT) gateway.

Network Access Control List (NACL): A stateless, rule-based firewall for controlling inbound and outbound traffic at the subnet level.

Route Tables: Control the traffic between subnets within the VPC. They contain rules that determine where network traffic is directed.

Internet Gateway: A horizontally scalable, redundant, and highly available VPC component that allows communication between instances in your VPC and the internet.

**2)What is difference between NACL and Security groups**

**Difference between NACL and Security Groups**

**NACL (Network Access Control List):** Operates at the subnet level, is stateless, and evaluates traffic based on rule numbers. It can be used to control traffic at the subnet level, allowing or denying traffic based on defined rules.

**Security Groups:** Operate at the instance level, are stateful, and act as a virtual firewall for your instance. They control inbound and outbound traffic by allowing or denying traffic based on rules, and they automatically allow return traffic.

**3)What is differenet types of load balance and tell difference between Network and application load balancer**

**Types of Load Balancers and Differences between Network and Application Load Balancers:**

Types of Load Balancers: There are two main types in AWS - Network Load Balancer (NLB) and Application Load Balancer (ALB).

**Network Load Balancer (NLB):** Operates at the transport (TCP/UDP) layer and is designed for extreme performance and high availability. It is used for routing traffic to a specific target group based on IP protocol data.

**Application Load Balancer (ALB):** Operates at the application layer (HTTP/HTTPS) and is used for routing traffic to different target groups based on content, such as URL or host-based routing.

**4) What is route 53, types of routing policy and routing used**

**Route 53, Types of Routing Policy, and Routing Used:**

**Route 53**: AWS's scalable Domain Name System (DNS) web service.

Types of Routing Policy:

Simple Routing: Routes traffic to a single resource.

Weighted Routing: Distributes traffic based on assigned weights.

Latency-Based Routing: Routes traffic based on the lowest network latency.

Failover Routing: Routes traffic to a backup resource when the primary resource is unhealthy.

Geolocation Routing: Routes traffic based on the geographic location of the user.

Routing can be used to direct traffic to various AWS resources like ELBs, S3 buckets, or even non-AWS resources.

**5)What is ASG and Launch config**

**ASG (Auto Scaling Group) and Launch Configuration:**

**Auto Scaling Group (ASG):** A group of EC2 instances that automatically adjusts its size based on policies to maintain application availability.

**Launch Configuration:** A template that defines the instance type, AMI, and other settings for instances launched by the ASG.

**6)How to restore the login into ec2 if pem file is lost**

Restoring EC2 Login if PEM File is Lost:

You can either create a new key pair and associate it with the instance during launch or stop the instance, detach the root volume, attach it to another instance, and modify the SSH configuration to add your new public key.

**7)How to encrypt the unencrypted AMI**

Encrypting an Unencrypted AMI:

You can create a copy of the unencrypted AMI and specify the option to encrypt the new image during the copy process. This will create an encrypted version of the AMI.

**8)What is EBS and EFS**

EBS (Elastic Block Store) and EFS (Elastic File System):

EBS provides block-level storage volumes that can be attached to EC2 instances.

EFS is a scalable and managed file storage service that can be mounted to multiple EC2 instances.

**9) What is VPC pearing and VPC endpoint**

VPC Peering and VPC Endpoint:

VPC Peering: A networking connection between two VPCs that enables the resources in those VPCs to communicate with each other as if they are on the same network.

VPC Endpoint: Allows private connectivity from your VPC to supported AWS services (e.g., S3, DynamoDB) without needing an internet gateway.

**10)What is S3/types of S3 bucket/S3 bucket policy/S3lifecycle**

S3 (Simple Storage Service):

S3 is an object storage service. Types of S3 buckets include standard, intelligent-tiering, one-zone, and Glacier. You can set bucket policies to control access and define lifecycle policies for automatic object management.

**11) What is IAM roles and how the cross region roles work**

IAM Roles and Cross-Region Roles:

IAM roles define permissions for AWS resources. Cross-region roles are IAM roles that allow AWS resources in one region to access resources in another region. Cross-region roles can be created by trusting another AWS account.

**12) How to tell one Ec2 to talk to other ec2 in other region**

Inter-Region Communication between EC2 Instances:

You can establish inter-region communication using public IP addresses, Virtual Private Network (VPN), Direct Connect, or by setting up VPC peering between VPCs in different regions.

**13)What is MYsql Aurora/Backup plan done/Master and reader endpoints/Failover mechanism**

MySQL Aurora:

Aurora is a fully managed, MySQL and PostgreSQL-compatible relational database engine. It provides features like automated backups, master and reader endpoints, and a failover mechanism for high availability.

**14)What is CFT template/how a resource depends on other resource**

CloudFormation Template (CFT) and Resource Dependencies:

A CFT is a JSON or YAML script that defines AWS resources and their configurations. Resource dependencies are declared in the template to specify the order in which resources are created and ensure they are available for use.

**15)What are importanat components of CFT**

Important Components of CFT:

Template, Resources, Parameters, Mappings, Outputs, and Conditions.

EBS vs. S3 for Storage Speed:

**16) Which is faster storage EBS or S3**

EBS is generally faster for block-level storage, providing low-latency access.

S3 is designed for object storage and is better suited for storing large amounts of data that don't require low-latency access. Speed can vary based on the storage class used in S3.

**17) How does networking and routing work?**

Networking is the communication of devices (nodes) over a network. It involves transferring data packets between devices using protocols like TCP/IP.

Routing determines the path data takes to reach its destination. Routers forward packets between networks using a routing table that keeps track of possible paths.

**18) What are ports?**

Ports are logical endpoints in a network connection, enabling a computer to communicate with multiple services simultaneously.

They are identified by numbers (0–65535). For example:

HTTP: Port 80

HTTPS: Port 443

SSH: Port 22

**19) How do you diagnose a slow Linux machine?**

To troubleshoot:

Check system load: top or htop commands show CPU and memory usage.

Disk I/O: Use iostat or iotop to monitor disk performance.

Memory usage: Use free -m or vmstat.

Processes: Identify resource-hungry processes using ps or top.

Network: Use netstat, iftop, or ping to check for latency issues.

Logs: Check system logs using dmesg or files in /var/log.

**20) How do you check running processes?**

Use ps to list processes, e.g., ps -aux shows detailed information.

Use top or htop for an interactive real-time view.

pgrep searches for processes by name, e.g., pgrep apache2.

**21) What’s the difference between a stateless and a stateful firewall?**

Stateless Firewall:

Filters packets based on predefined rules (e.g., IP, port).

Doesn't track the state of a connection.

Faster but less secure.

Stateful Firewall:

Tracks the state of connections (e.g., TCP handshake).

Makes decisions based on connection context.

Slower but more secure.

**22) How does a web server work?**

A client sends a request (e.g., via a browser) using HTTP/HTTPS.

The web server (e.g., Apache, Nginx) processes the request and maps it to resources (e.g., files or scripts).

If dynamic content is requested, the server may execute a script or query a database.

The response (HTML, CSS, JS) is sent back to the client.

**23) How does DNS work?**

DNS (Domain Name System) translates human-readable domain names (e.g., example.com) into IP addresses (e.g., 192.0.2.1).

Client queries a DNS server.

If the DNS server has the IP in its cache, it responds.

If not, it queries higher-level DNS servers (root, TLD, authoritative).

The IP is returned to the client, enabling a connection.

**24) What are DNS records?**

DNS records store mappings and configuration for a domain. Common types include:

A: Maps a domain to an IPv4 address.

AAAA: Maps a domain to an IPv6 address.

CNAME: Alias for another domain.

MX: Specifies mail servers for the domain.

TXT: Stores arbitrary text, often for verification or security.

NS: Lists authoritative name servers.

PTR: Reverse DNS lookup (IP to domain).

**AWS Scenario Based Interview Questions**

**You have to design a VPC architecture for two Tire application and the application has to be highly available and scalable how do you design this VPC architecture?**

ANS) In this Scenario, I would design a VPC Architecture in the following way:

I would create 2 subnets: public and private. The public subnet would contain the load balancer and be accessible from the internet. The private subnet would host the application servers.

I would distribute the subnet across multiple Availability Zones for high availability. Additionally, I would configure auto scaling group for the application server

**your organization has a VPC with multiple subnets perfect you want to restrict outbound internet access for resources in one subnet but allow outbound internet access for resources in another subnet how would you achieve?**

ANS): To restrict outbound internet access for resources in one subnet, we can modify the route table associated with that subnet. In the route table, we can remove the default route (0.0.0.0/0) that points to an internet gateway.

This would prevent resource in that subnet from accessing the internet. For the subnet where outbound internet access is required, we can keep the default route pointing to the internet gateway.

**You have a VPC with a public subnet and a private subnet. Instances in the private subnet need to access the internet for software updates. How would you allow internet access for instances in the private subnet?**

A: To allow internet access for instances in the private subnet, we can use a NAT Gateway or a NAT instance.

We would place the NAT Gateway/instance in the public subnet and configure the private subnet route table to send outbound traffic to the NAT Gateway/instance. This way, instances in the private subnet can access the internet through the NAT Gateway/instance.

**Q: You have launched EC2 instances in your VPC, and you want them to communicate with each other using private IP addresses. What steps would you take to enable this communication?**

A: By default, instances within the same VPC can communicate with each other using private IP addresses.

To ensure this communication, we need to make sure that the instances are launched in the same VPC and are placed in the same subnet or subnets that are connected through a peering connection or a VPC peering link.

Additionally, we should check the security groups associated with the instances to ensure that the necessary inbound and outbound rules are configured to allow communication between them.

**Q: You want to implement strict network access control for your VPC resources. How would you achieve this?**

A: To implement granular network access control for VPC resources, we can use Network Access Control Lists (ACLs).

NACLs are stateless and operate at the subnet level. We can define inbound and outbound rules in the NACLs to allow or deny traffic based on source and destination IP addresses, ports, and protocols.

By carefully configuring NACL rules, we can enforce fine-grained access control for traffic entering and leaving the subnets.

**Q: Your organization requires an isolated environment within the VPC for running sensitive workloads. How would you set up this isolated environment?**

A: To set up an isolated environment within the VPC, we can create a subnet with no internet gateway attached.

This subnet, known as an "isolated subnet," will not have direct internet connectivity. We can place the sensitive workloads in this subnet, ensuring that they are protected from inbound and outbound internet traffic.

However, if these workloads require outbound internet access, we can set up a NAT Gateway or NAT instance in a different subnet and configure the isolated subnet's route table to send outbound traffic through the NAT Gateway/instance.

**Q: Your application needs to access AWS services, such as S3 securely within your VPC. How would you achieve this?**

A: To securely access AWS services within the VPC, we can use VPC endpoints. VPC endpoints allow instances in the VPC to communicate with AWS services privately, without requiring internet gateways or NAT gateways.

We can create VPC endpoints for specific AWS services, such as S3 and DynamoDB, and associate them with the VPC.

This enables secure and efficient communication between the instances in the VPC and the AWS services.

**Q: What is the difference between NACL and Security groups ? Explain with a use case ?**

A: For example, I want to design a security architecture, I would use a combination of NACLs and security groups. At the subnet level, I would configure NACLs to enforce inbound and outbound traffic restrictions based on source and destination IP addresses, ports, and protocols. NACLs are stateless and can provide an additional layer of defense by filtering traffic at the subnet boundary.

At the instance level, I would leverage security groups to control inbound and outbound traffic. Security groups are stateful and operate at the instance level. By carefully defining security group rules, I can allow or deny specific traffic to and from the instances based on the application's security requirements.

By combining NACLs and security groups, I can achieve granular security controls at both the network and instance level, providing defense-in-depth for the sensitive application.

**Q: What is the difference between IAM users, groups, roles and policies ?**

A: IAM User: An IAM user is an identity within AWS that represents an individual or application needing access to AWS resources. IAM users have permanent long-term credentials, such as a username and password, or access keys (Access Key ID and Secret Access Key). IAM users can be assigned directly to IAM policies or added to IAM groups for easier management of permissions.

IAM Role: An IAM role is similar to an IAM user but is not associated with a specific individual. Instead, it is assumed by entities such as IAM users, applications, or services to obtain temporary security credentials. IAM roles are useful when you want to grant permissions to entities that are external to your AWS account or when you want to delegate access to AWS resources across accounts. IAM roles have policies attached to them that define the permissions granted when the role is assumed.

IAM Group: An IAM group is a collection of IAM users. By organizing IAM users into groups, you can manage permissions collectively. IAM groups make it easier to assign permissions to multiple users simultaneously. Users within an IAM group inherit the permissions assigned to that group. For example, you can create a "Developers" group and assign appropriate policies to grant permissions required for developers across your organization.

IAM Policy: An IAM policy is a document that defines permissions and access controls in AWS. IAM policies can be attached to IAM users, IAM roles, and IAM groups to define what actions can be performed on which AWS resources. IAM policies use JSON (JavaScript Object Notation) syntax to specify the permissions and can be created and managed independently of the users, roles, or groups. IAM policies consist of statements that include the actions allowed or denied, the resources on which the actions can be performed, and any additional conditions.

**Q: You have a private subnet in your VPC that contains a number of instances that should not have direct internet access. However, you still need to be able to securely access these instances for administrative purposes. How would you set up a bastion host to facilitate this access?**

A: To securely access the instances in the private subnet, you can set up a bastion host (also known as a jump host or jump box). The bastion host acts as a secure entry point to your private subnet. Here's how you can set up a bastion host:

Create a new EC2 instance in a public subnet, which will serve as the bastion host. Ensure that this instance has a public IP address or is associated with an Elastic IP address for persistent access.

Configure the security group for the bastion host to allow inbound SSH (or RDP for Windows) traffic from your IP address or a restricted range of trusted IP addresses. This limits access to the bastion host to authorized administrators only.

Place the instances in the private subnet and configure their security groups to allow inbound SSH (or RDP) traffic from the bastion host security group.

SSH (or RDP) into the bastion host using your private key or password. From the bastion host, you can then SSH (or RDP) into the instances in the private subnet using their private IP addresses.