AWS INTERVIEW QUESTIONS

Q1: Can we attach multiple target groups in one LB?
Yes, we can attach multiple target groups in one LB

Q2: How to create an internal load balancer?
There are 2 types of load balancers. One is internet-facing and other one is internal, just define the type while creating it.

Q3: How to resize the EBS volume?
Go to edit the volume and select modify the volume and change the size, IOPS, etc.

Q4: Can we attach the same EBS volumes to multiple instances?
No, we can't attach. We need to use the EFS for that.

Q5: What is the transit gateway in AWS?
A transit gateway is a network transit hub that you can use to interconnect your virtual private clouds (VPCs) and on-premises networks.

Q6: What is the use of lifecycle hooks in Autoscaling?
Amazon EC2 Auto Scaling offers the ability to add lifecycle hooks to your Auto Scaling groups. These hooks let you create solutions that are aware of events in the Auto Scaling instance lifecycle, and then perform a custom action on instances when the corresponding lifecycle event occurs. A lifecycle hook provides a specified amount of time (one hour by default) to wait for the action to complete before the instance transitions to the next state.

Q7: I want to create the VPC, what cloud is the maximum CIDR range you can provide?
10.0.0.0/16

Q8: How you will add your own system Ip under the security group?
Find the IP of the system using ifconfig command and add it to the SG

Q9: We have two subnets one is public and the second is private so how you will identify which is public and which is private?
Private Subnets:

Class A: 10.0.0.0 to 10.255.255.255
Class B: 172.16.0.0 to 172.31.255.255
Class C: 192.168.0.0 to 192.168.255.255

Public Subnets:
All other IP ranges not falling into the above private ranges are generally considered public.

Q10: If I have configured one application(HTTPD) on EC2 instance private subnet and now I want to access the URL from the web browser, how it will be possible?
Yes, it is possible. Here is the approach.

You can use a NAT Gateway
In the route table associated with your private subnet, add a route that sends all outgoing traffic (0.0.0.0/0) to the NAT Gateway

Q11: What is the use of a NAT gateway? and is it one-way communication or two-way communication?
NAT gateway facilitates two-way communication. Devices within the private network can initiate outbound connections to the internet, and the NAT gateway manages the translation of IP addresses and ports, allowing responses from the internet to be correctly delivered back to the respective devices within the private network.

Q12: If you want to access your private instance from your local machine so will you be able to access using private IP?
Private IP addresses are meant for internal communication within a private network only
VPN (Virtual Private Network): Set up a VPN connection between your local machine and the private network where the instance is located. Once connected to the VPN, your local machine will effectively be part of the private network and can access resources, including the private instance, using its private IP address.

SSH Tunneling (Port Forwarding): If the private instance has SSH enabled and you have SSH access to it, you can create an SSH tunnel from your local machine to the private instance.

Bastion Host or Jump Box: If you have a bastion host or jump box set up in the private network, you can first SSH into the bastion host and then SSH

from the bastion host to the private instance. This allows you to access the private instance indirectly.

Q13: How can we access the application through the web browser if the application is hosted on a private subnet instance? what are the ways?

Using NAT Gateway:
Proxy Server: Set up a proxy server in the public subnet, which acts as an intermediary between the user and the private application.
VPN-based Cloud Services: Some cloud providers offer managed VPN solutions, like AWS Client VPN

Q14: If you have your own S3 bucket and provide access to certain IP ranges/addresses not to everyone. How you will do? and which policy you will add? specify the type of policy you will add with your name ie. Role-based policy, service-based policy, inline policy, or custom policy
To provide access to a certain IP range/address and restrict access to everyone else in your own S3 bucket, you can use an IAM (Identity and Access Management) policy to control the permissions. IAM policies allow you to define who can access your S3 bucket and what actions they can perform.

In this scenario, you would use a custom IAM policy to define the specific IP range or addresses that are allowed access to the S3 bucket. The policy will be attached to the IAM users, groups, or roles that need access to the bucket.

Q15: If we are getting data under S3 standard storage class and I want to move that data every last of the month into Glacier, how you will do?
To automate the process of moving data from the S3 Standard storage class to Glacier every last day of the month, you can use AWS Lambda along with CloudWatch Events.
Set up an S3 Lifecycle policy to transition objects to the Glacier storage

Q16: Do you know the type of LB?
Classic Load Balancer (CLB): The original ELB, which provides basic load balancing across multiple Amazon EC2 instances. It operates at both the application and transport layers.

Application Load Balancer (ALB): A more advanced load balancer that operates at the application layer (Layer 7) and is designed to route traffic

to different targets based on the content of the request. ALBs are well-suited for HTTP and HTTPS traffic, supporting features like host-based routing, path-based routing, and support for WebSocket and HTTP/2 protocols.

Network Load Balancer (NLB): A high-performance load balancer that operates at the transport layer (Layer 4). It is designed to handle large amounts of traffic and is best suited for TCP, UDP, and TLS traffic. NLBs are often used for scenarios that require low latency and high throughput.

Q17: Do you know the flow of application LB? end to end?
Incoming Client Request: The flow begins when a client, such as a web browser or a mobile app, sends a request to access a specific application or service hosted on the load balancer's IP address or domain name.

Load Balancer Receives the Request: The application load balancer (ALB) sits between the client and the backend application servers. It receives the incoming client request on a specific port (e.g., port 80 for HTTP or port 443 for HTTPS).

Load Balancer Selection Algorithm: The ALB uses a load balancing algorithm to determine which backend server should handle the incoming request. The selection process can be based on various factors, such as round-robin, least connections, least response time, etc.

Load Balancer Routes the Request: Once the backend server is selected, the load balancer forwards the client request to that server. The backend server processes the request and generates a response.

Backend Server Processes the Request: The selected backend server processes the client request, performs any necessary computations, and generates a response. This response may include the requested data or an error message.

Response Sent to Load Balancer: After processing the request, the backend server sends the response back to the load balancer.

Load Balancer Sends Response to Client: The load balancer receives the response from the backend server and forwards it to the client that made the initial request.

Client Receives Response: The client (e.g., web browser or mobile app) receives the response from the load balancer. From the client's perspective, it appears as if the request was directly processed by a single server, unaware of the load balancing behind the scenes.

Scaling and High Availability: Application Load Balancers are often deployed in a redundant configuration with multiple instances to ensure high availability and fault tolerance. If any backend server becomes unavailable, the load balancer will automatically route traffic to the healthy servers.

Q18: If you have three instances in the target group so whenever any instance gets down, how you will come to know that one instance is down?

To detect when an instance in the target group goes down, you can set up an automated monitoring and alerting system. AWS, for example, provides various services that can help you achieve this. Here's one possible approach using AWS services:

Amazon CloudWatch: Set up CloudWatch to monitor the health of the instances in your target group. CloudWatch can collect and track metrics like CPU utilization, network traffic, and status checks.

CloudWatch Alarms: Create CloudWatch alarms based on specific conditions, such as when the status check fails for an instance or when the instance's health check status becomes "unhealthy."

Amazon SNS (Simple Notification Service): Configure an SNS topic to send notifications when an alarm is triggered. This topic can have one or more subscribers, such as email addresses, SMS numbers, or other AWS services.

Amazon Auto Scaling: If you are using Auto Scaling, it can automatically replace failed instances. In such cases, you can configure Auto Scaling to use the SNS topic as a notification mechanism.

Q19: Have you created any RDS instances? could you specify the multi-AZ and read replica?

Multi-AZ Deployment:

In Multi-AZ (Availability Zone) deployment, Amazon RDS automatically provisions and maintains a synchronous standby replica of your primary

database in a different Availability Zone. This standby replica provides data redundancy and automatic failover in the event of an infrastructure failure or maintenance event affecting the primary database. If the primary database becomes unavailable, Amazon RDS will automatically promote the standby replica to become the new primary database. Multi-AZ is a high-availability feature that helps enhance the resilience of your database.

Read Replicas:
Read replicas in Amazon RDS are copies of the primary database instance that are asynchronously updated from the primary instance. They are used to offload read traffic from the primary database, thereby improving read performance and scalability. Read replicas can be created in the same region as the primary database or in different regions, allowing you to distribute read traffic across various locations

Q20: Cloud formation vs Terraform
Cloudformation is the IaC tool that is limited to AWS only. You can use cloud formation templates to provision the infrastructure on AWS. We need to create a stack in Cloudformation and then need to deploy those stacks

While Terraform is an IaC tool that can interact with any cloud provider like AWS, GCP & Azure. We need to write the manifest files in Terraform and need to use Terraform commands to provision the infrastructure. Terraform maintains the information about the infrastructure in a state file called terraform.tfstate

Q21: Cloudtrail vs Cloudwatch
AWS CloudTrail:
CloudTrail is a service that enables you to monitor and audit the actions taken by users, applications, and services in your AWS account.
It provides detailed event history of activities that occur within your account, such as API calls made on resources, changes to resource configurations, and more.

Amazon CloudWatch:
Amazon CloudWatch is a monitoring and observability service that provides insights into the performance and operational health of your

AWS resources and applications. It collects and tracks metrics, creates alarms, and generates logs.

CloudWatch collects and stores metrics (such as CPU usage, network traffic, etc.) from various AWS services and custom sources.
You can set up alarms based on thresholds for certain metrics, and CloudWatch can take automated actions when those thresholds are breached.

Q22: When to use AWS Fargate?
You can use AWS Fargate in the following scenarios:

Containerized Applications: When you have containerized applications using Docker, Fargate can manage the deployment, scaling, and infrastructure for those containers.
Microservices: Fargate is well-suited for deploying and managing microservices architectures where each service can be containerized and deployed independently.

Serverless Container Management: Fargate provides a serverless computing model for containers, allowing you to only pay for the resources used by your containers without worrying about server provisioning and management.

Simplified Operations: If you want to reduce operational overhead and focus more on application development, Fargate abstracts away the need to manage the underlying instances.
Scalability: Fargate makes it easier to scale your containers as needed without the need to manage the scaling process manually.

Q23: Can we use EFS as the root volume?
No, we can’t use only EBS can be used as the root volume