Devops Real Interview Questions

Q.1: WHAT IS THE DIFFERENCE BETWEEN A SECURITY GROUP AND NACL? 1
Q.2: WHAT IS THE ROLE OF ROUTING TABLE IN VPC? 1
Q.3: WHAT IS VPC? AND WHAT ARE KEY COMPONENTS OF VPC? 2
Q.4: WHEN WE USE PRIVATE SUBNET AND WHEN PUBLIC. 2
Q.5: WHAT IS SSH? 2
Q.6: WHAT IS SSH TUNNELLING? 2
Q.7: WHAT ARE THE TYPES OF DOCKER NETWORK? 2
Q.8: WHEN WE CREATE ECS, WHICH NETWORK IT USE. 3
Q.9: CORE COMPONENT OF KUBERNETES. 3
Q.10: WHEN WE USE S3? 3
Q.11: WHAT TYPE OF DATA WE CAN STORE IN S3. 4
Q.12: WHAT IS IAM? WHAT ARE THE COMPONENT OF IAM. 4
Q.13: WHAT IS ROLE IN IAM? 4
Q.14: WHAT IS ROLE OF KUBE API? 5
Q.15: WHAT IS BRIDGE NETWORK? 5
Q.16: WHEN WE USE IGW? 5
Q.17: WHEN WE USE NAT GATEWAY? AND WHY. 6
Q.18: HOW MANY TYPES OF IAM ROLES DO WE HAVE? NAME SOME OF FOLLOWINGS. 6
Q.19: HOW CAN YOU CONNECT AN ON-PREMISES DATA CENTER TO AWS? 6

Q.1: What is the difference between a Security Group and NACL?

A **Security Group (SG)** acts as a **firewall at the instance level**, controlling inbound and outbound traffic for EC2 instances. It is **stateful**, meaning if an inbound rule allows traffic, the outbound response is automatically allowed. SGs only support **allow rules** and do not allow explicit denies.

A **Network ACL (NACL)** operates at the **subnet level**, controlling traffic for all instances within a subnet. It is **stateless**, meaning inbound and outbound rules must be defined separately. NACLs support both **allow and deny rules**, and rules are evaluated in **numerical order (lowest first)**.

Q.2: What is the Role of routing table in VPC?

A **Routing Table** in an AWS VPC is responsible for determining where network traffic should be directed. It contains **routes** that specify which subnet, gateway (Internet Gateway, NAT Gateway, VPN), or peering connection should be used for traffic.

- **Routes in the Routing Table:**
    - **Local Route** (default) allows communication within the VPC.

- **0.0.0.0/0 → Internet Gateway** for public subnets (Internet access).
- **0.0.0.0/0 → NAT Gateway** for private subnets (outbound internet access).
- Custom routes for VPC Peering, VPN, and Direct Connect.

Q.3: What is VPC? and what are key components of VPC?

A **VPC (Virtual Private Cloud)** is a logically isolated network in AWS where you can launch AWS resources like EC2, RDS, etc.

Key Components of VPC:

1. **Subnets:** Public and private sections of a VPC.
2. **Route Tables:** Defines traffic routing rules.
3. **Internet Gateway (IGW):** Enables public internet access.
4. **NAT Gateway:** Allows private subnets to access the internet.
5. **Security Groups (SG):** Controls traffic at the instance level.
6. **Network ACLs (NACL):** Controls traffic at the subnet level.
7. **Peering/VPN/Direct Connect:** Enables external network connectivity.

Q.4: When we use Private subnet and when Public.

- **Public Subnet:** Used when resources (like web servers, ALBs) need direct internet access. It has a route to the Internet Gateway.
- **Private Subnet:** Used when resources (like databases, backend servers) should be isolated from direct internet access. It requires a NAT Gateway for outbound internet access.**No table of contents entries found.**

Q.5: What is SSH?

SSH (**Secure Shell**) is a protocol used to securely connect to remote systems over an encrypted channel. It is commonly used to log into Linux-based servers for administration and execution of commands.

Q.6: What is SSH tunnelling?

SSH tunnelling is a technique that **creates a secure encrypted connection between a local machine and a remote server** and forwards network traffic securely. It is used for accessing private resources through an SSH connection.

Q.7: What are the Types of Docker Network?

Docker provides several network types to connect containers:

1. **Bridge Network (Default):** Containers on the same bridge can communicate.
2. **Host Network:** Uses the host machine's network stack directly.

3. **Overlay Network:** Used in Docker Swarm to connect multiple hosts.
4. **Macvlan Network:** Assigns MAC addresses to containers for direct LAN access.
5. **None Network:** No network access.

Q.8: When we create ECS, which network it use.
Amazon **ECS (Elastic Container Service)** can use:

- **AWS VPC Networking Mode (awsvpc)** - Provides each container with a private IP.
- **Bridge Networking** - Default Docker network mode.
- **Host Mode** - Containers use the host's networking.

Best Practice: Use awsvpc mode for better security & flexibility.

when you create an **Amazon ECS (Elastic Container Service)** cluster, it uses the **"bridge" network mode** for containers running on **EC2 launch type**.

However, if you use **AWS Fargate**, the default network mode is **"awsvpc"**, which provides each task with its own elastic network interface (ENI) and private IP address.

Default Network Mode Based on Launch Type:

ECS Launch Type	Default Network Mode
EC2	bridge
Fargate	awsvpc

Q.9: core component of Kubernetes.
Master Node Components:

- **API Server:** Handles requests.
- **Controller Manager:** Manages controllers (e.g., Node, Replication).
- **Scheduler:** Assigns workloads to nodes.
- **etcd:** Stores cluster configuration.

Worker Node Components:

- **Kubelet:** Communicates with the API server.
- **Kube Proxy:** Manages networking.
- **Container Runtime (Docker/Containerd):** Runs containers.

Q.10: When we use S3?

Amazon S3 (Simple Storage Service) is used whenever we need scalable, durable, and highly available object storage in the cloud**.** We use S3 in various scenarios**, including:**

1. **Storing Static Website Content** – Hosting images, CSS, JavaScript, and even full static websites.
2. **Backup & Archival** – Storing backups, logs, and long-term data retention (using lifecycle policies).
3. **Data Lake & Big Data Analytics** – Storing raw and processed data for analysis with AWS services like Athena, Redshift, and EMR.
4. **Media Storage & Streaming** – Storing and delivering videos, audio files, and images using Amazon CloudFront for CDN.
5. **Machine Learning & AI** – Storing large datasets for AI model training.
6. **Disaster Recovery** – Replicating data across multiple regions for fault tolerance.
7. **Application File Storage** – Storing user uploads such as documents, profile pictures, and reports.

Q.11: what type of data we can store in S3.

Amazon S3 is an object storage service, meaning we can store virtually any type of unstructured data in it. Some common types of data stored in S3 include:

- Documents, images, videos.
- Logs & backups.
- Big data analytics files.
- Static website files.

Q.12: What is IAM? what are the component of IAM.

AWS IAM (Identity and Access Management) is a service that helps securely control access to AWS resources. It enables authentication and authorization, allowing users and services to interact with AWS securely.

Key Components of IAM:

1. **Users** – Individual accounts representing people or applications that interact with AWS
2. **Groups** – A collection of users that share the same permissions (e.g., Admins, Developers).
3. **Roles** – Assignable permissions for AWS services or users to assume (used for cross-account access, EC2 roles, Lambda roles, etc.).
4. **Policies** – JSON-based permission rules that define what actions are allowed or denied on AWS resources.
5. **MFA (Multi-Factor Authentication)** – Adds an extra security layer by requiring a second authentication factor.
6. **Access Keys & Credentials** – Used for programmatic access (API calls, CLI).
7. **IAM Identity Center (AWS SSO)** – Manages multiple user logins across AWS accounts.

Q.13: What is Role in IAM?

An IAM Role in AWS is an identity that grants temporary permissions to users, applications, or AWS services without requiring long-term credentials like passwords or access keys.

IAM roles are used when:

- An **EC2 instance** needs access to S3, DynamoDB, or other AWS services.

- An **AWS Lambda function** needs permissions to read from S3 or write to a database.
- A user from **another AWS account** (cross-account access) needs access to resources.
- **AWS services (e.g., ECS, Redshift, RDS)** require permissions to interact with other AWS services.

Q.14: What is role of Kube API?

The Kubernetes API Server (Kube API) is the central control plane component that acts as an interface for interacting with the Kubernetes cluster. It handles all API requests, validates them, and updates the cluster state accordingly.

Key Roles of Kube API:

1. **Manages Cluster Communication** – Acts as the main entry point for managing the cluster, handling requests from kubectl, controllers, and external services.
2. **Validates and Processes Requests** – Ensures all API requests (such as pod creation, scaling, or updates) are properly authenticated, authorized, and follow Kubernetes policies.
3. **Stores Cluster State in etcd** – Persists cluster data in **etcd**, a key-value store used by Kubernetes to maintain the desired state of objects like pods, services, and deployments.
4. **Controls Workloads and Resources** – API requests trigger actions like scheduling pods, updating deployments, and scaling workloads.
5. **Serves as the Gateway for Controllers & Operators** – Other Kubernetes components (like the **scheduler, controllers, and operators**) interact with the cluster through the Kube API.

Q.15: What is Bridge Network?

A Bridge Network in Docker is the default network mode that allows containers on the same host to communicate with each other while isolating them from external networks.

Key Features of Bridge Network:

1. **Default Network Mode** – If no network is specified, Docker automatically creates a bridge network named bridge.
2. **Container-to-Container Communication** – Containers connected to the same bridge network can communicate using container names as hostnames.
3. **Isolation from Host** – Containers in a bridge network cannot communicate with the host machine unless explicitly mapped using port forwarding (-p flag).
4. **Manual Customization** – You can create a user-defined bridge network to improve networking control, name resolution, and performance.

Q.16: When we use IGW?

An **Internet Gateway (IGW)** is used **when resources in a public subnet need to communicate with the internet**. It allows inbound and outbound traffic between the VPC and the internet.

Use IGW When:

- You want **public-facing EC2 instances** to access the internet.
- You need to **host a web server** accessible globally.
- You want your instances to **download updates and patches** from external sources.

- You are setting up a **public-facing Load Balancer (ELB)**.

Q.17: When we use Nat Gateway? and why.

A **NAT (Network Address Translation) Gateway** is used **when private subnet instances need to access the internet but should remain inaccessible from the internet**.

Use NAT Gateway When:

- You need **instances in a private subnet** to download software updates.
- Your application **fetches data from external APIs** without exposing itself.
- You want a **database or backend service** to communicate with the internet securely.

Q.18: How many types of IAM roles do we have? name some of followings.

IAM Roles provide **temporary permissions** to AWS services, users, or applications.

Types of IAM Roles:

1. **Service-Linked Roles** – Automatically created for AWS services (e.g., AWSServiceRoleForEC2).
2. **AWS Managed Roles** – Predefined roles provided by AWS (e.g., AmazonEC2RoleforSSM).
3. **Custom IAM Roles** – User-defined roles with specific permissions.
4. **Cross-Account Roles** – Used to grant access between AWS accounts.
5. **Federated Roles** – Used with **SSO (Single Sign-On)** or identity providers (Okta, Azure AD).

Q.19: How can you connect an on-premises data center to AWS?

There are four main ways to connect an **on-premises** environment to AWS, depending on the use case, security, and performance requirements

1. **Site-to-Site VPN** (uses **encrypted tunnels over the internet**; cost-effective but higher latency).
2. **Direct Connect (DX)** (dedicated private fiber connection; low latency, high performance).
3. **Transit Gateway** (for managing **multiple connections** in a large-scale hybrid cloud).
4. **SD-WAN** (intelligent traffic routing between **multiple ISPs and clouds**).

For a quick setup, use VPN.
For low-latency enterprise workloads, use Direct Connect.
For large-scale networking, use Transit Gateway.
For multi-cloud environments, use SD-WAN.

1. **AWS Site-to-Site VPN (IPSec VPN) – Encrypted over Public Internet**

When to use:

- If you need a quick and cost-effective solution.
- When moderate latency is acceptable.

How it works:

- Uses **IPSec tunnels** to connect **on-premises routers** to AWS **Virtual Private Gateway (VGW)**.
- Traffic is **encrypted** but passes through the **public internet**.

Example Setup:

- Create a **Customer Gateway (CGW)** in AWS using the public IP of the on-premises router.
- Attach a **Virtual Private Gateway (VGW)** to the AWS **VPC**.
- Establish an **IPSec tunnel** between them.

Limitations:

- **Higher latency** since traffic flows over the public internet.

2. AWS Direct Connect (DX) – Dedicated Private Link

When to use:

- When you need **high bandwidth (1-10 Gbps)** with **low latency**.
- Suitable for **large-scale workloads** like databases and hybrid cloud applications.

How it works:

- AWS **provides a dedicated physical fiber** connection from **on-premises to AWS** via a colocation facility.
- Uses **BGP routing** for traffic control.
- More **secure and stable** than a VPN.

Example Setup:

- Order a **Direct Connect** line via the AWS console.
- Configure the **on-premises router** to establish the connection.
- Use **AWS Transit Gateway** if connecting multiple VPCs.

Limitations:

- **Expensive** and takes weeks to set up.
- Requires a **colocation provider** (e.g., Equinix).

3. AWS Transit Gateway – Centralized Hybrid Cloud Networking

When to use:

- If you need to connect **multiple on-premises locations** to **multiple AWS VPCs**.
- When scaling beyond **one VPN or Direct Connect link**.

How it works:

- Acts as a **centralized hub** to manage **multiple VPNs, Direct Connects, and VPC peering**.
- Reduces **complexity** compared to managing multiple VPN connections individually.

Example Setup:

- Create an **AWS Transit Gateway**.
- Attach **VPCs and VPN connections** to the Transit Gateway.
- Configure **route tables** to direct traffic appropriately.

Limitations:

- **Complex routing** setup.
- Can introduce **network bottlenecks** if not configured properly.

4. AWS VPN with SD-WAN – Intelligent Multi-Cloud Routing

When to use:

- When using **multiple ISPs or cloud providers**.
- When you need **dynamic routing and automated failover**.

How it works:

- Uses **SD-WAN software (Cisco, Palo Alto, Fortinet, etc.)** to manage multiple VPN tunnels dynamically.
- Automatically **routes traffic** based on latency, packet loss, or congestion.

Example Setup:

- Deploy an **SD-WAN virtual appliance** in AWS.
- Connect the **on-prem SD-WAN router** to AWS.
- Configure **policy-based routing** for optimal traffic flow.

Limitations:

- Requires **SD-WAN vendor support**.
- More **expensive** than traditional VPNs.

Choosing the Right Connectivity Option

Connectivity Option	Speed	Security	Cost	Best Use Case
Site-to-Site VPN	Medium	✅ Secure	✅ Low	Small businesses, quick setup
Direct Connect	High	✅✅ Secure	❌ High	Large enterprises, low latency

Connectivity Option	Speed	Security	Cost	Best Use Case
Transit Gateway	Medium-High	✅ Secure	✅ Moderate	Multi-VPC, multi-region networking
SD-WAN	High	✅ ✅ Secure	❌ High	Enterprises using multi-cloud