AWS

**What are the security components used in devops?**

For our AWS cloud we use IAM and its policies to provide access to users and coming to ec2 servers we have security groups and NACL's to get access on specific ports and protocols.

IAM, KMS, HSM, Security Groups, NACL’s, Directory Services …...........for security   
**1. Identity & Access Management**

|  |  |
| --- | --- |
| **Service** | **Purpose** |
| **IAM (Identity and Access Management)** | Manage users, groups, roles, and fine-grained permissions using policies |
| **IAM Roles** | Grant temporary credentials to services like EC2, Lambda |
| **IAM Policies** | JSON-based rules to control access to AWS resources |
| **AWS Organizations** | Manage and apply service control policies (SCPs) across accounts |

## **🔏 2. Network Security**

|  |  |
| --- | --- |
| **Service** | **Purpose** |
| **VPC Security Groups** | Virtual firewalls for EC2 instances (stateful) |
| **Network ACLs (NACLs)** | Stateless filtering rules for subnets |
| **VPC Flow Logs** | Capture IP traffic metadata for network interfaces |
| **AWS Firewall Manager** | Centralized firewall rule enforcement |
| **AWS WAF (Web Application Firewall)** | Protects against common web attacks (SQLi, XSS, etc.) |
| **AWS Shield** | DDoS protection (Standard is free, Advanced is paid) |
| **NAT Gateway** | Restrict outbound internet access for private subnets |

## **🔒 3. Data Protection**

|  |  |
| --- | --- |
| **Service** | **Purpose** |
| **KMS (Key Management Service)** | Manage encryption keys (integrated with most AWS services) |
| **CloudHSM** | Hardware-based key storage (FIPS 140-2 Level 3) |
| **S3 Encryption** | SSE-S3, SSE-KMS, SSE-C options to encrypt stored data |
| **EBS Encryption** | Encrypt volumes at rest and in transit |
| **TLS/SSL** | Encryption in transit (e.g., HTTPS for CloudFront, ALB, etc.) |

## **🧠 4. Threat Detection & Monitoring**

|  |  |
| --- | --- |
| **Service** | **Purpose** |
| **Amazon GuardDuty** | Intelligent threat detection using machine learning |
| **Amazon Inspector** | Automatically scans EC2 and containers for vulnerabilities |
| **Security Hub** | Centralized security dashboard for findings from GuardDuty, Inspector, etc. |
| **CloudTrail** | Logs all API calls across your AWS account |
| **AWS Config** | Tracks and audits configuration changes and compliance |
| **CloudWatch Logs/Alarms** | Monitor logs and set alarms for suspicious activity |

## **✅ 5. Authentication & Federation**

|  |  |
| --- | --- |
| **Service** | **Purpose** |
| **Cognito** | Add user sign-up, sign-in, and federation to web/mobile apps |
| **SSO (Single Sign-On)** | Federate access to AWS accounts and apps |
| **MFA (Multi-Factor Authentication)** | Strengthen access security for IAM users and roles |

## **🧱 6. Governance & Compliance**

|  |  |
| --- | --- |
| **Service** | **Purpose** |
| **AWS Config Rules** | Define and enforce compliance rules (e.g., “S3 buckets must be encrypted”) |
| **Service Control Policies (SCP)** | Restrict service usage across AWS Org accounts |
| **Resource Access Manager (RAM)** | Share resources securely across accounts |
| **Amazon Macie** | Finds and protects sensitive data like PII in S3 |

## **🛡️ Summary Diagram**

pgsql

CopyEdit

┌────────────────────────────┐  
│ IAM / MFA / SSO │ ← Identity & Authentication  
├────────────────────────────┤  
│ VPC, Security Groups, NACL │ ← Network Security  
├────────────────────────────┤  
│ S3/EBS Encryption, KMS │ ← Data Protection  
├────────────────────────────┤  
│ CloudTrail, GuardDuty │ ← Monitoring & Threat Detection  
├────────────────────────────┤  
│ AWS Config, Security Hub │ ← Governance & Compliance  
└────────────────────────────┘

1. **I have ecommerce-based website on a EC2 instance and we have got 100% Traffie, what do you suggest of a better infrastructure where the client should not get failed in accessing ecommerce website**

When an EC2 Instance is hosting ecommerce firstly we can attach an ELB with autoscaling group, so when the traffic towards the ecommerce website increases. decreases the autoscaling group automatically adjust the adding and removing of EC2 instance to balance the work load on the EC2 instances and it also registers and deregisters EC2 instances with the ELB and the ELB rebalance feature helps to distribute traffic evenly among the ec2 instances in different AZ’s. We can configure it with CW alarms and set the threshold limits for example if the CPU utilization is greater than 80 % trigger an alarm to initiate the ASG to add ec2 instance to balance the workload. We can even integrate it with SNS to get notified whenever the instance is launched, terminated, failed to launch or failed to terminate.

1. **I have a VPC in one region and VPC in another region I want to connect those two VPCs what is the best option you suggest?**

We can go with VPC peering connection across regions. To establish peering connection both the VPC’s shouldn’t have overlapping IP addresses. For example, we have VPC in region A want a peering connection with VPC in region B. Therefor the VPC owner in region should send requesting a peering connection to the VPC in region B. The region B VPC owner should accept the request to establish a peering connection between those VPC’s. Then update either side of the route tables with the CIDR of the VPCs for a private communication exchange.

1. **I have an ELB load balancer and there are multiple instances coming and I want specific user session to bind to specific VM what is the setting you need to do?**

We can enable sticky session and we can set the session cookie value and we can give time in minutes so that the user will bind to that particular instance for a particular period of time. Go to target groups -> select target group -> Attributes -> edit

Enable stickiness -> type load balancing -> give time in minutes

In this case request will flow to one server

1. **I have an RDS instance with multi-AZ deployment. Can I use standby instance to read/write operations?**

we can’t read/write to standby instance which is configured In a different AZ . It is basically for a failover to happen, whenever the main RDS goes down due to network breakdown or disaster it will automatically failover to the standby instance.

1. **Have you used Elastic Beanstalk and AWS CloudFormation?**

No don’t have hands-on experience on Elastic Beanstalk and CloudFormation. But instead of CloudFormation I used Terraform for provisioning infrastructure and in terraform I wrote terraform scripts to automate my EKS cluster creation on AWS.

1. **Whenever the system admin tries to access the ec2 instance he gets error host key not found. What is the troubleshooting method you require?**

That means your authentication has failed due to wrong username input or the wrong .pem file. You need verify that you are using correct username or not and also the correct .pem file

1. **Have you used fargate and EKS?**

I used EKS and in my project I have provisioned EKS cluster with terraform script and I even deployed 6 microservices by integrating EKS cluster with Jenkins where we build and deployed those microservices onto the EKS cluster.

I don’t have much hands-on experience on fargate that which allows you to run containerized applications  
- allows you to run containerized applications without managing the underlying EC2 instances. It’s a **serverless** option for running containers

1. **We have an ec2 instance and we want ec2 logs to CloudWatch. How do you configure?**

We can enable cloud watch detailed monitoring at the time of instance launch while configuring instance and also by installing CW log unified agent on a running EC2 instance. ( sudo yum install amazon-cloudwatch-agent\_) "Effect": "Allow", "Action":

[ "logs:CreateLogGroup",

"logs:CreateLogStream", "logs:PutLogEvents", "logs:DescribeLogStreams" ],

By attaching an Iam role with required permission, then installing awslogs package on a running ec2(sudo yum install -y awslogs), then start the awslogs service(**sudo systemctl start awslogsd**) and edit the awscli.conf file to point to the specific region which is at /etc/awslogs/awscli.conf

1. **A 3-tier architecture in AWS that you designed?**

First, I started creating a networking base by creating a VPC with a public subnet in each AZ, these subnets will have an internet facing load balancer. Then I created private subnet in each AZ for application deployment and finally I created private subnets for database instances.

Then I created internet gateway and attached it to the VPC and a NAT gateway with elastic IP in one of the public subnets for the private subnet ec2 instances to get to the internet.

Then created corresponding route tables and attached then to the respective subnets.

Then I launched required no of instances in public and private subnets and created internet facing load Balancer mapped to public subnets with a target group and registered ec2 instances so that the load balancer load balances traffic initiated among the ec2 instances.

1. **Where you will open ports?**

We will open required ports in security groups and NACL’s.’

1. **You lost ec2 .pem key and you want to access the instance how will you access the instance?**

Accessing the EC2 instance even if you lose the pem file is rather easy.

* 1. First, create a new instance by creating new access file, call it 'helper' instance with same region and VPC as of the lost pem file instance.
  2. Now stop the lost pem file instance. Remember not to terminate instance but to stop it.
  3. Go to EBS volumes, select the root volume of the lost pem file instance and detach.
  4. Now again select the detached volume and this time you have to attach this volume to helper instance which we created before. Since helper instance already has a root volume by default as /dev/sda1, the newly attached volume will be secondary(eg: /dev/sdf).
  5. Login to your helper instance with its pem file.
  6. Execute below commands:
  7. # mount /dev/xvdf1 /mnt

# cp /root/.ssh/authorized keys /mnt/root/.ssh/

# unmount /mnt

* 1. Detach the secondary volume from helper instance.
  2. Again, attach the volume back to our recovery instance. Start the instance. Terminate the helper instance.

1. **Can you detach an ebs volume and attach an ebs volume to the ec2 instance?**

Yes, we can detach an EBS volume and attach it to another ec2 instance

Select instance -> actions -> attach or detach EBS -> select req EBS and attach

1. **NAT Gateway**

**NAT Gateway** helps private subnet EC2 instances to get to the internet and NAT gateway need to be configured in public subnet by allocating an elastic IP because NAT gateway only works with elastic IPs.

**Example**: Let's say we have private subnet with EC2 instances and they need to reach internet for updates. Then you need to configure a NAT gateway in the public subnet with elastic IP and update the route table of the private subnet destination as 0.0.0.0/0 and target as NAT ID. And one thing though NAT gateway is configured in public subnet the public subnet ec2 instances no need to go through NAT to connect to the internet because the public subnet route table has an internet gateway entry.

1. **I have critical data on the ec2 instance but .pem key is lost then how to connect to the instance**

First create a new instance with the same configurations as the previous instance for which we lost the pem key and the take a snapshot of the EBS volume of the .pem key lost instance and then create a volume out of the taken snapshot and then attach this volume to the newly created instance. By this we can get access to the data of the instance for which we lost the .pem key

1. **S3 Storage Classes?**

• For frequently accessed objects **S3 STANDARD** **General purpose**

* + - For Infrequently accessed objects **S3 STANDARD\_IA** and **S3 ONE\_ZONE** (One AZ only)
    - For both frequently accessed objects and infrequently accessed objects **INTELLIGENT\_TIERING**

Suitable for 128kb objects and larger. It is most cost-effective storage tier.

* + - Archiving Classes 1. **S3 Glacier Instant Retrieval** used for archives where portions of data need to be retrieved Millisecond retrieval, great for data accessed once a quarter minutes.

Min storage duration is 90 days and you are charges for 90 days even if you move data before 90 days

2. **Amazon S3 Glacier Flexible Retrieval** : where portions of data need to be retrieved in minutes , standard 3 to 5 hrs , bulk 5 to 12 hrs.

Min storage duration is 90 days.

2. **GLACIER\_DEEP\_ARCHIVE**: used for archiving rarely accessed data and it is the lowest cost

storage option becasue less trasactions in this storage.

min storage duration is 180 days and default retrieval time are 12 hours and you

are charged for 180 days even if you move your data before 180 days

1. **CORS- Cross Origin Resource Sharing**

Is a way by which client web application (webpage or a domain) can request resources from another webpage (different Domain)

CORS can be used to allow web application to access resources on your S3(bucket/resources) As an example:

You can host web fonts on your S3 bucket, then configure your bucket to allow CORS requests for web fonts. Other domains will issue CORS requests to load web fonts from s3 bucket.

1. **Data Points IN CW**

Data point is a metric value at a point in time. Data points represents value of metrics over time. If you use 1 minute as an aggregation period then there will be one data point entry for one minute.

1. **A record**

*Address record* is a Host Record which maps domain name to IP address

1. **Alias Record** 
   * + Specific to Route 53
     + You can use alias record to create route 53 records and route queries to aws services.
     + If an alias record points to aws services like ELB, CF distribution, S3 Bucket.
     + Alias record can save you time because route 53 automatically recognize changes in records that alias record refers to.
     + Ex: If an IP address of an ELB changes route 53 automatically reflect those changes in DNS answers for this domain without any changes to the Hosted zone that contains records for the domain.

The word "proxy" describes about one server acting on the behalf of another computer.

1. **AAAA Record :** IPv6 to hostname
2. **CNAME Record:** Hostname to hostname
3. **Static website hosting on S3**

-create a bucket

* keep it ready your index.html file

-Enable the bucket for Static website hosting

Bucket -> properties -> static website hosting -> edit -> enable -> index.html (file) -> save

-write down the website endpoint URL

* Enable public access using Bucket Policies
* test it with the endpoint url

1. **.Golden AMI:** Install your applications, OS dependencies etc. beforehand and launch your

EC2 instance from the Golden AMI

1. **What is the use case of VPC Peering?**

It helps in transfer of data by creating a file sharing network accross accounts To allow other VPCs to access resources you have in your vpc.

1. **Why VPC peering connection is private?**

By communicating privately, it improves security and gives an advantage to separate traffic from the public internet.

1. **We have an ec2 instance with an application running on it and you need to auto scale it based on CPU utilization. Describe the entire flow?** 
   1. We can create an image of the ec2 instance that which is running application.

Select EC2 --> Actions --> Image and Templates --> create Image

* 1. EC2 --> Auto Scaling --> launch Template--> create Launch Template --> select AMI --> instance type --> select storage --> configure Security Groups --> choose key pair --> create Launch Template.
  2. Auto Scaling Group --> create --> Choose your Launch configuration --> next --> network --> choose VPC --> choose AZ’s and subnets --> NEXT --> Choose Load Balancer or create one --> create --> internet-facing (scheme) --> select TG --> HEALTH Checks --> ec2 and ELB HC --> give grace period 300 seconds --> configure Monitoring --> Next -->
  3. Select GROUP SIZE --> Desired, min, max --> Scaling Policies

Add Notification --> you can create an SNS topic --> Done

1. **Types of Scalability:** 
   1. Vertical Scaling 2. Horizontal Scaliing

1. Vertical Scaling: It means increasing the size of instance

Ex: If your application runs on t2.micro scaling it to t2.large is known as vertical scaling.

It is common for non-distributed system, such as database RDS and ElastiCache can scale vertically.

2. Horizontal Scaling : Increasing number of ec2 instances.

HS implies distributed systems. very common for web applications.

High Availability and HS goes hand in hand by running your instances in multiple AZ’s to survive AZ loss.

1. **Types of Load Balancers.**

***Load Balancer:***

Load Balancer forward traffic to different backend instances in multiple AZ’s , by exposing a -single point of access to your application.

-It forwards equal amount of traffic to all the instances in different AZ’s (Cross-Zone Load Balancing)

-It can handle failures of instances by doing regular health checks to your instances.

-Can enable Stickiness with cookies.

-Provides HA

-Separate Public traffic from Private traffic.

***TYPES*:**

**CLB, ALB, NLB, GWLB**

**CLB:** HTTP, HTTPS, TCP, SSL (secure TCP)

**ALB**: HTTP, HTTPS, WebSocket

**NLB**: TCP, TLS (secure TCP), UDP

**GWLB**: Operates at layer 3 (Network layer) – IP Protocol

**Classical Load Balancer:**

-Supports TCP (Layer 4), HTTP & HTTPS (Layer 7)

-Health checks are TCP or HTTP based

-Fixed hostname

**Application Load Balancer:**

-Application load balancers is Layer 7 (HTTP)

-Can load balance to multiple http applications across multiple target groups.

-Health checks are at TG level -Routing:

Routing based on path, hostnames, query strings and headers

-ALB is best fit for micro services and container-based applications.

**Network Load Balancer:**

-Forward TCP & UDP traffic to your instances

-Handle millions of requests per seconds

-Less latency ~100 ms (vs 400 ms for ALB)

-NLB has one static IP per AZ, and supports assigning Elastic IP (helpful for whitelisting specific IP)

-NLB are used for extreme performance, TCP or UDP traffic.

* NLB can have targets as EC2 instance IDs and IP address.
* NLB can be used with ASG if you registered targets by instance ID.
* TARGETS are referred by IP address and Instance ID.

**Gateway Load Balancer:**

-Deploy, scale, and manage a fleet of 3rd party network virtual appliances in aws -Can send traffic to any ip protocols -Combines the following functions:

* + Transparent Network Gateway – single entry/exit for all traffic
  + Load Balancer – distributes traffic to your virtual appliance -Uses the GENEVE protocol on port 6081

**Why use a load balancer?**

•Spread load across multiple downstream instances

•Expose a single point of access (DNS) to your application

•Seamlessly handle failures of downstream instances

•Do regular health checks to your instances

•Provide SSL termination (HTTPS) for your websites

•Enforce stickiness with cookies

•High availability across zones

•Separate public traffic from private traffic

1. **On what basis you choose your instance.**

We choose on the basis of workloads, pricing,

**ON-Demand Instances:**

--If we require instance for short-term and for un-interrupted workloads, where you can’t predict how the application will behave.

--per second billing is incurred on us -- No long-term commitment.

**Reserved Instances:**

-- If we require instances for long term we go with reserved instances that means for a period of 1 to 3 years.

-- Reserve a specific type of instance

-- For steady state usage applications like databases  **Convertible Reserved Instance:** -- here we can change instance type

**Scheduled Reserved Instance:**

-- launch within the time you reserve.

-- when you require for a fraction of day / week /month

-- commitment for 1 year only  **Spot Instances:**

-- Can get a better discount compared to On-Demand instances

-- You need to bid for spot instances

-- If your bid price exceeds the spot price you will lose the instance -- suitable for workloads with start and end time.

-- Not suitable for critical jobs or databases.

**Dedicated Hosts:** here we are renting an entire server in a data centre

allocated for 3-year reservation

More expensive

1. **What are life-cycle hooks?**

Lifecycle hooks are used for putting wait time before any lifecycle action i.e., launching or terminating an instance happens. The purpose of this wait time, can be anything from extracting log files before terminating an instance or installing the necessary software in an instance before launching it.

1. **RDS**

RDS is a relational database service, it follows a structure scheme by defining tables, columns, indexes Supported Relational Engine types are:

MS SQL Server

Oracle DB

MySQL

PostgreSQL

MariaDB

AWS Aurora

RDS has a multi-AZ option that which deploys a RDS primary instance in one AZ which is for reads and writes and a stand-by instance in another AZ. We cannot read and write to stand-by instance. But there is a synchronous data replication between the primary and stand-by instance. And we can have upto 5 Read-Replicas for a RDS to balance read workloads.

===>Failover may be triggered when

--Loss of primary AZ or primary DB instance failure

-- Loss of network connectivity

--Compute or storage failover

--Automated Backups

--Manual Snapshots

1. **Connect 2 ec2 machines:**

EC2-A===> need to connect to EC2-B===> then update the security group of EC2-A on the outbound rules add a rule

===>

custom TCP | protocol | TCP | port Range : 8080 | destination: SG-of-ec2-B ==> add

On the EC2-B ==> update the INBOUND rules on security group ===> custom TCP | TCP| 8080 | **source: SG**-of-ec2-A | Test connectivity using commands

telnet $destinationIP $PORT curl -v telnet://$IP:$PORT

1. **Use cases of NLB?**

When you need extreme performance to handle millions of request.

When you need to support high-volume inbound TCP request.

When you need to support static or elastic IPs

Need to support an IP address or an IP target outside of the VPC.

Want to support and monitor multiple services running on ports on an EC2 instance.

1. **TRANSIT GATEWAY -TGW**

A transit gateway is a network transit hub that you can use it to connect multiple VPCs and also to on –premise networks. In this connection VPCs are allowed to communicate with each other.

Why we use transit gateway means suppose let's say we have 4 vpcs and we need all vpcs to communicate with each other. Therefore, we need 6 peering connections. What if there are 20 or 30 vpcs at that time you should go with transit gateway.

It uses BGP to dynamically propagate routes.

1. **VPC Endpoints.**

**Without VPC endpoints** EC2 instances/Apps if they want access to aws services they need to go over the internet, VPN connection, NAT Gateway or need public IP addresses. Traffic Flow looks like:

Private subnet ---> Implied Router ---> NAT gateway (everything set) ---> (again) Implied Router ---> IGW ---> Internet ---> to respective AWS service.

**With VPC Endpoints** we can get higher performance and more secure connections via its private ip address to aws services without the need to go over the internet.VPC endpoint is powered by private link.

Private subnet ---> Implied Router ---> Route Table has an entry (prefix less ID for aws services) ---> vpc endpoint ID ---> VPC endpoint ---> resp. AWS service.

There are two types of endpoints. INTERFACE and GATEWAY endpoints.

**Interface endpoint supported services are** : API gateway, CloudFormation, CW, CW events/logs, EC2 APi, KMS, ELB, SNS **Gateway Endpoint supported services are**: Amazon S3 and DynamoDB.

1. **Public subnet vs private subnet**

|  |  |
| --- | --- |
| **Public Subnet** | **Private Subnet** |
| If a subnet is associated wiht an internet gateway so that the subnet route table has igw entry then it is a public subnet. | If a subnet route table does not have an igw entry in its route table then it is a private subnet. |
| It can communicate with outside world i.e., internet | It cannot communicate with outside world and internet cannot reach the subnet |

1. **Public IP vs Private IP**   
   **Public IP:**   
   - Public IP means the machine can be identified on the internet (WWW)

- Must be unique across the whole web (not two machines can have the same public IP)  
Class A – 1 to 126 – here 127 is for loopback address.  
Class B – 128 to 191

Class C – 193 to 223

Class D – 224 to 239

Classs E - 240 to 255  
**Private IP:**

It is for local usage , with in your organization.It is of dedicated range.

10.0.0.0 – 10.255.255.255

172.16.0.0 – 172.31.255.255

192.168.0.0 – 192.168.255.255

1. **Security Group Vs NACL**

|  |  |
| --- | --- |
| **Security Group** | **NACL** |
| Operates at instance level | Operates at subnet level |
| Supports allow rules only | Supports allow and deny rules |
| It is STATEFUL: Return traffic is automatically allowed, regardless of any rules | It is STATELESS: Return traffic is explicitly allowed by rules |
| We evaluate all rules whether to allow traffic or not | We process rules in numbers order while deciding whether to allow traffic |
| Applies to an instance for which the security group is associated with | Applies to all instances in the subnets it is associated with. |

1. **Elastic IP:**

When you stop and start your EC2 instance, it can change its public IP. In order to have a fixed IP for your instance you can assign an elastic IP to your instance.

1. **Elastic BeanStalk:**

•Automatically handles capacity provisioning, load balancing, scaling, application health monitoring, instance configuration, … Just the application code is the responsibility of the developer

1. **Suppose lets say of the user deleted an ec2 instance in your aws account how can you find out who did it.**  
   The main purpose ofCloudTrial is governance, compliance,auditing capabilities for your AWS account.  
     
   - It Logs all the API calls made to AWS services including who made changes and when those changes are made in your aws account.  
   - Provides info about unsual and unexpected activities or changes in your AWS environments such as unauthorized access or change.
2. **AWS Guard Duty:**  
   **Amazon GuardDuty Overview**

**Purpose**

* Intelligent Threat Discovery
* Continuously monitors and workloads Protects AWS Accounts

**Key Features**

* **Machine Learning Algorithms**
  + Performs anomaly detection
  + Uses third-party data for threat detection
* **Easy to Enable**
  + One-click setup
  + 30-day free trial
  + No software installation required

**Example**: We enabled Guard Duty and it started monitoring aws account resources and one of the ec2 instance making outbound connection to a known malicious IP then Guard Duty will generate a findings with detailed info about the activity à and the finding includes instance ID, type of threat detected, and severity of it also time of detection.

1. **AWS Organizations Overview:**  
   It is a global service which helps in managing multiple aws accounts.  
   Here we create a main account in the organization and all the member accounts will join the organization.  
   and also accounts can only belong to one organization.  
     
   Centralized management for multiple aws accounts.
2. **What is an EC2? Describe your experience working with it ?**

EC2 means Elastic Compute Cloud, where we can rent virtual machines according to our requirement based on compute power i.e no of vCPU required, based on RAM requirements.  
There are different types of instances classes i.e. T2, m5 etc.  
- We choose the instance based on the workloads and pricing.  
Ex: If we cannot predict how oue application behaves we can choose m5 type of instances.

***ON-Demand Instances:***

-- If we require instance for *short-term and for un-interrupted workloads*, where you can’t predict how the application will behave.

-- per second billing is incurred on us -- No long-term commitment.

***Reserved Instances:***

-- If we require instances *for long term we go with reserved instances* that means for a period of 1 to 3 years.

-- Reserve a specific type of instance .

-- For steady state usage applications like databases

***Convertible Reserved Instance***: -- here we can change instance type

***Scheduled Reserved Instance:***

-- launch within the time you reserve.

-- when you require for a fraction of day / week /month

-- commitment for 1 year only

***Spot Instances:***

-- Can get a better discount compared to On-Demand instances

-- You need to bid for spot instances

-- If your bid price exceeds the spot price you will lose the instance -- suitable for workloads with start and end time.

-- Not suitable for critical jobs or databases.

***Dedicated Hosts***: here we are renting an entire server in a data centre

allocated for 3-year reservation More expensive

**46. Explain what is a T2 instance in AWS ?**

- T2 instance is a typer of Virtual server in EC2. balance of compute, memory, and network resources and are ideal for workloads that don't require consistent, high levels of CPU performance  
 - T2 instances come in various sizes, such as t2.nano, t2.micro, t2.small, t2.medium, t2.large, t2.xlarge, and t2.2xlarge. Each size offers different amounts of CPU, memory, and network performance.

**47. What is the role of AWS CloudTrial?**   
 CloudTrial logs all the API calls activity in your AWS account which are made through AWS console, cli or sdk.  
 It provides logs for user activity, resource modifications and service interaction with in aws.

By default CloudTrial stores events for 90.

Ex: Lets say some one deleted an object in S3 bucket we can check in cloud trial logs to check who deleted an object, when it is deleted.  
1

**48. What is AWS Lambda?How would you manage environment provided by clients?**

Lambda is a serverless offering provided by AWS. It helps to run code by responding to a event. Lambda functions run automatically when triggered.  
We get 100 lamba concurrency this helps lambda to scale automatically.  
  
**49. How many elastic IP’s does AWS allocate by default?**

By default we get 5 Elastic IP’s we can raise a support request to increase the limit.

**50. What platform does AWS provide- Iaas, SaaS, PaaS ?**

Aws provides all the three like IaaS, SaaS, PaaS.

IaaS – VPC, EC2, EBS  
PaaS – RDS,Elastic Beanstalk, Lambda

SaaS – Workspace, Chime, QuickSight.

**51. How is data stored in S3 Buckets?**  
 In S3 buckets data is **stored as objects** and also, we can enable versioning on bucket level to maintain the versioning of the objects uploaded to S3.  
- We can assign bucket policies to control its access   
- Also we can assign IAM policies to users or roles to control what actions they perform in S3.  
- Suppose lets we we want to upload files to S3 bucket we can assign a role to the Ec2 instance to give permission to upload.

**52. What are trigger in AWS Lambda?**

Triggers are events that cause your Lambda function to execute. These triggers can come from various AWS services.  
Examples:  
\* In response to the SNS notification lambda functions can trigger to perform action req.  
\* When data is uploaded to the S3 we can set up lambda trigger to process the data.

**53. What are events in AWS?**   
Events are the changes that happen on any of the AWS resources and that change can trigger an action to perform.  
Ex:   
 **Amazon S3 Events:**

* + Event Types: Object creation (e.g., PUT, POST), object removal (e.g., DELETE).
  + Use Case: Trigger a Lambda function to process files when they are uploaded to an S3 bucket.

**Amazon EC2 Events**:

* + **Event Types**: Instance state changes (e.g., start, stop, terminate).
  + **Use Case**: Trigger notifications or automated actions when EC2 instances change state.

**54. Is an internet gateway required for VPC peering connection?**

No, an Internet Gateway is not required for a VPC peering connection in AWS. VPC peering allows you to connect two Virtual Private Clouds (VPCs) so that they can communicate with each other using private IP addresses.

**55. What is an Eastic BeanStalk? Can it be customized?**

Elastic Beanstalk is a Platform as a Service (PaaS) service.  
Where we only need to deploy the application version rest of the underlying infrastructure is completely managed by beanstalk.  
How It works?  
1. **Create an Application**: This is your starting point in Elastic Beanstalk, where you define your app and its components.   
2. **Upload a Version**: Upload your application code (e.g., a ZIP file with the code) and associate it with an application version.   
3. **Launch an Environment**: Deploy your application to a specific environment (e.g., dev, test, or prod).   
4. **Manage the Environment Lifecycle**: Once deployed, you can manage the lifecycle of your application (monitor health, scale, etc.).   
5. **Iterate with New Versions:** Upload new versions of your application to update the environment.

**Elastic BeanStalk can be customized:**  
- We can use configuration files to customize the env settings.  
- Also we can use custom AMI accord. To the requirements.  
- Set env variables to configure application settings dynamically  
- Choose instance type w.r.t to the performance need.

**53. Amazon SES:**

- SES is a fully managed service for sending emails.  
- Helps in securely delivering the emails.  
- We can send emails are receive replies.  
use cases:  
- Transactional emails  
- Marketing emails  
  
**55. Amazon SNS:**

- SNS is a fully managed push notification service.  
- Where we can send individual or Fan-out messages to subscribers endoints.  
Publisher -à create a topic -à Publisher send a message to topic -à SNS will send the message to subscribers.  
 - We can send notifications to registered mobile no’s.  
 - Message send as text email to the registered email addresses.

56.

|  |  |
| --- | --- |
| DynamoDB | 8000 |
| MySql | 3306 |
| PostgreSQL | 5432 |

**57. Internet Gateway?**  
It is the gateway through which VPC communicates with the internet and with other AWS services.  
That means if the traffic is coming from outside to the VPC from it will pass through the internet gateway and it’s going to send it to VPC.  
It performs NAT b/w private and public subnets.

**58. How many routing policies are available in Route53?**  
 We have

**Simple Routing Policy:** Routes traffic to a single resource without any special routing logic.

**Weighted Routing Policy:** Distributes traffic across multiple resources based on specified weights.

**Failover Routing Policy:** Routes traffic to a primary resource unless it is unavailable, then routes to a secondary resource.

**Latency-based Routing Policy**: Routes traffic to the resource with the lowest latency for the user.

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

**Multi-value Answer Routing Policy**: Returns multiple healthy resources in response to DNS queries.

**Geoproximity Routing Policy:** Routes traffic based on the geographic location of resources and users, with the ability to shift traffic by specifying a bias.

**58.Why do you dont want to secure your application using NAT gateway in publix subnet ?**

NAT Gateways are not designed to *secure* applications — their primary role is to *enable outbound*  *internet access* for private resources, not to provide inbound protection or act as firewalls.  
 The NAT Gateway *only allows outbound connections initiated from private instances*. No one can start a connection from the internet to the instances behind the NAT

59. You have an EC2 instance which do not have public IP attached to it but still the internet routes are happening though the internet gateway can i able to reach the ec2 instance over the internet?  
 No, **you cannot reach an EC2 instance over the internet** if:

* It **does not have a public IP or Elastic IP**, and
* There is **no other public-facing resource (like a Load Balancer or Bastion Host)** routing traffic to it —  
   **even if** the **route table points to an Internet Gateway (IGW)**.  
    
  **Public IP is required for direct access:**
* For any **inbound traffic from the internet**, the EC2 instance must have:
  + A **public IP or Elastic IP**, and
  + Be in a subnet with a **route to an IGW**.
* Without a public IP, **the IGW has no way to map the response traffic back to the instance**.

**60. Lets say now we have public Ip to the ec2 instance is it good idea to expose our ec2 instance directly to the internet?**  
No, it is not a good idea   
We need to register the ec2 instances as a targets to an internet facing load balancer.  
So that we get a single point of access to out ec2 instances that are hosting applications.  
And later if we want to load balance the ec2 we can use ASG to scale out and scale in EC2 instances.

**61. Is there any easy way to give your end user instead of a load balancer DNS name.**  
Ans. Idealy we should not provide the load balancer DNS name instead register a domain name from a domain register basically we use GoDaddy as a domain register. And map the load balancer DNS name as a Cname record then we can provide the DNS name to the end user.

**63. Lets say you have a website with some domain** [**www.example.com**](https://www.example.com/) **and we have /payments and another path /purchase which load balancer will you use ?**

Basically to route traffic based on t=path we can go with application load balancer. ALB can route requests to different target groups based on the **URL path**.   
 **ALB Configuration:**

|  |  |
| --- | --- |
| **Rule** | **Target Group** |
| Path is /payments\* | payment-target |
| Path is /purchase\* | purchase-target |

**64. Now How can you block a range of IP addresses?**  
**1. AWS WAF (Web Application Firewall) — Best for ALB/API Gateway/CloudFront**

**If you're using an Application Load Balancer (ALB), API Gateway, or CloudFront**, you can use **AWS WAF** to block IPs or CIDR ranges.

#### ***Steps:***

1. Go to **AWS WAF** in the console.
2. Create a **Web ACL**.
3. Add a rule:
   1. Type: **IP set**
   2. Add your **blocked CIDR ranges** (e.g., 203.0.113.0/24)
   3. Action: **Block**
4. Associate the Web ACL with your **ALB**, **CloudFront distribution**, or **API Gateway**.

### **2. Security Groups — Limited to Specific IPs (Not Ideal for Blocking Ranges)**

* **Security groups are stateful firewalls**, but they are **allow-lists**, not block-lists.
* You can **allow only specific IPs**, but you **cannot explicitly "deny"** an IP or CIDR.

**So — security groups can't block a range**, but you can **omit** the range you want to block when defining "allowed" IPs.

### **3. Network ACLs (NACLs) — Good for VPC-level Blocking**

If you want to block traffic **before it even hits your instance or load balancer**, use **Network ACLs**.

#### ***Steps:***

1. Go to **VPC > Network ACLs**.
2. Choose the ACL associated with your subnet.
3. Add an **inbound rule**:
   1. Rule number: e.g., 100
   2. Type: All Traffic or specific port
   3. Protocol: All or TCP
   4. Source: e.g., 203.0.113.0/24
   5. Action: **DENY**
4. Ensure there's a lower-priority ALLOW for legitimate traffic.

NACLs are **stateless**, so you must configure both **inbound and outbound** rules for complete control.  
  
66. Let say you have ec2 instances in multiple availability zones, Now how can you make your backend storage consistant.  
In this case we can use EFS for a distributed storage accross ec2 instances in multiple AZ’s.   
 Why not EBS? --> we cannot EBS because it is AZ specific.

67.Is there any benefit of using EBS over EFS ?  
Yes, there **are benefits of using EBS over EFS**, but it depends on **your specific use case**. Both are AWS-managed storage services, but they are **designed for different purposes**.  
  
**Key Differences: EBS vs EFS**

|  |  |  |
| --- | --- | --- |
| **Feature** | **EBS (Elastic Block Store)** | **EFS (Elastic File System)** |
| Type | Block storage (like a hard disk) | File storage (like a shared NFS) |
| Mounting | Only one EC2 instance at a time (unless using EBS Multi-Attach with some limits) | Multiple EC2s across AZs simultaneously |
| Performance | Higher IOPS and throughput options | Lower IOPS compared to EBS (unless provisioned) |
| Use Case | Databases, OS volumes, low-latency workloads | Shared file access, web content, logs |
| Pricing | Pay for provisioned size | Pay for actual storage used |
| Consistency | Strong, immediate | Strong |
| Access Across AZs | ❌ (AZ-specific) | ✅ Yes (cross-AZ mount targets) |

**68. Let's say you have an IAM account you gave administrator permissions to that account.Now you don’t want that admin user to create another user? How can you achieve it?**

## **Solution: Use an IAM Permissions Boundary or SCP (Service Control Policy)**

### **✅ Option 1: Use a Permissions Boundary**

A **permissions boundary** is an advanced IAM feature that **limits what permissions a user or role can have**, even if they have AdministratorAccess.

#### ***🔧 Steps:***

1. Create a **permissions boundary policy** that explicitly **denies IAM user creation**:

json

CopyEdit

{  
 "Version": "2012-10-17",  
 "Statement": [  
 {  
 "Sid": "DenyIAMUserActions",  
 "Effect": "Deny",  
 "Action": [  
 "iam:CreateUser",  
 "iam:DeleteUser",  
 "iam:UpdateUser",  
 "iam:PutUserPolicy",  
 "iam:AttachUserPolicy"  
 ],  
 "Resource": "\*"  
 }  
 ]  
}

1. Attach this policy as a **permissions boundary** when creating or updating the admin IAM user or role.

📌 This ensures:

Even though the user has AdministratorAccess, IAM will **evaluate the permissions boundary**, and **deny** any IAM user creation or modification.

### **✅ Option 2: Use Service Control Policies (SCPs) — If You Use AWS Organizations**

If you're in an **AWS Organizations** setup, you can use an **SCP** to restrict IAM actions across accounts.

#### ***Example SCP:***

json

CopyEdit

{  
 "Version": "2012-10-17",  
 "Statement": [  
 {  
 "Sid": "DenyIAMUserManagement",  
 "Effect": "Deny",  
 "Action": [  
 "iam:CreateUser",  
 "iam:DeleteUser",  
 "iam:UpdateUser"  
 ],  
 "Resource": "\*"  
 }  
 ]  
}

Apply this SCP to the **OU or account** that the admin user is in. It will override any IAM policy, including AdministratorAccess.  
  
69.Lets say you 2 servers trying to ssh its failing. You are getting an error connection refused? How will you troubleshoot?  
 Firstly check are you using correct username and pem file and validate the remote server security group is allowed the inbound traffic from this instance and again verify port 22 is in listening state using (netstat –tunlp)   
  
70. You hosted a website but the users are getting 401 error ?  
 401 means – unauthorised   
That means authentication is failed on the application. The user is not authorised.  
 When it says unauthorised is the server accepted the authentication.  
  
Yes the server accepted the authentication and completed the authentication.That means it authenticated and validated the request.  
  
**HTTP Error Codes (Web/API-related)**

|  |  |  |
| --- | --- | --- |
| **Code** | **Meaning** | **How to Fix** |
| 400 | Bad Request | Check request syntax, missing headers, or malformed parameters. |
| 401 | Unauthorized | Missing/invalid authentication (token, credentials). |
| 403 | Forbidden | Authenticated but **not authorized** — check IAM or permissions. |
| 404 | Not Found | Endpoint or resource doesn’t exist. |
| 500 | Internal Server Error | Server crash — check backend logs. |
| 502 | Bad Gateway | Issue with proxy/load balancer to backend. |
| 503 | Service Unavailable | Service is down or overloaded — retry with backoff. |
| 504 | Gateway Timeout | Backend took too long to respond. |

71) If your blue or green deployment in Kubernetes fails, they knew that is a green version and it doesn’t respond after switching traffic.how would you troubleshoot and rollback for this issue?

**1. Check Green Pod Health**  
kubectl get pods -l app=your-app,version=green  
kubectl describe pod <green-pod-name>  
kubectl logs <green-pod-name>  
 Look for:

* CrashLoopBackOff or ImagePull errors
* App-specific errors in logs (e.g., port issues, DB connectivity, config errors)

**2.Verify Readiness and Liveness Probes**  
Check if the new version fails to respond to health checks.

kubectl get endpoints your-service  
kubectl describe pod <green-pod-name>

* If readiness probe fails, the pod won’t receive traffic.
* If liveness fails, the pod keeps restarting.

### **3. Check Service Routing**

Confirm if the Service actually points to green version:

bash

CopyEdit

kubectl get svc your-service -o yaml

Make sure the selector has version: green and matches the green pods.  
**4. Application-Level Check**

If green pods are running but still unresponsive:

* Curl the service IP or endpoint directly from another pod (kubectl exec into a debug pod).
* Check environment variables, secrets, or configmaps used by the green deployment.

**72) Lets say your deployment pipeline runs fine util it reaches the deployment stage , so there was like an error it says kubectl not found or unauthorized ? What would be the issue?**

The CI/CD environment (e.g., GitLab CI, Jenkins, etc.) does not have kubectl installed or it's not in the system's PATH.  
**Fix:**

* Install kubectl in the pipeline job before using it.

**Unauthorized (when running kubectl commands)**

#### ***❌ Root Cause:***

* kubectl is installed, **but it cannot authenticate with the Kubernetes cluster**.
* Possible reasons:
  + KUBECONFIG file is missing or invalid.
  + Service account or token used is expired or has no permissions.
  + The pipeline is **not correctly configured with credentials**.

73) Lets says your docker image is buit in jenkins but when pushing to AWS ECR we face issue like 403? How to solve this ?

This issue might arise when there is authentication is successful but not authorised with ECR, we need to check the IAM permissions associated with the user and verify the credentials expiration as well.

74) You changed a vpc subnet configuration and running terraform apply for it. Then it tries to destroy existing EKS nodes How do you handle this safely?  
Firstly before executing terraform apply.  
- We should execute terraform plan to see what execution plan it is created for this operation, what are the resources marked for deletion.  
Then use a life cycle block to prevent the destroy operation on the eks node  
resource "aws\_eks\_node\_group" "example" { ...

lifecycle {

prevent\_destroy = true

}

}  
  
This will **block terraform apply** if it tries to destroy the node group.

* 75) Key difference between terraform format and terraform validate ?  
  **terraform fmt** command scans your current directory for configuration files. You can also provide a target argument to tell terraform fmt to scan:
* A directory
* A specific file  
  command formats Terraform configuration file contents so that it matches the canonical format and style.

**Terraform validate:** validates the terraform configuration syntax  
  
76) How do you describe the aws instance using aws cli  
aws ec2 describe-instances --instance-ids i-xxxxxxxxxxxxxxxxx  
  
77) Key differences between NAT-Gateway and Internet Gateway ?

|  |  |  |
| --- | --- | --- |
| **Feature** | **NAT Gateway** | **Internet Gateway** |
| **Purpose** | Allows **private subnets** to access the internet | Connects **public subnets** directly to the internet |
| **Traffic Direction** | **Outbound only** (from private instances to internet) | **Both inbound and outbound** |
| **Instance IP Needed?** | No public IP on instance (uses NAT's IP) | Instance needs a **public IP or Elastic IP** |
| **Use Case** | Private EC2 → Internet (e.g., updates, package installs) | Hosting websites, open SSH, public APIs on EC2 |
| **Subnet Location** | Deployed in a **public subnet** | Attached to the **VPC** |
| **Return Traffic** | Automatically handled | Requires route + public IP on instance |
| **Security Control** | Use security groups, NACLs, and route tables | Same |
| **Cost** | **Billed** (per hour + per GB) | **Free** |

77) Can a subnet span across multiple availability zones ?  
No, a subnet in AWS **cannot span across multiple Availability Zones (AZs)**.  
 Each **subnet is strictly tied to a single Availability Zone** within a VPC.

78) What is the difference between S3 and EBS ?  
S3 is an object storage , where we can store n no of files.  
There are different storage classes available in S3 like S3 standard , S3 standard IA, S3 one zone IA, Intelligent tiering etc.  
WE can create life cycle rules to transitions objects to less storage cost classes.  
  
  
EBS:  
EBS is an elastic block store that is attached to an EC2 instance to utilize it.  
Here we define the storage requirements accordingly to create an EBS   
In EBS we have different types like gp2, gp3, io1, io2 , st1, st2 HDD   
  
Signed url in AWS

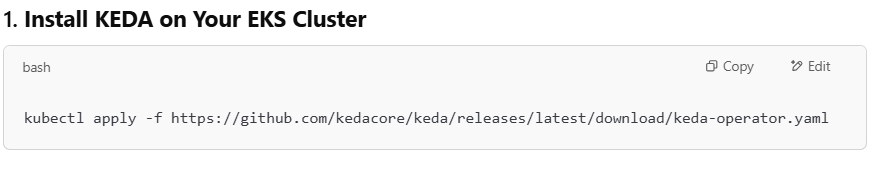
With signed URL we can give temporary access to an object in S3 without making it publicly accessible.

Generally for that specific object we can set an expiration time and can create a signed url to share it for upload, download.  
**\* Use case:** Share a private file securely without making it public.  
**\* How it works:** The URL contains a signature generated using your AWS credentials and includes an expiration timestamp. After expiration, the URL becomes invalid.  
\* The maximum expiration is **7 days**  
  
What are regions in AWS ?  
AWS has regions all over the world.  
In general a region is a cluster of data centers  
most of the aws services are region scoped.  
  
In EKS I want to scale my application using some custom metrics not like CPU and memory? How can we do ?

To scale your application in **EKS (Amazon Elastic Kubernetes Service)** using **custom metrics** (not just CPU or memory), the best approach is to use **KEDA (Kubernetes-based Event-Driven Autoscaler)**.

KEDA allows you to scale Kubernetes workloads (like Deployments, Jobs, etc.) based on **custom or external metrics** such as:

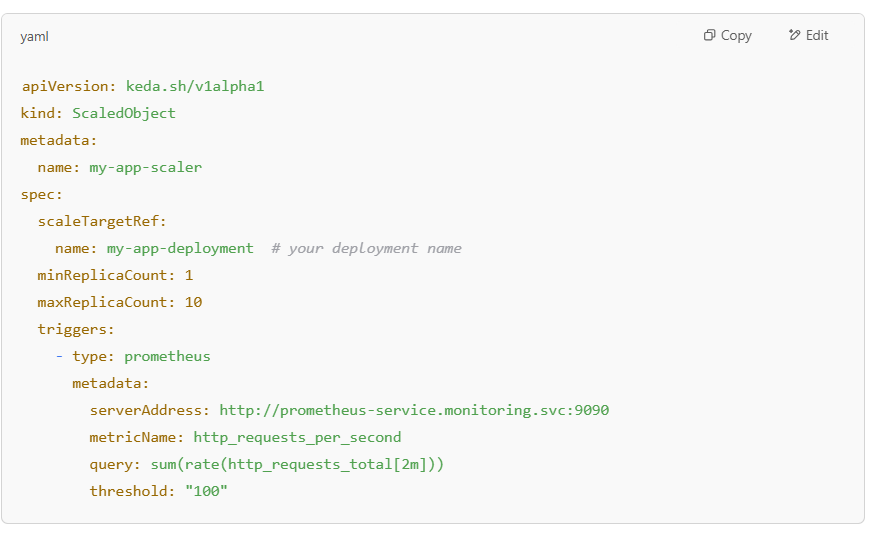
* Number of messages in SQS/SNS
* HTTP queue depth
* Prometheus queries
* Kafka lag

Steps to Scale EKS Using Custom Metrics with KEDA  


### **2. Define a ScaledObject**

A ScaledObject tells KEDA how to scale a deployment using a custom metric.

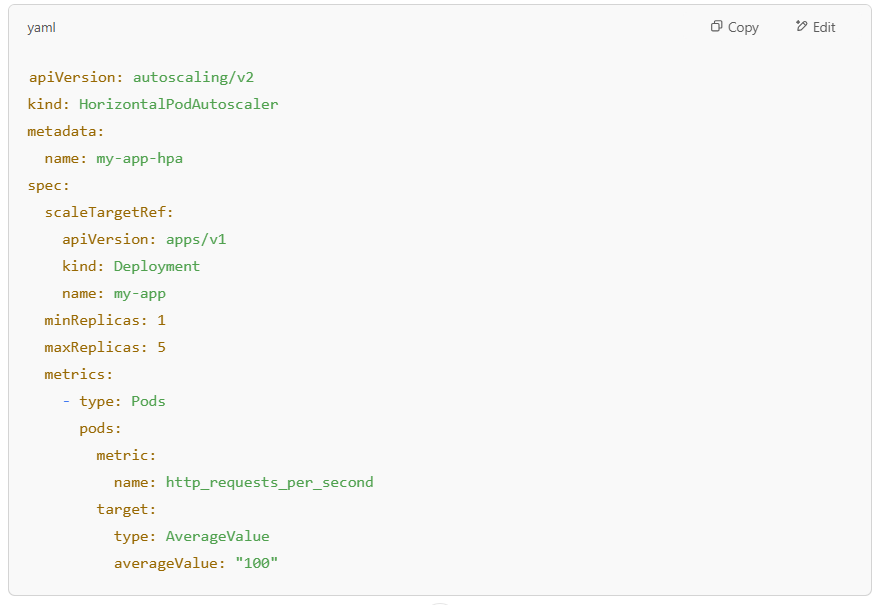
**Example**: Scale based on a Prometheus query (custom metric)



### **3. Install Metrics Adapter (if not using KEDA)**

If you choose not to use KEDA and want to use **Horizontal Pod Autoscaler (HPA)** with custom metrics:

* You must deploy **Prometheus Adapter** or a **custom metrics API**.
* Configure your HPA to use custom metrics like:



## **When to Use What?**

|  |  |
| --- | --- |
| **Tool** | **Use When** |
| **KEDA** | Event-based scaling from external/custom sources like SQS, Kafka, Prometheus, etc. |
| **HPA + Custom Metrics Adapter** | You have in-cluster metrics and want more control but more setup effort |

Can you design a three tier architecture? Just go through what all you think of to put in place? Make sure the application is secure ? We do not need high availability ?   
  
I would design the architecture using a **VPC** with:

* A **public subnet** for an **internet-facing ALB**
* A **private subnet** for **EC2 instances** (app tier)
* Another **private subnet** for **RDS** (database)

The EC2 instances have **no public IPs**. A **NAT Gateway** (in the public subnet) allows them to fetch updates from the internet securely.

I’ll configure **security groups** to:

* Allow **only ALB to access EC2**
* Allow **only EC2 to access RDS**

The **ALB** will be mapped to a **Route 53 domain**, using **HTTPS with an ACM certificate**.  
 For storage needs, I would use **EFS** if the app needs shared volume.

Finally, I’ll use **IAM roles**, **AWS Secrets Manager** for DB creds, and **CloudWatch** for monitoring.  
 **WAF** on the ALB protect against common attacks.

-----------------------------------------------------------------------------------------------------------------------------------  
  
Can you explain what a 3 tier Architecture ? What it is generally?

A **3-tier architecture** is a **software design pattern** that separates an application into **three logical layers (tiers)** — each with a specific responsibility. This separation improves **scalability**, **security**, and **maintainability**.  
  
**Frontend Tier (Presentation Layer)**

* **What it is**: The user interface — how users interact with the application.
* **Responsibilities**:
  + Displaying data
  + Capturing user input
  + Sending requests to middleware/API
* **Examples**:
  + Web pages (HTML/CSS/JavaScript)
  + Mobile apps (iOS/Android)
  + Single Page Applications (React, Angular, Vue)
* **Deployment Example**: Host on Amazon S3 + CloudFront, or in an EC2 instance or container

**Middleware Tier (Application/Logic Layer)**

* **What it is**: The brain of the application — where all business logic and processing happens.
* **Responsibilities**:
  + Processing requests from frontend
  + Validating input
  + Performing calculations
  + Communicating with the backend (database)
* **Examples**:
  + Web/API servers (Node.js, Java Spring Boot, Python Flask/Django, .NET Core)
  + Microservices
  + Serverless functions (like AWS Lambda)
* **Deployment Example**: EC2 instances in a private subnet behind an ALB, or containers in ECS/Fargate, or Lambda functions

### **Backend Tier (Data Layer)**

* **What it is**: The data store — where all application data is stored and retrieved.
* **Responsibilities**:
  + Storing structured/unstructured data
  + Executing database queries
  + Ensuring data consistency and integrity
* **Examples**:
  + Relational databases (MySQL, PostgreSQL, Oracle)
  + NoSQL databases (MongoDB, DynamoDB)
  + File storage (Amazon S3, EFS)

-----------------------------------------------------------------------------------------------------------------------------------

**Let say we are running a webserver like nginx or any other where it will be placed and how users can interact with the end-point ?**   
  
Like we can place the nginx webserver on our EC2 or we can run nginx as a container or pod in eks.  
Here users can interct with the endpoint   
1) user makes request to DOMAIN ex: httpd://www.example.com/app-name

2) DNS resolves the domain name to   
 ALB DNS if we map the DNS as a Cname recored with the ALB DNS

3) Request will reach the ALB.  
 - then it will route traffic to the nginx running on EC2/ container.

4) nginx handles the request  
 Proxies the request to backend (eg: app server/application containers)  
  
request flows like USER ------> ALB ---->NGINX ------> Backend  
Response flows back like: Backend ------> Nginx ----> ALB ----> User

Let say if we have a simple Setup of 1 ec2 where you will place the weserver nginx and how will you expose it ?

Option-1)   
we can place the EC2 in a public subnet with application hosted on it.  
Ensure the subnet route table has a route to internet gateway (igw).  
Associate a elastic IP or Public IP with the EC2 instance.  
Security Goup route to control inbound and outbound traffic.  
DNS Mapping – Register a domain and create an A Record that maps the EC2 public ip/elastic Ip

Option-2)   
We can place the EC2 in private subnet.  
Place an ALB in the public subnet to expose it to frontend.  
Her we need a Bastion host in public subnet to SSH into the private subnet EC2 instance.  
Register the EC2 as a target in the ALB target group   
Public access will be like USER ---> ALB ----> EC2 (private)  
-----------------------------------------------------------------------------------------------------------------------------------  
  
**Lets say your application [Middle-ware] needs to access the RDS but the RDS is in private subnet? How would you grant access to the application to access the RDS as securely as possible?**  
  
 ---> In our application setup we control access of the applciation running an EC2 instance with the help of security groups—like on the inbound of RDS security group – accept traffic from the SG\_of\_EC2 on port 3306/5432 similarly update the outbound rule on the EC2 SG to route traffic to SG\_of\_RDS  
  
---> suppose lete say we have our application on EC2 instance need to connect to RDS in private subnet.  
Then we can assign an IAM ROLE for EC2 instance to connect to RDS database instead of hardcoded credentials.  
Here for RDS – enable IAM database authentication.

Use Secrets Manager   
Store databse credentials in AWS secret Manager.  
Middleware app can retrieve creds at runtime securely via IAM roles access.

Let say we have the same setup.The traffic should not flow through public internet and it should access the S3 bucket   
  
We can use Gateway endpoint for S3.

- A **VPC Endpoint** enables private connections between resources in a VPC and supported AWS services like S3, without using an Internet Gateway, NAT Gateway,  
- Here I will create a Gateway Endpoint for S3 in the same VPC as of our EC2 is present and attach it to the route table that the EC2 is associated with by updating a Prefix less ID in the route table   
  
- This ensures that all the traffic from the EC2 destined to S3 will be routed within the AWS private network without the need to go through the public internet.  
- We need to make sure like required IAM roles is attached to the EC2 to access the specific S3 bucket.  
Optionally we can add a bucket policy that restricts to accept traffic coming through specific VPC

**Have you worked with LAMBDA or any ither serverless services in AWS ?**  
- Basically we use Lambda for the intialing start/stop of our EC2 instances and RDS databases in the off hours.  
By creating a lamba function with an event trigger at specific time schedule to trigget th almba to start/stop servers.

**Lets say I am implementing some data processing module using lambda when I try to import the particular module.The lambda is giving an errror that the particular module is not found? Can you think of any immediate solution?**

"ModuleNotFoundError", the issue is usually that the module is not available in the Lambda runtime environment.  
Immediate Solution: Package the missing module along with your Lambda code and deploy it.

Lets say you need a custom packsage that does not come with the default lambda? How do you install the package   
\* Bundle the Package with Your Lambda Function  
1) **Create a local folder** and move into it:  
mkdir lambda-package && cd lambda-package  
**2) Install the custom package** locally into that directory:  
pip install <package-name> -t .  
3) **Add your Lambda function file** (e.g., lambda\_function.py) into the same directory.  
4) **Zip the entire directory**:  
zip -r lambda-function.zip .  
5) **Upload this ZIP** to your Lambda function via:  
AWS Console or CLI

**Let say we have a lambda function which handles some endpoint.Whenever a user tries to hit a download pdf we have a lambda function that resolve then the PDF.**  
**If multiple users are hitting on download pdf we are giving ,=multiple requests to our lambda function? How will lambda scale and how will it server users ?**   
Example : You have a Lambda function that:

* Handles an endpoint (e.g., GET /download-pdf)
* Dynamically generates/returns a PDF
* Needs to handle **multiple users simultaneously**

AWS Lambda automatically **scales horizontally** by **creating multiple instances** of your function in response to the number of concurrent requests.  
**Each request gets a separate invocation**:  
 When multiple users hit the endpoint:

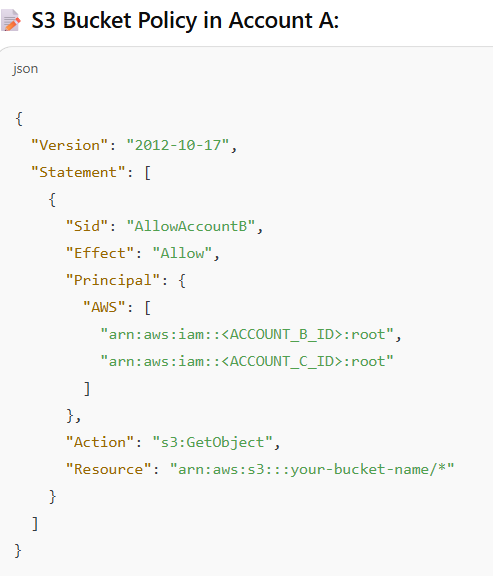
* Each user’s request triggers a **separate Lambda invocation**
* Each invocation is handled in **isolation**

By default, your AWS account allows **1,000 concurrent Lambda executions** per region.

**Lets say I have 3 different accounts I have Account A , Account B, Account C and I have a S3 bucket in account A, Now the Account B and C should be able to acces the s3 bucket in Account A. What all changes that we need in terms of account A and. What all changes that we need on S3 bucket and how do you grant access to account B and C**

### **In Account A (where the S3 bucket lives):**

1. **Update the S3 bucket policy** to allow **Account B and C** to access the bucket.
2. (Optional but recommended) Use **IAM roles** for better control and auditing.



### **In Account B and Account C:**

1. Create an **IAM role or IAM user** that can **assume a role** or make direct requests to the S3 bucket in Account A.
2. If using IAM role assumption, ensure the role in Account A **trusts B and C**.

We have an S3 bucket and a user Joe should only be able to access that S3 bucket

### **IAM Policy for User Joe (Best Practice)**

Attach a policy to **Joe's IAM user** that **grants access only to the specific bucket**.



**Lets say I have an application which require very high throughput like millions of requests per second ? Which type of load balancer can be used and why?**

WE can choose NLB as it can handle millions of requests per second and NLB has less latency supports TCP UDP protocols, can route to IP address as targets.  
  
Security Tip:  
If using **NLB with TLS termination**, enable **AWS Certificate Manager (ACM)** with TLS listeners, and configure **WAF** on a proxy layer  
  
Why we wont choose ALB in this scenario?  
We **won’t choose an Application Load Balancer (ALB)** in this scenario — where the application requires **very high throughput like millions of requests per second** — due to the following key reasons:

|  |
| --- |
| **Less Performance at Scale** |

|  |
| --- |
| ALB operates at Layer 7 (HTTP/HTTPS), which means it's **heavier** due to header parsing, routing decisions, and rule evaluations. This adds **latency** and limits max throughput.  Why ALB is slow than NLB ? |

* ALB (Application Load Balancer) is slower than NLB because it operates at Layer 7 (HTTP/HTTPS), which requires parsing requests, evaluating rules, and making routing decisions — adding processing overhead. It performs deep packet inspection and supports advanced features like path- or host-based routing, which increases latency. In contrast, NLB works at Layer 4 (TCP/UDP), forwarding traffic without inspecting content, resulting in much lower latency and higher throughput.  
    
  I have an application which req high availability and we have 3 diff AZ’s and all and EC2 is running on 3 AZ’s and data needs to be synchronised across all the 3 Ec2 AZ’s. Even when the single instance goes down the other 2 should be still available?  
    
  If your app is **stateful** (e.g., storing files or session data), use:
* **Amazon EFS (Elastic File System)** – mount it across all EC2s; it automatically syncs and is AZ-resilient.
* Or, use **S3** for object storage with client-side syncing.
* If you're using a **database**, choose:
* **Amazon RDS (Multi-AZ)** or **Aurora** with replication across AZs.  
  Use **IAM roles** to grant secure access to EFS/S3/RDS.

To meet high availability and data consistency across 3 AZs, I would deploy EC2 instances in all 3 AZs behind a Load Balancer. For data synchronization, I’d use EFS for shared file storage or RDS with Multi-AZ support for databases. The Load Balancer ensures traffic is routed only to healthy instances, and Auto Scaling replaces failed instances, maintaining availability even during failure in one AZ.

In that case in what scenarios, we use EBS over EFS?  
**Use EBS (Elastic Block Store) when:**

1. **Single EC2 instance needs high-performance storage**
   1. EBS is a **block-level storage**, like a virtual hard disk.
   2. Attached to **one EC2 instance at a time** (unless using EBS Multi-Attach with limitations).
2. **You need low-latency, high IOPS**
   1. Ideal for workloads like **databases (e.g., MySQL/PostgreSQL)**, where speed and consistency are critical.
3. **You want tight control over storage volume types**
   1. EBS offers **gp3, io2**, etc., letting you fine-tune performance and cost.
4. **Application doesn’t need shared access**
   1. If the application is **not distributed**, and runs on a single EC2, EBS is preferred.

**Lets say I have 2 petabytes of data.The client want to back then up to a S3 bucket. How can we do that ?**

### **Recommended Options**

#### **1. AWS Snowball or Snowmobile (Offline Transfer)**

* **Best for:** Extremely large datasets (like 2 PB) when network bandwidth is limited.
* **How it works:**
  + Order AWS **Snowmobile** (for exabyte-scale) or **Snowball Edge** (up to 100 TB per device).
  + AWS ships the device to your location.
  + You copy the data onto the device.
  + Ship it back to AWS.
  + AWS uploads the data to your **S3 bucket**.
* **Why this is ideal:** Fast, secure, avoids network bottlenecks, supports encryption.

#### **2. AWS DataSync (Online Transfer)**

* **Best for:** If you have high-speed internet (Gbps+), and want to migrate data over time.
* **How it works:**
  + Install **DataSync agent** on-prem.
  + Configure source and destination (S3).
  + It handles **incremental syncs**, **encryption**, **error handling**, and **throttling**.
* **Limitations:** Not ideal if network bandwidth is low or inconsistent.

Now lets say you have network issue while uploading objext to S3 and the network goes down for a couple of hours and what happens when the network comes back ?

### **Scenarios and Behavior**

#### **🔸 1. Single PUT Upload (≤5GB)**

* **If interrupted mid-upload:** The upload **fails**.
* You will need to **re-upload the object from scratch** when the network is back.
* **No partial progress is saved.**

#### **🔸 2. Multipart Upload (>5GB or large files)**

* **If using multipart upload and the network fails mid-way:**
  + The uploaded parts are **retained** by S3 for **7 days**.
  + When the network comes back, your client or upload tool (like AWS CLI, SDKs, or aws s3 cp) can **resume** from the last successfully uploaded part.
  + You **do not** need to restart the entire upload.

You are running an application on top of amazon linux machine the app team complained that 3 months ago the app use to run very smoothly nut now they are observing the lab with some latency or delayed response? What all thing you check to resolve this issue.  
**Summary of Checks**

|  |  |
| --- | --- |
| **Category** | **What to Check** |
| CPU/Memory | top, vmstat, CloudWatch |
| Disk | iotop, iostat |
| Network | iftop, netstat, packet loss |
| App Logs | Errors, timeouts, thread issues |
| System Logs | dmesg, /var/log/messages |
| Dependencies | DB, external services |
| Updates | yum history |
| Scaling | Instance type adequacy |

**Check System-Level Metrics**  
Use tools like top, htop, vmstat, iotop, or sar to inspect:

* **CPU Utilization**  
   High user or system CPU could indicate load. Look for any spikes.
* **Memory Usage**  
   Run free -m or top. Check for high memory usage or swap activity.
* **Disk I/O Bottlenecks**  
   Use iotop or iostat. High I/O wait time could slow your app.
* **Network Performance**  
   Check for packet loss, drops, or high bandwidth usage using iftop, nload, or netstat -s.  
    
  **Application-Level Logs**
* Check app logs (/var/log/, app-specific locations) for:
  + Errors or timeouts
  + Increased response time patterns
  + Thread starvation or connection pool issues

#### **3. System Logs**

* dmesg — for kernel-level issues like OOM killer, hardware errors
* /var/log/messages or /var/log/syslog — for overall system warnings/errors

#### **4. Resource Trends Over Time**

Use **CloudWatch metrics**:

* EC2: CPU, memory (if custom metric), disk I/O, network
* Check for anomalies around the time the latency started

#### **5. Process and Thread Analysis**

* Run ps aux --sort=-%cpu and ps aux --sort=-%mem to identify heavy processes
* Use strace, lsof, or perf top if CPU-bound

6. Application Dependency Health

Is the app calling external services or DBs?

Check DB performance (RDS metrics)

Check connection errors, DNS latency, or VPC security group/NACL .changes  
  
Keep on adding the EBS volume of 100 GB to an Ec2 untill and unless you get an error from AWS side , So how many EBS volume can be attached to an EBS volume?  
**answer depends on the instance type**.

### **General Limits:**

* Most EC2 instance types support **up to 27 EBS volumes** (including the root volume).
* The **limit includes the root volume**, so typically, you can attach **26 additional EBS volumes**.
* This is **based on the number of available NVMe or SATA block device mappings** supported by the instance.

### **To increase this limit:**

* You can’t directly increase the volume attachment limit per instance — it’s bound by the instance’s block device mapping capacity.
* But you can:
  + Use **RAID configurations** to combine multiple volumes.
  + Use **larger EBS volumes** instead of adding many small ones.
  + Use **Elastic File System (EFS)** for scalability if file system sharing is needed.

Max size of a single object uploaded to S3 ?

5 TB - Terabytes

Lets say you a 5tb of multiple files example we have 1 file of 200GB another file of 300GB total files costitute a 5TB of files and this S3 bucket is kept in S3 bucket of Acccount A you are the Admin of that account.Now another new project is coming in and it has a another aws account lets say it is account B and they also need this %TB of data. What they have done they have given you admin access to account B as well ? In how many possible ways you can transfer the data from Account A s3 bucket to Account B S3 bucket ?  
  
You have 5 TB of data in Account A’s S3 bucket, and you want to get this data into an S3 bucket in Account B. You have admin access to **both** accounts. Here are the main possible ways to transfer the data:  
**1. Cross-account S3 bucket replication (CRR)**

* Set up **Cross-Region Replication (CRR)** or **Same-Region Replication (SRR)** from Account A’s bucket to Account B’s bucket.
* Requires enabling replication and configuring proper IAM roles and bucket policies for cross-account access.
* Replication happens automatically as new objects are added (good for ongoing sync).
* For existing data, you might have to do an initial copy manually or use S3 batch replication.

### **2. AWS CLI / SDK Copy using AssumeRole**

* Use AWS CLI or SDK in Account B assuming a role from Account A that has read access to Account A’s bucket.
* Run a copy command to copy objects from Account A’s bucket to Account B’s bucket.
* Example: Use aws s3 cp or aws s3 sync with temporary credentials assuming cross-account role.
* Good for one-time bulk copy or periodic sync.

### **3. S3 Batch Operations**

* In Account A, create an S3 Batch Operations job to copy objects from Account A bucket to Account B bucket.
* You need to set up the destination bucket permissions to allow writes from the batch job.
* Useful for large-scale, managed batch copy jobs.

### **4. Data Pipeline or AWS Glue Jobs**

* Use AWS Data Pipeline or AWS Glue ETL jobs to read from Account A S3 bucket and write to Account B S3 bucket.
* Useful if some data transformation is needed during transfer.

### **5. AWS DataSync**

* Set up AWS DataSync between buckets (or use an S3 bucket as source and destination).
* Supports high-speed transfers and can handle large datasets efficiently.
* Works cross-account if roles and permissions are configured.

### **6. Manual Download and Upload**

* Download files locally from Account A bucket and then upload them to Account B bucket.
* Not recommended for very large datasets (like 5 TB) due to bandwidth and time.

### **Summary Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Use Case** | **Pros** | **Cons** |
| Cross-account Replication | Ongoing sync + new objects | Automatic, low maintenance | Initial sync requires extra |
| CLI/SDK Copy with AssumeRole | One-time or batch copy | Flexible, scriptable | Manual, can be slower |
| S3 Batch Operations | Large scale batch jobs | Managed by AWS, scalable | Requires setup and permissions |
| Data Pipeline / Glue | ETL + data movement | Integrated with AWS analytics | More complex to set up |
| AWS DataSync | Large scale fast transfer | High speed, efficient | Cost and setup |
| Manual Download/Upload | Small scale or one-off | Simple | Not practical for 5TB |

Lets say there is an application in capgemini datacenter in US region. In hyderabad location in aws you team has created a service. If someone from US data center wants to access your service in Hyderabad.What architecture component that will come into her in this scenario.

In this scenario, where an application in Capgemini’s US data center needs to access a service running in AWS Hyderabad region, the key architecture components involved would typically include:

1. **VPN or Direct Connect**
   1. To securely and reliably connect the on-premises US data center network with the AWS Hyderabad VPC network.
   2. **AWS Direct Connect** provides a dedicated, private, high-bandwidth connection.
   3. Alternatively, a **Site-to-Site VPN** over the internet can be used if lower bandwidth or quicker setup is acceptable.
2. **VPC (Virtual Private Cloud)**
   1. The service in AWS Hyderabad will be deployed inside a VPC, isolating it within a private network.
3. **Private Subnets & Security Groups**
   1. The AWS service likely runs in private subnets, secured by security groups and Network ACLs, ensuring only authorized traffic from the US data center network can reach it.
4. **Route Tables**
   1. The VPC route tables are configured to route traffic destined for the US data center network via the VPN or Direct Connect connection.
5. **DNS / Endpoint Access**
   1. If the service exposes APIs or web endpoints, either private endpoints or internal DNS names could be used to resolve and access the service securely.

Summary answer for interview:

To enable access from the Capgemini US data center to the AWS Hyderabad service, we typically establish a secure network connection using **AWS Direct Connect** or a **Site-to-Site VPN**. This connects the on-premises data center network to the AWS VPC in Hyderabad privately and securely. The AWS service runs inside a **VPC with private subnets** and appropriate **security groups** to restrict access. Routing is configured to direct traffic over the private connection, ensuring secure, low-latency communication without exposing the service over the public internet.

Let me more specific like it want to access eks cluster in Hyderabad.

* To enable secure communication between a Capgemini US data center and an Amazon EKS cluster deployed in the Hyderabad AWS region, we can set up a **Site-to-Site VPN connection**. This VPN establishes an **encrypted IPsec tunnel over the public internet**, allowing private network communication between on-premises resources and the AWS VPC.  
  **AWS VPN Gateway (VGW)**:
* Attach a **Virtual Private Gateway** to the **VPC** hosting the EKS cluster in Hyderabad.
* This acts as the target for the Site-to-Site VPN.
* **Customer Gateway (CGW)**:
* Represent the **on-premises VPN appliance** (at Capgemini US data center) in AWS.
* Configure it with the **public IP** and ASN of the on-prem device.
* **Site-to-Site VPN Connection**:
* Create a **VPN connection** between the VGW and CGW.
* AWS provides **2 VPN tunnels** for redundancy.
* **Routing Configuration**:
* Update **VPC route tables** in Hyderabad to route traffic destined for US private IP ranges through the VGW.
* Similarly, configure **on-prem routes** to reach the Hyderabad VPC CIDR via the VPN tunnel.

Recent Automation you have done in AWS?   
  
I have automated my EC2 Start/Stop to reduce cost.  
  
- Here I used services Lambda + Event Bridge to trigger start/stop schedules  
Firstly started with Creating 2 lambda functions with python code for invoking start and stop for EC2 instances.  
- For this I have assigned an EC2 IAM role for Lambda to execute tasks on EC2 instance.  
- Once the lambda function is created  
- I started to create my EventBrigde rule in that event bridge rule we have provided cron schedule.  
- And then associated an IAM role to invoke lambda function to Event Bridge.

In this way I automated start/stop ec2 instances and it significanty reduced cost.  
  
Suppose I want to delete 10,000 objects from the S3 bucket what is your ideal approach?  
  
We can use S3 Batch operations   
1. Create a manifest file (CSV) listing objects to delete)  
2. Submit a batch job in aws Console:  
Navigate to S3 → Batch Operations → Create Job.  
Choose Delete as the operation.  
Upload the manifest file.  
3. Run the Job: AWS process deletions in parallel   
-----------  
We can also use S3 default feature - life cycle rules   
**----> Set a lifecycle rule** to expire objects automatically:  
Go to S3 → Bucket → Management → Lifecycle Rules.   
  
Add a rule with:   
Prefix: temp/ (or leave blank for entire bucket).   
Expiration: "Delete after 1 day".   
AWS will auto-delete matching objects.

What is the purpose of health check in Route 53?  
- Health checks are only for public resources.  
Health checks are basically helpful for Automated DNS failover.  
These health checks will monitor an end point, (application, server, other aws resources) do the DNS failover   
For Exam: Our Route 53 Health check is monitoring an EC2 instance.Then we can have a primary instance with health check configured with threshold 3, intervlal 30 sec, protocol, http, https or TCP.  
If the health check fails, we can do DNS failover to secondary instance.

- We also have calculated Health Checks in this we have a parent health check that will monitor child health checks these child health check will be monitoring aws resouces like EC2 instances.  
- Here we specify how many health check need to pass to make the parent pass.  
  
---------> As Route 53 health checks are outside VPC and can’t access private endpoints. Here you can create CloudWatch alarm that will monitor the private resources and the health check will monitor the CW alarm directly.  
  
  
Is it possible to create a hosted zone for sub domain   
**Yes, it is possible to create a hosted zone for a subdomain in AWS Route 53.**

This is known as **delegating a subdomain** (like dev.example.com) from a parent domain (example.com) to another hosted zone — possibly even managed by a different AWS account.  
----> Hosted zone is container for records that define how to route traffic to its domain and subdomain.   
  
**IAM Identity Center:**

With Iam Identity center we can implement one Single Sing on SSO to all our AWS accounts in aws organizations.  
Business Cloud application (eg: MS 365,share point)

EC2 windows instances

We have identity providers like  
- Built in identity store in IAM Identity Center  
- 3rd Party: Active Directory, One Login , Okta  
Login flow:  
Browser ---> Iam Identity center Login page --> enter username & password--- > goes to AWS IAM Identity Center --> from there we can click on management console   
We can manage access accross multiple accounts in your AWS org.  
Permission sets- a collection of one or more IAM policies assigned to users or groups to define aws acccess/=,  
  
**Two employees work in different shifts Employee-1 = 10 AM to 5 PM Employee-2 = 6 PM to 2 AM How do you provide access based on timings**  
  
EventBridge Scheduler + Lambda  
- Create IAM access policy   
- setup Eventbridge rule to invoke a lambda function to attact the policy at 10 AM and detach the policy at 5 PM.  
  
Lambda will call: aws iam attach-user-policy --user-name employee-1 ...

**Transit gateway:**

- With transit gateway, we can connect multiple vpc’s together and can enable transitive communication between them.  
Ex:   
We have VPC peering that can enable communication between 2 vpc one to one communication lets say if we have 4 vpcs we need 6 peering connections between then.This is fine when we have 2 or 3 vpc's.  
  
- But what if we have 20 or 30 vpc’s then we should use transit gateway.  
with a transit gateway we can connect multiple vpc's together and enable transitive communication between them. For this to enable  
We first need to create a transit gateway from the vpc console lft pane select transit gateway and create in vpc console itself we have transit gateway attachments   
-->We need to create attachments for the vpc to attach them to transit gateway.  
once the attachments are created we can enable communication between them by updating the cidr of the vpcs in the subnet route tables add destination as vpc-cidr-block and target as tgw-id   
  
Another example   
- If we have vpcs in different regions. Then we need to create a transit gateway in each region and attach the vpcs to their local region transit gateway and We can peer the transit gateway in two different regions   
- Add routes in each VPC's route table pointing to the other VPC's CIDR via the local TGW.   
- Add routes in each TGW route table to forward traffic to the peered TGW.

Transit Gateway reduces the number of VPN connections needed for on-premises connectivity.   
Peering capabilities allow Transit Gateways to connect across regions and AWS accounts.

**Cloud Front:**  
- Cloudfront is a content Distribution Network.   
- We have edge locations All around the world. And then it will be connecting to your origin.   
- So it would be an S3 bucket or an Http endpoint or server and when the client connects and does an HTTP request into your edge location, then the edge location will see if it has it in cache if it does not have it in the cache.  
- Then it will go to the origin to get the requested result and then once retrieve the results, it will be cached so that if other client requests the same content from the same edge location, then the edge location does not need to go to the origin.   
- *Cloudfront Origins* can be. S3 bucket. An S3 bucket hosting a static website. Application Load Balancer. EC2 instances. Or any any HTTP backend?   
  
Content is cached at the edge location for a TTL so lets say our origin is updated with new content and we want our edge location to fetch the same new content then we can perform cache invalidation this will invalidate the cache on the edge location and when new request ismade edge location will reach the origin to fetch the new content and serve the CF users   
  
**Direct connect:**  
- With AWS Direct connect we get a dedicated private connection between our corporate data center and AWS   
- Here we commission an AWS Direct connect location [these are physical locations].they will have DC endpoints.  
- On-premise side we need to provision a customer router with firewall  
- On AWS VPC provision a VGW   
- Then we need to setup a private virtual interface in between all these locations to connect to virtual private gateway on the VPC side to access the aws vpc.  
- To connect public services such as s3 buckets and glaciar we need to setup a public virtual interface , then it goes through the same path but does not go into the VGW. It will directly connect to the aws services.  
- ***Direct Connect Gateway*** – to connect to one or more VPC’s in many different regions we need to use direct connect gateway  
data center ---> AWS DC location -----> DC gatway ----> VPC in diff regions   
  
**SCP:**  
**Service Control Policy:**  These are similar to IAM policies that are assigned to org Units or accounts to restrict users and roles.  
These SCP do not apply to management account because management account can do anything as there is a role that says no SCP ever apply to the management accoun.  
EX: lets say we have a scenario where we have an admin account that can do anything but we want that admin account to restrict not to create IAM users then we can use SCP here.