



## Table of Contents

Identity and Access Management ( IAM ) .....	6
1. What is AWS Identity and Access Management (IAM)? How does AWS IAM work? What are its key components? .....	6
2. What is the difference between authentication and authorization in AWS IAM? .....	7
3. How can you secure your AWS account using IAM? .....	7
4. How do IAM users differ from IAM roles? .....	8
5. What is the role of an IAM policy document? .....	8
6. How can you grant permissions to an IAM user? .....	8
7. How can you delegate permissions to AWS services? .....	8
8. What is cross-account access in AWS IAM? .....	8
9. How does IAM support identity federation? .....	8
10. What is the purpose of an IAM access advisor? .....	8
11. How does IAM enforce the principle of least privilege? .....	8
12. What is the difference between IAM policies and resource-based policies? .....	8
13. What is the IAM policy evaluation logic? .....	9
14. What is IAM Policy Conditions, and how can they enhance access control in AWS? .....	9
15. Explain the difference between IAM Roles and IAM Users .....	9
16. What is AWS Organizations, and how does it help in managing multiple AWS accounts? .....	9
17. How does IAM Cross-Account Access work, and what are the use cases? .....	9
18. What is AWS Single Sign-On (SSO), and how does it simplify identity management across multiple AWS accounts? .....	10
19. What are IAM Access Analyzer and IAM Access Advisor, and how do they contribute to security and compliance? .....	10
20. How does IAM Policy Simulations help in testing and validating access policies? .....	10
21. What is AWS Resource Access Manager (RAM), and how does it facilitate resource sharing across accounts? .....	10
22. Explain IAM Policy Variables and how they can be used in policies .....	11
Virtual Private Cloud ( VPC ) .....	12
1. What is Amazon Virtual Private Cloud (VPC)? What are its key components and features? .....	12
2. How can you connect your on-premises network to Amazon VPC? .....	15
3. What is a VPC peering connection? .....	15
4. How can you ensure private communication between instances in Amazon VPC? .....	15
5. Can you peer VPCs in different regions? .....	17

6.	How can you control public and private IP addresses in Amazon VPC? .....	17
7.	What is a VPN connection in Amazon VPC? .....	19
8.	How can you ensure high availability in Amazon VPC? .....	19
9.	Can you modify a VPC after creation? .....	20
10.	What is the purpose of the Amazon VPC Endpoint? .....	21
Elastic Cloud Compute ( EC2 ) .....		22
1.	What is Amazon EC2? .....	22
2.	How does Amazon EC2 work? .....	22
3.	What are the different instance types in EC2? .....	22
4.	Explain the differences between on-demand, reserved, and spot instances.....	22
5.	How can you achieve high availability for EC2 instances across multiple availability zones? .....	23
6.	What is an Amazon Machine Image (AMI)? What is the difference between an AMI and an instance store snapshot? .....	23
7.	Explain EC2 instance types and their use cases .....	23
8.	Explain the concept of EC2 Spot Fleet and how it differs from Spot Instances.....	23
9.	How does AWS Nitro System contribute to EC2 instance performance and security? .....	23
10.	What is EC2 instance metadata, and how can it be accessed? .....	24
11.	How does the Enhanced Networking feature improve EC2 instance performance? .....	24
12.	What is EC2 Auto Scaling, and how does it work? .....	24
13.	How to setup EC2 Auto Scaling? .....	24
14.	How can you secure your EC2 instances? .....	25
15.	Explain the difference between public IP and Elastic IP in EC2. ....	25
16.	What is Amazon EBS? .....	25
17.	How can you encrypt data on EBS volumes? .....	25
18.	How can you back up your EC2 instances? .....	25
19.	What is the difference between instance store and EBS-backed instances? .....	25
20.	What are instance metadata and user data in EC2? .....	25
21.	How can you launch instances in a Virtual Private Cloud (VPC)? .....	26
22.	What is the purpose of an EC2 security group? .....	26
23.	How can you automate the deployment of EC2 instances? .....	26
24.	How can you achieve high availability for an application using EC2? .....	26
25.	What is Amazon EC2 Instance Connect? .....	26
26.	What is an EC2 instance profile, and how is it different from IAM roles? .....	26

27.	What are the differences between NAT Gateways and NAT Instances? .....	26
28.	What is the maximum limit of elastic IPs anyone can produce? .....	27
	Storage Services ~ Simple (Object) Storage Service ( S3 ), EBS.....	28
1.	What are the storage services provided by AWS?.....	28
2.	What are the key features of Amazon S3? .....	29
3.	What is an S3 bucket? .....	30
4.	What is a pre-signed URL in S3? .....	31
5.	What is the difference between S3 Standard, S3 Intelligent-Tiering, and S3 One Zone-IA storage classes? .....	32
6.	How can you optimize costs in Amazon S3?.....	32
7.	What is the AWS Snowball device?.....	33
8.	What is Amazon S3 Select? .....	33
9.	Give an example of S3 select .....	34
10.	What is the difference between Amazon S3 and Amazon EBS?.....	35
11.	How can you enable server access logging in Amazon S3? .....	35
12.	How can you replicate data between S3 buckets within the same region? .....	36
13.	How many S3 buckets can be created? .....	36
	AWS Lambda .....	37
1.	What is AWS Lambda? How does it work? .....	37
2.	What is the maximum execution duration for a single AWS Lambda invocation? .....	38
3.	How do you pass data to and from AWS Lambda functions?.....	38
4.	Can AWS Lambda functions communicate with external resources? .....	38
5.	What are AWS Lambda layers? .....	38
6.	How can you handle errors in AWS Lambda functions?.....	38
7.	Can AWS Lambda functions access the internet?.....	38
8.	How can you configure environment variables for AWS Lambda functions? .....	38
9.	What is the difference between synchronous and asynchronous invocation of Lambda functions? .....	39
10.	What is the AWS Lambda Event Source Mapping? .....	39
11.	How can you manage the permissions and execution roles for AWS Lambda functions?.....	39
12.	What is AWS Step Functions? .....	39
13.	How can you automate the deployment of AWS Lambda functions?.....	39
14.	Can AWS Lambda functions connect to on-premises resources? .....	39
15.	What is the Cold Start issue in AWS Lambda? .....	39

Advanced Topics .....	40
1. What are the tools and techniques that you can use in AWS to identify if you are paying more than you should be, and how to correct it?.....	40
2. What services can be used to create a centralized logging solution? .....	41
3. How to create a centralized logging solution in AWS? .....	42
4. What are the native AWS Security logging capabilities? .....	43
5. What is a DDoS attack, and what services can minimize them?.....	43
6. How do you set up a system to monitor website metrics in real-time in AWS?.....	44
7. Name some of the AWS services that are not region-specific.....	44
8. What are the elements of an AWS CloudFormation template? .....	44
9. What are the different types of load balancers in AWS? .....	45
10. Can AWS Config aggregate data across different AWS accounts? .....	45
11. How do you troubleshoot issues with an application that's running into AWS cloud environment? .....	46
12. What is the difference between Amazon RDS and Amazon DynamoDB? .....	48
13. Does AWS support Cassandra database? .....	49
14. What is the difference between DynamoDB and Amazon Keyspaces? .....	50
15. What is the difference between Amazon SQS and SNS? .....	51
16. What is dead-letter queues? .....	52
17. What is the difference between Apache Kafka and Amazon SQS?.....	53
18. What is the difference between Apache Kafka and Amazon SNS?.....	54
19. What is AWS well architected framework? .....	56
21. What are the differences between Apache Kafka and AWS Kinesis? .....	58
22. How do you design a big data application on AWS?.....	60
23. What are big data tools provided by AWS? .....	62
24. What is AI? .....	63
25. How do you implement an AI application on AWS? .....	64
26. What are AI and ML tools provided by AWS?.....	66

# Identity and Access Management ( IAM )

1. What is AWS Identity and Access Management (IAM)? How does AWS IAM work? What are its key components?

**AWS Identity and Access Management (IAM)** is a web service that enables you to securely control access to AWS services and resources. IAM allows you to manage users, groups, roles, and permissions to ensure that only authorized individuals and applications can interact with your AWS environment. Here's an overview of how AWS IAM works:

- **Users:** Users represent individual AWS account holders or resources that need access to AWS services. Each user has a unique username and associated credentials (such as access keys, passwords, or multi-factor authentication devices).
  - **Access Keys:** Users can have access keys, which consist of an access key ID and a secret access key. Access keys are used to authenticate programmatic requests to AWS services using the AWS Command Line Interface (CLI), SDKs, or other tools.
  - **Multi-Factor Authentication (MFA):** MFA adds an extra layer of security by requiring users to provide a time-based, one-time password in addition to their regular username and password. MFA can be enabled for individual users to enhance account security.
- **Groups:** Groups are collections of IAM users. Instead of attaching policies directly to users, you can organize users into groups and attach policies to the groups. This simplifies the management of permissions, especially when users share similar roles or responsibilities.
- **Roles:** IAM roles are similar to users but are not associated with a specific person. Instead, roles are assumed by entities, such as AWS services, applications, or federated users. Roles define a set of permissions and policies that determine what actions the entity can perform.
  - **Temporary Security Credentials:** IAM allows you to grant temporary security credentials using roles. This is particularly useful for scenarios like cross-account access, where entities in one AWS account need temporary access to resources in another account

- **Policies:** IAM policies are JSON documents that define permissions and access controls. Policies specify what actions are allowed or denied, which resources can be accessed, and under what conditions. Policies can be attached to users, groups, or roles.

**Special Policy Types:**

- **Resource-Based Policies:** IAM enables you to attach policies directly to AWS resources. For example, you can define who can launch an Amazon EC2 instance or who can access an Amazon S3 bucket by attaching policies directly to those resources.
- **Service Control Policies (SCPs):** SCPs are used in AWS Organizations to set fine-grained permissions on the entire organization or specific organizational units. SCPs can restrict the maximum permissions that can be granted to IAM entities within an organization.

**Supported Features:**

- **Identity Federation:** IAM supports identity federation, allowing users to sign in to the AWS Management Console using credentials from their corporate directory or other identity provider (IdP). This is achieved through standards like Security Assertion Markup Language (SAML) or OpenID Connect (OIDC).
- **Cloud-Trail Integration:** AWS IAM integrates with AWS CloudTrail, providing audit trails of actions taken by IAM users, groups, and roles. CloudTrail logs capture details about who made requests, what actions were performed, and when they occurred

2. What is the difference between authentication and authorization in AWS IAM?

Authentication is the process of verifying the identity of users or entities, while authorization is the process of granting or denying access to resources based on policies and permissions.

3. How can you secure your AWS account using IAM?

You can secure your AWS account by enforcing the principle of least privilege, creating strong password policies, enabling multi-factor authentication (MFA), and regularly reviewing permissions.

4. How do IAM users differ from IAM roles?

IAM users are individuals or entities that have a fixed set of permissions associated with them. IAM roles are temporary credentials that can be assumed by users or AWS services to access resources.

5. What is the role of an IAM policy document?

An IAM policy document defines the permissions and actions that are allowed or denied. It is written in JSON format and attached to users, groups, or roles.

6. How can you grant permissions to an IAM user?

You can grant permissions to an IAM user by attaching policies to the user directly or by adding the user to groups with associated policies.

7. How can you delegate permissions to AWS services?

Use IAM roles. IAM roles allow you to delegate permissions to AWS services like EC2 instances, Lambda functions, and more, without exposing long-term credentials.

8. What is cross-account access in AWS IAM?

Cross-account access allows you to grant permissions to users or entities from one AWS account to access resources in another AWS account.

9. How does IAM support identity federation?

IAM supports identity federation by allowing users to access AWS resources using temporary security credentials obtained from trusted identity providers (e.g., SAML, OpenID Connect).

10. What is the purpose of an IAM access advisor?

IAM access advisors provide insights into the services that users accessed and the actions they performed. This helps in auditing and understanding resource usage.

11. How does IAM enforce the principle of least privilege?

IAM enforces the principle of least privilege by allowing you to define specific permissions for users, groups, or roles, reducing the risk of unauthorized access.

12. What is the difference between IAM policies and resource-based policies?

IAM policies are attached to identities (users, groups, roles), while resource-based policies are attached to AWS resources (e.g., S3 buckets, Lambda functions) to control access from different identities.



### 13. What is the IAM policy evaluation logic?

IAM uses an explicit deny model, which means that if a user's permissions include an explicit deny statement, it overrides any allow statements in the policy.

### 14. What is IAM Policy Conditions, and how can they enhance access control in AWS?

IAM Policy Conditions are elements in a policy that define the circumstances under which the policy is in effect. They provide fine-grained control over access based on attributes such as time, IP address, or the use of secure connections. For example, you can create a condition to allow access only during specific hours of the day or from a particular IP range, enhancing security and compliance.

### 15. Explain the difference between IAM Roles and IAM Users

- **IAM Users:** Represent an individual or an entity that interacts with AWS services. Users have permanent credentials (access key and secret key) and are often associated with humans or applications.
- **IAM Roles:** Provide temporary security credentials for tasks, services, or applications. Roles are assumed by users, services, or AWS resources, and they don't have long-term credentials. Roles are used for cross-account access and delegation of permissions

### 16. What is AWS Organizations, and how does it help in managing multiple AWS accounts?

AWS Organizations is a service that enables you to consolidate multiple AWS accounts into an organization that you create and centrally manage. Key features include

- **Consolidated Billing:** Allows you to receive a single bill for all accounts in the organization
- **Service Policies (SCPs Control):** Enables fine-grained control over access and permissions across accounts
- **Organizational Units (OUs):** Allows you to organize accounts into hierarchies for easier management

### 17. How does IAM Cross-Account Access work, and what are the use cases?

IAM Cross-Account Access allows one AWS account to access resources in another AWS account. Use cases include

- **Resource Sharing:** Accessing resources in another account, such as S3 buckets, Lambda functions, or EC2 instances.
- **Federated Access:** Allowing users from one AWS account to assume roles in another AWS account for temporary access.

- **Centralized Management:** Enabling centralized management of resources and permissions while keeping accounts separate

18. What is AWS Single Sign-On (SSO), and how does it simplify identity management across multiple AWS accounts?

AWS Single Sign-On (SSO) is a cloud SSO service that simplifies user access management for multiple AWS accounts and applications. Key features include

- **Centralized User Management:** Users sign in once to access all assigned accounts and applications.
- **Integrated with AWS Organizations:** Easily manage access across the organization and assign permissions centrally.
- **Integrated Applications:** SSO integrates with various AWS applications and third-party SAML 2.0 applications

19. What are IAM Access Analyzer and IAM Access Advisor, and how do they contribute to security and compliance?

- **IAM Access Analyzer:** Analyzes resource-based policies to identify and recommend access permissions that are overly permissive. It helps in detecting unintended access, reducing the risk of security breaches.
- **IAM Access Advisor:** Provides insights into service last accessed information for AWS services. It helps in understanding which services a user or role accessed recently and helps in rightsizing permissions

20. How does IAM Policy Simulations help in testing and validating access policies?

IAM Policy Simulations allow you to test and validate the effects of IAM policies before applying them. Key benefits include:

- **Policy Validation:** Simulations help ensure that policies provide the intended access and don't grant unnecessary permissions.
- **Risk Mitigation:** Reduce the risk of misconfigurations or unintended access by simulating real-world scenarios.
- **Resource Protection:** Verify that policies protect sensitive resources and prevent unauthorized actions

21. What is AWS Resource Access Manager (RAM), and how does it facilitate resource sharing across accounts?

AWS **Resource Access Manager** (RAM) allows resource owners to share their AWS resources (such as Amazon S3 buckets or Amazon Aurora databases) with

other AWS accounts. It simplifies resource sharing by centralizing and automating the process, ensuring consistent and controlled access

## 22. Explain IAM Policy Variables and how they can be used in policies

IAM **Policy Variables** allow you to create more flexible and dynamic policies. Commonly used variables include `${aws:username}`, `${aws:useruid}`, `${aws:requester}`, etc.

For example, you can use `${aws:username}` in a policy to grant permissions based on the IAM user's username, allowing for more granular control and automation

# Virtual Private Cloud ( VPC )

1. What is Amazon Virtual Private Cloud (VPC)? What are its key components and features?

VPC provides logical isolation of resources on AWS cloud. It allows you to control your network environment, including IP addresses, subnets, and security settings.

Key Property values to provide during VPC creation:

- IPv4 CIDR – This defines the overall address space available to the VPC and its subnets.
- Cost based Considerations
  - Shared or Dedicated Tenancy (Default is shared hardware)
  - VPC Flow Logs: to capture information about the IP traffic going to and from network interfaces in the VPC (logs are stored on S3 or CloudWatch logs)

Key Components of VPC:

1. **Subnets:** Subnets are logical subdivisions of an IP network in your VPC.
  - Each subnet is associated with an Availability Zone (AZ)
  - Subnet can be either public or private.
  - Public subnets have a route to the internet via an Internet Gateway, while private subnets do not.

Key Property values to provide during VPC creation

- IPv4 CIDR – Subset of its VPC CIDR
  - Availability Zone
  - Auto assign public IP (for internet access)
2. **Route Tables:** is subnet level security control and are used to create public and private subnets.
    - A route table contains a set of routes, each specifying a destination (CIDR block) and a target (e.g., another subnet, an internet gateway, a VPN connection, etc.).
    - Routes determine where traffic is directed based on its destination.
    - Each subnet in a VPC must be associated with a route table, which controls the traffic routing for that subnet.
    - Route tables are used to create public and private subnets. Public subnets have a route to the internet via an internet gateway, while private subnets route traffic internally or through a NAT gateway or NAT instance.

3. **Network Access Control Lists (NACLs):** NACLs are another security control at subnet level.
  - NACL act like a stateless firewalls at the subnet level.
  - NACL is an ordered set of rules that define the allowed and denied communication between subnets and the internet.
  - Each subnet in a VPC is associated with a NACL. NACLs complement the security groups
  - Network ACLs are stateless, meaning that they do not keep track of the state of the connections. In contrast, security groups are Stateful and automatically allow return traffic
  - Rules in a Network ACL are evaluated based on their rule numbers in ascending order. The rule with the lowest rule number that matches the traffic is applied
  - Network ACLs are part of the defense-in-depth security strategy, complementing security groups. While security groups control traffic at the instance level, Network ACLs control traffic at the subnet level
4. **Security Groups:** Instance level security control
  - Security groups are associated with instances and operate at the instance level, providing stateful control over traffic.
  - Security Groups act as virtual firewalls for your instances. They control inbound and outbound traffic based on rules you configure.
  - Each instance in a VPC must be associated with one or more security groups.
  - Each security group has a set of inbound and outbound rules that define the allowed traffic.
  - Security groups operate at the stateful level. If you allow inbound traffic from a specific IP address, the corresponding outbound reply traffic is automatically allowed
  - By default, all inbound traffic is denied, and you must explicitly define rules to allow specific traffic
  - Changes to security group rules take effect immediately, allowing for dynamic updates to access controls without requiring instance restarts

**5. Internet Gateway (IGW):**

- An Internet Gateway is a horizontally scaled, redundant component that allows instances in your VPC to connect to the internet and enables internet-based services to communicate with instances in your VPC.
- Public subnets typically have a route to the internet via an Internet Gateway.

**6. NAT Gateway/NAT Instance:**

- Network Address Translation (NAT) Gateways or NAT Instances allow instances in private subnets to initiate outbound traffic to the internet while preventing inbound traffic from reaching them.
- NAT Gateways are managed services provided by AWS.

**7. VPC Peering:**

- VPC Peering enables you to connect one VPC with another, allowing instances in the peered VPCs to communicate as if they are on the same network.
- VPC peering is a one-to-one relationship and is not transitive.

**8. Elastic IP Addresses (EIP):**

- Elastic IP addresses are static, public IPv4 addresses that you can allocate and associate with instances in your VPC.
- EIPs provide a consistent IP address for instances, facilitating external communication.

**9. VPC Endpoints:**

- VPC Endpoints enable private connectivity between your VPC and supported AWS services without the need for internet gateways, NAT devices, or VPN connections.
- Examples include Amazon S3 and Dynamo-DB endpoints.

**10. Virtual Private Gateway (VGW):**

- The Virtual Private Gateway represents the VPN endpoint on the AWS side of a VPN connection. It is associated with your VPC and allows secure communication between your on-premises network and your VPC.

**11. VPN Connections:**

- VPN Connections enable secure communication between your VPC and your on-premises network.

- It uses IPsec (Internet Protocol Security) to establish an encrypted connection over the internet.

#### 12. **Transit Gateway:**

- Transit Gateway is a fully managed service that allows you to connect multiple VPCs, on-premises networks, and remote networks in a hub-and-spoke model.
- It simplifies network architecture and reduces the need for complex peering relationships.

#### 13. **Direct Connect:**

- AWS Direct Connect provides dedicated network connections between your on-premises data center and AWS.
- It bypasses the public internet, offering higher bandwidth and more reliable connectivity.

#### 2. [How can you connect your on-premises network to Amazon VPC?](#)

You can establish a Virtual Private Network (VPN) connection or use AWS Direct Connect to connect your on-premises network to Amazon VPC.

#### 3. [What is a VPC peering connection?](#)

VPC peering allows you to connect two VPCs together, enabling resources in different VPCs to communicate as if they were on the same network.

#### 4. [How can you ensure private communication between instances in Amazon VPC?](#)

To ensure private communication between instances in Amazon VPC (Virtual Private Cloud), you can implement the following strategies:

- **Private Subnets:** {Network Segmentation} Place instances that require private communication in private subnets. A private subnet is a subnet that does not have a direct route to the internet. This prevents instances in the private subnet from being directly accessed from the internet.
- **Security Groups:** {Firewall Rules}: Use security groups to control inbound and outbound traffic to and from instances. Configure security groups to allow only necessary traffic and explicitly deny traffic that is not required for private communication. Security groups are stateful, and you can define rules based on protocols, ports, and IP addresses.
- **Network Access Control Lists (NACLs):** {Subnet-Level Firewall Rules}: NACLs operate at the subnet level and provide an additional layer of security. Configure NACLs to control traffic entering and leaving the subnets, allowing

- only the required communication. NACLs are stateless, meaning that rules for inbound and outbound traffic must be specified separately.
- **Private IP Addresses:** {Internal Communication}: Instances within the VPC communicate using private IP addresses. By default, instances receive private IP addresses from the IP address range of the associated subnet. Private IP addresses are used for internal communication within the VPC and are not accessible from the public internet.
  - **VPC Peering:** {Secure Communication Between VPCs}: If you have multiple VPCs and need private communication between them, use VPC peering. VPC peering enables instances in one VPC to communicate with instances in another VPC using private IP addresses. Ensure that security groups and NACLs are appropriately configured for secure communication.
  - **AWS PrivateLink:** {Private Connectivity to AWS Services}: Use AWS PrivateLink to enable private communication between your VPC and supported AWS services without using public IP addresses or traversing the internet. AWS PrivateLink provisions a private endpoint within your VPC for accessing the service.
  - **VPN Connections:** {Secure On-Premises Communication}: If you have an on-premises network, establish a VPN connection to your VPC. VPN connections provide secure communication between your on-premises network and instances in your VPC, allowing for private communication over the encrypted connection.
  - **Amazon VPC Endpoints:** {Private Access to AWS Services}: Use VPC endpoints to enable instances in your VPC to access AWS services (e.g., S3, DynamoDB) privately without going over the internet. VPC endpoints provide a secure and direct connection to the AWS service.

By combining these strategies, you can create a secure and private environment within your Amazon VPC, allowing instances to communicate with each other while minimizing exposure to external threats. It's crucial to carefully design and configure your VPC's network architecture, security groups, NACLs, and other components to meet your specific security and privacy requirements.



### 5. Can you peer VPCs in different regions?

**NO**, VPC peering in Amazon Web Services (AWS) is supported only within the same region. You cannot directly peer VPCs that are in different regions. Each VPC peering connection is specific to a particular AWS region.

If you need to establish communication between resources in VPCs located in different regions, you have a few alternative options:

- **AWS Direct Connect:** Use AWS Direct Connect to establish a dedicated network connection between your on-premises data center and each VPC in different regions. While this doesn't provide VPC-to-VPC peering, it allows secure and dedicated connectivity between your on-premises environment and each VPC.
- **Transit Gateway:** Use AWS Transit Gateway to simplify and scale your network connectivity. Transit Gateway allows you to connect multiple VPCs and on-premises networks within the same region. It is designed to scale horizontally and can be a central hub for connecting multiple VPCs.
- **Inter-Region VPC Peering (Limited):** AWS announced support for inter-region VPC peering in November 2021. However, note that inter-region VPC peering is currently limited and comes with certain restrictions. You should check the latest AWS documentation for the most up-to-date information on the availability and limitations of inter-region VPC peering.
- **Global Accelerator:** AWS Global Accelerator is a service that provides static IP addresses and any cast routing to improve the availability and fault tolerance of applications. While it's not specifically for VPC peering, it can be used to enhance global application availability.

### 6. How can you control public and private IP addresses in Amazon VPC?

Through the assignment of Elastic IP addresses (EIPs), the allocation of private IP addresses to instances, and the use of Network Address Translation (NAT) devices.

#### **Public IP Address:**

- **Elastic IP Addresses** (EIPs): Elastic IP addresses are public IPv4 addresses that you can allocate and associate with instances in your VPC.
- **EIP Association:** You can associate an Elastic IP address with an EC2 instance either during instance launch or by associating it later. EIPs are

not automatically assigned; you must explicitly allocate and associate them.

- **Releasing EIPs:** If you no longer need an Elastic IP address, you can release it back to the pool of available addresses, but be aware that releasing an EIP means you lose the ability to retain that specific IP address.

#### **Private IP Addresses:**

- **Private IPv4 Addresses:** Instances in a VPC are assigned private IPv4 addresses from the IP address range of the associated subnet. These addresses are used for internal communication within the VPC and are not directly accessible from the internet.
- **Dynamic or Static Assignment:** By default, instances are assigned private IP addresses dynamically. You can also configure static private IP addresses during instance launch or modify them later.

#### **Public and Private Subnets:**

- **Subnet Design:** The design of your subnets determines whether they are public or private. Public subnets are associated with a route to the Internet Gateway, allowing instances to have public IP addresses and access the internet. Private subnets do not have a direct route to the internet.
- **NAT Instances:** In private subnets, you can use NAT instances or NAT gateways to enable instances to initiate outbound traffic to the internet while preventing inbound traffic from reaching them. NAT instances or gateways have public IP addresses for internet communication on behalf of instances in private subnets.
- **Ephemeral Ports:** When using NAT instances, be aware that they use ephemeral ports for outbound connections, and you may need to manage the available ports if your instances have high outbound connection rates.
- **NAT Gateway:** NAT gateways are managed AWS services that simplify outbound internet connectivity for instances in private subnets. They are scalable and highly available, providing a more straightforward solution than managing NAT instances.

## 7. What is a VPN connection in Amazon VPC?

In Amazon VPC (Virtual Private Cloud), a VPN (Virtual Private Network) connection is a secure and encrypted communication link established between your on-premises network or data center and your Amazon VPC. It allows you to extend your on-premises network into the AWS cloud securely, creating a hybrid cloud environment. The VPN connection operates over the public internet but provides a secure and private connection between your on-premises network and your VPC.

AWS offers two types of VPN connections:

- **Site-to-Site VPN** - Site-to-Site VPN is used for connecting your on-premises network to your VPC
- **Client VPN** - Client VPN is designed for secure remote access to your VPC for individual users.

VPN connections are suitable for scenarios where a dedicated physical connection (AWS Direct Connect) is not practical or feasible, providing a secure and cost-effective option for extending your on-premises network into the AWS cloud.

## 8. How can you ensure high availability in Amazon VPC?

Ensuring high availability in Amazon VPC (Virtual Private Cloud) involves designing your network architecture and deploying resources in a way that minimizes single points of failure and provides redundancy. Here are some key strategies to achieve high availability in Amazon VPC:

- **Multi-Availability Zone (AZ) Deployment:** Distribute your resources across multiple Availability Zones. An Availability Zone is essentially a data center with redundant power, networking, and cooling. By deploying resources in multiple AZs, you ensure that if one AZ becomes unavailable, your application can continue running in another.
- **Elastic Load Balancing (ELB):** Use Amazon Elastic Load Balancing to distribute incoming traffic across multiple EC2 instances in different AZs. ELB automatically scales and routes traffic to healthy instances, improving the availability of your application.
- **Auto Scaling Groups:** Implement Auto Scaling Groups to automatically adjust the number of EC2 instances based on demand. Auto Scaling ensures that you

- have the right number of instances available across different AZs to handle varying levels of traffic.
- **Database Replication:** If you use a relational database like Amazon RDS, implement database replication across multiple AZs. This provides failover capability in case of a database instance failure in one AZ.
  - **Elastic IP Addresses (EIP):** Use Elastic IP Addresses for critical resources like load balancers or instances to ensure that they maintain the same IP address even if they need to be replaced or moved to another Availability Zone.
  - **Amazon Route 53 (DNS):** Leverage Amazon Route 53 for DNS management. Use health checks and failover routing policies to automatically redirect traffic to healthy resources in case of an availability issue in one AZ.
  - **Amazon CloudWatch Alarms:** Set up CloudWatch alarms to monitor the health of your resources and automatically trigger actions (e.g., scaling, failover) based on predefined thresholds. This proactive monitoring helps detect and address issues before they impact availability.
  - **Amazon CloudFront (CDN):** Use Amazon CloudFront to cache and deliver content from edge locations worldwide. This not only improves the performance of your application but also provides additional redundancy and availability.
  - **Redundant VPN Connections:** If you are using VPN connections to connect to your on-premises network, establish redundant VPN connections to different Virtual Private Gateways in different AZs to ensure continuous connectivity.
  - **Amazon S3 Cross-Region Replication:** If you use Amazon S3 for storing data, consider implementing cross-region replication for critical data. This provides redundancy and ensures data availability in case of a regional outage.
  - **AWS Direct Connect:** Use AWS Direct Connect to establish a dedicated network connection between your on-premises data center and your VPC. This can improve network reliability and reduce latency.

By combining these strategies, you can create a highly available architecture in Amazon VPC that can withstand failures at the instance, AZ, or even regional levels, providing a resilient environment for your applications and services

#### 9. Can you modify a VPC after creation?

While you can modify certain attributes of a VPC, such as its IP address range and subnets, some attributes are immutable, like the VPC's CIDR block.

## 10. What is the purpose of the Amazon VPC Endpoint?

An Amazon VPC Endpoint enables you to privately connect your VPC to supported AWS services and VPC endpoint services without needing an internet gateway or VPN connection.

Practical use cases for Amazon VPC Endpoints include:

- **Amazon S3 Access:** Secure Access to S3: VPC Endpoints for Amazon S3 enable you to access S3 buckets securely from within your VPC without the need for internet gateways or NAT devices. This is beneficial for enhanced security and reduced data transfer costs.
- **Amazon DynamoDB Access:** Private Access to DynamoDB: VPC Endpoints for DynamoDB provide a way to access DynamoDB tables securely from within your VPC, ensuring that data does not traverse the public internet.
- **AWS Lambda Integration:** Invoke Lambda Functions: VPC Endpoints for AWS Lambda allow your VPC resources to invoke Lambda functions without going through the internet. This enhances security and can improve performance.
- **Amazon SNS and SQS Access:** Private Communication with SNS/SQS: VPC Endpoints for Amazon Simple Notification Service (SNS) and Simple Queue Service (SQS) enable private communication between your VPC and these messaging services, ensuring data does not leave the AWS network.
- **AWS CloudWatch Logs and Events:** Secure Communication with CloudWatch: VPC Endpoints for CloudWatch Logs and CloudWatch Events allow your VPC resources to securely send logs and events data to CloudWatch without relying on internet connectivity.
- **Amazon ECR (Elastic Container Registry):** Secure Docker Image Access: VPC Endpoints for Amazon ECR allow your containerized applications within the VPC to pull Docker images from the registry without exposing the traffic to the public internet.
- **Amazon API Gateway:** Private API Access: If you use Amazon API Gateway to create APIs, you can use VPC Endpoints to privately access these APIs from within your VPC, improving security and avoiding public internet exposure.
- **AWS Secrets Manager and AWS Config:** Private Access to Secrets and Config: VPC Endpoints for AWS Secrets Manager and AWS Config enable private and secure communication for managing secrets and configuration compliance within your VPC.

# Elastic Cloud Compute ( EC2 )

## 1. What is Amazon EC2?

Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. It allows users to create, configure, and manage virtual servers (known as instances) in the AWS cloud.

## 2. How does Amazon EC2 work?

It enables users to launch instances based on pre-configured Amazon Machine Images (AMIs). These instances run within virtual private clouds (VPCs) and can be configured with various resources like CPU, memory, storage, and networking.

## 3. What are the different instance types in EC2?

Amazon EC2 offers a wide range of instance types optimized for different use cases, such as general-purpose, memory-optimized, compute-optimized, and GPU instances.

## 4. Explain the differences between on-demand, reserved, and spot instances.

- **On-Demand Instances:** Pay-as-you-go pricing with no upfront cost or commitment.
- **Reserved Instances:** Reserved capacity for a 1- or 3-year term, offering significant cost savings compared to On-Demand pricing.  
HOW: EC2 -> "Purchase Reserved Instances"
- **Spot Instances:** Allows users to bid on unused EC2 capacity, potentially leading to significantly lower costs.

HOW: EC2 -> Spot Request, Following are blueprints available

### Spot Blueprints:

Generate a reusable quick start template as CloudFormation or Terraform, based on your workload. Use the template to configure a compute environment using Spot Instances, following best practices.

Available blue prints are –

- Batch ~ for Batch workload using Spot Instances inside AWS Batch managed compute environment
- EC2 AutoScaling Group ~ template of a web services workload, using EC2 Autoscaling group with Spot Instance
- EMR ~ EMR cluster using Spot Instances as task nodes.
- EKS ~ EKS cluster with Kubernetes worker nodes as Spot Instances.
- ECS with EC2 ~ ECS cluster using Spot Instances to run tasks.
- ECS with Fargate ~ ECS cluster using Fargate Spot to run tasks.

5. How can you achieve high availability for EC2 instances across multiple availability zones?

- Use Auto Scaling groups with instances spread across multiple availability zones.
- Use Elastic Load Balancers (ELB) to distribute incoming traffic across instances in different availability zones

6. What is an Amazon Machine Image (AMI)? What is the difference between an AMI and an instance store snapshot?

An **Amazon Machine Image (AMI)** is a pre-configured template that contains the information required to launch an EC2 instance. AMIs can include an operating system, applications, data, and configuration settings. Can be created from both EBS-backed and instance store-backed instances

**Instance Store Snapshot:** A snapshot of the root volume of an instance store-backed EC2 instance. Can be used to create an AMI but is specific to instance store-backed instances

7. Explain EC2 instance types and their use cases

- General Purpose (e.g., t3, m5): Balanced compute, memory, and networking resources. Suitable for diverse workloads.
- Compute Optimized (e.g., c5): High-performance processors, ideal for compute-bound applications.
- Memory Optimized (e.g., r5): High-memory instances for memory-intensive applications, such as databases and analytics.
- Storage Optimized (e.g., i3, h1): High-performance storage for I/O-intensive applications and big data workloads

8. Explain the concept of EC2 Spot Fleet and how it differs from Spot Instances.

- EC2 Spot Fleet is a collection of Spot Instances, On-Demand Instances, and optionally, Reserved Instances.
- Spot Fleet allows you to provision capacity across different instance types and across multiple Availability Zones, optimizing for cost and availability

9. How does AWS Nitro System contribute to EC2 instance performance and security?

- The Nitro System is a combination of dedicated hardware and lightweight hypervisor designed to deliver high performance and strong security.
- It offloads networking, storage, and management functions from the host to provide better performance, efficiency, and security



#### 10. What is EC2 instance metadata, and how can it be accessed?

- EC2 instance metadata is a set of data about an instance that can be queried from within the instance.
- It can be accessed by making HTTP requests to a special URL: <http://<IP-Address>/latest/meta-data/> or <http://<IP-Address>/latest/user-data/>

#### 11. How does the Enhanced Networking feature improve EC2 instance performance?

- Enhanced Networking uses single root I/O virtualization (SR-IOV) to provide higher packet per second (PPS) and lower network jitter.
- It reduces the load on the host CPU and provides higher networking throughput and lower latency.

#### 12. What is EC2 Auto Scaling, and how does it work?

- EC2 Auto Scaling automatically adjusts the number of EC2 instances in a group based on policies defined by the user.
- It helps maintain application availability, automatically launching or terminating instances to match the desired capacity.

#### 13. How to setup EC2 Auto Scaling?

- Create launch configuration (provide ec2 instance details: instance type, key pair, and security groups.)
- Create Auto Scaling group : An Auto Scaling group is a collection of Amazon EC2 instances that are treated as a logical unit. You configure settings for a group and its instances as well as define the group's minimum, maximum, and desired capacity
- Configure Scaling Policies: you can set up policies to scale based on CPU utilization, network traffic, or custom metrics
  - Use Amazon CloudWatch to enable scaling policies and monitor metrics for your Auto Scaling groups and EC2 instances
  - Use Elastic Load Balancing to automatically distribute incoming application traffic across the instances in your Auto Scaling group



#### 14. How can you secure your EC2 instances?

You can enhance the security of EC2 instances by using security groups, Network ACLs, key pairs, and configuring firewalls. Additionally, implementing multi-factor authentication (MFA) is recommended for account access.

#### 15. Explain the difference between public IP and Elastic IP in EC2.

A public IP is assigned to an instance at launch, but it can change if the instance is stopped and started. An Elastic IP is a static IP address that can be associated with an instance, providing a consistent public IP even after stopping and starting the instance.

#### 16. What is Amazon EBS?

Amazon Elastic Block Store (EBS) provides persistent block storage volumes for EC2 instances. EBS volumes can be attached to instances and used as data storage.

#### 17. How can you encrypt data on EBS volumes?

You can encrypt EBS volumes using Amazon EBS encryption. You can choose to create encrypted volumes during instance launch or encrypt existing unencrypted volumes. Amazon EBS encryption uses AWS KMS key to encrypt volumes.

#### 18. How can you back up your EC2 instances?

You can create snapshots of EBS volumes, which serve as backups. These snapshots can be used to create new EBS volumes or restore existing ones. Alternatively, you can also create an image of your instance if you installed and configured a lot of softwares manually after launching your instance, creating an image would enable you to launch from that image without needing to reinstall and reconfigure everything.

#### 19. What is the difference between instance store and EBS-backed instances?

Instance store instances use ephemeral storage that is directly attached to the instance, providing high I/O performance. EBS-backed instances use EBS volumes for storage, offering persistent data storage.

#### 20. What are instance metadata and user data in EC2?

Instance metadata provides information about an instance, such as its IP address, instance type, and IAM role. User data is information that you can pass to an instance during launch to customize its behavior.

### 21. How can you launch instances in a Virtual Private Cloud (VPC)?

When launching instances, you can choose a specific VPC and subnet. This ensures that the instances are launched within the defined network environment.

### 22. What is the purpose of an EC2 security group?

An EC2 security group acts as a virtual firewall for instances to control inbound and outbound traffic. You can specify rules to allow or deny traffic based on IP addresses and ports.

### 23. How can you automate the deployment of EC2 instances?

You can use AWS CloudFormation to create and manage a collection of related AWS resources, including EC2 instances. This allows you to define the infrastructure as code.

### 24. How can you achieve high availability for an application using EC2?

You can use features like Amazon EC2 Auto Scaling and Elastic Load Balancing to distribute incoming traffic and automatically adjust the number of instances to handle changes in demand.

### 25. What is Amazon EC2 Instance Connect?

Amazon EC2 Instance Connect provides a simple and secure way to connect to your instances using Secure Shell (SSH). It eliminates the need to use key pairs and allows you to connect using your AWS Management Console credentials.

### 26. What is an EC2 instance profile, and how is it different from IAM roles?

- An EC2 instance profile is a container for an IAM role that you can use to pass role information to an EC2 instance when the instance starts.
- IAM roles are general permissions sets for AWS services, while an EC2 instance profile specifically refers to the IAM role attached to an EC2 instance

### 27. What are the differences between NAT Gateways and NAT Instances?

NAT Gateway	NAT Instances
NAT Gateways are a managed service provided by AWS. They are fully managed, meaning that AWS takes care of the maintenance, high availability, and scaling aspects of the service.	NAT Instances are EC2 instances configured to perform Network Address Translation. As such, they require manual configuration, monitoring, and management. The responsibility for the instance's availability and scaling falls on the user.

NAT Gateways are designed to scale automatically based on the traffic requirements	The scalability of NAT Instances depends on the instance type chosen and the user's management of the instances
NAT Gateways are highly available within an Availability Zone (AZ).	High availability for NAT Instances requires deploying them in an Auto Scaling Group across multiple AZs, setting up custom scripts or automation for failover, and managing Elastic IP addresses.
NAT Gateways automatically allocate and manage Elastic IP addresses. Users don't need to associate or disassociate Elastic IPs manually.	Users need to manually associate Elastic IP addresses with NAT Instances. Elastic IPs provide a static public IP address for the instances.
NAT Gateways are designed for high performance and low-latency	The performance of NAT Instances depends on the instance type chosen. Users need to consider the CPU and network performance characteristics of the instance type
NAT Gateways do not have associated security groups. They operate at the subnet level, and you control access using route tables.	NAT Instances have associated security groups, and users must configure security group rules to control inbound and outbound traffic.

28. What is the maximum limit of elastic IPs anyone can produce?

A maximum of five elastic IP addresses can be generated per location and AWS account.

# Storage Services ~ Simple (Object)

## Storage Service ( S3 ), EBS

### 1. What are the storage services provided by AWS?

Storage Type	Services	Use Cases
Object Storage	<b>S3</b> – Simple Object Storage Service  Object = Data + Metadata + Unique Identifier  Scalable, Durable, and Low-latency object storage designed for storing and retrieving any amount of data	Web hosting, backup and restore, big data analytics, mobile and gaming applications
Block Storage	<b>EBS</b> (Elastic Block Store)  Provides persistent block-level storage volumes for use with Amazon EC2 instances, suitable for databases, file systems, and applications requiring block storage.	Databases, Transactional workloads, Boot volumes for EC2 instances
File Storage	Amazon <b>EFS</b> (Elastic File System), Amazon <b>FSx for Windows</b> , Amazon <b>FSx for Lustre</b>  Fully managed file storage services that support scalable file systems, accessible across multiple EC2 instances	Shared file storage for Linux workloads, Windows file shares, high-performance file systems.
Backup and Archive	Amazon Glacier, Amazon S3 Glacier Deep Archive  Low-cost storage services designed for data archival and long-term retention	Archival storage, backup and restore, compliance data storage
Hybrid Cloud Storage	AWS Storage Gateway  Connects on-premises environments to AWS storage services, providing file, volume, and tape-based storage interfaces	Hybrid cloud storage, Data synchronization, Disaster recovery
Relational Database Storage	Amazon RDS (Relational Database Service)	Managed relational databases,

	Offers various storage engines for relational databases, including Amazon Aurora, MySQL, PostgreSQL, MariaDB, Oracle, and Microsoft SQL Server	OLTP and OLAP workloads
Data Transfer Acceleration	Amazon S3 Transfer Acceleration  Enables fast, secure, and efficient transfers of files over the internet using Amazon CloudFront's global edge locations	Accelerated data uploads and downloads to/from Amazon S3
Backup Snapshots	Amazon EBS Snapshots  Allows the creation of point-in-time snapshots of Amazon EBS volumes for backup and disaster recovery.	Data backup, Volume recovery, Data versioning

## 2. What are the key features of Amazon S3?

Amazon S3 (Simple Storage Service) is a highly scalable and durable object storage service. Key features and characteristics of Amazon S3:

- **Object Storage:** Amazon S3 is an object storage service, meaning that it stores data as objects. Each object consists of data, a key (unique within a bucket), and metadata.
- **Scalability:** S3 is designed to scale horizontally, allowing you to store an unlimited amount of data. It can handle large-scale applications and support high request rates
- **Durability and Availability:** S3 is designed for 99.999999999% (11 nines) durability. It achieves this by automatically replicating data across multiple facilities within an AWS region. S3 also provides high availability, ensuring that your data is accessible even in the event of hardware failures.
- **Data Lifecycle Management:** S3 allows you to define lifecycle policies to automatically transition objects between storage classes (e.g., Standard, Intelligent-Tiering, Glacier) based on specific criteria such as age or access patterns. This helps optimize costs by moving less frequently accessed data to more cost-effective storage classes.
- **Security and Access Control:** S3 provides robust security features, including access control lists (ACLs) and bucket policies. You can use AWS Identity and Access Management (IAM) to manage permissions for users and applications accessing S3 buckets. Additionally, S3 supports server-side encryption to encrypt data at rest.

- **Versioning:** S3 supports versioning, allowing you to preserve, retrieve, and restore every version of every object stored in a bucket. This feature is useful for data recovery, backup, and compliance requirements.
- **Event Notifications:** S3 can generate events based on operations within your buckets (e.g., object creation, deletion). You can configure event notifications to trigger AWS Lambda functions, SQS queues, or SNS topics, enabling automated workflows based on changes in your S3 data.
- **Cross-Region Replication (CRR):** S3 supports cross-region replication, allowing you to replicate objects between different AWS regions. This is useful for data redundancy, disaster recovery, or ensuring low-latency access to data in different geographic locations.
- **Static Website Hosting:** S3 can be used to host static websites by enabling website hosting on a bucket. This is a cost-effective way to serve static content, such as HTML, CSS, and images, directly from S3.
- **Multipart Uploads: For large objects,** S3 supports multipart uploads, allowing you to upload parts of an object in parallel. This improves efficiency and resiliency during the upload process.
- **Query-in-Place with Amazon S3 Select:** S3 Select allows you to run SQL queries directly against the data stored in S3, extracting only the needed data, and reducing the amount of data transferred.
- **Integration with AWS Ecosystem:** S3 is integrated with other AWS services, making it a foundational component for building a variety of applications, including data lakes, analytics, machine learning, and more.

### 3. What is an S3 bucket?

S3 bucket is a container for storing objects. It is similar to a directory or folder that holds objects such as files, images, videos, or any other type of data. Each object in an S3 bucket is assigned a unique key within that bucket.

- S3 bucket names must be globally unique across all of AWS
- All objects in a bucket share the same namespace. The combination of the bucket name and object key forms the unique identifier for each object within an S3 bucket.
- Access to S3 buckets and objects is controlled through bucket policies, Access Control Lists (ACLs), and AWS Identity and Access Management (IAM) policies.
- S3 buckets support data lifecycle management
- S3 buckets support versioning, enabling you to preserve, retrieve, and restore every version of every object stored in the bucket
- S3 buckets can be configured to generate access logs, which capture details about requests made to the bucket. These logs can be useful for auditing and analyzing access patterns

- S3 buckets support event notifications, allowing you to trigger AWS Lambda functions, SQS queues, or SNS topics based on specific events such as object creation, deletion, or replication

#### 4. What is a pre-signed URL in S3?

A pre-signed URL in Amazon S3 is a URL that provides temporary access to a specific S3 object, allowing users or applications to download or upload the object without the need for AWS security credentials. Pre-signed URLs are useful when you want to share access to an S3 object for a limited time without exposing your AWS security credentials.

##### **Here's how pre-signed URLs work:**

- **Generating a Pre-signed URL:** The owner of an S3 bucket (usually an AWS account holder) generates a pre-signed URL for a specific S3 object using the AWS SDKs, AWS CLI, or AWS Management Console.
- **Expiration Time:** When generating a pre-signed URL, the owner sets an expiration time for the URL. After this time, the URL becomes invalid, and the temporary access it provides is revoked.
- **Limited Permissions:** The pre-signed URL contains a cryptographic signature that includes the specific HTTP method (e.g., GET or PUT) allowed and the expiration time. It does not expose the AWS security credentials of the bucket owner.
- **Access for Third Parties:** The owner can share the pre-signed URL with third parties, such as users or applications, granting them temporary and controlled access to the specified S3 object.
- **Operations Supported:** Depending on how the pre-signed URL is generated, it can be used for various operations, such as:
  - GET: Allowing the recipient to download the object.
  - PUT: Allowing the recipient to upload a new version of the object.

##### **Use Cases:**

Common use cases for pre-signed URLs include:

- Allowing temporary access to private S3 objects for specific users or applications.
- Enabling time-limited, secure downloads or uploads without sharing AWS credentials.
- Facilitating temporary access for clients that don't have AWS credentials but need to perform S3 operations.

awscli example:

```
aws s3 presign s3://your-bucket/your-object --expires-in 3600
```

5. What is the difference between S3 Standard, S3 Intelligent-Tiering, and S3 One Zone-IA storage classes?

S3 Standard	S3 Intelligent- Tiering	S3 One Zone-IA
Ideal for frequently accessed data with high performance requirements.	Automatically adjusts to changing access patterns and provides a cost-effective solution for data with unpredictable usage.	Offers cost savings by storing data in a single availability zone, best suited for non-critical, easily reproducible, or infrequently accessed data

6. How can you optimize costs in Amazon S3?

Best practices for optimizing costs in Amazon S3:

- Choose the Right Storage Class: Select the appropriate S3 storage class based on the access patterns and requirements of your data. For example:
  - Use S3 Standard for frequently accessed data with low-latency requirements.
  - Use S3 Intelligent-Tiering for data with changing or unknown access patterns.
  - Use S3 One Zone-IA for infrequently accessed, non-critical data that can be easily recreated.
- Implement Data Lifecycle Policies: Define lifecycle policies to automatically transition objects between storage classes or delete them when they are no longer needed. This helps optimize costs by aligning storage costs with the usage patterns of your data.
- Enable Versioning Wisely: Use versioning in S3 only if necessary, as it can incur additional costs. Versioning allows you to preserve, retrieve, and restore every version of every object stored in a bucket.
- Optimize Data Transfer: Leverage AWS Direct Connect or AWS Snowball for large-scale data transfers to reduce data transfer costs. Consider using S3 Transfer Acceleration for faster uploads to S3 by utilizing the Amazon CloudFront global network.
- Monitor and Analyze Usage: Use Amazon S3 metrics, AWS CloudWatch, and AWS Cost Explorer to monitor and analyze your S3 usage. Identify trends, access patterns, and potential areas for optimization.



- **Configure Access Controls:** Implement fine-grained access controls using bucket policies, IAM policies, and Access Control Lists (ACLs) to restrict access to only authorized users and applications. This helps prevent unauthorized usage and potential cost increases.
- **Utilize S3 Object Tagging:** Use object tagging to categorize and label your objects. Tags can be used to track and manage costs, making it easier to allocate expenses to specific projects or departments.
- **Explore Reserved Capacity:** If you have predictable and steady-state storage needs, consider using S3 Reserved Capacity pricing. This allows you to commit to a specific amount of storage for one or three years at a discounted rate.
- **Implement Multipart Uploads:** For large objects, use multipart uploads to parallelize and accelerate the upload process. This can help reduce costs by optimizing data transfer.
- **Leverage S3 Select and Glacier for Archival:** Use Amazon S3 Select to run SQL queries directly against the data stored in S3, extracting only the needed data and reducing the amount of data transferred. For long-term archival, consider using Amazon S3 Glacier for cost-effective storage.
- **Consider Transfer Acceleration:** If your users are geographically distributed, consider using Amazon S3 Transfer Acceleration to speed up uploads to S3 by leveraging the CloudFront global network.
- **Review and Optimize Access Patterns:** Understand the access patterns of your data and optimize your storage strategy accordingly. For example, if certain data is infrequently accessed, consider transitioning it to a lower-cost storage class.

#### 7. What is the AWS Snowball device?

The AWS Snowball is a physical data transport solution used for migrating large amounts of data into and out of AWS. It's ideal for scenarios where the network transfer speed is not sufficient.

#### 8. What is Amazon S3 Select?

**Amazon S3 Select** allows you to run SQL-like queries directly against the data stored in S3, without the need to retrieve the entire object.

Key features of Amazon S3 Select include:

- **SQL-Like Query Language:** S3 Select supports a SQL-like query language that allows you to express complex filters, projections, and transformations on your data.
- **Selective Data Retrieval:** With S3 Select, you can retrieve only the data that is relevant to your query. This selective data retrieval can significantly

reduce the amount of data transferred over the network, leading to faster query execution.


- **Integration with Standard SQL Tools:** The SQL syntax used in S3 Select is similar to standard SQL, making it familiar to users who are accustomed to working with relational databases. This enables easy integration with standard SQL tools and applications.
- **Support for CSV, JSON, and Parquet Formats:** S3 Select supports querying data stored in CSV, JSON, and Parquet formats. You can use it to process and analyze structured data without the need to retrieve and process the entire object.
- **Parallel Processing:** S3 Select leverages parallel processing to accelerate query execution. This is particularly beneficial for large datasets, as it allows for efficient parallelization of data processing.
- **Data Compression and Encryption:** S3 Select works seamlessly with compressed and encrypted data. You can run queries on compressed files without the need to decompress them first, reducing both storage costs and query execution time.

#### Use Cases:

- Analyzing log files stored in S3 to extract specific information.
- Extracting insights from large datasets without the need to download the entire dataset.
- Filtering and transforming data in real-time for downstream processing.
- Integration with AWS SDKs and APIs: S3 Select is accessible through AWS SDKs and APIs, allowing developers to integrate it into their applications and workflows seamlessly.
- Cost Savings: By using S3 Select to retrieve only the relevant portions of your data, you can achieve cost savings on data transfer and processing, especially for large datasets.

#### 9. Give an example of S3 select

For example if following csv is stored on S3

CSV on S3	Requirement	S3 Select query example
 <pre> id,name,age 1,John,25 2,Jane,30 3,Bob,22 4,Alice,28 5,Charlie,35 </pre> <p>Sample.csv</p>	Retrieve only the rows where the age is greater than or equal to 30	<pre> aws s3api select-object-content \ --bucket your-s3-bucket-name \ --key sample.csv \ --expression "SELECT * FROM s3object s WHERE s.age &gt;= '30'" \ --expression-type 'SQL' \ --input-serialization '{"CSV": {"FileHeaderInfo": </pre>

		"Use"}}' \ --output-serialization '{"CSV": {}' \ --output text
--	--	---

#### 10. What is the difference between Amazon S3 and Amazon EBS?

Amazon S3	Amazon EBS
S3 is an object storage service designed for scalable and durable storage of large amounts of unstructured data, such as images, videos, log files, backups, and other types of files. It is suitable for data storage that is accessed over the internet.	EBS is a block storage service designed for providing persistent and high-performance block-level storage volumes for use with EC2 instances. It is used as a storage device for the operating system, applications, and data that require low-latency access.
S3 is accessed over HTTP/HTTPS and is commonly used for web-based object storage	EBS volumes are block-level storage devices that are attached to EC2 instances. They are accessed as block devices and are suitable for use as root volumes or additional data volumes for EC2 instances
S3 is designed for object storage, where each object consists of *data, **a key (unique within a bucket), and ***metadata. Objects can be of varying sizes and are stored in a flat namespace	EBS provides block-level storage volumes that are presented as raw block devices to EC2 instances. EBS volumes are organized into snapshots, and they provide a file system at the EC2 instance level.
S3 is optimized for high durability and scalability, but it may have higher latency compared to local block storage. It is well-suited for storing large amounts of data that do not require low-latency access.	EBS volumes offer low-latency access and high throughput, making them suitable for applications that require fast and consistent I/O performance, such as databases and transactional workloads

#### 11. How can you enable server access logging in Amazon S3?

Enabling server access logging in Amazon S3 allows you to capture detailed records for requests made to your S3 buckets. Logs are stored in a target bucket that could be the same bucket or different.

Steps to enable server access logging for an Amazon S3 bucket:

- AWS Management Console- Go to bucket properties from S3 and console and enable server access logging, configure target bucket
- AWS CLI –  

```
aws s3api put-bucket-logging --bucket your-source-bucket \
  --server-side-encryption "aws:kms" \
  --logging-configuration '{"logFilePrefix": "logs/", "destinationBucketName": "your-target-bucket", "logFilePrefix": "logs/"}'
```
- Can use AWS SDK to enable it too

## 12. How can you replicate data between S3 buckets within the same region?

Use S3 Cross-Region Replication (CRR) or S3 Same-Region Replication (SRR).

Both mechanisms allow you to automatically replicate objects across buckets, but SRR specifically addresses scenarios where replication is needed within the same AWS region.

Here are the steps to configure Same-Region Replication:

- Using the AWS Management Console:
  - Navigate to the Amazon S3 Console, Choose the S3 bucket that contains the data you want to replicate.
  - Navigate to "Management" (SRR): In the bucket details page, select the "Management" tab.
  - Enable Same-Region Replication:
  - Click on "Replication" and then "Create replication rule."
  - Choose "Same-region replication" as the rule type.
  - Configure Replication Rule:
  - Configure the replication rule by specifying the source and destination buckets, as well as any filtering criteria if needed.
  - Review and Save

### CLI Example:

```
aws s3api put-bucket-replication --bucket your-source-bucket \
  --replication-configuration '{"Role": "arn:aws:iam::123456789012:role/ReplicationRole", "Rules": [{"Status": "Enabled", "Prefix": "", "Destination": {"Bucket": "arn:aws:s3:::your-destination-bucket"}}]}'
```

```
aws s3api put-bucket-replication --bucket your-source-bucket \
```

## 13. How many S3 buckets can be created?

Up to 100 buckets can be created by default{"Bucket": "arn:aws:s3:::your-

# AWS Lambda

## 1. What is AWS Lambda? How does it work?

AWS Lambda is a serverless computing service provided by Amazon Web Services (AWS). It enables you to run code without provisioning or managing servers. With Lambda, you can execute code in response to events and automatically manage the compute resources required for that code

- **Serverless Architecture:** Lambda automatically provisions, manages and scales the infrastructure.
- **Event Driven :** Lambda functions are triggered by events such as changes to data in an Amazon S3 bucket, updates to a DynamoDB table, or HTTP requests via API Gateway
- **Pay-as-You-Go Pricing:** only pay for the compute time that you consume, measured in milliseconds
- **Multilanguage Support:** Lambda supports various programming languages, including Node.js, Python, Java, Go, Ruby, and .NET Core
- **Built-In Fault Tolerance:** Lambda automatically replicates functions across multiple availability zones to provide high availability and fault tolerance

### Working with Lambda (Step by Step)

- Develop and package your code along with all its dependencies as a Lambda function
- Specify event sources that will trigger the Lambda function. This can include changes to objects in an S3 bucket, updates to a DynamoDB table, or API Gateway requests
- Upload the Lambda function code and any associated artifacts to AWS Lambda
- Create an execution role that grants necessary permissions for the Lambda function to interact with other AWS services
- Connect the Lambda function to event sources by configuring triggers. Events will invoke the Lambda function automatically
- When an event occurs (e.g., new data in S3), Lambda triggers the function, which runs in a containerized environment
- Lambda automatically scales by running multiple copies (instances) of the function in response to high volumes of events.
- AWS Lambda automatically manages the compute resources, allocates memory, and executes the function code

Lambda provides logs and monitoring metrics through AWS CloudWatch, allowing you to troubleshoot and monitor the performance of your functions

## Lambda Use Cases

- Real Time File processing: Process and analyze data as soon as it's uploaded to S3
- Automate repetitive tasks, such as resizing images or cleaning up old data
- Handle and analyze data from Internet of Things (IoT) devices
- Build and deploy serverless APIs and microservices using API Gateway and Lambda

### 2. What is the maximum execution duration for a single AWS Lambda invocation?

The maximum execution duration for a single Lambda invocation is 15 minutes.

### 3. How do you pass data to and from AWS Lambda functions?

You can pass data to Lambda functions through event objects, which contain information about the triggering event. You can also return data by using the return statement or creating a response object.

### 4. Can AWS Lambda functions communicate with external resources?

Yes, Lambda functions can communicate with external resources such as databases, APIs, and other AWS services by using appropriate SDKs and APIs provided by AWS.

### 5. What are AWS Lambda layers?

AWS Lambda layers are a way to manage and share code that is common across multiple functions. Layers can include libraries, custom runtimes, and other function dependencies.

### 6. How can you handle errors in AWS Lambda functions?

You can handle errors by using try-catch blocks in your code. Lambda also provides CloudWatch Logs for monitoring, and you can set up error handling and retries for asynchronous invocations.

### 7. Can AWS Lambda functions access the internet?

Yes, Lambda functions can access the internet through the Virtual Private Cloud (VPC) or through public endpoints if your function is not configured within a VPC.

### 8. How can you configure environment variables for AWS Lambda functions?

You can set environment variables for Lambda functions when creating or updating the function. These variables can be accessed within your code.

9. What is the difference between synchronous and asynchronous invocation of Lambda functions?

Synchronous invocations wait for the function to complete and return a response, while asynchronous invocations return immediately, and the response is sent to a specified destination.

10. What is the AWS Lambda Event Source Mapping?

Event Source Mapping allows you to connect event sources like Amazon DynamoDB streams or Amazon Kinesis streams to Lambda functions. This enables the function to process events as they occur.

11. How can you manage the permissions and execution roles for AWS Lambda functions?

You can use AWS Identity and Access Management (IAM) roles to grant permissions to your Lambda functions. Execution roles define what AWS resources the function can access.

12. What is AWS Step Functions?

AWS Step Functions is a serverless orchestration service that lets you coordinate multiple AWS services into serverless workflows using visual workflows called state machines.

13. How can you automate the deployment of AWS Lambda functions?

You can use AWS Serverless Application Model (SAM) templates, AWS CloudFormation, or CI/CD tools like AWS CodePipeline to automate the deployment of Lambda functions.

14. Can AWS Lambda functions connect to on-premises resources?

Yes, Lambda functions can connect to on-premises resources by placing the function inside a VPC and using a VPN or Direct Connect connection to establish connectivity.

15. What is the Cold Start issue in AWS Lambda?

The Cold Start issue occurs when a Lambda function is invoked for the first time or after it has been idle for a while. The function needs to be initialized, causing a slight delay in response time.

# Advanced Topics

1. What are the tools and techniques that you can use in AWS to identify if you are paying more than you should be, and how to correct it?

## Tools Provided:

- **AWS Cost Explorer:** AWS Cost Explorer is a powerful tool for visualizing, understanding, and managing your AWS costs. It provides a range of predefined and customizable reports to analyze costs based on various dimensions, such as services, accounts, and time periods.
- **AWS Budgets:** AWS Budgets allows you to set custom cost and usage budgets that alert you when you exceed your thresholds. It helps you proactively monitor and control your AWS spending.
- **Trusted Advisor:** AWS Trusted Advisor provides recommendations across various categories, including cost optimization. It analyzes your AWS environment and suggests ways to save money, improve performance, and close security gaps.
- **Cost and Usage Reports (CUR):** Cost and Usage Reports provide detailed data on your AWS usage and costs. You can enable CUR to get granular information about resource usage, pricing, and incurred costs, which can be exported to Amazon S3 and analyzed using tools like AWS Athena or external business intelligence tools.
- **AWS Pricing Calculator:** The AWS Pricing Calculator allows you to estimate and compare costs for different AWS services and configurations. It helps you understand how changes to your infrastructure impact costs before implementing them.

## Techniques:

- **Tagging Strategies:** Implement a robust tagging strategy for your AWS resources. Tags can be used to categorize resources, making it easier to allocate costs to specific projects, departments, or environments. Cost Explorer and Budgets can then be filtered based on tags.
- **Reserved Instances (RIs) Planning:** Evaluate your EC2 instance usage patterns and consider purchasing Reserved Instances for instances with steady, predictable workloads. RIs offer significant cost savings over On-Demand instances.
- **Spot Instances and Savings Plans:** Leverage Spot Instances for fault-tolerant and flexible workloads that can handle interruptions. Savings Plans provide significant savings for commitment-based, consistent usage across various AWS services.
- **Right-Sizing:** Regularly review your EC2 instances, EBS volumes, and other resources to ensure they are appropriately sized. Use AWS tools like AWS Compute Optimizer to get recommendations on right-sizing EC2 instances.



- **Terminate Unused Resources:** Identify and terminate unused or underutilized resources. This includes instances, EBS volumes, and other services. Use tools like AWS Config to track resource changes and identify unused resources.
- **Monitor Data Transfer Costs:** Be mindful of data transfer costs, especially if you have data-intensive workloads. Consider using AWS Direct Connect for predictable, lower-cost data transfer between your on-premises environment and AWS.
- **Analyzing Spot Fleet Savings:** For workloads that can leverage Spot Instances, consider using Spot Fleet to diversify across multiple Spot Instance types and Availability Zones to maximize cost savings.
- **AWS Marketplace Subscriptions:** Review your AWS Marketplace subscriptions regularly. Unused or unnecessary subscriptions can contribute to additional costs.
- **Review and Optimize CloudFormation Templates:** If you use AWS CloudFormation, review your templates to ensure they are optimized. Unused or overprovisioned resources in templates can contribute to unnecessary costs.
- **Educate and Train Teams:** Ensure that your teams are educated on AWS best practices and cost optimization strategies. Regular training can empower them to make cost-conscious decisions.

2. [What services can be used to create a centralized logging solution?](#)

The essential services that you can use are Amazon CloudWatch Logs, store them in Amazon S3, and then use Amazon Elastic Search to visualize them. You can use Amazon Kinesis Firehose to move the data from Amazon S3 to Amazon ElasticSearch

### 3. How to create a centralized logging solution in AWS?

Creating a centralized logging solution in AWS involves collecting logs from multiple sources, aggregating them in a centralized location, and providing tools for analysis, monitoring and visualization.

1	Choose a log aggregation service	<p>Select a service that can aggregate logs from various AWS services and applications. Two commonly used services for this purpose are:</p> <p><b>Amazon CloudWatch Logs:</b></p> <ul style="list-style-type: none"><li>• Centralized logging service in AWS.</li><li>• Collects logs from AWS services, EC2 instances, and custom applications.</li><li>• Provides storage, search, and analysis capabilities.</li><li>• Allows you to create log groups and log streams for organizing logs</li></ul> <p><b>AWS CloudTrail:</b></p> <ul style="list-style-type: none"><li>• Records API calls made on your account.</li><li>• Captures events related to AWS resources.</li><li>• Useful for auditing and security-related logging</li></ul>
2	Configure Log Streams	<p>Set up log streams within your chosen log aggregation service to organize logs based on sources, applications, or components.</p> <p>In CloudWatch Logs:</p> <ul style="list-style-type: none"><li>• Create log groups for different applications or services</li></ul> <p>In CloudTrail:</p> <ul style="list-style-type: none"><li>• Enable logging and configure the S3 bucket where logs will be stored</li></ul>
3	Integrate with AWS Services	<p>Integrate AWS services and resources with the chosen log aggregation service.</p> <p>For CloudWatch Logs:</p> <ul style="list-style-type: none"><li>• Configure AWS services (e.g., EC2, Lambda) to send logs to CloudWatch Logs.</li><li>• Install and configure the CloudWatch agent on EC2 instances for custom logs</li></ul> <p>For CloudTrail:</p> <ul style="list-style-type: none"><li>• Enable CloudTrail for your AWS account.</li></ul>

		<ul style="list-style-type: none"> <li>• Configure CloudTrail to deliver logs to CloudWatch Logs or an S3 bucket</li> </ul>
4	Configure Custom Logs	<p>For applications running on EC2 instances or Lambda functions, configure custom logs to be sent to CloudWatch Logs</p> <ul style="list-style-type: none"> <li>• Install and configure the CloudWatch agent on EC2 instances.</li> <li>• Use the AWS SDK or CloudWatch Logs agent to send logs from Lambda functions</li> </ul>
5	Visualization and Analysis	<ul style="list-style-type: none"> <li>• Use CloudWatch Metrics and Alarms for monitoring and alerting.</li> <li>• Explore CloudWatch Insights for querying and analyzing logs interactively.</li> <li>• Consider using AWS services like Amazon Elasticsearch, Amazon Athena, or third-party solutions for advanced log analysis and visualization</li> </ul>

#### 4. What are the native AWS Security logging capabilities?

Most of the AWS services have their logging options.

Also, some of them have an account level logging, like in AWS CloudTrail, AWS Config

- **AWS CloudTrail**

This is a service that provides a history of the AWS API calls for every account. It lets you perform security analysis, resource change tracking, and compliance auditing of your AWS environment as well. The best part about this service is that it enables you to configure it to send notifications via AWS SNS when new logs are delivered.

- **AWS Config**

This helps you understand the configuration changes that happen in your environment. This service provides an AWS inventory that includes configuration history, configuration change notification, and relationships between AWS resources. It can also be configured to send information via AWS SNS when new logs are delivered.

#### 5. What is a DDoS attack, and what services can minimize them?

DDoS is a cyber-attack in which the perpetrator accesses a website and creates multiple sessions so that the other legitimate users cannot access the service. The native tools that can help you deny the DDoS attacks on your AWS services are:

- AWS Shield
- AWS WAF
- Amazon Route53
- Amazon CloudFront
- ELB
- VPC

6. How do you set up a system to monitor website metrics in real-time in AWS?

Amazon CloudWatch helps you to monitor the application status of various AWS services and custom events. It helps you to monitor:

- State changes in Amazon EC2
- Auto-scaling lifecycle events
- Scheduled events
- AWS API calls
- Console sign-in events

7. Name some of the AWS services that are not region-specific

AWS services that are not region-specific are:

- IAM
- Route 53
- Web Application Firewall
- CloudFront

8. What are the elements of an AWS CloudFormation template?

AWS CloudFormation templates are YAML or JSON formatted text files that are comprised of five essential elements, they are:

- Template parameters
- Output values
- Data tables
- Resources
- File format version

### 9. What are the different types of load balancers in AWS?

There are three types of load balancers that are supported by Elastic Load Balancing:

- Application Load Balancer :
  - When you need a flexible feature set for your applications with HTTP and HTTPS traffic
  - Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers
- Network Load Balancer
  - When you need ultra-high performance
  - TLS offloading at scale
  - Centralized certificate deployment
  - Support for UDP, and static IP addresses for your applications
  - Capable of handling millions of requests per second securely while maintaining ultra-low latencies
  -
- Classic Load Balancer (legacy)
  - Use if application is built within EC2 classic network
- Gateway Load Balancer:
  - When you need to deploy and manage a fleet of third-party virtual appliances that support GENEVE
  - These appliances enable you to improve security, compliance, and policy controls

### 10. Can AWS Config aggregate data across different AWS accounts?

Yes, you can set up AWS Config to deliver configuration updates from different accounts to one S3 bucket, once the appropriate IAM policies are applied to the S3 bucket.

### 11. How do you troubleshoot issues with an application that's running into AWS cloud environment?

Troubleshooting issues with an application running on AWS infrastructure involves a systematic approach to identify and resolve problems effectively. Here's a general guide to help you troubleshoot AWS-based applications:

Logs and Monitoring:	<p><b>CloudWatch Logs:</b> Check CloudWatch Logs for your application's logs. Investigate error messages or unusual patterns.</p> <p><b>CloudWatch Metrics:</b> Monitor relevant CloudWatch metrics for AWS resources (e.g., EC2 instances, RDS databases) to identify any resource-related issues.</p> <p><b>AWS X-Ray:</b> If your application is instrumented with AWS X-Ray, use it to trace and diagnose issues in distributed applications.</p>
Infrastructure Status:	Check the <b>AWS Service Health Dashboard</b> to ensure that there are no ongoing service disruptions or incidents in the AWS region where your resources are deployed.
Network Connectivity	<p>Examine <b>VPC Flow Logs</b> to identify any unusual network traffic patterns or connectivity issues.</p> <p>Verify the security group and network ACL settings for your resources to ensure that the necessary ports are open.</p>
Resource Utilization	Set up <b>CloudWatch Alarms</b> to notify you if certain resource metrics (CPU utilization, memory usage) exceed predefined thresholds.
Instance and Container Insights	For EC2 instances and ECS containers, enable detailed monitoring and review insights provided by AWS Systems Manager or Container Insights.
Database Performance	For AWS managed databases (RDS, Aurora), check database-specific metrics. Use the AWS RDS Performance Insights for detailed analysis.
Security and Access Controls	Review AWS Identity and Access Management (IAM) policies to ensure that your application has the necessary permissions. Check AWS Key Management Service (KMS) for any issues related to encryption keys.
Deployment and Configuration	Ensure that your deployment scripts and configurations are correct. Tools like AWS CloudFormation or AWS CDK can help with managing infrastructure as code.

	Check for any recent changes to configurations that might have introduced issues.
Error Responses	Examine error responses returned by your application to identify the root cause of issues. Use error messages to guide your investigation.
Third-Party Services	If your application relies on third-party services, check their status and logs. Ensure that integrations are working as expected.
AWS Support	If the issue persists and you can't identify the root cause, consider reaching out to AWS Support for assistance. Provide detailed information about the issue and any relevant logs.
Documentation and Best Practices	Consult AWS documentation and best practices for the specific services you are using. This can provide guidance on common issues and their resolutions.
Community and Forums	AWS forums and community resources can be valuable for seeking advice and solutions from others who may have encountered similar issues.

Remember to approach troubleshooting systematically, starting with the most likely causes and progressively narrowing down the possibilities. Document your steps and findings as you go to facilitate communication with other team members or AWS support if needed.

## 12. What is the difference between Amazon RDS and Amazon DynamoDB?

Amazon RDS	Amazon DynamoDB
<b>RDS</b> is a <b>relational</b> database service that supports various relational database engines, such as MySQL, PostgreSQL, Oracle, Microsoft SQL Server, and MariaDB. RDS is suitable for applications that require a traditional relational database structure.	<b>DynamoDB</b> is a <b>NoSQL</b> database service, offering a fully managed, serverless, and scalable database. DynamoDB is designed for fast and predictable performance on large amounts of unstructured or semi-structured data.
It uses a traditional relational data model with tables, rows, and columns. It supports SQL queries and transactions and is suitable for applications with complex relationships between data.	It is a NoSQL database, using a key-value and document data model. It is schema-less, allowing for flexible and dynamic data structures, making it suitable for applications with evolving or unpredictable data requirements.
Scaling in RDS is achieved through vertical scaling (increasing the instance size) or, in some cases, by using read replicas. Scaling can involve downtime during instance resizing.	DynamoDB offers automatic and seamless horizontal scaling. DynamoDB tables can scale up or down based on demand without any downtime. This makes it well-suited for applications with varying and unpredictable workloads.
Performance is influenced by the chosen database engine (MySQL, PostgreSQL, etc.) and the underlying hardware. Read replicas can be used to offload read queries.	DynamoDB provides consistent and single-digit millisecond latency for both read and write operations. Performance is predictable and can be adjusted by changing provisioned throughput capacity.
Suitable for e-commerce platforms, content management systems, and line-of-business applications	Suitable for gaming apps, IoT applications, and scenarios with large-scale data requirements
RDS handles routine database tasks such as backups, software patching,	DynamoDB is fully managed service, taking care of administrative tasks like



and hardware provisioning. Users have more control over the database configuration and tuning.	hardware and software maintenance, backups, and automatic scaling. Users have less control over the underlying infrastructure
--	---

### 13. Does AWS support Cassandra database?

YES, AWS provides a managed Cassandra service called Amazon Keyspaces (formerly known as Amazon Managed Apache Cassandra Service). Amazon Keyspaces is designed to simplify the management and operation of Cassandra clusters in the cloud. With Amazon Keyspaces, you can build applications using the Apache Cassandra API while offloading the operational aspects such as hardware provisioning, setup, and maintenance to AWS.

- Amazon Keyspaces allows serverless scaling based on your application's traffic
- Amazon Keyspaces can be deployed across multiple AWS regions to achieve low-latency access and high availability for global applications.
- Amazon Keyspaces is compatible with the Apache Cassandra API, which means that you can use existing Cassandra drivers and tools to interact with your Keyspaces clusters.
- Amazon Keyspaces provides encryption at rest and in transit. It integrates with AWS Identity and Access Management (IAM) for access control.

#### 14. What is the difference between DynamoDB and Amazon Keyspaces?

Amazon DynamoDB and Amazon Keyspaces are both managed NoSQL database services provided by AWS. Key differences-

<b>Amazon Dynamo-DB (AWS Proprietary NoSQL)</b>	<b>Amazon Keyspaces (Cassandra NoSQL)</b>
DynamoDB is a fully managed NoSQL database service developed by AWS. It is a proprietary database engine designed for high-performance, low-latency, and scalable applications. DynamoDB uses a key-value and document data model and is suitable for a variety of use cases, including web and mobile applications, gaming, and IoT.	Amazon Keyspaces, formerly known as Amazon Managed Apache Cassandra Service (MCS), is a managed Cassandra-compatible NoSQL database service. It is built to be compatible with Apache Cassandra, a popular open-source distributed database. Keyspaces allows you to use the Apache Cassandra API to interact with the database, making it suitable for applications with existing Cassandra code or expertise.
DynamoDB supports a key-value and document data model. It provides flexibility in defining attribute types, and its schema is dynamic, allowing you to add or remove attributes without altering the table structure.	Keyspaces is compatible with the Apache Cassandra data model, which is based on columns organized into families and rows identified by a unique key. Cassandra supports a wide range of data models, including wide-column store and time-series data.
DynamoDB provides a rich set of operations for querying and scanning data. It supports both primary key-based queries and secondary index queries.	Keyspaces uses CQL (Cassandra Query Language), which is similar to SQL and is specific to Cassandra databases. CQL allows you to interact with Keyspaces using a familiar query language.
DynamoDB offers two consistency models—eventual consistency and strong consistency. You can choose the level of consistency based on your application's requirements.	Keyspaces provides tunable consistency, allowing you to choose between eventual consistency and strong consistency for read and write operations.

## 15. What is the difference between Amazon SQS and SNS?

Amazon <b>SNS</b> (Simple Notification Service)	Amazon <b>SQS</b> (Simple Queue Service)
SNS is a publish-subscribe messaging service. It allows message publishers to send messages to multiple subscribers (endpoints) simultaneously. Subscribers can be various services or applications.	SQS is a message queuing service. It follows the point-to-point messaging model, where messages are sent to a queue by a producer and consumed by a single consumer. Each message is processed by exactly one consumer.
It is suitable for scenarios where multiple recipients need to receive the same message simultaneously. Examples include sending notifications, alerts, or updates to a large number of subscribers or endpoints (e.g., mobile devices, email addresses, HTTP endpoints).	It is designed for decoupling the components of a distributed application. SQS helps in building systems that can handle asynchronous communication and smooth out traffic spikes. It's commonly used for message buffering between components.
Supports different message formats, including plaintext, JSON, and other common formats. Publishers send messages to a topic, and SNS takes care of delivering the messages to all subscribed endpoints.	Messages are sent to and consumed from queues. Each message is a standalone unit of data, and SQS doesn't enforce a specific message format. The content of the message is entirely up to the sender and receiver.
It doesn't inherently provide retries or dead-letter queues. If a message delivery fails, SNS doesn't retain the message for future retries.	Supports redrive policies that allow you to specify a dead-letter queue for messages that couldn't be processed successfully after a certain number of attempts. This is useful for handling and analyzing failed messages.
SNS Pricing is based on the number of messages published and the number of deliveries to subscribers. Subscribers	SQS Pricing is based on the number of requests (sends, receives, etc.) and the volume of data transferred. Both

typically incur costs for the messages they receive.	producers and consumers incur costs based on their usage
Supports fanout, meaning a single message can be delivered to multiple subscribers simultaneously.	Follows a point-to-point model, and messages are consumed by a single consumer from the queue.

#### 16. What is dead-letter queues?

A dead-letter queue (DLQ) is a feature provided by messaging services, including Amazon Simple Queue Service (SQS) and others, to handle messages that, for some reason, cannot be processed successfully by the normal processing flow. When a message is deemed undeliverable or encounters repeated processing failures, it is moved to a dead-letter queue for further analysis and handling.

- Messages might become undeliverable due to various reasons, such as incorrect message format, processing errors, or exceeding the maximum number of processing attempts.
- Before a message is moved to a dead-letter queue, the messaging service typically provides a configurable retry mechanism. If a message fails to be processed, the service can attempt to deliver it again for a certain number of retries.
- The conditions under which a message is considered undeliverable and moved to the dead-letter queue are configurable. Common parameters include the maximum number of delivery attempts and a time-to-live (TTL) for the message.
- Dead-letter queues are designed to aid in troubleshooting and analysis of message processing issues. When a message is moved to the dead-letter queue, it provides an opportunity for developers and operators to investigate the root cause of the failure.
- Amazon SQS redrive policy allows you to specify a dead-letter queue for a source queue. If a message fails processing on the source queue after the specified number of retries, it is moved to the associated dead-letter queue.
- Once messages are in the dead-letter queue, they can be inspected, analyzed, and potentially reprocessed. This can involve fixing issues with the message content, updating processing logic, or taking other corrective actions.

- Dead-letter queues help prevent infinite processing loops caused by messages that consistently fail processing. The messages are eventually moved to the dead-letter queue, allowing the system to continue processing other messages.
- Dead-letter queues are an important tool in building resilient and robust distributed systems. They provide a safety net for handling exceptional cases and allow for proper analysis and resolution of issues that might otherwise disrupt the normal flow of message processing.

#### 17. What is the difference between Apache Kafka and Amazon SQS?

Apache Kafka and Amazon Simple Queue Service (SQS) are both messaging systems but have different architectures, use cases, and features.

Apache Kafka	Amazon SQS
Kafka is a distributed streaming platform that is designed for high-throughput, fault-tolerant, and scalable event streaming.  It uses a publish-subscribe model and stores messages in topics that are partitioned across multiple brokers. Kafka allows for real-time stream processing and is often used for building event-driven architectures.	SQS is a fully managed message queuing service that follows the point-to-point model. It is designed to decouple the components of a distributed system by allowing messages to be sent between different parts of the system via queues. SQS is not inherently a streaming platform.
Kafka retains messages for a configurable period, allowing consumers to replay or process historical data. The retention period is typically longer in Kafka compared to SQS.	SQS retains messages for a shorter duration (maximum of 14 days) compared to Kafka. SQS is more focused on delivering messages reliably and quickly to consumers
Kafka persists messages to disk, providing durability even in the event of broker failures. This persistence contributes to Kafka's ability to act as a	SQS is designed for high availability but doesn't persist messages for long-term storage. It's optimized for quick message delivery and short-term queuing.

long-term event log for data integration and analytics.	
Kafka is horizontally scalable and can handle large amounts of data and high-throughput scenarios. It can be scaled out by adding more brokers to the Kafka cluster	SQS is a fully managed service, and its scalability is handled by AWS. It automatically scales to accommodate the workload without the need for manual intervention.
Kafka is ideal for scenarios that require real-time event streaming, such as log aggregation, monitoring, and analytics. Kafka is commonly used in data pipelines, microservices architectures, and applications that need a distributed commit log.	SQS is suited for decoupling components of a distributed system, ensuring reliable message delivery between different parts of an application. Common use cases include task offloading, workload distribution, and building fault-tolerant systems.
Kafka typically involves managing your own Kafka cluster, which may incur infrastructure and operational costs. Open-source Kafka can be self-hosted or managed with tools like Confluent Cloud or Amazon MSK (Managed Streaming for Apache Kafka).	SQS follows a pay-as-you-go model where you pay for the number of requests and data transfer. It is fully managed, reducing operational overhead.

#### 18. What is the difference between Apache Kafka and Amazon SNS?

Apache Kafka	Amazon <b>SNS</b>
Kafka is a distributed streaming platform that is designed for high-throughput, fault-tolerant, and scalable event streaming. It uses a publish-subscribe model and stores messages in topics that are partitioned across multiple brokers. Kafka is often used for building real-time data pipelines, log aggregation, and event-driven architectures	SNS is a fully managed publish-subscribe service that allows message publishers to send messages to multiple subscribers (endpoints) simultaneously. Subscribers can include AWS services, mobile devices, email addresses, and more. SNS is not a streaming platform like Kafka but is focused on broadcasting messages to multiple subscribers.

Kafka persists messages to disk and provides configurable retention periods, allowing consumers to replay or process historical data. This makes Kafka suitable for use cases where durable, long-term storage of events is required.	SNS is designed for immediate message delivery and doesn't provide long-term message storage. It is optimized for sending messages to subscribers in real-time.
Kafka provides strong ordering guarantees within a partition, ensuring that messages are processed in the order they are produced within the same partition	SNS doesn't guarantee strict ordering of messages across multiple subscribers. While messages are delivered to subscribers in the order they are published, variations in delivery times can result in slightly different orders for different subscribers
Kafka is horizontally scalable and can handle large amounts of data and high-throughput scenarios. It can be scaled out by adding more brokers to the Kafka cluster.	SNS is a fully managed service, and its scalability is handled by AWS. It automatically scales to accommodate the workload without the need for manual intervention.
Ideal for scenarios that require real-time event streaming, such as log aggregation, monitoring, and analytics. Kafka is commonly used in data pipelines, microservices architectures, and applications that need a distributed commit log.	Suited for scenarios where messages need to be broadcast to multiple subscribers simultaneously. Common use cases include notifications, alerts, and communication to a diverse set of endpoints, such as mobile devices, email, and HTTP endpoints.
Kafka can be self-hosted, and managing a Kafka cluster involves infrastructure and operational considerations. Managed Kafka services, like Confluent Cloud or Amazon MSK, can reduce some operational overhead.	SNS is fully managed, meaning AWS takes care of infrastructure and operational aspects. Users can focus on publishing and consuming messages without managing the underlying infrastructure.

Choosing between Apache Kafka and Amazon SNS depends on the specific requirements of your application, such as the need for real-time streaming, message durability, ordering guarantees, and ease of management.

### 19. What is AWS well architected framework?

The AWS Well-Architected Framework is a set of best practices and guidelines for designing and operating reliable, secure, efficient, and cost-effective systems on Amazon Web Services (AWS). It provides a consistent approach for customers and AWS partners to evaluate architectures, and it offers guidance to help them build secure, high-performing, resilient, and efficient infrastructure for their applications.

The AWS Well-Architected Framework is built on five key pillars:

<b>Operational Excellence</b>	Focuses on operational best practices to ensure efficient day-to-day operations. This includes monitoring, incident response, automation, and evolving procedures over time
<b>Security</b>	Addresses the principles and best practices to protect data, systems, and assets. It covers areas such as data protection, identity and access management, detective controls, infrastructure protection, and incident response
<b>Reliability</b>	Focuses on the ability of a workload to recover from failures and meet customer demands. This includes topics like resiliency, availability, and fault tolerance
<b>Performance Efficiency</b>	Helps optimize performance based on workload requirements. This includes selecting the right resource types, monitoring performance, and making informed decisions to maintain efficiency as needs evolve over time
<b>Cost Optimization</b>	Focuses on avoiding unnecessary costs and ensuring that resources are used efficiently. This includes understanding and controlling where money is being spent, selecting the right resource types, and continuously optimizing over time

The framework provides a set of questions and considerations for each pillar, helping organizations assess their workloads against these best practices. AWS customers can use the Well-Architected Framework to review and improve their existing architectures or to ensure that new workloads are designed following best practices.

AWS Well-Architected Reviews, facilitated by AWS or AWS partners, are also available to help customers evaluate their architectures in alignment with the



framework. These reviews provide recommendations for improvement based on the specific workload's needs and goals.

## 21. What are the differences between Apache Kafka and AWS Kinesis?

Kafka and Amazon Kinesis are both popular distributed streaming platforms designed for handling real-time data, but they have some differences in terms of architecture, deployment, and integration. Here are some key distinctions between Kafka and Amazon Kinesis:

Apache Kafka	AWS Kinesis
Kafka is an open-source distributed streaming platform developed by the Apache Software Foundation. Users need to set up, configure, and manage Kafka clusters themselves, whether on-premises or in the cloud	Amazon Kinesis is a fully managed service provided by AWS. Users do not have to worry about the infrastructure or cluster management, as AWS takes care of the operational aspects.
Kafka allows users to deploy and manage their own clusters, providing flexibility in choosing deployment locations (on-premises, cloud, or hybrid). Users are responsible for scaling and maintaining the infrastructure.	Amazon Kinesis is a fully managed service that automatically scales based on demand. Users do not have to worry about provisioning or managing infrastructure, and the service is designed to handle scaling requirements seamlessly.
Kafka can be integrated with AWS services, but it requires additional configuration and setup. Confluent, a company that provides a distribution of Kafka, offers connectors to integrate Kafka with various AWS services.	Kinesis is tightly integrated with other AWS services, making it easy to stream data into and out of various AWS components such as S3, DynamoDB, Redshift, and Lambda.
Kafka requires more manual configuration and management, making it suitable for users who are comfortable with infrastructure management and tuning	Amazon Kinesis is designed for ease of use, with simplified setup and management. It's well-suited for users who prefer a fully managed service without the need for extensive configuration
Kafka is open source, and the cost is associated with infrastructure and operational expenses. Users need to manage their own clusters and handle associated costs.	Amazon Kinesis has a pay-as-you-go pricing model based on the number of shards, records, and data transfer. Users pay for the resources they consume without managing underlying infrastructure.

Kafka has a large and active open-source community, and it is widely adopted across various industries. It has a rich ecosystem of tools and connectors.	Amazon Kinesis benefits from integration with the broader AWS ecosystem and services. While it may not have the same open-source community as Kafka, it offers a comprehensive set of features within the AWS environment.
--	--

Ultimately, the choice between Kafka and Amazon Kinesis depends on factors such as the level of control and management preferences, integration with other AWS services, and the specific use case and requirements of the organization.

## 22. How do you design a big data application on AWS?

Designing a big data application on AWS involves careful consideration of various factors, including data storage, processing, analytics, security, and scalability. Here are the key steps and best practices for designing a big data application on AWS:

<b>Define Objectives and Requirements</b>	Clearly define the objectives and requirements of your big data application. Understand the types of data you'll be working with, the processing and analytics needs, and the desired outcomes
<b>Choose Appropriate Services</b>	Select AWS services that align with your application requirements. Common services for big data applications include Amazon S3 for storage, Amazon EMR for processing, Amazon Redshift for data warehousing, and Amazon Kinesis for real-time data streaming
<b>Data Storage and Management</b>	Consider using Amazon S3 as a scalable and durable object storage service for your data lake. Organize your data in a way that facilitates efficient querying and processing. For structured data, you might also use services like Amazon DynamoDB or Amazon RDS.
<b>Data Processing</b>	Utilize Amazon EMR for distributed processing of large datasets using frameworks like Apache Spark, Apache Hadoop, or Apache Flink. Configure clusters based on the workload and adjust the instance types accordingly
<b>Data Integration and ETL</b>	Implement ETL (Extract, Transform, Load) processes using AWS Glue or other ETL tools to prepare data for analysis. AWS Glue can automatically discover, catalog, and transform data stored in Amazon S3
<b>Real-Time Data Streaming</b>	If your application requires real-time data streaming, consider using Amazon Kinesis Data Streams for ingesting and processing streaming data. Kinesis Data Firehose can be used to load streaming data into other AWS services
<b>Data Warehousing and Analytics</b>	For data warehousing and analytics, use Amazon Redshift for high-performance querying and reporting. Connect

	visualization tools like Amazon QuickSight for business intelligence
<b>Monitoring and Logging</b>	Implement robust monitoring and logging using AWS CloudWatch, AWS CloudTrail, and additional logging tools. Monitor resource utilization, set up alarms, and use CloudWatch Dashboards for a centralized view of metrics
<b>Security</b>	Apply security best practices, such as using AWS Identity and Access Management (IAM) for access control, encrypting data at rest and in transit, and implementing fine-grained security policies. Regularly audit and review security configurations
<b>Scalability and Auto-Scaling</b>	Design your application to scale horizontally by leveraging AWS Auto Scaling. Use scalable services like Amazon S3 and Amazon EMR to handle growing volumes of data and processing demands
<b>Cost Optimization</b>	Optimize costs by selecting the right instance types, leveraging serverless services where possible, and implementing data lifecycle policies to manage storage costs effectively
<b>Resilience and High Availability</b>	Design for resilience by distributing your application across multiple Availability Zones. Use AWS services like Amazon S3 and Amazon EMR, which are designed to be highly available
<b>Documentation and Collaboration</b>	Document your architecture, configurations, and processes. Encourage collaboration among your team members by using AWS CloudFormation for infrastructure as code and version control for configurations
<b>Testing and Iteration</b>	Conduct thorough testing of your big data application, including scalability and performance testing. Iterate and refine your design based on feedback and changing requirements

By following these steps and best practices, you can design a robust, scalable, and efficient big data application on AWS.

### 23. What are big data tools provided by AWS?

Amazon Web Services (AWS) provides a comprehensive set of tools and services for handling big data workloads.

<b>Amazon EMR (Elastic MapReduce)</b>	A cloud-based big data platform that enables processing of large amounts of data using popular frameworks such as Apache Spark, Apache Hive, Apache Hadoop, and more.
<b>Amazon Redshift</b>	A fully managed data warehouse service that allows users to run complex queries and analyze large datasets with high-performance SQL queries.
<b>Amazon Athena</b>	A serverless query service that enables users to analyze data stored in Amazon S3 using standard SQL without the need to set up and manage servers
<b>Amazon Glue</b>	A fully managed ETL (Extract, Transform, Load) service that makes it easy to prepare and load data for analysis
<b>Amazon Kinesis</b>	A platform for real-time data streaming and analytics, consisting of services like Kinesis Data Streams, Kinesis Data Firehose, and Kinesis Data Analytics
<b>Amazon S3 (Simple Storage Service)</b>	A highly scalable and durable object storage service, often used as a data lake for storing and retrieving any amount of data
<b>AWS Glue DataBrew</b>	A visual data preparation tool that helps clean, normalize, and transform data without writing code
<b>AWS Lake Formation</b>	A service that makes it easy to set up, secure, and manage a data lake
<b>Amazon QuickSight</b>	A business intelligence (BI) service that enables users to create interactive dashboards and reports for data visualization and analysis

<b>Amazon Elasticsearch Service</b>	A managed service for Elasticsearch, commonly used for full-text search, log analytics, and real-time application monitoring
<b>AWS Data Pipeline</b>	A web service for orchestrating and automating the movement and transformation of data between different AWS services and on-premises data sources

These tools and services cater to various aspects of big data processing, storage, analysis, and visualization

#### 24. What is AI?

AI, or Artificial Intelligence, refers to the simulation of human intelligence in machines that are programmed to think and learn like humans. The goal of AI is to develop systems that can perform tasks that typically require human intelligence. These tasks include problem-solving, understanding natural language, recognizing patterns, learning from experience, and adapting to changing circumstances.

AI can be categorized into two main types:

<b>Narrow or Weak AI</b>	Weak AI is designed to perform a specific task or a narrow set of tasks. It operates within a well-defined context and is not capable of generalizing its knowledge to other domains. Examples of narrow AI include virtual personal assistants (like Siri or Alexa), image and speech recognition systems, and recommendation algorithms
<b>General or Strong AI</b>	Strong AI refers to a system with the ability to understand, learn, and apply knowledge across a wide range of tasks at a level comparable to human intelligence. Achieving strong AI remains an aspirational goal and is not yet realized in practice.

Key components and techniques within the field of AI include:

<b>Machine Learning (ML)</b>	A subset of AI that focuses on developing algorithms and models that enable computers to learn from data. Machine learning can be categorized into supervised learning, unsupervised learning, and reinforcement learning
------------------------------	---

<b>Deep Learning</b>	A subset of machine learning that uses neural networks with multiple layers (deep neural networks) to learn and make decisions. Deep learning has shown significant success in tasks such as image and speech recognition
<b>Natural Language Processing (NLP)</b>	The ability of machines to understand, interpret, and generate human-like language. NLP is often used in applications such as language translation, chatbots, and sentiment analysis.
<b>Computer Vision</b>	The field of AI that focuses on enabling machines to interpret and understand visual information from the world, similar to how humans perceive and interpret visual data
<b>Robotics</b>	The integration of AI into robotic systems to enable them to perform tasks autonomously. This includes tasks such as navigation, object manipulation, and decision-making.

AI technologies are applied in various industries, including healthcare, finance, education, manufacturing, and more. They are used to automate repetitive tasks, improve decision-making, enhance user experiences, and solve complex problems. As AI continues to advance, ethical considerations, transparency, and responsible development are becoming increasingly important aspects of its deployment and use

## 25. How do you implement an AI application on AWS?

Implementing an AI application on AWS involves several steps, from data preparation and model development to deployment and integration. Here is a high-level overview of the process:

<b>Define Objectives and Requirements</b>	Clearly define the objectives of your AI application. Understand the problem you are trying to solve, the data you have or need, and the outcomes you want to achieve.
<b>Data Collection and Preparation</b>	Gather and prepare the data required for training and testing your AI model. Ensure that the data is



	representative, clean, and well-annotated. AWS services like Amazon S3 can be used for data storage
<b>Select AI/ML Services</b>	Choose the appropriate AWS AI and machine learning services based on your application requirements. Common services include Amazon SageMaker for end-to-end machine learning workflows, Amazon Comprehend for natural language processing, Amazon Rekognition for image and video analysis, etc
<b>Model Development and Training</b>	Use Amazon SageMaker or other frameworks like TensorFlow or PyTorch to develop and train your machine learning model. SageMaker provides a managed environment for building, training, and deploying models
<b>Evaluation and Tuning</b>	Evaluate your model's performance using validation datasets. Fine-tune hyperparameters and adjust your model architecture based on evaluation results
<b>Deployment</b>	Deploy your trained model on AWS using Amazon SageMaker. This can involve creating an endpoint for real-time inference or creating batch transform jobs for large-scale batch processing. Ensure that your deployment is scalable and meets your application's latency requirements
<b>Integration</b>	Integrate your AI application with other AWS services or third-party tools. For example, you might integrate with AWS Lambda for serverless execution, Amazon S3 for data storage, or Amazon API Gateway for building RESTful APIs.
<b>Monitoring and Logging</b>	Implement monitoring and logging to track the performance of your AI application. AWS CloudWatch can be used for monitoring, and CloudWatch Logs can capture logs for debugging and analysis.
<b>Security</b>	Implement security best practices, including using AWS Identity and Access Management (IAM) for access control, encrypting sensitive data, and securing communication between services
<b>Scalability</b>	Design your AI application to be scalable to handle varying workloads. Utilize AWS Auto Scaling and other scaling strategies to ensure that your application can meet demand.
<b>Cost Optimization</b>	Optimize costs by choosing the right instance types, utilizing spot instances for cost savings (where applicable), and implementing strategies to scale down resources during periods of lower demand

<b>Testing</b>	Conduct thorough testing of your AI application, including unit testing, integration testing, and performance testing. Ensure that the application meets the expected accuracy and reliability standards
<b>Documentation</b>	Document your AI application, including the architecture, deployment procedures, and any configuration settings. This documentation will be valuable for future maintenance and updates.
<b>Continuous Improvement</b>	Implement a process for continuous improvement. Monitor model performance over time, retrain models with new data, and iterate on your application based on user feedback and changing requirements

By following these steps and leveraging AWS services, you can build and deploy AI applications that are scalable, reliable, and integrate seamlessly with other cloud services

## 26. What are AI and ML tools provided by AWS?

Amazon Web Services (AWS) offers a variety of AI (Artificial Intelligence) and ML (Machine Learning) tools and services.

<b>Amazon SageMaker</b>	A fully managed service that enables developers to build, train, and deploy machine learning models quickly. It provides an integrated environment for all stages of the machine learning lifecycle.
<b>Amazon Polly</b>	A text-to-speech service that uses advanced deep learning technologies to synthesize human-like speech
<b>Amazon Rekognition</b>	A service for image and video analysis that can identify objects, people, text, scenes, and activities.
<b>Amazon Comprehend</b>	A natural language processing (NLP) service that extracts insights and relationships from unstructured text
<b>Amazon Translate</b>	A neural machine translation service that provides fast and high-quality language translation

<b>Amazon Transcribe</b>	An automatic speech recognition (ASR) service that converts speech to text.
<b>Amazon Lex</b>	A service for building conversational interfaces using voice and text.
<b>AWS DeepLens</b>	A deep learning-enabled video camera that allows developers to experiment with and deploy deep learning models locally on the device
<b>AWS DeepComposer</b>	A service that uses generative AI models to create original music compositions
<b>AWS DeepRacer</b>	An autonomous 1/18th scale race car designed to help developers learn and experiment with reinforcement learning
<b>AWS Deep Learning AMIs</b>	Amazon Machine Images pre-installed with popular deep learning frameworks like TensorFlow and PyTorch, making it easy to set up deep learning environments.
<b>Amazon Augmented AI (A2I)</b>	A service that makes it easy to build the workflows required for human review of machine learning predictions