AWS Infrastructure

AWS Region

AWS regions are physical locations around the world having a cluster of data centers.

AWS Region	Code
US East (N. Virginia)	us-east-1
US East (Ohio)	us-east-2
US West (N. California)	us-west-1
US West (Oregon)	us-west-2
Africa (Cape Town)	af-south-1
Asia Pacific (Hong Kong)	ap-east-1
Asia Pacific (Mumbai)	ap-south-1
Asia Pacific (Osaka)	ap-northeast-3
Asia Pacific (Seoul)	ap-northeast-2
Asia Pacific (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Asia Pacific (Tokyo)	ap-northeast-1
Canada (Central)	ca-central-1
Europe (Frankfurt)	eu-central-1
Europe (Ireland)	eu-west-1
Europe (London)	eu-west-2
Europe (Milan)	eu-south-1
Europe (Paris)	eu-west-3
Europe (Stockholm)	eu-north-1
Middle East (Bahrain)	me-south-1
South America (São Paulo)	sa-east-1

You need to select the region first for most of the AWS services such as EC2, ELB, S3, Lambda, etc.

You can not select region for Global AWS services such as IAM, AWS Organizations, Route 53, CloudFront, WAF, etc.
Each AWS Region consists of multiple, isolated, and physically separate **AZs (Availability Zones)** within a geographic area.

AZ (Availability zones)

An AZ is one or more discrete data centers with redundant power, networking, and connectivity
All AZs in an AWS Region are interconnected with high-bandwidth, low-latency networking.
Customer deploy applications across multiple AZs in same region for high-availability, scalability, fault-tolerant and low-latency.
AZs in a region are usually 3, min is 2 and max is 6 for e.g. **3 AZs in Ohio are us-east-2a, us-east-2b, and us-east-2c**.
For high availability in us-east-2 region with min 6 instances required **either** place 3 instances in each 3 AZs **or** place 6 instances in each 2 AZs (choose any 2 AZs out of 3) so that it works normal when 1 AZ goes down.

Security, Identity & Compliance

IAM (Identity and Access Management)

IAM is used to manage **access** to users and resources
IAM is a global service (applied to all the regions at the same time). IAM is a free service.
Root account is created by default with full administrator, shouldn't be used
Users mapped to physical user, should login to AWS console with their own account and password
Groups can have one or more users, can not have other groups
Policies are JSON documents that **Allow or Deny** the access on **action** can be performed on AWS **resource** by any user, group and role

- **Version** policy language version. 2012-10-17 is latest version.
- **Statement** container for one or more policy statements
- **Sid** (optional) a way of labeling your policy statement
- **Effect** set whether the policy Allow or Deny
- **Principal** user, group, role, or federated user to which you would like to allow or deny access

Action one or more actions that can be performed on AWS resources
Resource one or more AWS resource to which actions apply
Condition (optional) one or more conditions to satisfy for policy to be applicable, otherwise ignore the policy

```
{ "Version":"2012-10-17", "Statement":[{ "Sid":
"Deny-Barclay-S3-Access", "Effect":"Deny", "Principal": {
"AWS": ["arn:aws:iam:123456789012:barclay"] }, "Action": [
"s3:GetObject", "s3:PutObject", "s3:List*" ], "Resource":
["arn:aws:s3:::mybucket/*"] },{ "Effect": "Allow", "Action":
"iam:CreateServiceLinkedRole", "Resource": "*", "Condition": {
"StringLike": { "iam:AWSServiceName": [ "rds.amazonaws.com",
"rds.application-autoscaling.amazonaws.com" ] } } }] }
```

Roles are associated with trusted entities - AWS services (EC2, Lambda, etc), Another AWS account, Web Identity (Cognito or any OpenID provider), or SAML 2.0 federation (your corporate directory). You attach policy to the role, these entities assume the role to access the AWS resources.
Least Privilege Principle should be followed in AWS, don't give more permission than a user needs.
Resource Based Policies are supported by S3, SNS, and SQS
IAM Permission Boundaries to set at individual user or role for maximum allowed permissions
IAM Policy Evaluation Logic ➔ Explicit Deny ⇨ Organization SCPs ⇨ Resource-based Policies (optional) ⇨ IAM Permission Boundaries ⇨ Identity-based Policies
If you got SSL/TLS certificates from third-party CA, import the certificate into **AWS Certificate Manager (ACM)** or upload it to the **IAM Certificate Store**

Access AWS programmatically

AWS Management Console - Use password + MFA (multi factor authentication)
AWS CLI or SDK - Use Access Key ID (~username) and Secret Access Key (~password)

```
$ aws --version $ aws configure AWS Access Key ID [None]: AES
Secret Access Key [None]: Default region name [None]: Default
output format [None]: $ aws iam list-users
```

AWS CloudShell - CLI tool from AWS browser console - Require login to AWS

Access AWS for Non-IAM users

Non-IAM user first authenticate from Identity Federation. Then provide a temporary token (IAM Role attached) generated by calling a AssumeRole API of **STS (Security Token Service)**. Non-IAM user access the AWS resource by assuming IAM Role attached with token.
You can authenticate and authorize Non-IAM users using following Identity Federation:-

SAML 2.0 (old) to integrate Active Directory/ADFS, use AssumeRoleWithSAML STS API
Custom Identity Broker used when identity provider is not compatible to SAML 2.0, use AssumeRole or GetFederationToken STS API
Web Identity Federation is used to sign in using well-known external identity provider (IdP), such as login with Amazon, Facebook, Google, or any OpenID Connect (OIDC)-compatible IdP. Get the ID token from IdP, use AWS Cognito api to exchange ID token with cognito token, use AssumeRoleWithWebIdentity STS API to get temp security credential to access AWS resources
AWS Cognito is recommended identity provider by Amazon
Amazon Single Sign On gives single sign-on token to access AWS, no need to call STS API

You can use **AWS Directory Service** to manage Active Directory (AD) in AWS for e.g.

AWS Managed Microsoft AD is managed Microsoft Windows Server AD with trust connection to on-premise Microsoft AD. Best choice when you need all AD features to support AWS applications or Windows workloads. can be used for single sign-on for windows workloads.
AD Connector is proxy service to redirect requests to on-premise Microsoft AD. Best choice to use existing on-premise AD with compatible AWS services.
Simple AD is standalone AWS managed compatible AD powered by Samba 4 with basic directory features. You cannot connect it to on-premise AD. Best choice for basic directory features.
Amazon Cognito is a user directory for sign-up and sign-in to mobile and web application using Cognito User Pools. Nothing to do with Microsoft AD.

Amazon Cognito

Cognito User Pools (CUP)

User Pools is a **user directory** for sign-up and sign-in to mobile and web applications.

User pool is mainly used for **authentication** to access AWS services

Use to **authenticate** mobile app users through **user pool directory**, or federated through **third-party identity provider (IdP)**. The user pool manages the overhead of handling the tokens that are returned from **social sign-in** through Facebook, Google, Amazon, and Apple, and from **OpenID Connect (OIDC)** and **SAML IdPs**.

After successful authentication, your web or mobile app will receive user pool **JWT tokens** from Amazon Cognito. JWT token can be used in two ways:-

- You use JWT tokens to **retrieve temporary AWS credentials** that allow your app to access other AWS services.
- You create group in user pool with IAM role to access API Gateway, then you can use JWT token (for that group) to **access Amazon API Gateway**.

Cognito Identity Pools (Federated Identity)

Identity pool is mainly used for authorization to access AWS services

You first authenticate user using **User Pools** and then exchange token with **Identity Pools** which further use **AWS STS** to generate **temporary AWS credentials** to access AWS Resources.

You can provide temporary access to write to S3 bucket using facebook/google login to your mobile app users.

Supports **guest users**

AWS Key Management Service (KMS)

AWS managed **centralized key management service** to create, manage and rotate **customer master keys (CMKs)** for encryption at rest.

You can create customer-managed **Symmetric** (single key for both encrypt and decrypt operations) or **Asymmetric** (public/private key pair for encrypt/decrypt or sign/verify operations) master keys
You can enable automatic master key rotation once **per year**. Service keeps the older version of master key to decrypt old encrypted data.

AWS CloudHSM

AWS managed **dedicated hardware** security model (HSM) in AWS Cloud
Enables you to securely generate, store, and manage **your own cryptographic keys**
Integrate with your application using industry-standard APIs, such as PKCS#11, Java Cryptography Extensions (JCE), and Microsoft CryptoNG (CNG) libraries.
Use case: Use **KMS** to create a CMKs in a **custom key store** and store non-extractable key material in AWS CloudHSM to get a **full control on encryption keys**

AWS Systems Manager

Parameter Store is centralized secrets and configuration data management e.g. passwords, database details, and license code
Parameter value can be type **String** (plain text), **StringList** (comma separated) or **SecureString** (KMS encrypted data)
Use case: Centralized configuration for dev/uat/prod environment to be used by CLI, SDK, and Lambda function
Run Command allows you to automate common **administrative tasks** and perform one-time configuration changes on EC2 instances **at scale**
Session Manager replaces the need for Bastions to access instances in private subnet

AWS Secrets Manager

Secret Manager is mainly used to store, manage, and rotate secrets (passwords) such as **database credentials, API keys, and OAuth tokens**.
Secret Manager has **native support to rotate database credentials of RDS databases** - MySQL, PostgreSQL and Amazon Aurora

For other secrets such as API keys or tokens, you need to use the **lambda for customized rotation function**

AWS Shield

AWS managed **Distributed Denial of Service (DDoS) protection** service
Protect against **Layer 3 and 4** (Network and Transport) attacks
AWS Shield Standard is automatic and free DDoS protection service for all AWS customers for CloudFront and Route 53 resources
AWS Shield Advanced is paid service for enhanced DDoS protection for EC2, ELB, CloudFront, and Route 53 resources

AWS WAF

Web Application Firewall protects web applications against common web exploits
Protect against **Layer 7** (HTTP) attack and block common attack patterns, such as **SQL injection** or **Cross-site scripting (XSS)**
You can deploy WAF on CloudFront, Application Load Balancer, API Gateway and AWS AppSync

AWS Firewall Manager

Use **AWS Firewall Manager** to centrally configure and manage AWS WAF rules, AWS Shield Advanced, Network Firewall rules, and Route 53 DNS Firewall Rules across accounts and resources in **AWS Organization**
Use case: Meet Gov regulations to deploy AWS WAF rule to block traffic from embargoed countries across accounts and resources

AWS GuardDuty

Read VPC Flow Logs, DNS Logs, and CloudTrail events. Apply machine learning algorithms and anomaly detections to discover **threats**
Can protect against **CryptoCurrency** attacks

Amazon Inspector

Automated Security Assessment service for **EC2 instances** by installing an **agent in the OS** of EC2 instance.
Inspector comes with **pre-defined rules packages**:-

Network Reachability rules package checks for unintended network accessibility of EC2 instances
Host Assessment rules package checks for vulnerabilities and insecure configurations on EC2 instance. Includes Common Vulnerabilities and Exposures (CVE), Center for Internet Security (CIS) Operating System configuration benchmarks, and security best practices.

Amazon Macie

Managed service to discover and protect your **sensitive data** in AWS
Macie identify and alert for sensitive data, such as **Personally Identifiable Information (PII)** in your selected S3 buckets

AWS Config

Managed service to assess, audit, and evaluate configurations of your AWS resources in multi-region, multi-account
You are notified via SNS for any configuration change
Integrated with CloudTrail, provide resource configuration history
Use case: Customers need to comply with standards like PCI-DSS (Payment Card Industry Data Security Standard) or HIPAA (U.S. Health Insurance Portability and Accountability Act) can use this service to assess compliance of AWS infra configurations

Compute

EC2 (Elastic Compute Cloud)

Infrastructure as a Service (IaaS) - virtual machine on the cloud
You must provision **nitro-based EC2** instance to achieve 64000 EBS IOPS. Max 32000 EBS IOPS with Non-Nitro EC2.
When you restart an EC2 instance, its public IP can change. Use **Elastic IP** to assign a fixed public IPv4 to your EC2 instance. By default, all AWS accounts are limited to five (5) Elastic IP addresses per Region.

Get EC2 instance metadata such as private & public IP from `http://169.254.169.254/latest/meta-data` and user-defined data from `http://169.254.169.254/latest/user-data`
Place all the EC2 instances in same AZ to reduce the data transfer cost
EC2 Hibernate saves the contents of **instance memory (RAM)** to the Amazon EBS root volume. When the instance restarts, the RAM contents are reloaded, brings it to last running state, also known as **pre-warm** the instance. You can hibernate an instance only if it's **enabled for hibernation** and it meets the **hibernation prerequisites**
Use **VM Import/Export** to import virtual machine image and convert to Amazon EC2 AMI to launch EC2 instances

EC2 Instance Types

You can choose **EC2 instance type** based on requirement for e.g. `m5.2xlarge` has Linux OS, 8 vCPU, 32GB RAM, EBS-Only Storage, Up to 10 Gbps Network bandwidth, Up to 4,750 Mbps IO Operations.

Instance Class	Usage Type	Usage Example
T, M	General Purpose	Web Server, Code Repo, Microservice, Small Database, Virtual Desktop, Dev Environment
C	Compute Optimized	High Performance Computing (HPC), Batch Processing, Gaming Server, Scientific Modelling, CPU-based machine learning
R, X, Z	Memory Optimized	In-memory Cache, High Performance Database, Real-time big data analytics
F, G, P	Accelerated Computing	High GPU, Graphics Intensive Applications, Machine Learning, Speech Recognition
D, H, I	Storage Optimized	EC2 Instance Storage, High I/O Performance, HDFS, MapReduce File Systems, Spark, Hadoop, Redshift, Kafka, Elastic Search

EC2 Launch Types

On-Demand - pay as you use, pay per hour, costly
Reserved - up-front payment and reserve for 1 year or 3 year, two classes:-

Standard unused instanced can be sold in AWS reserved instance marketplace

Convertible can be exchanged for another Convertible Reserved Instance with different instance attributes

Scheduled Reserved Instances - reserve capacity that is scheduled to recur daily, weekly, or monthly, with a specified start time and duration, for a one-year term. After you complete your purchase, the instances are available to launch during the time windows that you specified.

Spot Instances - up-to 90% discount, cheapest useful for applications with flexible in timing, can handle interruptions and recover gracefully.

Spot blocks can also be launched with a required duration, which are not interrupted due to changes in the Spot price

Spot Fleet is a collection, or fleet, of Spot Instances, and optionally On-Demand Instances, which attempts to launch the number of Spot and On-Demand Instances to meet the specified target capacity

Dedicated Instance - Your instance runs on dedicated hardware provides physical isolation, single-tenant

Dedicated Hosts - Your instances run on a dedicated physical server. More visibility of how instances are placed on server. Let you use existing server-bound software licenses and address corporate compliance and regulatory requirements.

You have a limit of **20 Reserved instances**, 1152 vCPU On-demand standard instances, and 1440 vCPU spot instances. You can increase the limit by submitting the EC2 limit increase request form.

EC2 Enhanced Networking

Elastic Network Interface (ENI) is a virtual network card, which you attach to EC2 instance in same AZ. ENI has one primary private IPv4, one or more secondary private IPv4, one Elastic IP per private IPv4, one public IPv4, one or more IPv6, one or more security groups, a MAC address and a source/destination check flag

While **primary ENI** cannot be detached from an EC2 instance, A **secondary ENI** with private IPv4 **can be detached and attached** to a standby EC2 instance if primary EC2 becomes unreachable **(failover)**

Elastic Network Adapter (ENA) for C4, D2, and M4 EC2 instances, Upto 100 Gbps network speed.

Elastic Fabric Adapter (EFA) is ENA with additional **OS-bypass** functionality, which enables HPC and Machine Learning applications to bypass the operating system kernel and communicate directly with EFA

device resulting in very high performance and low latency. for M5, C5, R5, I3, G4, metal EC2 instances.
Intel 82599 Virtual Function (VF) Interface for C3, C4, D2, I2, M4, and R3 EC2 instances, Upto 10 Gbps network speed.

EC2 Placement Groups Strategy

Placement groups can span across AZs only, **cannot span across regions**

Cluster - Same AZ, Same Rack, Low latency and High Network, High-Performance Computing (HPC)
Spread - Different AZ, Distinct Rack, High Availability, Critical Applications, Limited to 7 instances per AZ per placement group.
Partition - Same or Different AZ, Different Rack (or Partition), Distributed Applications like Hadoop, Cassandra, Kafka etc, Upto 7 Partition per AZ

AMI (Amazon Machine Image)

Customized image of an EC2 instance, having built-in OS, softwares, configurations, etc.
You can create an AMI from EC2 instance and launch a new EC2 instance from AMI.
AMI are built for a specific region and can be copied across regions

ELB (Elastic Load Balancing)

AWS load balancer provides a static DNS name provided for e.g. `http://myalb-123456789.us-east-1.elb.amazonaws.com`
AWS load balancer routes the request to **Target Groups**. Target group can have one or more EC2 instances, IP Addresses or lambda functions.
Three types of ELB - Classic Load Balancer, Application Load Balancer, and Network Load Balancer

Application Load Balancer (ALB):

Routing based on hostname, request path, params, headers, source IP etc.
Support **Request tracing**, add `X-Amzn-Trace-Id` header before sending the request to target
Client IP and port can be found in `X-Forwarded-For` and `X-Forwarded-Porto` header
integrate with WAF with rate-limiting (throttle) rules to prevent from DDoS attacks

Network Load Balancer (NLB):

Handle **volatile workloads** and **extreme low-latency**

Provide static IP/Elastic IP for the load balancer per AZ
allows registering targets by IP address
Use NLB with Elastic IP in front of ALBs when there is a requirement of whitelisting ALB

Stickiness: works in CLB and ALB. Stickiness and its duration can be set at Target Group level. Doesn't work with NLB

ELB Types	Supported Protocol
Application Load Balancer	HTTP, HTTPS, WebSocket
Network Load Balancer	TCP, UDP, TLS
Gateway Load Balancer	Thirdparty appliances
Classic Load Balancer (old)	HTTP, HTTPS, TCP

ASG (Auto Scaling Group)

Scale-out (add) or scale-in (remove) EC2 instances based on scaling policy - CPU, Network, Custom metric or Scheduled.
You configure the size of your Auto Scaling group by setting the minimum, maximum, and desired capacity. ASG runs EC2 instances at desired capacity if no policy specified. Minimum and maximum capacity are boundaries within ASG scale-in or scale-out. `min <= desired <= max`
Instances are created in ASG using **Launch Configuration** (legacy) or **Launch Template** (newer)
You **cannot change the launch configuration** for an ASG, you must create a new launch configuration and update your ASG with it.
You can create ASG that launches both Spot and On-Demand Instances or multiple instance types using **launch template**, not possible with launch configuration.

Dynamic Scaling Policy

Target Tracking Scaling - can have more than one policy for e.g. add or remove capacity to keep the average aggregate CPU utilization of your Auto Scaling group at 40% and request count per target of your ALB target group at 1000 for your ASG. If both policies occurs at same time, use largest capacity for both scale-out and scale-in.
Simple Scaling - e.g. CloudWatch alarm CPUUtilization (>80%) - add 2 instances

Step Scaling - e.g. CloudWatch alarm CPUUtilization (60%-80%)- add 1, (>80%) - add 3 more, (30%-40%) - remove 1, (<30%) - remove 2 more

Scheduled Action - e.g. Increase min capacity to 10 at 5pm on Fridays

Default Termination Policy - Find AZ with most number of instances, and delete the one with **oldest launch configuration**, in case of tie, the one closest to **next billing hour**

Cooldown period is the amount of time to wait for previous scaling activity to take effect. Any scaling activity during cooldown period is ignored.

Health check grace period is the amount of wait time to check the health status of EC2 instance, which has just came into service to give enough time to warmup.

You can add **lifecycle-hooks** to ASG to perform custom action during:-

scale-out to run script, install softwares and send `complete-lifecycle-action` command to continue

scale-in e.g. download logs, take snapshot before termination

Lambda

FaaS (Function as a Service), Serverless

Lambda function **supports many languages** such as Node.js, Python, Java, C#, Golang, Ruby, etc.

Lambda **limitations**:-

execution time can't exceed 900 seconds or 15 min

min required memory is 128MB and can go till 10GB with 1-MB increment

`/temp` directory size to download file can't exceed 512 MB

max environment variables size can be 4KB

compressed *.zip* and uncompressed code can't exceed 50MB and 250MB respectively

Lambda function can be **triggered** on DynamoDB database trigger, S3 object events, event scheduled from EventBridge (CloudWatch Events), message received from SNS or SQS, etc.

Assign IAM Role to lambda function to give access to AWS resource for e.g. create snapshot of EC2, process image and store in S3, etc.

Lambda can **auto scale in seconds** to handle sudden burst of traffic. EC2 require minutes to auto scale.

You are charged based on number of requests, execution time and resource (memory) usage. Cheaper than EC2.

You can use **Lambda@Edge** to run code at CloutFront Edge globally
You can optionally setup a **dead-letter queue (DLQ)** with SQS or SNS to forward **unprocessed** or failed requests payload
You can enable and watch the **lambda execution logs** in CloudWatch

Application Integration

SQS (Amazon Simple Queue Service)

Fully managed service with following specifications for Standard SQS:-
- can have unlimited number of messages waiting in queue
- default retention period is 4 days and max 14 days
- can send message upto 256KB in size
- unlimited throughput and low latency (<10ms on publish and receive)
- can have duplicate messages (At least once delivery)
- can have out-of-order messages (best-effort ordering)

Consumer (can be EC2 instance or lambda function) **poll** the messages **in batches** (upto 10 messages) and **delete** them from queue after processing. If don't delete, they stay in Queue and may process multiple times.
You should allow Producer and Consumer to send and receive messages from **SQS Queue Access Policy**
Message Visibility Timeout when a message is polled by a consumer, it becomes **invisible** to other consumers for timeout period.
You can setup a **Dead-letter queue (DLQ)** which is another SQS to keep the messages which are failed to process by consumers multiple times and exceed the **Maximum receives** threshold in SQS.
You use **SQS Temporary Queue Client** to implement SQS Request-Response System.
You can delay message (consumers don't see them immediately) up to 15 minutes (default 0 seconds). You can do it using **Delivery Delay** configuration at queue level or **DelaySeconds** parameter at message level.
Long polling is when the `ReceiveMessageWaitTimeSeconds` property of a queue is set to a value greater than zero. Long polling reduces the number of empty responses by allowing Amazon SQS to wait until a message is available before sending a response to a ReceiveMessage request, helps to reduce the cost.
You can create SQS of type **FIFO** which **guarantee ordering and exactly once processing** with limited throughput upto 300 msg/s without and 3000

msg/s with batching. FIFO queue name must end with suffix ***.fifo***. You can not convert Standard SQS to FIFO SQS.
Use case: Cloudwatch has custom metric on =(SQS queue length/Number of EC2 instances), which alarm ASG to auto scale EC2 instances (SQS consumer) based on number of messages in queue.

SNS (Amazon Simple Notification Service)

PubSub model, where publisher sends the messages on SNS topic and all topic subscribers receive those messages.
Upto 100,000 topics and Upto 12,500,000 subscription per topic
Subscribers can be: Kinesis Data Firehose, SQS, HTTP, HTTPS, Lambda, Email, Email-JSON, SMS Messages, Mobile Notifications.
You can setup a **Subscription Filter Policy** which is JSON policy to send the filtered messages to specific subscribers.
Fan out pattern: SNS topic has multiple SQS subscribers e.g. send all order messages to SNS topic and then send filtered messages based on order status to 3 different application services using SQS.

Amazon MQ

Amazon managed **Apache ActiveMQ**
Migrate an existing message broker using **MQTT** protocol to AWS.

Storage

S3 (Simple Storage Service)

S3 Bucket is an **object-based** storage, used to manage data as objects
S3 Object is having:-

- Value - data bytes of object (photos, videos, documents, etc.)
- Key - full path of the object in bucket e.g. `/movies/comedy/abc.avi`
- Version ID - version object, if versioning is enabled
- Metadata - additional information

S3 Bucket holds objects. S3 console shows virtual folders based on key.
S3 is a universal namespace so bucket names must be globally unique (think like having a domain name)

`https://<bucket-name>.s3.<aws-region>.amazonaws.com or https://s3.<aws-region>.amazonaws.com/<bucket-name>`
Unlimited Storage, Unlimited Objects from **0 Bytes** to **5 Terabytes** in size.
You should use **multi-part upload** for Object size **> 100MB**
All new buckets are **private** when created by default. You should enable **public access** explicitly.
Access control can be configured using **Access Control List (ACL)** (deprecated) and **S3 Bucket Policies** (recommended)
S3 Bucket Policies are **JSON** based policy for complex access rules at user, account, folder, and object level
Enable **S3 Versioning** and **MFA delete** features to protect against accidental delete of S3 Object.
Use **Object Lock** to store object using write-once-read-many (WORM) model to prevent objects from being deleted or overwritten for a fixed amount of time (**Retention period**) or indefinitely (**Legal hold**). Each version of object can have different retention-period.
You can **host static websites** on S3 bucket consisting of HTML, CSS, **client-side JavaScript**, and images. You need to enable Static website hosting and Public access for S3 to avoid 403 forbidden error. Also you need to add **CORS Policy** to allow cross-origin request.
`https://<bucket-name>.s3-website[.-]<aws-region>.amazonaws.com`
Generate a **pre-signed URL** from CLI or SDK (can't from the web) to provide temporary access to an S3 object to either upload or download object data. You specify expiry (say 5 sec) while generating url:-
`aws s3 presign s3://mybucket/myobject --expires-in 300`
S3 Select or **Glacier Select** can be used to query subset of data from S3 Objects using SQL query. S3 Objects can be CSV, JSON, or Apache Parquet. GZIP & BZIP2 compression is supported with CSV or JSON format with server-side encryption.
using `Range` HTTP Header in a GET Request to download the specific range of bytes of S3 object, known as **Byte Range Fetch**
You can create **S3 event notification** to push events e.g. `s3:ObjectCreated:*` to SNS topic, SQS queue or execute a Lambda function. It is possible that you receive single notification for two writes to a non-versioned object at the same time. Enable versioning to ensure you get all notifications.
Enable **S3 Cross-Region Replication** for asynchronous replication of object across buckets in another region. You must have **versioning** enabled on both **source** and **destination** side. Only **new S3 Objects** are replicated after you enable them.

Enable **Server access logging** for logging object-level fields object-size, total time, turn around time, and HTTP referrer. Not available with CloudTrail.
Use **VPC S3 gateway endpoint** to access S3 bucket within AWS VPC to reduce the overall data transfer cost.
Enable **S3 Transfer Acceleration** for faster transfer and high throughput to S3 bucket (mainly uploads), Create **CloudFront** distribution with OAI pointing to S3 for faster-cached content delivery (mainly reads)
Restrict the access of S3 bucket through CloudFront only using **Origin Access Identity (OAI)**. Make sure user can't use a direct URL to the S3 bucket to access the file.

S3 Storage Class Types

Standard: Costly choice for very high availability, high durability and fast retrieval
Intelligent Tiering: Uses ML to analyze your Object's usage and move to the appropriate cost-effective storage class automatically
Standard-IA: Cost-effective for infrequent access files which cannot be recreated
One-Zone IA: Cost-effective for infrequent access files which can be recreated
Glacier: Cheaper choice to Archive Data. You must purchase **Provisioned capacity**, when you require guaranteed **Expedite retrievals**.
Glacier Deep Archive: Cheapest choice for Long-term storage of large amount of data for compliance

S3 Storage Class	Durability	Availability	AZ	Min. Storage	Retrieval Time	Retrieval fee
S3 Standard (General Purpose)	11 9's	99.99%	≥3	N/A	milliseconds	N/A
S3 Intelligent Tiering	11 9's	99.9%	≥3	30 days	milliseconds	N/A

S3 Standard-IA (Infrequent Access)	11 9's	99.9%	≥3	30 days	milliseconds	per GB
S3 One Zone-IA (Infrequent Access)	11 9's	**99.5%**	**1**	30 days	milliseconds	per GB
S3 Glacier	11 9's	99.99%	≥3	90 days	**Expedite (1-5 mins)** Standard (3-5 hrs) Bulk (5-12 hrs)	per GB
S3 Glacier Deep Archive	11 9's	99.99%	≥3	180 days	**Standard (12 hrs)** Bulk (48 hrs)	per GB

You can upload files in the same bucket with different **Storage Classes** like *S3 standard, Standard-IA, One Zone-IA, Glacier* etc.
You can setup **S3 Lifecycle Rules** to transition current (or previous version) objects to cheaper storage classes or delete (expire if versioned) objects after certain days e.g.

transition from S3 Standard to S3 Standard-IA or One Zone-IA can only be done after 30 days.
transition from S3 Standard to S3 Intelligent Tiering, Glacier, or Glacier Deep Archive can be done immediately.

You can also setup lifecycle rule to **abort multipart upload**, if it doesn't complete within certain days, which auto delete the parts from S3 buckets associated with multipart upload.

Encryption

Encryption is transit between client and S3 is achieved via SSL/TLS
You can add default encryption at bucket level and also override encryption at file level.
Encryption at rest - Server Side Encryption (SSE)

SSE-S3 AWS S3 managed keys, use AES-256 algorithm. Must set header: `"x-amz-server-side-encryption":"AES-256"`

SSE-KMS Envelope Encryption using AWS KMS managed keys. Must set header: `"x-amz-server-side-encryption":"aws:kms"`

SSE-C Customer provides and manage keys. HTTPS is mandatory.

Encryption at rest - Client Side Encryption client encrypts and decrypts the data before sending and after receiving data from S3.

To meet **PCI-DSS or HIPAA** compliance, encrypt S3 using SSE-C and Client Side Encryption

Data Consistency

S3 provides **strong read-after-write consistency for PUTs and DELETEs** of objects. PUTs applies to both writes to new objects as well as overwrite existing objects.

Updates to a single key are atomic. For example, if you PUT to an existing key from one thread and perform a GET on the same key from a second thread concurrently, you will get either the old data or the new data, but never partial or corrupt data.

AWS Athena

You can use **AWS Athena** (Serverless Query Engine) to perform analytics directly against S3 objects using SQL query and save the analysis report in another S3 bucket.

Use Case: one-time SQL query on S3 objects, S3 access log analysis, serverless queries on S3, IoT data analytics in S3, etc.

Instance Store

Instance Store is temporary **block-based** storage physically attached to an EC2 instance

Can be attached to an EC2 instance only when the instance is launched and cannot be dynamically resized

Also known as **Ephemeral Storage**

Deliver very low-latency and high random I/O performance

Data persists on instance reboot, data **doesn't persist on stop or termination**

EBS (Elastic Block Store)

EBS is **block-based** storage, referred as EBS Volume

EBS Volume think like a USB stick

- Can be attached to only one EC2 instance at a time. Can be detached & attached to another EC2 instance in that same AZ only
- Can attach multiple EBS volumes to single EC2 instance. Data persist after detaching from EC2

EBS Snapshot is a backup of EBS Volume at a point in time. You can not copy EBS volume across AZ but you can create EBS Volume from Snapshot across AZ. EBS Snapshot can copy across AWS Regions.

Facts about EBS Volume **encryption**:-

- All data at rest inside the volume is encrypted
- All data in flight between the volume and EC2 instance is encrypted
- All snapshots of encrypted volumes are automatically encrypted
- All volumes created from encrypted snapshots are automatically encrypted
- Volumes created from unencrypted snapshots can be encrypted at the time of creation

EBS supports **dynamic changes in live production** volume e.g. volume type, volume size, and IOPS capacity without service interruption

There are two types of EBS volumes:-

- **SSD** for small/random IO operations, High IOPS means number of read and write operations per second, Only SSD EBS Volumes can be used as **boot volumes** for EC2
- **HDD** for large/sequential IO operations, High Throughput means number of bytes read and write per second

EBS Volumes with two types of RAID configuration:-

- **RAID 0** (increase performance) two 500GB EBS Volumes with 4000 IOPS - creates 1000GB RAID0 Array with 8000 IOPS and 1000Mbps throughput
- **RAID 1** (increase fault tolerance) two 500GB EBS Volumes with 4000 IOPS - creates 500GB RAID1 Array with 4000 IOPS and 500Mbps throughput

EBS Volume Types	Description	Usage
General Purpose SSD (gp2/gp3)	Max 16000 IOPS	boot volumes, dev environment, virtual desktop
Provisioned IOPS SSD (io1/io2)	16000 - 64000 IOPS, EBS Multi-Attach	critical business application, large SQL and NoSQL database workloads
Throughput Optimized HDD (st1)	Low-cost, frequently accessed, throughput intensive	Big Data, Data warehouses, log processing
Cold HDD (sc1)	Lowest-cost, infrequently accessed	Large data with lowest cost

EFS (Elastic File System)

EFS is a **POSIX-compliant file-based** storage
EFS supports **file systems semantics** - strong read-after-write consistency and file locking
highly scalable - can automatically scale from gigabytes to petabytes of data without needing to provision storage. With **burst mode**, the throughput increase, as file system grows in size.
highly available - stores data redundantly across multiple Availability Zones
Network File System (NFS) that can be mounted on and **accessed concurrently by thousands of EC2** in multiple AZs without sacrificing performance.
EFS file systems can be accessed by Amazon EC2 Linux instances, Amazon ECS, Amazon EKS, AWS Fargate, and AWS Lambda functions via a file system interface such as NFS protocol.
Performance Mode:

- **General Purpose** for most file system for low-latency file operations, good for content-management, web-serving etc.
- **Max I/O** is optimized to use with 10s, 100s, and 1000s of EC2 instances with high aggregated throughput and IOPS, **slightly higher latency** for file operations, good for big data analytics, media processing workflow

Use case: Share files, images, software updates, or computing across all EC2 instances in ECS, EKS cluster

FSx for Windows

Windows-based file system supports **SMB** protocol & Windows **NTFS** supports **Microsoft Active Directory (AD) integration**, ACLs, user quotas

FSx for Lustre

Lustre = Linux + Cluster is a **POSIX-compliant parallel linux file system**, which stores data across multiple network file servers
High-performance file system for **fast processing of workload** with consistent **sub-millisecond latencies**, up to hundreds of gigabytes per second of throughput, and up to **millions of IOPS**.
Use it for Machine learning, High-performance computing (HPC), video processing, financial modeling, genome sequencing, and electronic design automation (EDA).
You can use **FSx for Lustre as hot storage** for your highly accessed files, and **Amazon S3 as cold storage** for rarely accessed files.
Seamless integration with Amazon S3 - connect your S3 data sets to your FSx for Lustre file system, run your analyses, write results back to S3, and delete your file system
FSx for Lustre provides two deployment options:-

- **Scratch file systems** - for temporary storage and short-term processing
- **Persistent file systems** - for high available & persist storage and long-term processing

Database

RDS (Relational Database Service)

AWS Managed Service to create PostgreSQL, MySQL, MariaDB, Oracle, Microsoft SQL Server, and Amazon Aurora in the cloud
Scalability: Upto 5 Read replicas, replication is asynchronous so reads are eventually consistent.
Availability use Multi-AZ Deployment, synchronous replication
You can create a **read replica** in a **different region** of your running RDS instance. You pay for replication cross Region, but not for cross AZ.

Automatic **failover by switching the CNAME** from primary to standby database
Enable **Password and IAM Database Authentication** to authenticate using database password and user credentials through IAM users and roles, works with MySQL and PostgreSQL
Enable **Enhanced Monitoring** to see percentage of CPU bandwidth and total memory consumed by each database process (OS process thread) in DB instance
Enable **Automated Backup** for **daily storage volume snapshot** of your DB instance with retention-period from 1 day (default from CLI, SDK) to 7 days (default from console) to 35 days (max). Use **AWS Backup** service for retention-period of 90 days.
To encrypt an unencrypted RDS DB instance, take a snapshot, copy snapshot and encrypt new snapshot with AWS KMS. Restore the DB instance with the new encrypted snapshot.

Amazon Aurora

Amazon fully managed relational database compatible with MySQL and PostgreSQL
Provide 5x throughput of MySQL and 3x throughput of PostgreSQL
Aurora Global Database is single database span across multiple AWS regions, enable low-latency global reads and disaster recovery from region-wide outage. Use global database for disaster recovery having RPO of 1 second and RTO of 1 minute.
Aurora Serverless capacity type is used for on-demand auto-scaling for intermittent, unpredictable, and sporadic workloads.
Typically operates as a DB cluster consisting of one or more DB instances and a cluster volume that manages cluster data with each AZ having a copy of volume.

- **Primary DB instance** - Only one primary instance, supports both read and write operation
- **Aurora Replica** - Upto 15 replicas spread across different AZ, supports only read operation, automatic failover if primary DB instance fails, high availability

Connections Endpoints

- **Cluster endpoint** - only one cluster endpoint, connects to primary DB instance, only this endpoint can perform write (DDL, DML) operations

Reader endpoint - one reader endpoint, provides load-balancing for all read-only connections to read from Aurora replicas
Custom endpoint - Upto 5 custom endpoint, read or write from a specified group of DB instances from Cluster, used for specialized workloads to route traffic to high-capacity or low-capacity instances
Instance endpoint - connects to specified DB instance directly, generally used to improve connection speed after failover

DynamoDB

AWS proprietary, Serverless, managed **NoSQL database**
Use to store **JSON documents**, or session data
Use as distributed serverless cache with **single-digit millisecond** performance
Planned Capacity provision WCU & RCU, can enable auto-scaling, good for predictable workloads
On-demand Capacity unlimited WCU & RCU, more expensive, good for unpredictable workloads where read & write are less (low throughput)
Add **DAX (DynamoDB Accelerator) cluster** in front of DynamoDB to cache frequently read values and offload the heavy read on hot keys of DynamoDB, prevent `ProvisionedThroughputExceededException`
Enable **DynamoDB Streams** to trigger events on database and integrate with lambda function for e.g. send welcome email to user added into the table.
Use **DynamoDB Global Table** to serve the data globally. You must enable *DynamoDB Streams* first to create global table.
You can use **Amazon DMS** (Data Migration Service) to migrate from Mongo, Oracle, MySQL, S3, etc. to DynamoDB

ElastiCache

AWS Managed Service for **Redis** or **Memcached**
Use as distributed cache with **sub-millisecond** performance
Elasticache for Redis
Offers Multi-AZ with Auto-failover, Cluster mode
Use password/token to access data using **Redis Auth**

HIPAA Compliant

Elasticache for Memcached

Intended for use in speeding up dynamic web applications
Not HIPAA Compliant

Redshift

Columnar Database, OLAP (online analytical processing)
supports **Massive Parallel Query Execution (MPP)**
Use for Data Analytics and Data warehousing
Integrate with **Business Intelligence (BI) tools** like AWS Quicksight or Tableau for analytics
Use **Redshift Spectrum** to query S3 bucket directly without loading data in Redshift

Amazon Kinesis

Amazon Kinesis is a fully managed service for collecting, processing and analyzing **streaming real-time data** in the cloud. Real-time data generally comes from IoT devices, gaming applications, vehicle tracking, clickstream, etc.

Kinesis Data Streams capture, process and store data streams.

Producer can be Amazon Kinesis Agent, SDK, or Kinesis Producer Library (KPL)
Consumer can be Kinesis Data Analytics, Kinesis Data Firehose, or Kinesis Consumer Library (KCL)
Data Retention period from **24 hours (default)** to 365 days (max).
Order is maintained at Shard (partition) level.

Kinesis Data Firehose loads data streams into AWS data stores such as S3, Amazon Redshift and ElastiSearch. Transform data using lambda functions and store failed data in another S3 bucket.
Kinesis Data Analytics analyzes data streams with SQL or Apache Flink
Kinesis Video Streams capture, process and store video streams

Amazon EMR

EMR = Elastic MapReduce

Big data cloud platform for processing vast data using open source tools such as **Hadoop**, Apache Spark, Apache Hive, Apache HBase, Apache Flink, Apache Hudi, and Presto.
EMR can be used to perform data transformation workloads - Extract, transform, load (ETL)
Use case: Analyze Clickstream data from S3 using Apache Spark and Hive to deliver more effective ads

Neptune

Graph Database
Use case: high relationship data, social networking data, knowledge graphs (Wikipedia)

ElasticSearch

Amazon-managed **Elastic Search** service
Integration with Kinesis Data Firehose, AWS IoT, and CloudWatch logs
Use case: Search, indexing, partial or fuzzy search

Migration

AWS Snow Family

AWS snow family is used for **on-premises large-scale data migration** to S3 buckets and processing data at low network locations.
You need to install **AWS OpsHub** software to transfer files from your on-premises machine to snow device.
You can not migrate directly to Glacier, you should create S3 first with a lifecycle policy to move files to Glacier. You can transfer to Glacier directly using DataSync.

Family Member	Storage	RAM	Migration Type	DataSync	Migration Size
Snowcone	8TB	4GB	online & offline	yes	GBs and TBs

Snowball Edge Storage Optimized	80TB	80GB	offline	no	petabyte scale
Snowball Edge Compute Optimized	42TB	208GB	offline	no	petabyte scale
Snowmobile	100PB	N/A	offline	no	exabyte scale

AWS Storage Gateway

Store gateway is a **hybrid cloud service** to move on-premises data to the cloud and connect on-premises applications with cloud storage.

Storage Gateway	Protocol	Backed by	Use Case
File Gateway	NFS & SMB	S3 -> S3-IA, S3 One Zone-IA	Store files as object in S3, with a local cache for low-latency access, with user auth using Active Directory
FSx File Gateway	SMB & NTFS	FSx -> S3	Windows or Lustre File Server, integration with Microsoft AD
Volume Gateway	iSCSI	S3 -> EBS	Block storage in S3 with backups as EBS snapshots. Use **Cached Volume** for low-latency and **Stored Volume** for scheduled backups
Tape Gateway	iSCSI VTL	S3 -> S3 Glacier & Glacier Deep Archive	Backup data in S3 and archive in Glacier using tape-based process

AWS DataSync

AWS DataSync is used for **Data Migration** at a large scale from **On-premises storage** systems (using NFS and SMB storage protocol) to

AWS storage (like S3, EFS, or FSx for Windows, AWS Snowcone) over the **internet**
AWS DataSync is used to archive on-premises **cold data** directly to **S3 Glacier or S3 Glacier Deep Archive**
AWS DataSync can migrate data directly to **any S3 storage class**
Use DataSync with **Direct Connect** to migrate data over **secure private network** to AWS service associated with **VPC endpoint**.

AWS Backup

AWS Backup to centrally manage and automate the backup process for EC2 instances, EBS Volumes, EFS, RDS databases, DynamoDB tables, FSx for Lustre, FSx for Window server, and Storage Gateway volumes
Use case: Automate backup of RDS with 90 days retention policy. (Automate backup using RDS directly has max 35 days retention period)

Database Migration Service (DMS)

DMS helps you to migrate database to AWS with source remaining fully operational during the migration, minimizing the downtime
You need to select EC2 instance to run **DMS** in order to migrate (and replicate) database from source => target e.g. On-premise => AWS, AWS => AWS, or AWS => On-premise
DMS supports both **homogenous** migrations such as On-premise PostgreSQL => AWS RDS PostgreSQL and **heterogenous** migrations such as SQL Server or Oracle => MySQL, PostgreSQL, Aurora, or Teradata or Oracle => Amazon Redshift
You need to run **AWS SCT** (Schema Conversion Tool) at source for **heterogenous** migrations

AWS Application Migration Service (MGN)

Migrate virtual machines from VMware vSphere, Microsoft Hyper-V or Microsoft Azure to AWS
AWS Application Migration Service (new) utilizes continuous, block-level replication and enables cutover windows measured in minutes

AWS Server Migration Service (legacy) utilizes incremental, snapshot-based replication and enables cutover windows measured in hours.

Networking & Content Delivery

Amazon VPC

CIDR block — Classless Inter-Domain Routing. An internet protocol address allocation and route aggregation methodology. CIDR block has two components - Base IP (WW.XX.YY.ZZ) and Subnet Mask (/0 to /32) for e.g.

- **192.168.0.0/32** means 2^{32-32}= 1 single IP
- **192.168.0.0/24** means 2^{32-24}= 256 IPs ranging from *192.168.0.0 to 192.168.0.255* (last number can change)
- **192.168.0.0/16** means 2^{32-16}= 65,536 IPs ranging from *192.168.0.0 to 192.168.255.255* (last 2 numbers can change)
- **192.168.0.0/8** means 2^{32-8}= 16,777,216 IPs ranging from *192.0.0.0 to 192.255.255.255* (last 3 numbers can change)
- **0.0 0.0.0.0/0** means 2^{32-0}= All IPs ranging from *0.0.0.0 to 255.255.255.255* (all 4 numbers can change)

VPC (Virtual Private Cloud)

A virtual network dedicated to your AWS account.
VPCs are **region specific** they do not span across regions
Every region comes with default VPC. You can create **upto 5 VPC** per region.
You can assign **Max 5 IPv4 CIDR blocks** per VPC with **min** block size **/28 =** 16 IPs and **max** size **/16 =** 65,536 IPs. You can assign Secondary IP CIDR range later if primary CIDR IPs are exhausted.
Only private IP ranges are allowed in IPv4 CIDR block - 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16.
You VPC CIDR block should not overlap with other VPC networks within your AWS account.
Enable **DNS resolution** and **DNS hostnames** at VPC, EC2 instances created in that VPC will be assigned a domain name address

VPC Peering

VPC peering connect two VPC over a **direct network route** using **private IP addresses**

Instances on peered VPCs **behave** just like they are on the **same network**

Must have **no overlapping CIDR Blocks**

VPC peering connection are **not transitive** i.e. VPC-A peering VPC-B and VPC-B peering to VPC-C doesn't mean VPC-A peering VPC-C

Route tables **must be updated** in both VPC that are peered so that instances can communicate

Can connect one VPC to another in **same** or **different region**.

VPC peering in the different region called **VPC inter-region peering**

Can connect one VPC to another in **same** or **different AWS account**

Subnet

A range of IP addresses in your VPC

Each subnet is tied to one Availability Zone, one Route Table, and one Network ACL

You assign one CIDR block per Subnet within CIDR range of your VPC. Should not overlap with other Subnet's CIDR in your VPC.

AWS **reserve 5 IP address** (first 4 and last 1) from CIDR block in each Subnet. For e.g. If you need 29 IP addresses to use, your should choose CIDR /26 = 64 IP and not /27 = 32 IP, since 5 IPs are reserved and can not use.

Enable **Auto assign public IPv4** address in public subnets, EC2 instances created in public subnets will be assigned a public IPv4 address

If you have 3 AZ in a region then you create a total of 6 subnets - 3 private subnets (1 in each AZ) and 3 public subnets (1 in each AZ) for multi-tier and highly-available architecture. API gateway and ALB reside in the public subnet, EC2 instances, Lambda, Database resides in private subnet.

Route Table

A set of rules, called routes, are used to determine where **network traffic is directed**.

Each **subnet** in your VPC **must be associated** with a route table.

A subnet can only be associated **with one route table at a time**

You can associate **multiple subnets with the same route table** For e.g. you create 4 subnets in your VPC where 2 subnets associated with one route table with no internet access rules known as **private subnets** and another 2 subnets are associated with another route table with internet access rules known as **public subnets**

Each Route table route has **Destination** like IPs and **Target** like local, IG, NAT, VPC endpoint etc.

public subnet is a subnet that's associated with a route table having rules to connect to internet using Internet Gateway.

private subnet is a subnet that's associated with a route table having no rules to connect to internet using Internet Gateway.

When our Subnets connected to the Private Route Table need access to the internet, we set up a NAT Gateway in the public Subnet. We then add a rule to our Private Route Table saying that all traffic looking to go to the internet should point to the NAT Gateway.

Internet Gateway

Internet Gateway allows AWS instances **public subnet access to the internet and accessible from the internet**

Each Internet Gateway is associated with one VPC only, and each VPC has one Internet Gateway only (one-to-one mapping)

NAT Gateway

NAT Gateway allows AWS instances in **private subnet access to the internet but not accessible from the internet**

NAT Gateway (latest) is a managed service that launches redundant instances within the selected AZ (can survive failure of EC2 instance)

NAT Instances (legacy) are individual EC2 instances. Community AMIs exist to launch NAT Instances. Works same as NAT Gateway.

You can only have 1 NAT Gateway inside 1 AZ (cannot span AZ).

You should create a NAT Gateway in each AZ for **high availability** so that if a NAT Gateway goes down in one AZ, instances in other AZs are still able to access the internet.

NAT Gateway resides in public subnet. You must **allocate Elastic IP** to NAT Gateway. You must add NAT Gateway in

private subnet route table with Destination `0.0.0.0/0` and Target `nat-gateway-id`

NAT Gateways are **automatically assigned a public IP address**

NAT Gateway/Instances **works with IPv4**

NAT Gateway **cannot be shared across VPC**

NAT Gateway **cannot be used as Bastions** whereas Nat Instance can

Bastion Host

Bastian Host is an individual **small EC2 instance in public subnet**. Community AMIs exist to launch Bastion Host.

Bastian Host are used to access AWS instances in **private subnet with private IPv4 address via SSH at port 22**

Egress Only Internet Gateway

Works same as NAT Gateway, but **for IPv6**

Egress Only means - outgoing traffic only

IPv6 are public by default. Egress Only Internet Gateway allows IPv6 instances in private subnet access to the internet but accessible from internet

Network ACL

Network Access Control List is commonly **known as NACL**

Optional layer of security for your VPC that acts as a firewall for controlling traffic in and out of one or more subnets.

VPCs comes with a modifiable **default NACL**. By default, it **allows all inbound and outbound** traffic.

You can create **custom NACL**. By default, each custom network ACL **denies all inbound and outbound** traffic until you add rules.

Each subnet within a VPC must be associated with only 1 NACL

If you don't specify, auto associate with default NACL.

If you associate with a new NACL, auto-remove previous association

Apply to all instances in associated subnet

Support both **Allow** and **Deny** rules

Stateless means explicit rules for inbound and outbound traffic. return traffic must be explicitly allowed by rules

Evaluate rules in **number order**, starting with lowest numbered rule. NACL rules have number(1 to 32766) and higher

precedence to lowest number for e.g. `#100 ALLOW <IP>` and `#200 DENY <IP>` means IP is allowed

Each network ACL also includes a rule with **rule number as asterisk** `*`. If any of the numbered rule doesn't match, it's denies the traffic. You can't modify or remove this rule.

Recommended creating numbered rules in increments (for example, increments of 10 or 100) so that you can insert new rules where you need to later on.

You can **block a single IP address** using NACL, which you can't do using Security Group

Security Group

Control inbound and outbound traffic **at EC2 instance level**

Support **Allow** rules only. All traffic is **deny** by default unless a rule specifically allows it.

Stateful means return traffic is automatically allowed, regardless of any rules

When you first create a security group, It has no inbound rule means **denies all incoming** traffic and one outbound rule that **allows all outgoing** traffic.

You can specify a source in the security group rule to be an **IP range, A specific IP (/32), or another security group**

One security group can be associated with **multiple instances across multiple subnets**

One EC2 instance can be associated with **multiple Security Groups** and rules are **permissive** (instead of restrictive). Meaning if you have one security group which has no Allow and you add an allow in another then it will Allow

Evaluate all rules before deciding whether to allow traffic

Transit gateway is used to create transitive VPC peer connections between thousands of VPCs

hub-and-spoke (star) connection

Support **IP Multicast** (not supported by any other AWS service)

Use as gateway at Amazon side in VPN connection, not at customer side

Can be attached to - one or more VPCs, AWS Direct Connect gateway, VPN Connection, peering connection to another Transit gateway

VPC Flow Logs

Allows you to capture **IP traffic information** in-and-out of Network Interfaces within your VPC
You can turn on Flow Logs at VPC, Subnet or Network Interface level
VPC Flow logs can be delivered to **S3** or **CloudWatch logs**.
Query VPC flow logs using Athena on S3 or CloudWatch logs insight
VPC Flow logs have - Log Version `version`, AWS Account Id `account-id`, Network Interface Id `interface-id`, Source IP address and port `srcaddr` & `srcport`, destination IP address and port `dstaddr` & `dstport`
VPC Flow logs contain source and destination **IP addresses** (not hostnames)

IPv6 are all public addresses, all instances with IPv6 are publicly accessible. for private ranges, we still use IPv4. You can not disable IPv4. If you enable IPv6 for VPC and subnets, then your EC2 instance would get private IPv4 and public IPv6
Cost nothing: VPCs, Route Tables, NACLs, Internet Gateway, Security Groups, Subnets, VPC Peering
Cost money: NAT Gateway, VPC Endpoints, VPN Gateway, Customer Gateway

VPC endpoints

VPC endpoints allow **your VPC to connect to other AWS services privately** within the AWS network
Traffic between your VPC and other services **never leaves the AWS network**
Eliminates the need for an Internet Gateway and NAT Gateway for instances in public and private subnets to access the other AWS services through public internet.
There are two types of VPC endpoints:-

Interface endpoint are Elastic Network Interfaces (ENI) with a private IP address. They serve as an entry point for traffic going to most of the AWS services. Interface endpoints are provided by **AWS PrivateLink** and have an hourly fee and per GB usage cost.
Gateway endpoint is a gateway that is a target **for a specific route** in your **route table**, use to destined for a supported

AWS service. Currently supports only **Amazon S3 and DynamoDB**. Gateway endpoints **are free**

If EC2 instance wants to access S3 bucket or DynamoDB in **different region privately** within AWS network then we first need **VPC inter-region peering** to connect VPC in both regions and then use VPC gateway endpoint for S3 or DynamoDB.

AWS PrivateLink is VPC interface endpoint service to expose a particular service to 1000s of VPCs cross-accounts

AWS ClassicLink (deprecated) to connect EC2-classic instances privately to your VPC

AWS VPN

AWS Site-to-Site VPN connection is created to communicate between your remote network and Amazon VPC **over the internet**

VPN connection: A secure connection between your on-premises equipment and your Amazon VPCs.

VPN tunnel: An encrypted link where data can pass from the customer network to or from AWS. Each VPN connection includes two VPN tunnels which you can simultaneously use for high availability.

Customer gateway: An AWS resource that provides information to AWS about your customer gateway device.

Customer gateway device: A physical device or software application on the customer side of the Site-to-Site VPN connection.

Virtual private gateway: The VPN concentrator on the Amazon side of the Site-to-Site VPN connection. You use a virtual private gateway or a transit gateway as the gateway for the Amazon side of the Site-to-Site VPN connection.

Transit gateway: A transit hub that can be used to interconnect your VPCs and on-premises networks. You use a transit gateway or virtual private gateway as the gateway for the Amazon side of the Site-to-Site VPN connection.

AWS Direct Connect

Establish a **dedicated private connection** from On-premises locations to the AWS VPC network.

Can access public resources (S3) and private (EC2) on the same connection

Provide 1GB to 100GB/s network bandwidth for fast transfer of data from on-premises to Cloud
Not an immediate solution, because it takes a few days to establish a new direction connection

AWS VPN	AWS Direct Connect
Over the internet connection	Over the dedicated private connection
Configured in minutes	Configured in days
low to modest bandwidth	high bandwidth 1 to 100 GB/s

Amazon API Gateway

Serverless, Create and Manage APIs that act as a front door for back-end systems running on EC2, AWS Lambda, etc.
API Gateway Types - HTTP, WebSocket, and REST
Allows you to track and control the usage of API. Set **throttle limit** (default 10,000 req/s) to prevent being overwhelmed by too many requests and returns 429 `Too Many Request` error response. It uses the bucket-token algorithm where the **burst size** is the max bucket size. For a throttle limit of 10000 req/s and a burst of 5000 requests, if 8000 requests are coming in the first millisecond, then 5000 are served immediately and throttle the rest 3000 in the one-second period.
Caching can be enabled to cache your API response to reduce the number of API calls and improve latency

API Gateway Authentication

IAM Policy is used for authentication and authorization of AWS users and leverage **Sig v4** to pass IAM credential in the request header

Lambda Authorizer (formerly Custom Authorizer) use lambda for OAuth, SAML or any other 3rd party authentication

Cognito User Pools only provide authentication. Manage your own user pool (can be backed by Facebook, Google, etc.)

Amazon CloudFront

It's a **Content Delivery Network (CDN)** that uses **AWS edge locations** to cache and deliver **cached content** (such as images and videos)
CloudFront can cache data from **Origin** for e.g.

- S3 bucket using OAI (Origin Access Identity) and S3 bucket policy
- EC2 or ALB if they are public and security group allows

Origin Access Identity (OAI) can be used to restrict the content from S3 origin to be accessible from CloudFront only
supports **Geo restriction (Geo-Blocking)** to whitelist or blacklist countries that can access the content
supports **Web download** distribution (static, dynamic web content, video streaming) and **RTMP Streaming** distribution (media files from Adobe media server using RTMP protocol)
You can generate a **Signed URL** (for a single file and RTMP streaming) or **Signed Cookie** (for multiple files) to share content with premium users
integrates with AWS WAF, a web application firewall to protect from layer 7 attacks
Objects are removed from the cache upon **expiry (TTL)**, by default 24 hours.
Invalidate the Object explicitly for web distribution only with the cost associated, which removes the object from CloudFront cache. Otherwise, you can change the object name, and **versioning** to serve new content.

Amazon Route 53

AWS Managed Service to create DNS Records (Domain Name System)
Browser cache the resolved IP from DNS for TTL (time to live)
Expose public IP of EC2 instances or load balancer
Domain Registrar If you want to use Route 53 for domains purchased from 3rd party websites like GoDaddy.

- AWS - You need to create a **Hosted Zone** in Route 53
- GoDaddy - update the 3rd party registrar NS (name server) records to use Route 53.

Private Hosted Zone is used to create an internal (intranet) domain name to be used within Amazon VPC. You can then add some DNS records and routing policies for that internal domain. That internal domain is accessible from EC2 instances or any other resource within VPC.

DNS Record: Type

CNAME points hostname to any other hostname. Only works with **subdomains** e.g. `something.mydomain.com`
A or AAAA (Alias) points hostname to an AWS Resource like ALB, API Gateway, CloudFront, S3 Bucket, Global Accelerator, Elastic Beanstalk, VPC interface endpoint etc. Works with both root-domain and subdomains e.g. `mydomain.com`. **AAAA** is used for IPv6 addresses.

DNS Record: Routing Policy

Simple to route traffic to specific IP using a single DNS record. Also allows you to return multiple IPs after resolving DNS.
Weighted to route traffic to different IPs based on weights (between 0 to 255) e.g. create 3 DNS records for weights 70, 20, and 10.
Latency to route traffic to different IPs based on AWS regions nearest to the client for low-latency e.g. create 3 DNS records with region us-east-1, eu-west-2, and ap-east-1
Failover to route traffic from Primary to Secondary in case of failover e.g. create 2 DNS records for primary and secondary IP. It is mandatory to create health check for both IP and associate to record.
Geolocation to route traffic to specific IP based on user geolocation (select Continent or Country). Should also create default (select Default location) policy in case there's no match on location.
Geoproximity to route traffic to specific IP based on user geolocation and **bias** value. Positive bias (1 to 99) for more traffic and negative bias (-1 to -99) for less traffic. You can **control the traffic** from specific geolocation using bias value.
Multivalue Answer to return up to 8 healthy IPs after resolving DNS e.g. create 3 DNS records with an associated health check. Acts as client-side Load Balancer, expect a downtime of TTL, if an EC2 becomes unhealthy.

DNS Failover

active-active failover when you want all resources to be available the majority of the time. All records have the same name, same type, and same routing policy such as **weighted or latency**
active-passive failover when you have active primary resources and standby secondary resources. You create two records - primary & secondary with **failover** routing policy

AWS Global Accelerator

Global Service
Global Accelerator **improves the performance** of your application **globally** by **lowering latency and jitter**, and increasing throughput as compared to the public internet.
Use Edge locations and AWS internal global network to find an **optimal pathway** to route the traffic.
First, you create a global accelerator, which provisions **two anycast static IP addresses**.
Then you register **one or more endpoints** with Global Accelerator. **Each endpoint can have one or more AWS resources** such as NLB, ALB, EC2, S3 Bucket or Elastic IP.
You **can set the weight** to choose how much traffic is routed to each endpoint.
Within the endpoint, global accelerator **monitors health checks** of all AWS resources to send traffic to healthy resources only

Management & Governance

Amazon CloudWatch

CloudWatch is used to **collect & track metrics, collect & monitor log files, and set alarms** of AWS resources like EC2, ALB, S3, Lambda, DynamoDB, RDS etc.
By default, CloudWatch will aggregate and store the metrics at Standard 1-minute resolution. You can set max high-resolution at 1 second.
CloudWatch dashboard can include graphs from **different AWS accounts and regions**
CloudWatch has the following EC2 instance metrics - CPU Utilization %, Network Utilization, and Disk Read Write. You need to set up a custom metric for Memory Utilization, Disk Space Utilization, SwapUtilization etc.
You need to install **CloudWatch Logs Agent** on EC2 to collect custom metrics and logs on CloudWatch
You can terminate or recover EC2 instances based on **CloudWatch Alarm**
You can schedule a Cron job using **CloudWatch Events**
Any AWS service should have access to `log:CreateLogGroup`, `log:CreateLogStream`, and `log:PutLogEvents` actions to write logs to CloudWatch

AWS CloudTrail

CloudTrail provides **audit and event history** of all the actions taken by any user, AWS service, CLI, or SDK across AWS infrastructure.
CloudTrail is enabled (applied) by default for all regions
CloudTrail logs can be sent to CloudWatch logs or S3 bucket
Use case: check in the CloudTrail if any resource is deleted from AWS without anyone's knowledge.

AWS CloudFormation

Infrastructure as Code (IaC). Enable modeling, provisioning, and versioning of your entire infrastructure in a text (.YAML) file
Create, update, or delete your **stack of resources** using CloudFormation **template as a JSON or YAML file**
CloudFormation template has a following components:-

- **Resources:** AWS resources declared in the template (mandatory)
- **Parameters:** input values to be passed in the template at stack creation time
- **Mappings:** Static variables in the template
- **Outputs:** Output which you want to see once the stack is created e.g. return ElasticIP address after attaching to VPC, return DNS of ELB after stack creation.
- **Conditionals:** List of conditions to perform resource creation
- Metadata
- Template helpers: References and Functions

Allows **DependsOn** attribute to specify that the creation of a specific resource follows another
Allows **DeletionPolicy** attribute to be defined **for resources** in the template

- **retain** to preserve resources like S3 even after stack deletion
- **snapshot** to backup resources like RDS after stack deletion

Supports **Bootstrap scripts** to install packages, files and services on the EC2 instances by simply describing them in the template
automatic rollback on error feature is enabled, by default, which will cause all the AWS resources that CF created successfully for a stack up to the point where an error occurred to be deleted

AWS CloudFormation **StackSets** allow you to create, update or delete CloudFormation **stacks across multiple accounts, regions, OUs in AWS organization** with a single operation.
Using CloudFormation itself is free, underlying AWS resources are charged
Use case: Use to set up the same infrastructure in different environments e.g. SIT, UAT and PROD. Use to create DEV resources every day in working hours and delete them later to lower the cost

AWS Elastic Beanstalk

Platform as a Service (PaaS)
Makes it easier for developers to quickly deploy and manage applications without thinking about underlying resources
Automatically handles the deployment details of capacity provisioning, load balancing, auto-scaling and application health monitoring
You can launch an **application** with the following pre-configured platforms:-

- Apache Tomcat for Java applications,
- Apache HTTP Server for PHP and Python applications
- Nginx or Apache HTTP Server for Node.js applications
- Passenger or Puma for Ruby applications
- Microsoft IIS 7.5 for .NET applications
- Single and Multi Container **Docker**

You can also launch an **environment** with the following environment tier:-

- An application that serves HTTP requests runs in a **web server environment tier**.
- A backend environment that pulls tasks from an Amazon Simple Queue Service (Amazon SQS) queue runs in a **worker environment tier**.

It costs nothing to use Elastic Beanstalk, only the resources it provisions e.g. EC2, ASG, ELB, and RDS etc.
supports custom AMI to be used
supports **multiple running environments** for development, staging and production, etc.
supports **versioning** and stores and tracks application versions over time allowing easy rollback to prior version

AWS ParallelCluster

Deploy and manage High-Performance Computing (HPC) clusters on AWS using a simple text file
You have full control of the underlying resources.
AWS ParallelCluster is free, and you pay only for the AWS resources needed to run your applications.
You can configure HPC cluster with Elastic Fabric Adapter (EFA) to get OS-bypass capabilities for low-latency network communication

AWS Step Functions (SF)

Build serverless visual workflow to orchestrate your Lambda functions
You write **state machine** in **declarative JSON**, you write a **decider program** to separate activity steps from decision steps.

AWS Simple Workflow Service (SWF)

Code runs on EC2 (not Serverless)
Older service. Use SWF when you need external signal signals to intervene in the process or need the child process to pass value to the parent process, otherwise, use **Step Functions** for new applications.

AWS Organization

Global service to manage multiple AWS accounts e.g. accounts per department, per cost center, per environment (dev, test, prod)
Pricing benefits from **aggregated usage across accounts**.
Consolidate billing across all accounts - single payment method
Organization has multiple **Organization Units (OUs)** (or accounts) based on department, cost center or environment, OU can have other OUs (hierarchy)
Organization has **one master account** and **multiple member accounts**
You can apply **Service Control Policies (SCPs)** at OU or account level, SCP is applied to all users and roles in that account
SPC Deny take precedence over Allow in the full OU tree of an account e.g. allowed at the account level but deny at OU level is = deny
Master account can do anything even if you apply SCP
To merge Firm_A Organization with Firm_B Organization
Remove all member accounts from Firm_A organization

Delete the Firm_A organization
Invite Firm_A master account to join Firm_B organization as a member account

AWS Resource Access Manager (RAM) helps you to create your AWS resources once, and securely share across accounts within OUs in AWS Organization. You can share Transit Gateways, Subnets, AWS License Manager configurations, Route 53 resolver rules, etc.
One account can share resources with another individual account within AWS organization with the help of **RAM**. You must enable resource sharing at AWS Organization level.
AWS Control Tower integrated with AWS Organization helps you to **quickly setup and configure a new AWS account** with **best practices** from base called as **landing zone**

AWS OpsWorks

Provide managed instances of **Chef** and **Puppet** configuration management services, which help to configure and operate applications in AWS.
Configuration as Code - OpsWorks lets you use Chef and Puppet to automate how servers are configured, deployed, and managed across EC2 instances using Code.
OpsWork Stack let you model your application as a stack containing different layers, such as load balancing, database, and application server.

AWS Glue

Serverless, fully managed ETL (extract, transform, and load) service
AWS Glue Crawler scan data from data-source such as S3 or DynamoDB table, determine the schema for data, and then creates metadata tables in the AWS Glue Data Catalog.
AWS Glue provides **classifiers** for CSV, JSON, AVRO, XML or database to determine the schema for data

Containers

ECR (Elastic Container Registry) is Docker Hub to pull and push Docker images, managed by Amazon.

ECS (Elastic Container Service) ECS is a container management service to run, stop, and manage Docker containers on a cluster

ECS Task Definition where you configure task and container definition

Specify **ECS Task IAM Role** for ECS task (Docker container instance) to access AWS services like S3 bucket or DynamoDB

Specify **Task Execution IAM Role** i.e. `ecsTaskExecutionRole` for EC2 (ECS Agent) to pull docker images from ECR, make API calls to ECS service and publish container logs to Amazon CloudWatch on your behalf

Add container by specifying docker image, memory, port mappings, health-check, etc.

You can create multiple ECS Task Definitions - e.g. one task definition to run a web application on the Nginx server and another task definition to run a microservice on Tomcat.

ECS Service Definition where you configure cluster, ELB, ASG, task definition, and number of tasks to run multiple similar **ECS Task**, which deploys a docker container on EC2 instance. One EC2 instance can run multiple ECS tasks.

Amazon EC2 Launch Type: You manage EC2 instances of ECS Cluster. You must install **ECS Agent** on each EC2 instance. Cheaper. Good for predictable, long-running tasks.

ECS Agent The agent sends information about the EC2 instance's current running tasks and resource utilization to Amazon ECS. It starts and stops tasks whenever it receives a request from Amazon ECS

Fargate Launch Type: Serverless, EC2 instances are managed by Fargate. You only manage and pay for container resources. Costlier. Good for variable, short-running tasks

EKS (Elastic Kubernetes Service) is managed Kubernetes clusters on AWS

Cheat Sheet

AWS Service	Keywords
Security	
Amazon CloudWatch	Metrics, Logs, Alarms
AWS CloudTrail	Audit Events

AWS WAF	Firewall, SQL injection, Cross-site scripting (XSS), Layer 7 attacks
AWS Shield	DDoS attack, Layer 3 & 4 attacks
Amazon Macie	Sensitive Data, Personally Identifiable Information (PII)
Amazon Inspector	EC2 Security Assessment, Unintended Network Accessibility
Amazon GuardDuty	Analyze VPC Flow Logs, Threat Detection
AWS VPN	Online Network Connection, Long-term Continuous transfer, Low to Moderate Bandwidth
AWS Direct Connect	Private Secure Dedicated Connection, Long-term Continuous transfer, High Bandwidth
Application Integration	
Amazon SNS	Serverless, PubSub, Fan-out
Amazon SQS	Serverless, Decoupled, Queue, Fan-out
Amazon MQ	ActiveMQ
Amazon SWF	Serverless, Simple Workflow Service, Decoupled, Task Coordinator, Distributed & Background Jobs
AWS Step Functions (SF)	Orchestrate / Coordinate Lambda functions and ECS containers into a workflow
AWS OpsWork	Chef & Puppet
Storage	
EBS	Block Storage Volume for EC2
EFS	Network File System for EC2, Concurrent access
Amazon S3	Serverless, Block Storage - Photos & Videos, Website Hosting
Amazon Athena	Query data in S3 using SQL
AWS Snow Family	Offline Data Migration, Petabyte to exabyte Scale

AWS DataSync	Online Data Transfer, Immediate One-time transfer
AWS Storage Gateway	Hybrid Storage b/w On-premise and AWS
Compute	
AWS Lambda	Serverless, FaaS
Database	
Amazon RDS	Relational Database - PostgreSQL, MySQL, MariaDB, Oracle, and SQL Server
Amazon Aurora	Relational Database - Amazon-Owned
Amazon DynamoDB	Serverless, key-value NoSQL Database - Amazon-Owned
Amazon DocumentDB	Document Database, JSON documents - MongoDB
Amazon Neptune	Graph Database, Social Media Relationship
Amazon Timestream	Time Series Database
Amazon Redshift	Columnar Database, Analytics, BI, Parallel Query
Amazon Elasticache	Redis and Memcached, In-memory Cache
Amazon EMR	Elastic MapReduce, Big Data - Apache Hadoop, Spark, Hive, Hbase, Flink, Hudi
Amazon Elasticsearch Service	Elasticsearch, ELK
Microservices	
Elastic Container Registry (ECR)	Docker image repository, DockerHub
Elastic Container Service (ECS)	Docker container management system
AWS Fargate	Serverless ECS
AWS X-Ray	Trace Request, Debug
Developer	
AWS CodeCommit	like GitHub, Git-based Source Code Repository

AWS CodeBuild	like Jenkins CI, Code Compile, Build & Test
AWS CodeDeploy	Code deployment to EC2, Fargate, and Lambda
AWS CodePipeline	CICD pipelines, Rapid Software or Build Release
AWS CloudShell	CLI, Browser-based Shell
AWS Elastic Beanstalk	PaaS, Quick deploy applications - Java-Tomcat, PHP/Python-Apache HTTP Server, Node.js-Nginx
Amazon Workspaces	Desktop-as-a-Service, Virtual Windows or Linux Desktops
Amazon AppStream 2.0	Install Applications on Virtual Desktop and access it from Mobile, Tab or Remote Desktop through Browser
AWS CloudFormation	Infrastructure as Code, Replicate Infrastructure
AWS Certificate Manager (ACM)	Create, renew, deploy SSL/TLS certificates to CloudFront and ELB
AWS Migration Hub	Centralized Tracking on the progress of all migrations across AWS
AWS Glue	Data ETL (extract, transform, load), Crawler, Data Catalogue
AWS AppSync	GraphQL
Amazon Elastic Transcoder	Media (Audio, Video) converter

Important Ports

Protocol/Database	Port
FTP	21
SSH	22
SFTP	22
HTTP	80
HTTPS	443
RDP	3389

NFS	2049
PostgresSQL	5432
MySQL	3306
MariaDB	3306
Aurora	3306 or 5432
Oracle RDS	1521
MSSQL Server	1433

White Papers

Disaster Recovery

RPO - Recovery Point Objective - How much data is lost to recover from a disaster e.g. last 20 min data lost before the disaster
RTO - Recovery Time Objective - How much downtime require to recover from a disaster e.g. 1-hour downtime to start disaster recovery service
Disaster Recovery techniques (RPO & RTO reduces and the cost goes up as we go down)

- **Backup & Restore** – Data is backed up and restored, with nothing running
- **Pilot light** – Only minimal critical service like RDS is running and the rest of the services can be recreated and scaled during recovery
- **Warm Standby** – Fully functional site with minimal configuration is available and can be scaled during recovery
- **Multi-Site** – Fully functional site with identical configuration is available and processes the load

Use **Amazon Aurora Global Database** for RDS and **DynamoDB Global Table** for NoSQL databases for disaster recovery with stringent RPO of 1 second and RTO of 1 minute.

5 Pillars of the AWS Well-Architected Framework

The 5 Pillars of **AWS Well-Architected Framework** are as follows:-

Operational Excellence

Use **AWS Trusted Advisor** to get recommendations on AWS best practices, optimize AWS infrastructure, improve security and performance, reduce costs, and monitor service quotas

Use **Serverless application** API Gateway (Front layer for auth, cache, routing), Lambda (Compute), DynamoDB (Database), DAX (Caching), S3 (File Storage) and Cognito User Pools (Auth), CloudFront (Deliver content globally), SES (Send email), SQS & SNS (Publish & Notify events)

Security

Use **AWS Shield** and **AWS WAF** to prevent network, transport and application layer security attacks

Reliability

Performance Efficiency

Cost Optimization

Use **AWS Cost Explorer** to forecast daily or monthly cloud costs based on ML applied to your historical cost

Use **AWS Budget** to set yearly, quarterly, monthly, daily or fixed cost or usage budget for AWS services and get notified when actual or forecast cost or usage exceeds budget limit.

Use **AWS Saving Plans** to get a discount in exchange for usage commitment e.g. $10/hour for one-year or three-year period. AWS offers three types of Savings Plans – **1. Compute Savings Plans** apply to usage across Amazon EC2, AWS Lambda, and AWS Fargate. **2. EC2 Instance Savings Plans** apply to EC2 usage, and **3. SageMaker Savings Plans** apply to SageMaker usage.

Use **VPC Gateway endpoint** to access S3 and DynamoDB privately within AWS network to reduce data transfer cost

Use **AWS Organization** for consolidated billing and aggregated usage benefits across AWS accounts