AWS

Contents

AWS

1) Cloud:

- The cloud is a network of remote servers connected to the internet.
- It's like a vast digital storage space where data can be stored, managed, and accessed from anywhere with an internet connection.
- Instead of relying on physical storage devices, you store your data on these remote servers.

[Note: A server is essentially a computer or program that provides services to other computers, known as clients]

2) Cloud Computing:

- Store data/Apps on remote servers
- Process data/Apps on remote servers
- Access data/Apps on remote servers

➔ Cloud computing is the delivery of computing services (such as storage, servers, databases, networking, software, and analytics) over the internet (the cloud), allowing for flexible resources, scalability, and cost efficiency.

➔ So, instead of having everything on your local computer, the cloud allows you to access it from anywhere with an internet connection. This makes it a very attractive option for businesses and individuals alike.

(i) Cloud services Types (SaaS, PaaS, IaaS)

SaaS (software as a service) (consume)	PaaS (platform as a service) (build)	IaaS (infrastructure as a service) (host)
Ready-to-use applications accessible through a web browser or app.	Offers a platform and environment to develop, run, and manage applications without worrying about the underlying infrastructure	Provides virtualized computing resources over the internet, including servers, storage, and networking.
eg) Gmail, Dropbox, and Salesforce.	eg) GitHub, Kubernetes, docker	eg) AWS, Azure
Use cases: Buying applications like outlook, ondrive, skype	Use casees: Development framework. Analytics & Business intelligence.	Use cases: Migration of workloads. Test and deployment. Storage, backups and recovery.

3) Regions, Availability zones and Edge locations

(i) Regions: (geographic locations)

Geographic areas that contain multiple data centers.
Currently, 33 regions are available.

(ii) Availability Zones (AZs): (collection of datacentres with in a region)

Multiple, physically separate data canters located within a single region, providing fault-tolerance. (ability to keep running even if a software or hardware fails)
(Connected with each other)
Currently, over 105 Availability Zones across all regions
[Note: Region is a geographic location which hosts two or more Availability Zones]

(iii) Edge Locations: (where end users access services located at AWS)

Data centers that deliver content and services closer to end users to reduce latency.
Currently, over 600 CloudFront Points of Presence (PoPs) and 13 Regional Edge Caches

4) EC2 Dashboard:

(i) Elastic Compute Cloud (EC2):

EC2 is a core service offered by Amazon Web Services (AWS) that provides scalable computing resources in the cloud. It allows you to rent virtual computers (instances) on which you can run your own applications.

(ii) Instances:

Instance is a cloud-server used for deploying applications.

(a) Diff between instance families: t2, t3, a1, c1, c3, c4, c5

T2/T3: Flexible, burstable performance for general use.
A1: Cost-effective, energy-efficient with Arm processors.
C1/C3/C4/C5: Increasingly powerful compute-optimized instances for heavy CPU tasks. (High Performance)
For high Performance we can go with c or m series.

Instance Family	Key Features	Use Cases
T2	Burstable performance	Web servers, small databases
T3	Improved burstable performance, cost-effective	Developer environments, microservices
A1	Arm-based, energy-efficient	Web servers, containerized applications
C1	Older compute-optimized	Batch processing, media transcoding
C3	Enhanced networking, additional memory	HPC, scientific modelling
C4	High CPU performance, cost-efficient	Ad serving, web servers
C5	Latest generation, high performance	Machine learning, video encoding

[Note:
Burstable Performance: Allows instances to handle occasional spikes in CPU demand by using performance credits.
Arm Processors: Energy-efficient CPUs that provide cost savings and are ideal for specific workloads, particularly in cloud environments]

(b) Launch templates:

A launch template is a blueprint for launching EC2 instances. It specifies the configuration parameters for an EC2 instance, such as the AMI, instance type, key pair, security groups, and other settings.

(iii) Amazon Machine Images (AMI): (Images)

- It's essentially a preconfigured template containing the software, applications, and data required to launch an EC2 instance.
- You must specify an AMI when you launch an instance.
- You can launch multiple instances from a single AMI when you require multiple instances with the same configuration.
- You can use different AMIs to launch instances when you require instances with different configurations.

(iv) EBS: (Elastic Block Store)

It's a block-level storage service provided by AWS that offers persistent storage volumes for EC2 instances. Think of it as a virtual hard drive for your virtual server.

(a) Volume: (storage)

(i) General Purpose SSD (gp2/gp3)
(ii) Provisioned IOPS (Input output per sec) SSD (io1/io2 Block Express)
(iii) Magnetic (Standard HDD)

Feature	General Purpose SSD (gp2/gp3)	Provisioned IOPS SSD (io1/io2)	Magnetic HDD (st1)
Description	Balanced performance and cost-effective	High performance, low latency	High throughput, low cost
Real-World Use Cases	Web servers, application servers, development and test environments, small to medium-sized databases	NoSQL databases, in-memory databases, financial applications	Big data, data warehousing, log processing
IOPS	Up to 16,000	Up to 64,000	Up to 500
Throughput	Up to 250 MB/s	Up to 4,000 MB/s	Up to 500 MB/s
Cost	Moderate	High	Low
Other Features	Burstable performance	Consistent performance	Optimized for throughput

[Note:
For most applications, a general-purpose SSD (gp2) is a good starting point due to its balanced performance and affordability.
If you need the absolute fastest and most consistent performance, consider provisioned IOPS SSD. However, be prepared for a higher cost.
Use magnetic storage for data that doesn't require frequent access and prioritize cost-effectiveness]

(b) Snapshot: (Capture of EBS volume)

An AWS Snapshot is a point-in-time copy of an Amazon EBS (Elastic Block Store) volume. It's like a backup of your virtual hard drive.

(v) Diff b/w Image (AMI) and Snapshot

Feature	Image (AMI)	Snapshot
Definition	A complete copy of an instance at a specific point in time	A point-in-time copy of a disk or volume, capturing only changes since the previous snapshot
Size	Larger file size as it contains all data	Smaller file size as it only contains changed data
Cost	Higher storage cost due to larger size	Lower storage cost due to smaller size
Usage	Creating new instances, deploying applications	Primarily used for backups, disaster recovery, and restoring data to a specific point in time
Additional notes	Can be used as a base image for multiple instances	Can be chained to create incremental backups
Bootable	AMI is bootable	Snapshot is non-bootable

(vi) Network & Security

(a) Security groups

A security group acts as a virtual firewall for your EC2 instances, controlling the incoming and outgoing traffic.
It defines which traffic is allowed to reach your instances.

(b) Elastic IP

An Elastic IP address (EIP) is a static, public IPv4 address designed for dynamic cloud computing.
It's essentially a fixed IP address that you can associate with an EC2 instance.

(c) Keypairs: key for instance

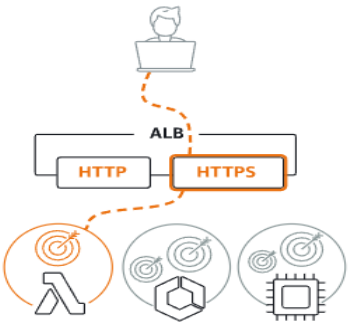
A key pair in AWS is a combination of a public key and a private key used for secure communication with EC2 instances.

(vii) Load Balancing

- A load balancer is a component that distributes incoming traffic across multiple targets (typically EC2 instances). It acts as a single point of contact for clients, ensuring high availability and scalability.
- The process of distributing traffic among multiple servers to improve a service or application's performance and reliability
- Load balancers improve application performance by increasing response time and reducing network latency.
- They perform several critical tasks such as the following:
  - Distribute the load evenly between servers to improve application performance.
  - Redirect client requests to a geographically closer server to reduce latency.

(a) 3 types: classic, application and network:

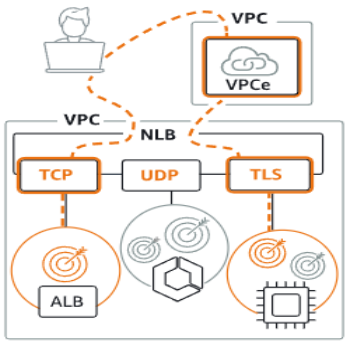
Application Load Balancer Info



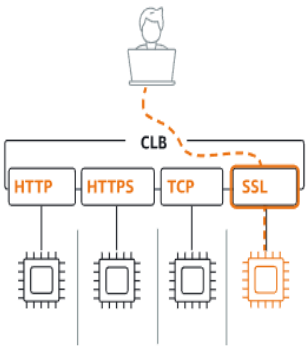
Choose an Application Load Balancer when you need a flexible feature set for your applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.

Create

Network Load Balancer Info



Choose a 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. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second securely while maintaining ultra-low latencies.



Choose a Classic Load Balancer when you have an existing application running in the EC2-Classic network.

Create

Types	Application Load Balancer (ALB)	Network Load Balancer (NLB)	Classic Load Balancer (CLB)
Definition	Operates at the application layer (Layer 7). Supports HTTP/HTTPS traffic. Offers advanced features like path-based routing, and health checks.	Operates at the transport layer (Layer 4). Supports TCP, TLS, and UDP protocols. Offers extremely low latency and high throughput.	Older generation load balancer, gradually being replaced by ALB and NLB. Supports HTTP, HTTPS, and TCP protocols.
Use Cases	Web applications, microservices, containerized applications, multi-protocol applications.	High-performance applications, gaming servers, network-based applications, load balancing across multiple Availability Zones.	Legacy applications, simple load balancing scenarios.
When to Use	When you need advanced routing, load balancing based on application layer attributes, and support for HTTP/HTTPS protocols.	When you need low latency, high throughput, and support for TCP, TLS, or UDP protocols.	For existing applications using CLB, but consider migrating to ALB or NLB for newer applications.
Protocol	HTTP/HTTPS	TCP, UDP, TLS	TCP, HTTP, HTTPS
Features	Path-based routing, sticky sessions, SSL termination, health checks, target grouping	Low latency, connection-oriented traffic, UDP support	Basic load balancing, health checks
Layer	Layer 7	Layer 4	Layer 4

(b) Target groups:

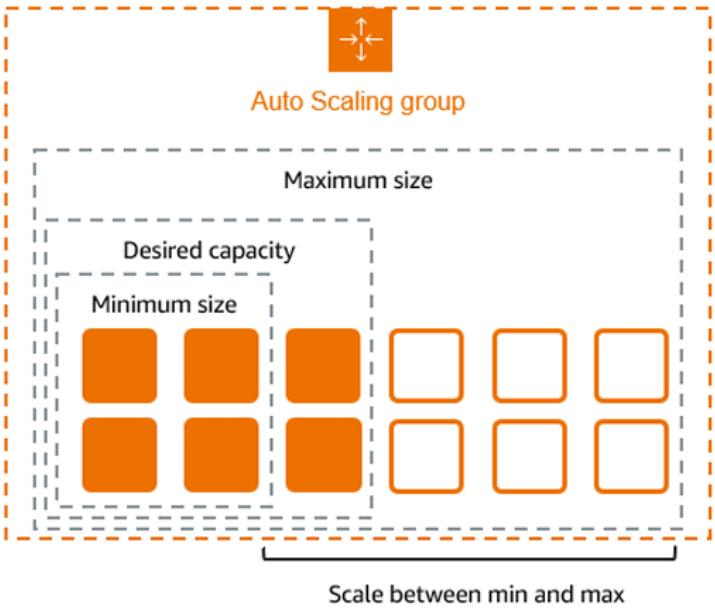
A target group is a collection of targets (EC2 instances, IP addresses, or Lambda functions) that a load balancer distributes traffic to.

It defines the protocol and port for traffic to be routed to the targets.

(viii) Auto Scaling

Cloud computing feature that automatically adjusts the number of compute resources (such as instances) in response to changing demand.

This ensures optimal performance and cost-efficiency.



(a) Auto-Scaling policies:

Policy Type	Description	Use Case
Dynamic Scaling	Combines various scaling policies (e.g., target tracking and step scaling) to dynamically adjust capacity based on demand and predefined metrics.	Useful for applications with varying load patterns.
Target Tracking	Automatically adjusts the number of instances to maintain a specified metric (e.g., CPU utilization) at a target value.	Useful for maintaining performance at a specific level.
Step Scaling	Scales the number of instances based on a set of scaling adjustments defined by thresholds. Each step specifies a range of values for a metric and the corresponding scaling action.	Useful for handling sudden spikes or drops in demand.
Simple Scaling	Scales the number of instances based on a single threshold. When the threshold is crossed, a scaling action (add or remove instances) is performed.	Useful for basic scaling needs with simple threshold-based rules.
Scheduled Scaling	Automatically scales the number of instances at specific times or on a recurring schedule.	Useful for predictable changes in load, such as increased traffic during business hours.
Predictive Scaling	Uses machine learning models to predict future traffic and scale resources in advance based on these predictions.	Useful for applications with predictable traffic patterns.

(b) Instance Refresh:

Instance Refresh is a feature in AWS EC2 Auto Scaling that allows to automatically deploy updates to instances within an Auto Scaling group.

Use Cases

- **Deploying new AMIs or user data scripts:** Update application code or configuration without manual intervention.
- **Updating instance configurations:** Apply new configurations, such as security patches or performance optimizations.
- **Migrating instance types:** Optimize performance or reduce costs by switching to new instance types.
- **Rolling back failed deployments:** Revert to previous configuration in case of issues.

  [**Note:** To update an instance's application or configurations, we perform an instance refresh. First, we apply the changes and create an AMI. Then, we use this AMI to create a launch template and replace the production instance with minimal downtime. If the process takes longer, a 15–30 minute maintenance window may be scheduled. After the updated instance is attached to the Auto Scaling group, the old instance is terminated.]

(c) TCP and UDP protocol

TCP and UDP are both fundamental protocols that handle data transmission over networks, but they take different approaches:

Feature	TCP (Transmission Control Protocol) (Connection-Oriented)	UDP (User Datagram Protocol) (Connectionless)
Definition	TCP is connection-oriented. It establishes a connection between sender and receiver before sending data.	UDP is connectionless. It sends data packets directly without creating a connection.
Reliability	Reliable (guaranteed delivery, error checking, retransmission)	Unreliable (no guaranteed delivery, no error checking)
Order of delivery	Guaranteed order of delivery	No guaranteed order of delivery
Flow control	Yes (prevents congestion)	No
Congestion control	Yes	No
Error checking	Extensive error checking (checksum, sequence numbers, acknowledgments)	Basic error checking (checksum)
Speed	Slower due to reliability and error checking	Faster due to simplicity and lack of overhead
Security	More secure due to connection-oriented nature	Less secure due to connectionless nature
Examples	Web browsing, email, file transfer,	Video streaming, online gaming,

(d) Error code 1XX, 2XX, 3XX, 4XX, 5XX

Code Range	Meaning	Description	Example
1XX (Informational)	Request Received	The server has acknowledged the request and is processing it.	100 Continue
2XX (Success)	Request Successful	The request was received, understood, and processed successfully.	200 OK
3XX (Redirection)	Further Action Needed	The server needs further information to complete the request, such as redirecting to a different URL.	301 Moved Permanently
4XX (Client Error)	Request Error	There's an issue with the request itself, like a typo in the URL or missing data.	404 Not Found
5XX (Server Error)	Server Error	The server encountered an error and couldn't complete the request.	500 Internal Server Error

Status Code Class	Code	Meaning	Description
1XX: Informational	100	Continue	Request received, continue sending the request body
	101	Switching Protocols	Switching to the protocol requested by the client
	102	Processing (WebDAV)	Server has received and is processing the request, but no response is available yet
2XX: Success	200	OK	Request succeeded
	201	Created	Request succeeded and a new resource was created
	202	Accepted	Request accepted, but processing not complete
	203	Non-Authoritative Information	Request succeeded, but the information may come from a third party
	204	No Content	Request succeeded, but no content is being returned
	205	Reset Content	Request succeeded, reset the view that sent the request
	206	Partial Content	Request succeeded, but only part of the resource is returned
	207	Multi-Status (WebDAV)	Multiple independent responses in a single response body

	208	Already Reported (WebDAV)	Member of a collection has already been reported in a previous response
	226	IM Used	Request succeeded; content has been transformed as indicated in the response headers
3XX: Redirection	300	Multiple Choices	Multiple options for the resource, user can choose one
	301	Moved Permanently	Resource has been permanently moved to a new URL
	302	Found	Resource temporarily moved to a different URL
	303	See Other	Resource can be found at another URL; use GET method to retrieve it
	304	Not Modified	Resource has not been modified since last requested
	305	Use Proxy	Resource must be accessed through a proxy specified in the response
	306	(Unused)	Previously used, no longer used
	307	Temporary Redirect	Resource temporarily moved to another URL, use the original URL for future requests
	308	Permanent Redirect	Resource permanently moved to another URL, use this new URL for future requests
4XX: Client Error	400	Bad Request	Server cannot process request due to client error
	401	Unauthorized	Authentication required to access the resource
	402	Payment Required	Reserved for future use
	403	Forbidden	Server understood the request but refuses to authorize it
	404	Not Found	Server cannot find the requested resource
	405	Method Not Allowed	Method used in the request is not allowed for the resource
	406	Not Acceptable	Requested resource not capable of generating acceptable content
	407	Proxy Authentication Required	Client must authenticate with the proxy

	408	Request Timeout	Server timed out waiting for the request
	409	Conflict	Request could not be processed due to conflict with the current state of the resource
	410	Gone	Resource is no longer available and will not be available again
	411	Length Required	Length of content is required and was not specified
	412	Precondition Failed	Server does not meet a precondition specified in the request
	413	Payload Too Large	Request payload is too large for the server to process
	414	URI Too Long	URI provided in the request is too long for the server to process
	415	Unsupported Media Type	Media type of the request is not supported by the server
	416	Range Not Satisfiable	Client requested a range not satisfiable by the server
	417	Expectation Failed	Server cannot meet the requirements of the Expect request-header field
	418	I'm a teapot	Joke status code indicating the server is a teapot, not a coffee machine
	421	Misdirected Request	Request was directed at a server unable to produce a response
	422	Unprocessable Entity (WebDAV)	Request was well-formed but could not be followed due to semantic errors
	423	Locked (WebDAV)	Resource that is being accessed is locked
	424	Failed Dependency (WebDAV)	Request failed due to failure of a previous request
	425	Too Early	Server is unwilling to risk processing a request that might be replayed
	426	Upgrade Required	Client should switch to a different protocol
	428	Precondition Required	Server requires request to be conditional
	429	Too Many Requests	User has sent too many requests in a given time period
	431	Request Header Fields Too Large	Server unwilling to process the request because its header fields are too large

	451	Unavailable For Legal Reasons	Access to the resource is denied due to legal reasons
5XX: Server Error	500	Internal Server Error	Generic server error message
	501	Not Implemented	Server does not recognize or cannot fulfill the request method
	502	Bad Gateway	Server received an invalid response from the upstream server
	503	Service Unavailable	Server is currently unable to handle the request due to temporary overload or maintenance
	504	Gateway Timeout	Server did not receive a timely response from the upstream server
	505	HTTP Version Not Supported	Server does not support the HTTP protocol version used in the request
	506	Variant Also Negotiates	Internal configuration error on the server
	507	Insufficient Storage (WebDAV)	Server unable to store the representation needed to complete the request
	508	Loop Detected (WebDAV)	Server detected an infinite loop while processing the request
	510	Not Extended	Further extensions to the request are required for the server to fulfill it
	511	Network Authentication Required	Client needs to authenticate to gain network access

(e) Port Numbers

Protocol	Port Number	Full Form	Description
HTTP	80	Hypertext Transfer Protocol	Standard web traffic for communication between browsers and servers.
HTTPS	443/ 8443	Secure Hypertext Transfer Protocol	Secure encrypted web traffic, often used for sensitive information like online banking.
SSH	22	Secure Shell	Secure remote login protocol for managing servers.

SMTP	25	Simple Mail Transfer Protocol	Sending emails.
DNS	53	Domain Name System	Translates website names (like [invalid URL removed]) into IP address for browsers.
DHCP	67	Dynamic Host Configuration Protocol	Assigns IP addresses to devices on a network automatically.
IMAP	143	Internet Message Access Protocol	Accessing emails on a server.
LDAP	389	Lightweight Directory Access Protocol	Managing user accounts and permissions in a network.
MSSQL	1433	Microsoft SQL Server	Relational database management system by Microsoft (default port).
MySQL	3306	MySQL	Open-source relational database management system (default port).
RDP	3389	Remote Desktop Protocol	Remote access to a graphical user interface of another computer.
NFS	2049	Network File System	Sharing file systems across a network.
RDS	Varies/ 1150-65535	Remote Desktop Services	Microsoft technology for remote desktop access (default port depends on configuration).
PostgreSQL	5432	PostgreSQL	Open-source relational database management system (default port).

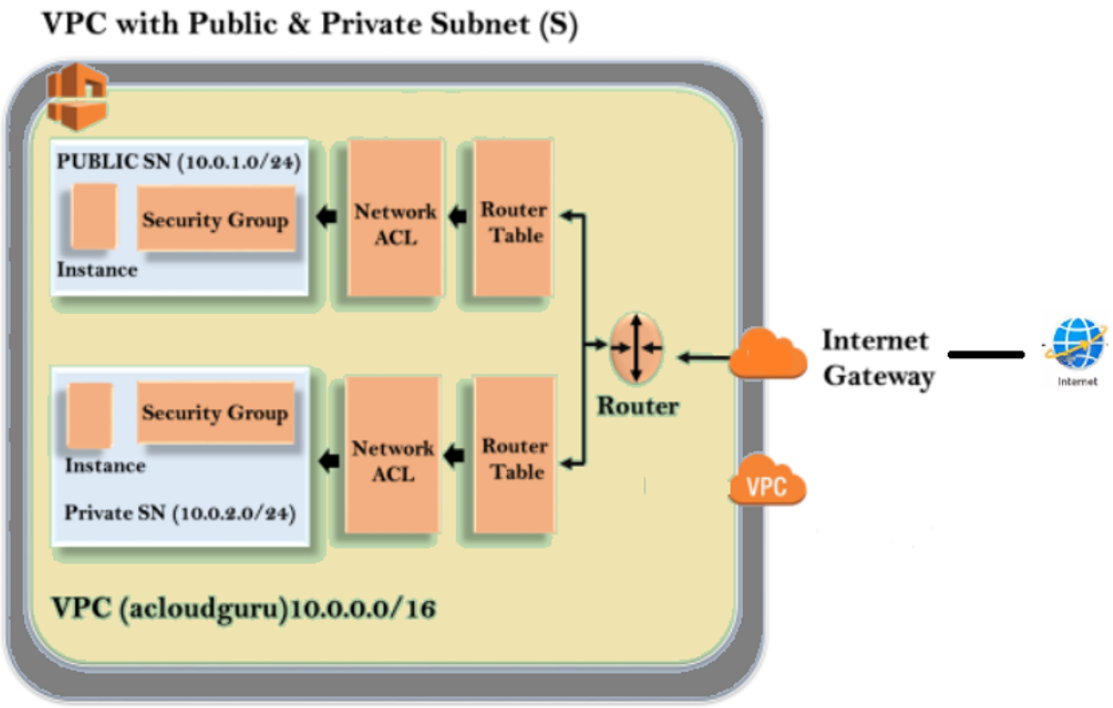
5) ***Virtual Private Cloud (VPC):***

- VPC (Virtual Private Cloud) is a secure, isolated section of the AWS cloud that we can define. It provides complete control over virtual networking environment, including the selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways.
- Size of VPC can be defined by IP address range.
- Max of 65536 IP address can be allocated in VPC.

VPC Components:

(i) Subnet (Public & Private Subnet)
(ii) NACL (Network Access Control List)
(iii) Internet Gateway
(iv) NAT Gateway (Network Address Translation)
(v) Route tables (Public & Private Route Tables)
(vi) Security Group

(i) Architecture of VPC



(i) Subnets: A range of IP addresses within a VPC that resides in a single Availability Zone.

(ii) IGW (Internet Gateway): Enables communication between VPC and Internet.

(iii) RT (Route Table): set of rules that determine where the network traffic from subnet or gateway is directed.

(iv) Security groups: instance firewall

- Act as virtual firewalls for EC2 instances, controlling inbound and outbound traffic.

(v) NaCl (Network Access Control List): subnet firewall

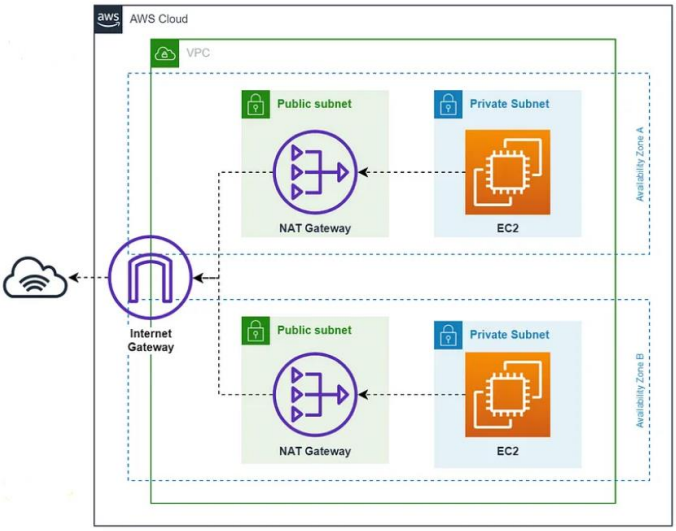
- Additional/optional layer of security that acts as a firewall for controlling traffic in and out of a subnet.

(vi) NAT Gateway (Network Address Translation): unidirectional (instance to internet)

- Allows instances in a private subnet to access the internet, while at the same time, preventing the internet from initiating connections with those instances.

[Public Subnet --> NAT --> Internet Gateway --> Private Subnet]

[Note: only request from Instance will be taken by NAT and through Internet Gateway it will download/upgrade required resources to the Private instance but any request coming from Internet will be denied]



How it works:

(i) **VPC Creation:** Create a VPC with a specific IP address range.
(ii) **Subnet Creation:** Divide the VPC into subnets, each residing in a specific Availability Zone.
(iii) **Internet Gateway Attachment:** Attach an Internet Gateway to allow public access.
(iv) **Route Table Creation:** Create route tables to define traffic flow.
(v) **Subnet Association:** Associate subnets with route tables.
(vi) **Security Group Creation:** Create security groups to control traffic for EC2 instances.
(vii) **Network ACL Creation:** Create optional Network ACLs for additional security.
(viii) **Firewalls:**
NACL --> subnet level
Security Groups --> Instance level

(ii) Difference b/w security group and NACL

Feature	Security Group	NACL
Scope	Instance-level	Subnet-level
Functionality	Acts as a firewall for individual EC2 instances	Controls traffic in and out of a subnet
Rules	Primarily allow inbound traffic. Outbound traffic is generally allowed by default.	Can allow or deny both inbound and outbound traffic.
Statefulness	Stateful, meaning return traffic is automatically allowed.	Stateless, meaning return traffic must be explicitly allowed.

(iii) Elastic IP addresses (EIPs):

An Elastic IP address (EIP) is a static, public IPv4 address designed for the dynamic nature of cloud computing.

(iv) Web server and application server

Feature	Web Server	Application Server
Main Function	Serve static content	Generate dynamic content
Data Processing	Minimal	Complex processing and logic
Protocols	Primarily HTTP, may support others	Supports various protocols including HTTP
Resource Usage	Lower	Higher
Examples	Apache, Nginx, Tomcat	JBoss, WebLogic

(v) 3 types of application:

(a) Static:

Applications with fixed content that doesn't change frequently.

- 1-Tier Architecture (all the components of an application reside within a single layer or location)
- These are web servers
  eg) Apache, nginx
- Used for small applications

AWS Services:

- **Amazon S3:** Ideal for storing static content like images, CSS, JavaScript, and HTML files.
- **Amazon CloudFront:** Distributes static content globally for faster delivery.
- **Amazon Route 53:** Manages DNS records for your static website.

(b) Dynamic:

Applications with content that changes frequently, often based on user interactions or real-time data.

- 2-Tier Architecture (A 2-tier architecture divides an application into two parts:
  Client Tier: Handles the user interface and application logic.
  Server Tier: Manages data storage and retrieval.)
- Servlets and Java server pages (JSPs): Java programs that run on a Java application server and extend the capabilities of the Web server.
- used to connect between application and Database
- eg) tomcat, Apache TomEE
- Used for medium applications

AWS Services:

- **Amazon EC2:** Provides virtual servers for running dynamic applications.
- **Amazon Elastic Beanstalk:** Simplifies deployment and management of dynamic applications.
- **AWS Lambda:** Serverless compute for running dynamic code without managing servers.

(c) Business Logic:

Applications that handle core business processes and rules.
Used for large applications where lot of logics, calculations are required like e-commerce, stock market.

AWS Services:

- **AWS Lambda:** For serverless execution of business logic.
- **Amazon EC2:** For running business logic on virtual servers.
- **AWS Elastic Beanstalk:** For deploying and managing business logic applications.
- **Amazon DynamoDB:** NoSQL database for storing business data

[**Note:** Depending upon application size and type, we need to choose application to deploy]

6) *Identity and Access Management (IAM):*

- Fundamental service for securely controlling access to AWS resources for user.
- It allows to manage users, groups, and roles, as well as define permissions that control what actions these identities can perform on AWS resources.

Key Components of IAM

- **Users:** Individuals who need access to AWS resources.
- **Groups:** Collections of users sharing similar permissions.
- **Roles:** Permissions granted to users or services for specific tasks or actions.
- **Policies:** Define permissions and access levels for users, groups, and role.

Benefits of Using IAM:

- **Enhanced Security:** Protects AWS resources by controlling access to sensitive information.
- **Improved Efficiency:** Streamlines user management and permission assignments.
- **Compliance:** Helps meet security and compliance requirements.
- **Granular Control:** Allows to define precise permissions for different users and roles.

Common IAM Use Cases:

- **Creating users and groups:** Managing access for employees and contractors.
- **Assigning permissions:** Granting appropriate access to AWS resources.
- **Using roles:** Providing temporary access to AWS resources for applications or services.
- **Implementing MFA:** Adding an extra layer of security to user accounts.

7) *Cloudwatch:*

It is monitoring service for AWS resources and applications.
It enables to collect and track metrics, monitor log files, set alarms, and automatically react to changes in your AWS resources.

Key CloudWatch Concepts

- **Metrics:** Numerical data points that represent the state of a system at a particular point in time.

- **Alarms:** Actions triggered based on metric thresholds.
- **Logs:** Textual data generated by applications and systems.
- **Events: Captures changes in AWS resources.**

Benefits

- Performance Optimization
- Cost Reduction
- Troubleshooting
- **Capacity Planning:** Forecast resource needs based on historical data.
- **Compliance:** Monitor for compliance with security and operational standards.

8) *Cloudtrail:*

CloudTrail is a web service that records API (Application Programming Interface) activity in your AWS account.
It continuously logs and monitors the activities and actions across your AWS account. It also provides the event history of your AWS account including information about who is accessing your AWS services.

Key Concepts

- **Trails:** A collection of log files.
- **Events:** Records of API calls.
- **S3 Bucket:** Storage location for log files.
- **Event History:** Allows viewing, searching, & downloading the past 90 days activity.
- **CloudTrail Lake:** Stores and queries log data for up to seven years.

➔ *Difference between Cloudwatch and Cloudtrial?*

CloudWatch	CloudTrail
CloudWatch is basically a monitoring service for AWS resources and applications.	CloudTrail is a web service that is mainly concerned with what is done on AWS and by whom.
By default, CloudWatch offers free basic services like monitoring our AWS resources.	CloudTrail is also enabled by default when we create our AWS Free Tier account.
Using CloudWatch we can track metrics and monitor logs.	CloudTrail provides greater visibility into user activity by tracking AWS console actions, including who made the call, from which IP address, and when.
CloudWatch records the application logs.	CloudTrail provides information about what occurred in your AWS account.
CloudWatch delivers metric data in 1-minute periods for detailed monitoring and 5-minute periods for basic monitoring.	CloudTrail delivers an event within 15 minutes of the API call.
CloudWatch stores data in its own dashboard in the form of metrics and logs.	CloudTrail centralizes all the logs across the regions and stores them on in S3 bucket.

9) SNS: (Simple Notifications Service)

SNS is a fully managed pub/sub messaging service provided by AWS.
It enables to send push notifications to various endpoints, including mobile devices, email addresses, and other AWS services.

Key Features of SNS:

- **Pub/Sub Model:** SNS operates on a publish-subscribe model, where publishers send messages to a topic, and subscribers receive those messages.
- **Push Notifications:** Delivers messages to various endpoints like mobile devices (iOS, Android), email addresses, and HTTP/HTTPS endpoints.
- **Fan-out:** A single message can be delivered to multiple subscribers efficiently.
- **Filtering:** Allows you to filter messages based on attributes.
- **Batching:** Enables sending multiple messages in a single request.
- **Ordering:** Guarantees message delivery order within a partition. (Order of messages is preserved)
- **Deduplication:** Prevents duplicate messages from being delivered.
- **Dead-Letter Queues (DLQs):** Provides a mechanism to handle undeliverable messages.
- **Integration with Other AWS Services:** Works seamlessly with other AWS services like SQS, Lambda and more.

10) EFS (Elastic File System):

Amazon Elastic File System (EFS) is a fully managed, scalable file storage service offered by AWS. It provides shared access to file data for multiple EC2 instances, containers, and Lambda functions.

Benefits:

- **Elastic scalability:** Automatically adjusts storage capacity to meet demand.
- **High throughput (efficiency):** Handles large amounts of data transfer efficiently.
- **Serverless:** No need to manage underlying file servers, simplifying operations.
- **Integration:** Seamlessly integrates with other AWS services (EC2, Lambda, etc.).
- **Pay-as-you-go:** Only pay for the storage you use, optimizing costs.
- **Shared file system:** Multiple EC2 instances can access the same file system simultaneously.

11) Amazon S3 (Simple Storage Service):

- S3 is a highly scalable, secure, and durable object storage service provided by Amazon Web Services (AWS).
- It is designed to store and retrieve any amount of data from anywhere on the web, making it ideal for use cases like data backup, web application hosting, etc.
- S3 is cost-effective as it's less expensive than EBS and EFS.
- It is global level as we can get same data in all regions.

Key Concepts

- **Bucket:** A container for objects in S3.
- **Object:** A file stored in an S3 bucket.

- **Versioning:** Enables you to keep multiple versions of an object.
- **Lifecycle Management:** Automatically transitions objects between storage classes.
- **Storage Classes:** Different storage options with varying cost and performance characteristics.
- **Access Control Lists (ACLs):** Control access to objects.
- **Bucket Policies:** Control access to buckets.
- **IAM:** Used for granular control over S3 resources.

(i) Diff b/w EBS, EFS and S3 storage:

Feature	EBS	EFS	S3
Storage Type	Block	File	Object
Access	Single EC2 instance	Multiple EC2 instances	Publicly accessible
Performance	High	Medium	Varies
Scalability	Up to 16 TB per volume	8 exabytes (8e+6) per file	Virtually unlimited
Cost	Higher	Medium	Lower
Use Cases	Boot volumes, databases	Shared file systems, content management	Backup, static websites, content delivery

(ii) Encryption:

- Encryption in Amazon S3 ensures that data is protected both at rest and in transit. Amazon S3 offers several encryption options, that allow to choose the level of control and security that meets our needs.
- If an unauthorized person gets access to the encrypted data, the data is unreadable without the key or password.

3types:

(a) Server-side encryption with Amazon S3 managed keys (SSE-S3)

- Key Management: Amazon S3 generates and manages encryption keys.
- Control: You have limited control over key management.
- Security: Provides basic data protection.
- Use Cases: Suitable for most use cases where high-level security is not a primary concern.

(b) Server-side encryption with AWS Key Management Service keys (SSE-KMS)

- Key Management: AWS KMS generates and manages encryption keys.
- Control: You have more control over key management through KMS policies.
- Security: Offers enhanced security compared to SSE-S3.
- Use Cases: Ideal for sensitive data where fine-grained control over key management is required.

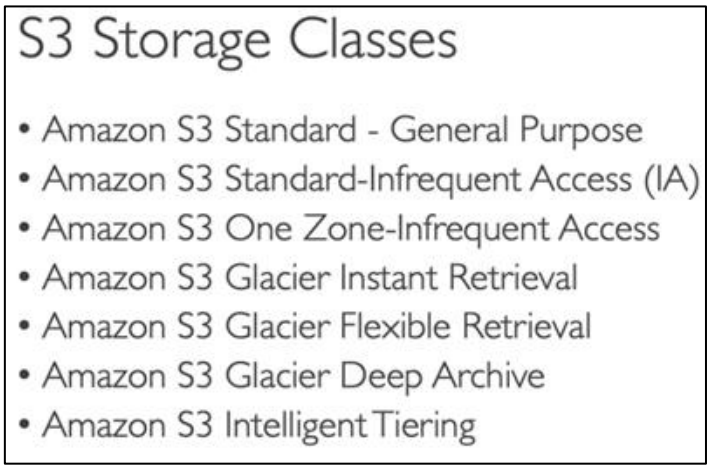
(c) Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)

- Key Management: Both Amazon S3 and AWS KMS manage encryption keys.
- Control: Provides an additional layer of security by using two encryption keys.
- Security: Offers the highest level of security among the three options.
- Use Cases: Suitable for highly sensitive data that requires maximum protection.

(iii) Lifecycle rules:

- S3 Lifecycle Rules are a set of automated actions that can be configured to manage the lifecycle of objects in S3 bucket.
- These rules help optimize storage costs and manage data efficiently over time by transitioning objects to different storage classes or expiring them when they are no longer needed.

[Note: can add multiple transitions so object will be moved to that storage once 30 days, 60 days, 90days, etc., are over so as to decrease price/billing]



(iv) Web Hosting:

- Amazon S3 can be used to host static websites, providing a simple and cost-effective solution for delivering content on the web. A static website hosted on S3 consists of HTML, CSS, JavaScript, and other static files like images or videos.
- It's best suited for sites where content doesn't need to change dynamically.

How it Works

- **Create an S3 Bucket:** Set up an S3 bucket with a unique name.
- **Configure Static Website Hosting:** Enable static website hosting for the bucket, specifying the index document (e.g., index.html) and error document (optional).
- **Upload Website Content:** Upload your website files (HTML, CSS, JavaScript, images) to the S3 bucket.
- **Access Your Website:** Use the generated S3 bucket endpoint to access your website.

(v) Global Service, Zone Specific Services, Region Specific Service

Feature	Global Services	Regional Services	Zone-Specific Services
Scope	Accessible from anywhere in the world.	Available within a specific geographic area.	Limited to a specific Availability Zone (AZ) within a region.
Latency	Generally higher due to geographic distance.	Lower latency for users within the region.	Lowest latency for highly localized access.
Fault Tolerance	High availability through redundancy across regions.	Fault tolerance through redundancy within a region.	High availability through redundancy within an AZ.
Data Residency	Data is stored regionally, but bucket names are globally unique.	Data is stored regionally.	Data is stored regionally and within a specific AZ.
Scalability	Highly scalable to meet global demands.	Scalable within a region.	Limited scalability within an AZ.
Use Cases	Content delivery, static data storage, backups, logs.	Compute resources, databases, virtual networks.	Critical components of highly available systems.
Examples	Amazon S3, IAM, Route 53, CloudFront	Amazon EC2, RDS, VPC, Lambda, EC2	Amazon EBS volumes, Network interfaces

[Note: Fault Tolerance is the ability of a system to continue operating correctly even when one or more of its components fail]

12) Lambda:

AWS Lambda is a serverless compute service that lets you run code without provisioning or managing servers. You pay only for the compute time you consume - there is no charge when your code is not running.
With Lambda, you can run code for virtually any type of application or backend service.

Key Concepts:

- **Serverless computing:** You focus on writing code; AWS manages the underlying infrastructure.
- **Functions:** Units of code that respond to events.
- **Triggers:** Events that initiate function execution (e.g., S3 object creation, API Gateway request, DynamoDB stream).
- **Runtime:** The programming language environment used for your function (Node.js, Python, Java, etc.).
- **Concurrency:** The number of instances of your function that can run concurrently.
- **Cold start:** The initial time taken to start a function instance.

13) Developers Tools:

(i) Codebuild:

AWS CodeBuild is a fully managed continuous integration service that compiles source code, runs tests, and produces ready-to-deploy software packages. It eliminates the need for provisioning, managing, and scaling your own build servers.

(ii) CodePipeline:

AWS CodePipeline is a fully managed continuous delivery service that helps you automate your release pipelines for fast and reliable application and infrastructure updates. It enables you to build, test, and deploy your code reliably and efficiently.

(iii) CodeCommit:

CodeCommit is a secure and highly scalable source control service that hosts private Git repositories. It enables you to store, manage, and collaborate on code within your organization.

(iv) CodeDeploy:

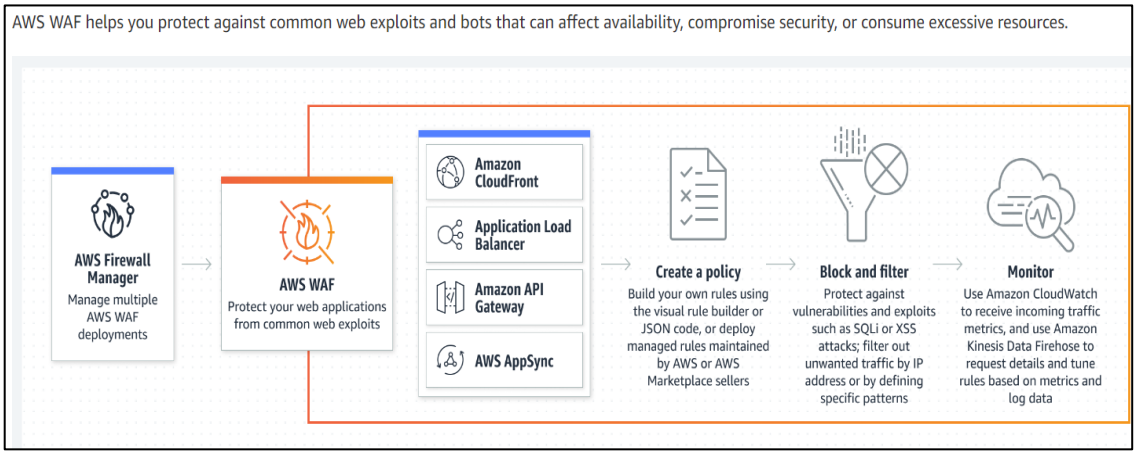
CodeDeploy is a fully managed deployment service that automates the deployment of applications to a variety of compute services, including Amazon EC2, AWS Lambda, etc

14) WAF: (Web Application Firewall)

A security application that filters and monitors incoming and outgoing HTTP traffic to protect web applications from attacks.

Key concepts:

- **IP Sets:** Allows to efficiently manage and group IP addresses or ranges for various security policies.
- **Management Interface:** Configures WAF settings, policies, and rules.
- **Rule Set:** Predefined rules to detect and block attacks.
- **Signature Library:** Database of known attack patterns.
- **Bot Management:** Detects and mitigates bot attacks.
- **Deployment Models:** Network-based, host-based, and cloud-based.



15) Cloudfront:

Amazon CloudFront is a content delivery network (CDN) service that securely delivers data, videos, applications, and APIs to customers with low latency, high transfer speeds, and increased availability.

CloudFront Distributions

A CloudFront distribution is the core configuration for delivering content. It defines how CloudFront will serve your content.

CloudFront Policies

- **Origin Access Identity (OAI):** Allows you to restrict access to your Amazon S3 bucket so that only CloudFront can access it.
- **Bucket Policy:** Controls access to your S3 bucket.
- **IAM Policies:** Manage user permissions for CloudFront resources

16) AWS CLI:

AWS CLI (Command Line Interface) is a unified tool that allows you to control multiple AWS services from the command line and automate them through scripts.

It provides a consistent interface for interacting with all parts of AWS.

17) AWS Route 53:

AWS Route 53 is a highly available and scalable Domain Name System (DNS) web service provided by Amazon Web Services.

It allows you to register domain names, route traffic to instances running in multiple AWS Regions, and monitor the health of your application.

18) RDS:

Amazon Relational Database Service (RDS) is a managed relational database service offered by AWS.

It makes it easy to set up, operate, and scale a relational database in the cloud.

19) Elastic Beanstalk

Amazon Elastic Beanstalk is platform-as-a-service (PaaS) offering from Amazon Web Services (AWS).

Easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx.

Why Elastic Beanstalk?

Elastic Beanstalk is a service for deploying and scaling web applications and services. Upload your code and Elastic Beanstalk automatically handles the deployment—from capacity provisioning, load balancing, and auto scaling to application health monitoring.

Pricing:

There's no additional charge for Elastic Beanstalk. You pay for Amazon Web Services resources that we create to store and run your web application, like Amazon S3 buckets and Amazon EC2 instances.

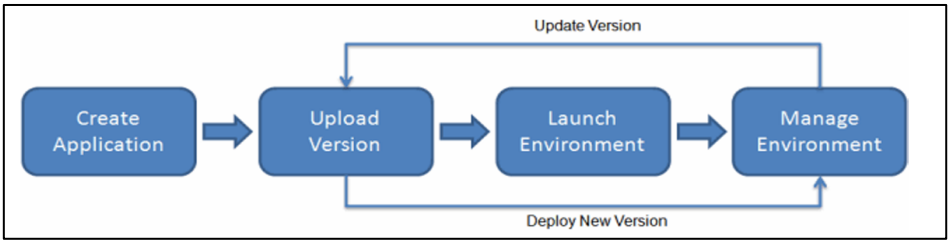
Key Benefits:

- **Simplified Deployment:** Handles infrastructure setup and management, allowing you to focus on your application code.
- **Auto-Scaling:** Automatically adjusts the number of instances based on traffic to optimize costs and performance.
- **Load Balancing:** Distributes incoming traffic across multiple instances for improved performance and reliability.
- **Platform Support:** Supports a wide range of programming languages and frameworks.
- **Cost-Effective:** Only pay for the resources you use.

How Does It Work?

To use Elastic Beanstalk, you create an application, upload an application version in the form of an application source bundle (for example, a Java .war file) to Elastic Beanstalk, and then provide some information about the application.

Elastic Beanstalk automatically launches an environment and creates and configures the AWS resources needed to run your code. After your environment is launched, you can then manage your environment and deploy new application versions.



20) SQS (Simple Queue Service):

SQS is a fully managed message queuing service provided by AWS. It enables to send, store, and receive messages between software components at any volume, without losing messages or requiring other services to be available.

How it works:

- Producers send messages to an SQS queue.
- Consumers retrieve these messages from the queue and process them.
- SQS ensures messages are delivered reliably and in order (for FIFO queues).

➔ *Difference between SNS and SQS*

Feature	SNS (Simple Notification Service)	SQS (Simple Queue Service)
Purpose	Sends messages to multiple subscribers	Stores messages in a queue for later processing
Message delivery	Pushes messages to subscribers	Messages are pulled by consumers
Message persistence	No, messages are not stored	Yes, messages are stored for a specific duration
Scaling	Automatically scales to handle many subscribers	Automatically scales to handle message volume
Use case	Notifications, alerts, fan-out	Asynchronous processing
Example	Email notifications, SMS alerts	Task queues, message buffering

21) ACM: (AWS Certificate Manager)

It's a service provided by Amazon Web Services (AWS) that simplifies the process of managing and deploying SSL/TLS certificates for your AWS-based websites and applications.

How ACM Works

(i) **Request a Certificate:** You can request a certificate for a specific domain, multiple domains, or a wildcard domain.

(ii) **Validation:** ACM verifies your ownership of the domain through DNS validation or email validation. Once validated, ACM issues the SSL/TLS certificate.

(iii) **Deployment:** You can deploy the certificate to various AWS services like Elastic Load Balancing, CloudFront, or API Gateway.

(iv) **Renewal:** ACM automatically handles certificate renewals before they expire.