३०. ५१

Amazon Web Services

→ 90 Minutes

→ 65 Question (MCQ's) => No Negative

50 (Scored) 15 (Unscored)

→ 70% = Total 1000 marks

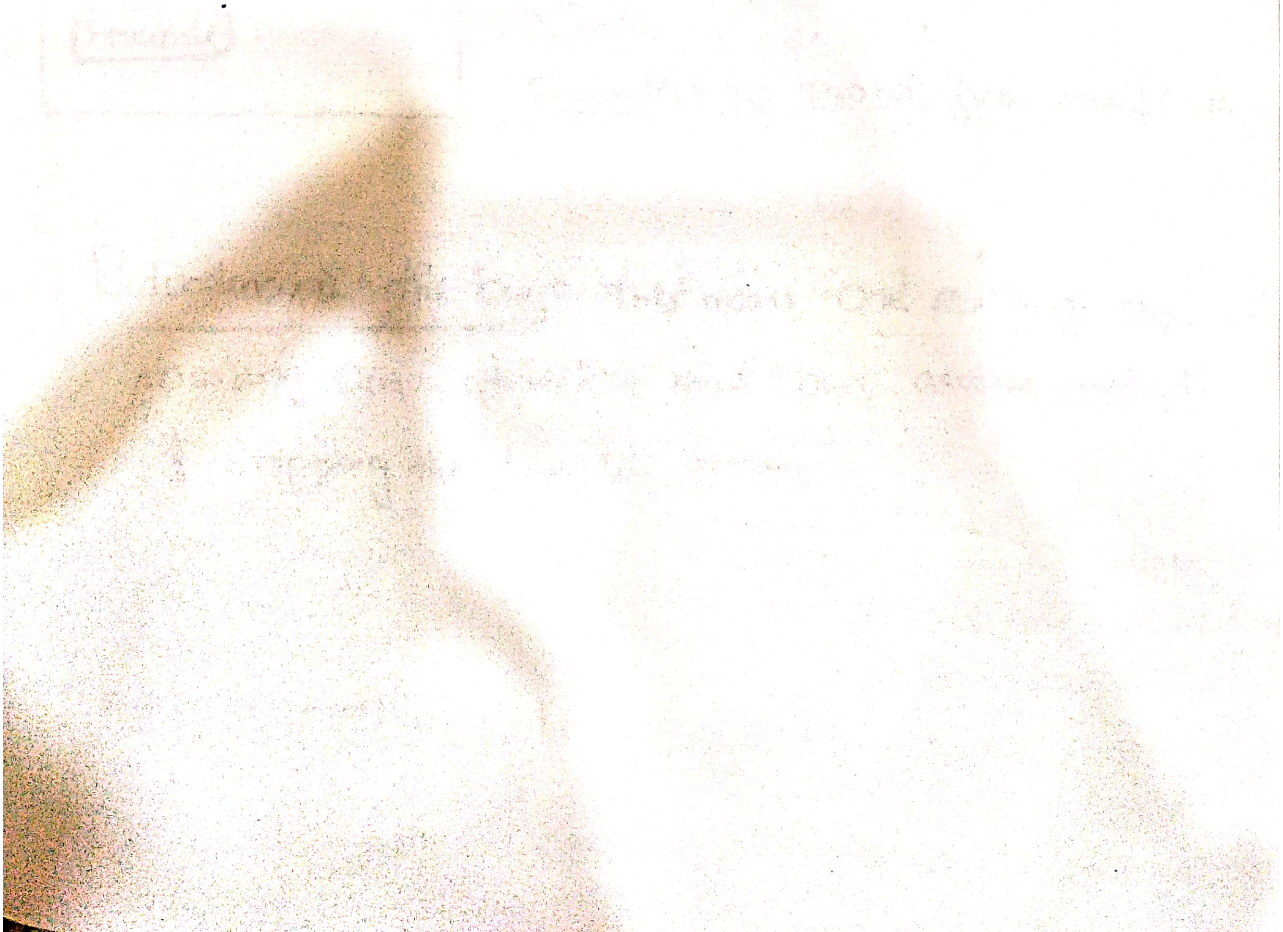
= 700 marks

Domain-1: Cloud Concept (24% Scored Content)

Domain-2: Security & Compliance (30% Scored Content)

Domain-3: Cloud Technology and Servic (34% Scored)

Domain-4: Billing, Pricing and Support (12% Scored)



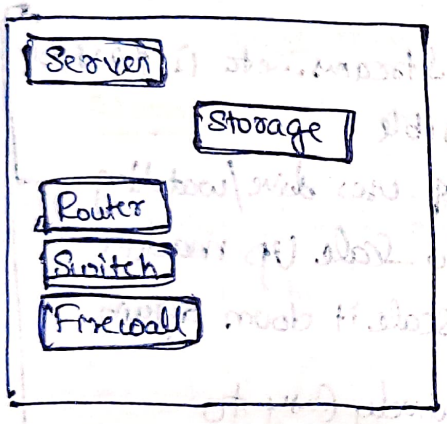
Cloud Computing

Traditional Infra
vs
Cloud

→ Objective is to move to cloud from Tradinal Infra (or) Have Servers racks etc in house ← On premises

=) Tradinal Infra - A person use to maintain all the servers and all.

* By which it costed alot
* Was tough to handle & manage.
* There must Run Day & Night
* You Own Everything (or) Rent the things.



Server: It is a Software running in a Powerful Hardware

Storage: These are SSD's & HDD's ... etc

Router: TO Connect to the Internet

Switches

Firewalls: TO Protect from viruses & Hacks.

Redundency is the key: This means One of Things stops working Our Software/Site must Supt another instead of Stopping the Complete Server.

Cloud Infra Structure: Cloud Infra Structure Refers to the collection of Hardware & Software services that power cloud Computing Services. In Includes Components like Servers, Storage, networking, Virtualization and management tools, all hosted in a cloud Provider's data Center.

* On Demand
* Just with a press of Button you move from 2GB to 200GB then Back to 2GB. (Impossible in cloud)
* Pay As You GO Model : You Pay As you Use/How much you Use. [Asked Again & Again]
* Shared Resources. [Majority Usage] (or) Advantage [Also we have some Govt Servers which are kept aside & Separate from public Use]
* It is For Any One. [You just Create your account, Pay money and deploy your Server]
* Elasticity [Because of this only Live Streams...etc is possible or Else this was Not Possible
  ⇒ When there are millions of users live/watching
  - It is Easy to Scale Up machine But very Difficult to Scale it down again
  - In Aws it is Comparatively Easy to Scale down your machine.
  - It is One of the Most Important feature of AWS [Asked again & again]

Key Components

1) Compute: Virtual Machines (VM's), Containers, or Serverless Computing for processing workloads.
Example: AWS EC2.

2) Storage: Cloud-based Storage solutions for storing data, files and Backups
Example: AWS S3 (Simple Storage Service)

Imagine building Dropbox from scratch

Traditional Infra	Cloud
Buy Hardware	login into AWS
Installation	Pick your service
Expansion	Scale on click
Operation → Software	Software.

(Those people Come we can do load balancing)

(3) Networking: Virtual networks, load balancers, and firewalls that manage communication between resource.
Example: AWS VPC

(4) Virtualization: Software that abstract physical resource to create virtual machine or Containers
Example: VMware.

(5) Management & Security: Tools for monitoring, logging, authentication, and Security
Example: AWS IAM

(long form network sent us)

DropBox [Cloud Storage Service] :- DropBox is a Popular cloud storage Service that allows users to store, sync, and share files online. It provides features like file Versioning, Collaboration, and automatic Backup.

Cloud Infra Structure

Components

=) cloud infrastructure is made up of various components that work together to provide Computing, Storage, networking and Security Service. Below is a Breakdown of these components and their role in cloud Computing.

1) Compute: (processor)

Definition: Compute Services provide the processing power needed to run application, work loads, and Virtual environments in the cloud.

Types of Compute Services:

- Virtual Machines (VM's):- Fully Managed instances of Virtualized Servers that Run Applications.
- Containers:- Light weight, Portable environment that include everything needed to Run an Application [Code, runtime, dependencies]
- Serverless Computing: cloud Providers automatically manage resource, scaling and Execute without the need for direct Server management.

Example:

AWS EC2 [Elastic Compute Cloud]: Scalable Virtual Machine in AWS.

2) Storage:-

Defination:- Cloud Storage provides scalable, durable, and Cost-effecient solutions for storing data, backup's and files.

Types of cloud Storage:

- Object Storage:- Stores data as Object with metadata (Eg: AWS S3)
- Block Storage: Provides high-Performance storage database and VM's [Eg: AWS EBS]
- File Storage: Shared files system accessible across multiple instance [Eg: AWS EFS]

Example:

→ AWS S3 (Simple Storage Service): Scalable Object storage

3) Networking

Definition: Cloud networking enables secure communication between different cloud resources and the Internet

Key Components of Cloud Networking

- Virtual Private Cloud (VPC):- Isolated cloud network that allows users to define IP range, subnets, and access control.
- Load Balancers:- Distributing incoming traffic across multiple servers to improve performance and Reliability.
- Firewalls:- Security systems that filters and monitor network traffic.

Examples:

- AWS VPC:- Creates an isolated virtual network within AWS.

4) Virtualization:

Defination: Virtualization enables multiple virtual environments to run on a single physical machine, improving resource efficiently.

Types of Virtualization:

- VMware :- A leading virtualization for running VM's
- Kubernetes: An Open source system for managing Containerd application.
- Docker: A platform for developing, shipping and Running Containers.

Examples:

- VMware: Provides VM-based Virtualization.
- Kubernetes: Orchestrates Containerized applications.
- Docker: Creates and runs Containers for application deployment.

5) Management & Security:

Definition:

Definition: Management and Security services help monitor, authenticate, and protect cloud resources

Key Features:

- Identity & Access Management (IAM): Controls Users Permission and authentication.
- Logging & Monitoring: Tracks systems performance and detect issues.
- Security Compliance: Ensures cloud environment meet Regulatory system standards.

Examples:

- AWS-IAM
  (Identity & Access Management)

  → Manages Users Permission and access to AWS Services.

Types of Cloud Service

⇒ Cloud Infrastructure can be deployed in different ways based on Business needs, Security Concerns, and Optional flexibility.

Types of cloud Infrastructure

1) Public cloud
2) Private cloud
3) Hybrid cloud
4) Multi cloud

1) Public Cloud: A public cloud is a cloud Computing model where Infrastructure and Service are owned and managed By third-Party cloud providers.

* The Resources [Computing, Storage, networking...etc] are shared among multiple users [multitenant model] over the Internet.

Key Features:

, Cost-efficient, Scalable, Maintenance-free.

Example: (AWS)

UseCase:- Startup's ~~Bbbb~~ Business needs, Scalable Infrastructure, Hosting SaaS applications, Companies with fluctuating workloads [E-Commerce]

2) Private cloud: A Private cloud is a cloud computing Environment dedicated exclusively to a single organization. It can be hosted On-Premises [within a Company's own Data Center] or Provided by a third-Party cloud.

Key feature: High Security, Customization, Compliance.

Use Case: Financial Institutions, Government agencies, etc.

3) Hybrid Cloud: A Hybrid cloud combines both Public and Private cloud environment, allowing data and applications to move between them. This approach provides flexible, Security and Cost-Optimization.

Example: AWS cloud Outpost

UseCase: → Business Handling Sensitive data [Health Care]

→ Enterprise wanting disaster recovery with On prim es Backup and public cloud failover.

4) Multi Cloud: A multi-cloud Strategy involves Using multiple cloud Service providers (AWS, Google, Azure etc) to Avoid reliance on a Single provider. This Approach enhances resilience, flexibility, and Cost Saving

UseCase: → Companies wanting to Optimize workloads across different cloud providers

→ Business needing High Availability and disaster Recovery.

→ Organizations looking to negotiate better prices By Using multiple providers.

Cloud Service Models

→ IaaS [Infrastructure as a Service]
→ PaaS [Platform as a Service]
→ SaaS [Software as a Service]
→ Control Over Cloud

* Cloud Computing is Category into three main Service models.
→ Each model Provides a different level of Control, flexibility and Responsibility for Users.

1) Infrastructure as a Service [IaaS]:

(SW)

* IaaS provides virtualized Computing resources over the Internet, such as virtual machines, storage, and networking.
* It allows businesses to Rent IT infrastructure instead of maintaining physical hardware.

→ Full Control over the OS, applications, Storage & networks.
→ Cloud Provider manages the physical infrastructure.

Eg: AWS EC2 UseCase: Hosting website.

2) Platform as a Service [PaaS]:

* PaaS provides a fully managed platform, including hardware, Operating System, and development tools, so developers can Build, test, and deploy applications without worrying about infrastructure

→ Control over the applications and data
→ Cloud Provider manages servers, networking, storage and Runtime Environment. UseCase: Web & app dev

Eg: Aws Elastic Beanstalk.

3) Software as a Service (SaaS):

* SaaS delivary fully functional software applications Over the Internet on a subscription basis. Users do not need to install, maintain, or update the software everything is managed by the Provider.

→ Users Control data and Configure [if allowed].
→ Cloud Provider manages the software, hosting, infrastructure and Security.

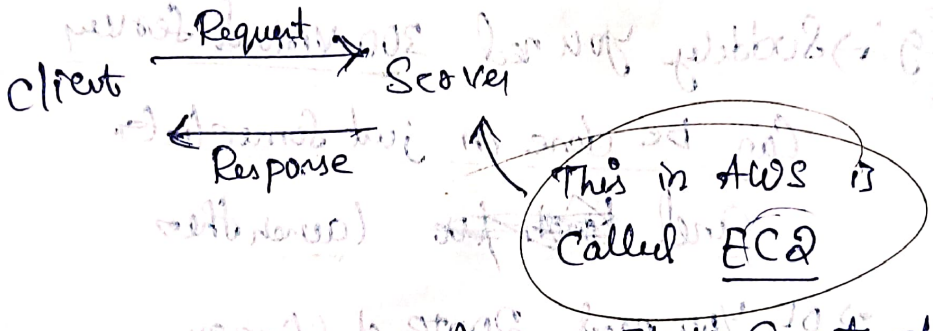
Skill-Builder

AWS Service offering

→ AWS offer.

* Compute
* Storage
* Network security
* Blockchain
* Machine learning
* Artificial Intelligence
* Robot development
* Video production
* Orbital Satellite.

=) It is a client-server model



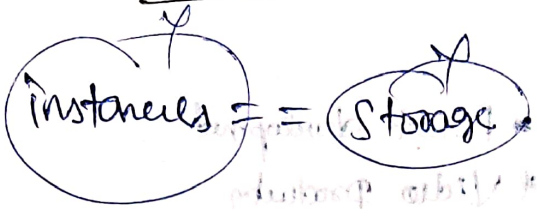
Amazon Elastic Compute Cloud
(Amazon EC2)

an EC2 instance, a virtual server.

→ This also validate and Gives Response to the client.

Key Concept For AWS

* You Only Pay For What You Use.
* Pay For What You Need.



Cloud Computing

→ On-Demand delivery.

↳ AWS has Resource when you need them. You Don't need to tell in advance that You need them.

Eg :→ Suddenly you need 300 virtual servers
Can be done in just some clicks
and ~~Each for~~ Launch the

→ If You need 200TB of storage.
You don't to tell in advance
Just Start Using them, when you them,
Don't need them anymore, just as Quickly.
You Can Return them and
Stop Paying.

Cloud: The On Demand delivery of IT Resource Over the Internet with Pay-as-go pricing

Undifferentiated heavy lifting of IT

↳ AWS will handle what you need, you Just focus on doing something need which Provide you unique/direct in IT.

Upfront Expense: You pay Before for data Centre, physical Server

Variable Expense: you only pay for computing servers you Consume instead of investing heavily in data Centre and Server b/f you know how you are going to utilize.

→ Trade upfront for Variable Expense

- Stop speculating to run & maintain data Centre.
- Stop guessing Capacity.

AWS Lambda is a AWS service that lets you run code without needing to manage or provision Servers. This description does not describe Cloud Computing as a whole.

AWS lambda is Explained.

Amazon Elastic Compute Cloud

→ You need Servers to Power Your Business

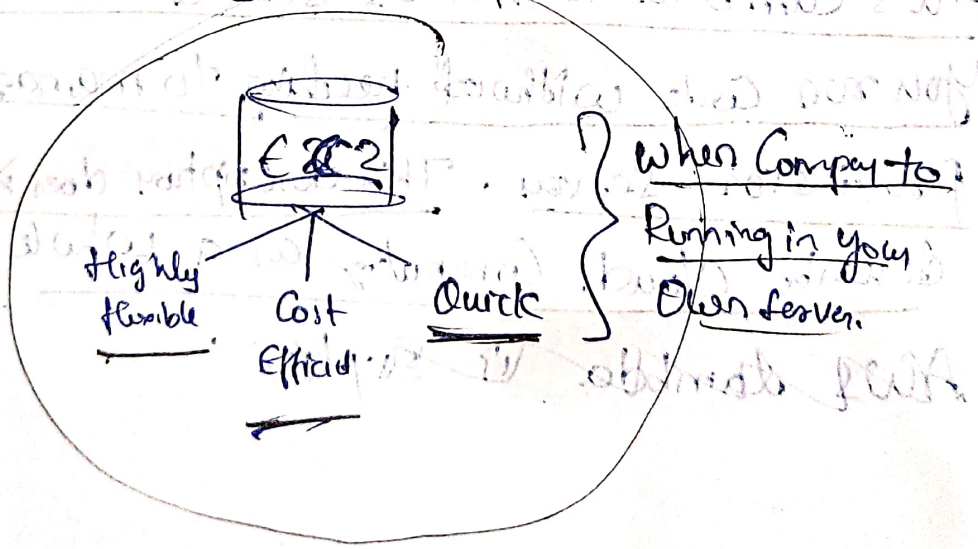
⇒ Eg it might be healthcare, manufacturer, Insurance or Delivering Video Content to millions of users. You need Servers for your Applicat

(or) you need to send data, products, services to end user. You need Servers.

In AWS (Server are Virtual)

* The Servers that You Use to Gain Access to Virtual Servers is Called (EC2)

⇒ EC2 ⇒ Elastic Compute Cloud.



M On-Prime Servey

1) You need to Rent Hardwy
2) Buy & wait for them.
3) Install them
4) Then Scale your Applicatn.

But In AWS.

* In EC2 it is much more Easier to Get Starty
* AWS took Care of it Already.

AWS EC2

→ AWS Build dataCenters ✓
→ AWS Secured data Centers ✓
→ AWS purchased Servers ✓
→ AWS installed Servers. ✓

=) The Seavers are Online and Ready to Use.

=) ~~AWS has high Capacity Servers or~~ massive

=) AWS has massive amount of Compute Capacity And you Can use what ever portion of that Capacity whom you need them.

=) All you need is to Requst the EC2 inst. you want and they will Launch and boot up. Ready to be Used in minutes

=) Once your are done you Can Stop or Terminat the EC2 Ser Instance.

=) Your Usage of EC2 instance can vary greatly over time. and you pay for what you use.

=) In EC2 you only pay for what you use. Because with EC2, you only pay for Running instances. Not stopped or Terminated Instances

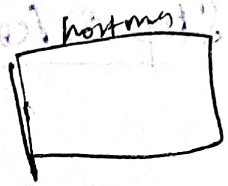
=) EC2 Runs on top of Physical host machine managed By AWS using Virtualization technology.

=) When you spin up an EC2 instance, you aren't necessarily taking entire host your self. Instead, you are sharing the host with multiple other Instances. Otherwise known as virtual machines.

=) A Hyper Visor running on the host machine is Responsible for sharing the underlaying physical resources between the Virtual machines

=) This idea of sharing the underlaying hardware is called Multitenancy.

=) The Hyper Visor is Responsible for isolating the Virtual machines from each other as they share Resource from the host.



=) These means EC2 instance are secure. Even though they may be sharing resources, one EC2 instance is not aware of any other EC2 instance also on the host. They are secure and separate from each other.

~~Host~~

* Think of a host as a powerful physical Computer (like a super-strong laptop or desktop) owned and managed by AWS. This Computer has a lot of memory, Processing Power (CPU) and storage.

* When you start an EC2 instance, you are basically renting a portion of this powerful Computer. But you are not the Only One - AWS Allows multiple users to Rent portions of the same physical machine.

* AWS uses something called a **hypervisor** a software that Divides the physical Computer into **smaller virtual machines (VMs)** Each VM behaves like a separate Computer, even though it is actually Running on a shared machine.

* Multiple EC2 instance (from diff VMs) run on same **physical machine (Host)** this is called **multi tenancy**.

* The hypervisor makes sure the each VM is Completely Isolated from other. Even though they share same physical medium, your EC2 instance Cannot See or Interact with other instances on the same host.

* When you create (provision) an EC2 instance, you get to pick which Operating System (OS) it will use either windows or linux.

* You're not limited to just One EC2 instance. You can create thousands of them at anytime, each with different Configuration to Run different parts of your Business.

* Once your EC2 instance is up and Running, you can install whatever software you need.

  → web application
  → Databases
  → Internal Business apps (tools used inside a company)
  → Third party software.

  Basically you have Full Control Over what software runs on the instance.

* If your application starts using too much memory (RAM) or CPU (processing power), you don't need to start over with a new server. Instead, you can increase the memory and CPU for the same EC2 instance.

* This process is called Vertical Scaling. It means upgrading the power of an instance instead of adding more instances.

* If your Application needs more power, You can make the Instance Bigger (more CPU, RAM, Storage). If you need less Power, You can make it smaller. This flexibility helps you save money and adjust Based on your need.

* You Can Even Control Network Access.

→ You Decide who can access your EC2 instance.

* Public access: Anyone on the Internet Can reach it

* Private access: Only Specific people or Services Can Connect

=) AWS makes Virtual Machines Easy:

* Virtual Machines (VMs) have been around for a long time, but AWS makes it super simple to create and manage them. Instead of Buying Expensive hardware, You can Rent Computing Power as need (Compute As a Service) (CaaS)

Vertical Scaling → Using too much Ram
make instance Bigger → need more power

Vertical Scaling means Ram & storage

* Virtual Machines are software-based computers that runs on a physical computer. They're not a new concept - They've been around for years, However AWS simplifies the whole process of creating, managing and using VM's, which we call EC2 instance.

Amazon EC2 Instance Types

=> Amazon EC2 provides a wide selection of instance types optimized to fit different use cases.

=> Instance type comprise varying combination of CPU, memory, storage, and networking capacity and give you flexibility to choose the appropriate mix of resources for your application.

=> Each instance type include one or more instance sizes, allowing you to scale your resources to the requirements of your target workloads.

AWS OFFERS different types of EC2 instances for different jobs/tasks. Each EC2 instance has specific strengths, making it suited for different types of work.

EC2 Instance families:

These are groups of instance optimized for different tasks. Think of them as different categories of employees in the coffee shop, where each employee is trained for a specific task.

1) General Purpose => Equal
2) Compute Optimized => Computing ↑, Processing or processor.
3) Memory Optimized => RAM ↑
4) Accelerated Computing => GPU ↑
5) Storage Optimized => Storage ↑

1) General Purpose Instance:

*There are well-rounded Instance that offer a balance of CPU, memory, and networking power.

→ They can handle various tasks, such as Running a simple website, managing a Code Repository, or handling general-purpose workloads.

2) Compute Optimized Instance:

*They are Build for Compute-Heavy tasks that require a lot of processing power.

→ They're Great for task like gaming servers, high-performance Computing (HPC), Scientific modeling, and simulations that need a lot of CPU Processing.

3) Memory Optimized Instance:

*These Instance are designed for tasks that need a lot of memory (RAM).

→ Ideal for Running application like large database, data analytics, and in-memory caches where you need to store a lot of data in memory.

4) Accelerated Computing Instance:

* These Instance have Specific hardware accelate for tasks that need intense mathematical Calculations, graphicas processing or data analysis.

→ They're great for things like machine learning, AI modelling, graphics rendering, or Simulation that requires heavy-duty floating point Calculations.

5) Storage Optimized Instance:

* These are designed for workloads that requires high-performance storage.

→ Ideal for tasks like running large data warehouse or managing large-scale database where fast data storage and retrieval are important.

Data warehouse Applicate ⇒ Storage Optimized

Balace Compute, memory, networking ⇒ General purpose

Ideal For high performal. Database ⇒ memory optimized

high performal processors ⇒ Compute optimized

Amazon EC2 Pricing

AWS EC2 offers multiple billing options to suit different needs, ranging from flexible pay-as-you-go models to long-term cost-saving plans.

1) On Demand Instances:

* On Demand pricing allows you to pay for the Compute Capacity you use, billed per hour or per second [Depend on the Instance type and OS].
* No long-term commitments or upfront payments are Required.

Use Case: Testing & Development ✓
Application with unpredictable traffic ✓
first-time AWS user exploring EC2 ✓

2) Savings plan: [EC2 instance saving Plans]

* The Saving plan offer low price in Exchange for a Commitment to a Specific usage level [measured in dollar per hour] for a One-year or three-year term.

→ Upto 72% Savings Compared to On Demand pricing.
→ Applies across any instance family, size, OS or AWS region.
→ Also Applies to AWS Fargate & AWS Lambda [Serverless Compute]

Use Case: Business with predictable Compute usage, long-term Applications seeking Cost optimization.

3) Reserved Instances [RI]: ~~Convertible Reserved Instance~~ Standard Reserved Instance

Reserved Instance provide significant discounts for EC2 usage when you commit to a One-year or three year term.

Key features: → upto 75% discount

→ Three payment options

* All upfront: pay full when you commit
* Partial Upfront: pay a portion upfront, with lower monthly cost
* No Upfront: Pay Nothing initially, but still get a discount

Use Case: * Steady State applications with predictable workload
* Databases, enterprise applications, and Backend systems.

4) Spot Instances: Let you use Spare EC2 Capacity at upto 90% off On-Demand Price. However, AWS can reclaim the Instance at anytime with a two-minute warning

Key features:
* Extremely low Cost
* Idel for Interruptible workload
* Can be Combined with On-Demand and Reserved Instance for Cost efficiency.

Use Case
* Batch Processing, Bigdata, CI/CD
* ML model training
* fault tolerant applications that can Resume later.

(5) Dedicated Hosts:

→ Dedicated Hosts are physical servers fully dedicated to One Customer, Ensuring no shared tenancy.

Key features:

* Required for Specific Compliance and regulatory needs.
* Provides greater Control over hardware.
* Reduces licensing Costs for Certain Software

Use Cases:

→ Companies needing strict Security and Compliance (Health Care, finance, Govt)

→ Organization running legacy Software required dedicated hardware

---

=) Specify Number of EC2 instance to run a Specific OS, instance family and Size and tenancy in One Region (=) Standard Reserved instance

You Specify amount of EC2 instance are Covered over a 1-year or 3-year term.

→ Amazon EC2 pricing option provides a discount when you make an hourly Spend Commitment to an instance family within a region for a 1-year or 3-year term.

Ans:- EC2 instance Savings plans.

EC2 - Pricing

Scaling Amazon EC2

Scalibility & Elasticity

=) When Using On-premises (physical) Servers. If they Buy too few, their system can't handle heavy traffic at peak times when they actually make money.

=) If they By Buy too many, they waste many when traffic is low.

=) How AWS Solve This issue

→ With AWS EC2, you don't have to buy physical servers, Instead, you Can Rent Virtual Machines (instances) and Only Pay for what you use.

Scaling Up (more Demand) → AWS adds more Servers automatically.

Scaling Down (low Demand) → AWS Removes Extra Servers to Save money.

→ If the Server fails

* AWS knows that failures happen. Instead of waiting for a fix, AWS automatically replaces the failed Instance so your Customer (users) never Experience downtime.

Scalability = Growing the System when more Customers arrive.

Elasticity = Automatically adjusting System based on Demand.

=) To Scale faster, you can use dynamic scaling and predictive scaling together.

=) If you want the Scaling process to happen automatically, which AWS service would you use?

(A) The AWS service that provides this functionality for Amazon EC2 instance is Amazon EC2 Auto Scaling.

More Demand => AWS adds more Servers automatically

Less Demand => AWS removes Extra Servers to save Cost

A Server fails => AWS replaces it instantly so the Web site stays up.

Within Amazon EC2 Auto Scaling, you can use two approaches: dynamic scaling & predictive scaling

(i) Dynamic Scaling: Reacts to real-time demand.

- If website traffic increases, new servers are added automatically
- If traffic decreases, extra servers are removed.
- This happens on the fly, as needed.

(ii) Predictive Scaling: plans ahead using AI

- AWS analyzes past traffic patterns and predicts when more servers will be needed.
- It prepares extra servers in advance before traffic surge happens.

> EC2 - Auto Scaling - Scaling Smarts in the Cloud

→ Minimum Capacity: The least number of servers always running

→ Desired Capacity: The number of servers you usually want

→ Maximum Capacity: The highest number of servers allowed during peak times.

How it works

1) At launch: you setup Auto scaling group with minimum of 1 instance to ensure your app is always running.

2) Normal Operation: you Instance keeps 2 instance Running most of the time (desired Capacity)

3) Traffic Surge: If demand rises.. Aws adds more instance - but never more that maximum instance you set

4) Traffic Drops: when Demand decrease, Aws Removes unneeded instances to save money.

5) Cost Saving: You Only Pay for active Instance, so No waste Spending on unused Server.

Elastic Scalability, Inherent power of Server

Directing traffic with Elastic Load Balancing

ELASTIC LOAD BALANCING [ELB]

What is Load Balancer?

A Load Balancer is like a Restaurant manager who directs customers to the least busy cashier. It takes incoming traffic request and routes them to EC2 instance that are ready to process them.

→ Spreads traffic evenly - preventing overloading any single EC2 instance

→ Improves performance - Ensure request get fast responses.

→ Handles failure - If an Instance fails, the Load Balancer redirects traffic to healthy ones.

Amazon ELB – Amazon Elastic Load Balancing

→ AWS Elastic Load Balancing is a fully managed service that automatically distributes traffic across EC2 instances.

Highly Available – ELB operates at the regional level, ensuring reliability

Automatically Scales – Handles traffic spikes without manual intervention.

Works with Auto Scaling – When new EC2 instances launches them, ELB automatically include them.

Graceful Shutdown – ELB stops sending traffic to instance before they shutdown, preventing disruptions

How ELB Works in Front End & Backend

FrontEnd Traffic

* ELB sits b/w users and your EC2 instances.
* Users Connect to One URL [ELBAddress]
* ELB distributes incoming request to the least Busy Instances

Backend Traffic

- In Multi-tier app, ELB manages Internal Communication.
- Backend Instances don't need to know about Each other – they Just Connect to ELB.
- When new Backend instance comes Online, it registers with ELB and starts handling requests.

→ This decoupling Architecture makes scaling Automatic, efficient and stress free

Messaging and Queing

Monolithic Applications and Micro Services

- Applications are made of multiple Components. The Components Communicate with each other to transmit data, fulfil request, and keep the application running.
- Suppose that you have an application with tightly Coupled Components. These Components might include databases, Servers, the user interface, business logic, and so on. This type of architecture can be Considered as monolithic application.
- In this approch to application architecture if a single Component fails, other Components fail and possibly the Entire application fails.

=) To help maintain application availability when a Single Component fails, you can design your application through micro service approch.

Random Comp Coupling.

→ The Communicate Using a Buffer in Between them. All the messages pass through the Buffer. If Even the One Component fails to work, Another still Communicate and his message stays in Buffer. One Another starts work & all the Request are full filled & work Normally.

Micro Services Approach

* In a microservices approach, application components are loosely coupled, so one fails, the other components continue to work because they are communicating with each other. The loose coupling prevents the entire application from failing.

→ As applications are loosely coupled they communicate using a Buffer in between.

* When designing applications on AWS, you can take a micro service approach with services and components that fulfill different functions.

Two Amazon services facilitate application integration.

1) Amazon Simple Notification Service [Amazon SNS].
2) Amazon Simple Queue Service [Amazon SQS].

1) Amazon Simple Notification System (Amazon SNS)

* Amazon SNS is a publish/subscribe service. Using Amazon SNS topics, a publisher publishes message to subscribers. ~~This is similar to the Coffee shop. The Cashier provides coffee orders.~~

→ In Amazon SNS, subscriber can be web servers, email Address, AWS lambda functions, or several other options.

* Subscribers will now receive updates immediately for only the specific topic which they have subscribed.

* It is possible for subscribers to subscribe to single topic or to multiple topics.

- Thy SNS is a Pub/sub model.

Pub/Sub. model = One message can fan out to many subscribers.

Supports Multiple End points: SQS, Lambda, HTTP, SMS, email, ...

How SNS work's

→ Create SNS topic [message channel]

→ Add Subscribers [Users, application, ..etc]

→ Publish One message, and all subscribers get it instantly.

Amazon Simple Queue Service [Amazon SQS]

→ Amazon Simple Queue Service [Amazon SQS] is a message queuing service.

* Using Amazon SQS, you can send, store, and receive message between software components, without losing message or requiring other services to be available.

* In Amazon SQS, an application sends message into a queue. A user or service retrieves a message from the Queue, process it, and then delete it from the Queue.

Decouples Services – Components don't need to be online at same time.

Stores message reliably: No message is lost, even if the receiver is down [Stored in Queue]

Scales Automatically: Handles any message volume, from a few to millions.

Ensures message delivery: Supports retries and dead-letter Queue for failed message.

⇒ Application A sends a message to an SQS queue

⇒ The message waits in the queue until a Consumer is ready

⇒ Application B retrieve the message, process it, and then deletes it.

AWS Lambda

=) In Amazon EC2 for your Application.

(1) Provision Instances [Virtual Servers]

(2) Upload your Code

(3) Continue to manage the Instance while your application is Running.

* The term "serverless" means that your Code runs on Server, but you don't need to manage these Servers. AWS will manage all the stuff.

* With Serverless Computing, you can focus more on innovating new products and features instead of maintaining Servers.

* Another benifit of Serverless Computing is that flexibility to Scale serverless application Automatically. It can adjust the application Capacity by modifying the Units of Consumption such as throughput and memory.

→ An AWS Serverless Computing is AWS Lambda

=) AWS lambda is a service that lets you run code without needing to provision or manage servers.

=) You only pay for compute time that you consume. Charges apply only when your code is running.

=) You can also run code for virtually any type of application or backend service, all with zero administration.

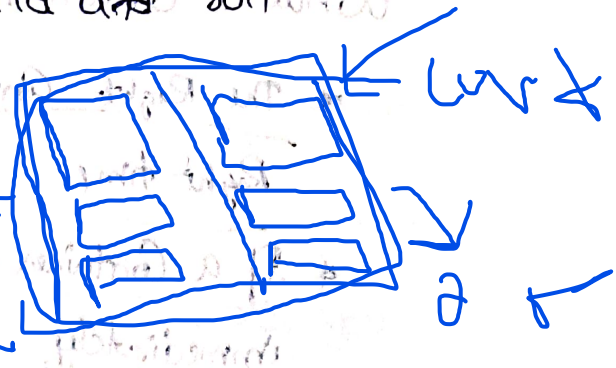
How It Works

1) Upload code to lambda
2) Set code to trigger from an event source (set triggers to run)
3) Code runs only when triggered.
4) You only pay for compute time you use.

Example: A simple lambda function might involve automatically resizing uploaded images to the AWS cloud. In this case, the function triggers when uploading a new image.

* You only pay for the compute time that you use. You would only for the compute time that you use when uploading new images. Uploading the image triggers lambda to run code for the image resizing function.

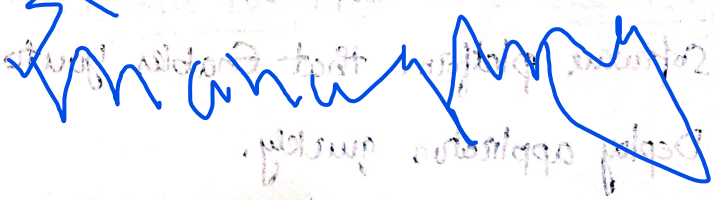
In AWS, You Can Also build and run Containerized applications.

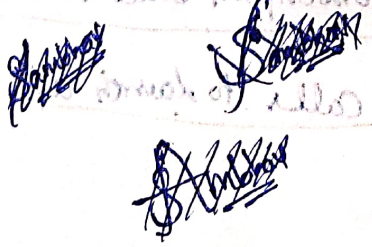


⇒ Containers:

* Containers provide you with standard way to Package your application's code and dependencies into a single object. You Can Only Containers for processes and workflows In which there are essential requirements for Security, reliability, and scalability.

→ When you run only a few Containers, You can manage them manually. But when you need hundreds or thousands of Containers running across many Computers, it Becomes impossible to manage them manually. You need a System to Organize and Control Everything efficiently - this is where Containers Orchestration Comes in.





$Ambhav

What is Container Orchestration

* The Right Containers are working at the Right time
* If a Container crashes, another Container must take over immediately.
* Resources like memory and CPU are used Efficiently.
* These can handle Rush of used Sudden [Scalun up when needs]

Amazon Elastic Container Service [Amazon ECS]

* Amazon ECS is a highly scalable, high-performance Container management system that enables you to run and scale Containered applicants on AWS.

Cont

* Amazon ECS Supports Docker Containers. Docker is a Software platform that Enables you to build test, and Deploy application quickly.
* AWS supports the use of open-source & Subscription based Docker.
* With Amazon ECS, You Can Use API Calls to launch and Stop Docker-enables Applications.

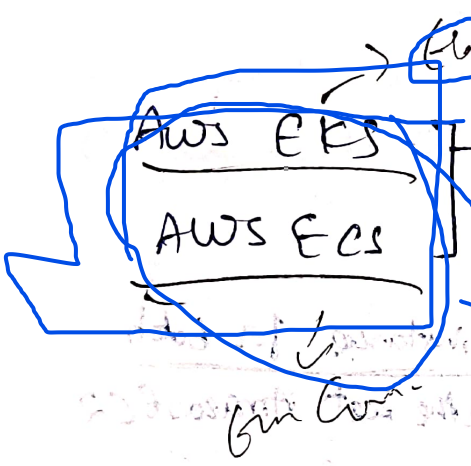
Docker Kubernetes

Amazon Elastic Kubernetes Service [EKS]

* Amazon EKS is a fully managed service that you can use to Run Kubernetes on AWS.

* Kubernetes is open source software that enables you to deploy and manage Containerized applications at scale.

* A large Community of Volunteers maintains Kubernetes, and AWS actively works together with the Kubernetes Community. As new features and functionalities release for Kubernetes applications, You can Easily apply these updates to your application managed by Amazon EKS.



There can be Controlled in 2 ways

* If you need full Control on you APP you need to manage scaling, memory etc Then you can use ECS

* If you Don't want to manage and automate Everything then you AWS Fargate

AWS Fargate

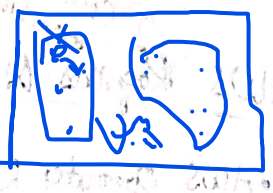
* AWS Fargate is a serverless Compute engine for Containers. It works with both Amazon ECS and Amazon EKS

* When Using AWS Fargate, you don't need to provision or manage servers.

* AWS Faogate manages your servers Infrastucture for you. You can focus more on Innovating and developing your applications, and you pay only for the Resource that are required to Run your Containers.

=) You want to Use an Amazon EC2 instance for batch Processing workload. What would be the best Amazon EC2 instance type to Use => Compute Optimized

AWS Global Infrastructure & Reliability



TO avoid disasters, big Companies like Amazon Web Services don't Relay on just One data Centers [warehouse for digital information]. Instead, they spread their services across different locations world wide.

* These locations are called Regions.
* AWS make sure you Applications says Online no matter what happens

(1) Multiple Data Centers [Not just one or two]

- Aws doesn't rely on single data Centers because if it fails Everything would stop working.

(2) AWS uses Regions

- Regions are different areas across the world wide. Where AWS has multiple data Centers
- If something bad happens in One Region, the other Region keep Running, Ensuring your website or application never goes down

(3) High Availability & Fault tolerance

- High Availability means your Applications stay Online Even if some part fail.
- Fault tolerance means AWS has Backup System ready to take over if something breaks.

AWS Global Infrastructure

multiple data centers cluster Availability Zone

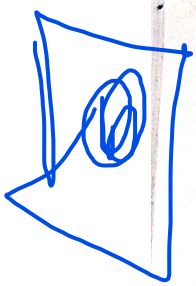
AWS Regions & Data Centers

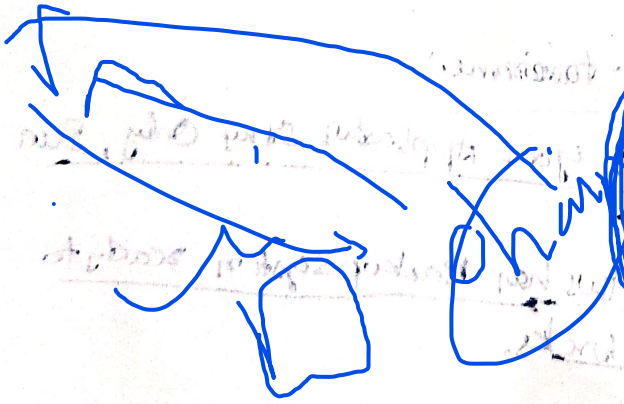
- AWS Operates Regions around the world to provide services like Computing, storage, database
- A Region is essentially a large area, like a city, and it contains multiple data centers [Building with servers]
- Each Region is separate from the other, meaning no data will leave the Region unless you allow it [great for privacy & security]

Four key Factors to choose a Region (R)

(1) Compliance [Legal Requirement] :-

- Some Business need to store data within a Specific Country due to legal Reasons
- If you need to keep your data in a Specific place like UK or China. you must choose the Region in that Country.
- Eg: if you required to keep data in Germany, you would choose the Frank Furt Region





- Compliance (Legal Rea) → only stay with Region
- Proximity (Close to Customer)
- Feature Availability
- Pricing

(2) Proximity [How close You Are to Your Customers]

- The closer you are to your Customers, the faster your app or website will be [Because of Speed of light limit]
- If most of your Customer are in Singapore, you should choose the Singapore Region, so the data travels a Shorter Distance.

(3) Feature Availability [Access to Services]:

~~The closer you are to your~~

- AWS is Constantly Adding new services. However, new feature may only be available in certain Region first, because they need physical infrastructure to work
- If you need a service [like Amazon Braket for quantum Computing Its new so its take some time to come to all regions], but its only Available in Specific Region, that will Influence your Region choice

(4) Pricing [Cost of Running Your Service]

- Some Regions are more Expensive than other because of the things like local taxes, electricity Cost.. etc.
- If Budget is a big Concern, you might choose a less Expensive Region

When Choosing Region you need to check

1) Do you need to comply with local law
2) How close is the Region with your Customer.
3) Does the Region have all the Services you need.
4) What's the Cost of using the Region.

What's the Problem with Using Only One DataCenter.

→ Imagine you have your Entire Application in a Single Building (dataCenter). what if Something happens (Natural Disaster..etc) the Building got damage or Loses power? Your Application would stop working. & you Business will be effected.

What is an Availability Zone (AZ):

- AWS use (Availability Zones (AZ's)) which are basically groups of data Centers located in the same area, but Spread Out Enough to avoid being effected by the Same disaster.
- Each AZ has its own power, network, Connectivity, So if One AZ goes down, the other keeps running.

How AWS Prevents Downtime:

- When you run something like an Amazon EC2 instance [a Virtual Machine], AWS makes sure it runs inside a Availability Zone
- But they don't put all AZ's next to each other. They are placed far enough [10 miles at least] from each other so even if something happens to one AZ other is still working.
- And, even though the AZ's are far apart, they can still talk to each other quickly [low latency, like in milliseconds]

Best Practice for ~~High~~ Availability

- To make sure your App doesn't go down if there's a disaster. AWS Recommends running your app across at least two Availability Zones.
- This way if one AZ fails, the other one will keep your application running without interruption.

How AWS Services Work Across AZ's

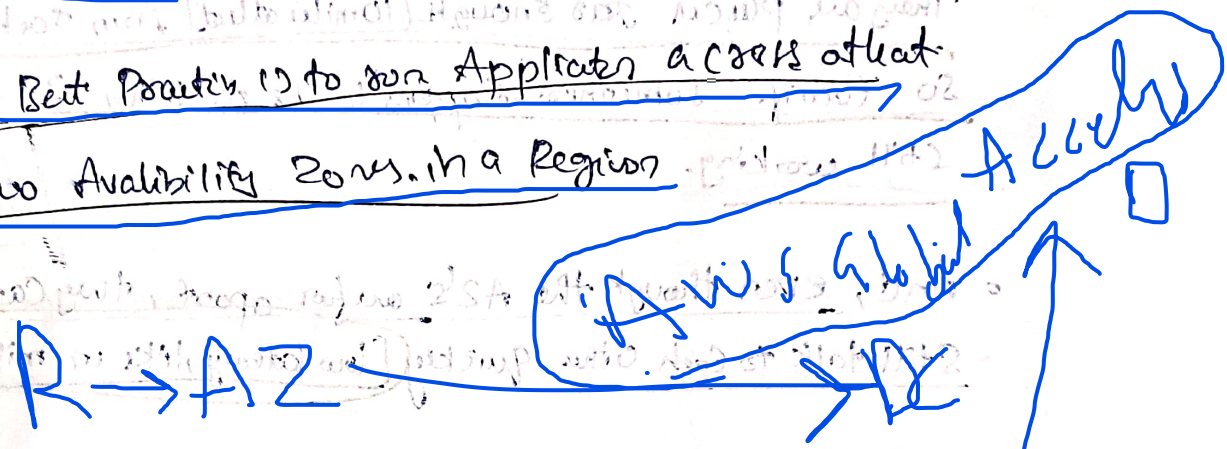
- Many AWS services [like Elastic Load Balancer] work across all AZ's in a Region automatically meaning they balance traffic across multiple AZ's for you without extra work.
- This make those services highly Available [they work even if one fails.

=) An Availability Zone is a single data Center or a group of Data Centers within a Region.

=) Availability Zones are located tens of miles apart from Each other.

=) This is close enough to have low latency between Availability Zones.

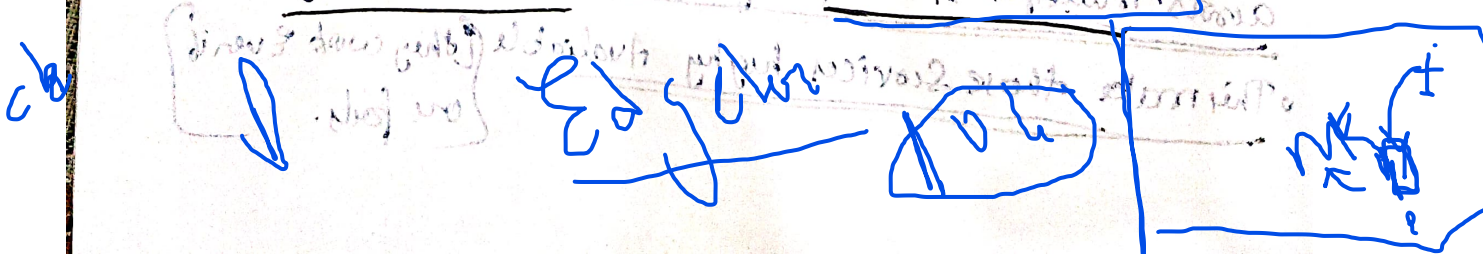
=) A Best Practice is to run Applications across atleast two Availability Zones in a Region



How AWS helps You serve Global Customers.

- Imagine You have Customers all over the world, but you only have your data in one AWS Region [Eg Tokyo]. If Customer in Mumbai want to access your data, they'd send request all the way to Tokyo, which take time.

- Solution: Instead of making all mumbai Customers Reach Tokyo. You can Cache a Copy of data in Mumbai. This makes it faster for Customers to get the data Because it's closer to them.



What is a Content Delivery Network (CDN)?

AWS Cloud Front

- A Content Delivery Network (CDN), helps deliver your Content, like video or applicatn, to users around the world faster.
- AWS version of CDN, it called Amazon Cloud Front
  - Cloud Front uses Edge locations Special data Centers spread around the world, to cache Content and speed up delivery to Customers, no matter where they are.

Edge locations

-> Edge location are different from Regions.

=) When you push Content from your AWS Region (like Tokyo) Cloud Front takes the Content and sends it to these Edge locations around the world.

=) Edge locations also help direct traffic with Amazon Route 53 (Domain Name System of AWS) where Entire Customer are routed to the Right place, again with low latency (fast Connection)

AWS Outposts (AWS in your Building)

⇒ AWS Outpost is a service where AWS setup a mini Region inside your Building.

- This means you get all of AWS benifits but within your own data Centre, for Special Cases. where you need things to stay local.
- It is not something most Business need, But for certain Situation it's available.

Region: Geographically isolated area you where you run your applications

Availability zone (AZ's): Multiple data Centre in each Region. They help you keep your app Running in case of a disster by Providing high availability.

Edge location: AWS's global network that delivers Content to User faster, Using Amazon Cloud front and Amazon Route 53.

Aws Outposts: A mini AWS data Centre setup in your Own Building for Special Case.

How to Interact with AWS Service

API in AWS:

- API (Application Programming Interface): A way to interact with AWS services programmatically.
- Everything in AWS is done through API Calls, which means when you want to create, configure or manage AWS Resources, you are making API Request.

Way to Interact with AWS Services:

1) AWS Management Console:
   - Browser Based Interface, you can view, visually manage AWS Services.
   - Great For Beginners, or for tasks like checking AWS bills, monitoring & building test environment.
   - However for Production environment, manually clicks can cause Error so we don't use there.

(2) AWS Command line Interface (AWS CLI) :-

- The CLI lets you use your terminal (Command line) to make API calls.
- Automation: You can script these API calls to repeat tasks like launching an EC2 instance, without needing to click through a UI.
- This makes your process faster, more consistent and less prone to human Error.

[3] AWS Software Development kit (SDK):

- SDK's allows you to interact with AWS using popular programming languages (like python, Java, Ruby).
- These kits lets you Build applications that interact with AWS without worrying about low-level API calls.
- Automation: SDK's help avoid the manual steps involved in creating services, just like the CLI, But with the added benefit of working within your Code.

Why Use Automation & Scripting?

- Reduce Human Error.
- Efficiency: with (CLI & scripting), you can automate repetitive task & ensure Consitny across deployments.
- Better managend: You can schedule task or trigger actions automatically based on certain events.

AWS Elastic Beanstalk

- With AWS Elastic Beanstalk, you provid Code an Configuration settings and Elastic Beanstalk deploys the Resources necessary to perform the following tasks
  - Adjust Capacity
  - load balancing
  - Automatic scaling
  - Application health monitoring.

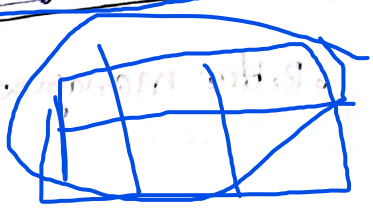
AWS Cloud Formation

- With AWS cloudformation, you can treat your Infrastru as Code. This means that you can build an Environment by writing lines of Code instead of using AWS management Console to manually Configure.
- AWS cloud formation provision your resources in a safe, Repeatable manner, enabling you to frequently build your Infrastructure and application without having to perform manual action.
- It determine the right operation to perform when managing your stack and roll back & changes automatically if it detects errors.

|| Networking ||

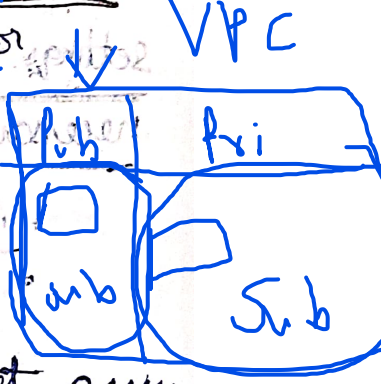
Amazon Virtual Private Cloud (VPC)

- In the world of AWS, VPC's (Virtual Private Clouds) are like creating a virtual sections within your cloud to keep things organised and secure.



What Is a VPC?

- A VPC is a virtual network in the AWS cloud where you can Run your Service.
- You can define the networks layout, control the traffic between services and isolate your services for Security.
- Resource within a VPC can be:
  - Public - these can access the Internet
  - Private - these don't have direct Internet access.



How does It work?

- Subnets:- A VPC is broken into subnets, which are like separate sections of the Coffee Shop.

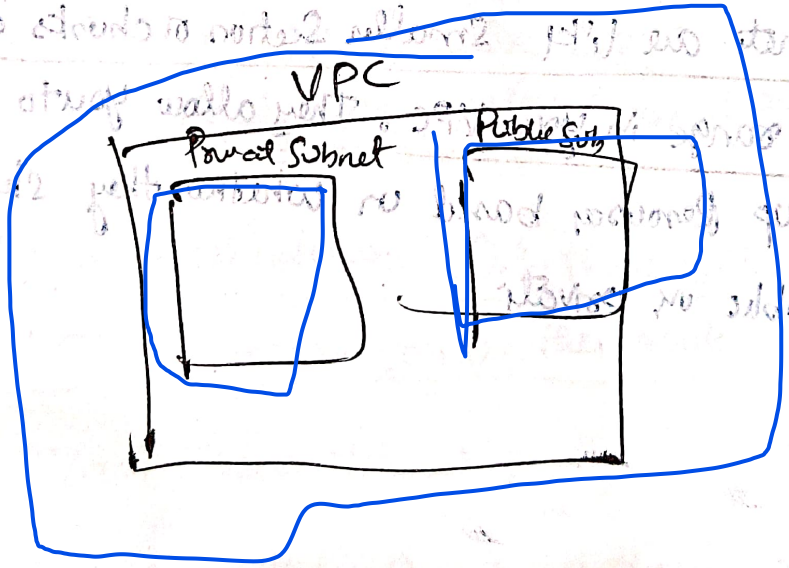
  → Public Subnet: Resources that need to interact with Customer and the Internet.

  → Private Subnet: Resources that shouldn't interact directly with Customers, usually for Back-end Services (like database)

Why Is this Useful?

- It keeps things organized: You can control which service can talk to each other or to the Outside world (Internet)
- It keeps your secure you Envirsoment: By keeping sensitive (like your databases or Internal serves) in a private subnet, they can't be directly accessed from the Outside.

=) VPCs give you control over how your cloud resources interact with each other, and the world outside.

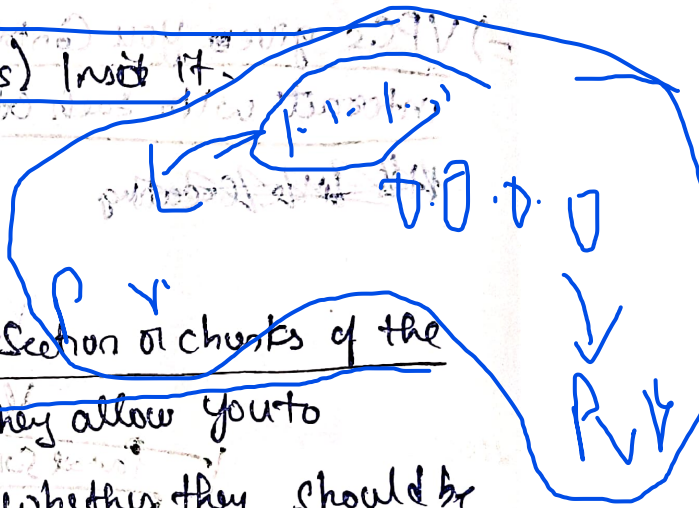


Amazon VPC

1) Virtual Private cloud

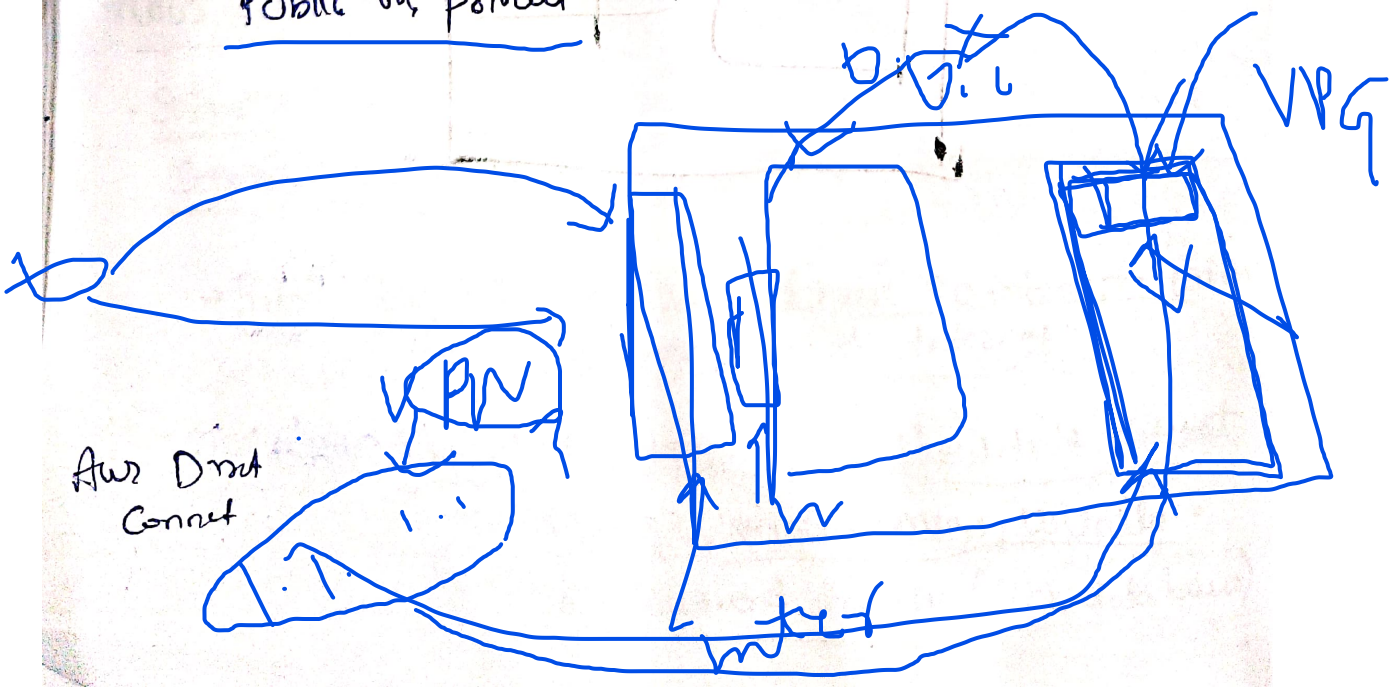
→ A network service that you use to Establish boundries around your AWS Resources i.e Amazon Virtual Private Cloud (Amazon VPC)

→ It is your Private Network in the cloud. You define your own IP address range within the VPC and Place resources like EC2 instance & Elastic Load Balancer (ELB's) Inside it.



2) Subnets:

- Subnets are like Smaller Section or chunks of the IP range in your VPC. They allow you to group Resources based on whether they should be Public or private.



(3) Public Facing Resources:

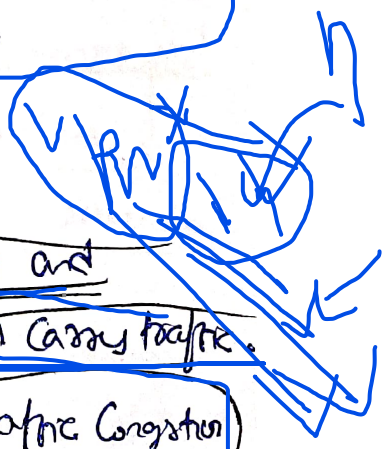
- If you have resource that need to be accessible from the internet (website), you need to attach an Internet Gateway (IGW) to your VPC.
- Internet Gateway (IGW): Think it as a front door to your VPC, without this, no one can reach your public resources inside your VPC.

(4) Private Resources: → Connect with VPN

- For Resource you don't want the public to access (like internal databases or applications), you don't attach an Internet Gateway. Instead you create a Virtual Private Gateway (VGW), which connects your private network (your office or on-premises) to your VPC securely using a VPN (Virtual Private Network)
- Virtual Private Gateway: Only those with the proper permissions can use this route.

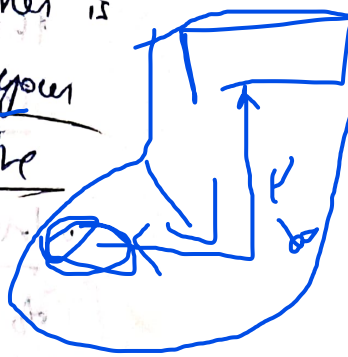
(5) VPN (Virtual Private Network):

- Even though a VPN Connection is secure and encrypted it still uses the public internet to carry traffic.
- This can lead to potential issues like traffic congestion or slower speed because internet is shared by many people.



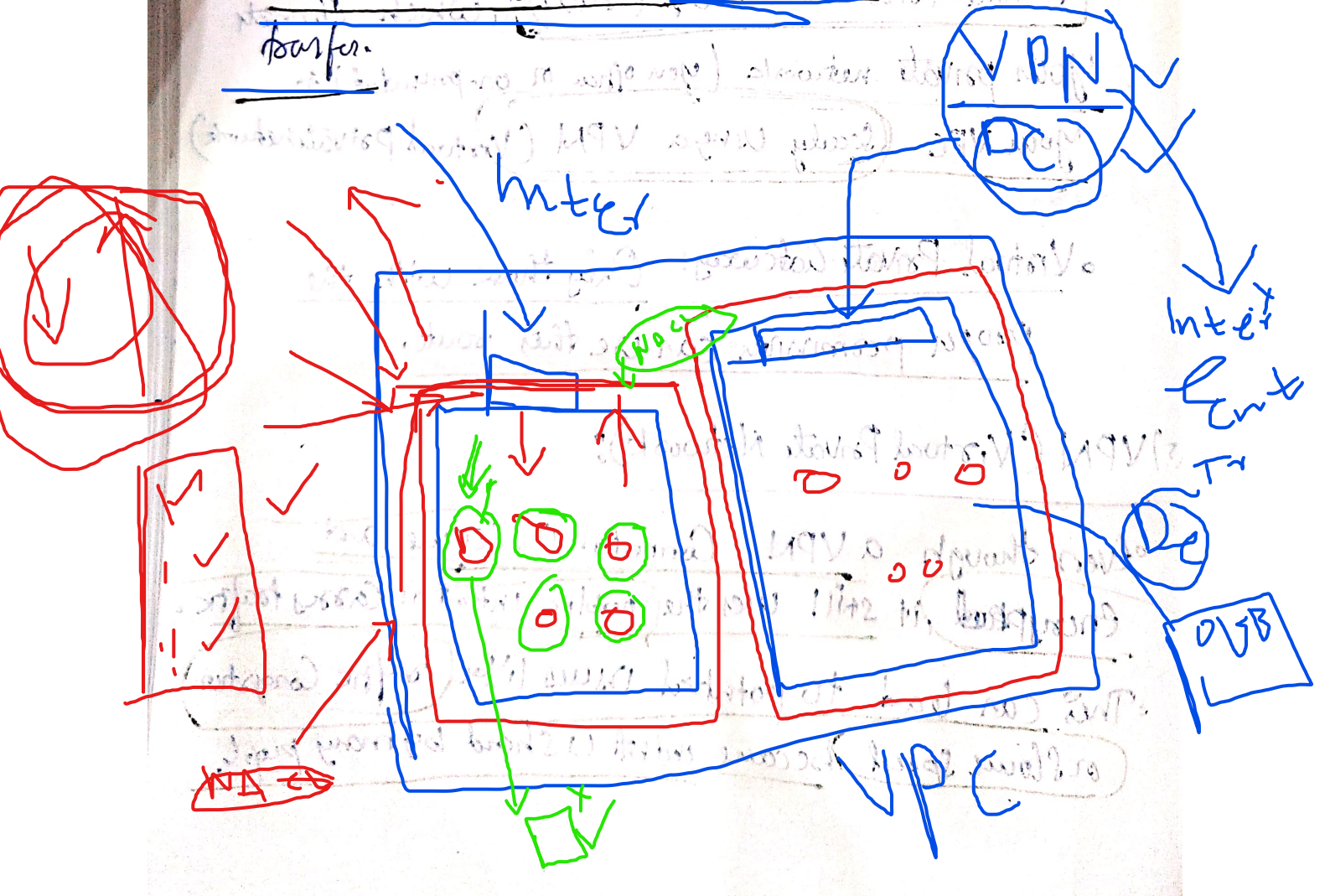
(6) AWS Direct Connect

- If you want a more reliable and faster Connection, you can use AWS Direct Connect. This is like a private magic doorway that Connects your Internet network directly to AWS, by passing the public internet entirely.



→ Its a dedicated fiber Connection that guarantees better performance with no shared bandwidth, making it ideal for high security and high-complicated needs.

- AWS Direct Connect ensure low latency and high security for your private Connection to AWS, and its especially helpful when you have a lot of critical data to transfer.



Subnet & Network access Control List

Network ACL (Network Access Control List)

- They are Boundry for Subnet inside VPC
- They Check every Packet that tries to pass in or out of a Subnet
  - If sender in in the list then he is allowed through it
  - If they're not on the list they are blocked
  - They check both incoming and Outgoing traffic everytime.

→ NACLs are Stateless.

- They forget past approvels. Everytime a packet enters or exits, they check the list again

Security Groups

- There are Bondry's Gate For your each EC2 instance inside a Subnet
  - By default, There Blocks everyone from Coming in Until you give them list of approved visitors.
  - But these are Stateful, meaning they remember who was let in
  - If a visitor leaves and Comes back, the Security Groups remember and lets them back without Checking the list again.

Network ACLs = Stateless, always checks, Boundry for Subnet

Security Group = Stateful, Remembers Past approvals, Boundry for instance.

=) Network ACLs is stateless and allows all inbound and Out bound traffic By default.

- By default, your account's default network ACL allows all inbound and outbound traffic, but you can modify it by adding your own rules.
- For Custom network ACLs, all in bound & out bound traffic is denied until you add rules. to specify which traffic to be allowed.
- Additionally all network ACLs have an Explicy deny rule. This Rule Ensures that if a Packet doesn't match any of the Rule on the list, the Packet is denied

Global Networking

DNS (Route 53)

→ Domain Name System

→ Customer can access our website by entering our website name just because of this DNS.

→ DNS resolution involves a Customer DNS resolver communicating with Company DNS servers.

→ It provide some Name mapped to the IP address.

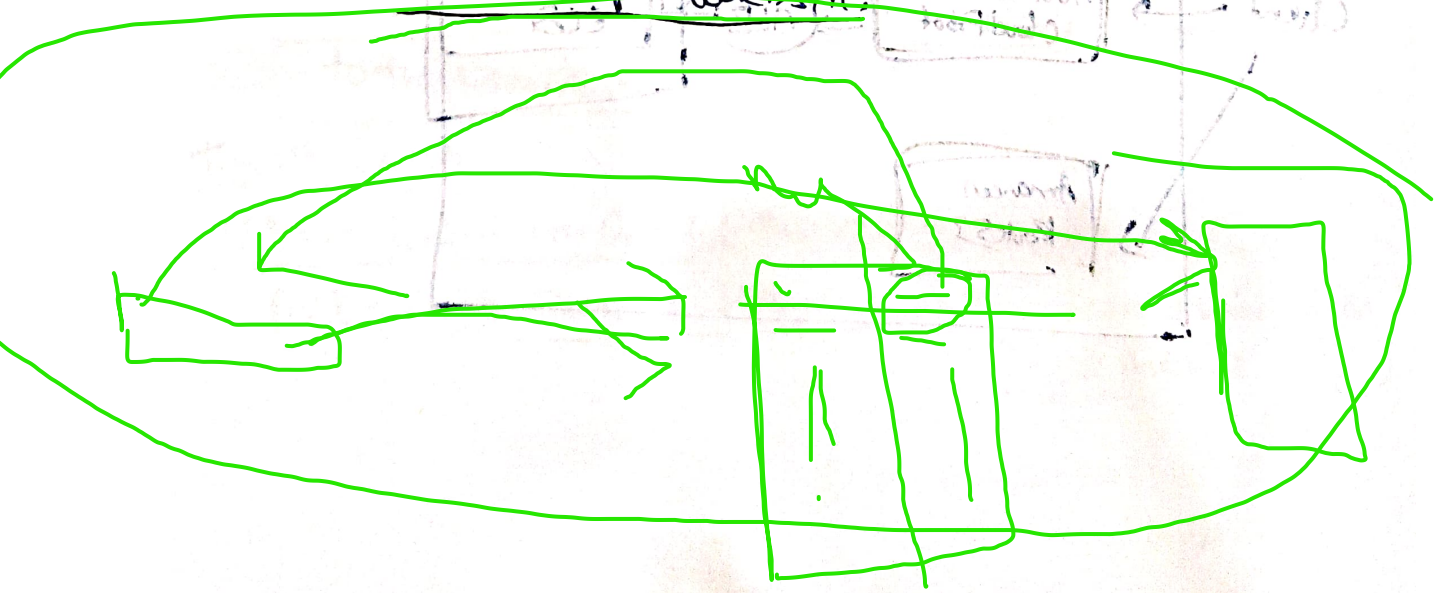
→ DNS resolution is the process of translating a domain name to an IP address.

Example: you visit any website

(1) when you enter any domain name in browser, this request is sent to Customer DNS resolver.

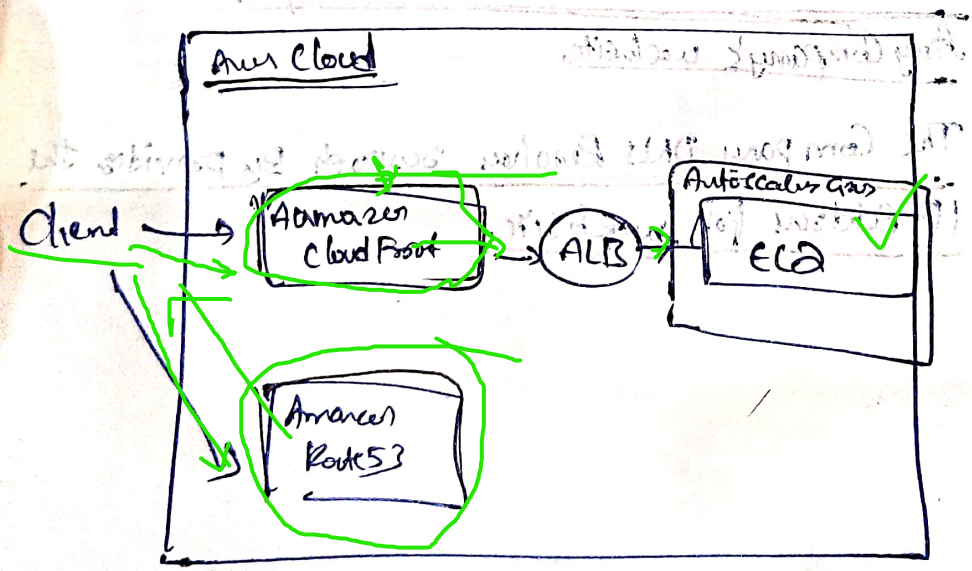
(2) The Customer DNS resolver asks the Company DNS servers for the IP Address that corresponds to Any Company's website.

(3) The Company DNS Resolver responds By providing the IP Address for website.



Amazon Route 53

- Amazon Route53 is a DNS webservice. It gives developer & Business a reliable way to route end users to Internet applications hosted in AWS.
- Amazon Route 53 Connect user requests to infrastructure running in AWS (EC2 etc). It can also be used to infrastructure outside of AWS.
- You can Register new Domain names directly in Route 53.
- You can also transfer DNS records for existing domain names managed by other domain registrars.
- This Enables you to manage all of your domain names within a single location.

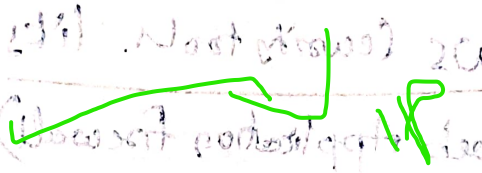


(1) A Customer requests from Browser to a website.
data

(2) Amazon Route 53 uses DNS resolution to identify the website's IP and sends the information to the Customer.

(3) The Customer request is send to the nearest Edge location through Amazon Cloud Front

(4) Amazon Cloud Front Connects to the Application Load Balancer, which sends the incoming packets to an Amazon EC2 instance.



Amazon cloud front - The Fast & Secure Content Delivery Network

* It does By storing a Copies of your Content (like images, videos, & webpage) in multiple locations closer to the user/customer.

How it works

1) Your website has Content: This Content can be images or videos Or Even Full website. They are stored in AWS (like in S3 bucket or EC2 instance

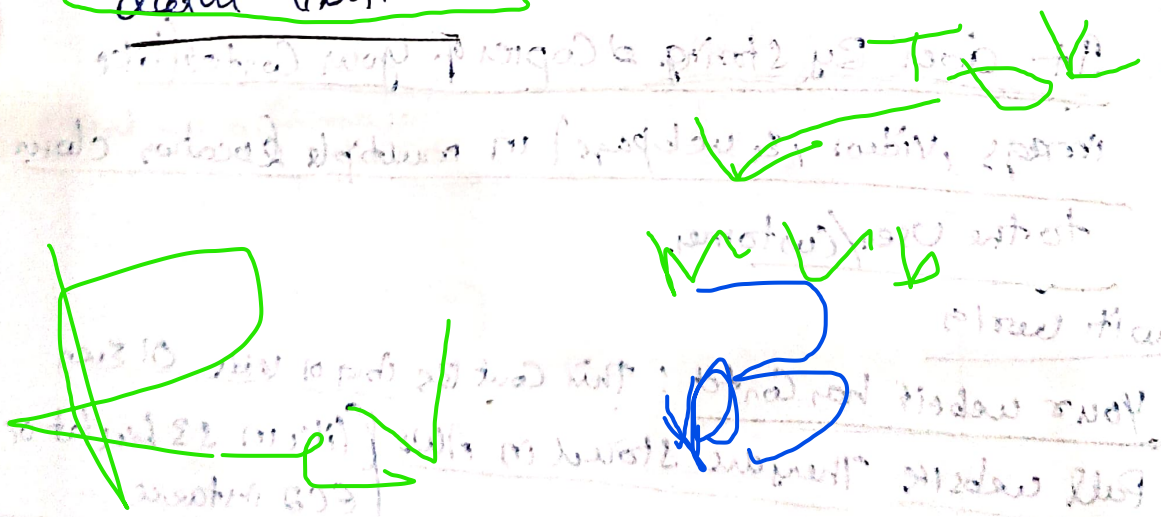
(2) CloudFront Stores the Copies of it worldwide

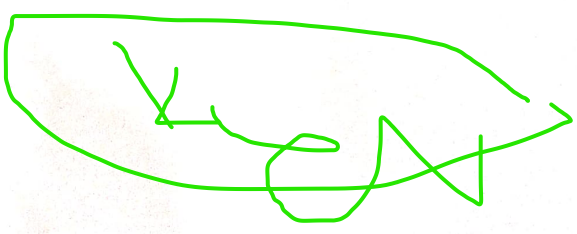
* CloudFront has Edge locations (Servers) all over the world. When someone visits your website, CloudFront serves the content from the nearest Edge location.

(3) As the data is closer to customer, It delivers fast & with low latency.

(4) Security & Protection: CloudFront helps protect against DDos attacks, encrypts data, and works with AWS security tools like AWS Shield and WAF (Web Application Firewall)

(5) Automatic Update: If your content changes, CloudFront updates its copies so user always gets the latest version.





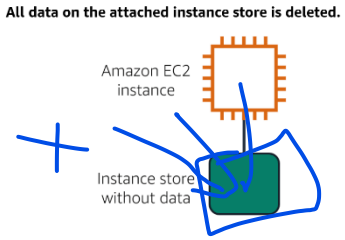
STORAGE AND DATABASE

Instance Stores and Amazon Elastic Block Store (Amazon EBS)

Instance stores

Block-level storage volumes behave like physical hard drives.

An instance store(opens in a new tab) provides temporary block-level storage for an Amazon EC2 instance. An instance store is disk storage that is physically attached to the host computer for an EC2 instance, and therefore has the same lifespan as the instance. When the instance is terminated, you lose any data in the instance store.



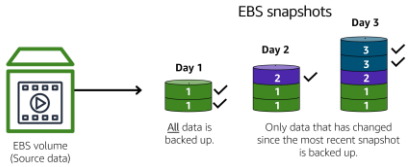
Amazon EC2 instances are virtual servers. If you start an instance from a stopped state, the instance might start on another host, where the previously used instance store volume does not exist. Therefore, AWS recommends instance stores for use cases that involve temporary data that you do not need in the long term.

Amazon Elastic Block Store (Amazon EBS)

Amazon Elastic Block Store (Amazon EBS) is a service that provides block-level storage volumes that you can use with Amazon EC2 instances. If you stop or terminate an Amazon EC2 instance, all the data on the attached EBS volume remains available.

To create an EBS volume, you define the configuration (such as volume size and type) and provision it. After you create an EBS volume, it can attach to an Amazon EC2 instance.

Because EBS volumes are for data that needs to persist, it's important to back up the data. You can take incremental backups of EBS volumes by creating Amazon EBS snapshots.



Amazon EBS snapshots

Incremental backups of EBS volumes with Amazon EBS snapshots. On Day 1, two volumes are backed up. Day 2 adds one new volume and the new volume is backed up. Day 3 adds two more volumes for a total of five volumes. Only the two new volumes are backed up.

An EBS snapshot is an incremental backup. This means that the first backup taken of a volume copies all the data. For subsequent backups, only the blocks of data that have changed since the most recent snapshot are saved.

Incremental backups are different from full backups, in which all the data in a storage volume copies each time a backup occurs. The full backup includes data that has not changed since the most recent backup.

Think of an **EC2 instance** like a computer running in the cloud. Just like your laptop or desktop, it needs **storage** to save files and data.

Two Types of Storage for EC2

1. **Instance Store (Temporary Storage)** 📂
    - This is like a built-in hard drive that comes with your EC2 instance.
    - It is fast but **temporary**—if you stop or delete the EC2 instance, all data on this storage is **lost**.
    - Good for storing temporary files, cache, or scratch data that isn't important.
2. **EBS (Elastic Block Store) – Permanent Storage** 💾
    - This is like an **external hard drive** you attach to your EC2 instance.
    - Even if you stop or restart the EC2 instance, the data remains safe.
    - You can **choose the size and type** of the EBS volume based on your needs.
    - Ideal for storing **databases, application files, or any important data** that must not be lost.

Backing Up Your Data – Snapshots 📸

To prevent losing important data, AWS allows you to take **snapshots** (backups) of your EBS volumes. If something goes wrong, you can **restore** your data from a snapshot.

Key Takeaways

✓ **Use Instance Store** for temporary data (fast but disappears if the instance stops).
✓ **Use EBS** for important data (remains even if the instance is stopped or restarted).
✓ **Take Snapshots** of EBS volumes to avoid losing important data.

You **can** take snapshots manually, but AWS also allows you to **automate** the process.

1 Manual Snapshots 📸

- You can go to the **AWS Console**, find your EBS volume, and click **"Create Snapshot."**
- Useful if you only need occasional backups.

2 Automated Snapshots 🔄

- AWS provides a service called **Amazon Data Lifecycle Manager (DLM)** to **schedule automatic snapshots** at regular intervals (daily, weekly, etc.).
- This is **better** than manual snapshots because it ensures you always have up-to-date backups.

Which one should you use?

✅ **For critical data (like databases or business applications):** Use **automated snapshots** to avoid forgetting backups.
✅ **For less critical data:** You can take **manual snapshots** as needed.

Amazon Simple Storage Service (Amazon S3)

In **object storage**, each object consists of data, metadata, and a key.

The data might be an image, video, text document, or any other type of file. Metadata contains information about what the data is, how it is used, the object size, and so on. An object's key is its unique identifier.

Amazon Simple Storage Service is a service that provides object-level storage. Amazon S3 stores data as objects in buckets.

You can upload any type of file to Amazon S3, such as images, videos, text files, and so on. For example, you might use Amazon S3 to store backup files, media files for a website, or archived documents. Amazon S3 offers unlimited storage space. The maximum file size for an object in Amazon S3 is 5 TB.

When you upload a file to Amazon S3, you can set permissions to control visibility and access to it. You can also use the Amazon S3 versioning feature to track changes to your objects over time.

Recall that when you modify a file in block storage, only the pieces that are changed are updated. When a file in object storage is modified, the entire object is updated.

Amazon S3 storage classes

With Amazon S3, you pay only for what you use. You can choose from a range of storage classes to select a fit for your business and cost needs. When selecting an Amazon S3 storage class, consider these two factors:

- How often you plan to retrieve your data
- How available you need your data to be

S3 Standard

- Designed for frequently accessed data
- Stores data in a minimum of three Availability Zones

Amazon S3 Standard provides high availability for objects. This makes it a good choice for a wide range of use cases, such as websites, content distribution, and data analytics. Amazon S3 Standard has a higher cost than other storage classes intended for infrequently accessed data and archival storage.

S3 Standard-Infrequent Access (S3 Standard-IA)

- Ideal for infrequently accessed data
- Similar to Amazon S3 Standard but has a lower storage price and higher retrieval price

Amazon S3 Standard-IA is ideal for data infrequently accessed but requires high availability when needed. Both Amazon S3 Standard and Amazon S3 Standard-IA store data in a minimum of three Availability Zones. Amazon S3 Standard-IA provides the same level of availability as Amazon S3 Standard but with a lower storage price and a higher retrieval price.

S3 One Zone-Infrequent Access (S3 One Zone-IA)

- Stores data in a single Availability Zone
- Has a lower storage price than Amazon S3 Standard-IA

Compared to S3 Standard and S3 Standard-IA, which store data in a minimum of three Availability Zones, S3 One Zone-IA stores data in a single Availability Zone. This makes it a good storage class to consider if the following conditions apply:

- You want to save costs on storage.
- You can easily reproduce your data in the event of an Availability Zone failure.

S3 Intelligent-Tiering

- Ideal for data with unknown or changing access patterns
- Requires a small monthly monitoring and automation fee per object

In the S3 Intelligent-Tiering storage class, Amazon S3 monitors objects' access patterns. If you haven't accessed an object for 30 consecutive days, Amazon S3 automatically moves it to the infrequent access tier, S3 Standard-IA. If you access an object in the infrequent access tier, Amazon S3 automatically moves it to the frequent access tier, S3 Standard.

S3 Glacier Instant Retrieval

- Works well for archived data that requires immediate access
- Can retrieve objects within a few milliseconds

When you decide between the options for archival storage, consider how quickly you must retrieve the archived objects. You can retrieve objects stored in the S3 Glacier Instant Retrieval storage class within milliseconds, with the same performance as S3 Standard.

S3 Glacier Flexible Retrieval

- Low-cost storage designed for data archiving
- Able to retrieve objects within a few minutes to hours

S3 Glacier Flexible Retrieval is a low-cost storage class that is ideal for data archiving. For example, you might use this storage class to store archived customer records or older photos and video files. You can retrieve your data from S3 Glacier Flexible Retrieval from 1 minute to 12 hours.

S3 Glacier Deep Archive

- Lowest-cost object storage class ideal for archiving
- Able to retrieve objects within 12 hours

S3 Deep Archive supports long-term retention and digital preservation for data that might be accessed once or twice in a year. This storage class is the lowest-cost storage in the AWS Cloud, with data retrieval from 12 to 48 hours. All objects from this storage class are replicated and stored across at least three geographically dispersed Availability Zones.

S3 Outposts

- Creates S3 buckets on Amazon S3 Outposts
- Makes it easier to retrieve, store, and access data on AWS Outposts

Amazon S3 Outposts delivers object storage to your on-premises AWS Outposts environment. Amazon S3 Outposts is designed to store data durably and redundantly across multiple devices and servers on your Outposts. It works well for workloads with local data residency requirements that must satisfy demanding performance needs by keeping data close to on-premises applications.

What is Amazon S3?

Amazon **Simple Storage Service (S3)** is an online storage service where you can **store and retrieve** as much data as you need. It's like a **giant online hard drive** that can hold anything—documents, images, videos, backups, and more.

💡 **Think of it like this:**
Imagine you own a coffee shop and need a place to store receipts, employee training videos, and customer feedback files. **Amazon S3 lets you store all of these safely and access them anytime.**

How Does Amazon S3 Store Data?

1️⃣ Objects & Buckets

- **Object** = A file (like a document, video, image, etc.)
- **Bucket** = A folder that holds multiple objects

💡 **Example:**
Imagine you have a folder on your computer called **"Coffee Shop Data."** Inside, you have files like **menu.pdf, employees.xlsx, and sales.csv.**

- Each file (menu.pdf, employees.xlsx, sales.csv) is an **object**
- The folder "Coffee Shop Data" is a **bucket**

You can create multiple buckets, organize your files however you want, and even **control who has access** to your files.

Amazon S3 Features

2️⃣ Object Versioning – Protecting Files from Accidental Deletion

- Every time you update a file, S3 can **keep old versions** instead of overwriting them.
- This means you **never lose** an older version by mistake.

💡 **Example:**
You update an Excel sheet **(sales.xlsx)** every day. If you accidentally delete it, **S3 keeps the older versions** so you can restore them.

3️⃣ Different Storage Classes (Tiers) for Different Needs

S3 has **different types of storage classes** depending on how often you need to access your data.

✅ **S3 Standard (For everyday use)**

- Stores **important and frequently accessed** data.
- **99.999999999% durability** (AWS calls this **"11 nines"**)
- Data is saved across **at least 3 locations**, so even if one storage facility fails, your data is safe.

💡 **Best for:** Storing website images, application files, and frequently used documents.

✅ **S3 Standard-Infrequent Access (S3 Standard-IA) – For less frequently accessed data**

- Cheaper than S3 Standard but **still quick to retrieve**
- Used for **backups, disaster recovery files, and archives**

💡 **Best for:** Storing monthly reports, backups, or customer order history that you don't check daily.

✅ **S3 Glacier Flexible Retrieval – For Long-Term Archiving**

- For **storing old data** that you **rarely** need but must keep for years.
- Takes **minutes to hours** to retrieve data.
- Cheaper than S3 Standard-IA.

💡 **Best for:** Audit records, compliance documents, old customer data, or tax records.

✅ **S3 Glacier Vault Lock – Ensuring Data Cannot Be Changed**

- Locks your data with a **"Write Once, Read Many" (WORM)** policy.
- Prevents files from being **deleted or changed**, ensuring compliance with regulations.

💡 **Best for:** Financial data, legal documents, and records that must be stored without modification.

4️⃣ Lifecycle Policies – Automate Data Movement Between Storage Classes

Instead of manually moving files from **S3 Standard to Glacier**, you can set up **lifecycle policies** to do it automatically.

💡 **Example:**
You want to:
✓ Keep files in **S3 Standard** for 3 months **(fast access)**
✓ Move them to **S3 Standard-IA** after 90 days **(cheaper storage)**
✓ Archive them in **S3 Glacier** after 120 days **(long-term storage)**

S3 **automatically moves** the data based on your policy without changing your application code.

5️⃣ Static Website Hosting – Hosting Websites on S3

You can **upload HTML, CSS, and images** to an S3 bucket and **host a static website.**

💡 **Example:**
You create a **coffee shop website** with an **HTML page, images, and a menu** and upload it to an S3 bucket. You **enable website hosting**, and **boom!** Your website is live with an Amazon S3 link.

6️⃣ Other Storage Classes for Special Cases

Besides the main ones, Amazon S3 also offers:

- **S3 One Zone-IA** (Cheaper but stored in only **one** AWS region)
- **S3 Glacier Instant Retrieval** (Archive storage with **faster retrieval**)
- **S3 Glacier Deep Archive** (For **long-term storage**, cheapest option, but takes **12+ hours** to retrieve files)

Final Takeaways ⏳

✓ **Amazon S3** stores all kinds of files (documents, videos, backups, etc.).
✓ **Buckets** hold files (like folders).
✓ **Versioning** protects files from accidental deletion.
✓ **Different storage classes** help you save money based on how often you need data.
✓ **Lifecycle policies** move data automatically between storage tiers.
✓ **S3 Glacier** is best for long-term storage.
✓ **S3 can even host static websites!**

Amazon EBS (Elastic Block Store) vs. Amazon S3 (Simple Storage Service)

Amazon EBS (Block Storage)

- **Best for:** Databases, virtual machines, and applications needing frequent read/write access.
- **How it works:** Stores data in **blocks**, like a traditional hard drive. Only the changed blocks are updated when modifying data.
- **Use case:** Ideal for workloads requiring **frequent updates**, such as **video editing, databases, and application storage**.
- **Persistence:** Data persists even if the EC2 instance is stopped.
- **Performance:** Provides **low-latency, high-speed access** to data.

Amazon S3 (Object Storage)

- **Best for:** Storing large amounts of static data, such as images, videos, backups, and logs.
- **How it works:** Stores data as **objects** in **buckets**. When an object is updated, the entire object must be replaced.
- **Use case:** Best for **write-once, read-many** scenarios like website assets, backups, and archival storage.
- **Durability:** 99.999999999% (11 nines), meaning data is highly reliable.

- **Scalability:** Virtually **unlimited storage** capacity.
- **Web Access:** Every object gets a unique URL for easy access and sharing.

Key Differences

Feature	Amazon EBS	Amazon S3
Storage Type	Block Storage	Object Storage
Use Case	Databases, frequent file edits	Static files, backups, media storage
Data Updates	Only changed blocks are updated	Whole object must be replaced
Scalability	Limited to 16 TiB per volume	Virtually unlimited
Durability	Tied to EC2 instance	99.999999999% durability
Performance	Low-latency, high-speed access	Web-accessible, scalable

Final Thought:

- **Use EBS when you need high-performance storage for applications that modify data frequently.**
- **Use S3 for storing and retrieving large amounts of static data at scale.**

Amazon Elastic File System (Amazon EFS)

File storage

In **file storage**, multiple clients (such as users, applications, servers, and so on) can access data that is stored in shared file folders. In this approach, a storage server uses block storage with a local file system to organize files. Clients access data through file paths.

Compared to block storage and object storage, file storage is ideal for use cases in which a large number of services and resources need to access the same data at the same time.

Amazon Elastic File System (Amazon EFS) is a scalable file system used with AWS Cloud services and on-premises resources. As you add and remove files, Amazon EFS grows and shrinks automatically. It can scale on demand to petabytes without disrupting applications.

Comparing Amazon EBS and Amazon EFS

Amazon EBS

An Amazon EBS volume stores data in a **single** Availability Zone.

To attach an Amazon EC2 instance to an EBS volume, both the Amazon EC2 instance and the EBS volume must reside within the same Availability Zone.

Amazon EFS is a regional service. It stores data in and across **multiple** Availability Zones.

The duplicate storage enables you to access data concurrently from all the Availability Zones in the Region where a file system is located. Additionally, on-premises servers can access Amazon EFS using AWS Direct Connect.

Amazon Elastic File System (EFS) Overview

What is EFS?

- A **managed file system** that multiple EC2 instances can access **simultaneously**.
- Automatically **scales up or down** based on the amount of data stored.
- Ideal for **shared storage** across multiple servers or applications.
- No need for manual capacity planning or hardware management.

Key Differences: EBS vs. EFS

Feature	Amazon EBS	Amazon EFS
Storage Type	Block Storage (like a hard drive)	File Storage (like a shared network drive)
Accessibility	Tied to a **single EC2 instance**	Shared across **multiple EC2 instances**
Availability Scope	Availability Zone (AZ)	Region-wide
Scalability	**Fixed size** (must provision storage in advance)	**Automatically scales** as data grows
Use Case	Databases, frequently modified files	Shared storage, big data, analytics, web applications

When to Use EFS?

- When **multiple servers** need to access the same data.
- For **big data analytics**, where multiple instances process the same dataset.
- When you need **automatic scaling** without manual intervention.
- For **web applications** that require shared storage across instances.

Final Thought:

- **Use EBS** when you need dedicated, high-performance storage for a **single instance**.
- **Use EFS** when you need **shared, scalable storage** across **multiple instances**.

Amazon Relational Database Service (Amazon RDS)

Relational databases

In a **relational database**, data is stored in a way that relates it to other pieces of data.

Relational databases use **structured query language (SQL)** to store and query data. This approach allows data to be stored in an easily understandable, consistent, and scalable way.

Amazon Relational Database Service

Amazon Relational Database Service (Amazon RDS)(opens in a new tab) is a service that enables you to run relational databases in the AWS Cloud.

Amazon RDS is a managed service that automates tasks such as hardware provisioning, database setup, patching, and backups. With these capabilities, you can spend less time completing administrative tasks and more time using data to innovate your applications. You can integrate Amazon RDS with other services to fulfill your business and operational needs, such as using AWS Lambda to query your database from a serverless application.

Amazon RDS provides a number of different security options. Many Amazon RDS database engines offer encryption at rest (protecting data while it is stored) and encryption in transit (protecting data while it is being sent and received).

Amazon RDS database engines

Amazon RDS is available on six database engines, which optimize for memory, performance, or input/output (I/O). Supported database engines include:

- Amazon Aurora
- PostgreSQL
- MySQL
- MariaDB
- Oracle Database
- Microsoft SQL Server

Amazon Aurora

Amazon Aurora(opens in a new tab) is an enterprise-class relational database. It is compatible with MySQL and PostgreSQL relational databases. It is up to five times faster than standard MySQL databases and up to three times faster than standard PostgreSQL databases.

Amazon Aurora helps to reduce your database costs by reducing unnecessary input/output (I/O) operations, while ensuring that your database resources remain reliable and available.

Consider Amazon Aurora if your workloads require high availability. It replicates six copies of your data across three Availability Zones and continuously backs up your data to Amazon S3.

How to Move a Database to AWS? (Migration Options)

A. Lift-and-Shift Approach (EC2 Hosting)

- Move your existing database **as it is** to an **Amazon EC2 instance**.
- You get full control over **OS, memory, CPU, storage**—just like your own server.
- Requires manual management of updates, backups, and scaling.

B. Using Amazon RDS (Relational Database Service)

- A **managed database service** that supports major databases like MySQL, PostgreSQL, and SQL Server.
- AWS takes care of:
    - **Automatic backups**
    - **Software updates (patching)**
    - **Disaster recovery**
    - **Security and scaling**
- **Saves time** and reduces **manual work** for database administrators.

C. Using Amazon Aurora (Highly Optimized RDS)

- A **high-performance** relational database designed by AWS.
- Compatible with **MySQL and PostgreSQL** but **10 times faster** and **cheaper** than traditional databases.
- **Key Features:**
    - **Automatic replication** across multiple locations (6 copies).
    - **Up to 15 read replicas** to handle more users without slowing down.
    - **Continuous backups** stored in S3.
    - **Point-in-time recovery**, allowing you to restore from any past time.

Amazon DynamoDB

Amazon DynamoDB(opens in a new tab) is a key-value database service. It delivers single-digit millisecond performance at any scale.

To learn about features of DynamoDB, select each flashcard by choosing them.

- DynamoDB is a NoSQL, non-relational database.
- It scales automatically without the need for manual management.
- It has a flexible schema, allowing items to have different attributes.
- Designed for high performance with millisecond response times.
- No SQL queries are used; instead, queries focus on primary and secondary keys.
- Ideal for applications that require high scalability and fast data access.

What is DynamoDB?

- **Serverless Database**: No need to manage servers or infrastructure.
- **Managed Service**: AWS handles the scaling, availability, and redundancy.

How Data is Organized in DynamoDB:

- **Tables**: Where data is stored.
- **Items**: Individual entries within the table.

- **Attributes**: Characteristics or fields of each item (like name, date, etc.).

Scalability & Availability:

- **Automatic Scaling**: DynamoDB automatically scales up or down based on usage without user intervention.
- **Data Redundancy**: Data is stored across multiple availability zones (AZs) for high availability.
- **Mirrored Data**: Data is stored redundantly across multiple drives.

Performance:

- **Millisecond Response Time**: DynamoDB is designed to handle applications with millions of users while delivering fast performance (typically measured in milliseconds).

DynamoDB vs SQL Databases:

- **NoSQL vs. SQL**: DynamoDB is a NoSQL database, which means it doesn't require a rigid schema like SQL databases.
- **Schema Flexibility**: Items in DynamoDB tables don't need to have the same attributes, and attributes can be added or removed at any time.
- **Non-Relational**: It does not require relationships between tables like traditional SQL databases.

Use Cases for DynamoDB:

- Best suited for applications with varying or unpredictable data types (e.g., IoT, mobile apps, gaming).
- Great for datasets that have flexible or dynamic attributes.
- **Not Ideal for Complex Queries**: DynamoDB focuses on simple queries that retrieve data based on primary or secondary keys (no complex joins across multiple tables).

Querying in DynamoDB:

- **Keys**: Queries in DynamoDB are based on a subset of attributes called keys (Primary Key and Optional Secondary Keys).
- **Simpler Queries**: Queries tend to focus on individual items from a single table rather than joining multiple tables.

Why Choose DynamoDB:

- **Fully Managed**: AWS manages all operational aspects (scaling, backups, etc.).
- **Highly Scalable**: DynamoDB can handle very high request volumes (e.g., 7.11 trillion API calls during Prime Day 2019).
- **High Throughput**: DynamoDB can handle millions of requests per second without performance degradation.

Key Benefits:

- **Low Latency**: Millisecond response time.
- **Scalability**: Handles massive scales with automatic scaling.
- **Fully Managed**: Reduces operational overhead.
- **Cost-Effective**: Pay only for what you use, as it is serverless.

Relational Database (RDS):

- **Amazon RDS** is a **managed relational database service** that simplifies database management tasks like backups, patching, and scaling.
- You **control the data, schema, and network**.
- **Ideal for complex analytics** and business use cases where you need to join multiple tables and perform intricate queries.
- **Use Case Example**: A **sales supply chain management system** requires complex relational joins and data analysis, making RDS the better choice.

NoSQL Database (DynamoDB):

- **Amazon DynamoDB** is a **serverless, NoSQL database** that uses a **key-value store**. It doesn't require an advanced schema and offers global scalability at the touch of a button.
- DynamoDB is designed for high throughput and can scale to **petabyte-scale** data.

- **Granular API access** allows for fast and scalable data storage.
- **Ideal for simple, non-relational data** that doesn't require complex joins or relationships between tables.
- **Use Case Example**: An **employee contact list** (names, phone numbers, emails, IDs) is a simple data set that doesn't need the complexity of relational databases. DynamoDB is more efficient for this kind of use case.

 Winner Depends on Use Case:

- **RDS** is the winner for **complex data analysis** and scenarios requiring **relationships between tables** (e.g., sales or supply chain management).
- **DynamoDB** wins for **simple, high-throughput data** that doesn't require complex relationships (e.g., look-up tables like employee contact lists).

 Summary:

- **RDS**: Best for **complex, relational data** with joins and analytics.
- **DynamoDB**: Best for **simple, fast, scalable data** that doesn't need relational structure.
- Your **specific use case** will determine which database is best for your application.

Amazon Redshift

Amazon Redshift(opens in a new tab) is a data warehousing service that you can use for big data analytics. It offers the ability to collect data from many sources and helps you to understand relationships and trends across your data.

Overview of Data Management and Business Intelligence

When dealing with data that requires fast, real-time transactions, traditional relational databases work very well. These databases are optimized for managing large volumes of data with high availability and fast response times. However, there are cases where this type of database doesn't perform as well, especially when dealing with **historical data analysis** and **big data**.

Traditional Relational Databases and Their Limitations

Relational databases excel at handling current data that is actively being read or written, such as transactions, inventory systems, and other real-time operations. However, when it comes to **historical data analysis**, such as reviewing past performance over time, they can become inefficient. The **volume** of data generated, particularly with the rise of the Internet of Things (IoT) and telemetry systems, can overwhelm even the most robust relational databases.

Additionally, the **variety** of data sources further complicates analysis. For instance, querying data from multiple systems (like inventory, financials, and retail sales) can be challenging for traditional relational databases because they are not designed to aggregate and analyze data across different sources efficiently.

Enter Data Warehouses for Historical Analytics

When data complexity increases, relational databases might no longer be suitable for handling the analysis. This is where **data warehouses** come in. Data warehouses are designed specifically for handling **big data** and **historical analytics** rather than real-time operational data. They focus on organizing and storing large volumes of data from various sources for analytical purposes.

In contrast to traditional databases that handle current transactional data, data warehouses handle **historical queries**, such as looking at sales numbers from the past hour or year. The data in a data warehouse is **static** in the sense that once it's collected, it's not changing. This is different from operational data that is constantly being updated and modified.

Challenges with Traditional Data Warehouses

While data warehouses solve many problems related to historical analysis, managing and maintaining a data warehouse can still be complex. Traditionally, data warehouse teams have to deal with **tuning**, **scaling**, and ensuring that the system remains resilient and performant. This can take significant effort and time, diverting focus away from the data itself and onto the underlying infrastructure.

Amazon Redshift: Data Warehousing as a Service

To simplify the process of using data warehouses, **Amazon Redshift** offers a **fully managed data warehouse service** that handles the scaling, tuning, and maintenance for you. Redshift is designed to handle **massive data sets**, including petabyte-sized data warehouses. Redshift can scale seamlessly, making it suitable for organizations that need to work with big data.

In addition to handling large volumes of data, Redshift has a significant advantage: **performance**. Redshift uses advanced technologies that allow it to deliver up to **10 times the performance** of traditional data warehouses when running **business intelligence** (BI) workloads. These innovations in architecture and design make Redshift highly optimized for data analysis and querying.

Redshift Spectrum: An Extension for Big Data

One key feature of Amazon Redshift is **Redshift Spectrum**, which allows you to run SQL queries directly against **unstructured data** stored in **data lakes** (like S3). With Redshift Spectrum, you can query **exabytes** of unstructured data, significantly enhancing the ability to integrate structured and unstructured data in your analysis.

Key Benefits of Using Amazon Redshift

1. **Fully Managed**: Redshift eliminates the need to manage the underlying infrastructure, meaning your teams can focus on the data rather than the engine.
2. **Massive Scalability**: Redshift can handle petabyte-scale data warehouses and can scale up or down as needed.
3. **High Performance**: Redshift can provide up to 10x higher performance than traditional databases for big data BI workloads.
4. **Ease of Use**: Starting with Redshift is as simple as making a single API call, making it easier to integrate into your data pipeline.
5. **Redshift Spectrum**: Enables direct querying of unstructured data in data lakes, adding flexibility for working with large datasets.

Conclusion

In summary, while traditional relational databases are excellent for real-time data, they struggle when dealing with the vast amounts of historical data and complex analysis needed for business intelligence. **Amazon Redshift** is a powerful tool specifically built for large-scale data analysis, offering a fully managed service that handles the complexities of scaling, performance, and maintenance, enabling your teams to focus on generating insights rather than managing infrastructure.

AWS Database Migration Service

AWS Database Migration Service (AWS DMS)(opens in a new tab) enables you to migrate relational databases, nonrelational databases, and other types of data stores.

With AWS DMS, you move data between a source database and a target database. The source and target databases(opens in a new tab) can be of the same type or different types. During the migration, your source database remains operational, reducing downtime for any applications that rely on the database.

For example, suppose that you have a MySQL database that is stored on premises in an Amazon EC2 instance or in Amazon RDS. Consider the MySQL database to be your source database. Using AWS DMS, you could migrate your data to a target database, such as an Amazon Aurora database.

Other use cases for AWS DMS

1. Development and test database migrations
   Enabling developers to test applications against production data without affecting production users
2. Database consolidation
   Combining several databases into a single database
3. Continuous replication
   Sending ongoing copies of your data to other target sources instead of doing a one-time migration

Amazon Database Migration Service (DMS) helps simplify this process. It allows you to **migrate existing databases** to AWS **securely** and **easily**, while ensuring minimal downtime.

What Does Amazon DMS Do?

Amazon DMS enables the migration of data between **source** and **target** databases. The best part is that the **source database remains operational** during the migration, so the applications relying on it are unaffected.

The source and target databases **don't have to be the same type**, meaning DMS can handle migrations between different database platforms as well.

Types of Migrations Supported by DMS

1. **Homogeneous Migrations**:
   - This type of migration happens when both the **source and target databases are the same type** (e.g., MySQL to MySQL).
   - These migrations are typically straightforward because the **schema**, **data types**, and **database code** are compatible between the source and the target database.
   - Example: You can migrate from MySQL on-premises to **Amazon RDS for MySQL** or from **Microsoft SQL Server** to **Amazon RDS for SQL Server**.

How It Works:

   - You create a migration task that connects the **source** and **target databases**.

- AWS DMS handles the actual migration after you initiate it with a single click.

2. **Heterogeneous Migrations**:
   - In this case, the **source** and **target databases are of different types** (e.g., from Oracle to MySQL).
   - The process involves two steps:
     1. **Schema Conversion**: The **AWS Schema Conversion Tool (SCT)** is used to convert the **source database's schema** and code to match the structure of the target database.
     2. **Data Migration**: After schema conversion, DMS is used to move the data from the source database to the target database.

Additional Use Cases for Amazon DMS

1. **Development and Test Database Migrations**:
   - This use case allows you to migrate a **copy of your production database** to your development or test environment without affecting the production environment.
   - You can migrate the database once or **continuously**, which is helpful for ongoing testing against production-like data.
2. **Database Consolidation**:
   - If you have several databases and want to merge them into a **centralized database**, DMS helps with that as well. This is useful when consolidating various databases into one for easier management and reporting.
3. **Continuous Database Replication**:
   - DMS can be used for **continuous data replication**. This is especially helpful in scenarios like **disaster recovery** or when the database systems are located in **different geographic regions**.
   - Continuous replication ensures that your data remains up-to-date across multiple locations or systems.

Additional Database Services

When choosing a database or storage platform, it's crucial to align the database type with your business needs rather than forcing your data to fit a specific database. AWS offers various specialized databases for different use cases to ensure you're using the best tool for the job:

1. **Amazon DocumentDB**: Ideal for **content management systems**, **catalogs**, and **user profiles**, where you need more than just key-value pairs, offering flexibility in storing document-based data.

   Amazon DocumentDB(opens in a new tab) is a document database service that supports MongoDB workloads. (MongoDB is a document database program.)

2. **Amazon Neptune**: A **graph database** designed for use cases like **social networking**, **recommendation engines**, and **fraud detection**. It efficiently handles relationships between data points, like who is connected to whom in a social network.
3. **Amazon QLDB (Quantum Ledger Database)**: If you need **immutable records** for use cases like **banking** or **financial systems**, QLDB offers a tamper-proof system of record, ensuring that data entries can never be deleted, making it more suitable than a blockchain solution.
4. **Amazon Managed Blockchain**: A blockchain service, but more for **decentralized applications** requiring distributed ledger technology, though it's not ideal for regulated industries that require tamper-proof centralized ledgers like QLDB.
5. **Amazon ElastiCache**: Provides **caching layers** to improve the performance of read-heavy applications, reducing response times from milliseconds to microseconds. It supports both **Memcached** and **Redis** engines to handle different caching needs.
6. **DynamoDB Accelerator (DAX)**: A **caching layer** specifically for **DynamoDB**, improving read performance for non-relational data by speeding up common queries.

The key takeaway is that AWS provides a variety of database options and accelerators tailored to different requirements, ensuring businesses use the most efficient and appropriate tool for their workloads.

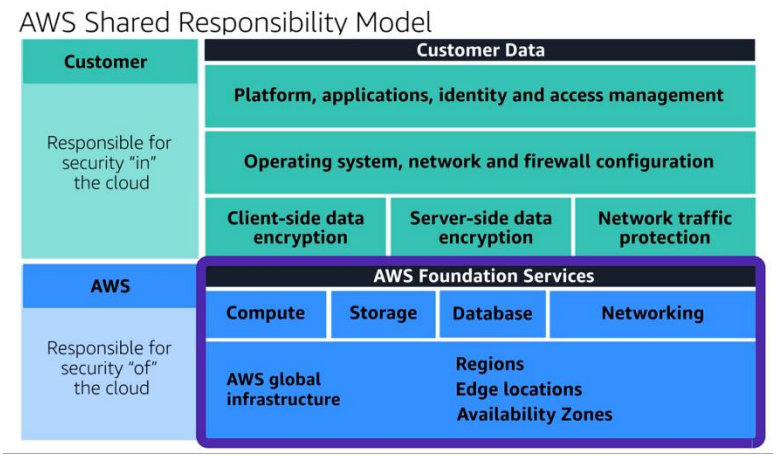
The correct response option is: ***EBS volumes store data within a single Availability Zone. Amazon EFS file systems store data across multiple Availability Zones***.

An EBS volume must be located in the same Availability Zone as the Amazon EC2 instance to which it is attached.

Data in an Amazon EFS file system can be accessed concurrently from all the Availability Zones in the Region where the file system is located.

Amazon Redshift is a data warehousing service that you can use for big data analytics. Use Amazon Redshift to collect data from many sources and help you understand relationships and trends across your data.

SECURITY



In AWS, **security** is a shared responsibility between AWS and the customer, which is outlined in the **Shared Responsibility Model**. Here's a breakdown of this model and other key AWS security services and mechanisms:

Shared Responsibility Model

- **AWS's Responsibility**: AWS is responsible for the **security of the cloud**, including the physical security of **data centers**, the **network infrastructure**, and **host infrastructure**. This encompasses hardware, software, networking, and the facilities that run AWS services.

- **Customer's Responsibility**: Customers are responsible for **securing the workloads in the cloud**. This includes configuring their applications, data, identity management, encryption, access control, and operating system patches. Essentially, customers manage everything from the **application layer** up.

The AWS shared responsibility model

The shared responsibility model divides into customer responsibilities (commonly referred to as "security in the cloud") and AWS responsibilities (commonly referred to as "security of the cloud").

The **Shared Responsibility Model** in AWS defines the division of security responsibilities between AWS and its customers. Here's how the model works:

1. AWS's Responsibility:

AWS is responsible for securing the **cloud infrastructure**, which includes:

- **Physical security**: AWS ensures the data centers where resources like EC2 instances are hosted are protected with strong physical measures (e.g., fences, security guards).

- **Network and hypervisor**: AWS secures the network infrastructure and the hypervisor layer (which controls virtualized environments). These layers are designed to be tamper-proof and are protected by AWS's internal security mechanisms.

2. Customer's Responsibility:

Customers are responsible for securing everything **in the cloud**, which includes:

- **Operating System (OS)**: When you choose an operating system for your EC2 instance, it's your responsibility to manage it. AWS cannot access or patch your operating system (except in specific cases). You are the only one with the encryption keys to access or modify it.

- **Applications**: You control and secure the applications you run on top of the OS. AWS does not manage these applications, so it's up to you to ensure they are secure and maintained.

- **Data**: Data security is entirely your responsibility. You decide who can access your data and under what conditions. AWS provides tools like encryption to protect data, whether it's open to the public or highly restricted (e.g., banking data).

Key Concept:

The division between AWS's responsibility (security of the cloud) and the customer's responsibility (security in the cloud) is crucial. AWS takes care of the infrastructure, but customers must manage everything above that, including the operating system, applications, and data. This shared responsibility ensures clarity and control for both AWS and its customers, allowing for secure cloud operations.

Analogy:

This model can be compared to owning a house:

- **AWS is the builder** who constructs the house (data centers) and ensures it's built securely.

- **You, the homeowner**, are responsible for securing the house (the OS, applications, and data) by locking the doors (securing the operating system) and managing who enters.

This approach ensures that security tasks are clearly defined, and both parties are responsible for specific layers in the cloud architecture.

User Permissions and Access

In AWS, **Identity and Access Management (IAM)** is used to control access to resources. Here's a breakdown of how IAM works, using a coffee shop analogy:

1. Root User:

- When you first create an AWS account, you get the **root user**. This user has **unlimited access** to all AWS services and resources, just like the owner of a coffee shop who can manage all parts of the business.
- **Security Tip**: Always enable **multi-factor authentication (MFA)** for the root user to add an extra layer of protection.
- Even though the root user can do anything, it's not ideal to use this account for everyday tasks.

2. IAM Users:

- You can create **IAM users** for different people in your organization. By default, these users have **no permissions** at all, similar to a cashier who can't access the inventory system unless explicitly granted permission.
- **Granular Control**: You **explicitly grant permissions** to users via **IAM policies** (JSON documents that define allowed actions on AWS resources).
- **Least Privilege Principle**: You grant only the permissions that a user needs, nothing more.

3. IAM Groups:

- To simplify permission management, you can group IAM users into **IAM groups** (e.g., cashiers, inventory managers). You can attach a policy to the group, and all users in that group get the same permissions, just like assigning all cashiers access to the register.

4. IAM Policies:

- **IAM policies** define permissions in a JSON format. For example, a policy might allow a user to list items in an S3 bucket but prevent them from uploading anything to it. Policies have an **effect** (Allow/Deny), an **action** (API calls), and a **resource** (specific AWS resource).

5. IAM Roles:

- **Roles** are identities with permissions that can be assumed temporarily, similar to assigning different tasks (e.g., register, inventory management) to an employee on different days.
- Unlike IAM users, **roles do not have a username or password**. Instead, they can be assumed by users, services, or applications, granting temporary permissions.
- **Temporary Access**: Roles are often used when you need to temporarily grant access, for example, to external users or other AWS services.

6. Federation:

- Instead of creating an IAM user for every person, you can **federate** users into AWS, allowing them to use existing corporate credentials (like Microsoft Active Directory) to log in and access AWS resources by mapping corporate identities to IAM roles.

Summary:

- **Root user**: Full access to all resources.
- **IAM users**: Specific, controlled access based on roles and policies.
- **IAM groups**: Easier management by grouping users with similar access.
- **IAM policies**: Define what actions a user can perform on specific resources.
- **IAM roles**: Temporary access with specific permissions.
- **Federation**: Allow users to log in with their existing credentials.

The shared responsibility model applies here too—AWS secures the infrastructure, but you're responsible for managing and securing access to your AWS resources using IAM.

You can enable MFA(Multi Factor Authentication) for the root user and IAM users. As a best practice, enable MFA for the root user and all IAM users in your account. By doing this, you can keep your AWS account safe from unauthorized access.

AWS Organizations: Managing Multiple Accounts Efficiently

When businesses first start using AWS, they usually begin with **a single AWS account**. But as the company grows, managing different teams, departments, or business units under one account can become chaotic.

For example:

- Developers need access to **development environments**
- Accounting staff require access to **billing information**
- Business units may want to experiment with AWS services **without impacting each other**

This can quickly lead to **AWS account sprawl**—a tangled mess of accounts with different permissions and access rules, making security, billing, and governance **hard to manage**.

Introducing AWS Organizations

AWS **Organizations** is a service that helps businesses **centrally manage multiple AWS accounts**. It simplifies governance, billing, security, and compliance across accounts.

Key Features of AWS Organizations

1. Centralized Account Management

- Instead of managing **each AWS account separately**, AWS Organizations allows you to group them into a **single organization**.
- This helps you **easily control access, apply policies, and manage billing** across multiple AWS accounts.

2. Consolidated Billing

- All AWS accounts under an organization **can be billed together**, reducing administrative overhead.
- **Cost Benefits**: AWS offers **bulk discounts** for services based on the total usage across all accounts in the organization.

3. Organizational Units (OUs)

- You can create **Organizational Units (OUs)** to **group accounts** based on their purpose.
- Example OUs:
    - **Development OU**: For developers working on test environments.
    - **Finance OU**: For teams handling AWS billing and financial reports.
    - **Compliance OU**: For teams that must follow regulatory policies.

4. Service Control Policies (SCPs)

- SCPs **restrict what AWS services and actions** can be used across different accounts.
- For example:
    - A **finance account** could be restricted to only access **billing and cost management services**.
    - A **developer account** could be allowed to use **EC2 and S3**, but **blocked from modifying security settings**.

Why Use AWS Organizations?

✅ **Simplifies Multi-Account Management** – Manage all AWS accounts from a **single control plane**.
✅ **Better Security & Compliance** – Apply **organization-wide security policies** to protect resources.
✅ **Cost Savings** – Benefit from **consolidated billing** and **AWS volume discounts**.
✅ **Granular Access Control** – Restrict services using **SCPs**, ensuring compliance and security.

Final Thoughts

AWS Organizations helps companies **stay organized, secure, and cost-efficient** as they scale their AWS usage. By grouping accounts, enforcing security policies, and managing billing centrally, it provides a structured approach to handling AWS at an enterprise level. 🚀

In AWS Organizations, you can apply service control policies (SCPs) to the organization root, an individual member account, or an OU. An SCP affects all IAM users, groups, and roles within an account, including the AWS account root user.

You can apply IAM policies to IAM users, groups, or roles. You cannot apply an IAM policy to the AWS account root user.

Compliance

Ensuring Compliance & Security in AWS

Every industry has **specific standards and regulations** that businesses must comply with. Whether it's **health inspections for coffee shops** or **tax audits for businesses**, organizations need to maintain documentation, records, and inspections to pass audits successfully.

Similarly, in AWS, businesses must **ensure compliance** with relevant regulations based on their industry, such as:

- **GDPR (General Data Protection Regulation)** – For companies handling consumer data in the EU.
- **HIPAA (Health Insurance Portability and Accountability Act)** – For healthcare applications in the US.
- **PCI-DSS (Payment Card Industry Data Security Standard)** – For businesses handling credit card transactions.

AWS Compliance Tools & Best Practices

AWS provides **infrastructure security**, but customers are responsible for ensuring that **their applications and data comply** with relevant regulations. This is part of the **AWS Shared Responsibility Model**:
✅ **AWS secures the cloud** (data centers, networking, physical security).
✅ **You secure what's in the cloud** (data, configurations, access policies).

Here's how you can ensure compliance in AWS:

1. Choosing the Right AWS Region 🌍

- Certain regulations **restrict data storage locations** (e.g., GDPR requires storing EU customer data within the EU).
- AWS **does not automatically replicate data across Regions**, so businesses must **choose Regions carefully** to meet compliance.

2. Data Security & Encryption 🔐

- **You own your data** in AWS, so you control **how it's stored and encrypted**.
- AWS provides built-in **encryption mechanisms** (S3 encryption, RDS encryption, KMS for key management).
- Enable security **configuration settings** for compliance (e.g., S3 bucket policies, IAM permissions).

3. Accessing Compliance Reports via AWS Artifact 📄

- **AWS Artifact** provides third-party **audit reports, certifications, and compliance documents**.
- Businesses can use these **official reports** to prove compliance without running their own audits on AWS infrastructure.
- The **AWS Compliance Center** consolidates compliance resources and security best practices.

4. Following AWS Security Best Practices 🛡️

- Use **AWS Security Hub** to monitor security configurations and compliance.
- Implement **AWS IAM roles and policies** for **least privilege access**.
- Enable **AWS CloudTrail** for auditing and tracking API activity.
- Use **AWS Config** to continuously monitor AWS resources against compliance standards.

Final Thoughts

AWS **helps businesses meet compliance** with industry regulations by providing secure infrastructure, compliance reports, and built-in security features. However, businesses must **architect their applications responsibly**, follow **security best practices**, and leverage AWS compliance tools to stay compliant.

📌 **Key Takeaways:**
✅ Choose AWS Regions that align with your compliance needs.

✅ Encrypt and secure your data using AWS security services.
✅ Use **AWS Artifact** for official compliance documentation.
✅ Follow AWS security best practices to meet **shared responsibility**.

Which tasks can you complete in AWS Artifact?

- *Access AWS compliance reports on-demand.*
- *Review, accept, and manage agreements with AWS.*

Denial-of-Service Attacks

Defending Against DDoS Attacks in AWS 🚨

A **Distributed Denial-of-Service (DDoS) attack** is an attempt to **overwhelm your infrastructure**, preventing real users from accessing your application. Attackers use **botnets**—armies of compromised machines—to flood your system with malicious traffic.

Common Types of DDoS Attacks

1️⃣ UDP Flood Attacks 🌊

- Attackers send **tiny UDP requests** that trigger **massive responses** from legitimate services (e.g., weather data services).
- The responses are directed to **your** server (using a spoofed return address), overloading it.

2️⃣ HTTP Flood Attacks 🏗️

- Attackers simulate **normal web requests** (e.g., search queries) from botnets at an overwhelming rate.
- The goal is to **consume resources** and slow down or crash the web server.

3️⃣ Slowloris Attacks 🐌

- Attackers open **slow, incomplete connections**, forcing your server to **wait indefinitely** for the request to complete.
- This **ties up server resources**, preventing real users from being served.

AWS DDoS Protection Strategies

1️⃣ Security Groups & Network-Level Protection 🔒

✅ **AWS Security Groups** filter out **unexpected traffic** at the AWS network level, blocking attacks like UDP floods.
✅ **AWS's massive infrastructure** absorbs brute-force attacks at a regional scale, making DDoS attacks cost-prohibitive.

2️⃣ Elastic Load Balancer (ELB) & Auto Scaling 🚀

✅ **Elastic Load Balancers (ELB) handle requests before they reach your web servers**, preventing Slowloris attacks.
✅ **Auto Scaling** ensures that even if an attack occurs, additional resources can **automatically scale** to handle the load.

3️⃣ AWS WAF & AWS Shield 🛡️

✅ **AWS WAF (Web Application Firewall)** detects and blocks **malicious HTTP requests** in real-time.
✅ **AWS Shield Standard** protects against common DDoS attacks **at no extra cost**.
✅ **AWS Shield Advanced** offers **enhanced DDoS protection** with detailed monitoring and real-time attack mitigation.

Final Takeaways 🎯

✓ **A well-architected AWS system is naturally resilient** to many DDoS attacks.
✓ **Security Groups, ELB, and Auto Scaling** offer **built-in DDoS resistance**.
✓ **AWS WAF & AWS Shield Advanced** provide **proactive DDoS defense**.

By leveraging **AWS security best practices**, you can **fortify your cloud infrastructure** against even the most sophisticated DDoS attacks. 🚀

AWS Shield Standard

AWS Shield Standard automatically protects all AWS customers at no cost. It protects your AWS resources from the most common, frequently occurring types of DDoS attacks.

As network traffic comes into your applications, AWS Shield Standard uses a variety of analysis techniques to detect malicious traffic in real time and automatically mitigates it.

AWS Shield Advanced

AWS Shield Advanced is a paid service that provides detailed attack diagnostics and the ability to detect and mitigate sophisticated DDoS attacks.

It also integrates with other services such as Amazon CloudFront, Amazon Route 53, and Elastic Load Balancing. Additionally, you can integrate AWS Shield with AWS WAF by writing custom rules to mitigate complex DDoS attacks.

Additional Security Services

AWS Key Management Service (AWS KMS)

The coffee shop has many items, such as coffee machines, pastries, money in the cash registers, and so on. You can think of these items as data. The coffee shop owners want to ensure that all of these items are secure, whether they're sitting in the storage room or being transported between shop locations.

In the same way, you must ensure that your applications' data is secure while in storage **(encryption at rest)** and while it is transmitted, known as **encryption in transit**.

AWS Key Management Service (AWS KMS)(opens in a new tab) enables you to perform encryption operations through the use of **cryptographic keys**. A cryptographic key is a random string of digits used for locking (encrypting) and unlocking (decrypting) data. You can use AWS KMS to create, manage, and use cryptographic keys. You can also control the use of keys across a wide range of services and in your applications.

With AWS KMS, you can choose the specific levels of access control that you need for your keys. For example, you can specify which IAM users and roles are able to manage keys. Alternatively, you can temporarily disable keys so that they are no longer in use by anyone. Your keys never leave AWS KMS, and you are always in control of them.

AWS WAF

AWS WAF(opens in a new tab) is a web application firewall that lets you monitor network requests that come into your web applications.

AWS WAF works together with Amazon CloudFront and an Application Load Balancer. Recall the network access control lists that you learned about in an earlier module. AWS WAF works in a similar way to block or allow traffic. However, it does this by using a **web access control list (ACL)**(opens in a new tab) to protect your AWS resources.

Here's an example of how you can use AWS WAF to allow and block specific requests.

Suppose that your application has been receiving malicious network requests from several IP addresses. You want to prevent these requests from continuing to access your application, but you also want to ensure that legitimate users can still access it. You configure the web ACL to allow all requests except those from the IP addresses that you have specified.

When a request comes into AWS WAF, it checks against the list of rules that you have configured in the web ACL. If a request does not come from one of the blocked IP addresses, it allows access to the application.

However, if a request comes from one of the blocked IP addresses that you have specified in the web ACL, AWS WAF denies access.

Securing Your Data in AWS 🔒🔐

Just like you'd lock up your coffee beans and equipment to prevent theft, your **data** also needs protection—both when it's **at rest** and **in transit**.

1️⃣ Encryption: Protecting Data Like Locking a Door 🚪🔑

🔒 **Encryption at Rest (Idle Data)**

- Think of **locking up your storeroom** at night.
- **AWS services like DynamoDB, S3, and RDS** offer **encryption at rest** by default.
- Uses **AWS Key Management Service (KMS)** to manage encryption keys.
- Without the key, unauthorized users **cannot access your data**.

🚀 **Encryption in Transit (Moving Data)**

- Similar to **transporting coffee beans securely between locations**.
- Uses **SSL/TLS connections** to encrypt data when it's moving between AWS services or from client to server.
- **AWS services like S3, Redshift, and SQS** use encryption in transit to prevent data interception.

2️⃣ Security & Threat Detection Tools 🛡️

🕵️ **Amazon Inspector: Automated Security Assessments**

- Checks for **vulnerabilities** in your AWS applications.
- Identifies **exposed EC2 instances** and deviations from security best practices.
- Generates a **report with recommended fixes** to strengthen security.

⚠️ **Amazon GuardDuty: Continuous Threat Detection**

- Monitors AWS activity for **malicious behavior and unauthorized access**.
- Uses **machine learning** and **threat intelligence** to detect suspicious activities.
- Works **independently** from your AWS services, ensuring security **without impacting performance**.

3️⃣ Additional AWS Security Services

✓ **AWS Shield Advanced** – Protects against large-scale DDoS attacks.
✓ **AWS Security Hub** – Centralizes security alerts across AWS services.

Final Takeaways 🎯

✓ **Encryption** secures your data **at rest** and **in transit**.
✓ **Amazon Inspector** finds vulnerabilities, **GuardDuty detects threats**.
✓ **AWS Security Services work together** to provide **multi-layered protection**.

AWS tools for monitoring and analytics

Monitoring: Observing systems, Collecting metrics, and then using that data to make decisions.

Amazon CloudWatch

Amazon CloudWatch

Amazon CloudWatch(opens in a new tab) is a web service that enables you to monitor and manage various metrics and configure alarm actions based on data from those metrics.

CloudWatch uses **metrics**(opens in a new tab) to represent the data points for your resources. AWS services send metrics to CloudWatch. CloudWatch then uses these metrics to create graphs automatically that show how performance has changed over time.

CloudWatch alarms

With CloudWatch, you can create **alarms**(opens in a new tab) that automatically perform actions if the value of your metric has gone above or below a predefined threshold.

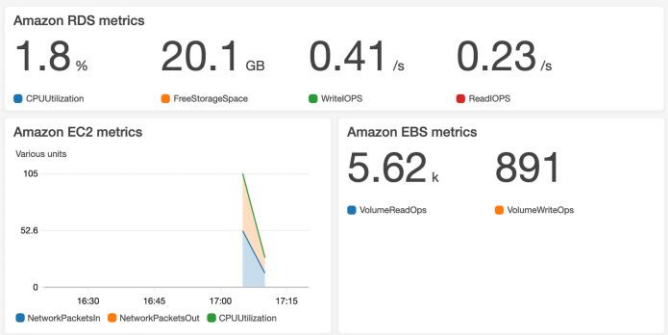
For example, suppose that your company's developers use Amazon EC2 instances for application development or testing purposes. If the developers occasionally forget to stop the instances, the instances will continue to run and incur charges.

In this scenario, you could create a CloudWatch alarm that automatically stops an Amazon EC2 instance when the CPU utilization percentage has remained below a certain threshold for a specified period. When configuring the alarm, you can specify to receive a notification whenever this alarm is triggered.

- **CloudWatch Alarms** trigger actions.
- **Amazon SNS (Simple Notification Service)** sends alerts via **SMS, email, or AWS Lambda functions**.

Result? 📢 You get notified before small issues become big problems.

CloudWatch dashboard



The CloudWatch **dashboard**(opens in a new tab) feature enables you to access all the metrics for your resources from a single location. For example, you can use a CloudWatch dashboard to monitor the CPU utilization of an Amazon EC2 instance, the total number of requests made to an Amazon S3 bucket, and more. You can even customize separate dashboards for different business purposes, applications, or resources.

Key Benefits of CloudWatch 🚀

✓ **Centralized Metrics** – View logs & metrics from all AWS services.
✓ **Full Visibility** – Monitor applications, infrastructure, and services.
✓ **Faster Issue Resolution (mean time to resolution- MTTR)** – Quickly pinpoint & fix issues.
✓ **Cost Optimization (total cost of ownership- TCO)** – Improve resource efficiency.

AWS CloudTrail

AWS CloudTrail

AWS CloudTrail(opens in a new tab) records API calls for your account. The recorded information includes the identity of the API caller, the time of the API call, the source IP address of the API caller, and more. You can think of CloudTrail as a "trail" of breadcrumbs (or a log of actions) that someone has left behind them.

Recall that you can use API calls to provision, manage, and configure your AWS resources. With CloudTrail, you can view a complete history of user activity and API calls for your applications and resources.

Events are typically updated in CloudTrail within 15 minutes after an API call. You can filter events by specifying the time and date that an API call occurred, the user who requested the action, the type of resource that was involved in the API call, and more.

Example: AWS CloudTrail event

Suppose that the coffee shop owner is browsing through the AWS Identity and Access Management (IAM) section of the AWS Management Console. They discover that a new IAM user named Mary was created, but they do not know who, when, or which method created the user.

To answer these questions, the owner navigates to AWS CloudTrail.

In the CloudTrail Event History section, the owner applies a filter to display only the events for the "CreateUser" API action in IAM. The owner locates the event for the API call that created an IAM user for Mary. This event record provides complete details about what occurred:

On January 1, 2020 at 9:00 AM, IAM user John created a new IAM user (Mary) through the AWS Management Console.

CloudTrail Insights

Within CloudTrail, you can also enable **CloudTrail Insights**(opens in a new tab). This optional feature allows CloudTrail to automatically detect unusual API activities in your AWS account.

For example, CloudTrail Insights might detect that a higher number of Amazon EC2 instances than usual have recently launched in your account. You can then review the full event details to determine which actions you need to take next.

1️⃣ Introducing AWS CloudTrail 🚀

CloudTrail = AWS's security camera 🎥

- Logs **every request** made to AWS
- Tracks **who made changes, when, from where, and what changed**
- Helps with **compliance, security, and troubleshooting**

Every API call—from **launching an EC2 instance** to **modifying IAM permissions**—gets recorded.

2️⃣ How CloudTrail Strengthens Security 🔐

👮 Example: Proving Security Compliance

- You configure a **Security Group** to block external access.
- An **auditor** wants proof that it hasn't changed.
- **CloudTrail logs confirm that no modifications were made.** ✅

- **CloudTrail Logs Include:**

✅ **Who made the request?** (User, Role, Service)
✅ **What action was performed?** (e.g., ModifySecurityGroup)
✅ **When was the change made?** (Timestamp)
✅ **Where did it originate from?** (IP Address, Region)
✅ **Was it successful or denied?** (Response Code)

3️⃣ Storing Logs Securely with S3 & Vault Lock 🔒

To ensure **audit logs remain untampered**, store them in:

- **Amazon S3** (secure, scalable storage)
- **Vault Lock** (prevents logs from being deleted or altered)

This provides **immutable proof** for security audits.

4️⃣ Why Use CloudTrail? 🌟

✓ **Automatic Logging** – No manual tracking required.
✓ **Security & Compliance** – Detect unauthorized changes.
✓ **Troubleshooting** – Pinpoint root causes of issues.
✓ **Tamper-Proof Audit Trails** – Vault Lock prevents deletion.

Final Thoughts 📜

Just like balancing a **cash register**, **CloudTrail ensures AWS activity is trackable and verifiable**. **You always know what happened, when, and by whom.**

AWS Trusted Advisor

AWS Trusted Advisor

AWS Trusted Advisor(opens in a new tab) is a web service that inspects your AWS environment and provides real-time recommendations in accordance with AWS best practices.

Trusted Advisor compares its findings to AWS best practices in five categories: cost optimization, performance, security, fault tolerance, and service limits. For the checks in each category, Trusted Advisor offers a list of recommended actions and additional resources to learn more about AWS best practices.

The guidance provided by AWS Trusted Advisor can benefit your company at all stages of deployment. For example, you can use AWS Trusted Advisor to assist you while you are creating new workflows and developing new applications. You can also use it while you are making ongoing improvements to existing applications and resources.

AWS Trusted Advisor dashboard

When you access the Trusted Advisor dashboard on the AWS Management Console, you can review completed checks for cost optimization, performance, security, fault tolerance, and service limits.

For each category:

- The green check indicates the number of items for which it detected **no problems**.
- The orange triangle represents the number of recommended **investigations**.
- The red circle represents the number of recommended **actions**.

AWS Trusted Advisor is a web service that inspects your AWS environment and provides real-time recommendations in accordance with AWS best practices. The inspection includes security checks, such as Amazon S3 buckets with open access permissions.

AWS pricing and support

AWS Free Tier

The AWS Free Tier(opens in a new tab) enables you to begin using certain services without having to worry about incurring costs for the specified period.

Three types of offers are available:

- Always Free
- 12 Months Free
- Trials

For each free tier offer, make sure to review the specific details about exactly which resource types are included.

Always Free

These offers do not expire and are available to all AWS customers.

For example, AWS Lambda allows 1 million free requests and up to 3.2 million seconds of compute time per month. Amazon DynamoDB allows 25 GB of free storage per month.

12 Months Free

These offers are free for 12 months following your initial sign-up date to AWS.

Examples include specific amounts of Amazon S3 Standard Storage, thresholds for monthly hours of Amazon EC2 compute time, and amounts of Amazon CloudFront data transfer out.

Trials

Short-term free trial offers start from the date you activate a particular service. The length of each trial might vary by number of days or the amount of usage in the service.

For example, Amazon Inspector offers a 90-day free trial. Amazon Lightsail (a service that enables you to run virtual private servers) offers 750 free hours of usage over a 30-day period.

How AWS pricing works

AWS offers a range of cloud computing services with pay-as-you-go pricing.

Pay for what you use.

For each service, you pay for exactly the amount of resources that you actually use, without requiring long-term contracts or complex licensing.

Pay less when you reserve.

Some services offer reservation options that provide a significant discount compared to On-Demand Instance pricing.

For example, suppose that your company is using Amazon EC2 instances for a workload that needs to run continuously. You might choose to run this workload on Amazon EC2 Instance Savings Plans, because the plan allows you to save up to 72% over the equivalent On-Demand Instance capacity.

Pay less with volume-based discounts when you use more.

Some services offer tiered pricing, so the per-unit cost is incrementally lower with increased usage.

For example, the more Amazon S3 storage space you use, the less you pay for it per GB.

AWS Pricing Calculator

The **AWS Pricing Calculator**(opens in a new tab) lets you explore AWS services and create an estimate for the cost of your use cases on AWS. You can organize your AWS estimates by groups that you define. A group can reflect how your company is organized, such as providing estimates by cost center.

When you have created an estimate, you can save it and generate a link to share it with others.

Suppose that your company is interested in using Amazon EC2. However, you are not yet sure which AWS Region or instance type would be the most cost-efficient for your use case. In the AWS Pricing Calculator, you can enter details, such as the kind of operating system you need, memory requirements, and input/output (I/O) requirements. By using the AWS Pricing Calculator, you can review an estimated comparison of different EC2 instance types across AWS Regions.

AWS pricing examples

This section presents a few examples of pricing in AWS services.

AWS Lambda

To learn more about AWS Lambda pricing(opens in a new tab), choose each of the following two tabs.

AWS Lambda pricing

Pricing Example

For AWS Lambda, you are charged based on the number of requests for your functions and the time that it takes for them to run.

AWS Lambda allows 1 million free requests and up to 3.2 million seconds of compute time per month.

You can save on AWS Lambda costs by signing up for a Compute Savings Plan. A Compute Savings Plan offers lower compute costs in exchange for committing to a consistent amount of usage over a 1-year or 3-year term. This is an example of **paying less when you reserve**.

Amazon EC2

To learn more about Amazon EC2 pricing(opens in a new tab), choose each of the following two tabs.

Amazon EC2 Pricing

Pricing Example

With Amazon EC2, you pay for only the compute time that you use while your instances are running.

For some workloads, you can significantly reduce Amazon EC2 costs by using Spot Instances. For example, suppose that you are running a batch processing job that is able to withstand interruptions. Using a Spot Instance would provide you with up to 90% cost savings while still meeting the availability requirements of your workload.

You can find additional cost savings for Amazon EC2 by considering Savings Plans and Reserved Instances.

Amazon S3

To learn more about Amazon S3 pricing(opens in a new tab), choose each of the following two tabs.

Amazon S3 Pricing

Pricing Example

For Amazon S3 pricing, consider the following cost components:

- **Storage -** You pay for only the storage that you use. You are charged the rate to store objects in your Amazon S3 buckets based on your objects' sizes, storage classes, and how long you have stored each object during the month.
- **Requests and data retrievals -** You pay for requests made to your Amazon S3 objects and buckets. For example, suppose that you are storing photo files in Amazon S3 buckets and hosting them on a website. Every time a visitor requests the website that includes these photo files, this counts towards requests you must pay for.
- **Data transfer -** There is no cost to transfer data between different Amazon S3 buckets or from Amazon S3 to other services within the same AWS Region. However, you pay for data that you transfer into and out of Amazon S3, with a few exceptions. There is no cost for data transferred into Amazon S3 from the internet or out to Amazon CloudFront. There is also no cost for data transferred out to an Amazon EC2 instance in the same AWS Region as the Amazon S3 bucket.
- **Management and replication -** You pay for the storage management features that you have enabled on your account's Amazon S3 buckets. These features include Amazon S3 inventory, analytics, and object tagging.

Billing Dashboard

Use the **AWS Billing & Cost Management dashboard** to pay your AWS bill, monitor your usage, and analyze and control your costs.

- Compare your current month-to-date balance with the previous month, and get a forecast of the next month based on current usage.
- View month-to-date spend by service.
- View Free Tier usage by service.
- Access Cost Explorer and create budgets.
- Purchase and manage Savings Plans.
- Publish AWS Cost and Usage Reports.

Consolidated Billing

Consolidated billing

In an earlier module, you learned about AWS Organizations, a service that enables you to manage multiple AWS accounts from a central location. AWS Organizations also provides the option for **consolidated billing**(opens in a new tab).

The consolidated billing feature of AWS Organizations enables you to receive a single bill for all AWS accounts in your organization. By consolidating, you can easily track the combined costs of all the linked accounts in your organization. The default maximum number of accounts allowed for an organization is 4, but you can contact AWS Support to increase your quota, if needed.

On your monthly bill, you can review itemized charges incurred by each account. This enables you to have greater transparency into your organization's accounts while still maintaining the convenience of receiving a single monthly bill.

Another benefit of consolidated billing is the ability to share bulk discount pricing, Savings Plans, and Reserved Instances across the accounts in your organization. For instance, one account might not have enough monthly usage to qualify for discount pricing. However, when multiple accounts are combined, their aggregated usage may result in a benefit that applies across all accounts in the organization.

AWS Budgets

AWS Budgets

In **AWS Budgets**(opens in a new tab), you can create budgets to plan your service usage, service costs, and instance reservations.

The information in AWS Budgets updates three times a day. This helps you to accurately determine how close your usage is to your budgeted amounts or to the AWS Free Tier limits.

In AWS Budgets, you can also set custom alerts when your usage exceeds (or is forecasted to exceed) the budgeted amount.

Example: AWS Budgets

Suppose that you have set a budget for Amazon EC2. You want to ensure that your company's usage of Amazon EC2 does not exceed $200 for the month.

In AWS Budgets, you could set a custom budget to notify you when your usage has reached half of this amount ($100). This setting would allow you to receive an alert and decide how you would like to proceed with your continued use of Amazon EC2.

AWS Cost Explorer

AWS Cost Explorer

AWS Cost Explorer(opens in a new tab) is a tool that lets you visualize, understand, and manage your AWS costs and usage over time.

AWS Cost Explorer includes a default report of the costs and usage for your top five cost-accruing AWS services. You can apply custom filters and groups to analyze your data. For example, you can view resource usage at the hourly level.

AWS Support Plans

AWS Support

AWS offers four different **Support plans**(opens in a new tab) to help you troubleshoot issues, lower costs, and efficiently use AWS services.

You can choose from the following Support plans to meet your company's needs:

- Basic
- Developer
- Business
- Enterprise On-Ramp
- Enterprise

Basic Support

Basic Support is free for all AWS customers. It includes access to whitepapers, documentation, and support communities. With Basic Support, you can also contact AWS for billing questions and service limit increases.

With Basic Support, you have access to a limited selection of AWS Trusted Advisor checks. Additionally, you can use the AWS Personal Health Dashboard, a tool that provides alerts and remediation guidance when AWS is experiencing events that may affect you.

If your company needs support beyond the Basic level, you could consider purchasing Developer, Business, Enterprise On-Ramp, and Enterprise Support.

AWS Support Plans Overview 📌

AWS provides **five** support plans to assist customers with **troubleshooting, cost optimization, and efficient service usage**.

1️⃣ AWS Support Plans ⏳

Support Plan	Cost 💰	Key Features 🔑
Basic 🆓	Free	Documentation, forums, billing support, limited Trusted Advisor, AWS Health Dashboard.
Developer 🛠️	Paid	Business-hour support via email, general guidance, and architectural best practices.
Business 📊	Paid	24/7 support via email, chat, and phone, AWS Trusted Advisor full checks, infrastructure event management.
Enterprise On-Ramp 🚀	Paid	Faster response times, designated technical support, and architectural reviews for growing businesses.
Enterprise 🏢	Paid	15-min response for critical issues, **Technical Account Manager (TAM)**, well-architected reviews, and concierge support.

2️⃣ Which Plan is Right for You? 🎯

✅ **Just Getting Started?** → **Basic** (Free).
✅ **Developers Testing Workloads?** → **Developer**.
✅ **Running Production Apps?** → **Business**.
✅ **Growing Business with High Availability Needs?** → **Enterprise On-Ramp**.
✅ **Large Enterprise with Mission-Critical Workloads?** → **Enterprise**.

Technical Account Manager (TAM) – Your AWS Cloud Guide 🚀

A **Technical Account Manager (TAM)** is a **dedicated AWS expert** available to customers who subscribe to **Enterprise On-Ramp** or **Enterprise Support** plans.

🔹 **What Does a TAM Do?**

✓ **Primary AWS Contact** – Your go-to AWS resource.
✓ **Cloud Strategy Guidance** – Helps **educate, empower, and evolve** your cloud journey.
✓ **Architectural Best Practices** – Assists in **resilient, cost-effective** cloud designs.
✓ **Service Integration** – Ensures multiple AWS services work seamlessly together.
✓ **Performance & Cost Optimization** – Helps reduce AWS costs while maintaining efficiency.
✓ **Exclusive AWS Access** – Connects you with **AWS experts, programs, and internal resources**.

🔹 **Example Use Case**

Scenario:
You're **building an application** that relies on multiple AWS services (e.g., **EC2, Lambda, RDS, S3**).

How a TAM Helps:
✅ Reviews **architecture** for scalability and cost-efficiency.
✅ Provides **best practices** for integrating services.
✅ Ensures compliance and security standards are met.
✅ Connects you with **specialized AWS teams** when needed.

A **TAM is your cloud partner**, ensuring your **AWS environment runs smoothly, securely, and cost-effectively**. 🔥

AWS Marketplace

AWS Marketplace

AWS Marketplace(opens in a new tab) is a digital catalog that includes thousands of software listings from independent software vendors. You can use AWS Marketplace to find, test, and buy software that runs on AWS.

For each listing in AWS Marketplace, you can access detailed information on pricing options, available support, and reviews from other AWS customers.

You can also explore software solutions by industry and use case. For example, suppose your company is in the healthcare industry. In AWS Marketplace, you can review use cases that software helps you to address, such as implementing solutions to protect patient records or using machine learning models to analyze a patient's medical history and predict possible health risks.

AWS Marketplace categories

AWS Marketplace offers products in several categories, such as Infrastructure Software, DevOps, Data Products, Professional Services, Business Applications, Machine Learning, Industries, and Internet of Things (IoT).

Within each category, you can narrow your search by browsing through product listings in subcategories. For example, subcategories in the DevOps category include areas such as Application Development, Monitoring, and Testing.

AWS Marketplace – Your Digital Software Catalog 🌐

The **AWS Marketplace** is a **curated digital catalog** that helps businesses **find, deploy, and manage** third-party software solutions within their AWS environment. It simplifies procurement, offers flexible pricing, and accelerates innovation.

🔹 **Why Use AWS Marketplace?**

✓ **One-Click Deployment** – Quickly deploy software without setting up infrastructure.
✓ **Flexible Payment Options** – Choose **on-demand, pay-as-you-go, or annual licenses**.
✓ **Free Trials & Quick Start** – Test software before committing.
✓ **Custom Pricing & Licensing** – Negotiate custom agreements with vendors.
✓ **Private Marketplace** – Maintain a **pre-approved catalog** that meets security & legal standards.
✓ **Seamless Procurement Integration** – Easily integrate into **enterprise purchasing workflows**.

🔹 **How AWS Marketplace Helps Enterprises**

✅ **Faster Deployment** – No need to build or install infrastructure.
✅ **Lower Costs** – Avoid unused software licenses with pay-as-you-go options.
✅ **Greater Control** – Customize software procurement for security and compliance.
✅ **Better Visibility** – Cost management tools help track spending efficiently.

🔹 **Example Use Case**

Scenario: Your business needs a **machine learning tool** to analyze customer data.

How AWS Marketplace Helps:
🚀 Find a **pre-vetted ML solution** from a trusted vendor.
🔧 Deploy with **one click**, skipping manual setup.
💰 Choose **on-demand pricing** to avoid upfront costs.
📊 Monitor spending with AWS cost management tools.

Migration and innovation in the AWS Cloud

Many businesses start their journey **on-premises** or with another cloud provider. AWS provides **migration tools, cost savings, and flexibility** to help you transition smoothly.

AWS Cloud Adoption Framework (AWS CAF)

AWS Cloud Adoption Framework (CAF) – A Guide to Cloud Migration 🚀

Migrating to AWS isn't as simple as flipping a switch—it requires **planning, expertise, and teamwork**. Fortunately, AWS has **captured best practices** from successful migrations and developed the **AWS Cloud Adoption Framework (CAF)** to guide organizations through this process.

🔹 **The Role of Different Teams in Migration**

Your role in the organization **affects what you need to know** about cloud migration. A **developer, cloud architect, financial analyst, HR, or business leader** will have different perspectives, and it's crucial to align them.

👷 **Developers** → Focus on application code & infrastructure changes.
👩‍💻 **Cloud Architects** → Design scalable, resilient architectures.
📊 **Business Analysts** → Ensure business value & cost optimization.
💰 **Financial Analysts** → Manage cloud spending & budgeting.
👥 **HR Teams** → Ensure the right talent is available for cloud initiatives.

🔹 **The 6 Perspectives of the AWS Cloud Adoption Framework**

CAF is structured into **six perspectives**, divided into **business-focused** and **technical-focused** groups:

📈 **Business-Focused Perspectives**

1️⃣ **Business Perspective** – Aligns IT strategies with business goals.
2️⃣ **People Perspective** – Ensures the right talent & training for cloud adoption.
3️⃣ **Governance Perspective** – Manages policies, compliance, and risk.

💻 **Technical-Focused Perspectives**

4️⃣ **Platform Perspective** – Optimizes cloud architecture & infrastructure.
5️⃣ **Security Perspective** – Ensures security, compliance, and risk management.
6️⃣ **Operations Perspective** – Manages cloud operations & incident response.

🔹 **AWS Cloud Adoption Framework (CAF) Action Plan**

🔍 Identify **gaps** in skills, processes, and tools across all **six perspectives**.
📌 Document these as **inputs** for migration planning.
🚀 Develop an **Action Plan** to ensure a smooth and efficient AWS migration.

🔹 **Why Use the AWS Cloud Adoption Framework?**

✅ **Clear Migration Strategy** – Aligns teams on goals & priorities.
✅ **Risk Mitigation** – Identifies gaps before they become issues.
✅ **Talent Readiness** – Ensures the right skills and hiring plans.
✅ **Optimized Cloud Spending** – Helps manage cloud costs effectively.

Migrating to the cloud **doesn't have to be overwhelming**. With the **AWS Cloud Adoption Framework**, your team has the structure and guidance needed to succeed.

Migration Strategies

6 strategies for migration

When migrating applications to the cloud, six of the most common migration strategies(opens in a new tab) that you can implement are:

- Rehosting
- Replatforming
- Refactoring/re-architecting
- Repurchasing
- Retaining
- Retiring

To learn more about migration stretiges, expand each of the following six categories.

Rehosting

Rehosting also known as "lift-and-shift" involves moving applications without changes.

In the scenario of a large legacy migration, in which the company is looking to implement its migration and scale quickly to meet a business case, the majority of applications are rehosted.

Replatforming

Replatforming, also known as "lift, tinker, and shift," involves making a few cloud optimizations to realize a tangible benefit. Optimization is achieved without changing the core architecture of the application.

Refactoring/re-architecting

Refactoring (also known as **re-architecting**) involves reimagining how an application is architected and developed by using cloud-native features. Refactoring is driven by a strong business need to add features, scale, or performance that would otherwise be difficult to achieve in the application's existing environment.

Repurchasing

Repurchasing involves moving from a traditional license to a software-as-a-service model.

For example, a business might choose to implement the repurchasing strategy by migrating from a customer relationship management (CRM) system to Salesforce.com.

Retaining

Retaining consists of keeping applications that are critical for the business in the source environment. This might include applications that require major refactoring before they can be migrated, or, work that can be postponed until a later time.

Retiring

Retiring is the process of removing applications that are no longer needed.

🔹 **The 6 R's of AWS Migration**

1️⃣ **Rehosting ("Lift & Shift")** – Move applications with minimal changes.
2️⃣ **Replatforming ("Lift, Tinker & Shift")** – Optimize without major architectural changes.
3️⃣ **Refactoring / Rearchitecting** – Redesign for cloud-native capabilities.
4️⃣ **Repurchasing** – Switch to **AWS-native** services like RDS, Lambda, etc.
5️⃣ **Retire** – Decommission unused workloads to save costs.
6️⃣ **Retain** – Keep certain apps on-premises (for now).

AWS Snow Family

When customers need to transfer **large volumes of data to AWS**, traditional internet-based methods are often **slow, costly, and inefficient**. Even with **AWS Direct Connect**, transferring massive datasets can take **days, weeks, or even months** due to bandwidth limitations.

To **solve this problem**, AWS introduced the **AWS Snow Family**, a suite of **physical devices** designed to **securely and efficiently transfer large amounts of data** to AWS.

AWS Snow Family members

The **AWS Snow Family**(opens in a new tab) is a collection of physical devices that help to physically transport up to exabytes of data into and out of AWS.

AWS Snow Family is composed of **AWS Snowcone**, **AWS Snowball**, and **AWS Snowmobile**.

These devices offer different capacity points, and most include built-in computing capabilities. AWS owns and manages the Snow Family devices and integrates with AWS security, monitoring, storage management, and computing capabilities.

AWS Snowcone

AWS Snowcone(opens in a new tab) is a small, rugged, and secure edge computing and data transfer device.

It features 2 CPUs, 4 GB of memory, and up to 14 TB of usable storage.

AWS Snowball

AWS Snowball(opens in a new tab) offers two types of devices:

- **Snowball Edge Storage Optimized** devices are well suited for large-scale data migrations and recurring transfer workflows, in addition to local computing with higher capacity needs.
    - Storage: 80 TB of hard disk drive (HDD) capacity for block volumes and Amazon S3 compatible object storage, and 1 TB of SATA solid state drive (SSD) for block volumes.
    - Compute: 40 vCPUs, and 80 GiB of memory to support Amazon EC2 sbe1 instances (equivalent to C5).
- **Snowball Edge Compute Optimized** provides powerful computing resources for use cases such as machine learning, full motion video analysis, analytics, and local computing stacks.
    - Storage: 80-TB usable HDD capacity for Amazon S3 compatible object storage or Amazon EBS compatible block volumes and 28 TB of usable NVMe SSD capacity for Amazon EBS compatible block volumes.
    - Compute: 104 vCPUs, 416 GiB of memory, and an optional NVIDIA Tesla V100 GPU. Devices run Amazon EC2 sbe-c and sbe-g instances, which are equivalent to C5, M5a, G3, and P3 instances.

AWS Snowmobile

AWS Snowmobile(opens in a new tab) is an exabyte-scale data transfer service used to move large amounts of data to AWS.

You can transfer up to 100 petabytes of data per Snowmobile, a 45-foot long ruggedized shipping container, pulled by a semi trailer truck.

🔹 **Security & Compliance**

🔐 **End-to-End Encryption**
All AWS Snow Family devices **encrypt data with 256-bit encryption** before transfer.

🔑 **AWS Key Management Service (KMS)**
Customers can use **AWS KMS** to **generate, manage, and control encryption keys**.

🔒 **Tamper-Resistant Hardware**
Devices are **designed to resist unauthorized access** while in transit.

🔹 **The AWS Snow Family – Overview**

The **AWS Snow Family** consists of **three key solutions**, each catering to different data transfer needs:

1️⃣ **AWS Snowcone** – For small-scale data migration (up to **8TB**).
2️⃣ **AWS Snowball Edge** – For medium-scale migration with computing capabilities.
3️⃣ **AWS Snowmobile** – For massive-scale migration (up to **100PB**).

Each device is **designed for security, portability, and efficiency**, enabling customers to move data **without relying on slow network transfers**.

🔴 **AWS Snowcone – Small but Powerful**

- 🔹 **Capacity: Up to 8TB of storage.**
- 🔹 **Edge Computing: Supports Amazon EC2 instances & AWS IoT Greengrass for on-device processing.**
- 🔹 **Use Cases:**

✅ **Backup & restore operations.**
✅ **Data collection from remote locations.**
✅ **Video & image archiving.**
✅ **IoT data aggregation.**

How It Works

1️⃣ **Order a Snowcone from the AWS Management Console.**
2️⃣ **AWS ships the device to your location.**
3️⃣ **Connect and transfer your data.**
4️⃣ **Ship it back to AWS.**
5️⃣ **AWS uploads the data to your AWS account (Amazon S3).**

🔐 **Security: All data is automatically encrypted with 256-bit encryption keys, managed by AWS Key Management Service (KMS).**

🟡 **AWS Snowball Edge – More Storage & Compute Power**

- 🔹 **Two Versions:**

✅ **Snowball Edge Compute Optimized – More computing power for data processing on-site.**
✅ **Snowball Edge Storage Optimized – More storage for larger data transfers.**

- 🔹 **Storage: Supports tens of terabytes of data.**
- 🔹 **Computing Capabilities:**

✅ **Runs AWS Lambda functions.**
✅ **Supports Amazon EC2-compatible AMIs.**
✅ **Integrates AWS IoT Greengrass for real-time data processing.**

Use Cases

- 🔹 **Industrial applications (IoT, machine learning).**
- 🔹 **Video transcoding & image compression.**
- 🔹 **Data transfer from remote sites with limited connectivity.**

How It Works

1️⃣ **Order the device via the AWS console.**
2️⃣ **AWS ships it to your location.**
3️⃣ **Integrate with your existing infrastructure (rack-mountable).**

4️⃣ Process & store data locally before transferring to AWS.
5️⃣ Return the device to AWS for secure data upload.

🟢 AWS Snowmobile – The Ultimate Data Transfer Solution

- 🔹 Capacity: Transfers up to 100PB in a 45-foot rugged shipping container.
- 🔹 Physical Transport: Delivered via truck, appearing as a network-attached storage device.
- 🔹 Security Features:

✅ Tamper-resistant, waterproof, & temperature-controlled.
✅ Fire suppression system.
✅ GPS tracking & 24/7 video surveillance.
✅ Dedicated security team with escort vehicles.

Use Cases

- 🔹 Large-scale data center migrations.
- 🔹 Massive data transfers for government, finance, and healthcare.
- 🔹 Enterprise backups for disaster recovery.

How It Works

1️⃣ AWS drives the Snowmobile to your data center.
2️⃣ Connect and transfer your data (as if using a massive storage device).
3️⃣ AWS transports the Snowmobile back to its data centers.
4️⃣ AWS uploads the data to your AWS account.

Exploring More AWS Innovations 🚀

AWS offers far more than we can cover in a single discussion. Whether it's cloud migration, AI & machine learning, IoT, or even space technology, AWS provides solutions for businesses to innovate at scale.

🤖 AI & Machine Learning: AWS ML Services

AWS provides the most extensive suite of AI & ML tools, making machine learning accessible to all businesses.

1️⃣ Amazon SageMaker – Build & Deploy ML Models

✅ Train and deploy custom ML models at scale.
✅ Supports all major open-source ML frameworks.
✅ Managed infrastructure optimized for high-performance ML.

2️⃣ Pre-trained AI Services

AWS offers ready-to-use AI for various business needs:

- 🔹 Computer Vision – Amazon Rekognition (face/object detection).
- 🔹 Language Processing – Amazon Comprehend (sentiment analysis).
- 🔹 Forecasting – Amazon Forecast (predictive analytics).

3️⃣ Amazon Augmented AI (A2I)

✅ Human-in-the-loop ML models for better accuracy.
✅ Ideal for sensitive tasks like document processing & content moderation.

4️⃣ AWS DeepRacer – Hands-on ML Training 🏎️

AWS DeepRacer lets developers:

✅ Experiment with reinforcement learning (RL).
✅ Train and test autonomous racing models in a fun environment.

👉 Use Case: Great for businesses wanting to train teams in ML without needing PhD-level expertise.

💬 AI-Powered Chatbots: Amazon Lex (Alexa's Core AI)

- 🔹 Build interactive voice & text chatbots for customer support.
- 🔹 Natural language processing (NLP) for intelligent conversations.
- 🔹 Power virtual assistants like Alexa.

👉 Use Case: Companies can use Amazon Lex to automate customer interactions with AI chatbots.

📄 Document AI: Amazon Textract

Extract text & data from scanned documents with AI.

✅ Converts PDFs/images into editable text.
✅ Reads tables & forms (unlike traditional OCR).
✅ Automates document processing & compliance workflows.

👉 Use Case: Banks, healthcare, and legal firms use Textract to digitize and analyze documents efficiently.

🌎 Internet of Things (IoT) with AWS

AWS enables connected devices to communicate globally:

✅ AWS IoT Core – Securely connect IoT devices.
✅ AWS IoT Greengrass – Run ML at the edge (without internet).
✅ AWS IoT Analytics – Process massive IoT data streams.

👉 Use Case: Smart homes, industrial automation, & real-time data monitoring.

🚀 AWS Ground Station – Satellite Connectivity

Always wanted a satellite but found it too expensive? 🛰️

✅ AWS Ground Station lets you rent satellite time instead of owning one.
✅ Download satellite data on demand (e.g., weather, mapping, or surveillance).
✅ No expensive infrastructure needed.

👉 Use Case: Governments, research institutions, & global communications companies use AWS Ground Station for cost-effective satellite operations.

Overview of Amazon Q Developer

Amazon Q Developer is an AI-powered tool that integrates with your IDE to analyze comments and surrounding code, offering real-time code generation and security vulnerability detection. It helps developers by automating repetitive tasks, adhering to security best practices, and improving productivity and code quality.

Key Features:

1. Code Generation & Natural Language Processing: Generates code based on English comments and surrounding code, adhering to style and conventions.
2. Security Scanning: Scans code for vulnerabilities (OWASP, AWS standards, crypto best practices) and offers remediation suggestions.
3. AI-Driven Code Suggestions: Provides code completion and generation options, optimizing developer time.
4. Open-Source Reference Tracker: Ensures proper attribution for code samples derived from open-source projects.
5. Automated Security Updates: Keeps codebase secure by automatically scanning for new vulnerabilities.
6. Integration with IDEs: Works with VS Code, JetBrains, and supports languages like Python, Java, and JavaScript.

The Security Perspective of the AWS Cloud Adoption Framework also helps you to identify areas on non-compliance and plan ongoing security initiatives.

The other response options are incorrect because:

- ***The Governance Perspective helps you to identify and implement best practices for IT governance and support business processes with technology.***
- ***The Operations Perspective focuses on operating and recovering IT workloads to meet the requirements of your business stakeholders.***
- ***The Business Perspective helps you to move from a model that separates business and IT strategies into a business model that integrates IT strategy.***

AWS Well-Architected Framework and benefits of the AWS Cloud.

Overview of Well-Architected Framework:

The AWS Well-Architected Framework helps evaluate the quality of your cloud architectures across five key pillars: Operational Excellence, Security, Reliability, Performance Efficiency, and Cost Optimization. It provides guidance for building robust, secure, and cost-effective systems on AWS.

The 5 Pillars of the Well-Architected Framework:

1. **Operational Excellence**: Focuses on operations in the cloud, such as monitoring, incident response, and optimizing procedures to ensure continuous improvement.
2. **Security**: Ensures data protection, access control, and security best practices are implemented to safeguard the system and maintain privacy.
3. **Reliability**: Ensures the system can recover from failures and meet business and customer expectations for uptime and availability.
4. **Performance Efficiency**: Optimizes cloud resource usage, leveraging AWS's scalability and flexibility to meet performance goals effectively.
5. **Cost Optimization**: Focuses on controlling costs through efficient use of resources, such as right-sizing and eliminating waste.

Operational excellence

Operational excellence is the ability to run and monitor systems to deliver business value and to continually improve supporting processes and procedures.

Design principles for operational excellence in the cloud include performing operations as code, annotating documentation, anticipating failure, and frequently making small, reversible changes.

Security

The **Security** pillar is the ability to protect information, systems, and assets while delivering business value through risk assessments and mitigation strategies.

When considering the security of your architecture, apply these best practices:

- Automate security best practices when possible.
- Apply security at all layers.
- Protect data in transit and at rest.

Reliability

Reliability is the ability of a system to do the following:

- Recover from infrastructure or service disruptions
- Dynamically acquire computing resources to meet demand
- Mitigate disruptions such as misconfigurations or transient network issues

Reliability includes testing recovery procedures, scaling horizontally to increase aggregate system availability, and automatically recovering from failure.

Performance efficiency

Performance efficiency is the ability to use computing resources efficiently to meet system requirements and to maintain that efficiency as demand changes and technologies evolve.

Evaluating the performance efficiency of your architecture includes experimenting more often, using serverless architectures, and designing systems to be able to go global in minutes.

Cost optimization

Cost optimization is the ability to run systems to deliver business value at the lowest price point.

Cost optimization includes adopting a consumption model, analyzing and attributing expenditure, and using managed services to reduce the cost of ownership.

Sustainability

In December 2021, AWS introduced a sustainability pillar as part of the AWS Well-Architected Framework.

Sustainability is the ability to continually improve sustainability impacts by reducing energy consumption and increasing efficiency across all components of a workload by maximizing the benefits from the provisioned resources and minimizing the total resources required.

To facilitate good design for sustainability:

- Understand your impact
- Establish sustainability goals
- Maximize utilization
- Anticipate and adopt new, more efficient hardware and software offerings
- Use managed services
- Reduce the downstream impact of your cloud workloads

The Operational Excellence pillar includes the ability to run workloads effectively, gain insights into their operations, and continuously improve supporting processes to deliver business value.

The Performance Efficiency pillar focuses on using computing resources efficiently to meet system requirements, and to maintain that efficiency as demand changes and technologies evolve.

The Security pillar includes protecting data, systems, and assets, and using cloud technologies to improve the security of your workloads.

AWS Well-Architected Tool:

The Well-Architected Tool is a self-service tool available in the AWS Management Console, designed to help you assess and improve your workloads. The tool runs checks against the framework's pillars, generates reports, and highlights areas for improvement using a traffic light system:

- **Green: Good, no action needed.**
- **Orange: A concern, needs attention.**
- **Red: High-risk area, immediate action required.**

The tool helps you stay on top of best practices, guides architectural changes, and is customizable to your unique requirements.

Advantages of cloud computing

Advantages of cloud computing

Operating in the AWS Cloud offers many benefits over computing in on-premises or hybrid environments.

In this section, you will learn about six advantages of cloud computing:

- **Trade upfront expense for variable expense.**
- **Benefit from massive economies of scale.**
- **Stop guessing capacity.**
- **Increase speed and agility.**
- **Stop spending money running and maintaining data centers.**
- **Go global in minutes.**

Six Main Benefits of Using AWS:

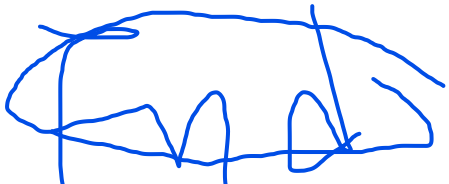
1. **Pay-as-You-Go Pricing: Unlike traditional data centers that require large upfront investments, AWS allows you to pay only for what you use, which means lower initial costs and flexibility in managing resources based on demand.**
2. **Massive Economies of Scale: AWS benefits from its large-scale operations and purchasing power, allowing you to leverage lower costs for hardware and infrastructure than if you ran your own data center.**
3. **Stop Guessing Capacity: AWS allows you to provision only the resources you need and scale them up or down automatically based on actual usage, preventing over- or under-provisioning that can occur with traditional data centers.**
4. **Increase Speed and Agility: AWS enables rapid experimentation and innovation by allowing you to quickly deploy and decommission resources, which speeds up development and testing cycles without the high cost of failed experiments.**
5. **No Need to Maintain Data Centers: AWS handles the undifferentiated heavy lifting of running and maintaining data centers, allowing you to focus on core business operations rather than hardware management.**
6. **Global Expansion in Minutes: AWS makes it easy to expand into new regions globally, often within minutes, without needing to establish physical data centers, thanks to its globally distributed infrastructure and tools like AWS CloudFormation.**

Which pillar of the AWS Well-Architected Framework includes the ability to run workloads effectively and gain insights into their operations?

The correct response option is Operational Excellence.

The other response options are incorrect because:

- **The Cost Optimization pillar focuses on the ability to run systems to deliver business value at the lowest price point.**
- **The Performance Efficiency pillar focuses on using computing resources efficiently to meet system requirements and to maintain that efficiency as demand changes and technologies evolve.**
- **The Reliability pillar focuses on the ability of a workload to consistently and correctly perform its intended functions.**



📘 Exam Domains

The exam is divided into four main domains:

Domain	Weight
1. Cloud Concepts	24%
2. Security and Compliance	30%
3. Technology	34%
4. Billing and Pricing	12%

📚 Study Plan (2-4 Weeks)

Week 1: Cloud Concepts

- Learn benefits of cloud computing: elasticity, scalability, agility.
- Understand IaaS, PaaS, SaaS models.
- Familiarize yourself with AWS Global Infrastructure: Regions, Availability Zones, Edge Locations.

Week 2: Security and Compliance

- Shared Responsibility Model.
- IAM (Users, Groups, Roles, Policies).
- Compliance programs (HIPAA, GDPR, etc.)
- AWS Organizations, SCPs, and security tools like AWS Shield, WAF.

Week 3: Core AWS Services (Technology)

- Compute: EC2, Lambda, Elastic Beanstalk
- Storage: S3, EBS, Glacier
- Networking: VPC, Route 53, CloudFront
- Databases: RDS, DynamoDB
- Deployment & Monitoring: CloudWatch, CloudTrail, Trusted Advisor

Week 4: Billing and Support

- Understand pricing models (On-Demand, Reserved, Spot).
- Use AWS Pricing Calculator and TCO tool.
- Understand support plans (Basic, Developer, Business, Enterprise).
- Cost Explorer and Budgets.

AWS Elastic Beanstalk

✅ What is AWS Elastic Beanstalk? *(Simple Definition)*

AWS Elastic Beanstalk lets you **run your web apps without managing servers**. You just upload your code, and AWS does the rest—like setting up servers, scaling, and monitoring.

📖 In Simple Words

Normally, to run a website or app, you'd need to:

- Set up servers (EC2)
- Install software
- Handle traffic changes

- Monitor performance

With Elastic Beanstalk, you just **upload your app**, and it **automatically**:

- Creates the servers
- Sets up networking and load balancing
- Scales when traffic goes up
- Monitors your app

You stay in control, but AWS handles the heavy lifting.

☕ **Coffee Shop Analogy**

You made a new coffee recipe and want to open stores.

- **Doing it yourself (EC2)**: Rent space, buy machines, hire staff.
- **Using Elastic Beanstalk**: Hand over your recipe, and AWS opens stores, hires staff, and manages everything for you.

You focus on your recipe (your app), AWS runs the shop (the app environment).

🌎 **4. Real-World AWS Example**

Imagine:

You're a startup that builds a **job posting website** using **Node.js**. You want it online fast, but you don't have time to learn how to manually launch EC2, configure networking, set up logs, etc.

You just:

1. Zip your app code
2. Go to Elastic Beanstalk in AWS Console
3. Upload your code and choose "Node.js"
4. Click "Create Application"

In a few minutes:

- AWS sets up the environment
- Launches EC2 servers
- Adds a load balancer
- Enables auto-scaling
- Provides a URL to access your site

Now you're live on the internet **with no deep cloud knowledge** needed.

✅ **Key Points to Remember**

Feature	**What It Does**
Fully managed	AWS takes care of servers, scaling, and more
Supports multiple languages	Java, .NET, Python, Node.js, Ruby, PHP, etc.
Automatic scaling	Handles traffic spikes
Monitoring built-in	Via CloudWatch
Free to use	You pay only for the AWS resources it uses (like EC2, S3, etc.)

AWS Config

🧠 **1. Simple Definition**

AWS Config is a service that **tracks and records changes** to your AWS resources and checks if they follow rules you set (called **compliance rules**).

Think of it like a **security camera and auditor** for your AWS environment.

📖 **2. Detailed Explanation (Easy Words)**

When you're using AWS, you have lots of resources like:

- EC2 instances (servers)
- S3 buckets (storage)
- IAM roles (permissions)
- Security groups (firewall rules)

These can **change often**, sometimes **by accident** or without you knowing:

- Someone opens public access to a private S3 bucket
- A security rule allows traffic from all IPs (not secure)
- A user is given full admin permissions

AWS Config does two things:

1. **Tracks changes** – It records when something is created, deleted, or changed.
2. **Checks compliance** – It compares those changes against rules you define. If something breaks a rule, it alerts you.

It gives you:

- A timeline of changes (like a version history)
- A snapshot of the current settings
- A way to **audit and prove** your setup is secure (great for security & compliance teams)

☕ **3. Coffee Shop Analogy**

📍 **Scenario:**

You run a **coffee shop franchise** with many locations. You want all stores to:

- Open at 8:00 AM
- Use a standard recipe
- Keep the cash register locked after hours

You hire a **regional manager** to:

1. **Visit each store daily**
2. **Check if they follow the rules**
3. **Write reports** on any violations

AWS Config is like that manager:

- It checks every AWS "store" (resource)
- Watches for **changes** in behavior or settings
- Alerts you if **rules are broken**

Example: "Store #7 opened at 7:30 AM instead of 8:00."
That's like an S3 bucket being made public when it shouldn't be.

🌍 **4. Real-World AWS Example**

Let's say your company has a rule:

"No S3 bucket should allow public access."

You create a **compliance rule in AWS Config** for that.

Then:

- DevOps team accidentally sets a bucket to public
- AWS Config immediately detects the change
- It marks the bucket as **non-compliant**
- Sends an alert or logs the issue for audit

You can:

- See **who made the change**
- When it happened
- What the configuration was **before and after**

This helps:

- Catch security risks early
- Prove to auditors you follow best practices
- Stay safe and compliant without checking things manually

✅ **Key Points to Remember**

Feature	What It Does
Tracks changes	Records changes to AWS resources over time
Compliance checks	Alerts if resources don't follow your rules
Audit history	View past settings and who made changes
Supports automation	Can trigger alerts or auto-remediation
Useful for security	Helps meet regulatory requirements like HIPAA, GDPR, etc.

🔁 **Example Compliance Rules You Can Create**

Rule	Description
S3 bucket should not be public	Prevent data leaks
EC2 instances must be in a specific region	For cost or regulation control
IAM users must not have admin rights	Security best practice

AWS Maice

🧠 **What is AWS Macie?**

AWS Macie is a security tool that **automatically finds sensitive data** in your S3 buckets, like credit card numbers, passwords, names, or personal info. It helps you keep that data safe and private.

📖 **Easy Explanation**

Sometimes you store files in S3, not realizing they contain **private or personal info**. AWS Macie scans your files, looks for sensitive content, and warns you if it finds something risky — especially if the bucket is public. It uses **machine learning** to recognize patterns like social security numbers or emails.

☕ **Example (Coffee Shop Style)**

Imagine you run a coffee shop and collect customer feedback forms. Some customers write their names and phone numbers. You keep these forms in a box, but one day a staff member accidentally leaves the box on the sidewalk where anyone can read it.

Macie is like a privacy checker who opens the box, reads the forms, and says:

"Hey! This one has personal info and the box is in public view — please move it somewhere safe."

That's how AWS Macie helps with your **S3 data privacy**.

✅ **Key Points to Remember**

- 🔍 **Finds sensitive data** like names, emails, credit cards in S3
- ⚠️ **Alerts you** if that data is exposed (e.g. in a public bucket)
- 🧠 Uses **machine learning** to spot private info
- ✅ Helps meet **compliance** standards like GDPR or HIPAA
- 📦 Focuses only on **Amazon S3**
-

AWS GuardDuty

🧠 **What is it?**

GuardDuty is a **threat detection service** that watches your AWS environment for **suspicious or malicious activity** — like a security guard keeping an eye on your building.

📖 **Easy Explanation**

GuardDuty constantly **monitors your AWS accounts, logs, and network traffic**. If something unusual happens — like someone trying to hack into your server or accessing it from a strange country — GuardDuty alerts you right away.

☕ **Example (Coffee Shop Style)**

You run a coffee shop, and you have security cameras at the doors. One night, someone tries to break in through a window. The camera detects movement and sends you a phone alert. That's what GuardDuty does for your AWS resources — it spots threats **before they become big problems**.

✅ **Key Points to Remember**

- 👀 Watches for **suspicious activity** in AWS (like hacking or data theft)
- 🔁 Monitors logs (like CloudTrail, VPC Flow Logs, DNS)
- ⚠️ Sends **security alerts** when it finds threats
- 💡 No setup needed — just **turn it on**, and it starts working

---

AWS Inspector

🧠 **What is it?**

AWS Inspector is a **security assessment tool**. It scans your EC2 instances or containers and checks for **vulnerabilities** — like missing updates or weak settings.

📖 **Easy Explanation**

Inspector looks at the software running on your servers. It checks things like:
"Is this app missing a security patch?"
"Is this server open to the internet without protection?"

Then it gives you a **report** with issues and how to fix them.

☕ **Example (Coffee Shop Style)**

In your coffee shop, an inspector comes in to check if the **kitchen equipment is safe** and up to code. They tell you, "This oven needs maintenance" or "This fridge is leaking." AWS Inspector does the same for your cloud servers — it helps you fix weaknesses **before hackers find them**.

✅ **Key Points to Remember**

- 🛠️ Scans EC2 instances and containers for **security flaws**
- 📋 Gives **detailed reports** with fix recommendations
- 🔐 Helps meet **compliance and security standards**
- ⚙️ Works automatically on a schedule or when you deploy new code

AWS Detective

🕵️ **What is AWS Detective?**

AWS Detective is a tool that helps you **investigate and understand security issues** in your AWS environment. It **analyzes logs and shows you what happened, when, and how** — like a digital detective solving a mystery.

📖 **Easy Explanation**

Let's say AWS GuardDuty alerts you to suspicious activity — like someone logging into your account from another country. That's helpful, but now you want to know:
"How did they get in? What did they do? What resources were affected?"

This is where **AWS Detective** helps.

It **gathers log data** (like CloudTrail, VPC flow logs, GuardDuty findings) and **creates a visual timeline** of events. You can click through and **see the full story** of what happened, which helps security teams investigate and fix the problem faster.

☕ **Example (Coffee Shop Style)**

Imagine someone broke into your coffee shop at night. The next day, you get an alert (like GuardDuty). You now want to investigate:

- How did they get in?
- What time did it happen?
- What did they touch or steal?

You check your **camera footage, keycard logs, and delivery records** — and piece everything together.

AWS Detective does this for your cloud environment. It gathers all the clues and shows you a clear picture of the security incident.

✅ **Key Points to Remember**

- 🧩 Helps **investigate security issues**
- 🔎 Analyzes data from **CloudTrail**, **GuardDuty**, **VPC Flow Logs**, etc.
- 📊 Provides a **visual timeline and graphs** to understand what happened
- 🛠️ Useful **after a threat is detected**, to find the root cause
- 🤝 Works with **GuardDuty**, **Inspector**, and other security tools

🌍 AWS Ground Station

✅ **What it does (Easy):**

AWS Ground Station lets you **communicate with satellites** directly through AWS. You can download satellite data and process it in the cloud.

🛠️ **How it helps:**

No need to build your own satellite ground stations. You use AWS to schedule access, receive data, and analyze it in real time.

📌 **Key Points:**

- Works with **satellite data**
- No need to maintain **ground station hardware**
- Integrates with AWS services like S3 and EC2

👥 AWS Cognito

✅ **What it does (Easy):**

AWS Cognito helps you **add sign-up, login, and user management** to your app securely.

🛠️ **How it helps:**

You don't have to build your own authentication system. Cognito handles things like user registration, login, and multi-factor authentication (MFA).

📌 **Key Points:**

- Handles **user sign-up and sign-in**
- Supports **social logins** (Google, Facebook, etc.)
- Scales automatically for millions of users

🌐 AWS Global Accelerator

✅ **What it does (Easy):**

It **speeds up** the access to your app for users **around the world** by routing traffic through the AWS global network.

🛠️ **How it helps:**

It improves **performance and availability** by directing users to the nearest healthy AWS endpoint.

📌 **Key Points:**

- Uses **AWS global network** (not the public internet)
- Improves **latency and performance**
- Automatically reroutes traffic if something fails

☁️ AWS CloudStart

✅ **What it does (Easy):**

AWS CloudStart is a program (not a technical service) to help governments quickly adopt cloud by providing resources, training, and support.

🛠️ **How it helps:**

Helps public sector orgs move to the cloud **faster and safely**.

📌 **Key Points:**

- Meant for **governments and public sector**
- Provides **cloud training and adoption support**
- Helps with **cloud readiness**

👨‍💻 AWS Developer Center

✅ **What it does (Easy):**

It's a **hub for developers** — with tutorials, SDKs, tools, and sample code to build with AWS.

🛠️ **How it helps:**

Makes it easier for developers to get started with AWS services using code.

📌 **Key Points:**

- Central place for **developer resources**
- Contains **tools, docs, and sample projects**
- Supports **multiple languages and SDKs**

🛠️ AWS OpsWorks

✅ **What it does (Easy):**

AWS OpsWorks is a **configuration management** tool — it helps you **automate server setup** using Chef or Puppet.

🛠️ **How it helps:**

You can **define how servers should be set up** (like what software to install) and OpsWorks applies those settings automatically.

📌 **Key Points:**

- Uses **Chef/Puppet** automation
- Good for **automating complex server setups**
- Alternative to AWS Systems Manager for config

🚚 AWS Migration Hub *(with analogy)*

✅ **What it does (Easy):**

Migration Hub is a **central dashboard** that tracks and manages your application migrations from on-prem to AWS.

🛠️ **How it helps:**

It keeps all your migration tools and progress **in one place**, even if you're using different AWS services or partner tools.

🧉 **Analogy:**

Imagine moving a coffee shop chain to a new city. Migration Hub is like your **moving checklist** that tracks what's packed, what's shipped, and what's left.

📌 **Key Points:**

- **Tracks** migrations across services
- **Central dashboard** for visibility
- Supports **discovery, planning, and tracking**

💾 Amazon FSx *(with analogy)*

✅ **What it does (Easy):**

Amazon FSx gives you **fully managed file systems** in the cloud, like Windows File Server or Lustre (for high performance).

🛠️ **How it helps:**

Instead of running your own file server, you get a **cloud-based one that's fast, scalable, and supports standard file protocols**.

🧉 **Analogy:**

Imagine having a shared folder in your office for all employees. FSx is like putting that folder **in the cloud**, and it's managed by AWS.

📌 **Key Points:**

- Supports **Windows File Server**, **Lustre**, and others
- Fully managed and **high-performance**
- Good for **shared storage** needs

🚀 AWS Launch Wizard

✅ **What it does (Easy):**

Launch Wizard helps you **quickly deploy enterprise applications** like Microsoft SQL Server or SAP on AWS — step by step.

🛠️ **How it helps:**

It automates the setup so you don't need to manually configure servers, storage, and networking for complex apps.

📌 **Key Points:**

- Simplifies **enterprise app deployment**
- Supports **SAP, SQL Server, and more**
- **Step-by-step setup wizard**

⚡ AWS DataSync

✅ **What it does (Easy):**

AWS DataSync **moves large amounts of data** quickly between **on-premises storage and AWS services like S3 or EFS**.

🛠️ **How it helps:**

You can sync your data **fast, securely, and automatically**, instead of copying manually.

📌 **Key Points:**

- **Transfers data** between on-prem and cloud
- Works with **S3, EFS, FSx**
- Supports **scheduled and automated** transfers

🗂️ Amazon S3 File Gateway

✅ **What it does (Easy):**

It lets you **access Amazon S3 like a normal file server** from your on-premises system.

🛠️ **How it helps:**

Your local apps can store and retrieve files from S3 **using standard file protocols** (like NFS/SMB), without needing to change.

📌 **Key Points:**

- Part of **AWS Storage Gateway**
- Connects **on-prem apps to S3**
- Works like a **local file share backed by the cloud**

🌐 AWS Transit Gateway

✅ **What it does (Easy):**

Transit Gateway connects **multiple VPCs and on-prem networks** through a **central hub**, like a router for your AWS networks.

🛠️ **How it helps:**

Instead of creating lots of direct connections between networks (called VPC peering), you connect all to **one place** — the Transit Gateway.

📌 **Key Points:**

- Central hub for **network connectivity**
- Scales better than VPC peering
- Supports **on-prem + multiple VPCs**

🛠️ AWS Cloud Development Kit (CDK)

✅ **What it does (Easy):**

AWS CDK lets you **write code to define your cloud infrastructure** instead of manually clicking in the AWS Console. You use familiar programming languages like Python, JavaScript, or Java to build your AWS resources.

🛠️ **How it helps:**

Instead of setting up servers, databases, or networks piece-by-piece in the console, you write code that describes what you want. Then CDK **automatically creates those resources for you** — making infrastructure easier to manage, version, and reuse.

📌 **Key Points for Exam:**

- Infrastructure as code (IaC) tool
- Supports multiple programming languages (Python, JavaScript, TypeScript, Java, C#)
- Converts your code into AWS CloudFormation templates
- Helps automate and simplify AWS resource creation
- Makes infrastructure **repeatable, testable, and version-controlled**

🌟 AWS Well-Architected Framework & Benefits of AWS Cloud

1️⃣What is the AWS Well-Architected Framework?

It's a **set of best practices and guidelines** that help you design and operate reliable, secure, efficient, and cost-effective systems in the cloud.

Think of it like an architect's blueprint — but for cloud infrastructure. It ensures you build things the right way, avoid mistakes, and can improve your setup over time.

AWS designed this framework around **6 pillars** — each focusing on a key area for a successful cloud design.

🏛️ **The 6 Pillars of the AWS Well-Architected Framework**

---

1. Operational Excellence

Easy meaning: Run your systems smoothly and improve them over time.

In detail: This pillar is about managing your infrastructure and applications well — monitoring, deploying updates, and learning from failures. It encourages you to automate as much as possible and make processes repeatable.

Analogy: Imagine running a coffee shop. Operational excellence means having a clear process for opening/closing, cleaning machines, taking inventory, and quickly fixing problems when the espresso machine breaks. You keep improving your daily routines to serve customers better.

With AWS Well-Architected Tool: You use the tool to check if you have good monitoring, automated deployments, and incident response plans.

---

2. Security

Easy meaning: Protect your data, systems, and people.

In detail: This pillar focuses on protecting your information, controlling access, detecting and responding to threats, and following best practices like encryption and strong authentication.

Analogy: In your coffee shop, security is like locking doors, having cameras, training staff on safety, and protecting customer payment info. You also check who can enter the store and restrict access to sensitive areas.

With AWS Well-Architected Tool: You check your encryption, IAM roles (who can access what), logging, and threat detection setups.

---

3. Reliability

Easy meaning: Make sure your system works well and recovers quickly if something breaks.

In detail: This pillar is about designing systems to avoid failure, handle errors gracefully, and recover fast from problems. It involves backups, failover plans, and testing.

Analogy: Your coffee shop has a backup espresso machine in case one breaks, and a plan to get it fixed fast so customers aren't waiting. If the power goes out, you might have a generator.

With AWS Well-Architected Tool: You assess your backups, failover mechanisms, and how your system reacts to failures.

---

4. Performance Efficiency

Easy meaning: Use resources smartly to give the best experience.

In detail: This pillar is about choosing the right resources, scaling when needed, and using technologies efficiently to maintain good speed and performance.

Analogy: You make sure your coffee shop has enough baristas during rush hour and the right size espresso machines so customers don't wait long. You optimize the layout for quick service.

With AWS Well-Architected Tool: You check if you're using scalable infrastructure, right-sizing resources, and monitoring performance metrics.

---

5. Cost Optimization

Easy meaning: Spend money wisely and avoid waste.

In detail: This pillar helps you understand and control your cloud spending, by only paying for what you need, and choosing cost-effective resources.

Analogy: You avoid buying more coffee beans or cups than needed, negotiate good supplier deals, and turn off machines when the shop is closed to save electricity.

With AWS Well-Architected Tool: You analyze your costs, identify unused resources, and get suggestions on cheaper alternatives.

---

6. Sustainability

Easy meaning: Build in a way that reduces environmental impact.

In detail: This new pillar encourages designing systems that use energy efficiently and reduce carbon footprints.

Analogy: Your coffee shop uses energy-efficient appliances, recycles waste, and sources eco-friendly products.

With AWS Well-Architected Tool: You evaluate your resource use and get recommendations to lower environmental impact.

🔧 **AWS Well-Architected Tool**

AWS provides this **online tool** that helps you **review your workloads against the 6 pillars**. It asks questions, gives a report on your strengths and weaknesses, and recommends improvements — helping you build better systems.

🌈 **6 Main Benefits of AWS Cloud**

---

1. Flexibility & Scalability

You can quickly scale resources up or down based on demand. No more buying expensive hardware that sits idle.

---

2. Cost Savings

Pay only for what you use, avoid upfront costs, and benefit from AWS's low prices and pricing options.

---

3. Security

AWS offers strong security features — data encryption, identity management, compliance certifications — to keep your data safe.

---

4. Global Reach

AWS has data centers all over the world, so you can serve users globally with low latency.

---

5. High Availability & Reliability

AWS infrastructure is designed for fault tolerance and disaster recovery, so your apps stay up and running.

6. Innovation Speed

With AWS managed services, you can experiment and deploy new ideas faster, without worrying about managing infrastructure.

AWS Shield

What it does (Easy):

AWS Shield is a service that **protects your websites and applications from DDoS attacks** (Distributed Denial of Service), which are attempts to overwhelm your system with traffic to make it unavailable.

How it helps:

It automatically detects and **blocks malicious traffic** before it can harm your resources, keeping your services available to legitimate users.

AWS Shield Standard

What it does (Easy):

This is the **basic, free version of AWS Shield** that protects all AWS customers from the most common and frequently seen DDoS attacks automatically.

How it helps:

You get automatic protection without extra setup or cost, guarding your AWS services like CloudFront and Elastic Load Balancers from simple DDoS attacks.

Other Important AWS Security Services

1. AWS WAF (Web Application Firewall)

- Protects your web apps from common web exploits like SQL injection or cross-site scripting by filtering HTTP(S) requests.
- Helps create custom security rules to block bad traffic.

2. AWS IAM (Identity and Access Management)

- Manages who can access your AWS resources and what they can do.
- Lets you create users, roles, and permissions for secure access control.

3. AWS GuardDuty

- Continuously monitors your AWS accounts for suspicious activity or threats and sends alerts.
- Detects unauthorized behavior or unusual patterns.

4. AWS Inspector

- Scans your EC2 instances for vulnerabilities and security issues and gives you a detailed report.
- Helps you fix weaknesses before attackers exploit them.

5. AWS Macie

- Uses machine learning to discover and protect sensitive data like personal information stored in S3.
- Alerts you if sensitive data is exposed.

✅ **Key Points to Remember for the Exam**

- **AWS Shield** protects against **DDoS attacks**.
- **Shield Standard** is **free and automatic** for all AWS customers, protecting against common DDoS threats.
- **AWS WAF** protects web apps by filtering HTTP requests with custom rules.
- **IAM** controls user access and permissions securely.
- **GuardDuty** detects suspicious activity across your AWS environment.

- **Inspector** scans EC2 instances for vulnerabilities.
- **Macie** discovers and protects sensitive data in S3.

⏰ AWS CloudWatch Alarms

What it does (Easy):

CloudWatch Alarms **monitor your AWS resources** (like servers or databases) and **send alerts** when something is wrong or crosses a threshold, like CPU usage being too high.

How it helps:

You can react quickly if your app is overloaded or if costs are rising, by getting notified via email, SMS, or automatic actions (like shutting down a server).

Key points:

- Monitors **metrics** (CPU, disk, network)
- Sends **alerts** based on thresholds
- Can trigger **automated actions**

💰 AWS Budgets

What it does (Easy):

AWS Budgets lets you **set custom cost or usage budgets** and notifies you if you exceed or are forecasted to exceed your budget.

How it helps:

Keeps your AWS spending under control by sending alerts before costs get too high.

Key points:

- Set budgets for **cost, usage, or reservations**
- Sends **notifications** via email or SNS
- Helps with **cost management**

📊 AWS Cost Explorer

What it does (Easy):

Cost Explorer helps you **visualize and analyze your AWS spending** over time with reports and graphs.

How it helps:

Shows you where your money is going, so you can find cost-saving opportunities.

Key points:

- Visualizes **historical costs and usage**
- Creates **custom reports**
- Helps identify **cost trends and waste**

Other Important AWS Services

1. AWS CloudTrail

- Tracks and records all AWS API calls for auditing and security.
- Helps you know “who did what and when.”

2. AWS Lambda

- Lets you run code without managing servers — pay only for compute time used.
- Useful for event-driven, scalable apps.

3. Amazon RDS

- Managed relational database service (MySQL, PostgreSQL, etc.)
- Takes care of backups, patching, and scaling.

4. Amazon S3

- Scalable object storage for any type of data (files, images, backups).
- Durable and highly available.

✅ **Key Points to Remember for the Exam**

- **CloudWatch Alarms:** Alerts when resource metrics cross thresholds.
- **AWS Budgets:** Set and track spending limits with notifications.
- **Cost Explorer:** Visualize and analyze your AWS spending.
- **CloudTrail:** Records API calls for auditing.
- **Lambda:** Serverless computing for running code on demand.
- **RDS:** Managed databases with automated maintenance.
- **S3:** Durable object storage for data and backups.

☁️ AWS Cloud Adoption Framework (CAF)

1️⃣ Why do we need the AWS Cloud Adoption Framework?

Moving to the cloud is a big change for any company — it's not just about technology, but also people, processes, and business goals. Without a clear plan, migrations can be chaotic, expensive, and risky.

AWS CAF is like a **guidebook** that helps organizations **plan and manage their cloud journey** successfully. It makes sure the move to AWS happens smoothly by addressing every important area, not just tech.

2️⃣ How is it done and where is it useful?

AWS CAF breaks down the migration into **clear focus areas** (called perspectives) that cover business, technical, and organizational parts. Teams work together to identify gaps, skills needed, and plan actions.

It's useful for:

- Planning **cloud strategy**
- Preparing teams for changes
- Managing risks and security
- Designing technical solutions
- Aligning cloud adoption with business goals

3️⃣ What problems does AWS CAF solve?

- Avoids missing important migration steps
- Helps manage **people and process changes**
- Ensures **security and compliance** are maintained
- Improves **communication between teams**
- Reduces risks and unexpected costs
- Accelerates successful cloud adoption

---

4. Roles of Different Teams in Migration

Migrating to cloud involves different teams, each with specific roles:

- **Business team:** Defines goals, budgets, and ensures cloud adoption aligns with business strategy. They champion cloud benefits and manage change across the organization.
- **Security team:** Ensures compliance, data privacy, identity management, and sets security policies.
- **Operations team:** Prepares infrastructure management, monitoring, backups, and incident response in the cloud.
- **Application team:** Updates or re-builds applications to work in the cloud environment.
- **People/team management:** Handles training, change management, and helps staff adapt to new cloud skills and workflows.
- **Finance team:** Tracks costs, plans budgets, and forecasts cloud spending.

---

5. The 6 Perspectives of AWS Cloud Adoption Framework

AWS CAF organizes its approach into 6 perspectives — each focusing on key capabilities to prepare for cloud success.

---

1. Business Perspective

- Focus: Business strategy, governance, and value realization.
- Goal: Align cloud adoption with business outcomes like faster innovation, cost savings, or market growth.
- Example: A retail company wants to launch online sales faster; the business team drives cloud use to meet that goal.

---

2. People Perspective

- Focus: Organization's culture, skills, and change management.
- Goal: Prepare and train staff for cloud, manage shifts in team roles.
- Example: Training IT staff to manage AWS instead of on-prem servers, ensuring smooth transition.

---

3. Governance Perspective

- Focus: Risk management, compliance, policies, and budget controls.
- Goal: Set up rules and guardrails to control cloud use and spending safely.
- Example: Finance and security teams working together to set budget alerts and data privacy policies.

---

4. Platform Perspective

- Focus: Designing, building, and operating cloud infrastructure and services.
- Goal: Build a scalable, secure, and efficient AWS environment.
- Example: Architects designing VPCs, storage, compute resources optimized for workload needs.

---

5. Security Perspective

- Focus: Protecting data, systems, and compliance with regulations.
- Goal: Implement security best practices in cloud environment.
- Example: Applying encryption, IAM policies, and continuous threat monitoring with GuardDuty.

---

6. Operations Perspective

- Focus: Day-to-day operations, monitoring, incident response, and reliability.
- Goal: Ensure smooth running of cloud applications with automated monitoring and quick issue resolution.
- Example: Using CloudWatch alarms to detect and fix problems before customers notice.

6️⃣ The 6 Migration Strategies (6 R's)

When moving applications to the cloud, AWS suggests 6 common approaches called the **6 R's**:

---

1. Rehost ("Lift and Shift")

- Move applications as-is from on-premises to AWS without changes.
- Fast and simple but may miss cloud optimization.
- Example: Moving a website server from a physical data center to an EC2 instance.

---

2. Replatform ("Lift, Tinker, and Shift")

- Move apps with minor changes to use some cloud features.
- Example: Moving a database to Amazon RDS instead of running it on an EC2 server.

---

3. Refactor / Re-architect

- Redesign applications to fully leverage cloud capabilities like auto-scaling, serverless, microservices.
- More time and cost upfront but better performance and scalability.
- Example: Breaking a monolithic app into microservices running on AWS Lambda and ECS.

---

4. Repurchase

- Replace your current app with a new cloud-based version or SaaS solution.
- Example: Moving from a self-hosted CRM to Salesforce cloud service.

---

5. Retire

- Identify applications that are no longer needed and decommission them.
- Saves costs and reduces complexity.

---

6. Retain

- Keep some apps on-premises temporarily or permanently due to compliance, latency, or other reasons.
- Plan to revisit migration later.

🌟 **Real-World Analogy: Coffee Shop Chain Moving Online**

Imagine a coffee shop chain expanding to an online delivery system:

- **Business team** sets the goal: "Reach more customers online in 6 months."
- **People team** trains baristas and staff on new online order systems.
- **Governance team** creates rules on online payments and customer data privacy.
- **Platform team** builds the website and delivery app infrastructure on AWS.
- **Security team** ensures all customer info is safe and payments are secure.

- **Operations team** monitors the website 24/7 and fixes issues fast.

The shop chooses to **lift and shift** their current website (Rehost) while planning to **refactor** their app later to add new features and scalability.

✅ **Summary & Key Points for Exam:**

- **AWS CAF** guides organizations to adopt cloud successfully across people, process, and tech.
- It solves business, cultural, security, and operational challenges in cloud adoption.
- **Teams:** Business, People, Governance, Platform, Security, Operations all have roles.
- **6 Perspectives:** Business, People, Governance, Platform, Security, Operations.
- **6 R's (Migration strategies):** Rehost, Replatform, Refactor, Repurchase, Retire, Retain.