What is a cloud?

-> It's an extensive network of remote servers, globally.

-> These servers used to store, run applications and manage data over the internet.

-> cloud is not a physical entity.

-> It enable on-demand access to a shared pool of configurable resources includes rapidly elasticity, resource pooling and broad n/w access.

-> There are various deployment models, including public, private and hybrid clouds.

Benifits of cloud:

* Cost efficiency
* Scalability
* Enhanced security.
* Instead of owning and maintaining physical servers and datacenters, users access these resources on-demand from cloud providers like AWS, GCP, MS Azure, etc.

Key characteristics of cloud computing :-

- → On-Demand self-Service
- → Broad network access
- → Resource pooling
- → Rapid Elasticity
- → Measured Services.

Types of Cloud Computing :-

- → Public Cloud
- → Private cloud
- → Hybrid cloud.

Cloud Service Models :-

- → IaaS
- → PaaS
- → SaaS

Delivery of computing services, such as storage, DB, networking, software and analytics over the internet is called cloud computing.

Public Cloud :-

→ A public cloud is one of the cloud computing model where services & resources are made available to the general public over the internet.

→ These services are hosted by a third-party providers such as AWS, GCP, MS Azure & users can access them on a pay-as-you-go basis.

→ They offers scalability, cost efficiency and ability to share resources among multiple users.

→ Suitable for businesses requiring flexibility & minimal managment.

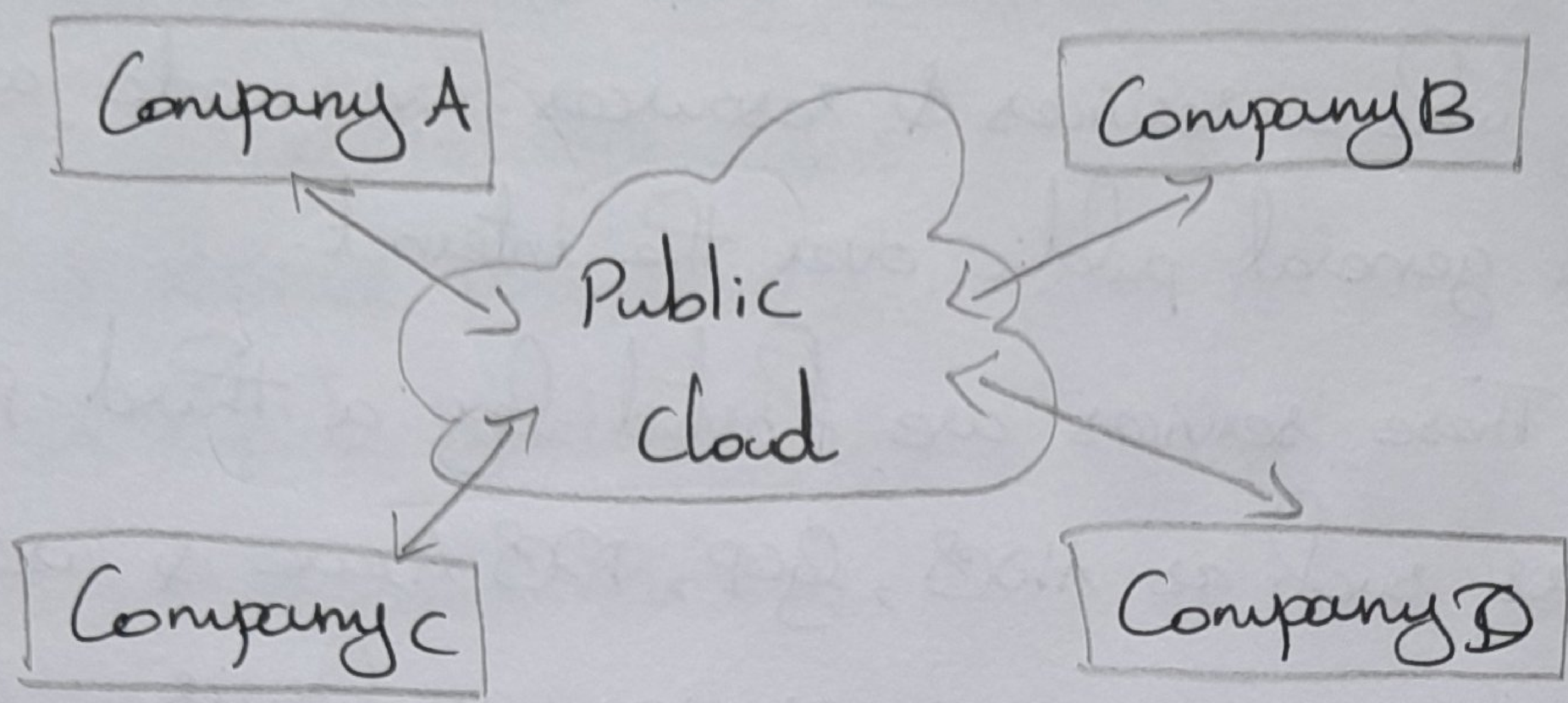
Benefits of public cloud :-

Cost Efficiency → No need to invest in physical hardware. you pay only for the resources you use

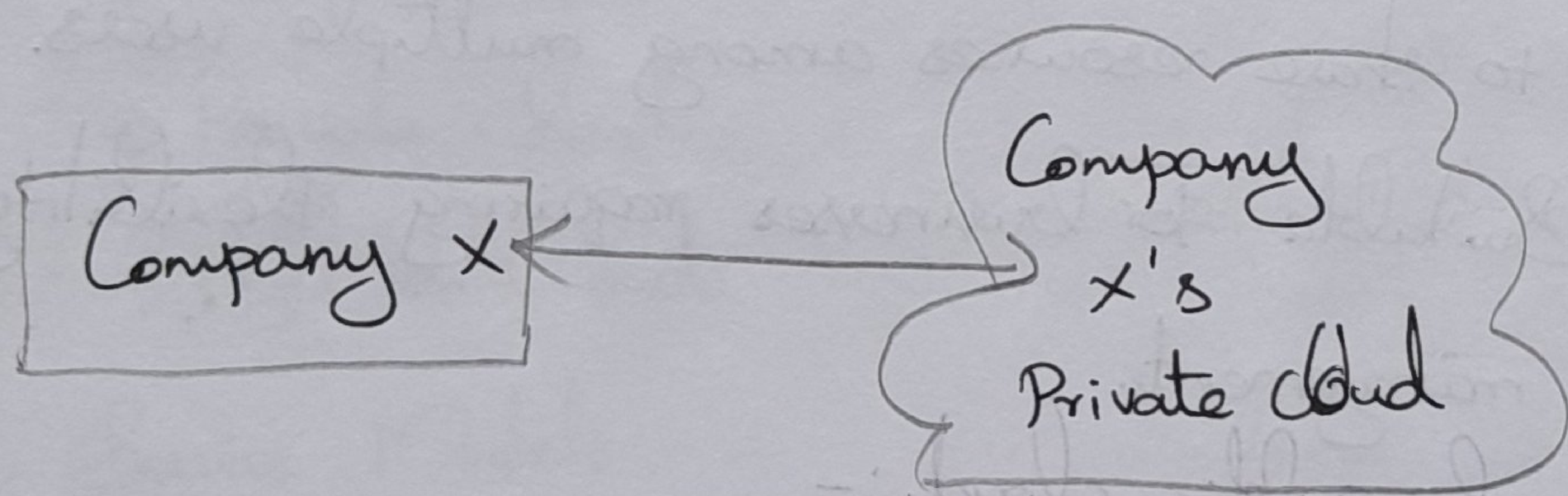
Scalability & Flexibility } → PCC provides offers on-demand resour that can help scale up & scale down based on users needs.

Reliability & Availability } → Cloud providers have multiple data centers across various regions, offeri high availability & fault tolerance thr LB & redundancy.

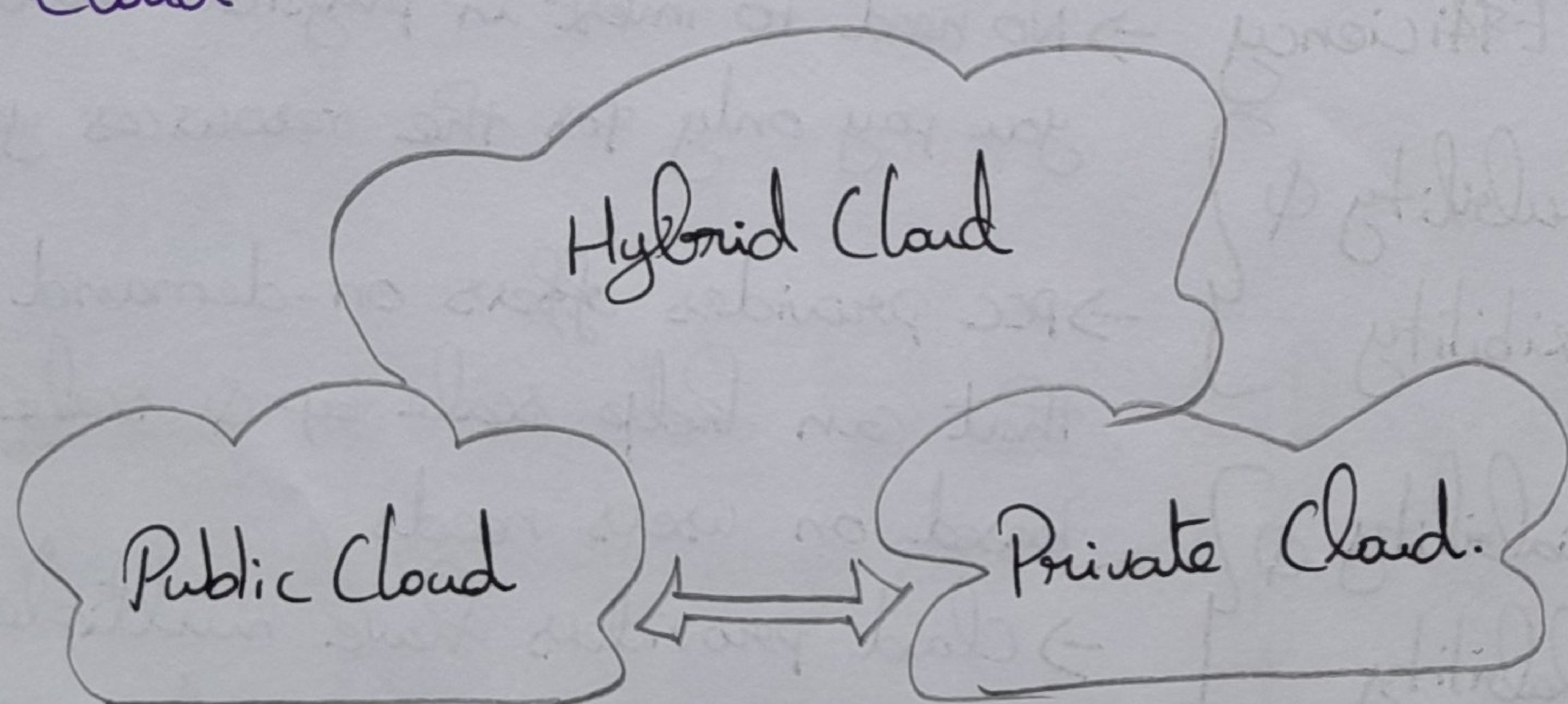
Public Cloud



Private Cloud



Hybrid Cloud



Security → Public cloud providers provides high level security, strong data encryption, firewalls & compliance with global security standards. (ISO 27001, HIPAA, GDPR)

Global Reach → Provides global n/w of data centers, improving performance & reducing latency.

Managed services → provides a wide range of managed services like DB, ML, AI & serverless computing which reducing the need for business to manage and maintain infra.

Innovation & Updates → Public cloud platforms regularly update & improve their services.

Limitations :-

Data privacy & Compliance → Strong & sensitive data in a public cloud can raise privacy, compliance issues.

Less control over Infrastructure → In public CC you relay on the provider for infrastructure management, limiting customization.

Latency Issues → n/w latency can affect real-time performance for distributed application & bussinesses.

Cost Overruns → Though pay-as-you-go model is cost effect without proper monitoring on resources will affect the cost-efficiency.

Limited customization → Designed for general-purpose & may not offer the level of customization needed for highly specialized.

Private Cloud :-

-> A private cloud is a dedicated cloud infrastructure exclusively used by one-organization, either hosted on-prem or by a third-party cloud provider.

-> It offers more control, customization & security.

-> Ideal for organizations with strict compliance or security co requirements. (Ex: Healthcare, finance)

-> However, it requiress higher upfront investment and ongoing maintenance compared to public cloud solutions.

On-Premises :-

-> Refers to servers, storage & networking hardware maknaged and hosted by an organization within its own facilities.

-> It provides full control over data, security and hardware but requires significant capital investment, mainterance & in-house IT expertise.

-> Many organization continue using on-prem infrastructure for legacy applications or compliance-driven workloads while leveraging hybrid cloud models for flexibility.

Cloud Repatriation :-

→ Cloud repatriation is the process of moving/migrating workloads or applications from the public cloud back to an on-premises or private cloud environment.

→ This occurs due to cost considerations, security concerns, performance requirements or the need for more control over data.

→ Repatriation is usually seen in organizations with matured infrastructure teams.

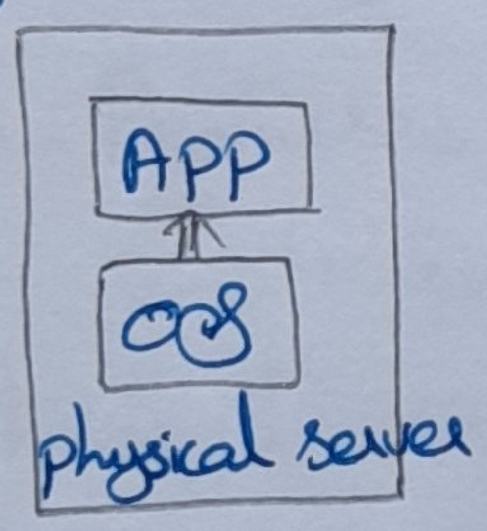
A Server configuration which includes,

→ RAM → Storage → OS → CPU → Processor

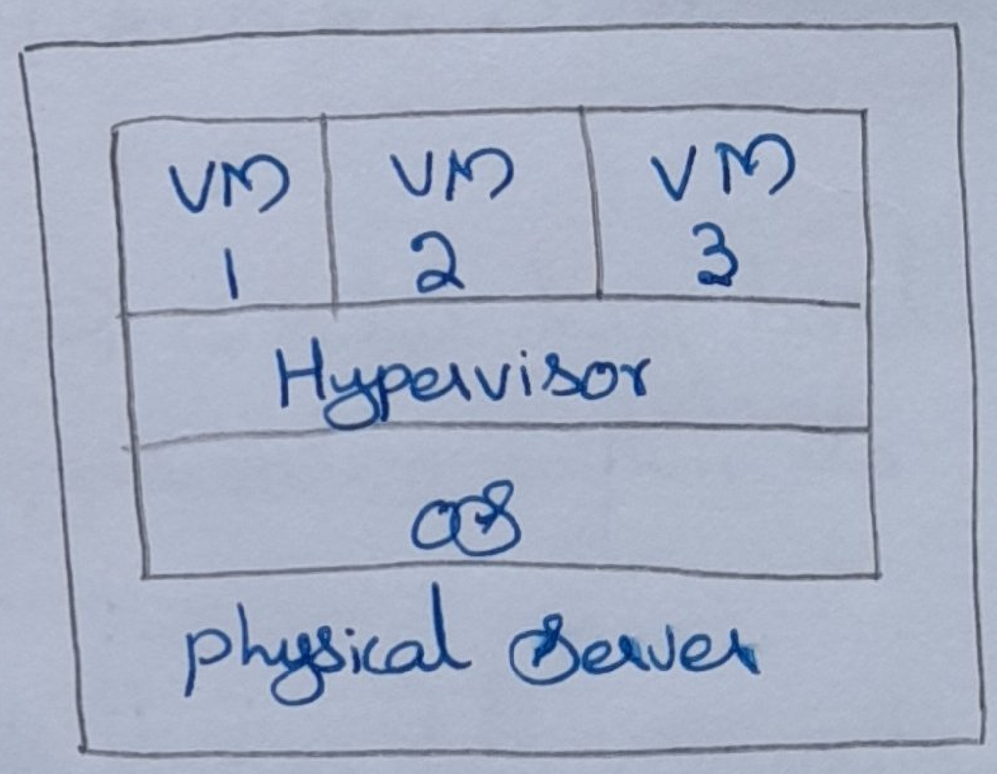
Computer ⇒ Commonly Operated Machine
Particularly Used for Technology
Education & Research.

Model 1 → Used before 20 years.

Physical Server



Model 2



→ Hypervisor divides the resources of the physical server b/w the virtual machines.

→ VMware, Openstack, XEN are commonly known as Hypervisor providers.

→ Server maintainance is a very tough job in model-2

Requirements of a Data Centre :-

* Security
* Patching & Wiring
* Seamless Internet
* Temperature
* Networking
* Power supply for 24X7.

Hypervisor :-

-> It is a s/w that creates and manages VM's on a host system.

-> Allowing multiple OSs runs concurrently on a single physical server.

-> Acts as a layer b/w hardware & VMs ensuring resources allocation & isolation b/w them.

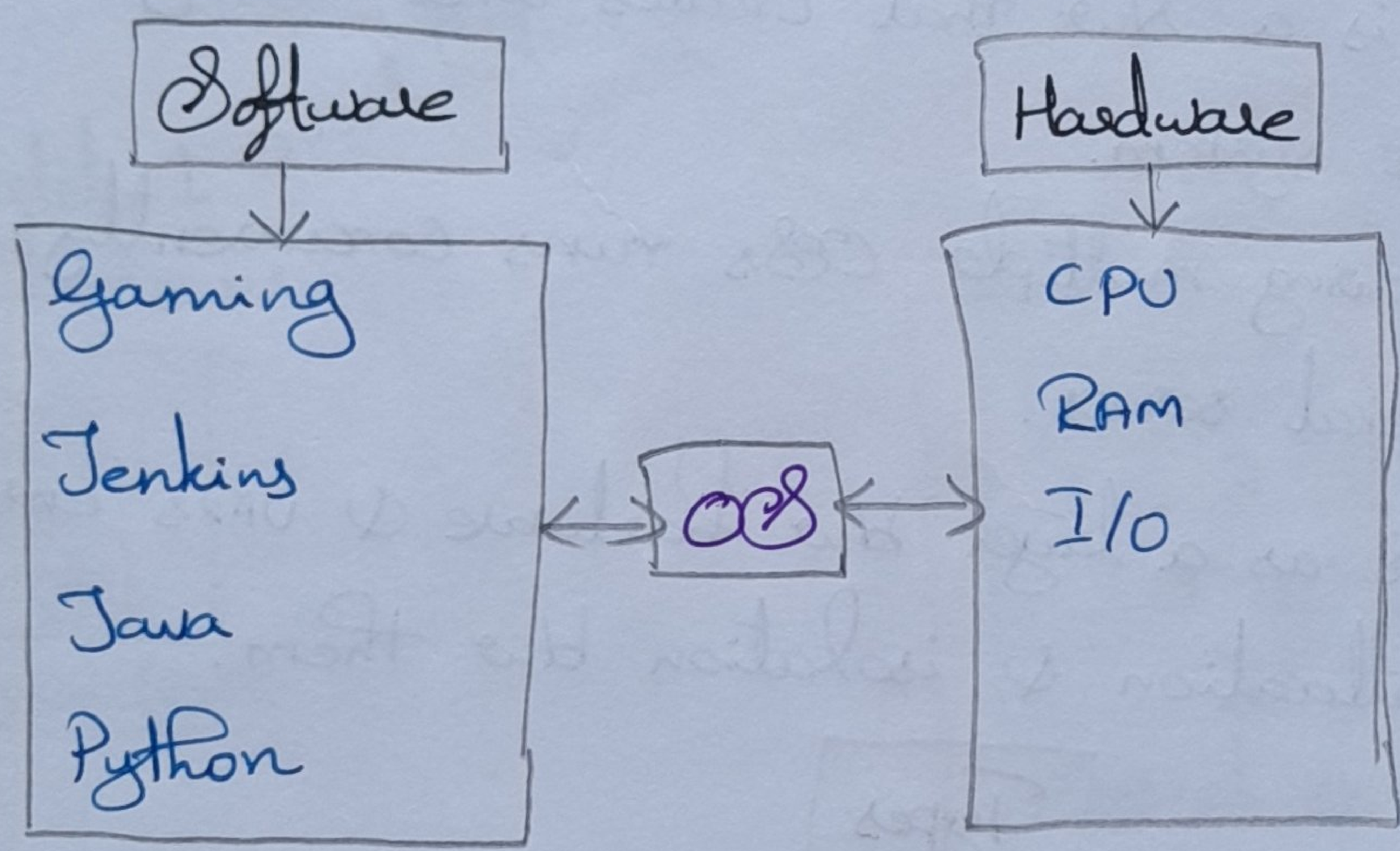
Types

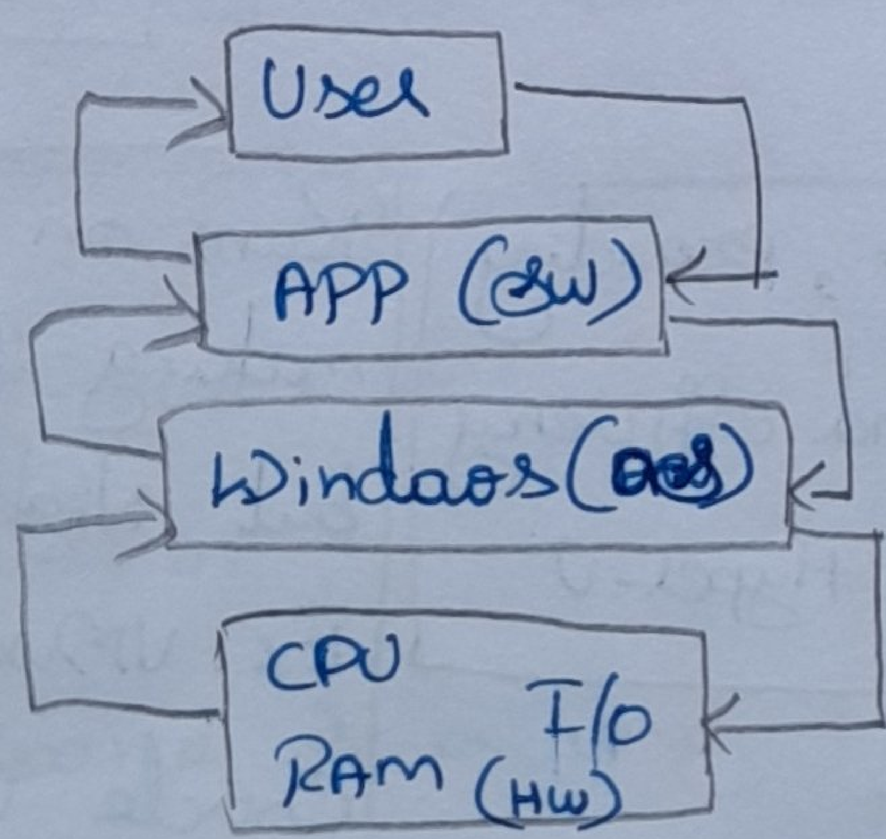
- Base-Metal HP
  - Runs directly on h/w, providing better performance and efficiency
  - Ex. VMware ESX, MS Hyper-V
- Hosted HP
  - Runs on top of existing OS making it easier to setup but slightly less efficient.
  - Ex. VMware workspace & Oracle virtual box.

Operating System:-

17-09-2024

An Operating System is the software that manages a computer's hardware and software resources and allows a user to interact with the computer.





Cascading:-

→ Cascading in an OS refers to a chain of failures or terminations triggerd by one initial failure.

Why Linux is most used in the production system? IQ*

i) Linux is free to use

ii) It's open source

iii) We can modify the features of Linux as needed. (Ex: Ubuntu, Alpine, CentOS, Fedora, Redhat)

iv) These are called distributions. (Flavours)

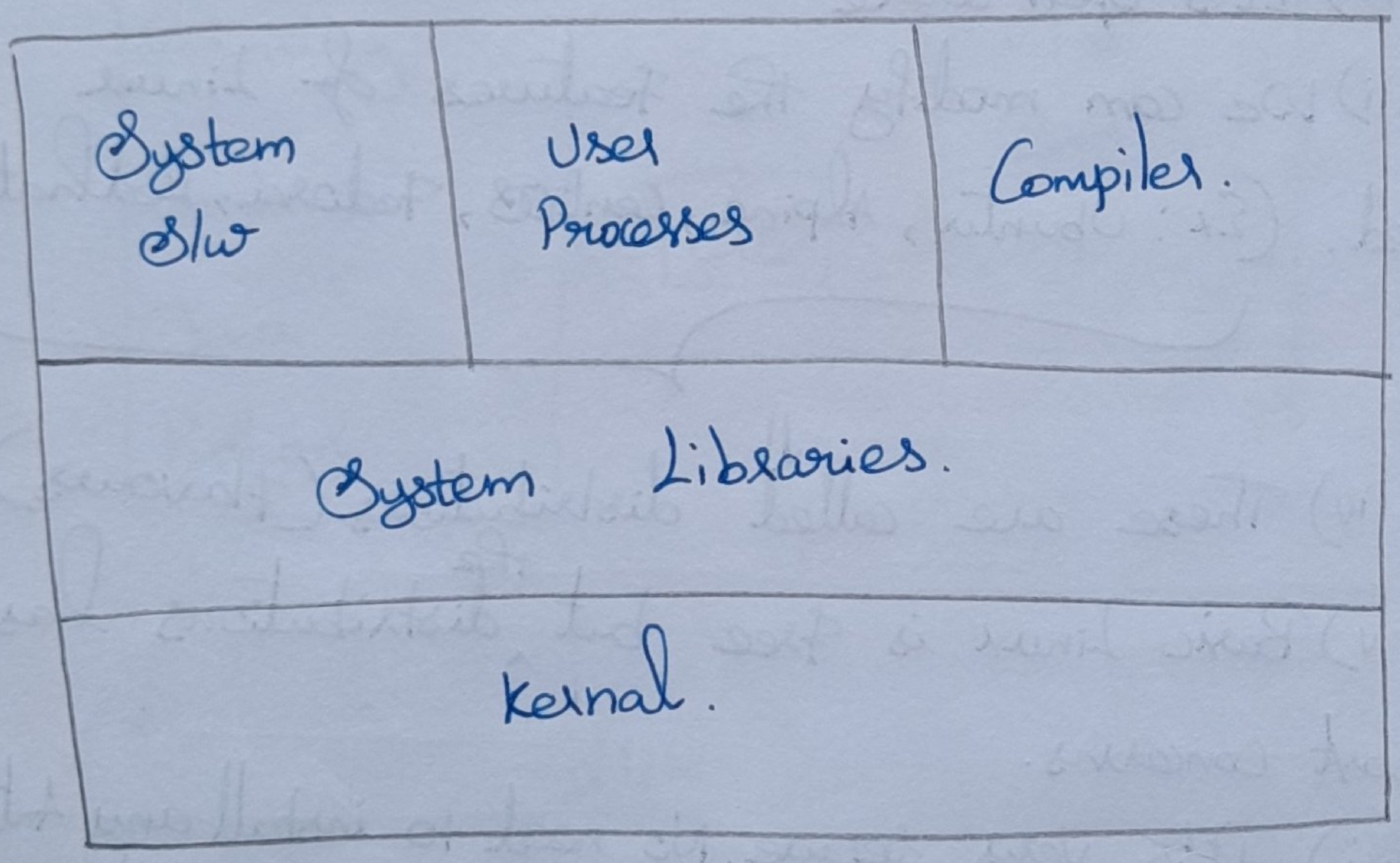
v) Basic Linux is free but the distributions has some cost concerns.

vi) It's very secure. No need to install any Anti virus (If we are using windows OS, we need to install some antiviruses to protect the system & OS from malwares)
Ex:- Mcafe, NQ...

vii) It's very fast.
(Lite weighted rather than windows. Windows using GUI)

Laptops	OS
DELL	Windows
Apple	MacOS
ThinkPad (IBM)	Linux.

OS Architecture :-



Kernal :-

→ It acts as a bridge between applications and the hardware, ensuring secure and efficient execution of processes.

→ Kernel in a OS which establishes the communication between s/w & h/w.

Other Responsibilities of Kernel:-

i) Device Management
ii) Memory Management
iii) Process Management
iv) Handling System Calls.

System Libraries:-

→ Library is a collection.

→ System libraries are collections of pre-written code that provide essential functions for an OS.

→ Ex:- libC

Compiler:-

→ A compiler is a program that translates code written in a high-level programming languages such as C, C++ or Java into machine code or bytecode. (0&1)

User Processes & System S/w:-

→ Processes created by the user are User processes

→ Gaming s/w, Application comes under System s/w

Why we are using Linux in the production system?

-> Linux uses CLI (Command Line Interface)

-> For every actions, it's enough to give a command in Linux.

-> Ex:- Create a file => touch file,
Delete a file => rm file,

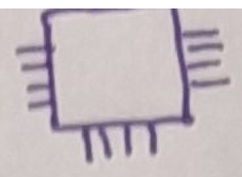
-> So that, Linux is lite weighted.

-> we can add permissions for a sensitive file/data. (Using the command <chmod>) in Linux.

-> Whereas, windows using GUI (Graphical User Interface) for every actions.

-> Comparing to windows, Linux is more secure.

-> Linux commands are commonly called shell scripting.

 => EC_2 Instance

EC_2 - Elastic Compute Cloud

-> Virtual Server or Virtual Machine provided by AWS.

-> Server is used to serve (or) deploy applications.

-> AWS EC_2 provides on-demand, scalable computing capacity in the AWS cloud.

-> CPU, RAM, Disk } => Compute

-> Scale up, Scale down } => Elastic

-> We should specify the region to launch an EC_2.

-> Latency will be very less in choosing and launching the server in a same region / nearby region. (Ex: NV client, Indian Service provider)

-> Indian company launch the server which hold the app in NV region.

Region :-

→ A region is a physical geographic area that contains multiple data centers known as Availability Zones.

→ Each region is isolated to ensure fault tolerance & data sovereignty.

Fault tolerance is the ability ~~to~~ of a system to continue operating correctly even if one or more components fail.
Redundancy is the practice of duplicating critical components or systems so that, in the event of a failure, a backup can take over seamlessly.

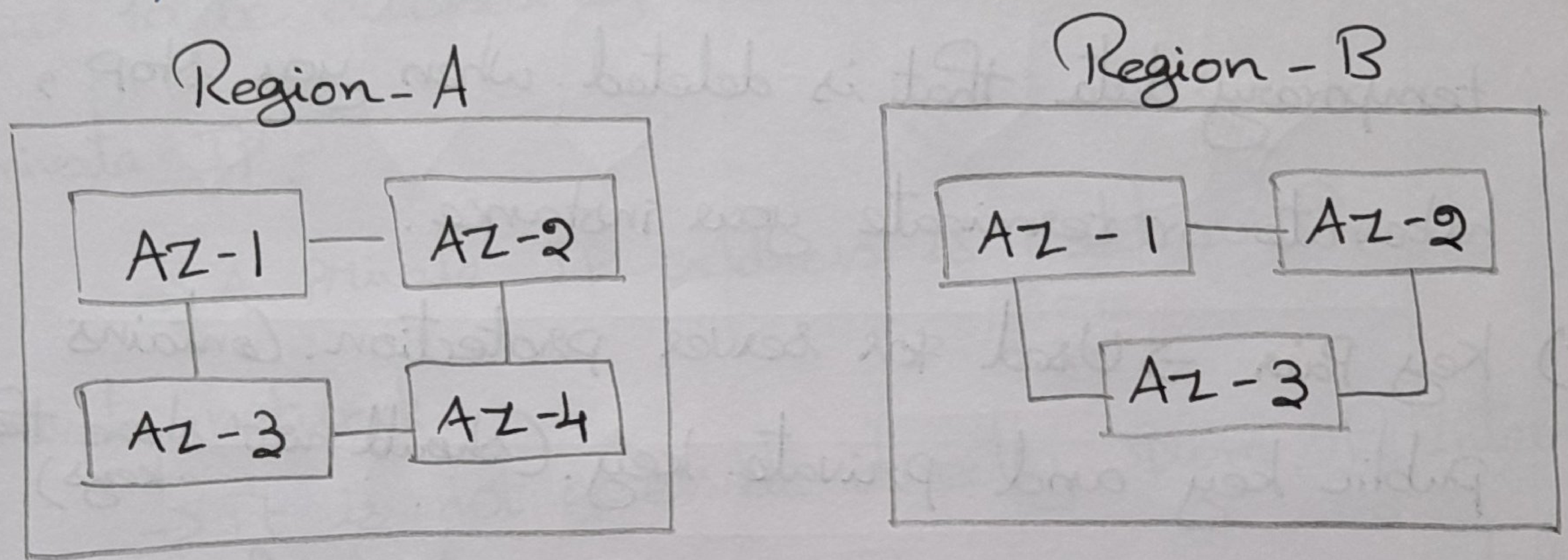
→ Group of servers is called Datacenters.

Availability Zones :-

-> An availability zone is a seperate data center within an AWS region.

-> It has its own power, network and cooling.

-> Using multiple datacenters AZn helps keeps your applications running even if one zone has issues



Types of EC2 Instances :-

* General Purpose
* Compute Optimized
* Memory Optimized
* Storage Optimized
* Accelerated Computing
* High-Performance Computing

Previous Generation Instances.

- For applications which are using heavy calculative ops.
- Memory Operations
- Store huge amount of data

Current Generation Instances.

Features of EC2 Instance :-

i) AMIs → Pre-configured templates for your instances.

ii) Instance types → Various configurations of CPU, memory, storage, n/w capacity & graphics hardware for your instances.

iii) Amazon Instance Volumes → Storage volumes for temporary data that is deleted when you stop, hibernate or terminate your instance.

iv) Key Pair → Used for server protection. Contains public key and private key. (should not share the keys)

v) Network Settings → Contains the config of VPC, subnet & Security Group.

vi) Security Group → A virtual firewall that allows you to specify the protocols, ports and source IP ranges that can reach your instances and the destination IP ranges to which our instances can connect.

Public IP:-

→ A public IP address is assigned to devices that need to be accessed over the internet.

→ It is unique across the entire internet and allows communication b/w different networks.

→ Ex: Used by instances in AWS when they need to be accessed by clients from outside the VPC.

Private IP:-

→ A private IP address is used within a private network.

→ It is not accessible directly from the internet

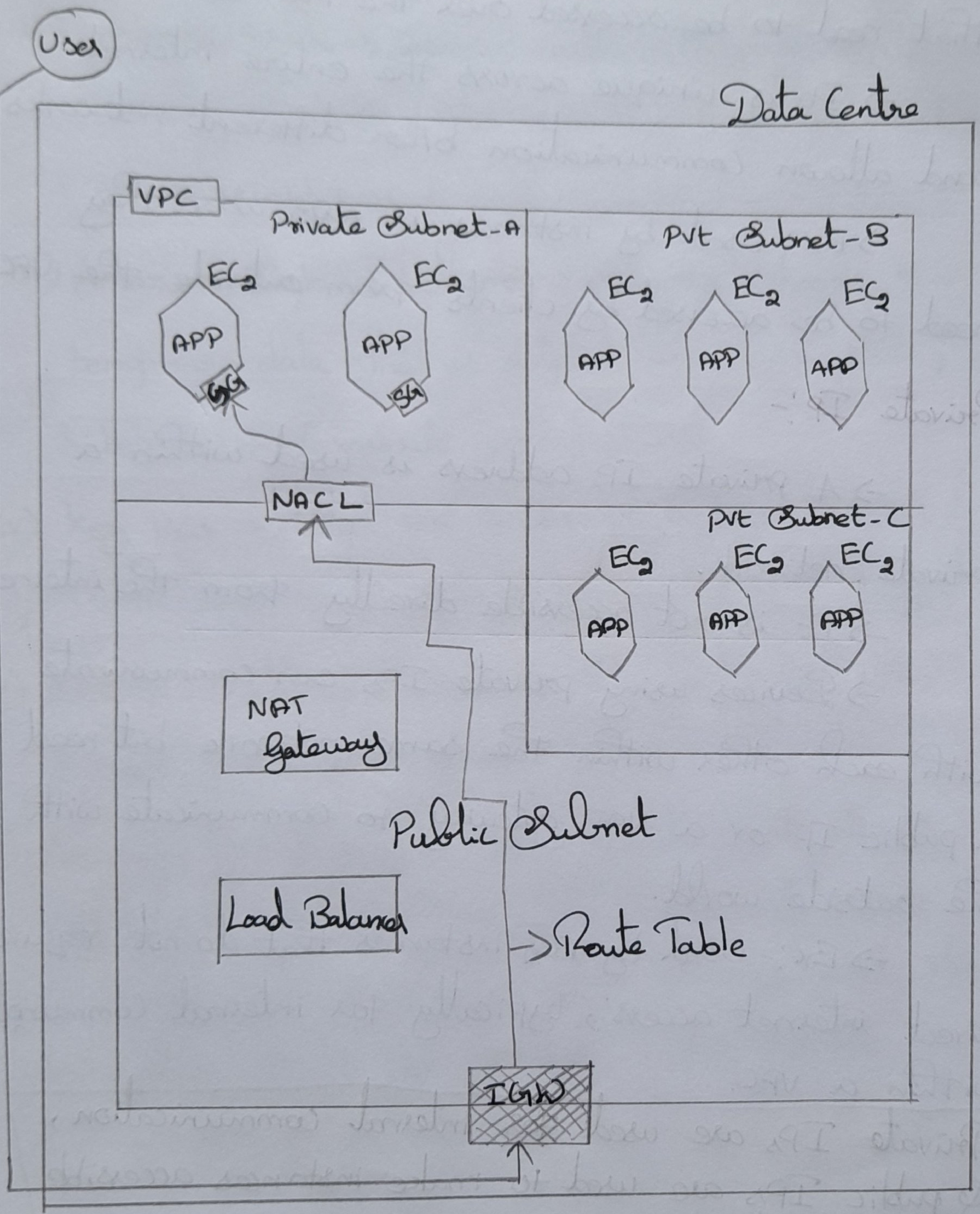
→ Devices using private IPs can communicate with each other within the same network but need a public IP or a NAT gateway to communicate with the outside world.

→ Ex:- Used by EC_2 instances that do not require direct internet access, typically for internal communication within a VPC.

Private IPs are used for internal communication, & public IPs are used to make instances accessible from the internet

23-09-2024

VPC - Virtual Private Network



VPC - Virtual Private Network :-

→ AWS VPC is a logically isolated network within AWS where you can launch AWS resources like EC_2 instances with control over network configurations.

→ VPC is a virtual network in AWS that is dedicated to an AWS account.

Components of VPC :-

Public Subnet → Has direct access to the internet through an internet gateway.

→ Instances in a public subnet can send and receive traffic from the internet.

Private Subnet → Does not have direct internet access.

→ Instances in a private subnets typically use a NAT gateway for outbound internet connectivity while remaining inaccessible from outside.

NAT Gateway → Network Access Translation Gateway that enables instances in a pvt subnet to connect to the internet while preventing incoming connections from the internet.

→ NAT Gateways allow secure outbound internet traffic by masking private IPs behind the gateway's public IP.

→ It is often used to download updates or communicate externally without exposing internal resources.

Route Table → A route table is a set of rules, called routes, used to determine where n/w traffic is directed within the VPC.

→ Each subnet in a VPC is associated with a RT.

→ Helps to control traffic flow in a VPC.

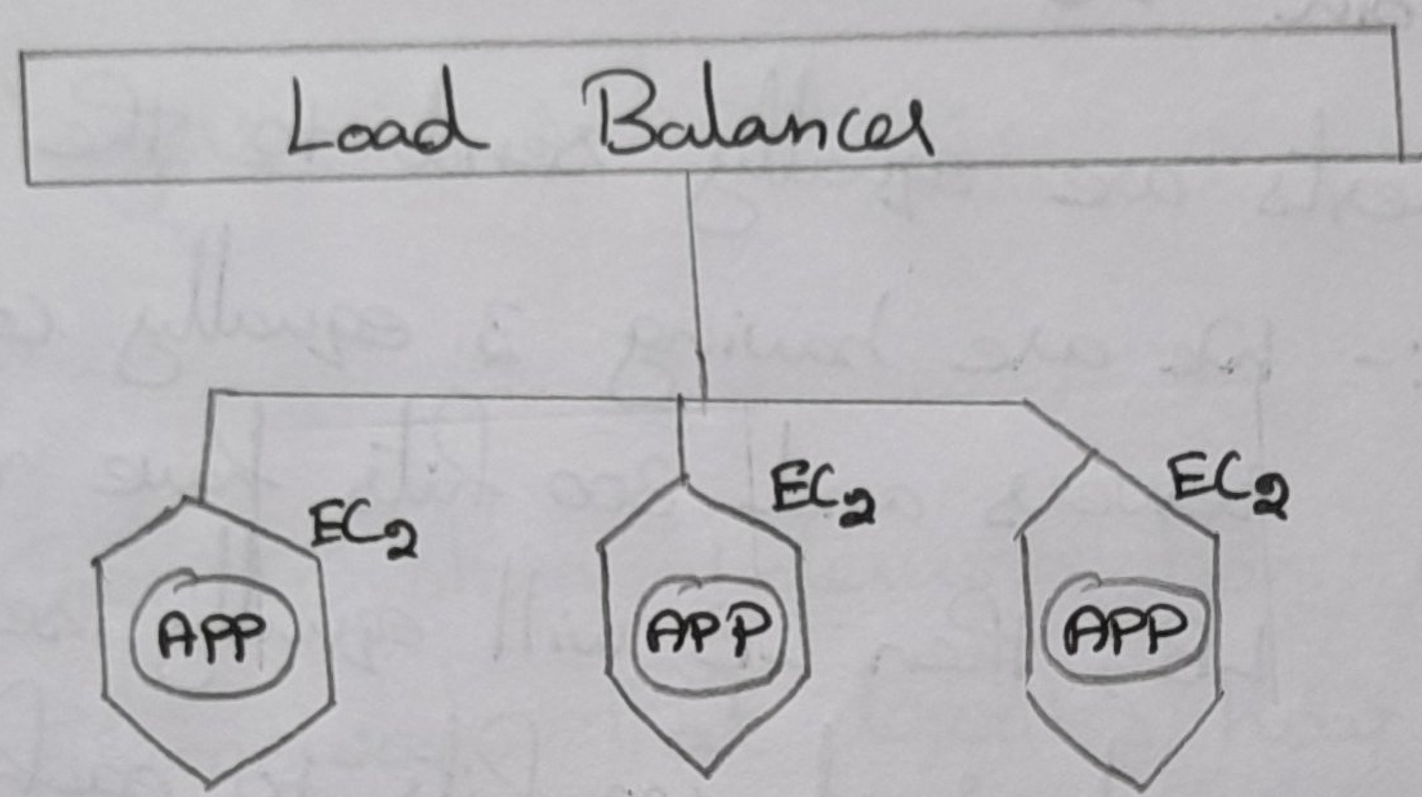
Internet Gateway → An Internet Gateway is a component in a VPC that enables communication b/w instances in a public subnet and the internet.

SG & NACLs → Controls inbound & outbound traffic at the instance level and subnet level, respectively.

24-09-2024

LB – Load Balancer

A Load Balancer is a critical component in AWS and cloud infrastructure that distributes incoming application traffic across multiple instances or resources to ensure reliability, efficiency and scalability.



Load Balancing Mechanisms :-

- → Round Robin
- → IP Hash
- → Weighted RR
- → Random.
- → Least Connections
- → Least Response Time
- → Weighted Least Connections

Software Lifecycle:-

-> Code
-> Build
-> Test
-> Deploy

Language	Build Tool.
Java	Maven
.Net	VB
C#	make file

Round Robin LB Mechanism:-

-> In RR Requests are equally send to the servers.

-> For ex:- We are having 3 equally configured servers and 300 hits have received by the LB, then LB will equally seperate the hits and send 100 hits to each server.

-> Most used by organizations.

Weighted LB Mechanism (or) Ratior Based LB Mechanism

-> Requests are sent to the servers based on the server configuration.

-> Ex:- If one of the sever is huge in size and the other 2 are small in size and receiving 100 hits then the LB has sent the request in the following manner.

$S_1 \Rightarrow 60$ $S_2 \Rightarrow 20$ $S_3 \Rightarrow 20$.

Types of Load Balancers :-

- -> Classic LB (Deprecated)
- -> Application LB
- -> Network LB
- -> Gateway LB.

Before discussing these types, we need to know about OSI Layers in Networking.

OSI Layers :-

-> The Open System Interconnection model is a conceptual framework that describes how data is sent and received through a networking system.

-> It made up of 7 layer.

Application Layer	=> End user Layer
Presentation Layer	=> Syntax layer
Session Layer	=> Synch & send to port
Transport Layer	=> End to End Connection
Network Layer	=> Packets
Data Link Layer	=> Frames
Physical Layer	=> Physical Structure

Monolithic Architecture :-

-> A single, unified application structure where all components are tightly coupled and run as one services.

-> Suitable for small or medium-sized applications.

Microservices Architature :-

-> An architectural style where an application is divided into independent, loosely coupled services,

-> Each focused on a specific business function.

-> Suitable for large, complex applications that requires scalability, flexibility & frequent updates.

-> Ex. phonepay, Amazon, flipkart, etc.,

Load Balancers are mainly used for High availability of an application.

If an user going to hit the server like,
Amazon.com/clothing/kids/shirts, then the connection b/w client & server held as by the below,

Client System

Layers.

L_7 Application layer → HTTP layer. The application layer add the http to the request by default.

L_6 Presentation Layer → The request will be encrypted in this layer.

L_5 Session Layer → The records of the request like, which browse when, how long are stored in this layer.

L_4 Transport Layer → The full request will splited into small chucks in this layer.

L_3 Network Layer → The request has pass through many routers.

L_2 Data Link Layer → It ensuring the data packets are properly framed, transmitted and received without errors.

L_1 Physical Layer → In this layer the physical structure of the system will be described.

Server System.

OSI layer & Load Balancer are interlinked with each other.

→ Application LB acts on L7 (Application layer)

→ Network LB acts on L4 (Transport layer)

→ Gateway LB doesn't acts on any layer.

Application Load Balancer :-

→ If we want to URL based routing, host based routing, Domain based routing & path based routing then we chose to use ALB.

→ In ALB we are intercept the request at L7.

→ It is designed to handle HTTP/HTTPS traffic & makes routing decisions based on content of the request such as the URL paths, hostname.

→ Use cases :- Microservices architecture, web applications, Real-Time Applications & Multi-region Apps.

-> A bit slower since it has to make decisions and routing to the server.

-> Due to its perfect decision making (on which server the routing should be connected / the request should be send) it is very costly when comparing to other LBs.

-> ALB is a powerful tool for distributing HTTP & HTTPS traffic based on content.

-> Well-suited for modern-apps especially those involving microservices or containerized environments.

Network Load Balancers :-

-> It is used in gaming platforms & youtube servers.

-> It is very fast. No place for latency

-> NLB is suitable for real-time applications, gaming, financial services and any application that requires fast, reliable load balancing for non-HTTP protocols.

-> Designed to handle high-performance, low-latency traffic at the n/w layer of the OSI model.

Gateway Load Balancer :-

→ It is very secured compared to ALB & NLB.

→ Used for VPN applications like online banking.

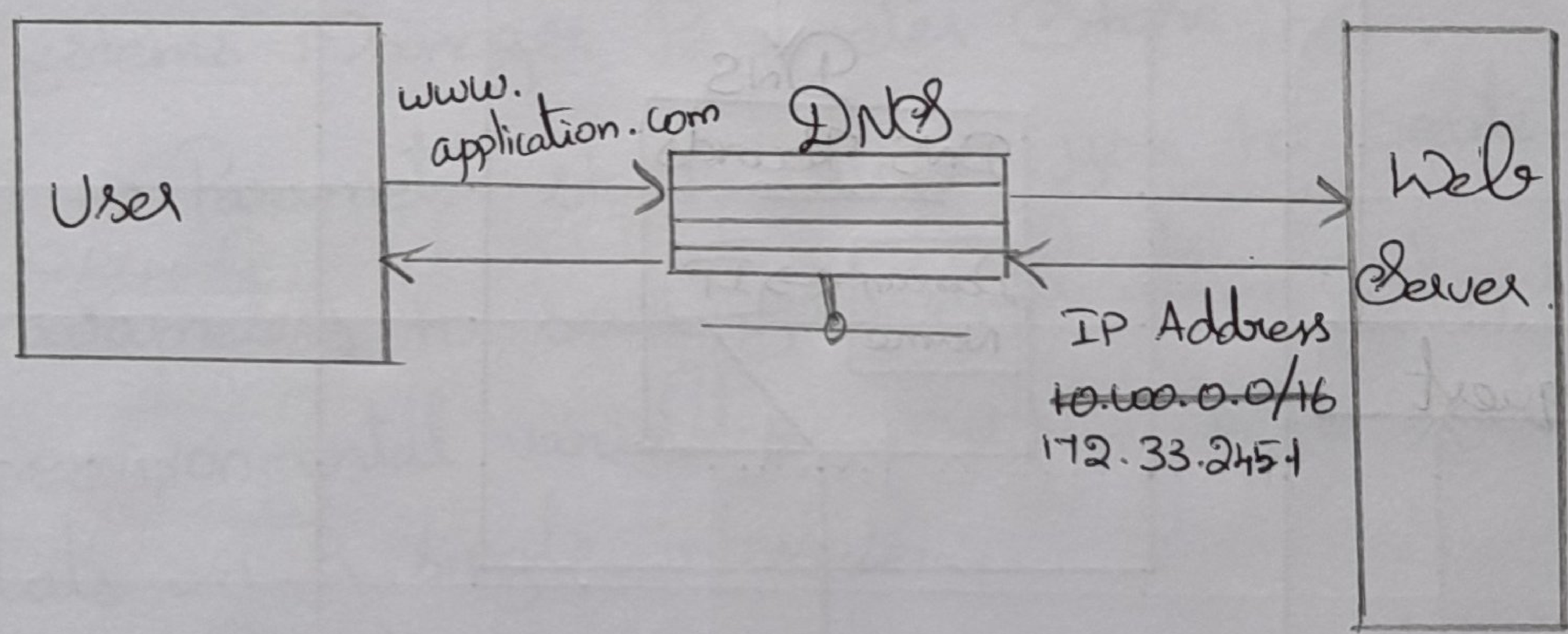
ALB	NLB	GW LB
Best for HTTP/ HTTPS traffic.	Best for TCP, UDP and TLS Traffic	Best for third-party virtual appliances.
Routes traffic based on URL paths, hostnames.	High performance, Low latency.	Integrates with virtual appliances for traffic inspection & filtering.
Ideal for web-apps, microservices & containerized apps.	Real-Time apps, IoT, & services requiring ultra-low latency.	Security appliances, n/w monitoring & traffic inspection.
Layer-7 Application Layer routing	Layer-4 Transport Layer routing	Layer-3. Network Layer routing.

25-09-2024

Route-53:-

→ Route-53 is a service provided by AWS for DNS as a service.

→ Domain Naming System is a system that translates domain names into IP addresses, which allows users to access websites by entering a human-readable name instead of IP addresses.



Public IP:-

→ Public IP is subjected to change. It is dynam

→ It is impossible to remind IP address for humans.

→ For that we are using domain name.

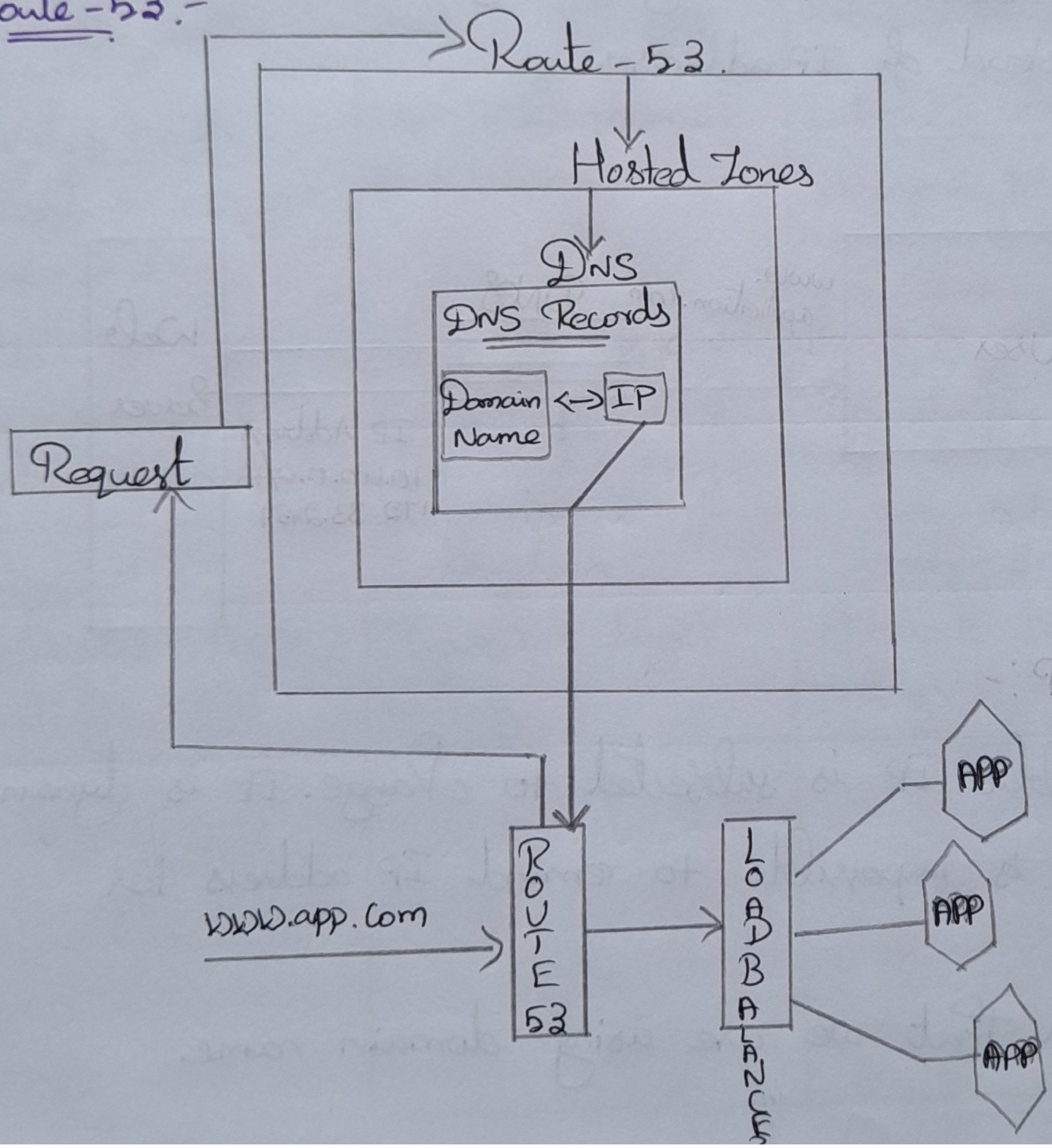
DNS :-

-> DNS resolving the domain name into the IP

-> Every Domain name will be mapped to an appropriate IP.

-> When somebody / user access the domain name, the DNS records will check the IP address mapped for that domain name.

Route-53 :-



Systems Manager & Secret Manager :-

-> AWS provides Systems manager & secrets manager for managing & securing the data such as environmental variables, DB passwords, API keys, product keys.

-> Systems manager parameter stores & secrets manager are two distinct services but offers similar functionalities that allows you to centrally managed and secure your secret information.

AWS Systems Manager Parameter Store :-

-> Parameter stores allows you to create-key value parameters to save your application configurations, custom environmental variables, product keys and credentials on a single interface.

-> It allows you to secure your data by encryption which is integrated with AWS KMS.

-> The secret information / sensitive data of our app should not be stored in plain text.

AWS Secrets Manager:-

-> Secrets manager enables you to rotate, manage & retrieve DB credentials, API keys and other secrets throughout their lifecycle.

-> We can integrate secrets manager with lambda function.

-> So that, the secrets will be changed through setting the conditions in lambda functions.

-> We can store passwords, API Tokens, SSL certificate, etc..,

Systems Manager	Secrets Manager
-> We can store the sensitive data. No-Automatic Rotation	-> We can rotate the sensitive data. Automatic Rotation
-> We Used to store important credentials.	-> Used to store most sensitive data like password.

* -> Hashicorp Vault is a tool that helps organizations manage and protect sensitive data & secrets.

-> It is open-source & very popular when comparing to systems manager & secrets manager

* To store sensitive data we can use the combination of systems manager & secrets manager.

29-09-2024

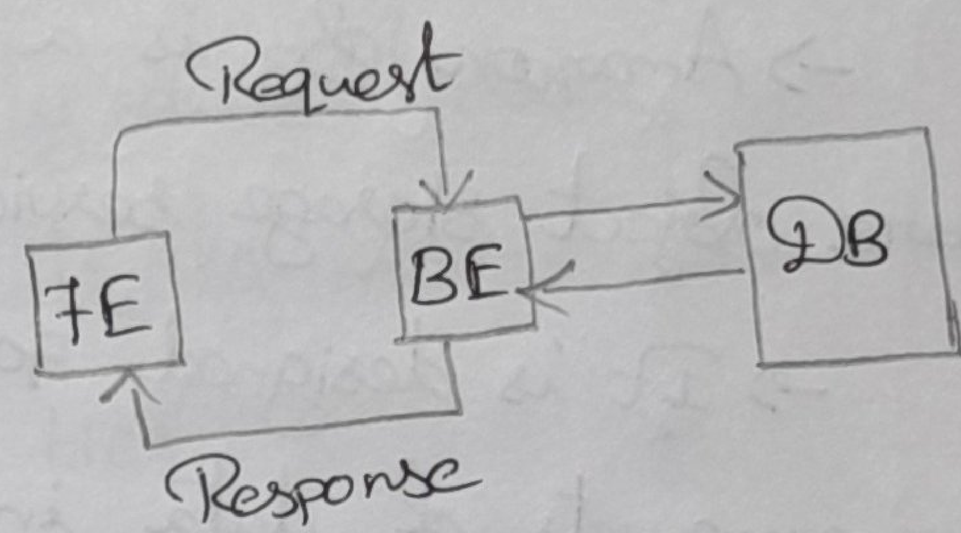
Three tier Architecture :-

-> It has 3 parts

i) Front End

ii) Back End

iii) Database

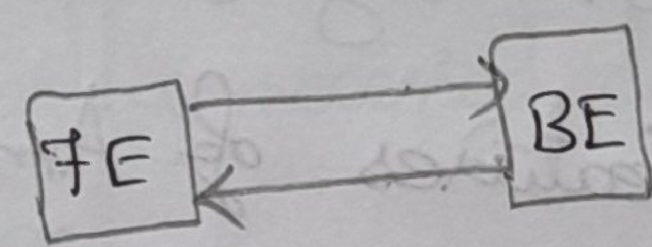


Two Tier Architecture :-

i) Front End

ii) Back End.

Ex :- Calculator



One Tier Architecture :-

i) Front End.

Ex :- Calender.

01-10-2024

S_3 - Simple Storage Service

-> Amazon S_3 is a highly scalable, durable and secure object storage service provided by AWS.

-> It is designed for storing and retrieving large amounts of data in a cost-effective way and can handle almost any type of data such as backups, archives, analytics data or content for applications.

Key features of Amazon S_3 :-

Storage for Any Data Type:

-> Stores objects as individual units of data in buckets.

-> We can store any kind of data such as text, images, videos, backups, .. etc.

-> Each objects can be up to 5 terabytes in size.

Durability & Availability:

-> S_3 offers 99.999999999 (11 9s) durability by automatically replicating data across multiple devices & facilities in a region.

-> High availability ensures that data remains

Storage Classes:

S3 provides multiple storage classes to help costs with access needs.

Standard -> For frequently accessed data.

Intelligent-Tiering -> Automatically moves data to the most cost-effective access tier.

Standard-IA (Infrequent Access) -> For less frequently accessed data, lower cost with retreival fees.

One-Zone-IA -> For infrequently accessed data stored in a single availability zone.

Glacier -> For long-term archiving with retrieval times of minutes to hour.

Glacier Deep Archive -> For data that can be retained for years & retrieved in hours.

Scalability:-

-> We can create n no. of object in a bucket.

Data Security:-

-> Data can be encrypted both at rest and in transit

-> We can use object lock to prevents deletion or modification of objects.

Key features:-

* High Availability & Durability
* Reliability
* Scalability
* Performance
* Security
* Cost - Effectiveness

-> Applications deployed in the server, there will be user data dumps & logs.

-> User data dumps are the files that listing out the objects & the metadata (username & passwords)

-> Logs are records of like who is using the app, how long he/she was in the app. Logs maintainging this kind of details for a security purpose.

The multipart utility is a Amazon S3 feature that allows large objects to be uploaded in smaller parts. This is useful for uploading large files as it increases efficiency & fault tolerance.

08-10-2024

Security

→ For network level security AWS provides the following services

i) VPC

ii) Security Groups

iii) NACL (Network Access Control Lists)

VPC :- (Virtual Private Cloud) -

→ Inside a public cloud we can create a private network called VPC.

→ It is a secure, isolated network environment within AWS where you can deploy and manage AWS resources such as EC2 Instances.

Key Components of VPC :-

* Subnets
* Route Table
* Internet Gateway
* NAT Gateway (Network Address Translation)
* NACL (Network Access Control Lists.

Inbound Rules:-

-> Inbound rules are the traffic that is flowing into the application

Outbound Rules:-

-> Outbound rules that controls the traffic flowing out of the application and connects to the other application.

-> They are set of configurations that controls the traffic that leaves a network to an external destination.

In AWS security is a shared responsibilities. b/w AWS and the customer.

-> AWS Responsibility => Security of the cloud

-> Customer " => Security in the cloud.

-> Opening port 25 can reveal your server's IP address when sending emails, as it's included in email headers. This exposure can attract hackers, who may scan or target your IP for vulnerabilities.

What sort of automation can we doing in NACL?

→ NACL is providing subnet level security.

→ We can launch multiple instance in a subnet. So that NACL providing a common rule for security to all those instances which are located in a subnet. NACL controlling the traffic by the rules to all the instances.

→ One common rule in NACL can control all the instances. This is the manner of automation done by NACL.

Security Groups:-

→ Providing the security at the instance level

→ Has inbound and outbound rules, evaluated as a single set for all rules.

→ If you allow inbound traffic, return traffic is permitted.

→ more flexible and widely used for instance-level security.

Network Access Control Lists (NACLs) :-

-> Providing security at the subnet level.

-> Has separate inbound and outbound rules, evaluated in number order (lowest first)

-> Both inbound and outbound rules must permit traffic for bidirectional communication.

-> Provides an additional layer of security for subnet.

SGs are instance level, stateful & commonly used for instance-specific control. Whereas NACLs are subnet-level & stateless; Ideal for blocking ranges.

10-10-2024

IAM - Identity & Access Management :-

-> IAM is a service that provides securely control access to AWS resources.

-> IAM manages permissions for users, applications and services & controlling who can access which resource in AWS and what actions the can perform.

Authentication :-

-> The process of verifying the identity of a user, application or service.

-> Passwords, biometrics, MFA & tokens are commonly used authentication methods.

-> Ex :- Logging into AWS Management Console with a username, password & MFA.

Authorization :-

-> The process of granting permissions to authenticated entities, determining what actions they can perform.

-> Policies, roles & permission settings are the authorization methods.

-> Ex :- Allowing an authenticated user access to an S3 bucket based on their IAM role.

Root user in AWS:-

→ The AWS root user is the account's primary identity with full access to all resources and settings.

→ We shouldn't give the root credentials to anybody. So that, we can prevent our account from malicious activities.

IAM User:-

→ An IAM user is an individual identity in AWS with specific permissions to access resources and perform actions based on assigned policies.

→ Each IAM user has unique credentials and can be granted limited privilage, following the principle of least privilage.

→ We can create IAM user for a person by giving authentication & authorization.

Policies:-

→ Policies in AWS are JSON documents that define permissions, specifying which actions are allowed or denied on specific resources.

→ They can be attached to IAM users, groups or roles to control access & to give authorization for an IAM user.

Groups:-

-> Groups in AWS IAM are collection of IAM users that simplify permission management by assigning policies to multiple users at once.

-> Used to automate the authorization.

Roles:-

-> Roles are used to access grant permissions/access from one service to another service, one AWS account to another AWS account.

-> Roles are a type of identity used to define a set of permissions.

-> IAM roles provide a way to delegate permission to AWS services or users temporarily, without the need to share a long-term credentials (like passwords or access keys.)

* Policies are created to provide permissions to the user.
* If one service in AWS wants to access an another service (or) if one AWS a/c wants to access another AWS a/c (or) if one service is in private cloud and another one is in public cloud then we can create the role for them.

11-10-2024

AWS CLI - AWS Command Line Interface.

-> The AWS CLI is a tool that enables users to interact with AWS services directly from the command line or terminal.

-> It allows you to manage and automate AWS resources and services by issuing commands rather than using the AWS Management Console.

-> If our PM asked to create 100 EC_2's & 100 S_3 bucket, then it will take more that 5 hours when we are using UI in AWS.

-> This is not preferable & it'll consuming more man power and time.

-> This manual method is not automation friendly.

-> AWS CLI is used to do this kind of automation tasks.

* Automation is the use of technology to perform tasks with minimal human intervention.
* It involves creating systems, processes or scripts that execute repetitive or complex tasks automatically.
* The main goal of automation is to reduce manual effort, minimize errors & increase productivity.

-> There are two ways to reach an application.

i) User Interface (UI)

ii) Application Programming Interface (API)

-> We can approach the application programatically

-> We will write the program to invoke the API of AWS to create, update or delete any AWS resources or services.

-> There are multiple ways to invoke API of AWS.

i) AWS CLI ii) AWS CDK

iii) Terraform iv) AWS cloud formation.

iii) & iv) are IAC tools.

-> AWS CLI written in python language

-> It is a python utility.

-> In AWS CLI, we will write commands using parameters.

-> AWS CLI convert these commands into API requests.

-> It is acts as an interface between user and AWS UI.

-> That is used to build infrastructure.

13-10-2024

CloudWatch:-

-> Amazon CloudWatch is a monitoring & observability service for AWS resources and applications.

-> It collects and tracks metrics, logs and events from various AWS services.

-> Monitoring, reporting, logging and alerting are all done by cloudwatch.

-> It's like gate keeper of our AWS account.

Monitoring:-

-> We can do infrastructure monitoring with the help of CW. (Ex: EC_2, S_3, VPC...etc.,)

-> Real-time metrics like CPU utilization, disk-in, disk-out are monitored by CW.

-> CW is done the monitoring based on these real-time monitoring.

-> In EC_2, CPU utilization, disk-in, disk-out, status check, disk-read & disk-write are the real life metrics.

-> We can do application monitoring but we should configure as we need. (Ex. how many hits the application might receive (or) how many people are trying to use my app.)

-> Application monitoring means how many API request is our application getting.

* AWS cloudwatch supports hundreds of built-in metrics across AWS services, and you can also define custom metrics if needed.

Alert or Alarm:-

-> We can set alert or alarm for our monitoring.

-> By integrating CW with SNS, this can be acheived.

-> The messages are received by email, text or mobile number.

Log Insights:-

-> It helps us quickly analyze, search & visualize log data from our AWS services & apps.

-> It's designed to make it easy to work with large volums of log data.

-> We can do application monitoring but we should configure as we need. (Ex. how many hits the application might receive (or) how many people are trying to use my app.)

-> Application monitoring means how many API request is our application getting.

* AWS cloudwatch supports hundreds of built-in metrics across AWS services, and you can also define custom metrics if needed.

Alert or Alarm:-

-> We can set alert or alarm for our monitoring.

-> By integrating CW with SNS, this can be acheived.

-> The messages are received by email, text or mobile number.

Log Insights:-

-> It helps us quickly analyze, search & visualize log data from our AWS services & apps.

-> It's designed to make it easy to work with large volums of log data.

Custom Metrics :-

-> We can configure the metrics as we need.

-> Memory utilization is not available by default. It's not a built-in metrics.

Cost Optimization & Scaling :-

-> CW will take the help of auto scalling to scale up or scale down the resources.

-> Ex. Launching a new EC_2 Instance.

-> When its coming to cost optimization, all the unwanted resources are deleted through run a program like python.

-> Through acheiving this points, the organization will get ~~bef~~benifit automatically.

Automatic dashboards => Default metrics

Custom dashboards => Customized by the user.

16-10-2024

Auto Scaling :-

→ AWS Auto Scaling is a service that automatically adjust the capacity of your AWS resources to maintain optimal performance and cost-efficiency.

→ We ~~can~~ can scaling resources up or down based on demand, helping to handle traffic spikes or periods of low activity without manual intervention

Types of Scaling :-

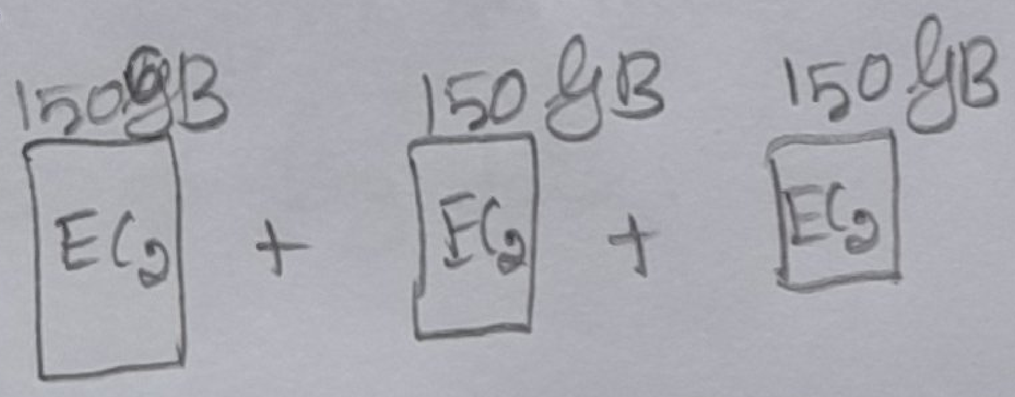
i) Horizontal Scaling

ii) Vertical Scaling.

Horizontal Scaling :-

→ Increases the count of the servers.

→ Ex: Adding more EC_2 instances to an Auto scaling group to handle higher traffic for an application.



⇒ Size remaing the same.

16-10-2024

Auto Scaling :-

-> AWS Auto Scaling is a service that automatically adjust the capacity of your AWS resources to maintain optimal performance and cost-efficiency.

-> We ~~can~~ can scaling resources up or down based on demand, helping to handle traffic spikes or periods of low activity without manual intervention

Types of Scaling :-

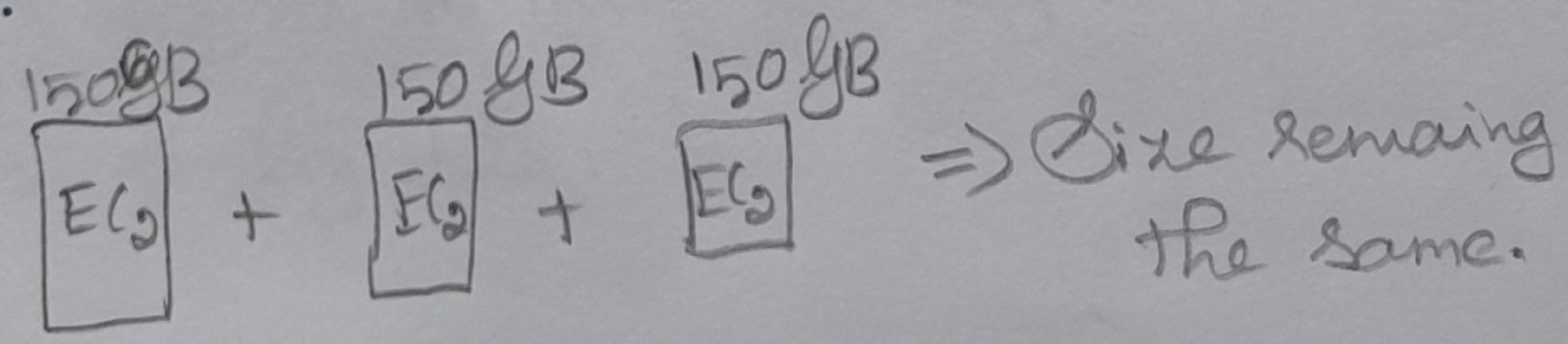
i) Horizontal Scaling

ii) Vertical Scaling.

Horizontal Scaling :-

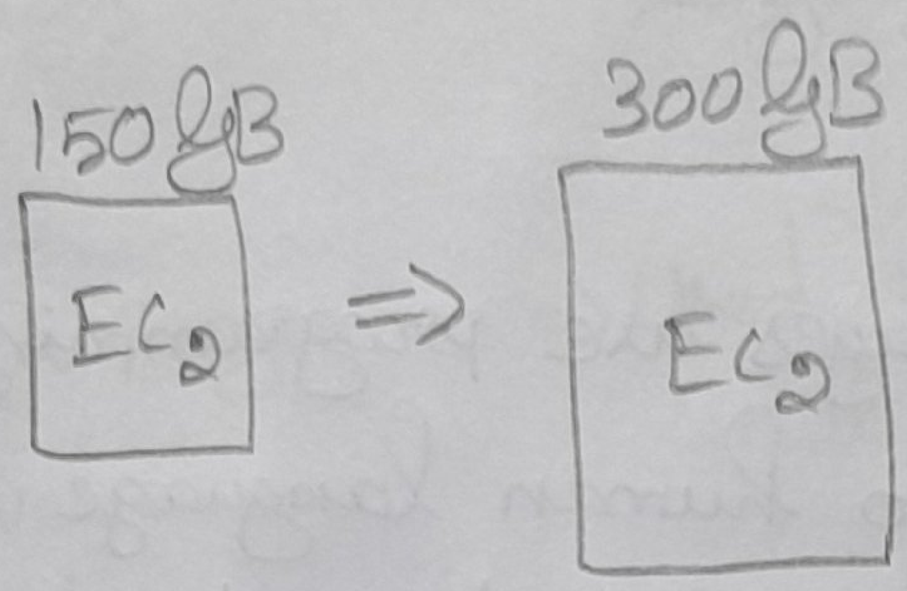
-> Increases the count of the servers.

-> Ex: Adding more EC_2 instances to an Auto scaling group to handle higher traffic for an application.



Vertical Scaling:-

-> Increasing the size of the server or resources.



=> Same server; the size of the server will be increased.

-> While the time period of increasing the size of the server, the server's functions will be stopped for while.

-> Which makes SPOF => Single Point of Failure.

-> This will cause bad user experience.

* Horizontal scaling adds capacity by adding more units, while vertical scaling increases the capacity of a single unit.

* Horizontal scaling is prefered in cloud environmental for better scalability & resilience.

Languages :-

i) High level Languages

ii) Low level Languages. => Binary / 0s & 1s

High Level Languages :-

-> High Level Languages are programming language that is closer to human language and abstracts away most hardware details, making it easier to read, write and maintain cod.

-> Ex :- C, C++, Java, Python... etc.,

Syntax :-

-> They refers to the set of rules and structure that define how code should be written in a programming language.

-> It includes keywords, symbols & puctuation.

Compiler :-

-> It is a program that translates code written in a high-level programming language into machine code.

High Level Languages.

- **Programming Languages**
  - => It's a program to develop an application.
  - => Ex:- Java, Python
- **Scripting Languages**
  - => Used to perform small task.
  - => Ex:- Shell Scripting, Python.
- **Markup Languages.**
  - => In which to give an I/P & get an o/p.
  - => Ex:- YAML, HTML, etc., JSON

-> Python is programming language & scripting language as well.

18.10.2024

Lamlda Function :-

→ AWS Lamlda is a fully managed, serverless compute service that automatically runs your code in response to events, scaling instantly as needed.

→ It eleminates the need for server management, charging only for actual compute time, making it ideal for scalable, cost-efficient applications.

→ EC_2 is an another compute service provided by AWS.

→ But in EC_2, the user only have to taking care of the dependencies & prerequisites, for the application which we are going to deploy in our EC_2.

→ When its regarding to Lamlda, AWS Lamlda will install and take care of all the dependencies & prerequisites needed by the application.

18.10.2024

Lamlda Function :-

-> AWS Lamlda is a fully managed, serverless compute service that automatically runs your code in response to events, scaling instantly as needed.

-> It eleminates the need for server management charging only for actual compute time, making it ideal for scalable, cost-efficient applications.

-> EC_2 is an another compute service provided by AWS.

-> But in EC_2, the user only have to taking care of the dependencies & prerequisites, for the application which we are going to deploy in our EC_2.

-> When its regarding to Lamlda, AWS Lamlda will install and take care of all the dependencies & prerequisites needed by the application.

Prerequisites:-

-> Prerequisites are the conditions or requirements that need to be met before starting a project / task or even installing an app.

-> Ex:- If we wanted to install Jenkins in our server, then our server must be having 4GB+ of RAM & 50GB+ of drive space.

Dependencies:-

-> These are resources or components that an application or a project/function relies on to work properly.

-> Ex:- For Jenkins application Java is a dependency.

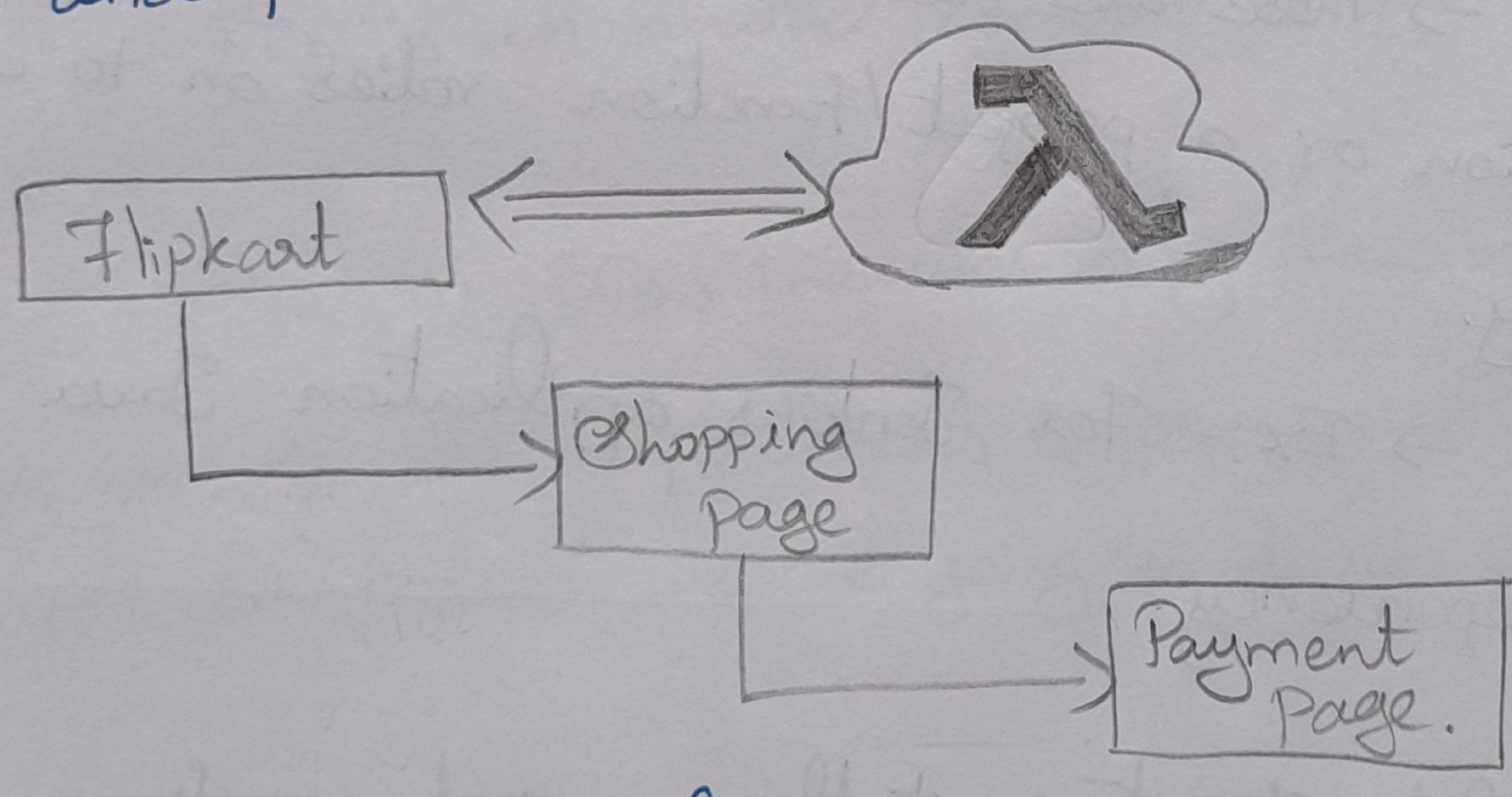
Lambda function itself can scale up & scale down of the usage of the resources based on the hits received ~~of~~ the application.

Tear down Process in Lambda:-

→ In AWS Lambda, tear down refers to the process of cleaning up or deallocating resources after a Lambda function has completed execution.

→ This is important to ensure resources are not left in use, which could incur unnecessary costs or create potential conflicts in subsequent executions.

→ It helps cost optimize resource use, control costs and prevent issues with re-used execution.



→ In the above example, after using the shopping page, the user will get directed to the payment page.

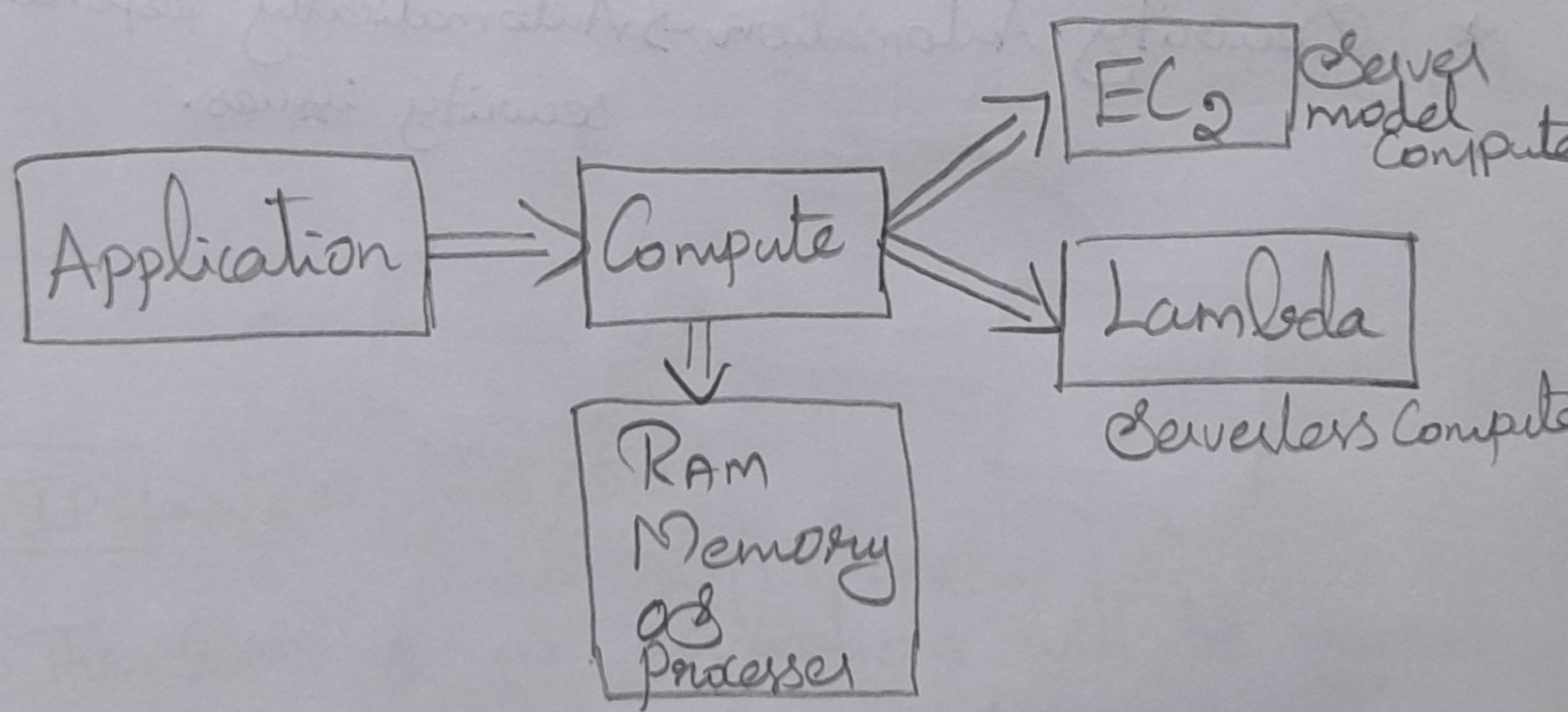
→ Once the payment done by the user the lambda function will tear down the payment page

→ When we deploy an application in EC_2, We will know about in which region the application will get deployed and in which AZ the application will get deployed and even IP addresses we will get to know.

→ But when a Lambda function involves in deploying application, it will deliver a DNS only to access the application.

→ It will not provide the detail about region, AZ & IP address.

→ That's why lambda function is called serverless.



Lambda function is great for running codes without managing servers.

AWS Lambda can be used for :-

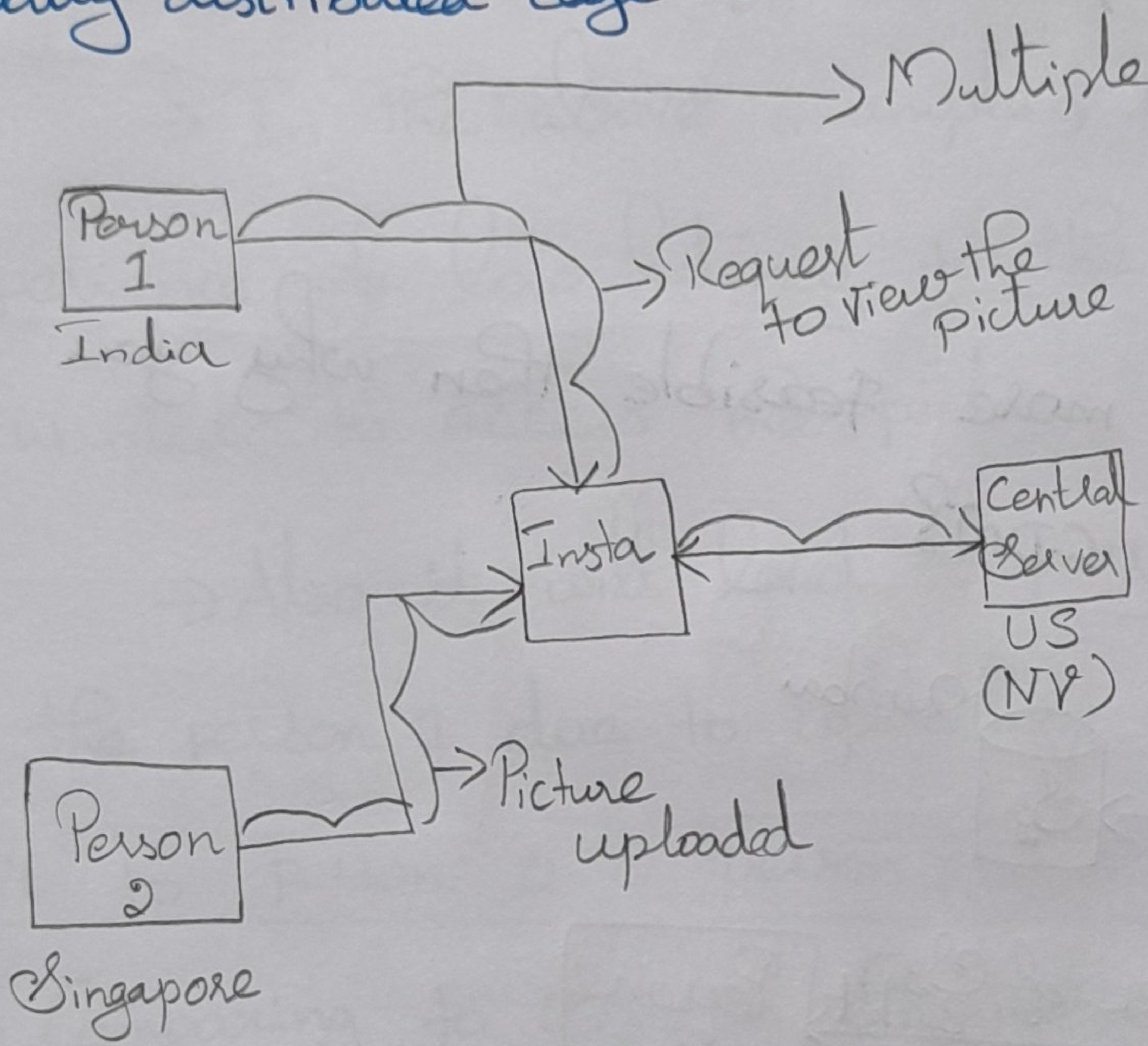
* File processing → Automatically resize images/convert files when uploaded to S_3
* Data Transformation → Clean/modify data befor storing it in DB.
* Automated Backups → Schedule regular backups for of our EC_2, RDS or EBS volumes.
* Event Handling → Process data from sources like SNS
* MicroServices → Build small, independent services
* Microservices
* Scheduled Tasks → Run tasks like cleanups or reports at specific times.
* Security Automation → Automatically respond to security issues.

19-10-2024

Cloud Front :- (CDN as a Service)

→ Amazon cloudfront is a Content Delivery Network service that provides to deliver your content, such as website, videos, images & APIs to users with low latency & high transfer speeds.

→ It distributes content through a network of globally distributed edge locations.



i) High latency
ii) Lead to bad user experience
iii) It'll take more time to access the picture

IP → DNS → CDN

→ The DNS of an applications will be mapped to with CDN. So that, we access only the CDN

→ There will be multiple edge locations around the world.

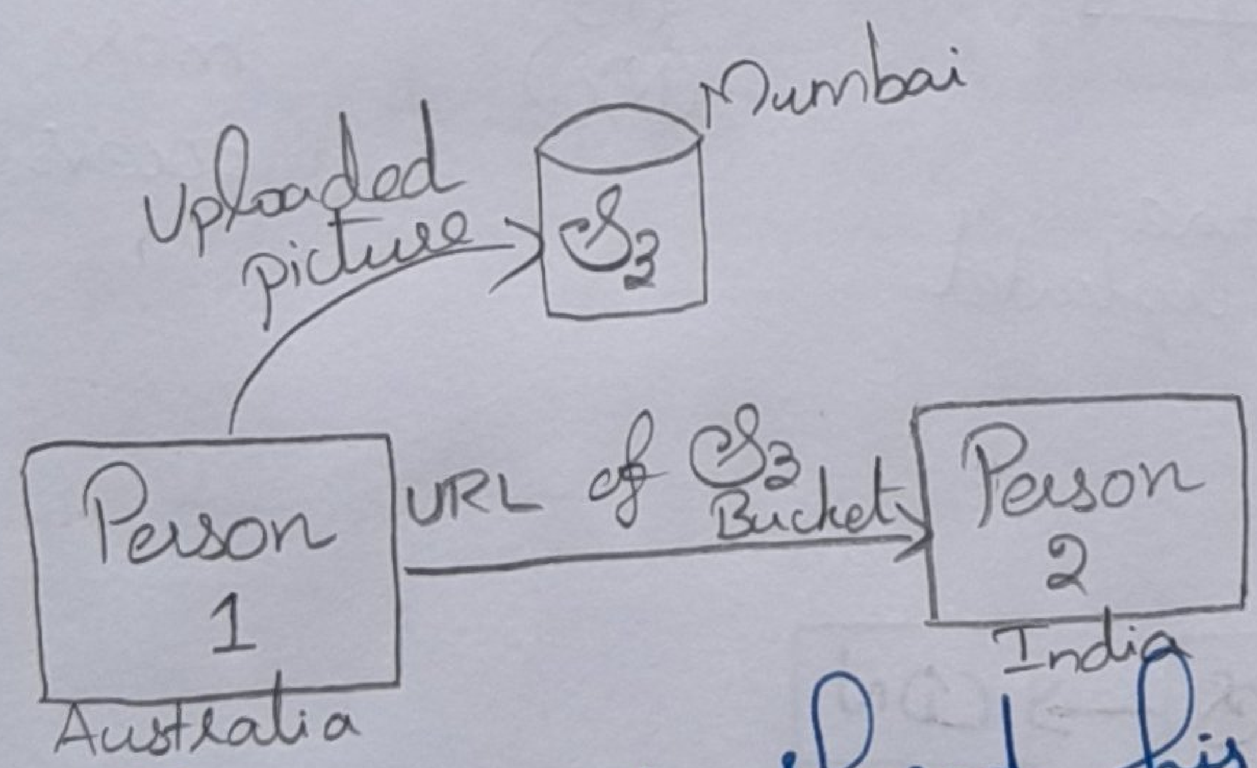
→ In each edge locations there will be a copy sent from central server.

→ It will overcome the problem of low latency.

→ CDN provides the better low latency service.

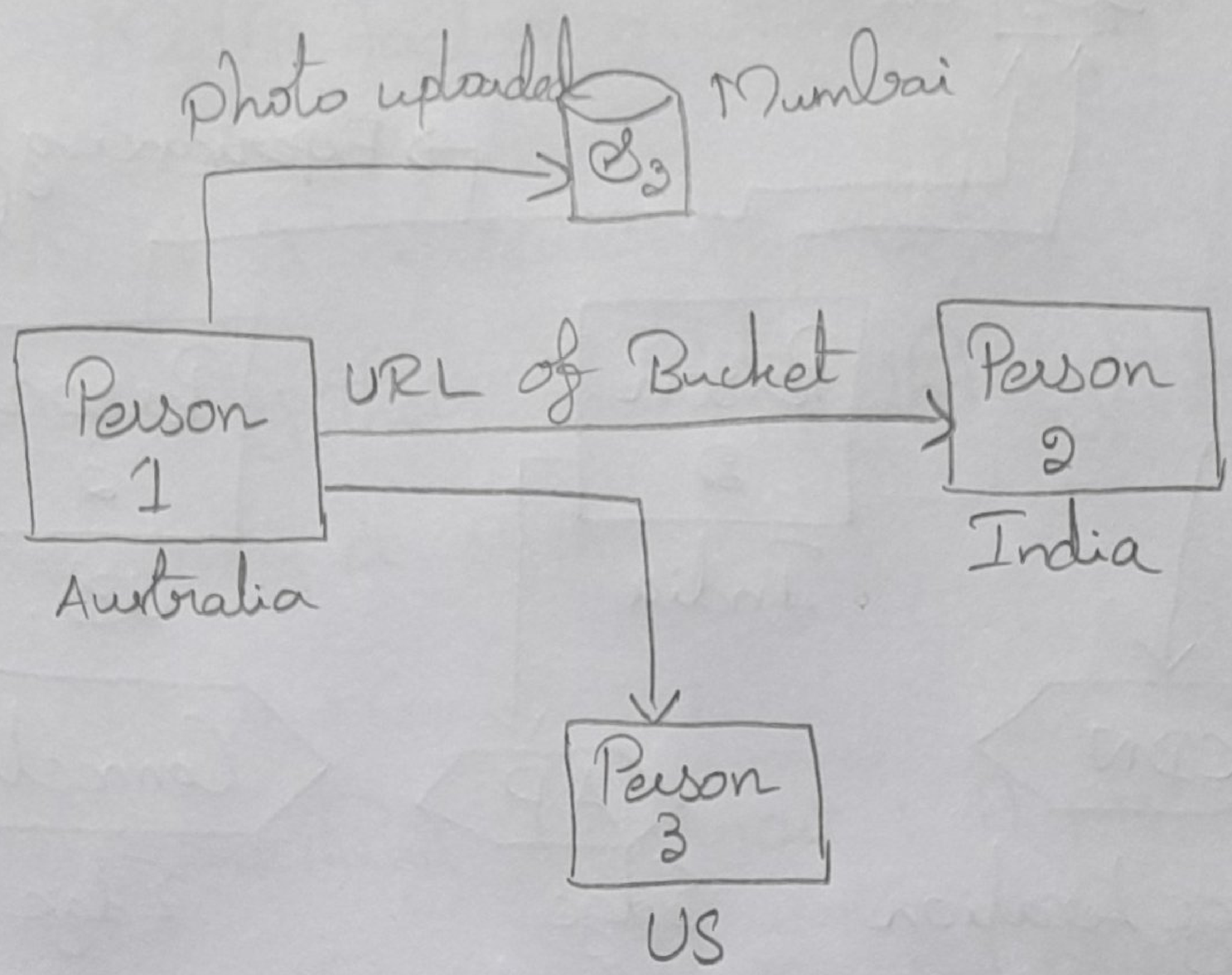
S_3 Vs CDN :-

* If S_3 is more feasible then why you wanted to go to CDN?



→ If person 1 shared his bucket URL to access the picture to person 2, it means he gave him the access of S_3 bucket to person 2.

→ It is comes under privacy breach.



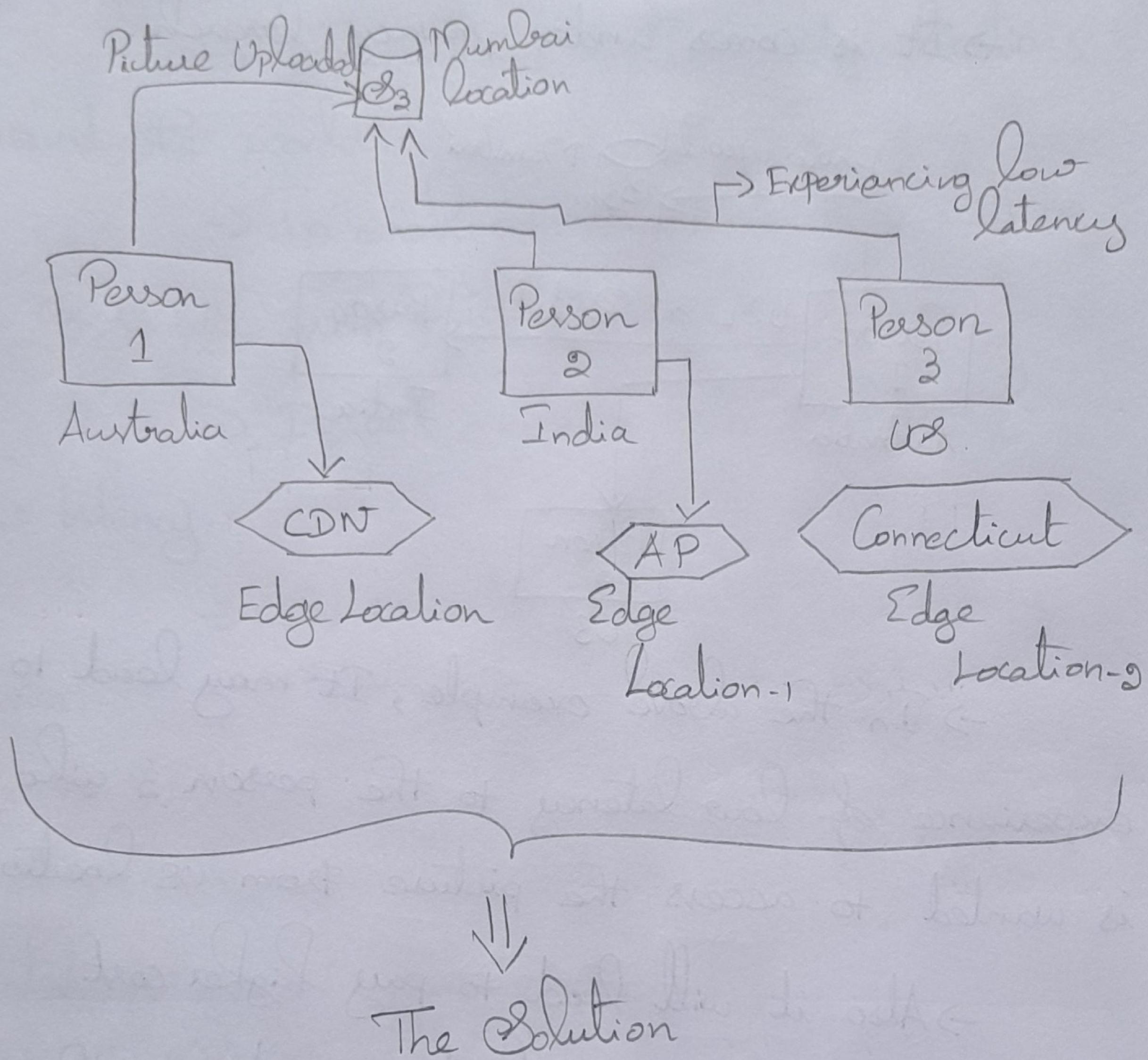
→ In the above example, It may lead to experience of low latency to the person 3 who is wanted to access the picture from US location.

→ Also it will lead to pay higher cost to the person 1 due to uploading & sharing the URL to person 2 & person 3.

Comparing to S_3, in the perspective of

i) Security
ii) Low Latency
iii) Cost

CDN is more feasible to users.


Picture Uploaded
S3
Mumbai location
Experiencing low latency
Person 1
Australia
Person 2
India
Person 3
US
CDN
Edge Location
AP
Edge Location-1
Connecticut
Edge Location-2
The Solution

20-10-2024

MFA - Multi Factor Authentication :-

→ MFA is a security mechanism that requires users to verify their identity through multiple methods before accessing a system.

→ It combines :

i) Something you know : A password or a PIN

ii) Something you have : A physical device, like smartphone app or security token.

iii) Something you are : Biometrics such as fingerprints / facial rec.

→ MFA greatly enhances the security by reducing the risk of unauthorized access, even if passwords are compromised.

→ In AWS, MFA comes under Identity & Access Management.

→ It is a two way protection.

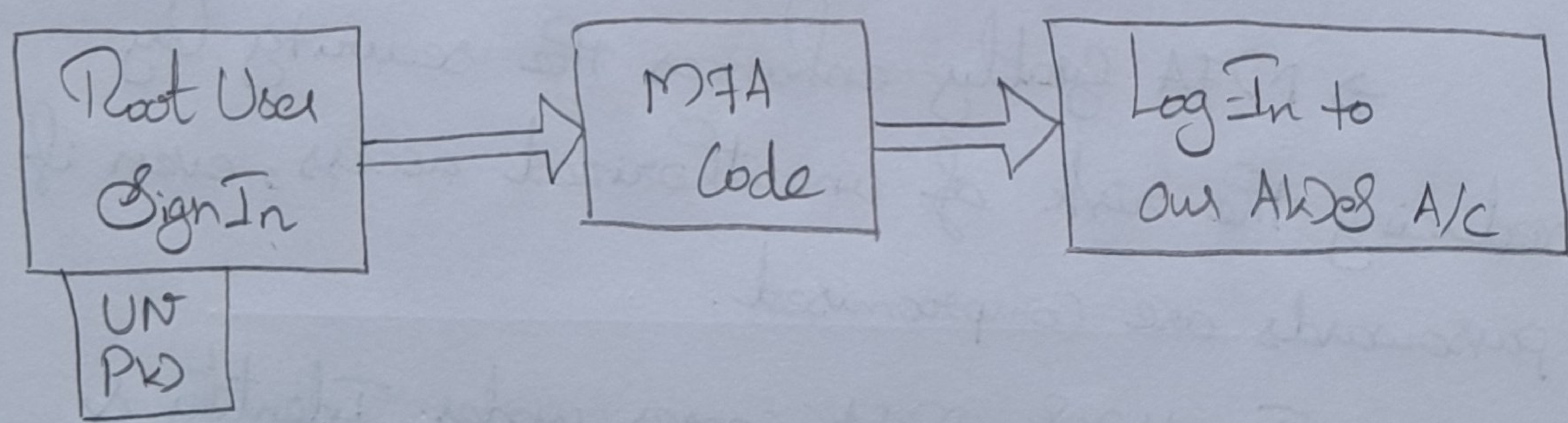
→ The 2 layer protection will perform as below,

i) Username & Password
ii) MFA device PIN number.

→ The PIN number will change over time.

→ In AWS, there are 3 way to enable MFA for our AWS a/c.

i) Passkey or Security key ⇒ fingerprint,
ii) Authenticator app. ⇒ App installed on your mobile device or computer.
iii) Hardware TOTP Token. ⇒ Satilite generated password.



MFA is an important best practice in AWS to secure user access to cloud resources.

21-10-2024

AWS KMS - AWS Key Management Service

-> AWS KMS is a managed service that helps you create, manage and control cryptographic keys for your applications and data across AWS.

-> It allows you to encrypt and decrypt data by ensuring sensitive information is secure.

Data Encryption:-

-> Data Encryption is the process of converting data into a coded format to protect it from unauthorized access.

-> It ensures, data protection, compliance, data integrity.

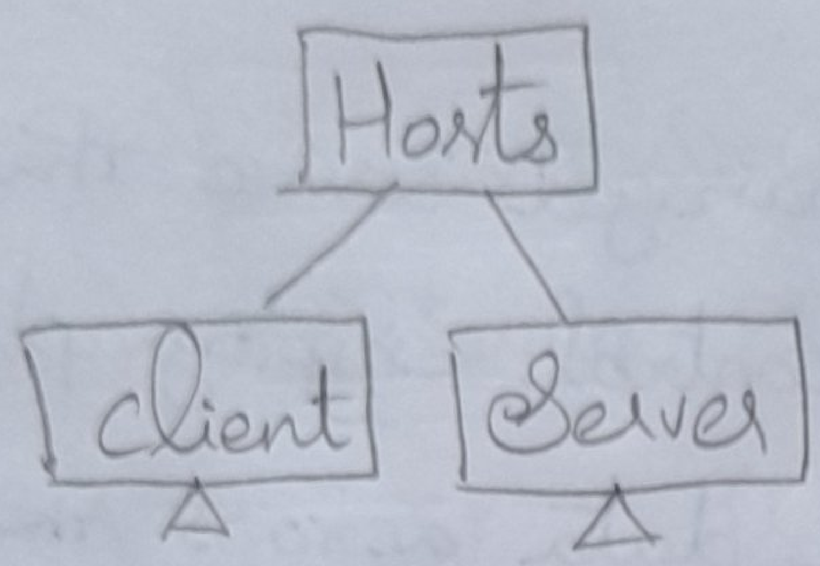
Plain Text:-

-> Plain text is unencrypted, readable data in its original form.

Plaintext : "Hello, World"

Cyphertext : "GVOZ!@*4!@YZ"

→ AWS KMS is used to encrypt our data.



→ If we want to encrypt our data, then we need some keys.

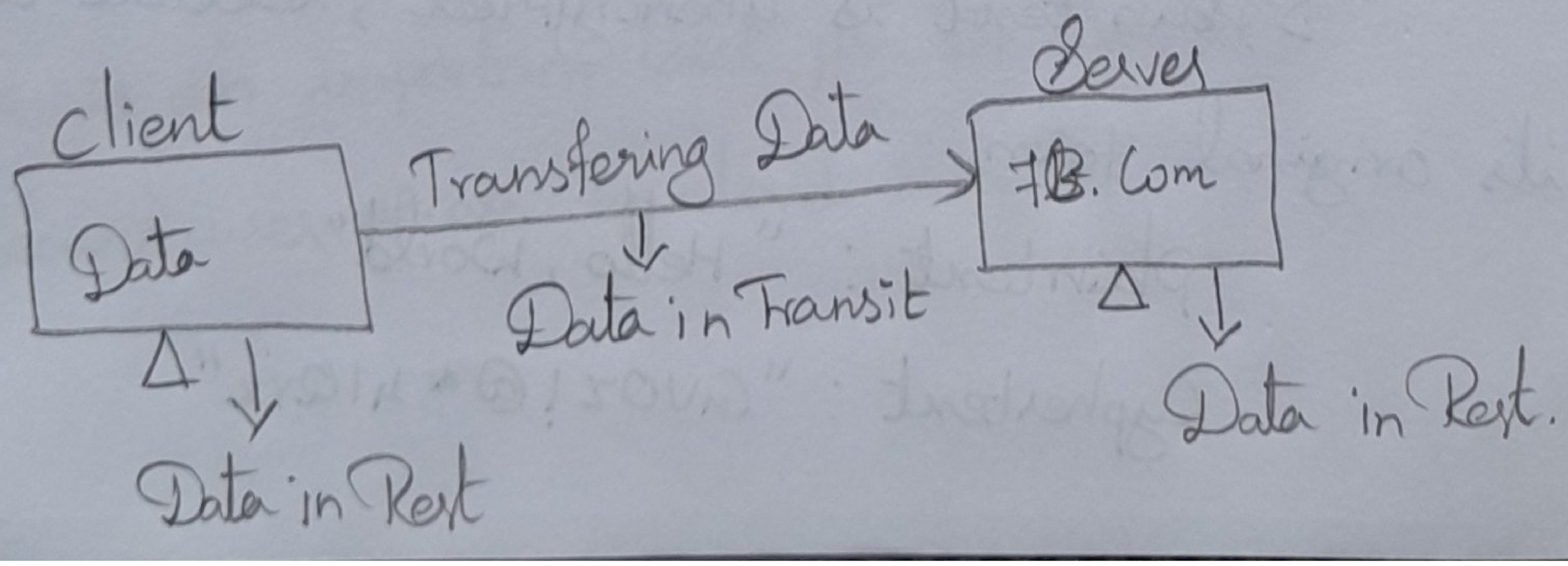
→ AWS KMS is used to store those keys.

→ Those keys are called Customer Master Keys. (CMKs)

→ There are two situations to encrypt our data.

i) Server Side Encryption
ii) Client Side Encryption.

→ The Customer Master Keys are used to encrypt our data.

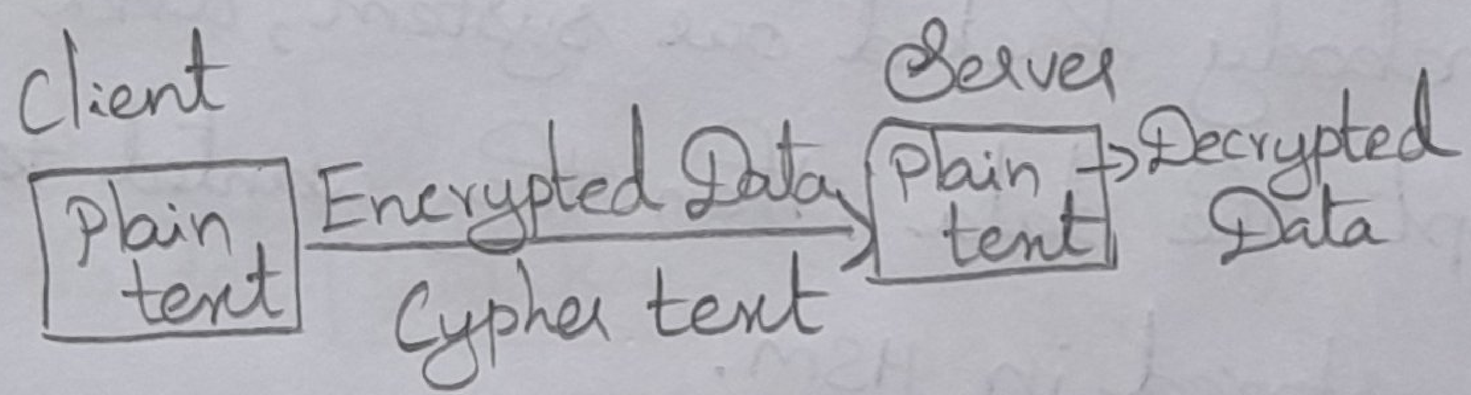


→ While transferring the data from our device to an application which is hosted in a web server, we will use some protocols like HTTP / HTTPS.

→ Usually we are using HTTPS.
↓
Having SSL certificate.

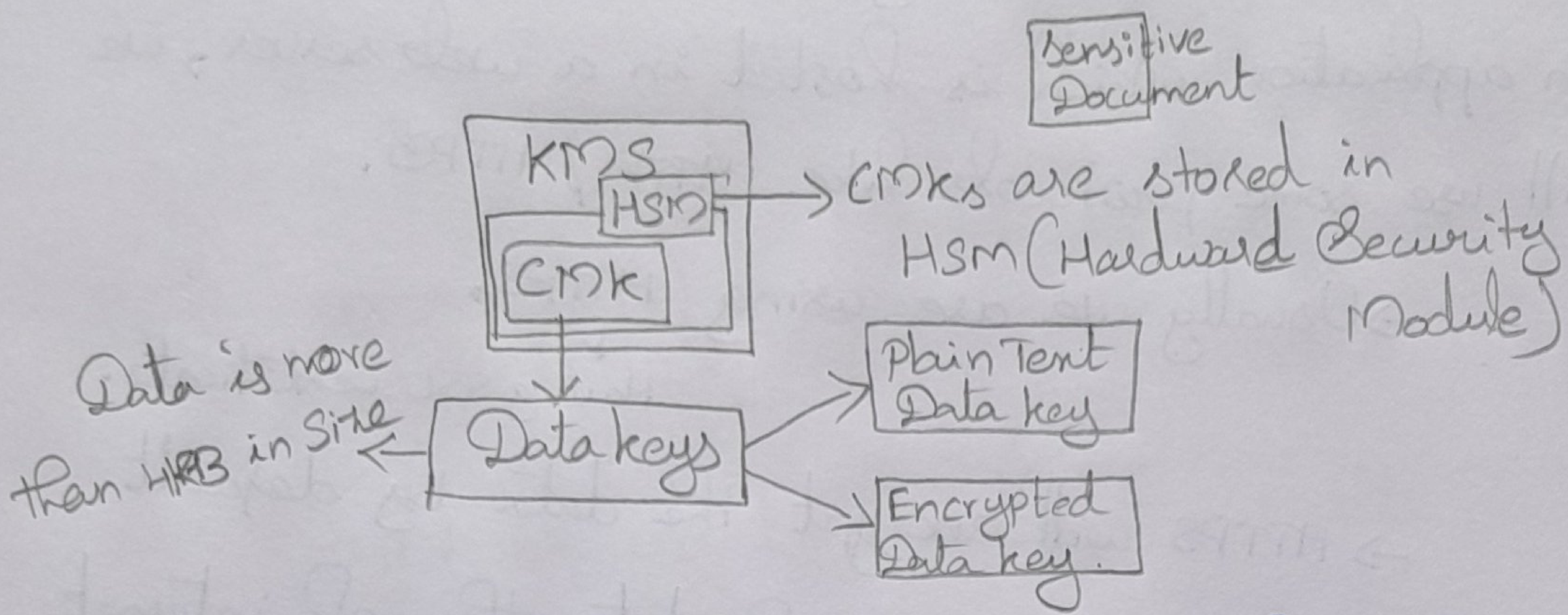
→ HTTPS will encrypt the data by default when we are transferring the data through internet.

→ So, there may be a chance of hacking our data while it is in rest.



→ When our data is in rest, either it may be in client or it may be in server then we can use KMS to encrypt these data.

The Encryption Technique of KMS:-



-> KMS will store CMKs in HSM (Hardware Security Module).

-> These HSMs are Tamper-free.

-> If somebody hacked our system, and they wanted to decrypt the data, then they wanted to these CMKs which are stored in HSM.

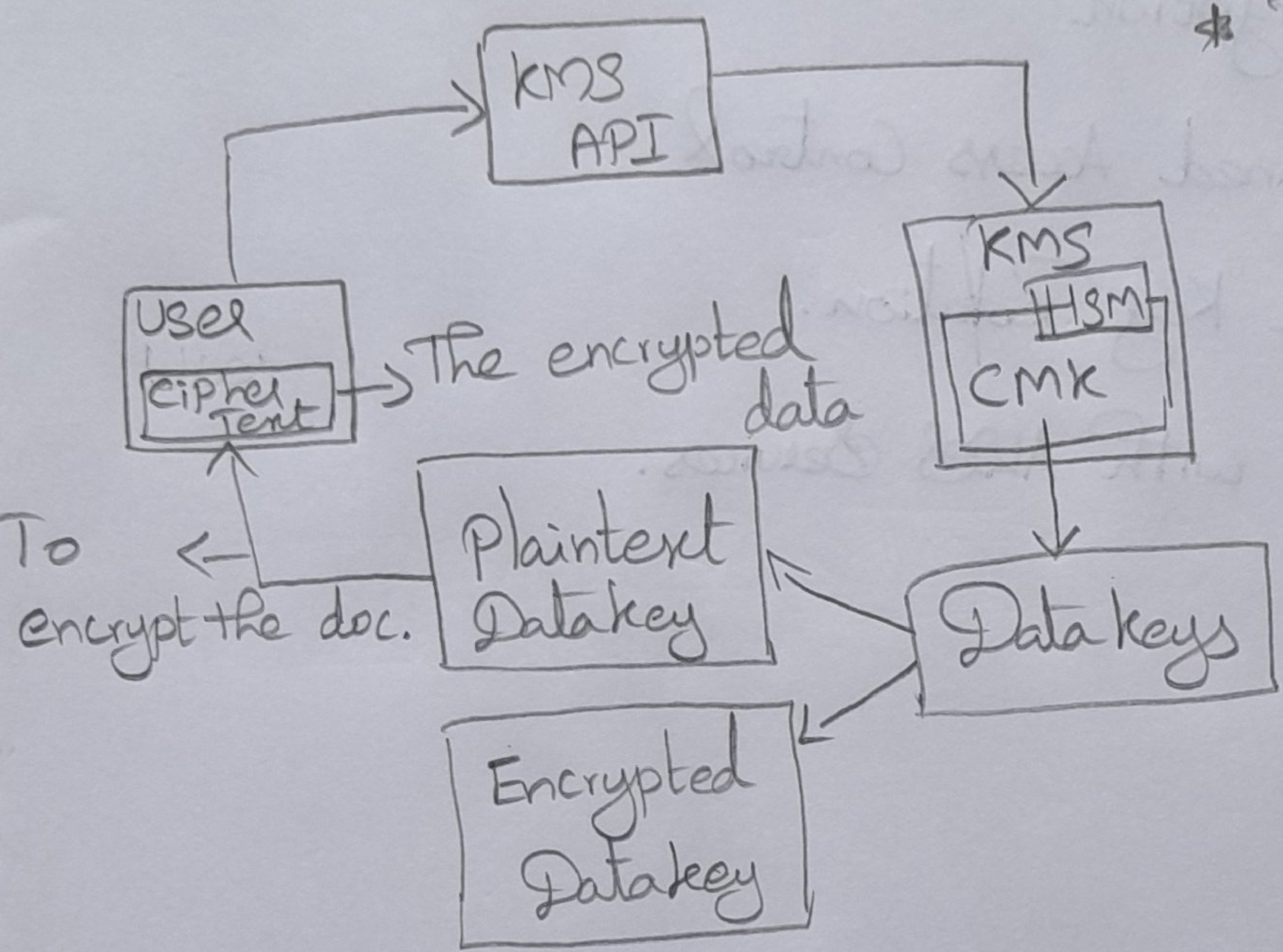
-> The HSM once will detect the malicious activities on these keys, then the HSM will auto delete the CMKs.

-> The algorithm will be written by us only.

-> However the CMK has the capability to encrypt the data which is in size of 4KB.

→ For The data which is more than 4KB, and to encrypt that data, the CMK will produce Data keys to encrypt the data (More than 4KB in size).

→ CMK has the capability to produce n no. of data keys.



* To Decrypt the cipher text also involves these steps.

→ The encrypted document is called cypher text doc

→ Once the document got encrypted then the plaint text data key should be deleted.

→ With the help of encrypted plaintext data key we can encrypt & decrypt the data as well.

→ KMS API is used to communicate with KMS.

→ We cannot approach the KMS directly.

Key features of AWS KMS:-

* Key creation & Management
* Data Encryption
* Fine-Grained Access Control
* Automatic Key Rotation.
* Integrate with AWS Services.

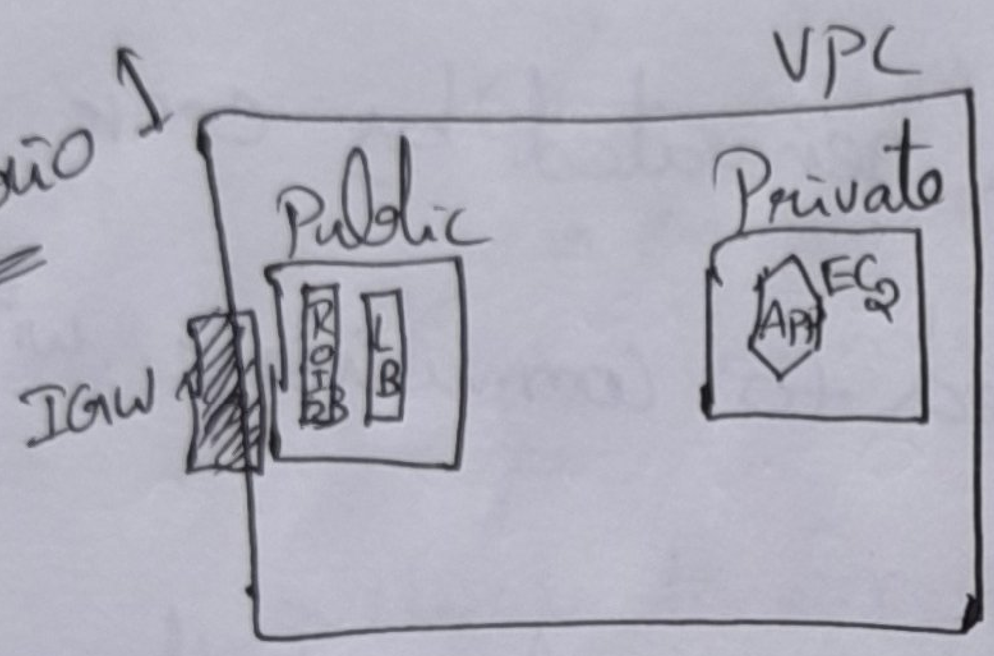
25-10-2024

NAT Gateway, NAT Instance & Elastic IP.

NAT Gateway:-

Problem:-

Scenario 1



* Organizations used to launch apps in pvt subnet only

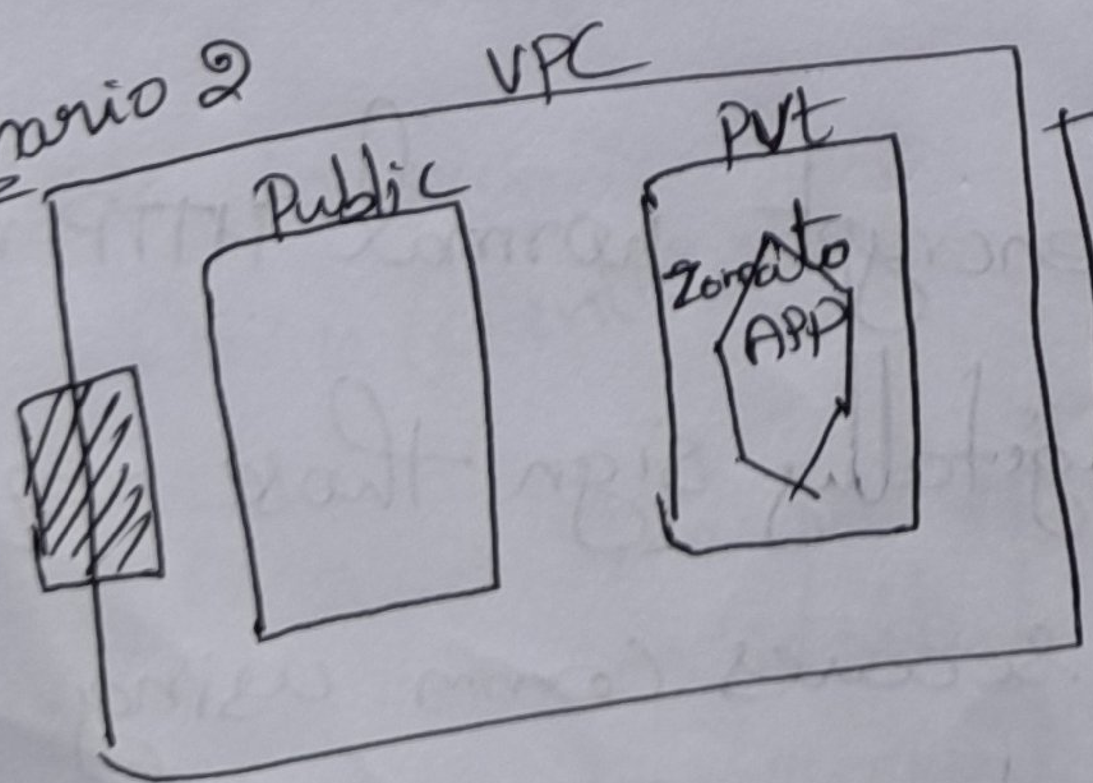
Public subnet => Internet access

Pvt subnet => No Internet access.

Public subnet -> Public IP
-> Pvt IP.

Pvt Subnet -> application -> pvt IP.

Scenario 2



Problem.

Zomato app wants to download some package through Internet

& Why they shouldn't be provide [illegible] access to the pvt subnet.

NAT gateway will change the original IP of the pvt subnet instance to the duplicate IP (Mask)*

NAT - Network Address Translation
IP Address.

*

NAT gateway	NAT instance
Costly	Cheap
High performance	Low performance.
Used in big organization	Used in start ups like.
Different method.	Deployed like EC_2 instance
100% AWS Security	50% users responsibility
Using Elastic IP	

Elastic IP => Static IP.

public IP is dynamic in nature

-> ~~we can max~~ 5 elastic IP's given ~~by~~ for an AWS account.

-> If we wanted to increase the count we shall request to AWS.

-> We have to attach the Elastic IP to an EC_2 instance for reducing cost.

running EIP => Low cost

-> EIP will not be changed.

-> We cannot attatch one EIP to multiple EC_2 at a time.

RDS → is a managed DB by AWS.

- → Patching
- → Upgrades
- → Maintainance
- → High availability
- → Scaling
- → Backups
- → Snapshots

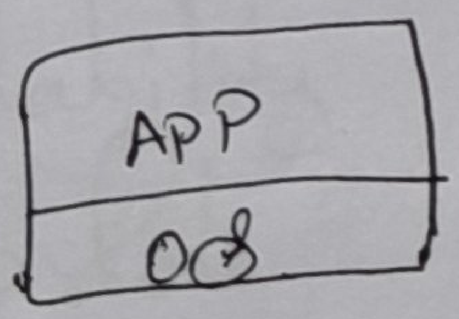
Volume	DB
unorganized	Organized

→ Within the AWS services the integration is very smooth.

→ We can integrate RDS to S_3 & EBS.

→ No need to migrate data, we can only connect.

→ RDS is a SQL based DB.

Method-1	Method-2	Method-3.
EC2 ↓ DB ↓ Data	On-Prem	AWS
		i) Upgrades ii) Patching iii) Security iv) High Availability.

28-10-2024

Cloud Trail.

-> Logs are stored in S3 bucket.
-> for that we need to provide IAM role for CT to access S3.

-> Tracks & records API activity in your AWS account. (IAM User's activity)

-> Used for logging and tracking user activities.
-> Every 15 min. it'll take logs.

RDS => Relational Database ~~System~~ Service.

Storage
Memory
Volume
Database

} Used to store data.

RDS is also a server ~~service~~

Storage => Dis organized, Ex : S3

DB => Organized, Ex : RDS

→ We can filter data in DB. And can't filter in S_3.

→ S_3	DB
Unorganized	Organized.
Any kind of data can be stored.	Any kind of data can be stored.
we can't filter	we can filter
Metrics	Metrics in dashboard. shown in graphs, diagrams, etc.)

→ RDS is a managed database service by AWS.

- ★ Security patching (Need a package for our DB, ↳ installing and AWS will download it) dependency.
- ★ Updates
- ★ Maintainance
- ★ High availability (without latency) ⇒ Through snapshots, RDS provide High availability.
- ★ Scalability
- ★ Snapshots.

~~EC2 - Elastic Compute Cloud~~

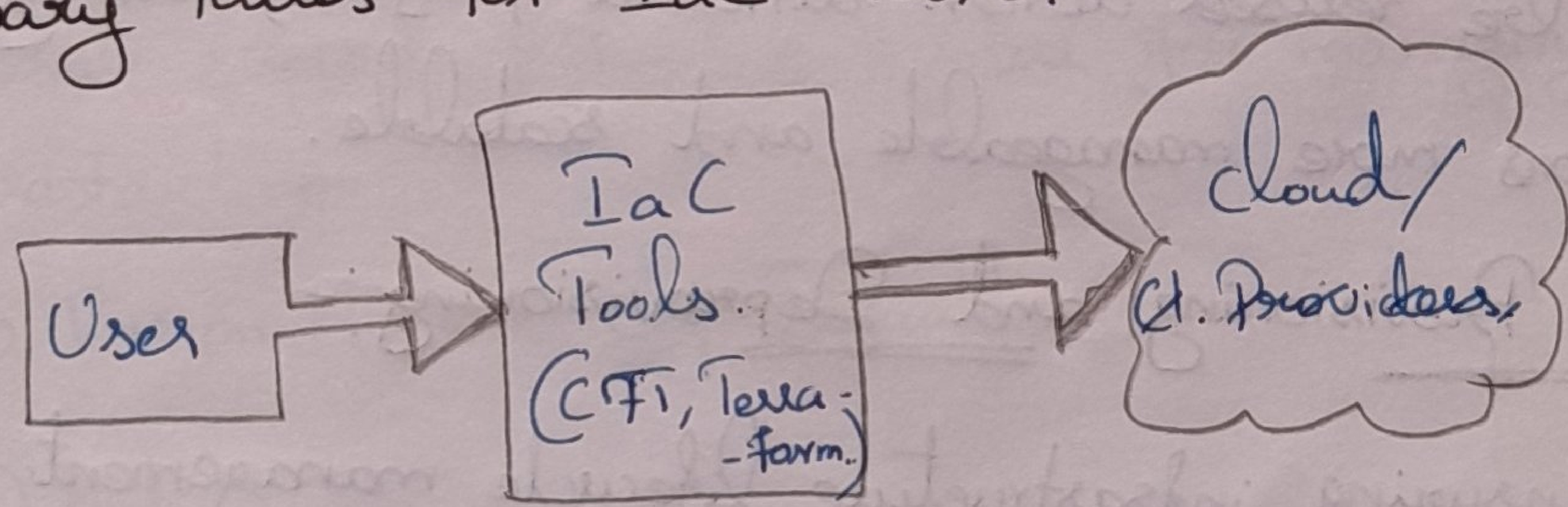
Cloud Formation :-

-> AWS Cloud Formation (CFT) is a service that allows you to model, provision, and manage AWS resources using code templates.

-> With CFT, you can define infrastructure as Code in Json or YAML file, describing the resources and their configurations.

-> It automates resource creation and management, making it easier to deploy, update, and scale infrastructu consistently across environments.

The Primary Rules for IaC tools :-



-> An IaC tool must be an interface between the user and the cloud. The cloud might be single/multi cl

Declarative or Imperative Syntax :-

-> Declarative describes the desired end state.

-> Imperative specifies the exact steps to create resources.

Cloud Formation Template

API calls are the understandable language for any cloud.

YAML	JSON
Simpler to understand	Complex to understand.
Using Indentation	Using curlybraces.

Templates contains the following features :-

optional

i) Version
ii) Description (we may define the purpose of template)
iii) Metadata (Author name & Email Id)
iv) Parameters
v) Rules (Prod, test, Dev => Naming Topology)
vi) Mappings.
vii) Conditions. (like rules)

viii) Resources. (EC_2, VPC, S_3, ... etc) => Mandatory

Idempotence :-

-> IaC should support idempotency.

Applying the configuration multiple times should result in the same outcome.

Version Control Integration :-

-> An IaC should be stored in version control.

Versioning is the process to store the previous set of codes for rollbacks & collaborative development.

Modularity & Reusability :-

-> IaC tools encourage modular templates or modules that can be reused across different projects, making configurations more manageable and scalable.

Automated Provisioning and Deprovisioning :-

-> To ensuring infrastructure lifecycle management, IaC tools should support automated deployment and teardown of resources.

Validation & Testing :-

-> IaC tools often support testing configurations before deployment.

-> Which can help catch issues early.

Why AWS CLI is not an IaC Tool?

The AWS CLI is a command-line tool that Provides a way to directly interact with AWS services by executing commands to create, modify, and delete resources manually.

-> It's used to only execute short & quick tasks.

-> Lack of State Management
  * It doesn't maintain any record of what resources exist or need to be updated.

-> No Idempotency Guarantees

-> Manual and Imperative
  * AWS CLI requires users to specify each step & doesn't support declarative syntax for managing complex infrastructure.

-> No Version Control by Default

Drift Detection :-

Drift Detection is the process of identifying differences or drift b/w the current state of infrastructure resources in the cloud and the desired state defined in IaC templates.

Key Aspects of Drift Detection :-

- → Maintains Consistency
- → Automation
- → Improved Troubleshooting
- → Security & Compliance.

Example :-

If an administrator manually changes the instance type of an EC2 instance outside of the Cloudformation template, DD will highlight this change and allowing you to decide whether to update the template or revert the instance type to its original configuration.

04-11-2024

Dynamo DB :-

-> Amazon DynamoDB is a fully managed, serverless, NoSQL database service provided by AWS, designed for high-performance applications at any scale.

↳ AWS handles all infra management, scaling & maintenance behind the scenes removing the need for users to manage servers directly

-> DynamoDB stores data in a flexible, schema-less format using tables, items and attributes.

-> Each item can have a unique structure, allowing for easy adaptation to changing data needs.

-> Mongo DB, Cassandra, Neo4j, HBase.

Key Features of Dynamo DB :-

* Highly available
* Replicas around 3 AZs
* Highly scalable
* Stores key value pairs
* Fast & Consistent in Performance
* Security & patching
* Easy to use
* No SQL Database.

Eventually Consistent Read :-

-> In DynamoDB, an Eventually Consistent read is a read option where data might not include the latest updates right away, due to replication delays.

-> DynamoDB uses eventually consistent reads by default

-> Social media feeds, product catalogs, Recommendation systems are some of the use cases of eventually consistent read.

Strongly consistent read :-

→ Strongly consistent read in DynamoDB ensures that the data returned reflects all previous writes, guaral guaranteeing that you're always reading the latest, most accurate version of the data.

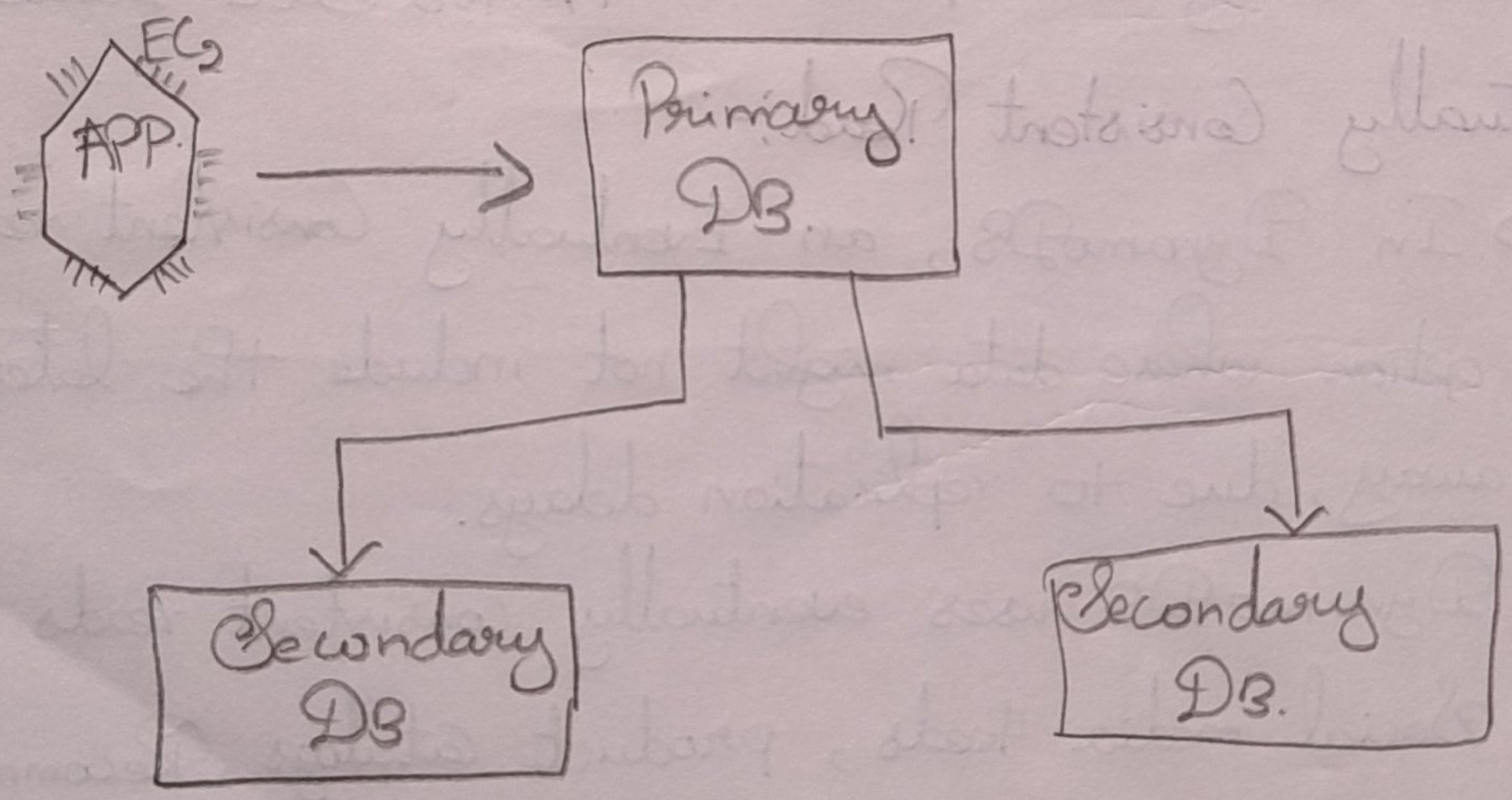
→ This type of read is especially useful when real-time data accuracy is essential.

→ Like Banking & Financial Transactions, Inventory management, order processing systems.

→ Comparing to eventually consistent read, strongly consistent read is pretty much costlier.

Notes :-

→ The application we have launched is directly connect with primary DB only.



05-11-2024

VPC Peering:-

-> Used to establish the communication between two Virtual Private Clouds.

EBS:->

-> Elastic Block Store

-> is a scalable block storage service provided by AWS, designed to work with Amazon EC2 instances.

-> Provides persistant, high-performance storage that can be used as a primary storage.

-> Provides block level storage volums for EC2 instance.

-> EBS is not highly scalable like RDS, S3 & DynamoDB

->