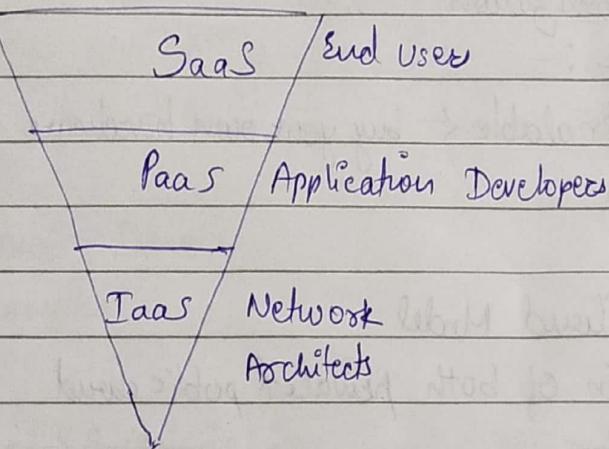


## ★ UNIT-3 - AWS (Topic 2)

★ five Essential Characteristics of Cloud Computing :-

- ① On demand self service
- ② Broad network access
- ③ Resource pooling
- ④ Rapid elasticity
- ⑤ Measured service



## 01/06/24 Cloud Deployment Model :-

### ① Public Cloud Model

- Adv :
  - Minimal Investment
  - Dynamic Scalability
  - No setup cost
  - Infrastructure Management
  - No Maintenance

### Dis-Adv

- Less Secure
- Low Customization

## ② Private Cloud Model

- One to one environment for single user.
- Adv :-
  - Better Control
  - Data Security and Privacy
  - Supports Legacy Systems
  - Customization
- Dis-Adv :-
  - Less Scalable ← buy your own hardware
  - Costly

## ③ Hybrid Cloud Model

- Combination of both private & public cloud

### • Adv :-

- flexibility and control
- Cost
- Security

### Dis-adv :-

- Difficult to manage
- Slow data transmission

IAM user : Admin  
Acc ID : 4711 1278 0810  
P → Admin @123#

S3Admin

S3admin @123

## ④ Community Cloud Model

- Adv :-
  - Cost effective
  - Security
- Shared Resources
- Collaboration & Data Sharing

P Dis :-

- Limited Scalability
- Rigid in customization

### (5) Multi Cloud Model

Adv :-

- Reduce Latency - response time is <sup>less</sup> ~~more~~
- High availability of service

Disadv :-

- Security Issue
- Complex

### \* Creating S3 Admin :-

- IAM → Users → Create Users → Add users to grp → Create grp
- User group Name → S3fullAccess
- Permission Policies → search for S3 then click on Amazon S3fullAccess

Create user group

→ Add it into the grp.

20 June/24

\* S3 Services :- S3 provides developers & IT teams with secure, durable, highly-scalable object storage. Amazon S3 is easy to use, with a simple web services interface to store and retrieve any amount of data from anywhere on the web.

- Shift
- 1) Object Based Storage
  - 2) Block Based Storage
  - ⇒ Differentiate between ↑
- types of storage
- so on 0  
→ files can be stored in 1TB  
→ virtualized storage

- \* S3 basics :-
- S3 is object based - ie allows
 

S3 is a universal namespace, That is names must be unique globally.
- HTTP Code 200 whenever the operation is successful.
- whatever you <sup>upload</sup> in your bucket will be considered as your object.

- \* Versioning in S3 :-
- Stores files in Version
- Eg :- On Monday you work on a code; when you leave your office; the US guy will work on your code; days pass; you get an error on Thursday; but you know that on Wed the # code was working properly; you know that the US guy must have made a mistake so you check the previous versions.

Objects consists of the following

Key

Value

Version ID

Metadata

Subresources;

Access Control Lists → (ACL)

Torrent;

\* How does data consistency work for S3?

- Read after write consistency for PUTs of new Objects.
- Eventually Consistency for overwrite PUTs & DELETES

how fast

means it atleast takes 1 sec to delete

Replication works → If you have 2 servers in different loc. and you upload a file on 1 server it reflects on 2<sup>nd</sup> server if you overwrite

Eventual Consistency

Read & write consistency → upload file it will immediately update but if you ~~do~~ delete or update it will take time.

\* S3 Features :-

- Tiered Storage Available → Versioning
- Lifecycle Management → Encryption
- MFA Delete → Securing your data using Access Control Lists & Bucket Policies

we want to save the file at lower cost.

- MFA can be used in S3 when if you delete a file then you have to add MFA code

\* Various S3 Tier

① S3 Standard :-

- Expensive storage type in S3 / Expensive tier
- 99.99% availability
- 99.9999999999 Durability
- You can retrieve your data from any server even if your nearest server is down. Will charge for data transfer if retrieval is from farther place.

② S3 - IA (Infrequently Accessed)

- Amazon will use availability zones
- same (last point) of ① S3 standards
- If entire region goes down you might not get the data.

③ S3 - One Zone - IA

- IA in same zone ; can be replicated
- If that zone is down the no data can be retrieved
- You will get the data once the zone is available

④ S3 - Intelligent Ticking (newest one)

- use ML Algo to determine the usage of file
- According to the analysis it will move the file according to your usage
- Can move your files between different tiers

⑤ S3 Glacier :-

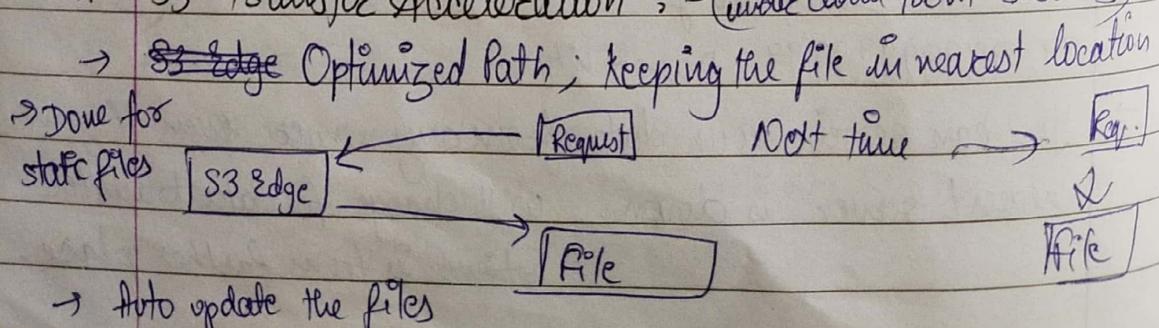
- You can store unlimited data here but retrieval data is more
- 

⑥ S3 Glacier Deep Archive :-

\* Cross Region Replication

in Amazon console

\* S3 Transfer Acceleration :- (under cloud front services)



## &gt; Permissions &gt;

- Go to your bucket > Enable ACL > to make the file public

There is a URL > bucket owner enforced

## 27/June/24 S3 Bucket Theory :-

standard has expensive  
you cannot install OS in S3.  
Has distributed data

S3 bucket versioning  
enables versioning for  
entire objects.

- ★ S3 Objects :-
- Whatever you upload in bucket is known as object.
- It is an object based storage not block based storage.
- ~~Block based is only for db, random read/write~~  
loads and structured db.
- Block storage divides the data to be stored in evenly sized blocks

Block  
based  
storage

### ★ Object Based Storage:-

- stores the file as whole and does not divide them
- object can be a file itself / data ; metadata ; object global unique ID
- Dropbox ; Aws ; FB ← eg.

About  
Buckets  
continues

You can create a file inside the bucket but cannot create bucket inside bucket

- Bucket ownership is non-transferable [even root cannot]
- S3 Bucket is Region Specific

You can share files but cannot transfer ownership.

read/write operation to whole world.

- Max cap. of Bucket is 5TB.

- Bucket must be empty before deleting it.
- You can have 100 Buckets in each account
- Once a bucket is created at specific region then you cannot change the region but can replicate.
- CRR → cross Region Replication → unless region replication can happen

### \* S3 Bucket Naming Conventions

- Bucket name acts like a username which has to be unique - unless someone deletes the name
- 3 - 63 characters long
- Underscores, special characters are not allowed

### \* S3 Bucket - Sub Resources

- S3 standards are costly.
- Lifecycle - To decide an objects' lifecycle management.
- Website → To hold configuration related to static website on S3
- Versioning →
- Access Control List →

### \* CRR - Cross Region Replication :-

- Object stored in a S3-bucket in a region will never leave their region unless you specifically do it or CRR.

→ <sup>use</sup> Individual user; AWS user; Make it public;  
authenticate all specific user

- Symbiosis Pune & Nagpur eg :

### \* Transfer Acceleration :-

- transferring files to edge loc then Buckets.

- ★ S3 Versioning Another Example :-
- You usually delete files in windows it goes to recycle bin; similarly enabling versioning helps you to retrieve data if its deleted in S3;
- Once enabling we enable versioning in S3 objects you cannot disable it; it can be suspended.
- Versioning is always enabled on S3, not objects.
- Bucket Versioning State
  - Suspend
  - Enable
  - Unversioned (Disable default)
- Once enabled ; after 2-3 updates :- understand the process
 

enabled	v - 6    X wont store
version 1	
v - 2	Enabled
v - 3	v - 7
suspended	Here the third version will be updated to 7 <sup>th</sup> version
v - 4    X	
v - 5    X wont store	
- after suspension ; you delete the file you won't be able to retrieve the file.

### ★ Multi - Factor - Authentication (MFA)

- If some body tries to
  - Delete object
  - Change MFA / Remove MFA
- It will ask code.

You must use it for objects larger than 5 GB

Date:

### \* S3 something = Multipart :-

→ ~~1TB~~ is the max capacity to upload

→ upload it by parts

→ It is used to upload an object in parts.

→ Parts are uploaded independently & in parallel, in any order.

→ Eg:- file of 500 GB

$$1\text{GB} \times 50$$

uploads 10 GB by parts ; in parallel process.

→ Recommended for objects stores of 100 MB or larger.

### \* Copying S3 Objects

— XX — XX — XX —

General Purpose — Balanced Memory & CPU

Compute Optimized — More CPU than RAM

Memory — II — More RAM

Storage — II — Low latency

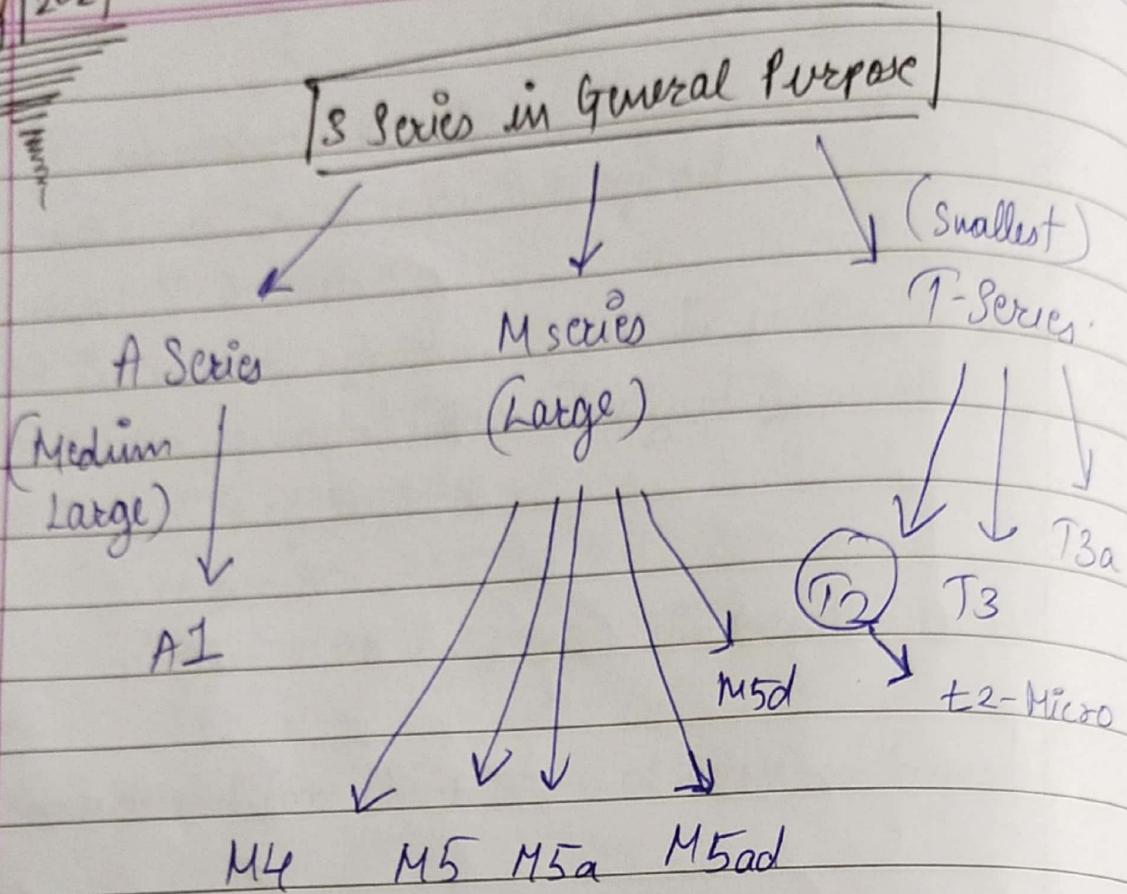
Accelerated Computing — Graphics Optimized

HMX

High Memory — High RAM/Memory system.

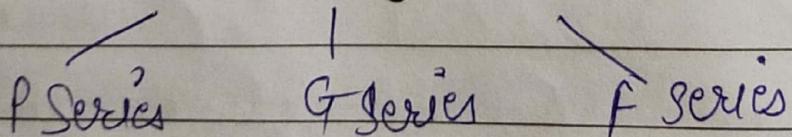
04 July 2024

Date:



- \* Memory Optimized [YT] - Based on RAM
- \* Storage Optimized Instances [YT]
  - L D2 Instances
  - H1 Instances
  - I3 & I3 en Instances

- \* Accelerated Computing Instances



## \* EC2 - Instances Purchasing options.

→ On Demand - costly → Spot Instance.

→ Dedicated Instance & Dedicated Host

→ Reserved - Standard RI  
Instance

- Convertable RI  
- Schedule RI

} Types

You can convert  
(instance or v)

If you buy 1 instance  
and you get T2 Nano now  
you cannot convert you have  
to use it

→ You don't need the server everyday

→ Eg - Attendance you don't need it ; once in a month

- Employee Attendance - keep server on during :-  
(specific timings) :-

8:30am - 9:00am } monday -  
4:30pm - 5:00pm } friday

## \* On demand & Spot Instance :-

→ Linux:- Once you click on start billing starts even if  
the EC2 is booting

→ Windows - charges for 1hr ; Even if you use  
for 15 minutes you will charge for 1hr,

# EC2 Instances Purchase Options

On demand

spot  
Instance

schedule  
Instances

dedicated  
Instances

Dedica-  
ted  
HOST

Reserved  
Instance

standard  
R.I

convertible  
R.I

schedule  
R.I

★ connect > connect

→ Launch Instance  
then check if you can connect ; If not  
then download MobaXterm

① sudo su -  
| switch user  
switch user do

② yum update -y [to update]  
③ yum install httpd -y [Apache server on Linux]  
④ service httpd start [start the server]  
|  
to stop → service httpd stop

⑤ chkconfig httpd on  
⑥ cd /var/www/html  
⑦ echo <html><h1>Hello</h1></html>

|  
to exit  
OR nano ; ctrl x & then save

HTTP

mb → make bucket

HTTPS

cp → to copy

SSH

Stop / terminate

## ★ CIDR :-

- Classless Inter Domain Routing
- IP Addresses - types
  - ↳ Physical IP
  - ↳ Logical IP

### ① Physical IP Address

→ you go on road & ask "who is happy"; people will not know who the person is; But if you ask about Kores Towers people will show you the route

→ In IT terms :-

Physical Address is MAC Address  
48 Bit Address

Logical is IP Address

versions

IPv4

32 Bit

versions

IPv6

64 Bit

Roster

It is a internetworking device; knows who is the sender & receiver;  
if he finds the two

Public  
IP  
Address

Private  
IP Address

similar networks

it gets confused;  
that is what we call  
it as switch.

## ★ ★ IPv4 :-

- ipconfig :-
- IP helps you in travelling from one place to another
- 32 Bit Address
- 192.168.10.10 ← this is a 32 bit address

why?

192	168	10	10
↓ 8	↓ 8	↓ 8	↓ 8

8 bit division format

(Octet) = 4 octets

① Here we have 4 octets

② 255.255.255.0      This is a subnet

and why 255? why not 255.255.0.0



You can connect 256 devices

from  $\Rightarrow$  0-255 but you

can connect 254 devices because

your network id 192.168.0.0 & then

the last network id is 192.168.0.255

## ★ IP classes

192.168.10.10

→ what class of IP Address is this?

Imp

Class A : 1.0.0.0 - 126.0.0.0

Class B : 128.0.0.0 - 191.255.0.0

Class C : 192.0.0.0 - ~~223~~ 223.255.255.0

Multicast Class D : 224.0.0.0 - 239 } Reserved

Research Class E : 240 - 255

To test NIC :- Network Network Interface Card

127.0.0.0 =

Ping 127.0.0.0

→ One more missing IP :- 0.0.0.0/0  
 and 127.0.0.0 → missing IP's

Theoretically it is in class A

→ 137.20.20.10 ← class B

Q you check for first octet & compare it with range.

Q What is a Network ID & Host ID?

→ 15.200.200.20 (Answer)

→ Class A ; Here Network ID :-  $\boxed{N|H|H|H}$   
 Host ID :-

So; Network ID :- 15

Host ID 200, 200, 20

Representation :-

Network ID : 1

Host ID = 0

$\boxed{1|5|0|0|0}$

Since its class B :-  
2 octates are reserved  
in class A; 1<sup>st</sup> octate is reserved

∴ Network ID is represented by 1  
and Host ID by 0.

Class A	N	H	H	H
class B	N	N	H	H
class C	N	N	N	H

B

∴ Network host :-

Eg 2 Network host will not change

~~172 | 128 | 0 . 0~~ class B

↓

Next IP

172 | 128 | 0 | 1

↓

172 | 128 | 0 | 255

↓

172 | 128 | 1 | 0

172 | 128 | 1 | 1

172 | 128 | 1 | 255

172 | 128 | 2 | 0

$\Rightarrow$  115. 0. 0. 0 ← class A  
 Binary / subnet mask [Octet 1]

11111111, 00000000, 00000000, 00000000  
 255. 0. 0. 0

$\Rightarrow$  192. 16. 1. 0 — class B  
 [2 octets]

11111111, 11111111. 00000000. 00000000  
 255. 255. 0. 0

- Private      Public
- No need to buy      need to buy

- You get public IP address for a website ; static IP needs to be there because Amazon changes it (public wala)
- public IP changes by Amazon
- private you have to buy
- so you buy a website domains → you actually host using IP next day your IP changes but your domain is same.

→ class C bits reserved 8  
 class B → 16

$$2^{16} = 65536 - 2 \\ = 65534$$

class B you can connect upto 65534  
 Bits reserved in class A : -  $8 \times 3 = 24$

$$\therefore 2^{24} = 11677216 - 2 \\ = 11677214$$

17 July '24



CIDR

N/W Bit represented by 1  
Host Bit - 11        0

Class A :- 1 Network bit + 3 Host bit

| N | H | H | H | - ~~SM~~

Class B

| N | N | H | H | 2 Network bit 2 Host bit

Class C

| N | N | N | H | 3 Network bit 2 Host bit

① 15.200.20.20

- Class A

- Network ID of this  $\rightarrow$  15.0.0.0

- Network Bit  $\rightarrow$  ~~15.0.0.0~~ N/W - 1  
Host - 0

② 196.128.1.10

- Class C

- N/W ID  $\rightarrow$  196.128.1.0

- Binary format  $\rightarrow$  11000100.10000000.11

$2^7 \quad 2^6 \quad 2^5 \quad 2^4 \quad 2^3 \quad 2^2 \quad 2^1 \quad 2^0$

128 64 32 16 8 4 2 1

196 1 1 0 0 0 1 0 0

128 1 0 0 0 0 0 0 0

1 0 0 0 0 0 0 0 0

0

$\therefore$  Bit format  $=$  11000100.10000000.00000000.

Ans = 11000100.10000000.00000000

S = 011000100.10000000.00000000

Ans = 11000100.10000000.00000000

★ Subnetting :- creating N/W inside N/W

① 115.10.10.20

- class A

- N/W ID - 115.0.0.0

- Subnet mask - ~~115.255.255.255~~

↳ To find

① 1<sup>st</sup> octet is N/W

② Find class

③ Converting host bit to Zero

④ Check N/W octet ie 1

⑤ Write SM ↴

255.0.0.0

② 172.20.20.20

- class B

- N/W ID = 172.20.0.0

- Subnet Mask = 255.255.0.0

Because Host Bits = 0 ; N/W B → 1 [Representation]

→ In Binary :- [255.255.0.0]

1111111.111111.0000 0000.00000000

③ 192.168.0.0

- N/W ID :- 192.168.0.0 - class C

- SM - 255.255.255.0

- Computers that will connect 254

- Broadcast ID (last IP) - 192.168.0.255  
for pc.

→  $172 \cdot 17 \cdot 10 \cdot 10$  } this is a separate network  
 $+ 172 \cdot 16 \cdot 10 \cdot 10$

- Router is an internetworking device for the network device

#

→  $10 \cdot 0 \cdot 0 \cdot 0 \rightarrow$  class A }  
 $172 \cdot 16 \cdot 0 \cdot 0 \rightarrow$  class B } Private IP Addresses.  
 $192 \cdot 168 \cdot 1 \cdot 0 \rightarrow$  class C }

→ network within a

→ Subnetting means creating network within a network.

18 July '24

### ★ IPv4

- Router is an internetworking device not an networking device

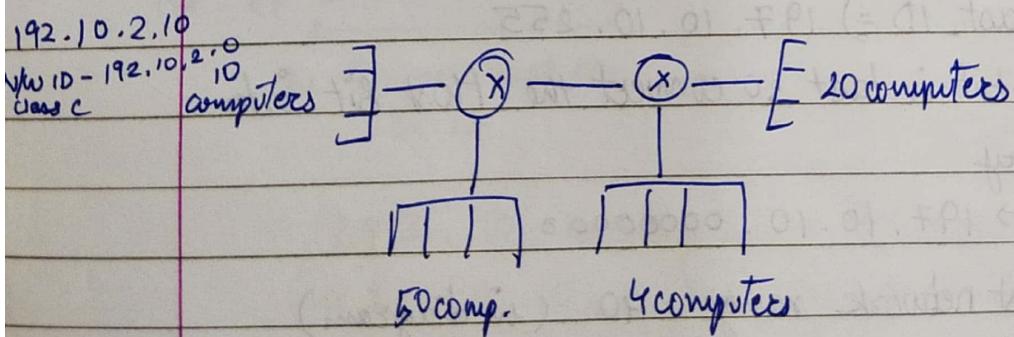
→ 3 methods for

① Separate Private addresses

② IPv4 is big - because it has large range

③ Do subnetting

↳ dividing the network logically



- ★ In any organization there are 3 IP ranges
- 192.168.1.0 - for private org Class C
  - 172.16.x.x - for class B
  - 10.x.x.x - for ~~per~~ class A

Q Why is it needed? Private IPs?

- ① Public IP is expensive; One public IP for entire org. so that it can communicate outside the network.
- ② IP is the only way computers communicate.
- ③ Network NAT that converts the request b/w private & ~~com~~ public IPs.
- ④ Router is responsible for NATING;

New Eg 15.10.20.30

- class A
- N/W ID : 15.0.0.0
- SM :- 255.0.0.0
- Broadcast ID :- ~~15.255~~ 15.255.255.255  
(last IP of this range)

★ How many host can we connect in class A :-

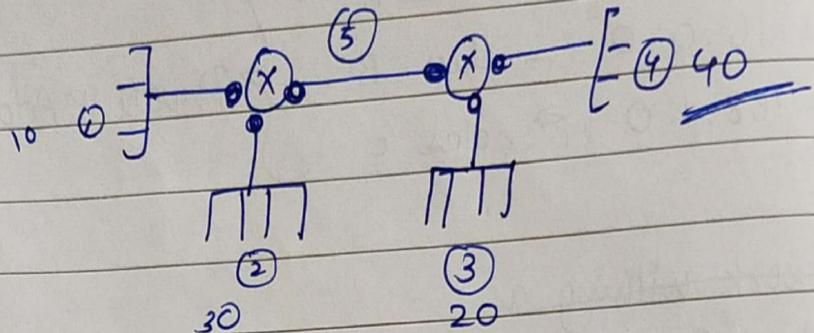
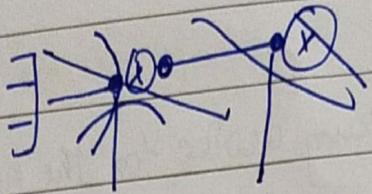
- 192.168.0.0
- class C ; - Bits  $2^{24}$   $\Rightarrow$  192.168.0.0/24
- Devices - 254 devices  $192.168.0.0/20$  - only 20 bits are there

~~B~~  $\Rightarrow$  class B :- 172.168.0.0  $\Rightarrow 2^{16} - 2 = 65536 - 2$   
= 65534

class A  $\Rightarrow$  10.168.0.0  $\Rightarrow 2^{24} - 2 =$

Eg:-

15.10.0.1



We need 5 networks

- 15.10.0.1/8
- Now let's create a network (steps to follow)
  - ① No of networks = 5
  - ② IP address given = 197.10.10.0  
default subnet = 29

### Step I

Detect class IP :-

class C

N/W ID :- 197.10.10.0

Subnet mask =) ~~255~~ 255.255.255.0

Broadcast ID =) 197.10.10.255

### Step II

Host bit is last so convert the Host Bit into

Binary

↳ 197.10.10.00000000

Highest network range = 40 (in diagram)

$$2^n - 2 \Rightarrow 2^8 - 2 = 256 - 2 \\ = 254$$

n = no. of devices

Step

(3)

$$2^n - 2 \gg 40$$

$$64 - 2 = 62$$

$$62 \geq 40 \quad \checkmark$$

$$\therefore n = 6$$

because of highest net/w

$$2^1 = 2; 2-2 = 0 \times$$

$$2^2 = 4; 4-2 = 2 \times$$

$$2^5 =$$

Step (4)

197.10.10.11 ~~10000000~~

~~~~~

6

197.10.10.00000000

6 [keeping 6 bits at host]

remaining 2 give it to net/w]

$128 + 64$

subnet ID

197.10.10.11 0000000

→

197.10.10.192

← your new subnet ID

~~255.255.255.255~~

~~197.10.10.192/26~~

your new subnet mask

1 subnet mask

Subnet ID :-

Subnet ID 1

197.10.10.64/26

Subnet ID 2

$$2^n = 2^6 = 64$$

Ans 2

|     |    |   |   |   |   |   |
|-----|----|---|---|---|---|---|
| 1   | 1  | 0 | 0 | 0 | 0 | 0 |
| 128 | 64 | 0 | 0 | 0 | 0 | 0 |

the first ~~subnet~~ ID  
is 64

∴ 197.10.10.63/26 ← Broadcast ID

197.10.10.64/26 ← N/W ID

∴ Subnet ID 3 :-  $64 + 64 = 128$

∴ 197.10.10.127/26

∴ 197.10.10.127/26 Broadcast ID

Subnet ID :-

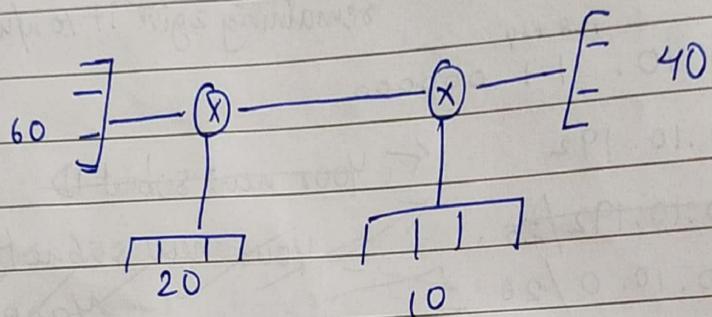
$$128 + 64 = 192$$

192.10.10.192 - New ID  
192.10.10.191/26

No. of networks we can create 4

\* with class class A IP address :-

12.0.0.0



- class A

- No of N/W - 5

- N/W ID - 12.0.0.0 [ octate - 1<sup>st</sup> octate for N/W ]

Step II - convert int binary

12, 00000000, 00000000, 00000000

Step III largest N/W = 60

$$2^n - 2 \geq 60$$

$$2^6 - 2 = 64 - 2 = 62$$

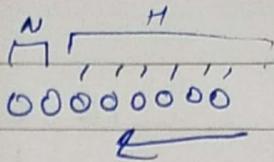
$$n=6$$

Keep 6 bits with the host

∴ Max waste of IP = 2

Step IV

12. 00000000 , 00000000 . 00000000

~~12. 000~~

12. 1111111. 1111111. 11000000

12. 255. 255. 192 / 26

$$\begin{array}{r} 12 \cdot \underline{1111111} \cdot \underline{1111111} \cdot 11000000 \\ \hline 8 + \underline{8} + \underline{8} + \underline{2} \end{array}$$

New subnet mask :-

12. 255. 255. 192 / 26

when ;

subnet mask

12. 0. 0. 0 / 8 - 12. 255. 255. 255

/ 26 - ~~255~~. 12. 255. 255. 192

∴ New subnet ID :-

12. 0. 0. 64 ← subnet ID

G3 ↓ ← Broadcast ID

12. 0. 0. 128      ~~Subnet ID~~12. 0. 0. 192      ~~Subnet ID~~12. 0. 1. 0      ~~Subnet ID~~~~12. 0. 1. 192      Subnet ID~~

12. 0. 1. 64

12.0.1.128

↓

12.0.1.192

↓

12.0.2.0

↓

12.0.2.64

↓

12.0.2.128

↓

12.0.2.192

↓

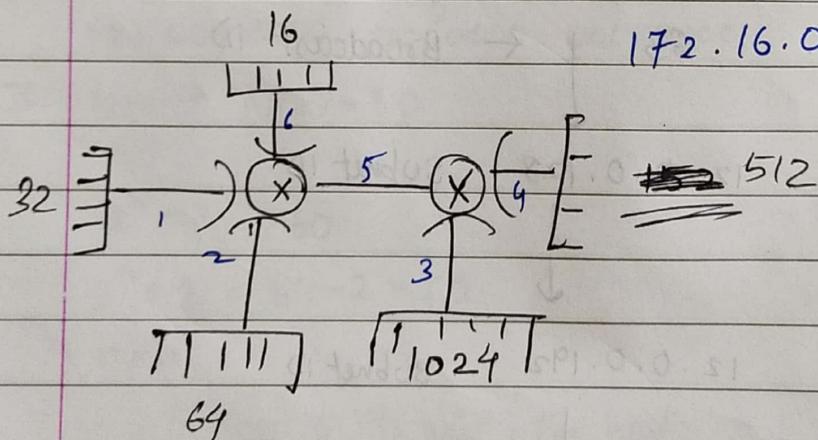
12.0.3.0

~~12.255.255.255~~

12.255.255.192

// last IP

172.16.0.0



Class B ; No. of N/W = 6

172.16.0.0  
N      H

II convert into binary :-

172.16.00000000.00000000

$$\text{III } 2^7 - 2 = 2^6 - 2 = 2048 - 2 \\ [n=11] = 2046$$

$$\therefore 172.16.00000000.00000000$$

$$\begin{array}{r} 128 \\ + 64 \\ + 32 \\ + 16 \\ + 8 \\ \hline 248 \end{array}$$

$$\therefore 172.16.1111000.0000000$$

$$\therefore 172.16.\cancel{248}.0$$

$$\therefore 172.16.248.0/21$$

How?

$$172.16.0.0/21$$



$$172.16.7.255$$



$$172.16.8.0$$

Broadcast

~~25 July 24~~

Q1 what is cloud computing & mention its characteristics of cloud computing?

Q2 what do you understand by SaaS explain with example

Q3 ————— || ————— PAAS ————— ||

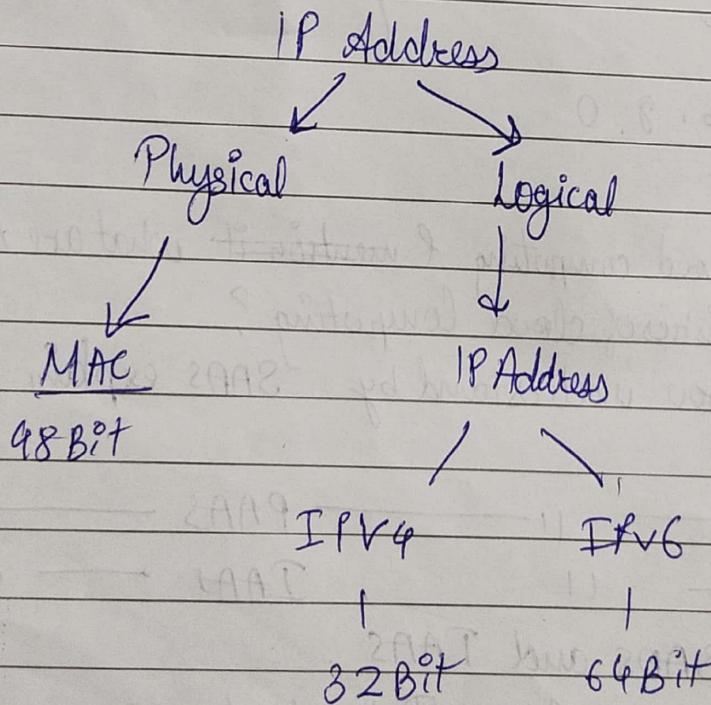
Q4 ————— || ————— IAAS ————— ||

Q5 Compare SaaS and IAAS.

Q6 Write a note on AWS Infrastructure?

Q7 What is S3 in AWS; Explain following terms related to S3 → Buckets, Objects, Life cycle Management system in S3

- Q8 What are the diff tiers in S3 and life cycle  
 unqut policies for each tier. - 60yrs + eg
- Q9 what do you understand by EC2.  
 what are the diff. instance types.
- Q10 Explain below terms related to EC2:-  
 ① Security groups  
 ② Image management  
 ③ Key Pairs  
 ④ EBS
- Q11 Differentiate b/w object based storage & block based storage. (any 5 points)
- Q12 Explain ~~adv~~ of IAM services. Explain authentication features and authorization.
- Q13 Write a note on IP classes and subnets.



Subnet Mask :-

Class A - 255.0.0.0 -  $2^{24} - 2 = 16777214$

Class B - 255.255.0.0  $\rightarrow 2^{16}$

Class C - 255.255.255.0

Class A

$$2^{24} - 2 = 16777216 - 2 \\ = 16777214$$

Class C

$$2^8 - 2 = 256 - 2 \\ = 254$$

Class B

$$2^{16} - 2 = 65536 - 2 \\ = 65534$$

calculating subnets

(host req.)

You have to buy IP addresses; it's expensive

Example

\* 157.110.88.0 Consider an IP of 157.110.88.0  
subnet - 30 30 subnets are needed;  
Each subnet - 64 Hosts (each subnet may have 64 hosts)  
(concern point of view)

$\begin{array}{c} 1 \\ 2 \\ 2 \\ 2 \\ 1 \end{array}$

16 bit 16 bit  
Class B: H = 2 ; N/W = 2

157.110.00000000.00000000

$\therefore 2^n - 2 \geq 30$ ; n=5

(157,110.11110000.0000 0000

157,110.248,0/21

157.110.0.0  $\rightarrow$

157.110.16,0/21

157.110.24,0/21

157.110.32,0/21

157.110.40,0/21

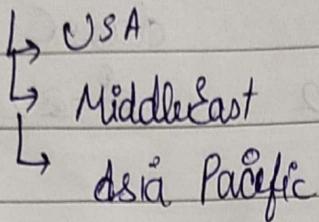
157.110.8.0/21] - 2  
157.110.7.0/21  
157.110.16.0/21

157.110.48.0/21

## \* Google Cloud :-

- Why Russia doesn't have a google DC?  
↳ Economical + Geographical reasons

→ Google has divided their infra in 3 parts :-



USA | ME | ASPC

Delhi  
Mumbai

→ Resources

- Global
- Regional
- Zonal

## \* Multi-Regional Services

→ One region goes down services will still work  
just there will be a delay while accessing  
any resources due to change of region

→ One Zone datacenter (discussed previously)

\* Every region will have 3 zones

Mumbai  $\leftrightarrow$  3 zones

Delhi  $\leftrightarrow$  3 zones

- Pova has 4 zones; Birthplace of Google C.P.
- fully qualified name for a zone is made up of
  - └ <region><zone>
  - └ GCP you cannot change use the name again  
(bucket case in AWS)
- ↗ <region><zone>
- Cross replication in GCP
  - └ in 2003; whatever you search in Google you used to have latency as the request were sent to US; now the servers are placed in India  
~~so the latency~~

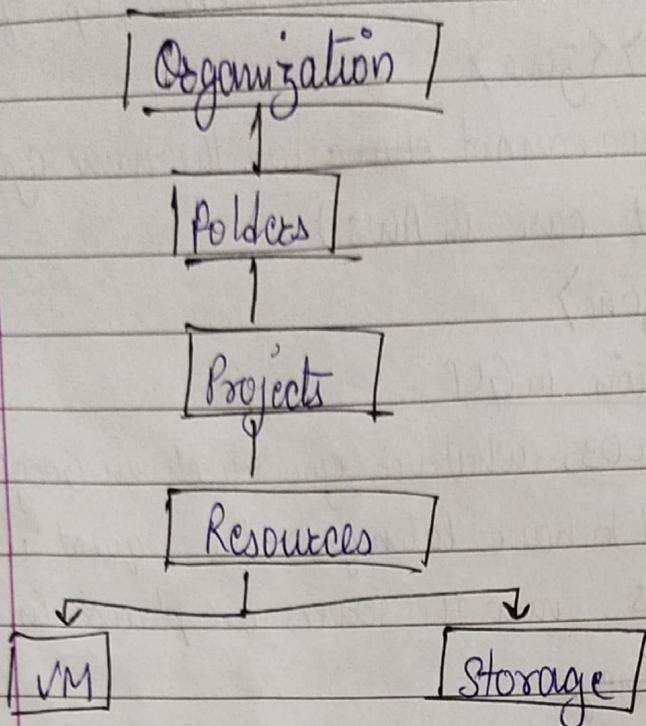
more latency less response time

- Nearest DC ~~is~~ for Pune :- Mumbai.
- More distance more charges ← redundant system.

#### \* 4 Basics of Cloud

- ① fault Tolerance - when a server fails, cloud detects the fault in the server and then it will replicate the server
- ② High Availability → Ensuring the services you are using should ~~to~~ work
- ③ Latency
- ④ ~~Replication~~ Redundancy → Replication

## \* Resource Hierarchy in GCP



- \* Folders :- one folder one CTO;
- \* Projects :- Multiple projects; payment for these projects will be paid by organization.  
Entire organization bill.
- \* Resources :- one resource allocation to one project
- \* Multiple Resources connected to one project or multiple Res... conn - many project
- \* Project can be built
- \* The purpose of the GCP Resource Hierarchy is two folds.
  - Provide a hierarchy of ownership, which binds the lifecycle of a resource to its immediate parent in the hierarchy.
  - ...

→ Inheritance

C:/Users/username/Documents/trip

↑  
Deleting this will delete subfolders below it

Organization

Company

Folder

Department X / Department Y

Shared Infrastructure

Team A

Team B

Product 1.

Product 2

Project

Development Project

Test Project

Production Project

Resources

~~Compute~~  
Engine  
Instances

App Engine  
Services

Cloud  
Storage  
Bucket

There can be folders inside ~~folders~~ folders but not for projects.

- ★ Benefits of the Organisation Resources.
- With an organization Resource, projects belongs to your organization instead of the emp. who created the project. This means that the projects are no longer deleted when an emp. leaves the company.
  - You can grant roles

14<sup>th</sup> Aug '24

## Search Engines

- Huge db resources such as web pages, newsgroups, programs, images etc.

3 components in Search Engine :-

(3 parts)

① Crawling



② Indexing

③ Ranking — has 3 parts

→ Analyze user query

→ finding matching pages

→ Present the results to the users.

- ① Analyze user query :- understand what user is searching for
- ② Finding matching pages :- find and present results; 10 ~~say~~ results per page; will popup advertisements

### \* Web Crawler :-

- what is web crawler
- functionality :-
  - ① Discovery - web crawler are also known as spiders or bots,
  - ② Retrieval - now the w/p is discovered by the crawler; retrieve contents including HTML, images, text, images, & other resources.

### ③ Parsing -

- ④ Filtering - filtering duplicate info; unwanted info is removed

### ⑤

### \* Key features :-

- ① Scalable Crawling

### → Underlying Technologies :

- 5 points
  - ① Distributed Crawling Frameworks
  - ② URL frontier Management
  - ③ HTTP Protocol Libraries
  - ④ Data storage & persistence
  - ⑤ Crawling policies & Configuration

Date:

ADSL → Asymmetric Digital Subscriber Line

★ Database :-

- All info on web is stored on database.

★ Search Interfaces :-

22 Aug 24

Q What is Routers?

Q What is Gateways?

Q Diff. b/w Routers & Gateways.

→ When 2 computers wants to connect with each other within same n/w, switch helps, router will allow to connect as ~~router~~ it works on identifying the source & destination.

→ The routers provided at home are ADSL that works like router;

→ DHCP - Dynamic Host Configuration Protocol

→ NAT - N/w address Translation ; - converting private IP to public IP & viceversa ; eg. there are 4 computers in a n/w - this helps routers to send response to the computers.

\* functionality of Internet Gateway :-

→ configuration of Router ; doesn't known as a diff devia

\* DNS - Domain Name Server

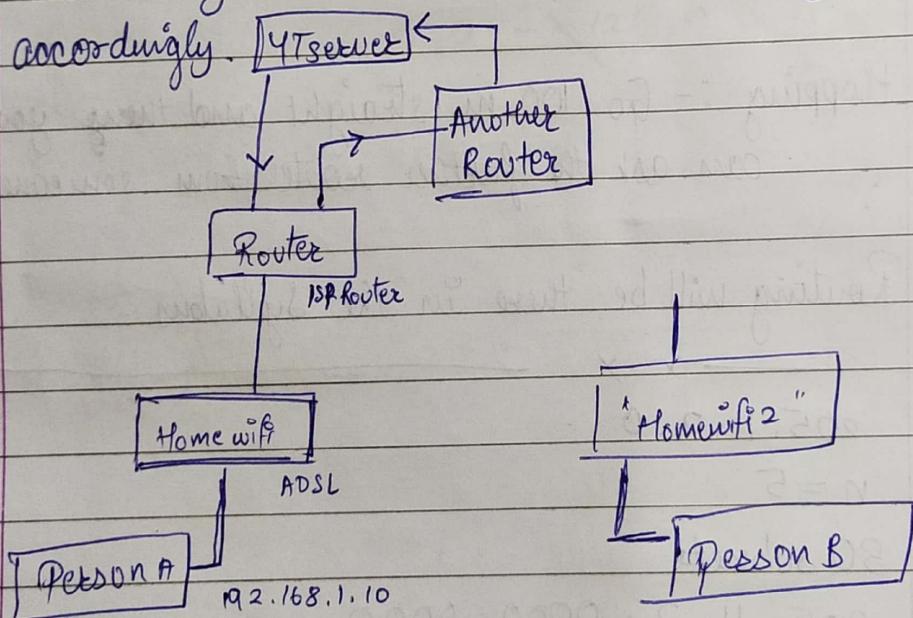
→ converter www.google.com → IP Address

\* Routing table :-

→ It will have the IP ; the address

→ Protocols is the common language these routers use to comm with each other.

→ If you send request to search for www.google ..  
and the router will communicate with the  
nearest router present nearby to www.google's server  
and then your router will update its Routing table  
accordingly.



① You search for www.google.com

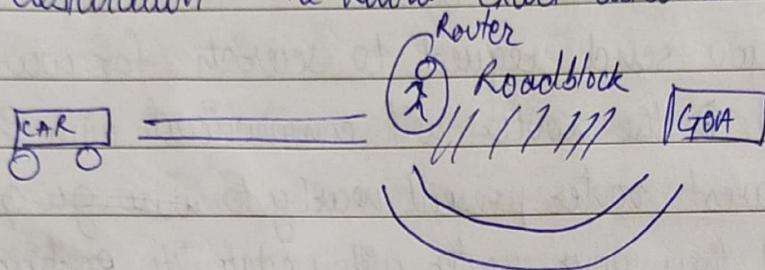
② Every router has a routing table which contains address of the nearest router

③ Your private IP will be converted into public at "Home wifi"

- ④ Even the second router has the routing table
- ⑤ The 1<sup>st</sup> Router will connect to another router to calculate the destination; it's searches inside the table.

### New Eg

- ① You're planning for a trip and on the way there's a road block; a construction person says that there's an alternate route there you can reach your destination - he knows exact alternate route



- ② Hopping :- Go 100 m straight and then you can ask the further route from someone else,

\* Routing will be there in the syllabus.

— X X —

205.11.2.0

n = 5

30 subnets

Bin ↓      Hex ↓

205.11.2.0000 0000

205.11.2.1111 0000 → Zeros

205.10.2.248 /29      24+3

205.10.2.0 /29

205.10.2.8 /29

16

32

$110 \cdot 8 \cdot 8 \cdot 0 - 30$  Subnets

$$n=5$$

$157 \cdot 110 \cdot 88 \cdot 0/16$  30 sub Doing for  
 $n=5$ ; Class B Host  
 $16+5 = 21$

$157 \cdot 110 \cdot 88 \cdot 21$

$\boxed{0000 \ 0000}$

Subnet Mask

$157 \cdot 110 \cdot 88 \cdot 0$

$157 \cdot 110 \cdot 8 \cdot 0$

~~class A - 1.0.0.0 - 126.0.0.0~~

~~class B - 128.0.0.0 - 223.255.0.0~~

~~class C - 224.255.255.0 - 255.255.255.0~~

~~class D -~~

~~class E -~~

~~class A - 1.0.0.0 - 126.0.0.0~~

B  $128.0.0.0 - 195.255.0.0$

C  $196.0.0.0 - 223.255.255.0$

D  $224.0.0.0 - 239.255.255.255$

E  $240 - 255.255.255.255$

★ Revision  $\Rightarrow$  Last Class :-

\* IP Addressing

① Physical  $\Rightarrow$  Addressing  $\Rightarrow$  MAC address - 48 Bits

② Logical  $\Rightarrow$  IP Addressing

Private IP Addresses :-

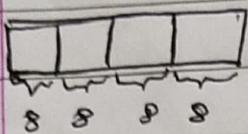
192. x. x. x

IPV4

IPV6

32 bit Address

128 bits



Class A 1. 0. 0. 0  $\rightarrow$  126. 0. 0. 0  $\rightarrow$  255. 0. 0. 0 /8

B 128. 0. 0. 0  $\rightarrow$  191. 255. 0. 0  $\rightarrow$  255. 255. 0. 0 /16

C 192. 0. 0. 0  $\rightarrow$  223. 255. 255. 0  $\rightarrow$  255. 255. 255. 0 /0

D 224. 0. 0. 0  $\rightarrow$  239

E 240. 0. 0. 0  $\rightarrow$  255

\* 10. 0. 0. 0

- Default subnet mask  $\Rightarrow$  255. 0. 0. 0

- 1 octate for Network ; 3 for hosts

- Bits with network  $\Rightarrow$   $2^8$   $\rightarrow$  166777216

- Bits with host  $\Rightarrow$   $2^{24} = 166777216$  [mostly asked]

- for class B  $\Rightarrow$  N/W  $\Rightarrow$   $2^{16}$

$$\text{Host} = 2^6 = 65536$$

for class C  $\Rightarrow$  N/W  $\Rightarrow$   $2^{24}$

$$\text{Host} = 2^8 = 256$$

Page No. \_\_\_\_\_  
Date: \_\_\_\_\_

|       |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|
| $2^7$ | $2^6$ | $2^5$ | $2^4$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ |
| 128   | 64    | 32    | 16    | 8     | 4     | 2     | 1     |

Example

\* IP 115.10.10.20

→ class A.

Subnet mask  $\Rightarrow$  255.0.0.0

→ Bits reserved for N/W  $\Rightarrow$  8

convert into binary

01110010

from host side

\* Subnetting

from network

$\Rightarrow$  115.10.10.20

Type  
of  
Q.

calculate no. of subnet

(2) →

provided subnets

9.

$\Rightarrow$  115.0.0.0

$\Rightarrow$  115.1.0.0

$\therefore \Rightarrow$  115.10.0.0

} belongs the same network.

\* Private IP Addresses

192.x.x.x

16.10.20.200

172.16.x.x

172.31.x.x

02 < 2-2

02 < 2-2

02 < 2-2

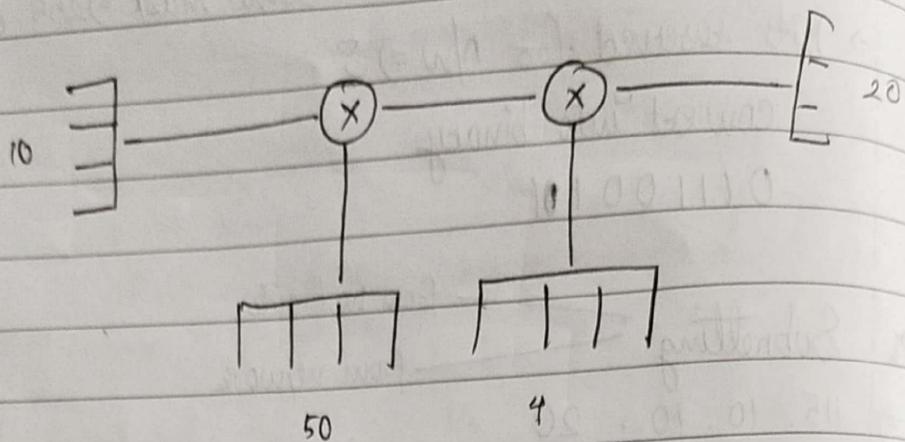
3 = a

of this way this method is termed as

private addressing

unit 3 + 4

## \* Class Test Question



- Max no. of computers = 50 (maximum no. of hosts) (given)
- Max no. of networking / connection / wires / etc = 5
- IP Address =) class A

Step I) Identify the class from the IP address given

|                |                   |                  |
|----------------|-------------------|------------------|
| 15.10.0.1      | class A           | Default subnet/8 |
| Host bits =    | { N   H   H   H } | Subnet mask      |
| Network bits = |                   |                  |

15.10.0.1/8

→ Convert N into binary & Host as 0

15. 0000 0000 . 0000 0000 . 0000 0000

Step II find out the maximum no. of hosts → "given 50"

$$\begin{array}{ll} 2^n - 2 \geq 50 & 64 - 2 \geq 50 \\ 2^6 - 2 \geq 50 & 62 \geq 50 \end{array}$$

$n = 6$  ; calculating broadcast ID

now requirement is from host side ; you go right to left 6 times

15. 0000 0000 . 0000 0000 . 000 000000

15. 1111 1111 . 1111 1111 , 11 000000

15. 255 , 255 , 192

15. 1111 1111 , 1111 1111, 11000000

$$\cancel{8+8+8+2} \rightarrow 8+8+8+2 = 26$$

Step IV

Last subnet ID =)

$\rightarrow 15. 255. 255. 192/26$  subnet ID last

New subnet mask 255. 255. 255. 192 /26

Net subnet ID List

~~15. 0. 0. 64 /26~~

~~15. 0. 0. 128 /26~~

15. 0. 0. 0 /26 } gap = 62 are there then we are

15. 0. 0. 64 /26 } using 50 because of max

15. 0. 0. 128 /26 devices so we are wasting

15. 0. 0. 192 /26 (= 91 IP addresses)

15. 0. 1. 0 /26

15. 0. 1. 64 /26

15. 0. 1. 128 /26

15. 0. 1. 192 /26

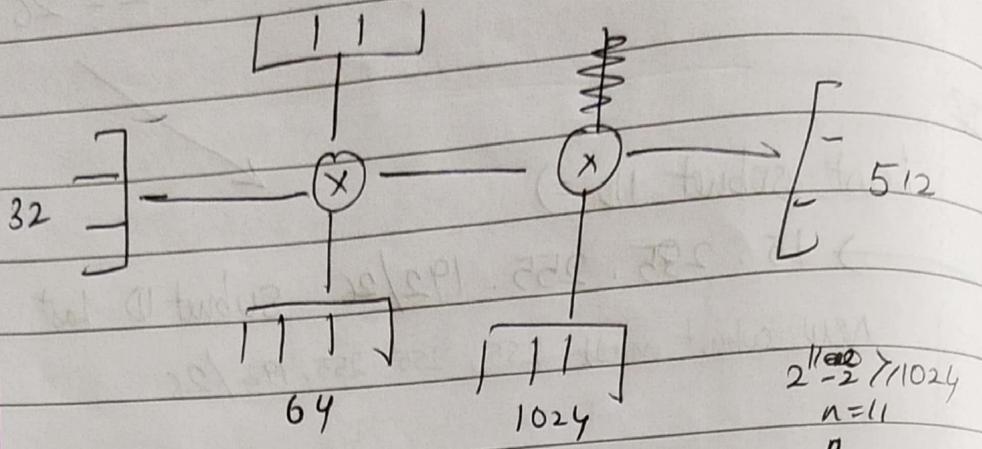
15. 0. 2. 0 /26

:

15. 255. 255. ~~192~~ /26 (= 190 IP addresses)

12.0.0.0/6

16



Class A

12.0.000.0000, 0000.0000.00000000

12.111111.1111000.00000000  $\Rightarrow 8+8+1$

12.255.248.0

~~255.255.~~

Last subnet 1 $\Rightarrow$

12.255.248.0/21

~~255.255.248.0/21~~

Broadcast ID  
12.255.248.255  
 $\Rightarrow$  1 $\rightarrow$  12.0.0.0/21  
 $\Rightarrow$  2<sup>nd</sup> subnet  $\Rightarrow$  12.0.8.0/21

12.0.7.255/21

3<sup>rd</sup> subnet  $\Rightarrow$

thus are asking

new point of view [refer saanchi's Q] ~~new~~ [class test]

Consider the IP 157.110.88.0 we ~~had~~ 30 subnets

\* Class B  $\leftarrow$  Step I

Step II

Convert Host bits to 0

157.110.00000000.00000000

$$\text{Now } \cancel{2^n} - 2 > 30$$

$$32 - 2 > 30 \quad n = 5$$

$$30 > 30$$

Now networking is taking so left to right  
while previously host was giving

$$\cancel{2^3} = 8$$

157.110.11110.00.00000000  $\because n=5$

$\therefore 157.110.248.0/21$  Last subnet ID  $\therefore \rightarrow 5$  times

157.110.0.0/21 - ①<sup>st</sup>

157.110.8.8/21 - ②

1<sup>st</sup> bit that  
got converted  
 $2^3 = 8$

16

24

32

40

48

255.255.248.0

Subnet mask

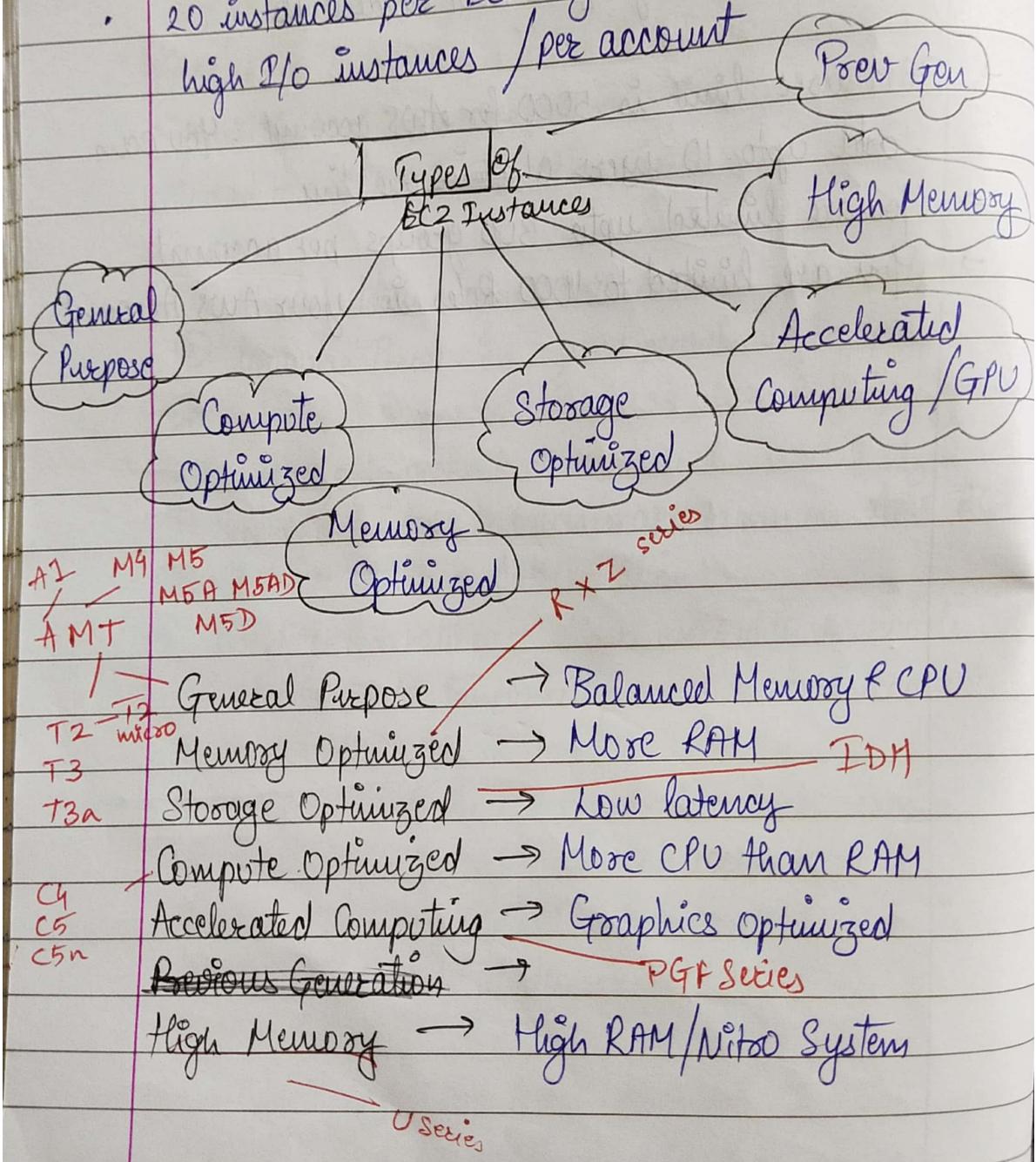
## ★ REVISION :-

### IAM - Identity Access Management

- What is it?
- what does AWS recommends? - Root account
- How do we access the account that Root has created? IAM AC
- IAM user limit? 5000/10/-
- Groups per AWS Acc? 300
- IAM Roles? 1000
- Default limits of managed policies attached to an IAM Role & IAM user is 10
- Features -
  - ① Shared access to your AWS Account (Globally)
  - ② Granular Permission - grant permission for specific resources such as S3, EC2, Redshift, Dynamo, - read only access to others; Amazon
  - ③ Securely access AWS Resources for apps to run on EC2.
  - ④ MFA - Multifactor Authentication
  - ⑤ Identity Federation - allows users with passwords elsewhere to temporarily access an AWS Account.
- AWS Identity & Access Mgmt is a web service that helps you securely control access to AWS Resource you use IAM ~~to~~ control who is authenticated & authorized
- When you create an AWS account it gives you single sign in options which has ~~the~~ the ~~admin~~ access to all the ~~set~~ services & resources in the account.

- account
- The identity is called AWS Root account user & is accessed by signing in with the email address & password that you can use to create the account.
  - Aws strongly recommends you to make an ~~not~~ not to use the root account for any administrative tasks
  - Use the services using your IAM account to manage administrative tasks to your account & securely lockaway your root credentials
- IAM user limit is 5000 for Aws account . You can add upto 10 users at ~~time~~ one time
  - You are limited upto 300 groups per account
  - You are limited to 1000 Roles in your Aws Account

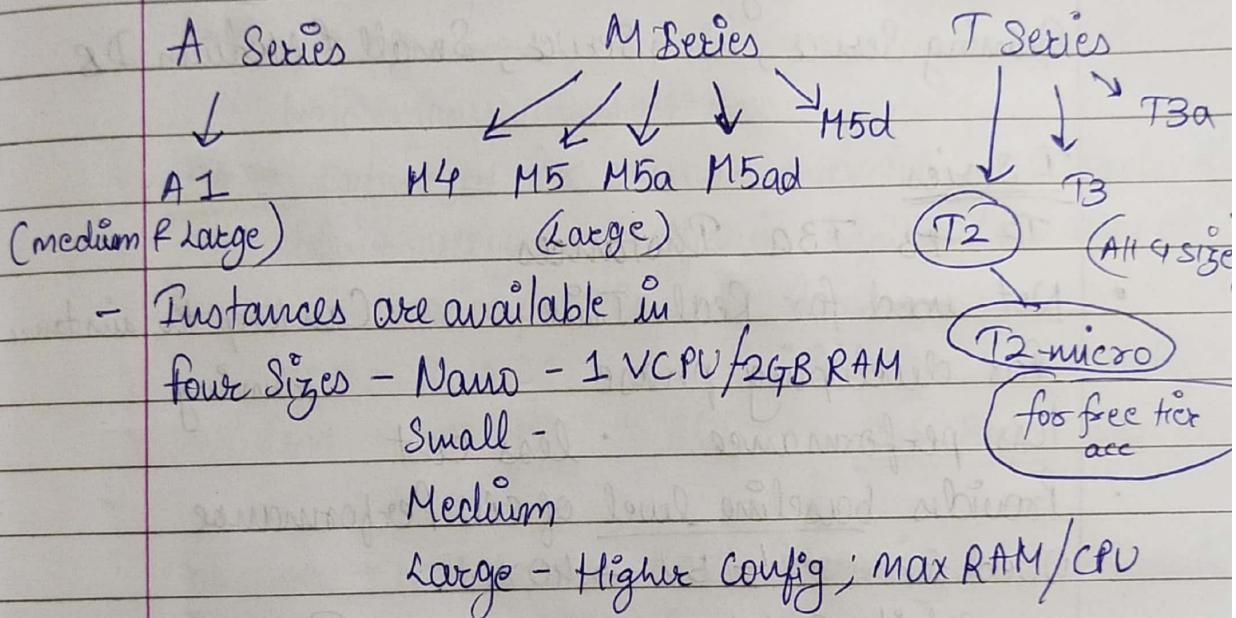
- ★ EC2 - Amazon Elastic Compute Cloud
- provides scalable computing capacity
- we to launch virtual servers as you need.
- Configure security & w/o & manage storage
- scale up or scale down the instance
- 2 storage options - EBS & instance store
- Pre configure templates known as Amazon Machine Images
- 20 instances per EC2 region with 2 default high I/O instances / per account



## ① General Purpose Tustance :-

- Balance of compute, memory and n/w resources
  - Avg - not high or low - everything
  - Normal purpose/type of application
  - for variety of workloads.

3 series in General Purpose



## ~~A1~~ A1 Instances :-

- ~~Not~~ suited for scale out workloads that are supported by the Open Ecosystem.
  - well suited for following application

- ① Webserver
  - ② Containerized microservices
  - ③ Caching fleets
  - ④ Distributed data stores
  - ⑤ Application that Requires Atom Instruction set

- majorly used
- M-Series - ~~M4~~ M4, M5, M5a, M5ad, M5d
- M4 {
- VCPU - 2 to 40 (Max)
  - RAM - 8GB to 160GB (Max)
  - Instance Storage → Elastic Block Storage

M5, M5a, M5ad, M5d (Higher level)

- VCPU - 2 to 96 (max)
- RAM - 8 to 384 (max) Nitro Virtualization

Instance Storage → EBS & NVMe SSD

Gaming Server, Web Server, Small & Medium DS

### T-Series

T2 T3 T3a Instances

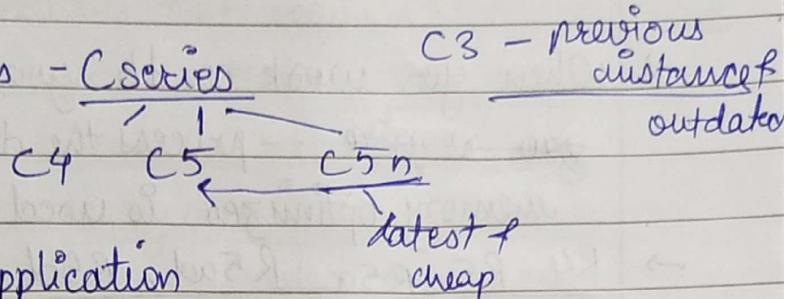
- Not used for Real Time
- for developing phase
- less performance
- Provides baseline level of CPU Performance
- Used for
  - Website & Webapp
  - Core Repository
  - Development, build, test
  - Microservices
- Can keep the instances running
- less cost
- Basic RAM & CPU
- 5 Gbps N/W ~~start~~

VCPU → 2 to 8

RAM → 0.5 to 32GB

## ② Compute Optimized Instances:

- More CPU than RAM
- A machine that can process multiple tasks ~~at~~ <sup>together</sup>
- parallel processing
- Cheap
- Only on Series - C series



- Compute bound application  
↳ Bank eg

In many processes are running

C4 → optimized for compute intensive workloads & deliver very cost effective high performance at a low price per Compute Ratio.

VCPU - 2 to 36

Storage : EBS Only

RAM → ~~8~~ ~~16~~ to 60GB

3.75

N/W BW → 10Gbps

Use cases → Web Server, Batch Processing, MMO Gaming, Video Encoding

C5 - cost effective high performance at a low price per compute ratio

VCPU = 2 to 72      RAM - 4 to 192 GB

BW → 25 Gbps

Storage → EBS & NVMe SSD

Mostly used in Banks

### ③ Memory Optimized.

R - series

X - series

Z - series

All 3 are similar but the major issue is you should know what type of db you will be working on.

→ When you work on db you need huge RAM to process the data, this is where memory optimized is used.

→ R4 R5 R5a, R5ad, R5d

- working on small db ; cheapest
- high performance, Cassandra, MongoDB, SQL
- Relational (My SQL) ; NoSQL (MongoDB, Cassandra)
- In memory cache ;
- VCPU - 2 to 96
- RAM - 16 to 768 GB
- Instance Storage - EBS or NVMe SSD

→ X1, X1e

- Maximum Ram
- Monthly cost is high
- Used in ; Automation ; SAP HANA
- VCPU - 4 to 128
- RAM - 122 to 3904 GB
- Instance Storage - SSD

→ Z1d Instance (Middleware)

- Core freq. of 4.0 GHz
- Fastest of any cloud instance
- AWS Nitro System , xeon processor, upto 1.8TB
- VCPU - 2 to 48
- RAM - 16 to 384 GB
- Storage - NVMe SSD
- Usecase - Electronic Design, Automation & certain DB workload

(4)

Youtube eg / songs eg

Storage Optimized Instances :- / sequential <sup>and</sup> ~~or~~ with

T Series - low capacity

D Series - mid capacity

H Series - high capacity

all 3 are used for data storing

for warehouse but how large

is the app that matters

- designed for workloads that requires high, sequential read & w access to very large data sets on local storage
- not parallel; one by one; you cannot upload files parallelly; it gets uploaded one by one
- Optimized to deliver few of thousands of Low Latency Random I/O Operations per Second (IOPS) to application

D2 Instance - D series

- Devised storage; applications which has ~~which~~ has Amazon prime, Massive Parallel Processing (MPP) data warehouse i.e netflix → warehouse containing lots of ~~of~~ data stored such as 4T,
- Map Reduce & Hadoop distributed Computing
- Log or data Processing app

vCPU → 4 to 36

RAM - 30.5 to 244 GB

Storage - SSDH1 Instances - H Series

- ~~16 TB~~ per local storage
- Virtual CPU & local storage are good enough
- direct attached storage ← can access to large amounts of data on J

- Applications that req<sup>to</sup> access large amount of data
- VCPU ~~RAM~~ → 8 to 64
- RAM → 32 to 256 GB
- Storage → HDD

### I3 and I3en Instances :-

- Large scale Instances for big/large companies
- FB; YT
- Costly
- Online payments
- Relational DB; NOSQL DB; DFS; Data Ware... App
- VCPU - 2 to 96

RAM - 16 to 768 GB

Local Storage - NVMe SSD

N/w Performance - 25 Gbps to 100 Gbps

Sequential Throughput :-

Read - 16 GB/s

Write - 6.4 GB/s (I3)

8 GB/s (I3en)

### ⑤ Accelerated Computing Instances

- use H/w accelerators, floating point num, calculations, graphics processing or data pattern matching
- used by AI; ML; DL; Gaming; Live Streaming
- \* **F1 Instances :-**
- offers customizable h/w acceleration with field programmable gate arrays (FPGA) - digital cameras uses for signal processing; DSLR

- ↑ FPGAs ↑ the performance
  - Live streaming should be good enough (ie FPGAs)
  - Each FPGA contains 2.5 million logic elements & 6800 DSP engines  
digital signal processing
  - Designed to accelerate computationally intensive algo,
  - f1 provides Local NVMe SSD storage
- VCPU - 8 to 64  
FPGA - 1 to 8
- RAM → 122 to 976 GB
- Storage → NVMe SSD Scientific; AI; DA; ML; DL
- Used in - Genomics Research, financial analytics, video processing

### \* P2 & P3 Instances

- Uses NVIDIA Tesla GPU
- Provides high b/w N/w
- upto ~~32~~ 32 GB of memory per GPU

#### P2 Instances

VCPU 4 to 64

GPU 1 to 16

RAM 61 - 732 GB

GPU RAM 12 - 192 GB

N/w B/w 25 Gbps

#### P3 Instances

VCPU 8 to 96

GPU 1 to 8

RAM 61 - 768 GB

Storage SSD + EBS

N/w B/w - 10 Gbps

- used in ML; DB; Seismic analysis; Genomics  
natural disasters study

molecular modeling; Deep Learning; AI

- API used in P3 is CUDA 9.1 & Open CL 1.2

P2 CUDA 8 & Open CL 1.2 ✓

## \* G2 & G3 Instances

- Optimized for Graphics Intensive
  - 3D visualization ; modeling & Gaming
  - NVIDIA Tesla M60 GPU ; cost effective ; high performance
- VCPU 4 to 64  
GPU 1 to 4  
RAM 30 to 488 GB  
GPU Memory 8 to 32 GB  
New performance 25 GB PC
- Uses :- Video creation Services ; Streaming : SD-Video

## (6) \* High Memory Instance :- — U Series

- Super comp
  - works on dedicated host
  - Bare metal instances
- purpose built to run large - in - memory db including production developments of SAP HANA in the cloud
- Features :-
- ① Latest Gen Intel Xeon Pentium 8176 M Processor
  - ② 6, 9, 12 TB of ~~RAM~~ instance RAM ; the largest of any EC2 Instance
  - ③ Powered by the AWS Nitro System , a combination of dedicated fw & light weight hypervisor
  - ④ Bare metal ~~Instances~~ Performance with direct access to host h/w
  - ⑤ EBS optimized by default at no additional cost

- ⑥ N/W Performance - 25 Gbps } R/W time is  
Dedicated EBS b/w - 14 Gbps fast
- ⑦ Each Instance offers 448 Logical Processor

⑧ Previous Generation :-

- They are available and can be purchased
- If you ~~do~~ already own an instance and then Amazon has shifted that particular instance into new gen so will it still affect your costing process? NO!
- T1, M1, C1, CC2, M2, CR1, CG1, i2, HS1, M3, C3 & R3
- When the billing starts? as soon as you purchase & it boots up your ~~per sec~~ billing starts unless you terminate it
- Stopping or shutdown will still ~~not~~ count as bill but won't charge for instance.
- Pay per sec is only for Linux & Ubuntu
- for Windows → Hourly bill
- Monthly bill is hourly bill
-

## \* Cloud Computing

- use of hosted services such as data storage, servers, databases, n/w & softwares over the internet.
- data stored in physical servers → maintained by cloud service provider.

## \* Characteristics of Cloud Computing :-

### ① On demand Services -

- cloud computing services doesn't require any human administrative; as users themselves monitor, manages & provision the computing resources as needed.

### ② Broad Network Access -

- The computing services are generally provided over standard networks and heterogeneous devices.

### ③ Resource Pooling - allows an IT Service provider to provide multiple clients access the same tool and storage space in a pool of resources.

### ④ Rapid Elasticity -

- Cloud services should have IT resources that are able to scale in and out quickly as user's need

### ⑤ Measured Service -

- The resource utilization is tracked for each application & occupant, it will provide both the user & resource provider with an account of what has been used, for monitoring billing & effective use of resources.

# Cloud Services / Types of Services

Date:

- ★ PaaS → N/w; storage; servers; virtualization
- PaaS → N/w; storage; S; V; OS; Middleware; Runtime
- SaaS → — ; Applications

| IaaS           | PaaS           | SaaS           |
|----------------|----------------|----------------|
|                |                |                |
|                |                |                |
|                |                |                |
| Virtualization | Runtime        | Application    |
| Servers        | Middleware     | Runtime        |
| Storage        | O.S.           | Middleware     |
| Networking     | Virtualization | O.S.           |
|                | Services       | Virtualization |
|                | Storage        | Servers        |
|                | Networking     | Storage        |
|                |                | Networking     |

Dotnet  
 Java Ruby  
 App you have to  
 manage as your  
 own

Everything  
 is managed  
 by cloud  
 you just  
 have to use  
 its services

## ★ Types of Cloud / Cloud Deployment Models.

### ① Public Cloud :- (Money locker)

- anyone can use this cloud via Internet across the globe
- Eg : Azure, AWS, GCP, Oracle
- Single Org only
- sold to the public, mega-scale infrastructure
- available to the general public
- Even if your data is in public cloud it's safe
- Pay as you go model

### ② Private Cloud - (Fortune)

- Single Org
- managed by the org
- Costly
- More Secure
- Eg - Openstack ; VMWare

### ③ Hybrid Cloud

- Public + Private
  - very imp data
  - normal data
- combining more than one cloud service provider

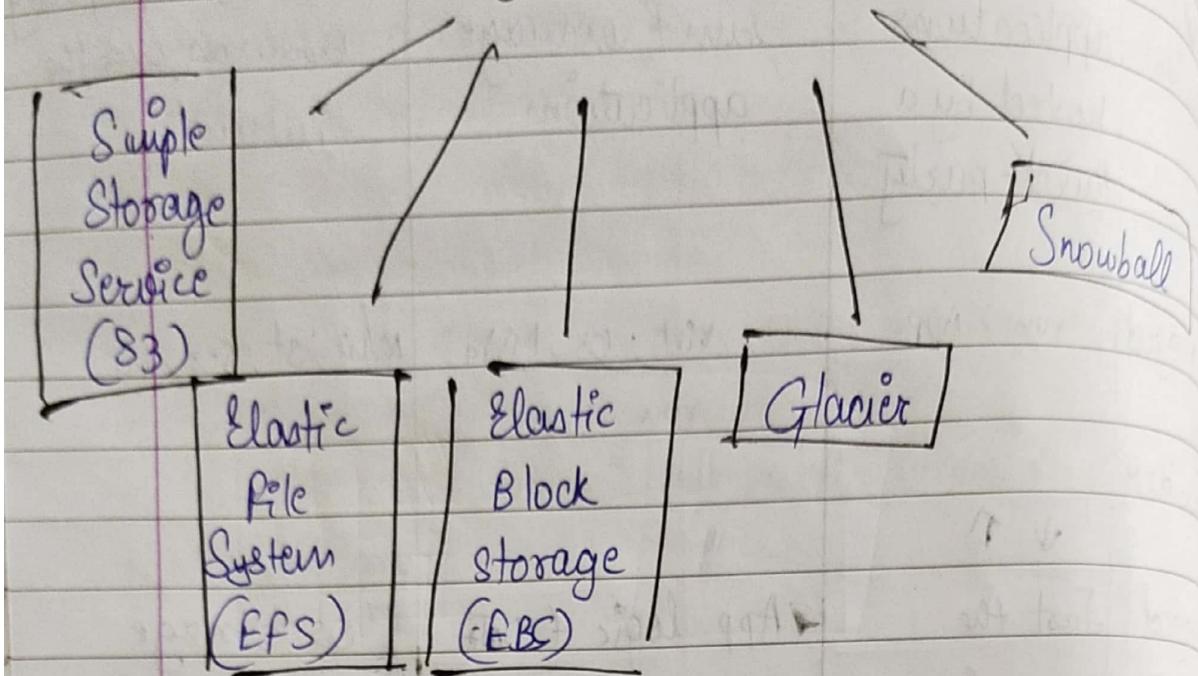
### ④ Community Cloud :-

- Shared infrastructure for specific community
- few org have shared concerns
- managed by 3<sup>rd</sup> party or org

| feature                             | SaaS                                              | PaaS                                           | PaaS                                                     |
|-------------------------------------|---------------------------------------------------|------------------------------------------------|----------------------------------------------------------|
| what it provides                    | complete S/W applications hosted by a third-party | Platform to develop, run & manage applications | Virtualized computing resources over the internet        |
| Managed by provider                 | run ; App                                         | Virt ; OS ; Midd ; run                         | N/W ; st ; ser ; Vir                                     |
| Managed by user                     | Just the usage & config of S/W                    | App logic & DATA                               | OS ; storage                                             |
| Users' Responsibility using the S/W | Accessing &                                       | Developing &<br>Deploying application          | Configuring &<br>management of infra                     |
| Eg:-                                | Google Workspace, Dropbox, Salesforce             | Google App Engine, Heroku, Microsoft Azure     | Amazon EC2, Microsoft Azure VMs, G Cloud Compute Engine. |
| Target Users                        | End users who need S/W ready use                  | Dev who need platform to build applications    | fully customizable infrastructure                        |

## ★ AWS Storages :-

### Types of Storage



#### ① Simple Storage Service (S3) :-

- Most commonly used in AWS
- Object level storage - can store any type of file ;
- can access it from anywhere .
- can make it public or can share with specific person → store unstructured data
- NOT so costly

#### ② Elastic file System (EFS) :- (Linux based system)

- Eg :- you have multiple branches available around the world ; they have an app of their own ; to update they share a same space which contains the app data ; they can update their app by downloading the resources.
- Storing data at common space & share among different zones
- s/w update ; patch files

### ③ Elastic Block Storage (EBS)

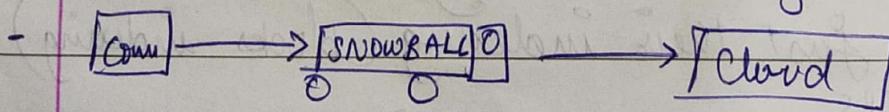
- Block level storage
- You can only access this through EC2
- ~~Server~~ Storage provided to the Server will be based on EBS

### ④ Glacier :-

- No more a Type - cheapest
- S3 Glacier
- Official type of S3
- Reliable, Durable, Highly available storage
- use :- suppose a company : T series → many song -  
singing song based on you - less demand on old  
songs - they cannot delete old song as that's imp  
for future needs (events)
- 10 Years ; 20 Years; etc.
- You don't want to use the data but want to store it  
safely.
- 1/- per GB

### ⑤ Snowball :-

- portable storage
- Used for Data Migration
- Consider a company who wants to shift the data  
from datacenter to cloud
- It's a like truck which contains storage

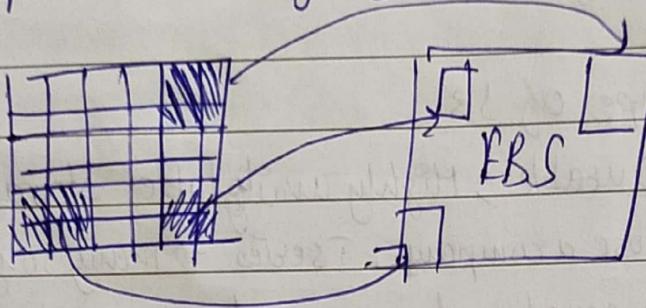


Migration

# \* Obj. Block Based Storage vs Object Based Storage

## ① Block Level Storage

- good for structured database where parallel processing is required
- storing data ~~parallelly~~ parallelly
- Block storage divides the data to be stored in evenly sized blocks for instance, a file can be split into evenly sized blocks before it is stored.  
*(data chunks)*



- To retrieve we use indexing
- Those blocks can store at any location
- Used in the EC2 instances that are created  
~~it also creates EBS for storing the data.~~  
*(scamajha ja)*

- Access through EC2 Instance
- Data blocks stored in block storage would not contain Metadata (data created, data name, data modified)

→ Index no :-

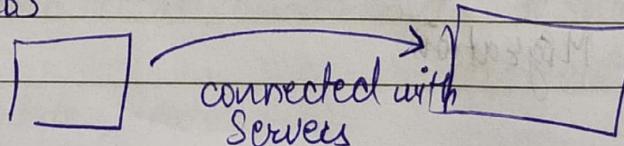
File type -

File Size -

File stored -

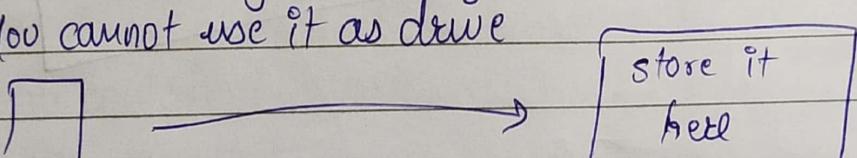
This wont store

- Its just their indexing (blocks indexing)
- Eg EBS



→ Block storage only keeps the address (index) where the data blocks are stored, it doesn't care what is there in the block; Just remembers how to retrieve it.

## ② Object Storage :-

- 5 MB/5GB file will be stored as it is
- In Object storage; an object is
  - The file/data itself
  - Its metadata
  - Object Global unique ID
- The object Global Unique ID, is a unique Identifier for the Object (can)
- Basically the file name should be unique; so that it can be accessed anywhere from the world.
- Access through Internet
- You cannot use it as drive
- 

```
graph LR; A[PC Full] --> B["store it here"]
```

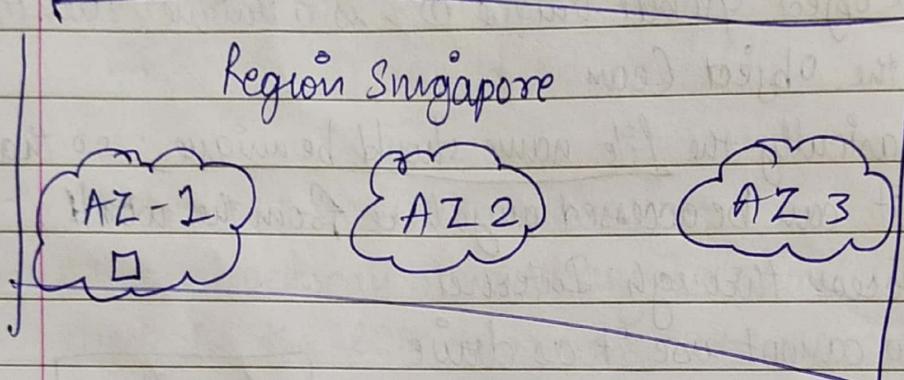
PC Full

No worries

- Works as an External hard disk; access whenever needed
- Eg :- Dropbox; AWS; S3; fB

## ★ S3 - Simple Storage Service :-

- storage for the internet → (http & https) use to access data - unless you give permission
- can share with anyone only the object
- Only be created using your AWS Account
- S3 is Object Based Storage
- You cannot install OS in S3
- S3 has a distributed data-store architecture where objects are redundantly stored in Multiple Locations [Min 3 loc in same Region]



- S3 creates min 3 copies ; even if one destroys other can retrieve it (only inside one region)
- ↳ Highly available  
↳ Redundant  
↳ Scalable
- Data is stored in Buckets
- You cannot create Bucket inside Bucket  
↳ Flat Container of objects
- Max cap of Bucket is 5TB.
- No limitations on creation of bucket but soft limitation is 100 Buckets (can be extended)

- You can create folders in your Bucket
- Bucket ownership is non-transferable
- S3 bucket is Region Specific

### \* Naming Rules :-

- S3 Bucket names (keys) are globally unique across all AWS Regions (globally unique)
- Bucket Names cannot be changed after created
- If the bucket is deleted, its name becomes available ~~for you again to use~~ you or other accounts.
- Bucket names must be atleast 3 & no more than 63 characters long.
- Bucket names are part of the URL need to access a bucket
- must be a series of one or more tables.
- can contain lowercase; numbers ; hyphen cannot use uppercase

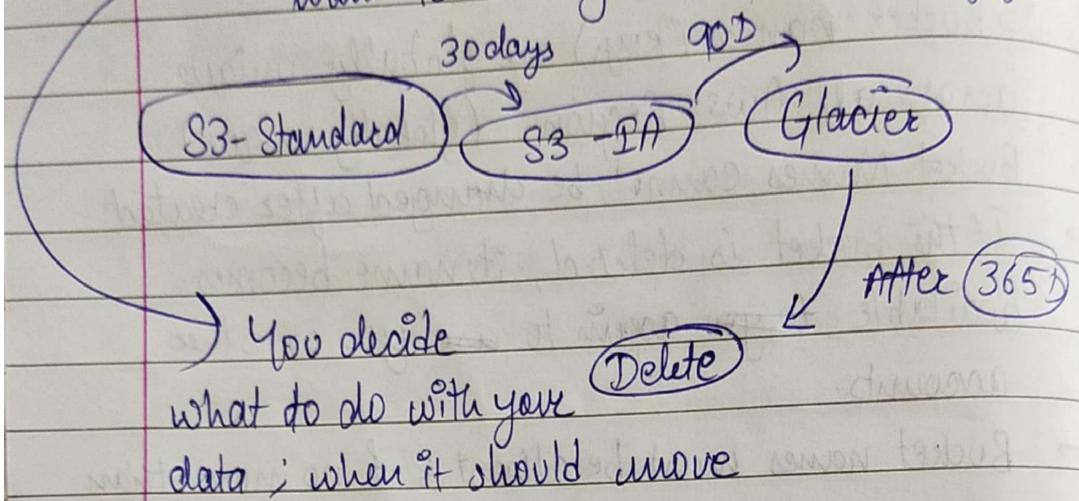
|       |  |       |  |              |
|-------|--|-------|--|--------------|
| ABC X |  | abc ✓ |  | - 123abc - X |
|-------|--|-------|--|--------------|

abc-123 ✓

- Bucket name should not be an IP Address
- Each table must start & end with letter or num.
- After creation of bucket it is ~~also~~ private unless you make it public .

## ★ S3 Subresources :-

- ① **LifeCycle** :- To decide an objects Lifecycle right
- Once your data is huge enough but you want to move your data accordingly like



- ② **Website** :- Static website hosting

- to hold configurations related to static website hosted in S3 Buckets.

- ③ **Versioning** →

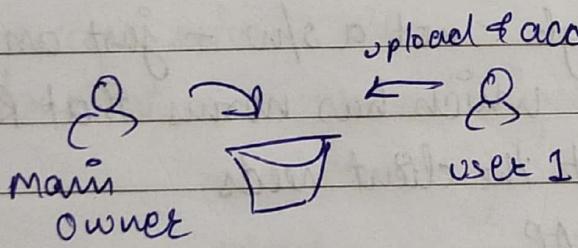
enable or suspend - stop for sometime  
not disabled

- ④ **Access Control List** - Security on Bucket

- who should access your bucket

## \* S3 - Objects :-

- Object can be stored from 0 byte to 5 TB
- Each object is stored & retrieved by a unique key (ID or name)
- File name globally unique is not req; Bucket  $\rightarrow$  bucket
- An object in AWS S3 is uniquely identified & addressed through
  - Service endpoint - region
  - Bucket name
  - Object key(name)
  - Optionally Object Version
- Object stored in a S3-bucket in a region will never leave that Region unless you specifically move them to another Region or CRR.
- A Bucket owner can grant Cross account permission to another AWS account to upload objects



- You can grant S3 bucket/object permission to
  - Individual users
  - AWS Account
  - Make the resource public
  - or to all authenticate user

12. 0. 0. 0

12. 0000 0000 . 0000 0000 . 0000 0000

$n=11$

$$8 + 8 + 5 = 2^8$$

12. 1111 1111 . 1111 1000 . 0000 0000

12. 255 . 248 . 0      ↴ subnet ID

→ 12. 255 . 248 . 0 /21 ← ~~Subnet mask~~

last  
subnet ID

Last subnet mask ↴

255 . 255 . 248 . 0

1st subnet ⇒ 12. 0 . 0 . 0

Next IP ⇒ 12. 0 . 8 . 0

12. 1111 1111 . 1111 1000 . 0 000 0000

4<sup>th</sup> Sept '24

#### \* REST -

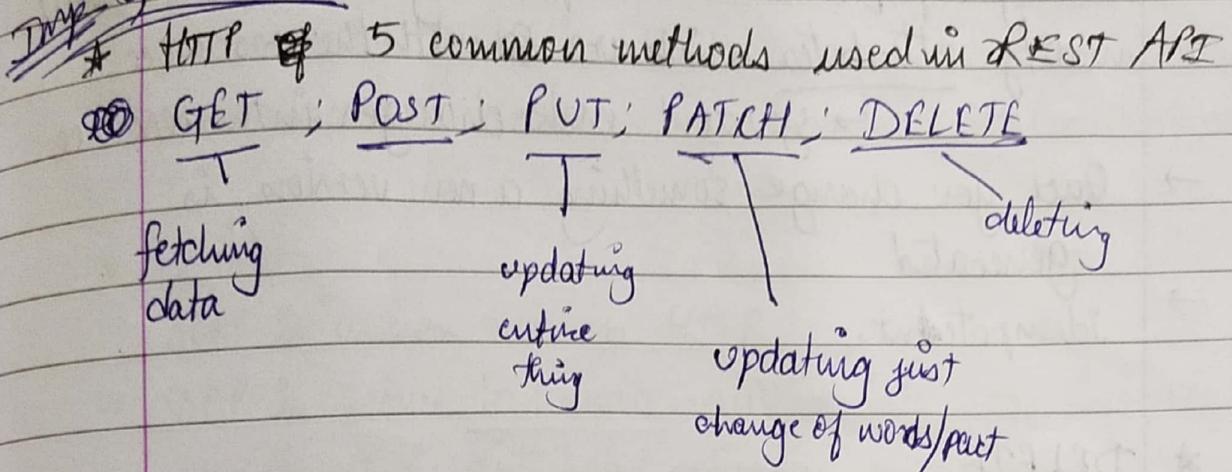
- Architectural style - not a s/w - just an architectural plan to follow which has norms that REST Has
- Just send what the client needs
- Preferred over SOAP

#### \* Simple Object Access Protocol (SOAP)

sending req → Basic operations from rest ⇒ GET; PUT; POST; DELETE  
response → the response will be in the form of HTML; XML; JSON

Why JSON? → lightweight

for exam



### ① GET :-

- Reading or retrieving from server Not found
- HTTP code 200 for OK ; 404 for ~~error~~ ; 400 BAD Req.
- Returns a representation in XML or JSON
- 501 - Service Error

### ② POST :-

- create subordinate resources
- on success returns 201
- idempotent ⇒ - repeating same task
  - performing same task without changing process results.
- You cannot ~~use~~ the same task again & again

### ③ PUT :-

- updating facilities
- can also create resources like POST
- update the resource by the ID chosen by client rather than the server

## \* PATCH :-

- modify capabilities like → updating surname or email.
- modify attributes which are ~~so~~ with the resource
  - ↳ you don't make changes in the resource
- Once you change something a new version is generated
- idempotent.

## \* DELETE

- used to delete resource identified by uri
- Returns 200 if OK

~~5/Sept/24~~

## SOAP API :-

- Simple Object Access Protocol
- Older, established and dependable but it can be slower than REST
- It has structured archi.
- uses XML as the data format for messages being sent & received by a SOAP client

- Why XML? → widely used language

Q) Compare between REST and SOAP API?

\* What is SOAP?

## Benefits of SOAP :-

- follow the structure while sending the message
- It has strict guidelines
- SOAP uses ~~uses~~ HTTP; but can leverage SMTP - simple mail...

### \* Adv

- SOAP is independent of HTTP
- SOAP is secure to handle sensitive data, eg - financial se...
- works well in distributed enterprise environment,

### \* Dis-adv

- complicated than REST; can have performance implications.
- doesn't support API calls
- less adaptable than REST
- slower than REST

### \* Use cases

- Transfers at bank
- Booking flights
- Billing services
- Navigation companies like Google Map.

Q. What is Amazon Machine Images (AMI)?

\* Four types

- Amazon provided AMI
- Community AMI
- Custom AMI
- Marketplace AMI

- ⇒ Special type of virtual appliance used to instantiat  
e (use) a ~~host~~ VM within EC2,
- ⇒ whenever you want to launch an EC2 you need  
to ~~not~~ have an AMI unless you create one

\* Properties :-

- A template for the root volume for the instance
- Launch permission that controls which AWS acc can use the AMI to launch instances
- A block storage mapping
- A block based storage is used

\* Depends on :-

Region

permission for launching AMI

Root device Storage

Architecture

- \* Buy, share & sell AMEs :-
- you will get paid if someone downloads your AME.
- 

## \* MAPREDUCE :-

- social framework & programming model used for processing huge amount of data.
- GPS & Hadoop are almost the same.
- programming model for writing appli. That are .....

• Mapreduce works in 2 phases : → Map  
→ Reduce

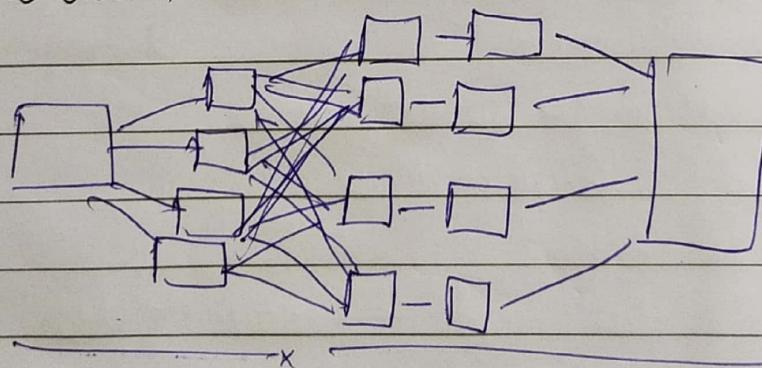
- Map : Splitting & mapping up the data
- Reduce : Shuffling, Reducing

→ The complete execution

↳ How will it organize the work ? [Hadoop organise the work]  
[Diagram has namenode]

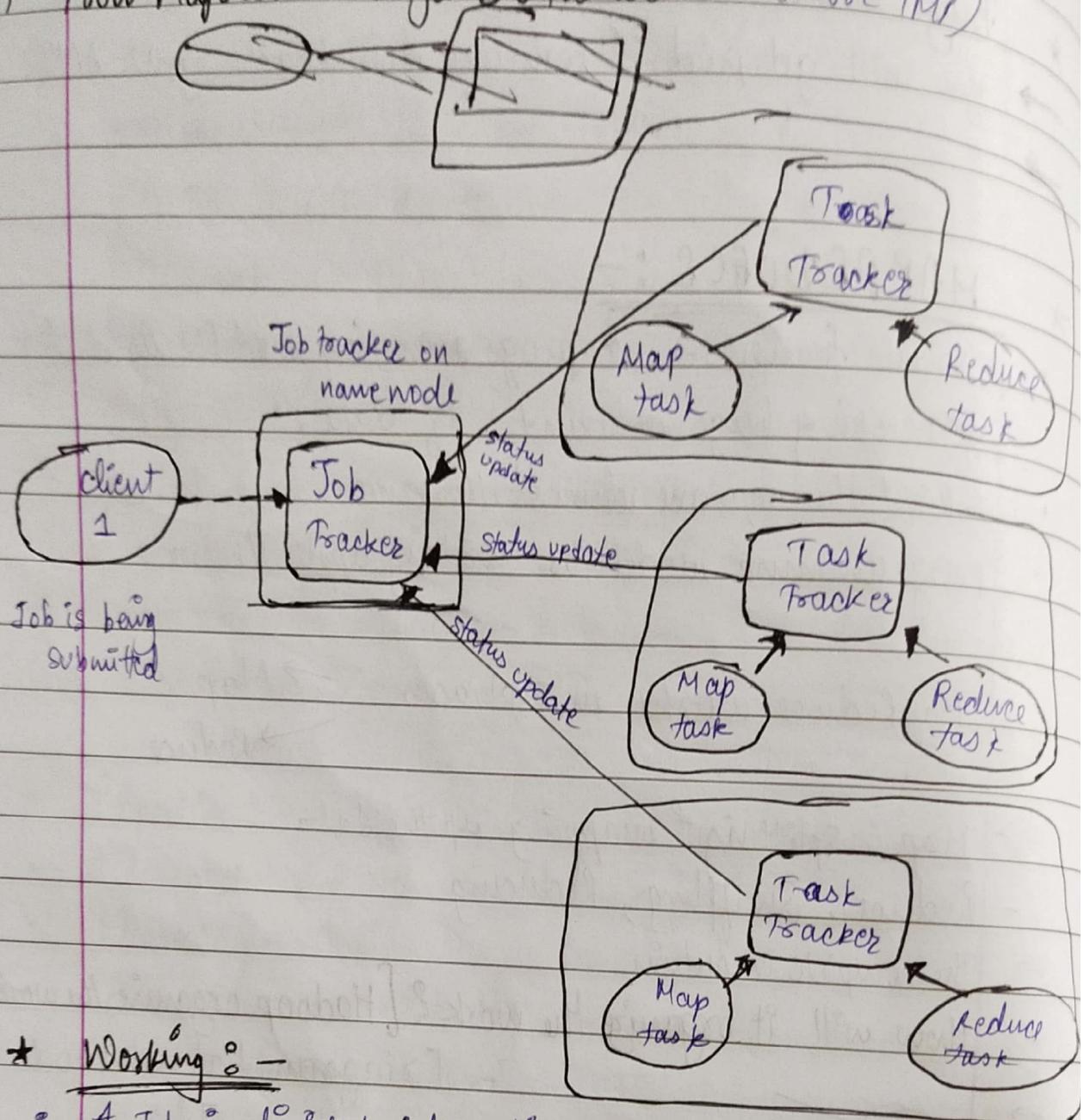
+ Map, Reduce & describe it

Q How does the archit



→ Jobtracker, Multiple tracker

Q How Mapreduce organizes the work? (More IMR)



\* Working :-

- A Job is divided into multiple tasks which is then run into multiple modes.

Types of computing → Parallel / Distributed ;

Edge ; Cloud ; fog

## \* GRID COMPUTING → IMP Question

- Distributing the load
- Defined distributed architecture of multiple computers connected by a n/w that work together.
- How does it work?

[Diagram]

- consists of 3 machines
  - Control node / service
  - Provider / grid node
  - User

## \* Key Components :-

[Imp Question]

- ① User Interface
- ② Security
- ③ Scheduler
- ④ Data management
- ⑤ Workload & Resource management

① UI → Interface is simple ;

## \* Component

## ★ Components of GFS :-

- Group of computers make GFS.
- A cluster is a group of ~~multiple~~ connected computers.
- There could be thousands of computers in cluster.
- Components :-

### GFS Clients :-

- computers or programs which may be used to request files
- These requests are generated to access or modify ~~to~~ a new file or an already existing file

### GFS Master Server :-

- Server as cluster's coordinator
- keeps track of the data that describe chunks
- Brain of the system
- coordinates everything & keeps a log of all activities happening in the cluster.
- It also tracks the location of the pieces of file & which file belongs to whom?

chunks

### GFS ~~Client~~ Server :-

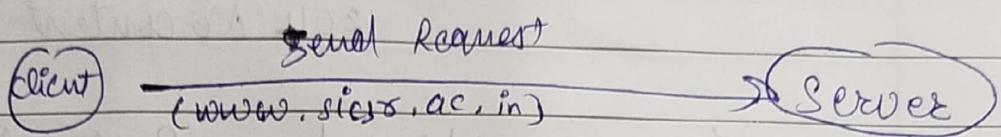
- These are the workers/slaves
- They store the actual file pieces, ~~&~~ each 64MB
- They don't give the pieces to the master server but gives ~~the~~ the file directly to the client when requested
- Safe & reliable
- GFS keeps a copy of each piece on diff chunk server.

| Parameters  | REST                                             | SOAP                                                         |
|-------------|--------------------------------------------------|--------------------------------------------------------------|
| Fullform    | Representational State Transfer                  | Simple object Access protocol                                |
| Protocol    | HTTP, HTTPS, FTP etc                             | HTTP & SMTP                                                  |
| Data format | JSON, HTML, XML, Plain text                      | only supports XML                                            |
| base of use | Simpler to implement & understand                | More complex due to strict standards                         |
| Performance | Lightweight & easy to use                        | complex due to XML payloads                                  |
| Security    | HTTPS & OAuth                                    | Built-in security standards                                  |
| ACID Prop.  | No.                                              | Yes                                                          |
| Caching     | Supports C for better performance                | No.                                                          |
| Standards   | Less rigid in terms of standards                 | High standards with strict rules                             |
| use case    | Best for web services like mobile apps, websites | Suitable for enterprise level transactions & complex systems |

## ★ REST API Information

- Simpler than SOAP
- write some of the points from differences
- All communications done via REST API uses only HTTP Request.

## Working



- Req sent to client in the form of URL as HTTP  
GET POST PUT DELETE, PATCH
- corresponds to CRUD operations.

① GET - retrieve data from resource

- If successful → 200 OK status
- NOT → 404 - Not found

② Post - create a new resource

- often returns 201 created with a link to the new resource.
- Not safe or repeatable
- means multiple post can create multiple copies.

③ Put - update or create resource

- If updated 200 OK or 204 No content.
- If creating, returns 201 creating
- repeating PUT gives the same result like POST

- Date: \_\_\_\_\_
- part of resource
- PATCH - modify the ~~resource~~. It only sends changes not the full resource. Not safe
  - Not safe
  - repeating can cause different results
  - DELETE - used to delete/remove a resource.
    - if successful      200 OK
    - 204 No content

### \* SOAP (Simple Object Access Protocol)

- protocol used for exchanging information b/w systems & applications
- formalized way of building APIs
- Key features :-
- uses XML for encoding messages
- Provides strict guidance to 4 components
  - Envelope
  - Encoding
  - Request
  - Response
- Protocols → HTTP, SMTP, TCP
- Built-in error handling

### \* Advantages :-

- Works across different platforms & operating systems
- Suitable for enterprise level & distributed systems
- Strong security features.

### \* Disadvantages :-

- slower & complex than REST
- less flexible & adaptable
- Doesn't support caching

### \* Usecases :- Banking, flight booking system, city mgmt, navigation.

### \* VERSIONING IN S3

- enable or suspend
- Explain process
  - ↳ save the file with same name
- can retrieve previous versions if deleted
- Provide example

## ★ AWS EC2 :- Elastic Cloud Compute

- a key pair is a set of 2 keys
  - public key :- stored on the EC2 instance
  - private key :- stored in your local machine

How does it work?

- you launch an EC2 instance, & you associate a key pair with it.
- The public key is automatically placed onto the instance
- To connect, you use the private key stored in your computer.

Example:-

- When you create a new instance, AWS asks you to either create a new key pair or use an existing one.
- You download the private key file (.pem) & keep it safe in your computer.
- To access the EC2 instance, you run a command in your terminal

`ssh -i "private-key.pem" ec2-user@your-instance-ip`

- The private key on your computer matches the public key on the EC2 instance, letting you securely connect.
- This protects your instance from unauthorized access.

## Routees

**function:** directs data packets b/w different n/w, typically within the same ~~diff~~ protocol

**function:** Routes data b/w devices on the ~~different~~ n/w but within the same protocol.

**Layers:**

**Protocol Handling:** works with IP addressing

**NAT** Commonly uses NAT to N/w addresses allow multiple devices translation to connect with ~~single~~ pub. IP address.

**Security firewalls & filtering**

**eg:- Home WiFi**

## GATEWAYS

acts like a bridge b/w 2 diff. n/w that may use different protocols.

**② Translates & converts data b/w different n/w with diff. protocols.**

**7**

Handles protocol conversion b/w diff devices

Translates b/w different n/w types, may also perform protocols ~~translation~~ <sup>translation</sup>

**firewall**

Office WiFi translates with other pub. WiFi.

## GRID COMPUTING

- Distributed system where multiple computers work together over a n/w to solve a complex task.
- The work is divided into smaller subtasks which is simultaneously on diff. machines
- The final result is combined to complete the task, enabling efficient use of resources like processing power, storage bandwidth.

### \* Key Components :-

- ① UI :- grid portal that allows users to interact with the grid & launch applications.
- ② Security :- GSI ensures secure communication through authentication, authorization & encryption.
- ③ Scheduler - Manages the distribution of tasks across grid nodes & handles job coordination.
- ④ Data Mgmt :- Ensures secure & efficient data transfer & access across nodes.

### \* Working

- A control server manages the grid & allocates resources.
- Providers (computers) contribute resources.
- Users submit tasks to be processed by the grid.
- Subtasks are executed in parallel, reducing time to complete large tasks.

\* Eg  $\frac{(4 \times 7)}{c_1} + \frac{(3 \times 9)}{c_2} + \frac{(2 \times 5)}{c_3}$

$$\begin{bmatrix} 28 & 27 & 10 \end{bmatrix}$$

combine together.

### \* Web Crawler :-

- a s/w program that navigates the internet to gather information from websites.
- Its a key part of Search engines like -

### \* Working :-

#### ① Discovery :-

- starts by visiting the known pages.
- From there it follows links on these pages to discover more pages.

#### ② Retrieval :-

- Once the data is found, it copies all the content from the page.

#### ③ ~~Reading~~ Parsing (Parsing)

- After retrieving the page, the crawler examines the page for useful information like heading, links & metadata.

#### ④ Filtering :-

- Some pages may be skipped, like login pages, pages restricted by the site owner or pages that are duplication of others.

### \* Search Engine Components :-

#### ① Web Crawler :-

- Also known as spider bots, these are SW programs that automatically browse the web to find new or updated web pages.
- Crawler starts with a set of URLs, follows link retrieve data from the pages to index them for search engine.

#### ② Databases :-

- Stores all content gathered by web crawler.
- includes information from web pages like text, images & metadata.
- Serves as the main storage where the search engine retrieves data when a user performs a search query.

#### ③ Search Interfaces :-

- user ~~the~~ inputs their query.
- Includes search boxes, result lines, filters & pagination. Allows to interact with the system & view search results.

#### ④ Ranking Algorithm :-

- Algo determine the order in which the search ~~eng~~ results are displayed to user.
- factors such as relevance, keywords, page quality, user behaviour, backlinks contribute to the ranking decisions.



#### How Search Engine works?

- ① Crawling
- ② Indexing
- ③ Ranking

#### GCP Resource Hierarchy

