21/05/25

Date: _____ Page No. _____

Topic: _____

AWS.

AMAZON WEB SERVICES.

1.1-

CCC-1 Cloud computing is the practice of using remote servers on the internet for carrying out a task, rather than using our own computers/servers.

⇒ Instead of buying a stack of servers, manually configuring, and handling any error in the servers.

⇒ We just rent it from a cloud computing company.

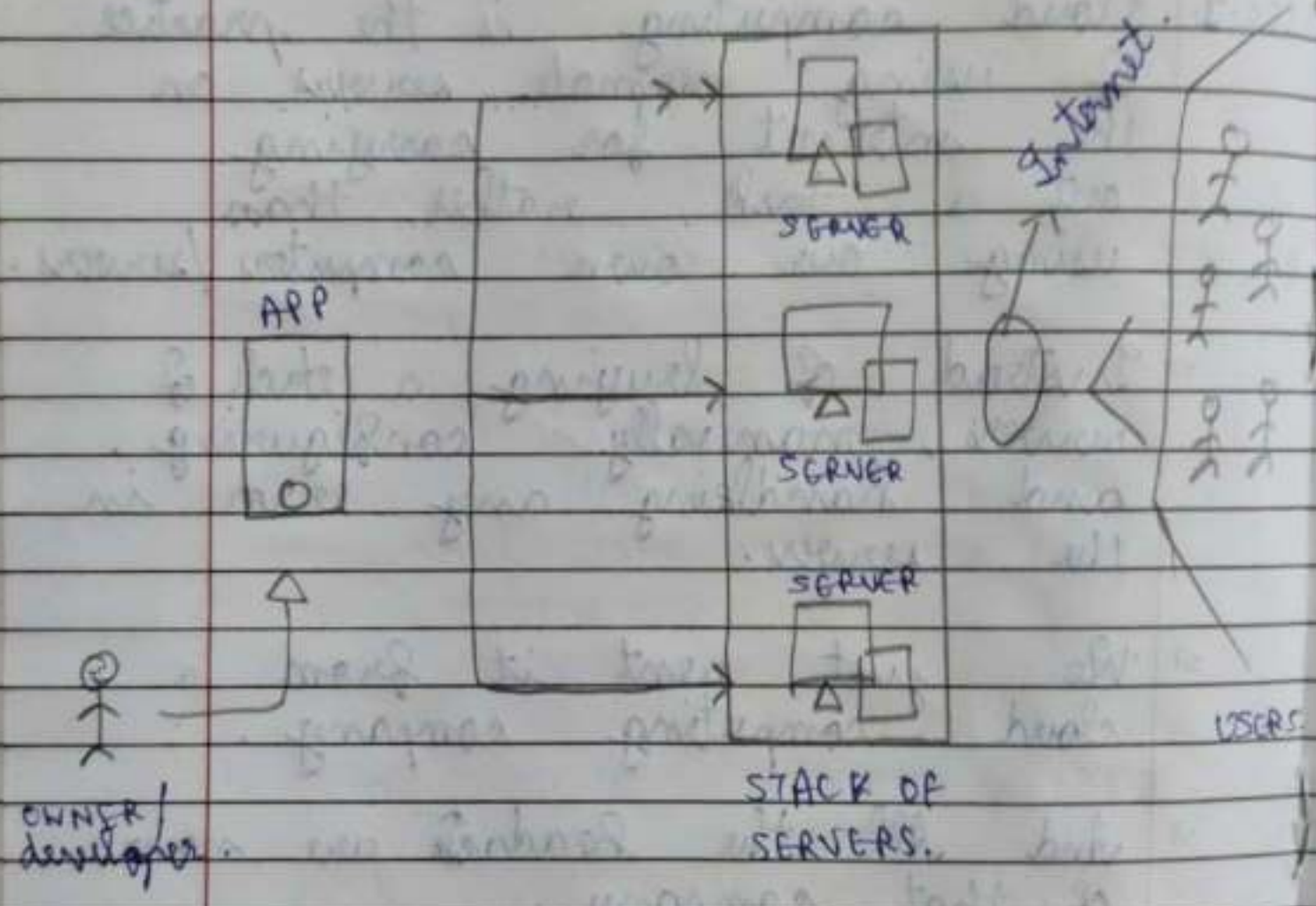
⇒ And All the headache are now of that company.

⇒ Better rent a servers than buying them for a huge amount plus overhead & headache. even upgrading any security patch.

Date _____ Page No. _____

Topic _____

12. OLD TECHNIQUE.



⇒ owner has a app and he has no idea about his app getting famous, still he has

⇒ still he bought the stack of servers to make his app run.

Date: _____ Page No. _____

Topic: _____

⇒ He will then have to get & set a local IP address (static IP) for the servers.

⇒ domain buying.

⇒ Then in order to show the app to the users he must connect it to the ~~servers~~ internet the servers to the internet (domain).

⇒ Then at after all the correct configurations the app will be available to the users.

⇒ HEADACHE :-

· He will have to hire a team to look after the sowers function properly.

· That team will look at scaling up or down the servers, upgrading hardware, any malfunctionality.

⇒ to overcome these problems we use CLOUD COMPUTING.

Date: _____ Page No. _____

Topic: _____

In AWS.

⇒ created an app.

⇒ Then go to AWS and launch a stack of servers & put my app over there.

⇒ AWS automatically gives me a static IP address, it also has a server that can give me a domain name as well.

⇒ I just need to configure my domain to these IP addresses & done.

⇒ I don't have to hire a team to manage the servers, because AWS says whatever you want to launch I'll handle it for you.

⇒ all the servers are maintained by cloud service provider, all hardware upgrades as well, even security patch.

Date: _____ Page No. _____

Topic: _____

4.3.

ADVANTAGES.

⇒ Little or no money investment.

⇒ More focus on app development.

⇒ Requires less workforce.

⇒ renting cost is minimal as 0.0004 $ per hour. (0.40 paisa). and it depends on the services used within the AWS.

⇒ rent a server use it (it will charge only for the amount of time the services were used), and give them back to the cloud computing Company.

Date: _____ Page No. _____

Topic: _____

1.4.

Cloud Products:

⇒ Google Drive.

⇒ Netflix.

⇒ airbnb.

⇒ amazon. E-Commerce website.
  ⇒ videos of amazon uses AWS S3.

⇒ prime video.

⇒ AWS is highly scalable, availability, reusability.

⇒ In 2008 the netflix got complete blackout which experienced the database lose.

⇒ Then it started to move towards cloud services, specially AWS.

⇒ By 6 January 2016, it has completed its cloud migration & shut down the last streaming services.

it is delivered through CDN

Date: _____ Page No. _____

Topic: _____

L1.5

⇒ # Cloud Computing Models.

⇒ 2 kinds of models.

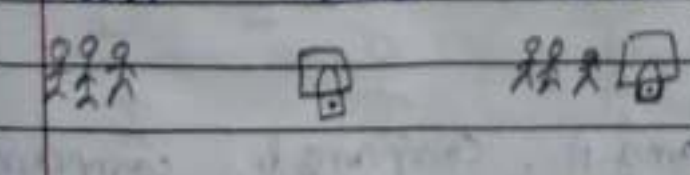
i.) Deployment model :

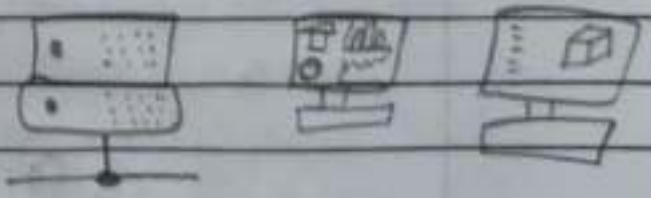
a. public cloud,
b. private cloud, &
c. hybrid cloud.

ii.) Service Models :

a. Infrastructure as a service. IaaS
b. Platform as a service. PaaS
c. Software as a service. SaaS

Deployment Models.	Service Models.
Public Cloud. Private Cloud. Hybrid Cloud.	IaaS. PaaS SaaS.





Date: _____ Page No. _____

Topic: _____

i) DEPLOYMENT MODELS.

⇒ various ways using which we can deploy our ~~model~~ application.

⇒ a) PUBLIC CLOUD :-

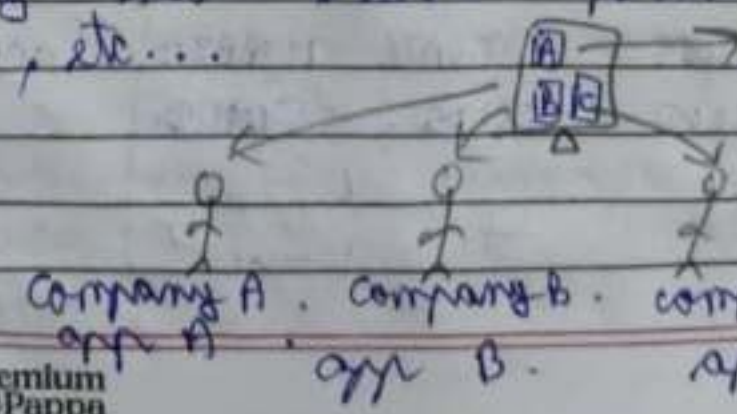
public servers are servers that can be owned by multiple companies.

⇒ same/one server but multiple companies can/have deployed their application on them.

⇒ server is shared among different companies.

companies

⇒ Ex: do not have confidential data or may not have privacy policy, etc...



Date: _____ Page No. _____

Topic: _____

b) PRIVATE CLOUD.

⇒ i) want a seperate server all for yourself : no matter how much space is empty, I just don't want any other companies data on my server.
⇒ I want my data to be Isolated.

ii) Buy our own server & create your own cloud on our data center.
⇒ Buy all the stack of servers required & Host our application on it.

⇒ Bought server, made our own cloud, Hosted our data/application onto the server.

PRIVATE CLOUD.

GCP ⇒ is a suite of cloud computing services offered by Google.

GCC ⇒ is a GUI for managing GCP resources.

c) Hybrid Cloud:

⇒ We want a kind of infrastructure where in the we are using some of the public clouds and some of the servers in the private cloud, in this case it becomes a hybrid cloud.

⇒ Ex :- A Research website.

⇒ Research company's website is on servers of public cloud.

⇒ But after the login one can see all the research material.

⇒ These research materials is not on the server of public cloud instead it is on servers of private cloud.

⇒ user will look like everything is on the same website.

website hosted ⇒ public cloud.

GCP ⇒ Google Cloud Platform.
OCC ⇒ Google Cloud Console.

Date _____ Page No. _____
Topic _____

sensitive & your research files are on the server of private cloud.

⇒ Mix of on premises & private cloud.

⇒ CLOUD WORKS (SERVERS).

⇒ Cloud companies buy a big machine or a big server with lot of RAM & processors in pair.

⇒ they launch multiple instances of virtual machines on it.
with 8GB of RAM.

⇒ i3/i5 can launch around 3 OS on the same server at the same time using a virtual machine.

⇒ Similarly, cloud providers are doing.

⇒ buy a stack of server launch multiple VM on that servers, these VM's are owned by the people that launch servers on AWS or on any cloud.

AWS console or Azure console or GCP console.

Date: _____ Page No. _____

Topic: _____

⇒ when your instance is up & ready it is actually a part of VM that is a part of the server.

⇒ it could also be a server can also be hosting 7/8 VM on the same server that can be owned by different companies who have created there Azure or GCP a/c.
people

⇒ Ex:- Government agencies CIA in US, they have very confidential data that can only be accessed by this employee irrespective of there location.

⇒ In these cases they use a stack of server that are only for their own company. isolated from the other companies.
⇒ pay as you go model.
⇒ they will charge for only the amount of time we use them.
⇒ it can be a little costly.

Date: _____ Page No. _____

Topic: _____

ii) SERVICE MODELS.

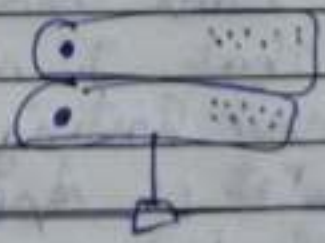
⇒ various services i can get from the cloud provider.

a] IaaS. [EC2].

⇒ cloud provider will give you the access to the ~~server~~ infrastructure, whole whole SYSTEM, the whole virtual M as a service.

⇒ get access to the OS of the server & we can install anything we want in that server.

⇒ It can become, a DB server, a website server, anything.



IaaS.

m the other companies.

⇒ Ex:- EC2.

Date: _____ Page No. _____

Topic: _____

b) PaaS.

⇒ Here you don't get the whole OS, Rather access is given at a dashboard level, where user can upload the data, & the rest is taken care by the cloud provider.

⇒ Ex: [Elastic Beanstalk.]

⇒ I want to upload the files & folders & the cloud provider automatically places it, where it has to be.

⇒ & we will get our app ready to go

a) IaaS.

⇒ In IaaS we'll get a fresh OS & on that I will have to install a web hosting software like tomcat or Apache. Then do a FTP transfer files from local to or the server. then place them on the particular place of files or folders.

Date: _____ Page No. _____

Topic: _____

c) SaaS.

⇒ Here, we get the software that we can use

⇒ We do not have any access to the OS of the server or the dashboard level access too.

⇒ What do we get is the software itself that's ready to be used.

⇒ Ex:- NETFLIX, FACEBOOK .. or say google excel, word, docs, etc...

⇒ software that is hosted on the cloud.

Date: _____ Page No. _____

Topic: _____

CLOUD PROVIDERS.

⇒ AWS.

⇒ MS AZURE.

⇒ GOOGLE CLOUD.

} Top cloud provider services. (almost 90%).

⇒ Digital Ocean.

⇒ IBM Cloud.

Date: _____ Page No. _____

Topic: _____

Ep 6. ⇒ WHY AWS only:-

(1.) ⇒ AWS owns 35% market share.
⇒ 35 out of 100 people uses it.

⇒ AWS was incepted since 2006, 14th March
⇒ first & foremost cloud provider.

(2.) ⇒ while MS Azure holds 13% & 2010
(3.) google cloud holds 6.5%. 2008
↳ 1st February, 2010. was also known as windows Azure.

⇒ More job opportunities.

Nowadays :-

AWS ⇒ 29%, 14th March 2006.

MS AZURE ⇒ 23%, 1st February, 2010.

GOOGLE CLOUD ⇒ 12%, 2008.

Date: _____ Page No. _____

Topic: _____

⇒ AWS (29%). ⇒

⇒ MS AZURE (22%) ⇒

⇒ GOOGLE CLOUD PLATFORM (GCP) : (12%). ⇒

⇒ Overall 63% of the total market share.

⇒ AWS S3 → March 2006.
AWS EC2 → August 2006.

⇒ MS AZURE → Windows AZURE → October 2008.
↳ Became commercially available on 1st February, 2010.

⇒ GCP → 2011, with google app engine as its initial product.

~~₹ 0.0005 ₹~~ per hour.

Date: _____ Page No. _____

Topic: _____

11.7

INTRODUCTION TO AWS :-

⇒ AWS is a subsidiary of amazon.com.

⇒ first player to enter in the market.

⇒ AWS provides on-demand cloud computing platforms to individuals, companies & governments.

⇒ on a metered pay-as-you-go basis.

⇒ pay for only the time you used the service. (approx 0.0005/hr).

⇒ it offers services in various domains like, storage, compute, networking, etc, security, etc., management, etc..

AWS ⇒ S3 ⇒ Amazon simple storage service.

Date: _____ Page No. _____

Topic: _____

⇒ AWS FOUNDATION.

EC2 ⇒ Elastic compute cloud.

2.1. ⇒ * pre-EC2 PART-I *

"AGENDA"

i.) pre EC2 - VM.

ii.) pre EC2 - Intel Processor Generation.

iii.) EC2 - Regions & Availability zones.

⇒ i.) pre-EC2 ⟶ VM.

MULTIPLE OS.

windows. Apple. Linux

ISO → image ⇒ install() ⇒ native / hypervisor / type I / Bare Metal

server / hardware

Date: _____ Page No. _____

Topic: _____

⇒ POPULAR AMAZON SERVICES :-

Elastic compute cloud. Relational database service.
EC2, S3, Lambda, RDS, VPC,
Simple Storage service. Virtual private cloud.
Cloudfront, DynamoDB, EBS, IAM,
Elastic Block Store. Identity and access management
Cloudwatch, Elastic Beanstalk,
Redshift, Route 53, Fargate..

⇒ First on the server we'll install hypervisor.

⇒ then we'll tell the hypervisor to launch/install the multiple OS on VM, using different ISO images.
MULTIPLE

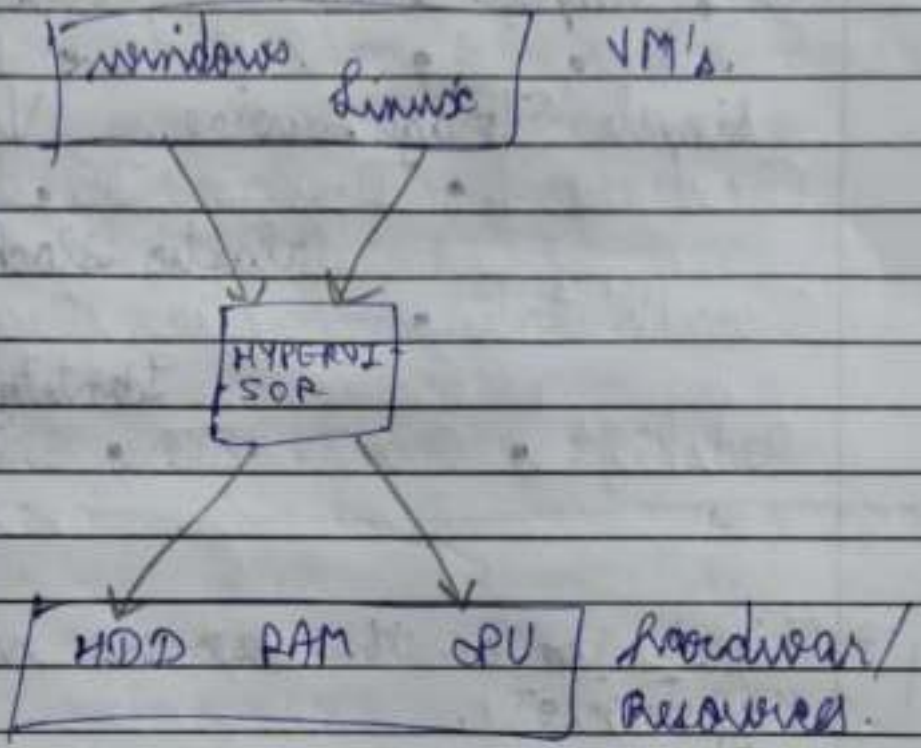
⇒ VM is a emulation of a computer system having OS, RAM, CPU or compute capacity.

⇒ It provides the exact same functionality as that of an actual computer system.

Date: _____ Page No. _____

Topic: _____

⇒ How does VM work.



⇒ pre-EC2 Intel processor generation

⇒ 1st Gen. Nehalem (2006) -
1st Gen. CPU introduced Hyper-threading.

⇒ 2nd Gen. Sandy Bridge (2011) -
2nd Gen. introduced Pentium
xeon E3
xeon E5
Mainly for enterprise servers or say

Date: _____ Page No. _____

Topic: _____

for cloud based VM's.

⇒ Entrypise servers do not use core i3, i5, i7, instead they use much higher or heavier known as Xeon.

⇒ 3rd Gen. Ivy Bridge (2012) -
they introduced Pentium
Xeon E3 v2
Xeon E5 v2
Xeon E7 v2
Advancement of previous gen. Xeon. CPU's.

⇒ 4th Gen. Haswell (2013) -
they came up with Xeon E3 v3.
Xeon E5 v3.
Xeon E7 v3
An upgradation of the previous gen. processors.

Date: _____ Page No. _____

Topic: _____

⇒ 5th Gen. Broadwell (2015) -
introduced Xeon D
Xeon E3 v4
Xeon E5 v4

⇒ 6th Gen. Skylake (2015) -
Xeon E3 v5.

⇒ 7th Gen. Kabylake (2016) -
introduced Optane memory support.

⇒ 8th Gen. Coffee Lake (2017) -
6-core i5/i7 CPU's.

⇒ 9th Gen. Coffee Lake refresh (2018) -
8 core i9 CPU's.

⇒ 10th Gen. Comet Lake/ice lake (2019) -
Comet lake (14 nm) 10-core i9.
Ice lake (10 nm), Gen. 11 CPU.

⇒ 11th Gen. Tiger lake (2020) -
Willow cove cores.

Date _____ Page No. _____

Topic _____

⇒ 12th Gen. Alder lake (2021) -
Hybrid architecture (P+E cores).

⇒ 13th Gen Raptor lake (2022) -
improved hybrid core design.

⇒ 14th Gen Raptor lake refresh (2023) -
minor performance refresh over 13th Gen.

⇒ 15th Gen. Arrow lake (Expected 2024-25) -
expected shift to Intel 20A process.

⇒ Elastic Compute (EC2 Concepts.) cloud.

⇒ Elastic :- It is the level at which a system is able to adapt to workload changes by provisioning & de-provisioning resources such that the resource meet current demand as closely as possible.

use
⇒ will have to pay only for the increased or decreased ~~amount of time active only~~ capacity only.
when 1.5 GHz pay for 1.5 GHz only.
if 3 GHz pay for 3 GHz only.

EC2

Date: _____ Page No. _____

Topic: _____

⇒ say : "Elastic processor.

Ex :- at 12 AM 8 gigahertz of CPU.

at 3AM it needs 3 GHz of CPU

at 9 AM it needs 1.5 GHz of CPU.

⇒ so, it increases & decreases the processor speed as per the workload increases or decreases.

⇒ needs more power it increas the processor speed & vice-versa.

⇒ Compute; Cloud;

it is a virtualize computant.

→ Internet / Intranet

EC2 ⇒ Elastic Compute cloud adapts capacity dynamically by altering compute resources to meet the need of a varying workload.

⇒ It is a combination of 3 words "Elastic", "Compute" & "Cloud".

Date: _____ Page No. _____

Topic: _____

12.2. EC2 Concepts ⇒ Regions & Availability zones.

⇒ Amazon has chosen few location around the globe, to put their data centers.

⇒ US East, US West, Mumbai, Seoul, Tokyo, Sao Paulo, Canada, EU, Singapore, Sydney.

⇒ All these places are called regions. totally around 36 regions & 114 availability zones.

⇒ within the regions itself there are multiple data centers these are called Availability zones.

Date: Page No.

Topic:

↳ has 6 data centers like us-east 1a, us-east 1b, 1c, 1d

these are the actual data centers, actual location

LOCATION REGIONS.

⇒ US East :- 2 regions. (2)
N. virginia (us-east-1).
Ohio (us-east-2).

⇒ US West :- 2 regions. (2)
N. california (us-west-1).
Oregon (us-west-2).

⇒ Asia Pacific :- 5 regions. (12)
Mumbai (ap-south-1).
Seoul (ap-northeast-2).
Singapore (ap-southeast-1).
Sydney (ap-southeast-2).
Tokyo (ap-northeast-1).

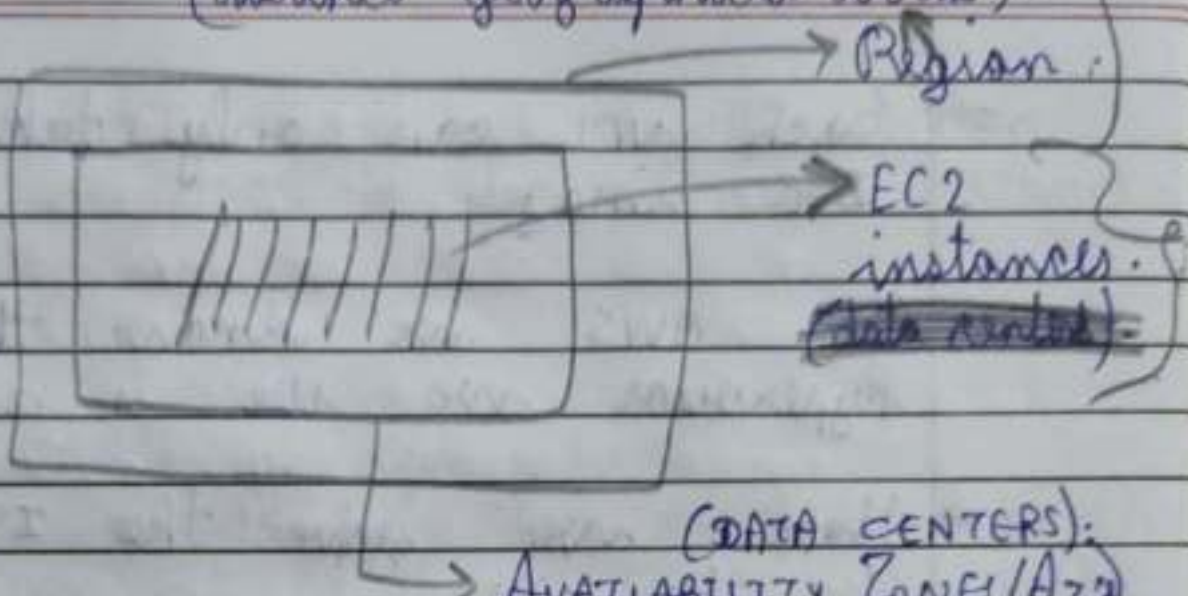
⇒ EU :- 2 ~~3~~ regions. (8)
⇒ Canada :- 2 regions. (2)
⇒ South america :- 1 region. (1)
⇒ Middle east :- 3 regions. (3)
⇒ Africa :- 1 region. (1)
⇒ China :- 2 regions. (2)
⇒ Israel :- 1 region. (1)

34.

these are the actual VM's.

Date: _____ Page No. _____

(distinct ~~geographical areas~~).



L2.3. Compute => EC2 - PART - II.

"AGENDA."

i. Instance Types, vCPU.
ii Root Device & AMI.
iii DEMO.
iv Elastic Network Interface, Tenancy.
v. Placement Group.
vi Pricing.
vii. Purchasing Options.
viii. Design Patterns.

Date: _____ Page No. _____

Topic: _____

⇒ Each VM can only take one ISO image.

⇒ In AWS we know the hypervisor we use is a Xen

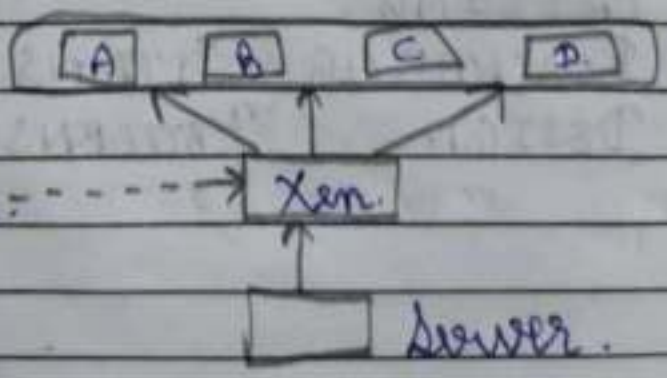
⇒ then we give the ISO image to the Xen.

⇒ Xen launches EC2 instances using this ISO image.

⇒ Each EC2 can have same or different OS.

⇒ "AMI".

⇒ Amazon Machine Image.

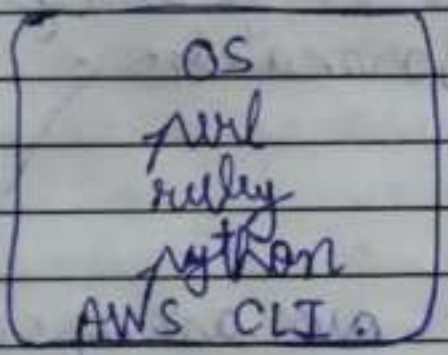


Date: _____ Page No. _____

Topic: _____

⇒ AMI :- AMI contains information required to launch an instance.

- ⇒ OS
- ⇒ Architecture
- ⇒ Storage for root device. (Instance Store or EBS backed).
- ⇒ Virtualization Type (HVM or PV).



⇒ If we config the ISO image with MS office in it with WINDOWS.

⇒ And then, it is Install on VM A, then the VM A has windows as well as MS office installed with it.

AMI

⇒ Here ISO image is similar to ISO image.

Date: _____ Page No. _____

Topic: _____

⇒ AMI is provided to the Xen hvr, then the Xen installs the on files, os, there on that AMI the VM.

⇒ if it installs the os and all the

⇒ AMI contains the os, & along with other information.

⇒ practical example:-

⇒ after login to aws.amazon.com.

⇒ go to all services.

⇒ go to EC2.

⇒ go to launch.

⇒ go to choose instance type.

Date: _____ Page No. _____

Topic: _____

INSTANCE TYPES.

⇒ different applications has different resource requirements.

⇒ like web app, or
DB based app, or
informatica based app.

⇒ one might need more CPU & less storage, other
⇒ might need more storage & more CPU, etc...

⇒ Resource requirements changes as per the applications.
⇒ some may need ⇒ 8 GB RAM 3GHz 1TB
some ⇒ 4GB RAM 1.5GHz 3TB.
⇒ different Instance type ~~may~~ for different types of applications.

⇒ Instance type determine the hardware of the underlying host computer on which EC2 instances are launched.

Date: _____ Page No: _____

Topic: _____

⇒ These are segregated into compute (processor), memory (RAM), storage optimized, general purpose, etc. types.

→ Instance type has instance family, vCPU, GiB Memory, current generation.

⇒ Family types :-

→ General purpose,
→ Compute optimized,
→ Memory optimized,
→ Storage optimized,
→ Accelerated computing,
→ High performance computing (HPC),
→ Mac instance.
→ Bare metal instance.

Date: _____ Page No. _____

Topic: _____

2.4.

⇒ vCPU :- virtual Central Processing Unit.

⇒ vCPU :- Each vCPU is a hyper-thread of an Intel Xeon core except for t2 & m3.medium instances (AWS definition).

⇒ CPU is not the one who does your work instead they are the processors or the cores that actually does your work.

⇒ CPU contains sockets, & processors or cores are housed on this sockets.

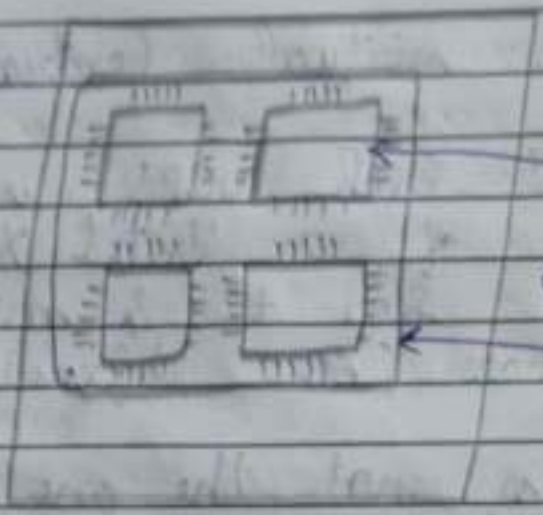
dual-core processors → 2 processors on 1 socket.
quad-core processors → 4 processors on 1 socket.

* 4 CPU can run 4 program at a same time.
⇒ 1 CPU can only run 1 program at a same time.

⇒ Intel changed these and made it :-
1 CPU then it can run 2 programs at a time.
each is a threads/vCPU.

Date: _____ Page No. _____

Topic: _____



CPU.

no. of vCPU are odd ⇒ processors is equal to no. of vCPU.

no. of vCPU are even ⇒ no. of processors equal to no. of vCPU ÷ 2.

Ex :- 1 vCPU has 1 core, 2 has ~~core~~, 8 vCPU → 4 cores divided by 2 each has threads.

→ Root device volume :- Contains the image using which the instance is booted.

Ex :- c drive.

Date: _____ Page No. _____

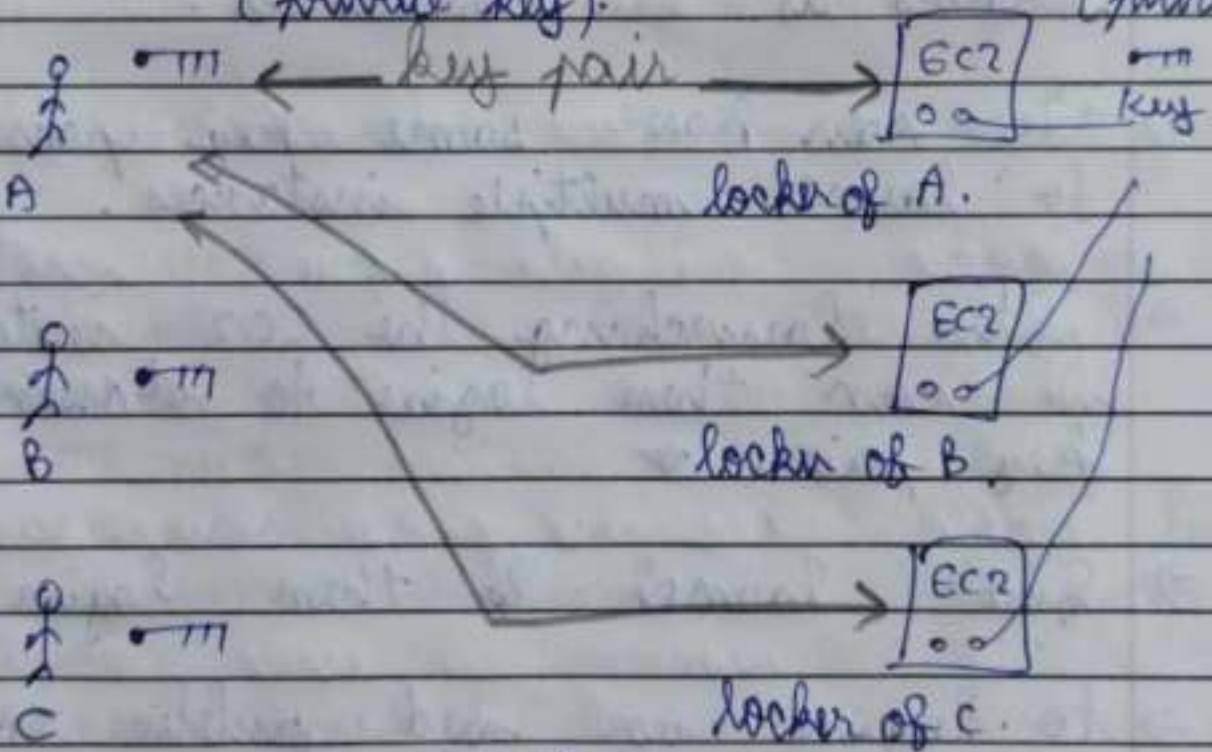
Topic: _____

⇒ Key - pair for security purposes.

aws - foundation - demo - key - pair

Name :- AWSFoundationDemo.

⇒ SUPPOSE :- AWS BANK.



⇒ A locker needs both the 2 keys to open it.

⇒ the one common key for all lockers is a public key that is with the bank itself.

⇒ private key every individual has their lockers 1 key known as private key.

22/05/25. Last 5 min of L 2.4.

Date: Page No.

Topic:

⇒ Both of these keys are combined known as Key pairs

⇒ there are 3 key pairs here (A, B, C).

⇒ public key is same where as the private key differs.
think of.

⇒ lockers as EC2 instances.

⇒ We can have same key-pair ~~for~~ to launch multiple instances.

⇒ after launching the EC2 instance. we can then login to it using the key-pair.

⇒ first launch & then login.

⇒ to login we need public IPv4 address & private key.

Date: _____ Page No. _____

Topic: _____

⇒ command to login:-

ssh -i "key.pem" ec2-user@<public IPv4 address> -

⇒ key.pem → is the private key.

ec2-user → default username for AMI Type (Amazon linux).

default username.	AMI Type.
ec2-user	Amazon linux.
ubuntu	Ubuntu.
centos	CentOS.
ec2-user	RHEL
ec2-user	SUSE linux.
admin or debian	Debian.

⇒ Common Errors:-

→ ensure port 22 is open in Security groups.
→ Make sure instance is in running state.
→ Key file must have only read-only permissions.

Date: _____ Page No. _____

Topic: _____

⇒ In (bash / WSL) :-

⇒ chmod 400 mkey.pem -

⇒ ssh -i "key\path\p.pem" ec2-user@<public IPv4 address>

⇒ In windows open powershell as run as administration.

⇒ STEP 1 :- Remove all existing permissions

icacls "key\path\p.pem" /reset

⇒ STEP 2 :- Disable inherited permissions.

icacls "key\path\p.pem" /inheritance:r

⇒ STEP 3 :- remove all users except current window user.

icacls "key\path\p.pem" /remove:g "Authenticated Users" "Users" "SYSTEM" "Administrators" "Everyone"

Date: _____ Page No. _____

Topic: _____

⇒ STEP 4 :- Grant read only permission to our current user.

icacls "key/path/x.pem" /grant:r "$(env:USERNAME):R"

⇒ STEP 5 :- Try SSH again.

ssh -i "key/path/x.pem" ec2-user@<IPv4>

⇒ after successfully connecting the user of will be :-

[ec2-user@ip-172-31-44-53 ~]$ ls ✓

↓

this is private IPv4 address.

my for :- public IPv4 => 13.232.70.31
this ex. :- private IPv4 => 172.31.44.53

⇒ terminologies of Instance :-

Instance ID :- each instance has their own unique ID (identification)

Date: _____ Page No. _____

Topic: _____

⇒ Instance state :- current state of the running instance.

⇒ Instance type :- type of instance choose at launching time

⇒ Availability zone :- data center where our data is stored.

⇒ AMI ID :- AMI chosen before launching instance.

⇒ key-pair name :- name of our key pair

⇒ launch time :- of the instance or when was the instance re-started

⇒ root device :-

⇒ public dns :- It is similar to our domain name.
if we paste this public dns it will resolve us to the public ipv4 address.
By default this public DNS is ugly but one can rename it using AWS route 53.

Date: _____ Page No: _____

Topic: _____

⇒ private DNS ⇒ It is within the Intranet
⇒ this is a DNS name or domain name if we are using a private network.

⇒ private IPv4 ⇒ It is again within the intranet.

⇒ from onwards we'll do everything in the user only (AWS EC2 use).

⇒ to install apache (Httpd) :-

⇒ sudo yum install -y Httpd

:

complete !

⇒ creating a basics web page & accessing it through public DNS & public IPv4 address.
⇒ public DNS will resolve this to public IPv4 address

⇒ starting the apache server :-
sudo service Httpd start

: ⇒ enter command.
w ⇒ means write/save the file.
q ⇒ quit the editor.

Date: _____ Page No. _____

Apache's default web root directory:

vi/vim <

press ESC ⇒ vi /var/www/html/index.html
type :wq. <body>
ENTER. <h1> Hello </h1>
save & exit < </body>

:q! ⇒ exit without saving.

⇒ then, paste public DNS or public IPv4, both will resolve in the above page/.

i ⇒ inserting.
⇒ nslookup < public DNS >
resolve it to public IPv4 address.

a ⇒ appending.
⇒ If any error :-

o ⇒ open a new line below
Check EC2 security groups inbound rules :-
Go to AWS Console → EC2 → Instances :
select instance → scroll to Security groups:
click security group → Inbound rules.
→ add if not.

Type.	protocol.	port range.	source.
HTTP	TCP	80	0.0.0.0/0

⇒ AWS EC2 console → security groups → launch-wizard-1.
⇒ click on inbound rules tab, edit, add that rule & save rules.

Logout ⇒ exit [ENTER]
CTRL + D.

AGL | Premium ka Pappa START FROM LEC. - 2.5:00

Elastic IP, Instance Metadata, Public Key.

21/06/20

Date: _____ Page No. _____

2.5

Topic: _____

⇒ Every time we start & stop our Instance in AWS, the IPv4 & Public DNS (IPv4) gets changed.

⇒ No Suitable if we have our app hosted one one / static public IP (IPv4).

⇒ to solve this we use Elastic IP (under NETWORK & SECURITY tab).

⇒ When we allocate & attach a Elastic IP to our application, No matter how many times we start & stop our Instances the IPv4 won't change (it will be static once given).

⇒ Elastic IPs → Allocate Elastic IP address → keep the default settings (IPv4 pool address, network border group, tags, Global static IP addresses) → click on allocate →

Date: _____ Page No. _____

Topic: _____

Allocated successfully. (43.205.221.10).

After allocating a Elastic IP ⇒

under actions option → (click on associate Elastic IP address → choose (resource type → Instance. (Instance id. Instance → name of Instance on which we want to allocate this Elastic IP (in our case AWS FOUNDATION DEMO) → Private IP (172.31.44.53) address (choose) → Reassociation (check yes or NO).

⇒ If Reassociation (yes).

NOTE:- If you associate an Elastic IP for instances address with an instance that already has an Elastic IP address associated, the previously associated IP address will be dissociated, but the address will still be allocated to your ACCOUNT.

Date: _____ Page No. _____

Topic: _____

⇒ If no Private IP address is specified, the Elastic IP address will be associated with the primary private IP address.

⇒ click associate.

[associated successfully with instance i-01a9a0945/a65addb].

⇒ Now, even if we start/stop our instance many times, the PUBLIC IPv4 address & the PUBLIC DNS(IPv4) WON'T change.

⇒ curl http://169.254.169.254/latest/
output :- dynamic, metadata.

⇒ curl http://169.254.169.254/latest/metadata
meta-data/, (this returns)
ami-id, events/, mac, network/, public-ipv4, instance-id, instance-type, etc. etc...

...meta-data/instance-id
...meta-data/ami-id

Date: _____ Page No. _____

Topic: _____

Q. ⇒ How to check no. of vCPU, threads, no. of cores, no. of cores per thread, & no. of sockets.

A. ⇒ lscpu

Q. ⇒ WHERE the public Key is stored in EC2 instance

A. ⇒ we ~~are~~ logged in in EC2 instance with user-id

⇒ type ⇒ id.

⇒ returns the id name of currently logged in EC2 instance (ec2-user).

⇒ type ⇒ cd.

⇒ we will reach to the home directory.

⇒ type ⇒ ls -la

⇒ we will find a ~~file~~ directory (.ssh).

⇒ type ⇒ [cd .ssh] then [ls]
we will find authorized_keys file.
⇒ open it ⇒ [more authorized_keys]
⇒ returns us the public key of the key pair.

AMI Copy, AMI permissions.

Date: _____ Page No. _____

2.6

Topic: AMI Again.

⇒ private key ⇒ we downloaded as key-pair (already used to log in EC2 instance).

⇒ public key ⇒ is stored in the home directory of the user which we have used to login into this EC2 instance.

PUBLIC KEY

⇒ ssh-rsa AAA.....TEbw/ aws-foundation-demo-key-pair

⇒ ssh-rsa public key private key

// 2.6 //.

⇒ what if we redraint all these things (created launch an instance, logged in, installed apache, started apache services, also created a basic web page with some text) in multiple EC2 instances.

⇒ say in 25 EC2 instances.

⇒ well welcome to AMI Again.

Date: _____ Page No. _____

Topic: _____

OS
JVM
Additional Volumes
Tomcat
Java SDK
Application

How to do these same installation or configuration on 25 different EC2 Instances.

2 ways :-

1.) Manually :- login each instance & installed set up configurations.

easy way :- 2.) AMI Again :- using AMI.

Ex :- our own EC2 instance.

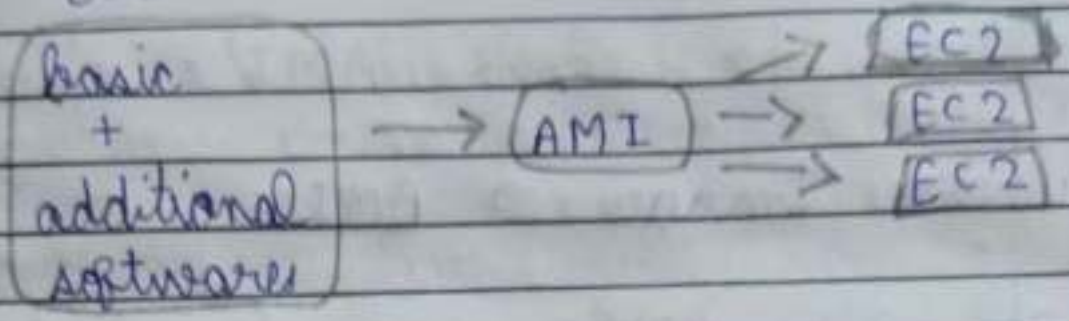
Amazon linux
Python, Ruby, Perl
AWS CLI
} → basic softwares used in AMI (built-in).

Apach, configured apache basic web page
} → additional softwares

Date: _____ Page No. _____

Topic: _____

EC2



⇒ all EC2's will have the exact configuration like basic + additional softwares (EC2).

⇒ this AMI is custom AMI that contains the basic + additional softwares.

⇒ sudo chkconfig httpd on

⇒ this command will automatically start httpd or apache for us no need to start it manually when restarted the EC2 instance.

⇒ CUSTOM AMI:-

select EC2 instance → actions option → Image → create image (give image name & discription) → click create Image

⇒ Successfully message.

Date: _____ Page No. _____

Topic: _____

→ How to check AMI's :-

under Images → AMIs

all the AMIs will be listed.

they have :- AMI name, AMI id, status, source, visibility, Creation date, Owner, etc etc, ...

status → pending, available.

Now ⇒ we will create 2 instances from this custom AMI (EC2)

2 way :-

1.) from Images → AMIs → select AMI → directly click launch.

2.) traditional way by Instances → launch instance → choose AMI (our own custom AMI) ⇒ (from My AMIs) → & the rest of the process is same.

⇒ we will create 2 EC2 instance & both should result the same

Date: _____ Page No. _____

Topic: _____

output as of our EC2 instance whose AMI we have created.

→ Yes, Same output.

1.) Create AMI from an instance.

2.) Launch multiple instances from it.

3.) Copy AMI (from one region to another)

4.) AMI permissions.

3.) [AMI (us-east-1)] ---- Copy ----→ [AMI (ap-south-1)]

⇒ Under Images section → AMIs → select AMI → go to Actions → copy AMI → select (destination region, name, description, encryption) → click copy AMI → close.

⇒ change the region & check in AMIs & create & run that Instance & it will also give the same output.

Date: _____ Page No. _____

Topic: _____

→ For our example we have selected destination region as Europe (London). done.

// make sure of security rules // check NOTES.

4.) AMI permissions.

public :- everyone that has an AWS account will be able to use my AMI.

private :- only those users will be able to use my AMI whose AWS account no. i have added in add permissions.

under Images → AMIs → select AMI → Actions → Modify Image permissions → public or private.

Edit AMI permissions

NOTE :- we won't be able to change from private to public once created. (today's date 21/oct/25).

21/05/25.

ENI ⇒ Elastic Network Interface.

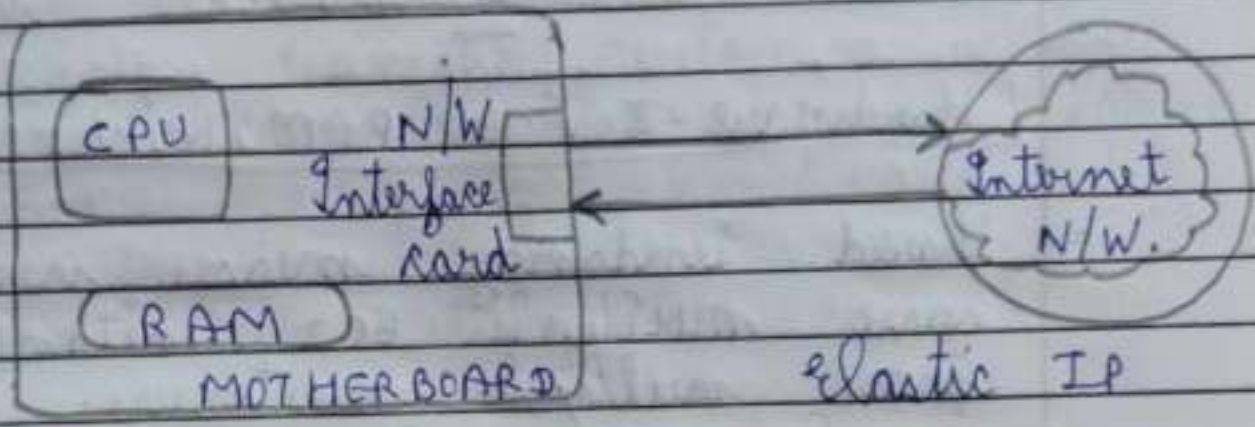
ENI, Instance Tenancy, dedicated Instance & host.

Date: _____ Page No. _____

2.7

Topic: _____

⇒ Network Interface is the interface between a computer and an internet or network.
Network IO happens through n/w Interface cards.



⇒ N/W Interface contain:- Elastic IP, public IP, private IP, security groups.

⇒ Instance Tenancy:- 3 types.

1) Shared / Default.

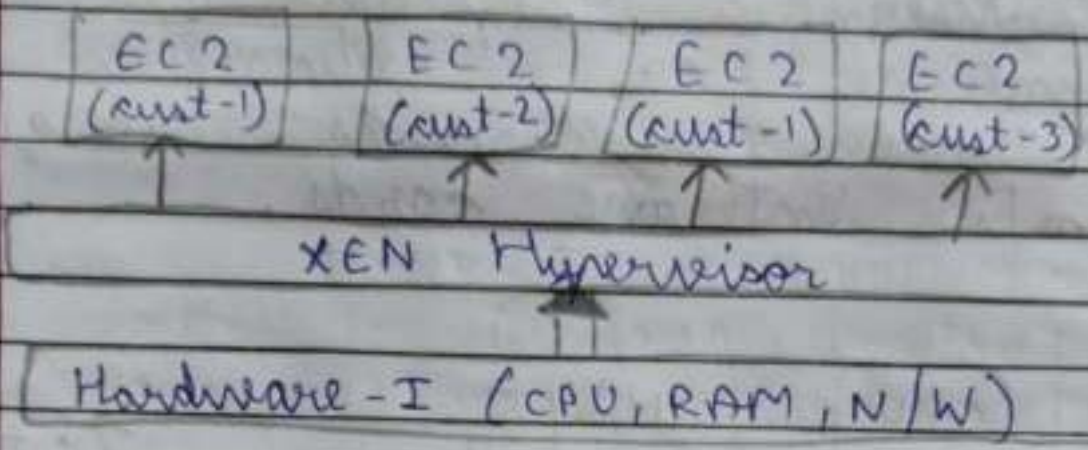
2) Dedicated.

3) Dedicated host.

Date: _____ Page No. _____

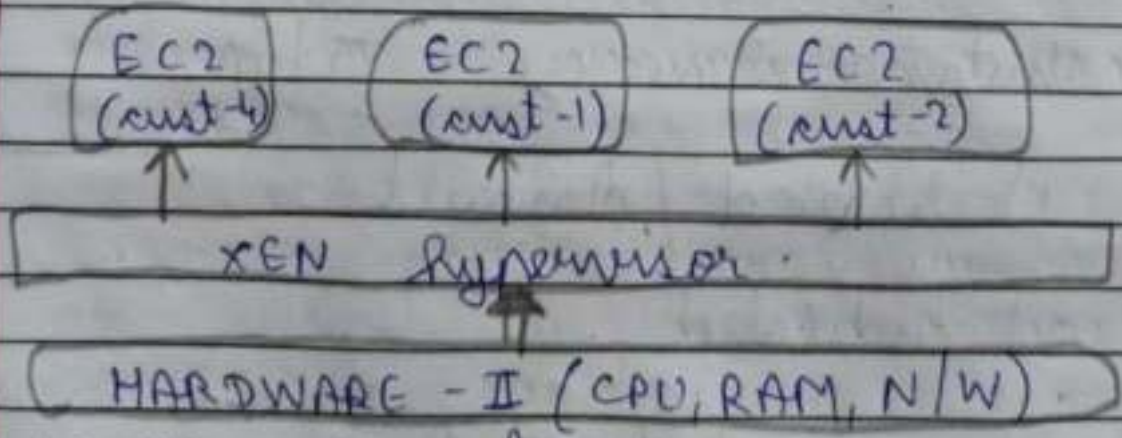
Topic: _____

1.) Shared / Default :-



→ Shared Tendancy means we can have multiple EC2 instances from multiple customers.

→ XEN hypervisor has launched all these EC2 instances that belongs to different AWS Account but on the same hardware.



→ same as above.

Date: _____ Page No. _____

Topic: _____

⇒ So, there are multiple hardwares in an AWS data center & XEN Hypervisor is installed on top of all of them.

⇒ In short, multiple EC2 instances from different customers accounts can be launched in the same ~~the~~ underlying hardware.

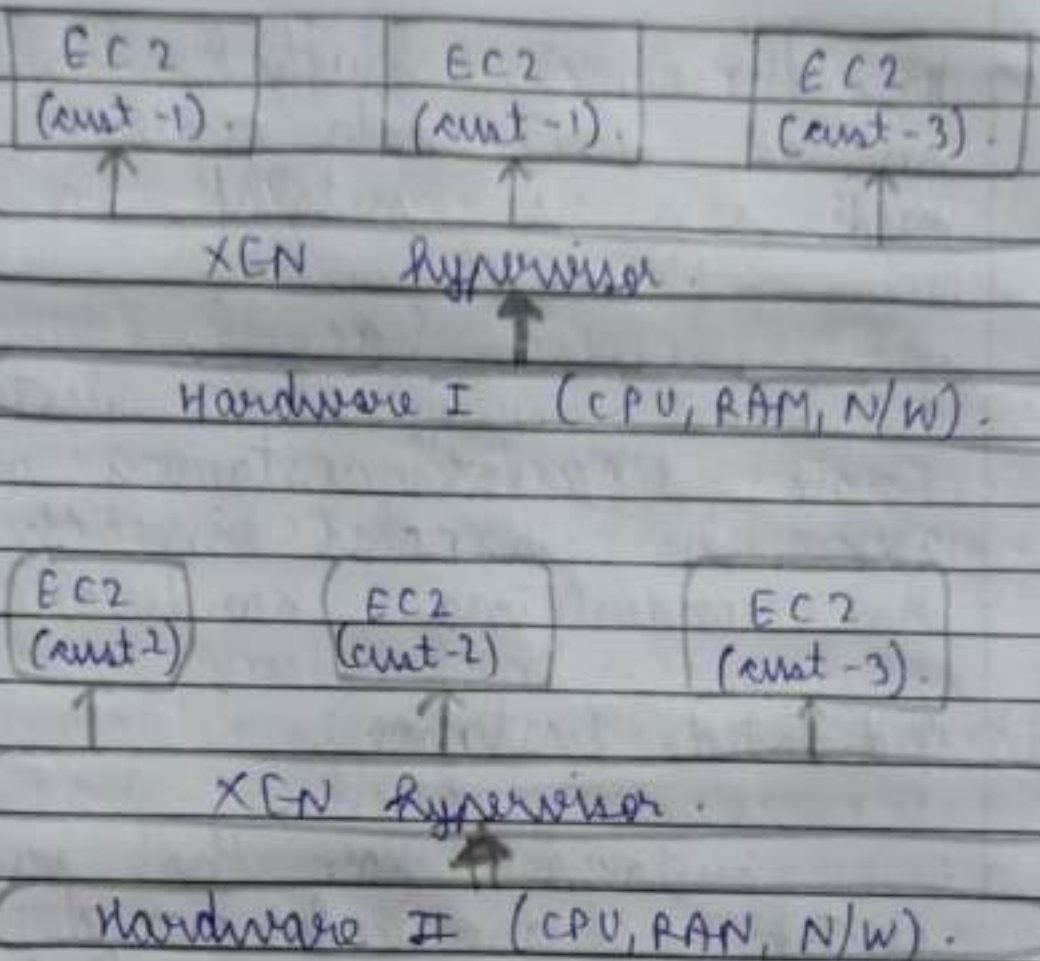
2.) Dedicated Instance :-

⇒ EC2 instances for the same customers are launched in the same hardware but ~~there~~ it can also ~~might be~~ have other instances of the ~~system~~ different customers as well.

⇒ EC2 Instances from the same customers will always be launched in the same hardware only.

Date: _____ Page No. _____

Topic: _____

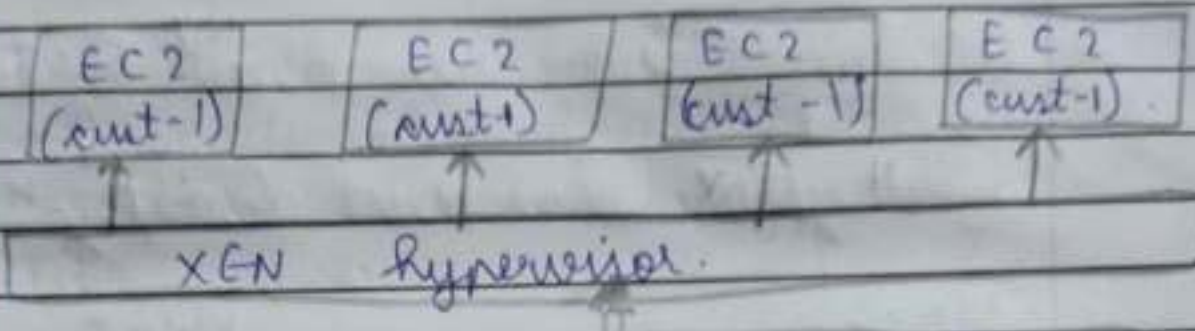


⇒ EC2 of cust 1 & cust 2 are running dedicated Instance, where as cust 3 ~~does~~ is not running a dedicated EC2 instance.

3) dedicated host :- You keep or lease the hardware for a particular customer.

Date: _____ Page No: _____

Topic: _____



IT IS CHARGEABLE

⇒ purchase → under Instance section
→ dedicated hosts → allocate a host
→ select instance type, AZs, Quantity, etc. → allocate Host.

Tenancy ⇒ dedicated host, then select Host ⇒ name of dedicated host.
affinity ⇒ when instance is restarted it will again ~~select the host~~ start in dedicated host.

⇒ // Instance Restart //.

what happens during a restart.
→ Say, first the EC2 instance is launched on hardware-I & when restarted there are chances it might get restarted on hardware-II. (Reason of IPv4 change).

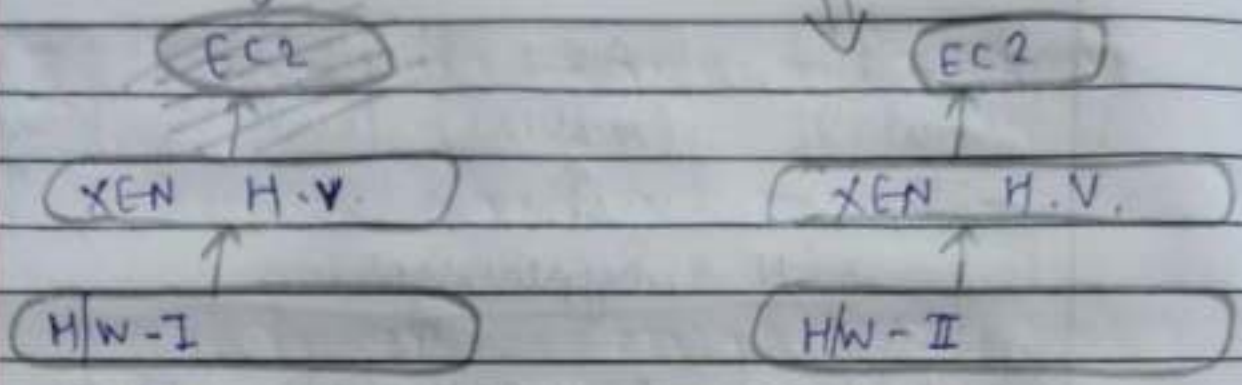
launched

AGL | Premium ka Pappa

Date: _____ Page No. _____

Topic: _____

first when restarted.



→ During the Instance restart it might change its tenancy.

→ to prevent this either use dedicated Instance or dedicated host.

⇒ // ~~Enhanced~~ NETWORKING //
Enhanced

⇒ for faster network performance & increase in bandwidth.

⇒ It is supported for few instance types, not all supports Enhanced networking.
It uses :-

⇒ SR-IOV ⇒ Single Room I/O virtualization technique.

Date: _____ Page No. _____

Topic: _____

- faster N/W performance.
- No additional charge for using enhanced networking.
- Two types :-

> Intel 82599 Virtual Function (VF) Interface - up to 10 GBPS.
> Elastic Network Adapter [ENA] - up to 25 GBPS.

Intel 82599 VF.	ENA.
C3, C4, D2, I2, R3, M4.	C5, M5, R4, I3, X1, G3, P2, P3, F1, H1.

⇒ // Placement Group //.

- It is a logical grouping of multiple EC2 instances which are placed together in a single or multiple AZs.
- needed for workload, wherein the connectivity between your servers is very critical.

- Ex :- High performance Computing clusters (HPC clusters).

Date: _____ Page No. _____

Topic: _____

- 2 types of placement groups.

1.) Cluster placement group :-

- created in a single AZ's.
- puts all the EC2 instances under one single AZ, & maintain network connectivity b/w them.

2.) Spread placement Group :-

- EC2 instances are spread ~~over more than~~ across multiple AZ's & maintain N/W connectivity b/w them.

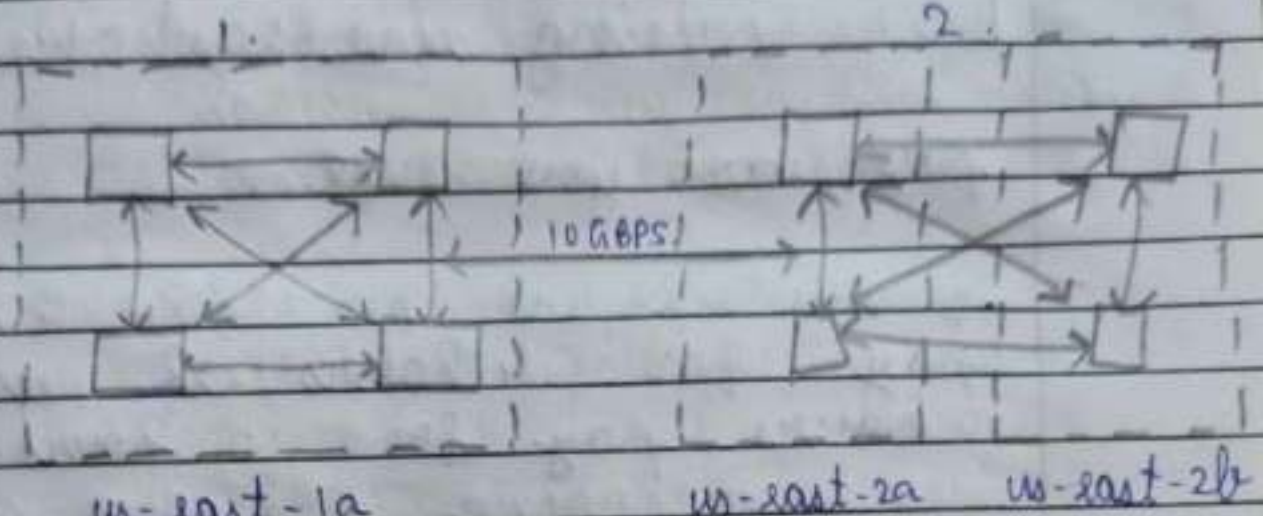
NOTE :- all the EC2 instances launched in the placement group should support enhanced N/Wing.

3.) Partition (came (today 21/06/25).

- a logical grouping of instances that are placed on separate hardware racks to ensure fault ~~to~~ isolation & improve performance.

Date: _____ Page No. _____

Topic: _____



⇒ under NETWORK & Security Section → Placements Groups → create placement groups → give name & strategy → click create.

⇒ while creating a new instance add a placement group in instance.

21/05/25.

Pricing, Reserved, & spot Instances, design patterns.

Date: _____ Page No. _____

Topic: _____

2.8

⇒ EC2 pricing (us-east-1) :-

- pay as you use.
- if instance is on for 4 hours, pay for 4 hours, if up for 10 hours, pay for 10 hours.
- Free Tier : 750 Hours per month of Amazon linux, RHEL, SLES, Windows t2.micro single instance usage.
- on demand price : (us-east-1) :-
  - m5.large = $0.096/hour.
  - c5.large = $0.085/hour.
  - r4.large = $0.133/hour.
- data transfer in : free from anywhere.
- data transfer out : from EC2 to -S3, SES, Glacier, Dynamodb, etc. in same region = free.

while

Date: _____ Page No: _____

Topic: _____

- to another EC2 instances say S3, Glacier, SES, in different region = $0.02 / GB.

- EC2, RDS, ELB, Elasticache in same AZ = free with private IP. $0.10 / GB with public IP. ENI

- EC2, RDS, ELB, Elasticache in different AZ = $0.010 / GB.

SLA = 99.99% uptime / month for EC2 instances. service level agreements.

(PO)

⇒ // Purchasing Option - RI //.

- PO is nothing but discounted rates which we can get for our EC2 instances.
- Can save huge amount of cost with different purchasing options.
- Reserved instance ? 1 or 3 year term.

Date: _____ Page No. _____

Topic: _____

- Pricing (on-demand us-east-1 region)

M5. large = $0.192/hr.

for yearly = $0.192 * 24 * 365
= $1681.92.

- In purchasing options we have different payment type/ options.

	Payment type.	One time payment.	Total yearly cost	Saving
1.	No Upfront : $0	$89.79*12 = $1077.48		36%
2.	Partial Upfront : $512	$1020.08.		39%
3.	Full Upfront : $1003	$1003.		40%

⇒ Other Reserved Instance attribute for calculating price.

① - Instance type → size - L, XL, 2XL.
→ family - M4, C5, i3.

Date: _____ Page No: _____

Topic: _____

① - Scope → Region.
→ AZ (zonal).

② - Tenancy → Shared / Default.
→ Dedicated & Instances.

NOTE:- We are not able apply purchasing options discount on dedicated hosts (3rd type of tenancy does not come in picture here).

④ - Platform (OS) → Windows.
→ Linux.

(combining all 4) ↙

- Offering class → Standard.
→ Convertible.

- Standard Reserved Instance (SRI) :-
Instance type - size can be modified during the period/term of 1 or 3 years. But MOST of the attributes can not be modified.

Convertible (RI) :-

- Entire RI ~~can~~ be bought can be exchanged for another convertible RI.
all 4 can be changed.

Date: _____ Page No. _____

Topic: _____

- Normalization Factor :- (NF).

Instance size	Normalization factor.
nano	0.25.
micro	0.5
small	1
medium	2
large	4
x large	8
2x large	16
4x large	32
8x "	64
9x "	72 (8×l + x large)
10x "	80
12x "	96
16x "	128 (8×l + 8×l)
18x "	144
24x "	192

- NF is a no. associated with each every instance size.

- ~~Nor~~
- Normalization factor :-

Date: _____ Page No: _____

Topic: _____

- Regional RI → AZ & Instance size flexibility (default tenancy only).

- zonal RI capacity Reservation :-
  - us-east-1a
  - c4.xlarge.

DT → default tenancy.

⇒ Running Instance — RI bought

BOUGHT RI IN - ~~ZONE~~ AZ-ZONAL RI.

4 M3.large linux, default tenancy in AZ us-east-1a

, 4 M3.large, linux, DT, AZ- us-east-1a applied

REGIONAL RI.

Instance-type(M4) matters NOT size

(64 → ¼ price is discounted 16)

② M4.4xlarge Amazon linux, default tenancy in us-east-1b (xlarge || large)

, 4 M4.large, Amazon linux, DT, region us-east-1. applied (Ex. :- 100$ → 25$ off).

- c4.xlarge RHEL, dedicated tenancy in AZ us-east-1c.

c4 large, RHEL, DT, region us-east-1.

~~c4.xlarge in zonal us-east-1a~~, regional RI can have discount on AZ zonal RI won't. (c4.xlarge in us-east-1a & us-east-1b).

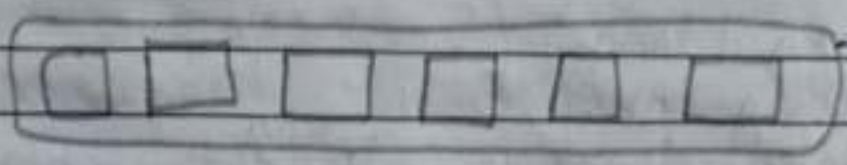
AGL | Premium ka Pappa

Date: _____ Page No. _____

Topic: _____

// SPOT INSTANCES //.

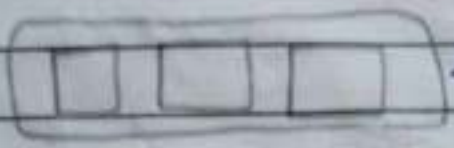
- final purchasing option is Spot Instances.
- spot instances enables us to request unused EC2 instances that furthermore reduces our cost.
- hourly price ~~for EC2~~ spot instances ~~are decided by EC2~~ is set by EC2. price can gradually increase or decrease depend on demand & supply.
- Works like stock markets - supply & demand.
- Spot Instance terminologies :-



- set of unused instances with a set of same OS, same instance type, within the same AZ.

Date: _____ Page No. _____

Topic: _____

-  ← SPOT fleet.

no. of EC2 instances that we will get it from the pool to run our workload.

- we will have to pay some price for spot fleet.

NOTE :- Instance is terminated if spot price increases than bid price.

- bid price : current running price of c4.xlarge instance is say 10$, if i give a bid price of 15$. then, i will get the no of EC2 instances that i have asked for. instance
- after some time spot price gets increased ~~by~~ to 16$, then my instances will be terminated, because my spot price is higher than my bid price
- Significant price reduction.

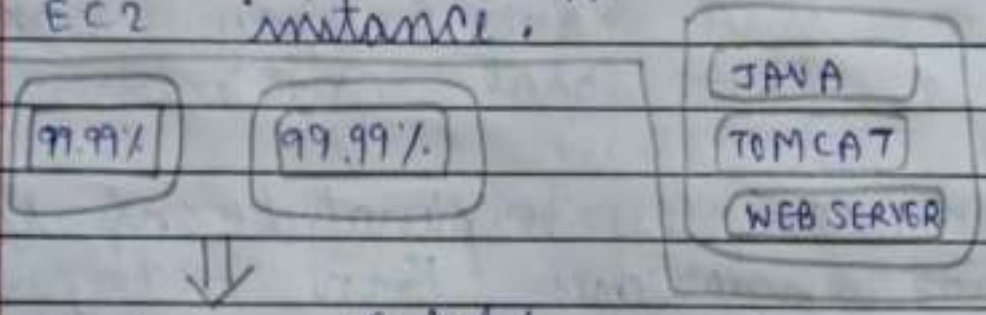
Date: _____ Page No. _____

Topic: _____

- use cases :-

// design patterns - Application Hosting //

- not much design pattern in EC2.
- create your application inside of EC2 instance.



- High availability using multiple EC2 instances.

- Migrate on-premise servers to EC2 instances.

EC2 SUMMARY.

Date: _____ Page No: _____

Topic: _____

2.9.

- Regions & Availability Zones (AZs).
- Amazon Machine Images.
- Instance types.
- vCPU, Key Pair, Root Device Volume.
- Instance metadata
- Instance tenancy attributes - 3.
- ENI
- Cost Optimization - Reserved Instances.

QUIZ (2)

⇒ size of limit of Amazon General purpose EBS ⇒ 16 TB.

⇒ EBS ⇒ Elastic Block Store.

⇒ check the already created file system details ? ⇒ df -h.

22/06/25:

AWS FOUNDATION, Storage

3.1.

Pre EBS-file system & storage Layer Concepts.

Date _____ Page No. _____

EBS.

Topic _____

⇒ Agenda :-

- Pre-EBS : Storage layers.
- Elastic Block store.
- Volumes types.
- Snapshots.
- Demo.
- Pricing.
- Design Patterns.

⇒ Pre EBS ⇒ File System basics as a

- AWS SOLUTION Architect, we will never see working inside an EC2 Instance, we will never create any file system.
- But its good to have Knowledge.

L = level.

Date: _____ Page No. _____

Topic: _____

WINDOWS.

WHAT is a file SYSTEM?

HDD
↓
drives
↓
folders
↓
files.

// HDD is partitioned into C & D drives.

C:
- fol1
  - resfol1
  - file1
- fol2
  - file1
  - resfol1
- file1.

D:
- fol1
  - resfol1
  - resfol2
  - file.

UNIX/LINUX.

L1 (d1) / → root directory.

L2: bin, fileabc, avs

L3: xyz (d2) abc, abc.txt, bnd (d3), filesss

L4: filess, file1, file2

bnd → (/avs/bnd)

(/bin/abc/file2).

⇒ d2 ⇒ (/bin/abc).

⇒ drives (d1, d2, d3). we can have multiple drives as shown.

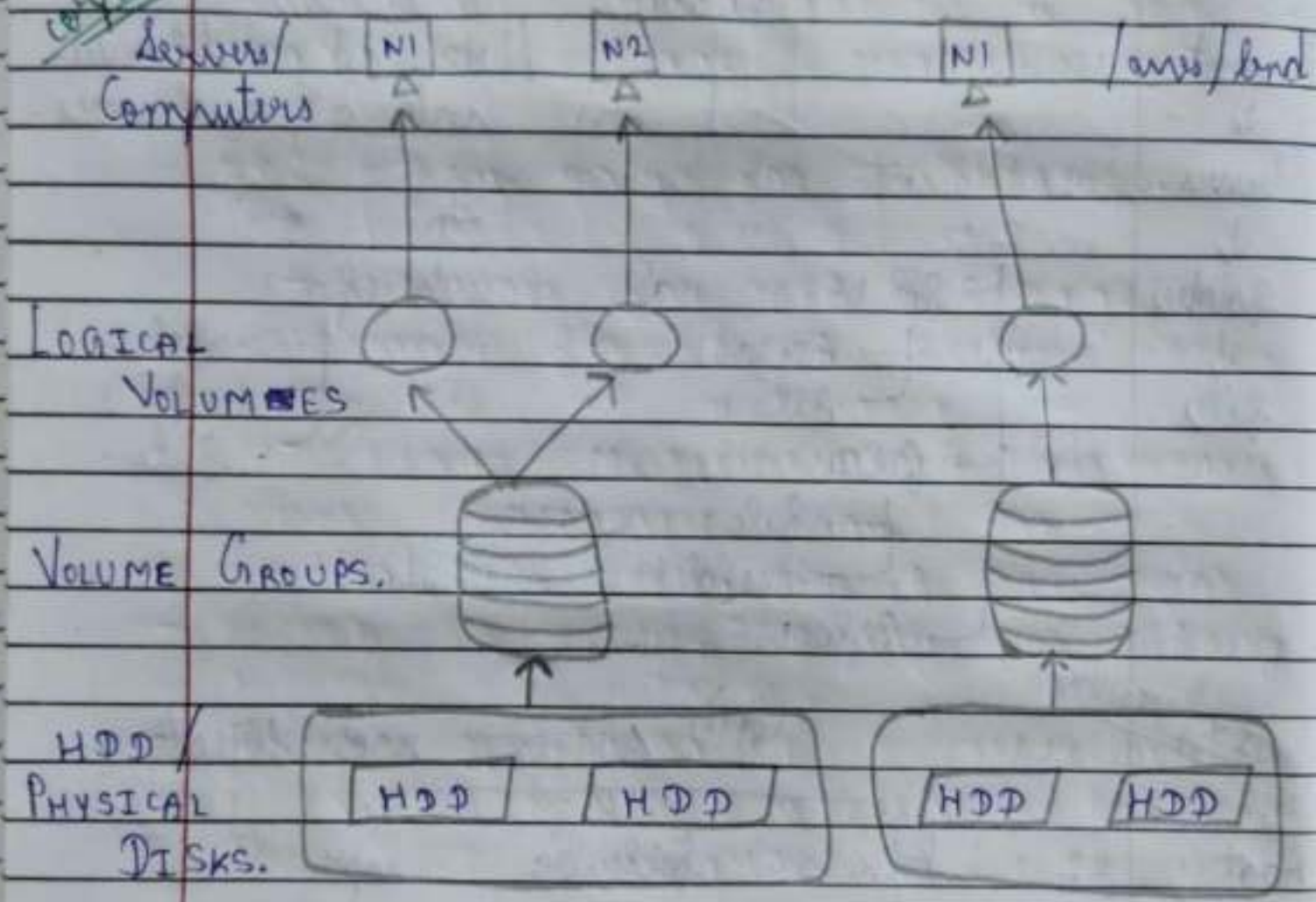
⇒ drives (hard disks).

Date: _____ Page No. _____

Topic: _____

Corporate World :-

// Storage layers. //



Img 1

Img 2.

EBS Concepts, Throughput - IOPS.

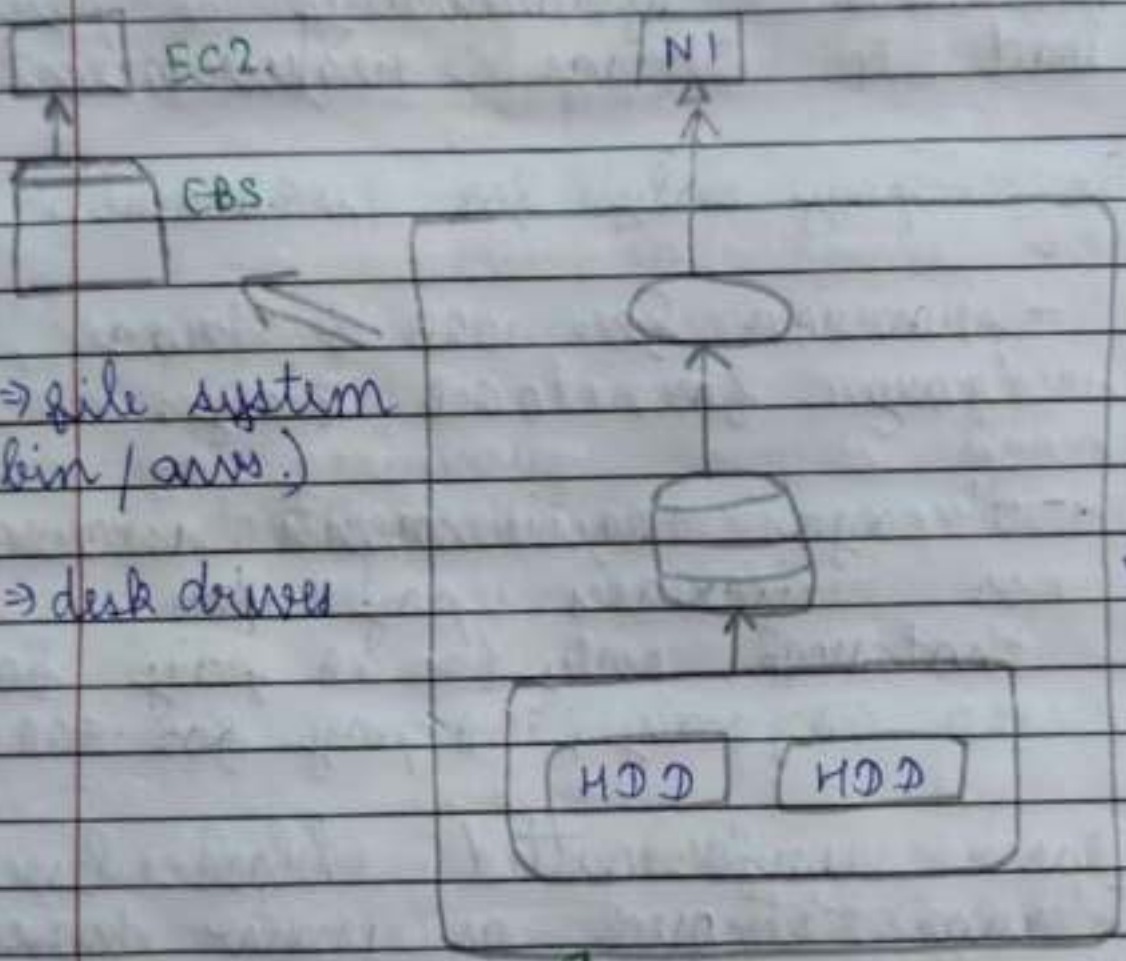
22/06/25.

Date: _____ Page No: _____

Topic: _____

3.2.

// EBS //



Img. 2.

⇒ when we use EBS, NO NEED TO WORRY about these things (logical Volumes, Volume groups, HDDs).

⇒ we get the EBS Volume, & Attach it to EC2 Instance & use it.

Date: _____ Page No. _____

Topic: _____

- "Elastic" because it can be increased dynamically as per load & storage requirement.

Ex:- pay only for what use.

- provision for 10GB of storage, pay for 10GB only.

- storage requirements increases or decreases pay for that much only (20 GB pay for 20 GB only, 5GB pay for 5GB only).

- Raw unformatted block level storage exposed as raw device to the EC2 instance.

- EBS Volumes persits independently from the life of EC2 instance.

- If EC2 instance is terminated, but the Volume is still persits (there) and can be used for later purposes.

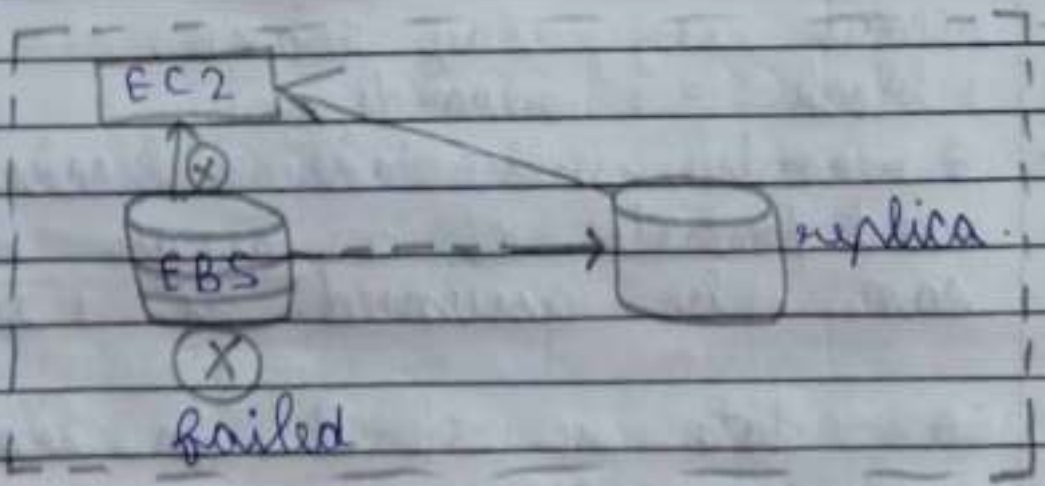
Date: _____ Page No. _____

Topic: _____

- An EBS Volume is automatically replicated within an Availability Zone.

- Use Case created a EBS Volume that can be Volume would be replicated within the same AZ.

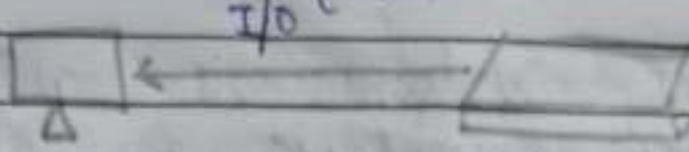
- if primary volume gets down the secondary Volume would be up without even letting you know.

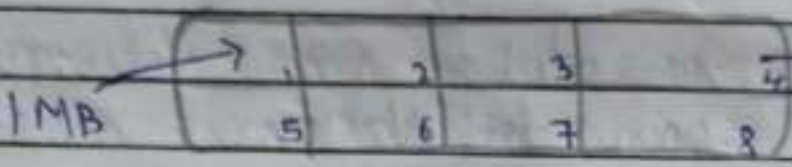


- failure tolerance
- Random read & Write.
- long ~ reads & Writes
  Continuous

Date: _____ Page No. _____

Topic: ~~(read or write operations)~~





- minimum unit of I/O is block.
- block is a grouping of one or more records depending on the size.

- record size say 500KB.
- 1 block = 2 records.
- 5 records will take 3 blocks.
- read / write of a single block can be assumed as 1 I/O operation

- EBS data is stored in blocks.

1) Throughput - Sequential access.

2) IOPS - Input/output operations per second.
- Random access.

Date: _____ Page No. _____

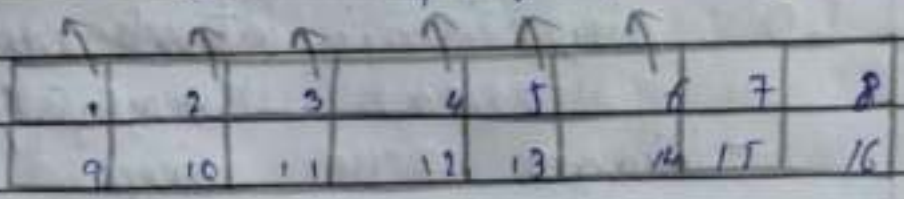
Topic: _____

1.) Throughput :-

- Sequential Transfer rate ~~at~~ which an SSD or HDD will maintain continuously.

~~2.)~~ - read / write data sequentially one after the other is measured as throughput.

- Ex :- → sequentially read/write.



ASSUMING :-

1 Block = 1 MB.

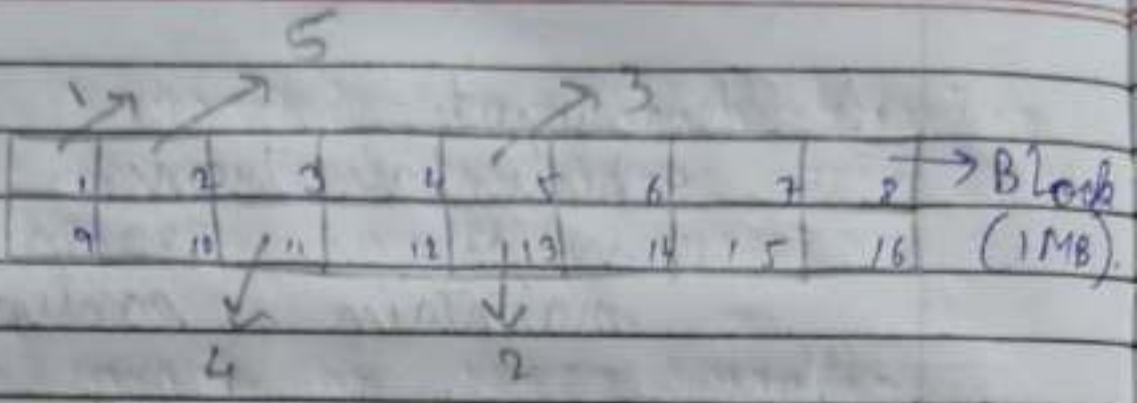
6 Block sequentially per second = 6 MBPS (6 MB/S).

2.) IOPS - It is the measure of no. of I/O operations a drive, SSD or HDD, will handle per second with each block being read from or written to a random location of the disk.

Date: _____ Page No. _____

Topic: _____

Ex:-



5 operations ⇒ 1, 13, 5, 11, 2.

IOPS = 5.

- IO size :- important role in IOPS & throughput.
  - bigger IO size better throughput.
  - smaller IO size gives better IOPS.

Volume types - GP2, IO1, ST1 & SC1.

22/06/25.

Date: _____ Page No. _____

Topic: _____

3.3

- Volume types

- SSD backed Volumes :-
  - very good for random IO operations
- HDD backed Volumes :- better for sequential reads/writes.

SSD backed.

(1) - GP2 - general purpose SSD.
- single-digit milliseconds.
- Baseline performance is 3 IOPS/GB with a min of 100 IOPS & max of 10000 IOPS.

⇒ Ex :- - provision of 100 GB of GP2 storage
- baseline performance would be of 300 IOPS.
- provision of 1500 GB of GP2 storage
- baseline performance would be of 4500 IOPS.
- for 1 GB of GP2 storage, will still be 100 IOPS, because it can not get less than that.

Date _____ Page No. _____

Topic _____

- Burst bucket :- (works similar to saving a/c)
- Max burst performance 3000 IOPS.
- capability of burst bucket.

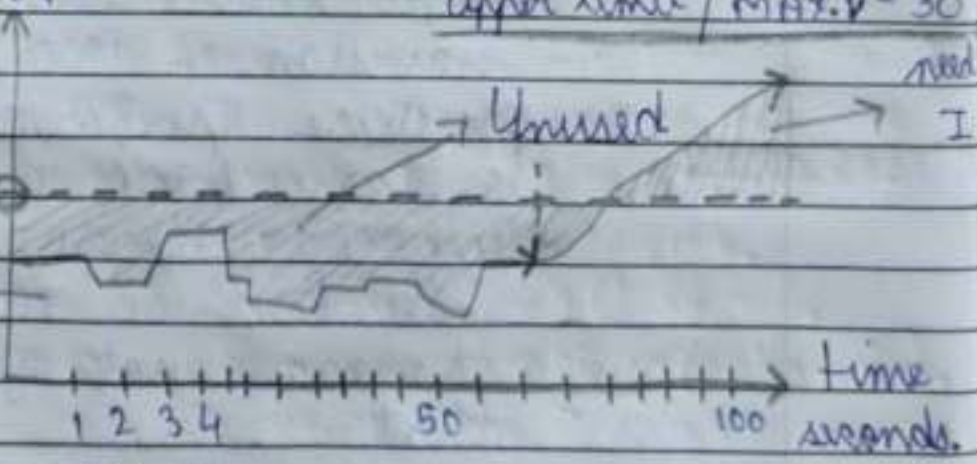
Ex :- salary of 10000/month.
- analogy thi. saving some amount of salary & getting a car.

- If our application does not consume EBS Volumes baseline performance, then the ~~IOPS~~ unused IOPS, would be stored in burst bucket. (similar to saving A/c).

Better understand with a graph.

Ex :- IOPS

upper limit / MAX. burst = 3000.



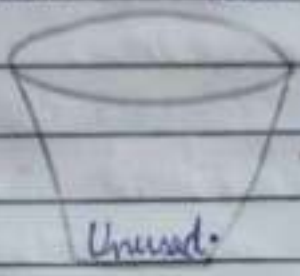
Storage = 500 GB.

Baseline IOPS = 1500 (500x3).

Date: _____ Page No. _____

Topic: _____

BURST BUCKET.



MAX = 5.4 Million.

3000 IOPS for 30 mins. Continuously

GP2 of 1300 GB, then
Baseline IOPS = 3900 ~~GB~~ IOPS. [More than ~~max~~ 3000 IOPS, which is a max burst performance. [In this case, concept of burst bucket will not play any role].

this additional IOPS is provided from the burst bucket.

00.
of more
OPS.

→ our application should not exceed the upper limit.

- Max throughput per Volume 160 MB/S (16KB IO size), if used for sequential read & write.

orange ⇒ gp2.

Date: _____ Page No. _____

Topic: _____

⇒ Provision ~~EC2~~ GP2 Volume in the Console.

2 Ways.

1) • in the launch of the instance.
- launch instance > configure storages. (Volume type).

12/04/25.

2.) EC2 Dashboard :- below is the example & steps of gp3 in blue. under Elastic Block Store section > create Volume > select

gp3, ← Volume type,
gp2, 100 Size (min 1GiB, MAX 16384 GiB),
io1, 300 IOPS (MIN 3000 & MAX 16000 IOPS),
io2, NOT applicable Throughput (MiB/s) 3000 IOPS
sc1, (MIN: 125MiB, MAX: 1000MiB,
st1, Baseline : 125 MiB/s),
Magnetic. AZ, Snapshot id (optional),
Encryption > create Volume.

JAI SHREE GANESHA.

Date: _____ Page No. _____

Topic: _____

(2.) IO1 - Provisioned SSD. (SSD backed).

- designed mainly for database workloads using io1,
- IOPS can be explicitly mentioned and is not dependent on the Volume size.

> From 100 to 32000 IOPS can be provisioned per Volume.

- thing to Know maximum ratio of provisioned IOPS to Volume size can not be more than 50:1 times.
- Ex :- 100GiB IO1 than, IOPS can not be more than 5000 IOPS.
  size
- we can provide IOPS & Size explicitly & IOPS irrespective of size but not More than 50 times.

> Max throughput per Volume 500MB/s.
> same process of creation.

Date: _____ Page No. _____

Topic: _____

(3) ST1 - Throughput optimized HDD
HDD based backed.

- good for sequential workload such as data warehouse, log processing, & ETL (extract, transform, load), big data analytics.

- performance is measured in terms of throughput not in terms of IOPS

> Baseline performance is 40 MB/s per TB of provisioned storage, with a max of 500 MB/s per Volume.

- Ex:- 3TB storage size than, the baseline performance will be 120 MB/s.

- min 20 MB/s & max 500 MB/s.

Date: _____ Page No. _____

Topic: _____

EC2 → dashboard > EBS section > Volumes > create Volume > select

Volume type, st1

Size, [MIN :-> 125 GiB, MAX :-> 16384 GiB]

IOPS, (NOT APPLICABLE)

throughput, (5/31), MAX :-> 500/500.

baseline : 40 MB/s per TB.

AZ, Snapshot id (optional),

Encryption.

- provides concept of burst performance (same concept as gp 2).

> Burst performance 250 MB/s per TB with a max of 500 MB/s per Volume.

(4.) SC1 - Cold Storage HDD. (HDD backed).

performance

- lesser baseline ; than st1, & performance is measured in throughput.

Date: _____ Page No. _____

Topic: _____

- also provides concept of burst performance.

> Baseline performance is 12 MB/s per TB of provisioned storage, with a max of 192 MB/s per Volume.

> Burstable performance varies from 80 MB/s per TB with a max of 250 MB/s per Volume.

- EC2 dashboard > EBS section > Volumes > create Volume > select
Volume type, * SC1.
1024. Size, min 125 & MAX 16384 GiB.
IOPS, NOT APPLICABLE.
12/80. throughput, 2/10 * to 192/250.
AZ, Snapshot-id (optional),
Encryption.

Volume operations.

22/06/25.

Date: _____ Page No. _____

Topic: _____

3.4

Volume type.	Size limit.	Maximum IOPS	Maximum throughput	Maximum Burst.
GP2	~~1GB-16TB~~ 1GB-16TB	10000	160MB/s	3000
IO1	4GB-16TB	32000	500MB/s	NA
ST1	500GB-16TB	500	500MB/s	500MB/s
SC1	500GB-16TB	250	192MB/s	250MB/s

⟹ // VOLUME OPERATIONS //

2 types.

1] at the time of creating EC2 Instance
2] creating Volumes separately & attaching them.

1.] launch Instance > give name ➤, select Amazon linux server, select instance type (t2.micro), key pair (aws-foundation-key-pair.pem) select network (existing security group (launch-wizard-1)), configure storage (add new Volume> 5GiB, gp2 > launch Instance.

⇒ go to key pair location & run cmd,
⇒ ssh -i "key-pair.pem" ec2-user@3.110.219.77,

running Instance's public key.

Date: _____ Page No. _____

Topic: _____

which devices are

⇒ to know all the ; connect to your EC2 instance type :-

```
lsblk -p (LSBLK -P)
/dev/xvda        8 G.  disk        dev
├─/dev/xvda1     8 G   part  /
├─/dev/xvda127   1 M   part
└─/dev/xvda128   10 M  part  /boot/efi
/dev/xvdb        5 G   disk
```

⇒ under EBS selected Instance go to storage tab option > root device name (/dev/xvda), block devices (/dev/xvda and /dev/sdb)

⇒ this is the attached Volume.

⇒ name has changed from /dev/sdb to /dev/xvdb due to virtualization.

⇒ Now, we will attach a file system to our custom / created Volume.

Date: _____ Page No. _____

Topic: _____

pwd
/home / ec2-user
mkdir ebs-demo
ls -l
drwxr-xr-x ..... ebs-demo

directory.

Before attaching our device to file system we will have to format the device.

ebs

⇒ think this way :-
we buy a USB drive & plug it & start using because they are pre-formatted.
⇒ this is not the case in our corporate storages all these devices are raw devices, they are not formatted (unformatted) devices.
⇒ format ~~them~~ it as per our file systems & then use it

⇒ Windows file format are :-
FAT (FILE allocation table), NTFS (New Technology file system, & exFAT (EXTENDED file allocation table.

Date: _____ Page No. _____

Topic: _____

⇒ linux / unix based file formats are :- ext2, ext3, ext4, xfs, (Commonly used.)

⇒ Command to check whether the device is formatted or not.

sudo file -s /dev/xvdb
/dev/xvdb : data

⇒ if the output is data means the device is not formatted

⇒ Command to format the device is :-

sudo mkfs -t ext4 /dev/xvdb

mkfs → make a file system
-t → type of
ext4 → with device /dev/xvdb

⇒ now check again this time there will be no data.

⇒ hence, we have formatted our raw device.

Date: _____ Page No. _____

Topic: _____

⇒ now attach our ebs-demo created file system to this device.

Command to do so is :-

ls
ebs-demo

sudo mount /dev/xvdb /home/ec2-user/ebs-demo/

⇒ now check the mountpoints using :-
lsblk -f
/dev/xvdb 5G disk /home/ec2-user/ebs-demo

⇒ Hence, we have mapped our file system (ebs-demo).

2.) Creating Volume separately & attaching them.

AZ of our creating volume should be same as EC2 instance's AZ.

Date _____ Page No. _____

Topic _____

⇒ create volume > GP2 > 2 > AZ same as EC2 instance (ap-south-1b) > create volume.

⇒ State of this volume will be "available" (NOT attached so), others that are being used are having State as "in-use".

⇒ to attach this volume :-

~~ato~~ select this volume > actions > attach volume > give our instance name, & device name (/dev/sdf) > attach Volume.

⇒ attached successfully (state = in-use).

⇒ to see use "lsblk -f".

NAME	SIZE	TYPE	MOUNTPOINT
/dev/xvdf	2G	disk	

/dev/sdf to /dev/xvdf automatically

⇒ NO MOUNTPOINT, NOT mapped/attached to any file system device.

Date: _____ Page No. _____

Topic: _____

⇒ repeat the same process did in 1. WAY.

sudo file -s /dev/xvdf

/dev/xvdf : data

⇒ checking formatted or not, here unformatted.

sudo mkfs -t ext4 /dev/xvdf

⇒ formatting.

sudo file -s /dev/xvdf

⇒ checking again this time no data.

pwd

/home/ec2-user

mkdir ebs-demo2

ls -l

d--------- ebs-demo2

sudo mount /dev/xvdf /home/ec2-user/ebs-demo2/

lsblk -p

:

:

/dev/xvdf 26 disk /home/ec2-user/ebs-demo2

⇒ hence, created Volume, formatted & attached to now in-use, successfully.

Date _____ Page No. _____

Topic _____

⇒ for description or more details of Volume select the volume & it will be shown in the side or at the below of the page.

- Volume id - vol-0321.....21b
- size - 2 GiB
- type - gp2
- Status check - okay
- Volume state - in-use
- IOPS - 100
- throughput - -
- created - sun Jun 22 2025 19:23:54
- AZ - ap-south-1b
- many more.

⇒ to modify.
select Volume > actions > modify Volume > select Volume type, size [can not be decreased, can only be increased. > Modify > one more confirmation (Modify).

⇒ resized successfully done.

Date: _____ Page No. _____

Topic: _____

⇒ status in EC2 instance.

lsblk ↑
/dev/xvdf 4G disk .....
resized here.

⇒ status in file system.

df -h # filesystem mapping information
Filesystem Size used Avail Use% Mounted on
/dev/xvdf 2.0G 24K 1.8G 1% /home/...

⇒ sudo resize2fs /dev/xvdf
increase the size or resize the device as per the filesystem.

⇒ check now :- df -h
/dev/xvdf 3.9G 24K 3.7G 1% /home/....

⇒ Now in total we have 3 Volumes :-
/dev/xvda → ROOT Volume.
/dev/xvdb → Additional Volume.
/dev/xvdf → Additional Volume.

Date _____ Page No. _____

Topic _____

⇒ How to detach or delete or release a Volume.

→ First remove the mapping (unmount)

→ sudo umount /home/ec2-user/
els-demo2

→ removed or NOT check with "lsblk -f".

→ second go to AWS console & detach the Volume.

→ select Volume > actions > detach Volume > Detach.

→ Once detached, Volume State will be "available" (in-use to available)

→ select Volume > actions > delete Volume > delete (type delete) > delete.

⇒ once deleted can not be retrieved in any way, until & unless we have a snapshot.

SNAPSHOTS.

Date: _____ Page No. _____

Topic: _____

3.5

AWS

as a Solutions architect & SysOps administration you dont have to use these Commands.

// Snapshots. //

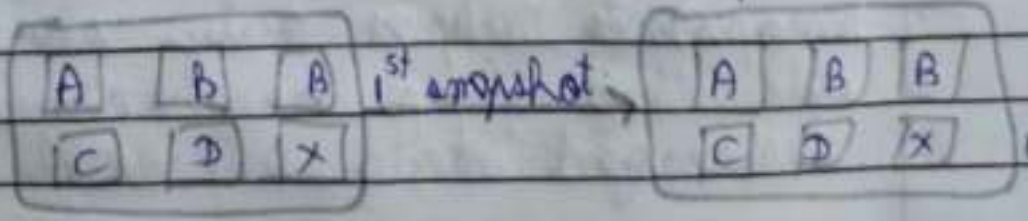
- data on EBS Volumes can be backed up - to provide point in time snapshots

> Snapshots are used to backup data on the EBS Volumes.
> All snapshots are incremental backups except for the first one.
> Snapshots are copied to Amazon S3.

S3 provides high amount of durability (99.99999999%) (11 nines) to be precise.

all blocks are copied.

Ex:

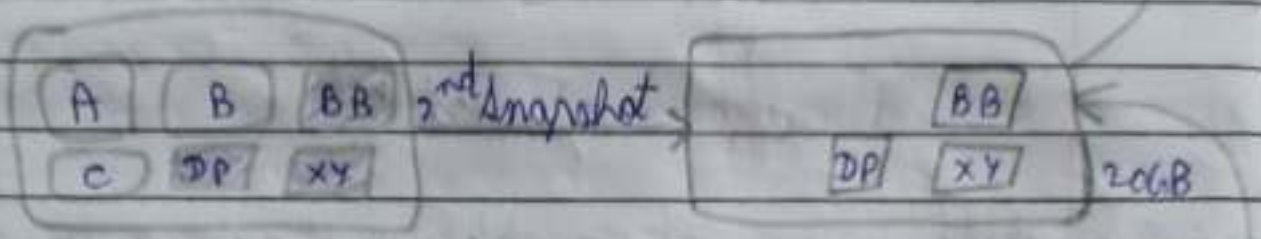


Volume with 6 blocks.

Date: _____ Page No. _____

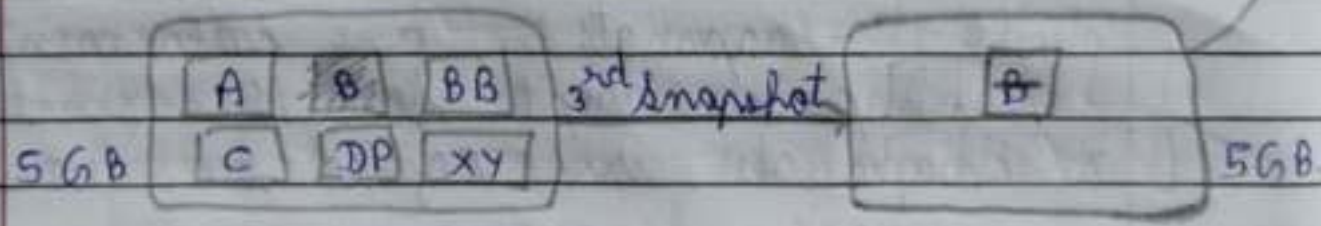
Topic: _____

- after some time data blocks gets modified.



only the modified blocks are copied of total (3) 20 GB size.

- then one block is deleted.



The 3rd snapshot will only contain the marker saying that 2nd Block (B) is deleted.

⇒ Snapshots copies only changed data, still they keep reference of the unchanged data as well.

Date: _____ Page No. _____

Topic: _____

- for unchanged data from 2nd snapshot it refers to snapshot 1.

- similarly snapshot 3 will refer to both snapshot 2 as well as to snapshot 1.

- means for data block A, B, C, B snapshot 2 will refer to snapshot 1, similarly for snapshot 3.

- if snapshot 2 is deleted, than entire snapshot 3 will refer to snapshot 1. → image of the modified data will merge with snapshot 3 & (5+20 GB).

- the reason is snapshot 3 was taken after snapshot 2 so it should contain the state of data that was captured by snapshot 2 & for rest unchanged data it will refer to snapshot 1.

- if snapshot 1 is deleted.

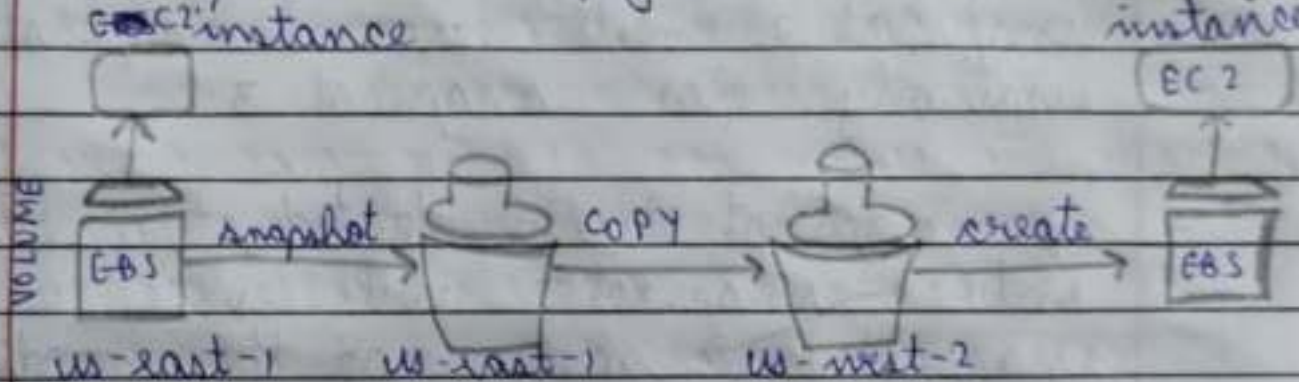
Date _____ Page No. _____

Topic: _____

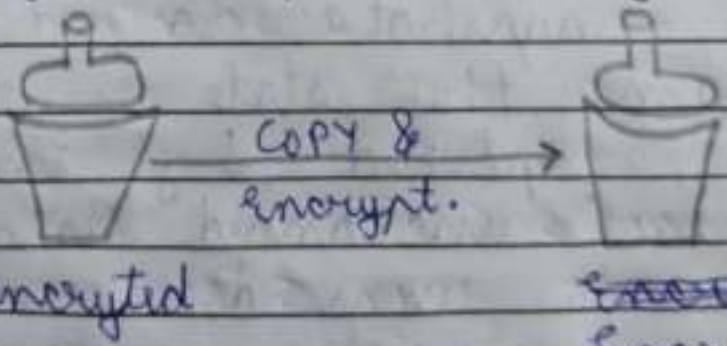
- all the data is merged with snapshot 3 & size becomes 95GB. (because 5GB is deleted).

// snapshot copy. //

> snapshot copy to a different Region instance.



> Encryption can be performed during snapshot copy.



Snapshot - demo

23/06/25

Date: _____ Page No. _____

3.5. Topic: copy operations.

- we have a EC2 instance & 2 Volumes are attached to it.
  /dev/xvda → root Volume.
  /dev/xvdb → our Volume.
  ↳ MOUNTED TO /home/ec2-user/ebs-demo

- copy a file from our Windows machine to this folder in Volume.

- open Windows & type this command :-
  ~~ssh~~ -i "path of private key-pair.pem"
  scp "file to copy (path of this)"
  ec2-user@publicIPv4 : path where to copy in EC2 instance.

- ~~ssh~~ -i "D:\Piyush\~~abc~~\key-pair.pem"
  scp "D:\Piyush\abc\abc.pdf"
  ec2-user@3.110.219.77:/home/ec2-user/ebs-demo

⇒ if permission denied :- Go To EC2 instance login & write :
⇒ sudo chmod 755 /home/ec2-user/ebs-demo
⇒ sudo chown ec2-user:ec2-user /home/ec2-user/ebs-demo
run this again & copied successfully.

Date: _____ Page No. _____

Topic: _____

→ create a snapshot from Volumes tab :-

under EBS section > snapshots > create snapshot >

select Volume

→ under EBS section > Volumes > actions tab > create snapshot > give description > create snapshot. Successfully created

→ we can check snapshot progress, status, other details :- pending / completed 99%/100%.
under EBS section > snapshots.

→ Now Copy this snapshot to a different region → create a Volume from it → and attach it to an EC2 instance.

⇒ Copy :- select snapshot
under EBS section > Snapshots > actions > copy snapshot > give description, give a destination region, encryption (if you want) > Copy snapshots. Success.

⇒ Change location/region & go ahead.

Date: _____ Page No. _____

Topic: _____

→ In London region.
→ EBS section > Snapshots > our snapshot is visible & is ready to use.

→ login in to our instance here if any, if not create EC2 instance & log in.

// Volume From Snapshot. //

→ EBS section > Snapshots > select snapshots > actions > create Volume > select Volume type, size (can only be increased, not decreased → reference to snapshot size), AZ (same as this Instance region) > create Volume. Success.

→ check Volume in ⇒ EBS > Volumes > our created Volume is there.

// Attach Volume to EC2 Instance. //

→ EBS section > Volumes > select Volume > actions > attach Volume > select Instance, select device name (/dev/sdb) > attach Volume. done.

→ check Volumes from AWS Console (Instances > Instance > select Instance > Storage > block devices (2 devices. 1 ROOT. 2 our Volume.)

Date: _____ Page No. _____

Topic: _____

→ check from CMD as lsblk -f :-

→ sudo file -s /dev/xvdb
No data, ext4 from start.
(because we have created this Volume from the snapshot.)

// MOUNTING //.

→ mkdir demo-ebs-snapshot
→ sudo mount /dev/xvdb /home/ec2-user/demo-ebs-snapshot/
/dev/xvdb → where to MOUNT.
/home/ec2-.... → what to MOUNT.

→ lsblk -f
/dev/xvdb 5G disk /home/ec2..

OR

→ df -h
/dev/xvdb 4.9G 2.1M 4.6G 1% /home/
Filesystem Size used avail use% Mounted on.

→ let us see the data we had (pdf file) is there in this device (/dev/xvdb).

Encryption, Bidding, EBS optimization, design pattern (RA, IO, DP).

22/6/215

3.7 Topic

→ cd demo - ls - snapshot
→ ls -l
. . . . . lost + found
. . . . . pdf - file . pdf

→ we got entire data from one Volume in a region from snapshot copy to Volume in another region.

→ snapshot can be used to restore Volumes in same or different region.

↳ In our case :- MUMBAI (ap-south-1) to LONDON (eu-west-2).

// ENCRYPTION //.

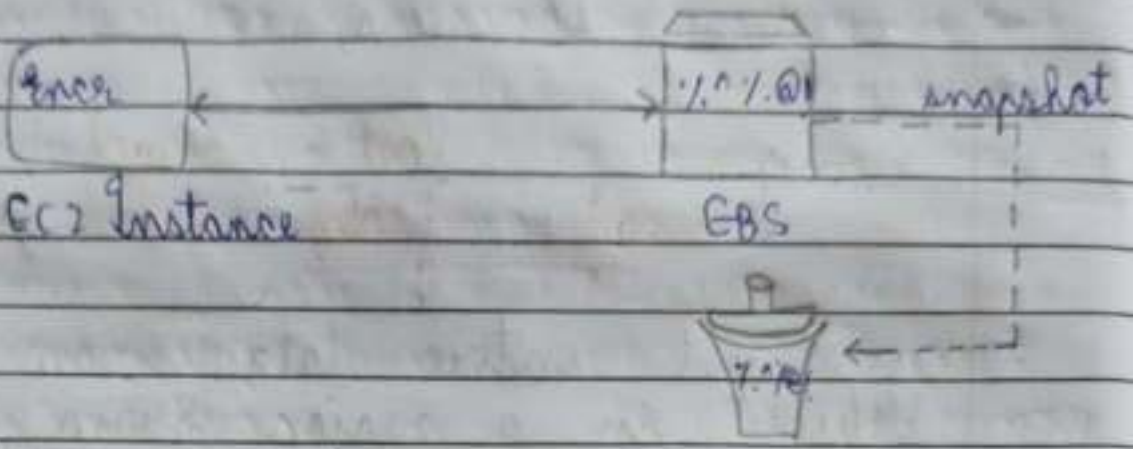
→ EBS Encryption can be used to encrypt data inside of EBS Volumes.

on

> Supported by all Volume types, but not by all Instance types.

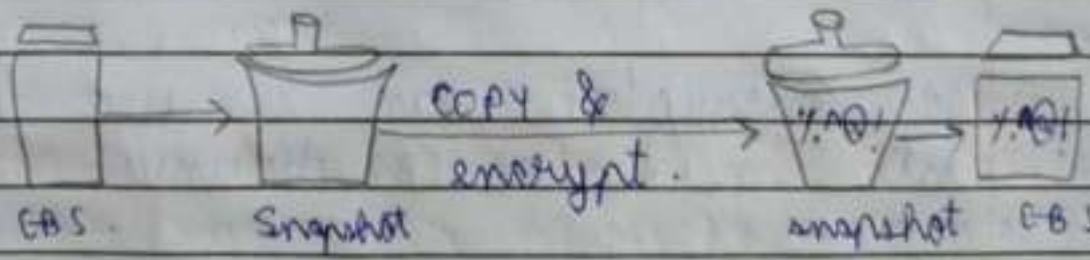
Date _____ Page No. _____

Topic: _____



- Entire Encryption process occurs in this EC2 Instance.
- So, the encrypted data is written to the EBS Volume.
- creating a snapshot from this Volume will create of encrypted data.

> Unencrypted to Encrypted.

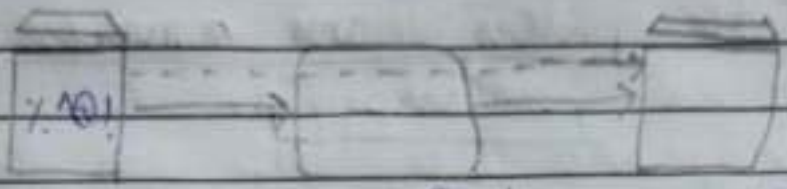


- from unencrypted EBS we created a snapshot (it will also be unencrypted)
- while copying snapshots we encrypt data & the created EBSVolume (it will contain encrypted data).

Date: _____ Page No. _____

Topic: _____

> Encrypted to Unencrypted Unencrypted.



EBS EC2 Instance EBS

- Encrypted EBS Volume is attached to an EC2 Instance.
- created a Unencrypted Volume an attach it to an EC2 instance as well
- Then filesystem from EC2 Instance can be used to copy data from Encrypted Volume to Unencrypted Volume.
- all data is copied, then we will have same data in both the Volumes.
- Then, the encrypted one can be detached & deleted
- converted unencrypted Volume to unencrypted.
- Because there is no direct way of doing it.

Date _____ Page No. _____

Topic _____

(MUMBAI) → ap-south-1

// EBS Pricing (for ~~us-east-1~~ region) //

⇒ refer to https://aws.amazon.com/ebs/pricing/

⇒

Volume Type	Price.
gp3 - Storage	$0.0912 / GB-MONTH
gp3 - IOPS	3000 IOPS free & $0.0057 / provisioned IOPS-MONTH over 3000.
gp3 - Throughput	125 MB/s free & $0.046 / provisioned MB/s - MONTH over 125
gp2 - Volumes	$0.114 / GB-MONTH of provisioned Storage.
io2 - Storage	$0.131 / GB - MONTH.
io1 - Volumes	$0.131 / GB-MONTH & $0.063 / provisioned IOPS-MONTH.
st1 - Volumes	$0.051 / GB - MONTH.
sc1 - Volumes	$0.0174 / GB - MONTH.

⇒ there are many more of them as well as of snapshot pricing, Have a look on the above website/link

Date: _____ Page No. _____

Topic: _____

Service-level-agreement-SLA

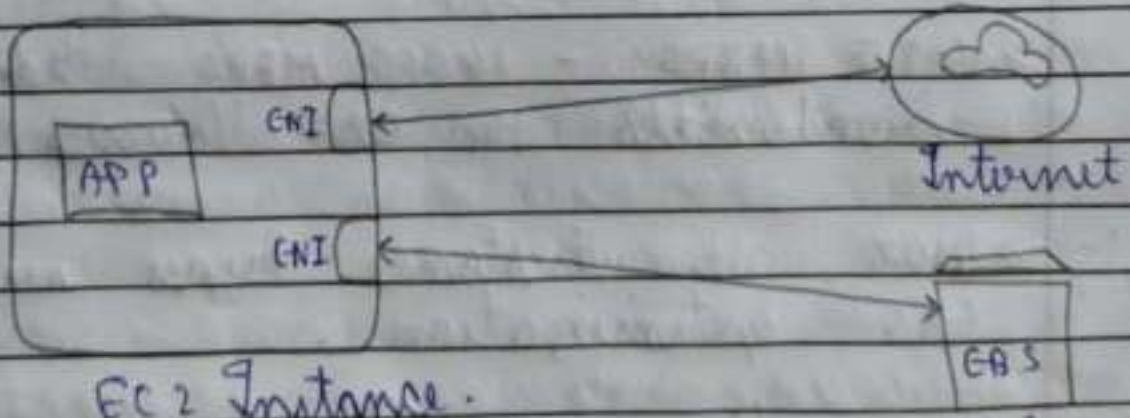
AWS also provides a Uptime of 99.99% (11 nines to be precise).

Free tier :- 30 GB/ MONTH, combination of gp2 & magnetic.
20,00,000 IO with magnetic.
1GB of snapshot storage.

// EBS Optimization //

- How to make sure of IOPS, & throughput are very high.
- How to optimize our EBS Instance.

ENI → Elastic Network Interface.



- our app (application) running inside EC2 Instance needs to have access to the internet & EBS Volume.

Date _____ Page No. _____

Topic _____

- without optimization we have only one ENI.

- with / for optimization we can have multiple ENI for different services (Internet, EBS Volume).

- all the ~~other~~ communication is carried out through these network interfaces.

- one ENI can not be used for any other purpose if it is been ~~using~~ used for one purpose.

> 425 MBPS - 14000 MBPS dedicated bandwidth to EBS Volumes.

NOTE :- > NOT all Instance types supports EBS optimization.

// design Patterns - RAID //

- Redundent array of Independent disks
- provides high availability, high throughput performance through multiple Volumes / disks.

Date: _____ Page No: _____

Topic: _____

onto multiple devices.

⇒ RAID 0 (Striping) :- Stripes data in

BLOCK 1
BLOCK 2
:: 3
:: 4
:: 5
:: 6
:: 7
:: 8

Size = 100 GB Size = 100GB

Throughput / speed.

speed = 10 MB/s. speed = 10 MB/s.

total size = 200 GB.
Usable size = 200 GB.
speed = 20 MB/s (∵ 2 disks).

can use more disks we have used 2 disks.

- data is striped into 2 disks.
- usecase :- when we don't have high speed disks, but our workload demands it.
- We can use multiple EBS Volumes, to create a ~~a~~ RAID 0 ~~array~~ array.

Date: _____ Page No. _____

Topic: _____

- used in cases :-
> suppose there is a budget constraint & only GP2 Volumes can be used.

> when we need parallel IO processing for our application of databases.

- used for higher performance (speed),
- & IO parallelism

⇒ // RAID-1 :- Mirroring //.

- we will consider the same example :-

- In RAID-1 each block is stored twice (once in each disk).
- that provides high availability &
- fault tolerance.

- if disk 1 goes down, NO WORRIES we have disk 2 available & the communication will be done with disk 2 & vice versa.

Date: _____ Page No. _____

Topic: _____

- No application will have the direct access to these disks, they will come via RAID 1 setup.

total : block (1-3).

total size (200 GB).
Usable size (100 GB).
Speed (10 MB/s).

size = 100 GB , 100 GB.
speed = 10 MB/s, 10 MB/s

- This kind of setup in EBS is ~~uncertain~~ unnecessary.
- Because EBS disks are replicated within the same AZ.

NOTE:- Throughput will be as fast as the slowest disk.

- Ex:- RAID - 0 ⇒ 5 MB/s & 10MB/s ⇒ 15 MB/s. (total)
RAID - 1 ⇒ 5 MB/s & 10MB/s ⇒ 5 MB/s.
∴ slowest disk is of 5 MB/s.

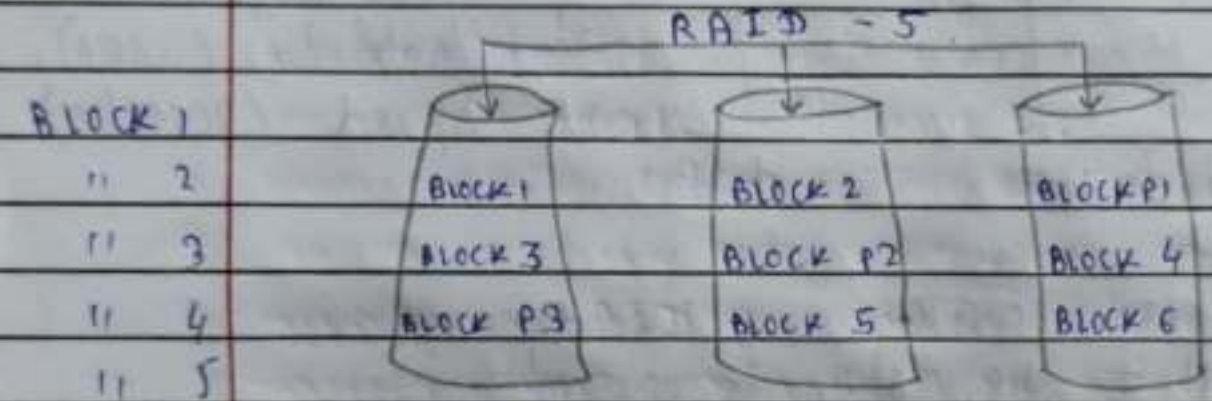
Date: _____ Page No. _____

Topic: _____

|| RAID - 5 (Parity) ||

- Parity provides fault tolerance & helps in reconstructing the data after a disk failure.

- it uses Striping & Parity Both.



size →	100 GB	100 GB	100 GB
speed →	10 MB/s	10 MB/s	10 MB/s

total size ⇒ 300 GB.
Usable size ⇒ 200 GB.
Speed ⇒ 20 MB/s.

→ RAID - 5 is totally unnecessary in GBS.

Date: _____ Page No. _____

Topic: _____

→ similar to RAID - 0.
- block 1 & 2 is stored in disk 1 & 2.
- block P1 (Parity block) is stored in disk 3.
- Parity block (P1) is a logical combination of block 1 & 2. configuration
- This setup of RAID - 5 ~~has~~ can handle a failure of an entire disk.
- Ex :- if disk 1 fails.
Then ~~block 1~~ entire data can be reconstructed.
block 1 from parity block (P1) from disk 3.
block 3 from block of parity (P2) & block 4 from disk 2 & disk 3.
block 5 & 6 are already available.

NOTE :- RAID 5 can never be configured as EBS Volume.
because this parity blocks (P1, P2, P3) these will take additional IOPS or throughput

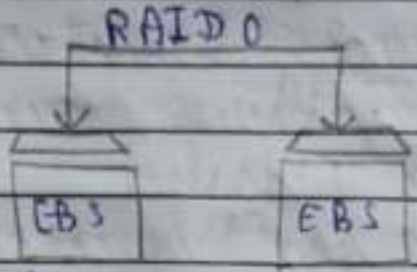
- If a provisioned say 120 IOPS & we have a RAID-5 setup & expecting 120 IOPS but we will only get 80 IOPS & 40 IOPS will be taken by these parities blocks (P1, P2, P3).

Date: _____ Page No. _____

Topic: _____

RAID-0:

- RAID using EBS, only RAID 0 is to be configured to get parallel IOPS processing & high throughput.



Volume 1. Volume 2.
2500 IOPS 2500 IOPS.

- using RAID 0 with these Volumes we can get a combined IOPS of 5000.

RAID-1:

- RAID-1 :- should not be configured as it only does replica.

RAID-5:

- RAID-5 :- should never be configured as parity blocks takes up additional IOPS.

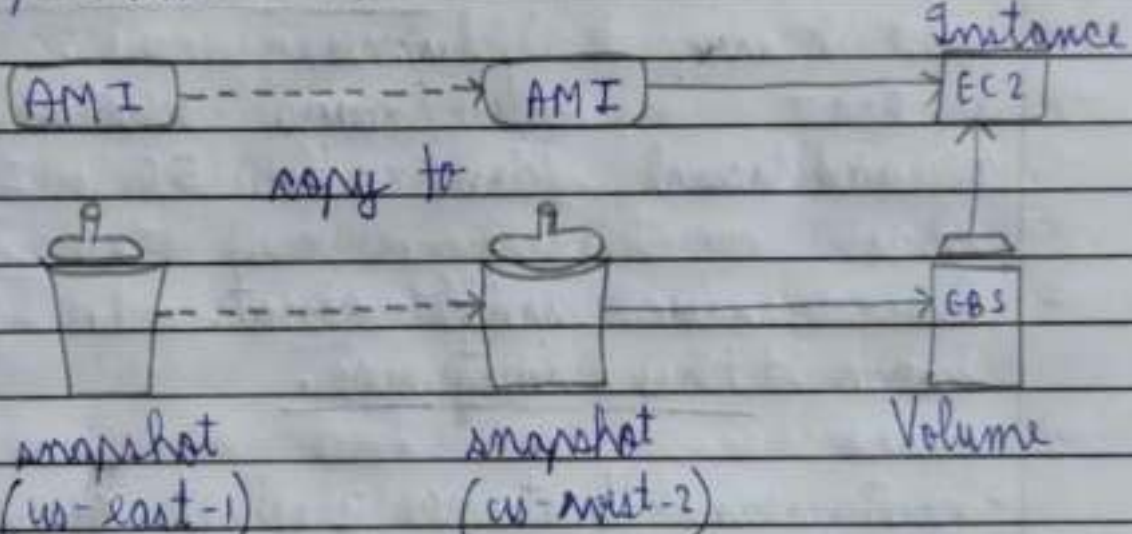
Date: _____ Page No. _____

Topic: _____

22/Oct/25.

DA - Disaster Recovery & Increase SLA.

- How do you plan DA using EBS & EC2?
- AMI & snapshots can be used to provide DR scenerios.



- we can attach this Volume to the EC2 instance. after this;
- we will get the exact same application & its corresponding data which was running in us-east-1 region.
- for automation of this process, few steps have to be written, to take snapshots & AMI at regular intervals in us-east-1 region, & copy them to us-west-2, launch Volume & Instance & attach them.

Date: _____ Page No. _____

Topic: _____

- A point to note that here basically we only have our root Volume with the EC2 instance until & unless attached one (additional Volume).

- EC2 Instance → aws-ebs-instance-demo.
  we have 2 Volumes -
  1 Root /dev/xvda 8G
  2 additional /dev/xvdb 5G

- Now, we will create an AMI from this instance.

- Instances > Instance > select our instance > actions tab > Image & templates > create Image > give image name, give image description, reboot instance, Instance Volumes > create image. done.
  Currently creating AMI from Instance.

- Status (Pending & wait for available)

Date: _____ Page No. _____

Topic: _____

- Checking goto :- Images section > AMIs > our AMI will be visible here.
- this AMI itself contain 2 block devices (root & additional), check in its details by selecting that AMI.

⇒ Now, launch EC2 instance from this AMI.

Images section > AMIs > ~~launch~~ > select our AMI > click launch instance from AMI > give name of instance, select AMI, select instance type, select / create key-pair (login), select / create Network settings, configure storage > launch instance.

⇒ > from-ami > already there > t2.micro > key-pair .pem > select existing security group (launch-wizard-1) > Storages (2 :1 is root, 2 is additional now built-in from AMI) > launch instance.

⇒ success

⇒ check in all instances (Instances section > instances > visible here).

⇒ login & check states of devices.

Date: _____ Page No. _____

Topic: _____

⇒ ssh -i "key-pair.pem" ec2-user@13.127.160.18

type: yes

check: ⇒ lsblk ↑ MAPPED TO

/dev/xvdb 5G disk

⇒ NOT mapped to anywhere.

⇒ but remember this instance is launched from our AMI which was created from our other instance.

⇒ the original instance had filesystem in it. (ebs-demo2).

⇒ check if we had it in our new instance by "ls" Command:-

⇒ ls

ebs-demo ebs-demo2

⇒ MAP it using "mount" Command:-

sudo mount /dev/xvdb /home/ec2-user/ebs-demo2

⇒ go to this path using ("cd") Command:

cd ebs-demo2

⇒ ls # look for files & folders.

----- file-pdf.pdf

d----- lost+found

Date: _____ Page No: _____

Topic: _____

- Even AMI can include additional Volumes in an image (AMI), then why we should take our EBS Volumes snapshots separately.

- This is were incremental ~~model~~ snapshot plays an important role.

- EBS snapshots are incremental in nature except for the first one.

- So, if we change or modify data(add, remove, etc) than the newest snapshots will only contain the modified / changed data.

- Whereas the AMI ; ~~is~~ ~~the~~ are the images of the full copies of the entire Volumes attached to the instances.

- if we have very large amount of data, it is better to take EBS snapshots separately for faster backups.

23/06/25.

Pre-EFS Shared file system, EFS, Mount target, Mount a file system (1). Storage EFS ⇒ Elastic file system.

Date: _____ Page No. _____

Topic: _____

3.8.

AGENDA.

1 - Pre-EFS : Shared file System.

2 - Elastic File Store.

3 - Mount targets.

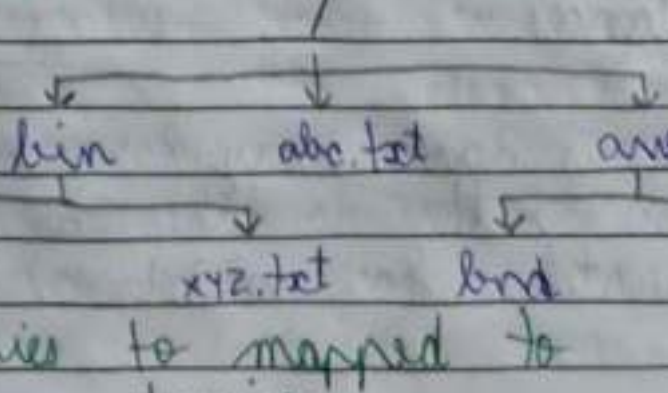
4 - Mounting file systems.

5 - Demo.

6 - Pricing.

7 - Design Patterns.

⇒ TOPIC-1. File ~~basic~~ SYSTEM BASIC. already covered ⇒ just a recap.



⇒ directories to mapped to storage devices.

Date _____ Page No. _____

Topic _____

directories :- bin, aws, abc, bnd

files :- abc.txt, xyz.txt, foundation.txt.

storage devices :- S1(/), S2(abc), S3(bnd).

⇒ these directories are called file systems

S1 ⇒ /

S2 ⇒ /bin /abc

S3 ⇒ /~~to~~ aws/ bnd

⇒ all these file systems as storage ~~are~~ devices which are basically the block ~~of storage~~ devices.

term of

// shared file system //

- shared file system is different from block devices.
- the protocols are different too.
- ext2, ext3, ext4, ... do not work with shared file system.

Date: _____ Page No. _____

Topic: _____

> file system shared across multiple servers.
> exposed as file systems to servers
> uses NFS protocol (NETWORK FILE SYSTEM)
> NFS client.
⇒ to access shared file system we need NFS client.
⇒ each servers need to have installed NFS client installed in them to access the shared file system.

Date: _____ Page No. _____

Topic: _____

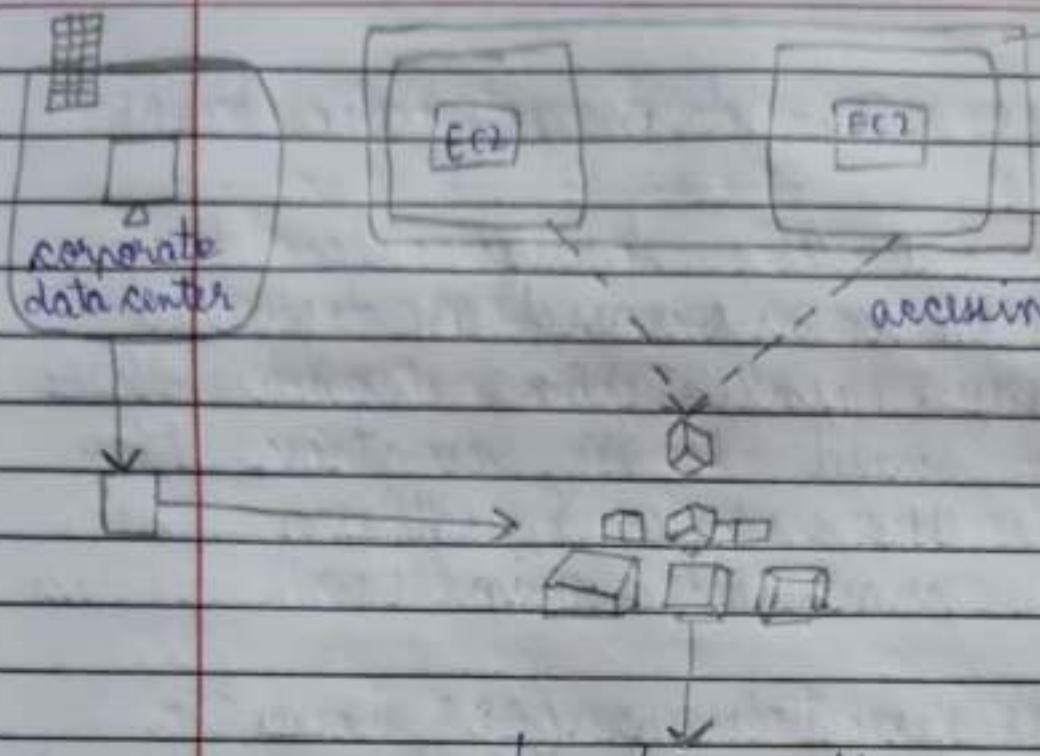
⇒ TOPIC - 2. ELASTIC FILE STORE.

- Elastic because storage for the file system is increased & decreased dynamically. (unlike EBS where ~~there was~~ it ~~is~~ need to ~~be~~ mention the size of the storage), nothing is to be provisioned in EFS.

> uses file level storage. (EBS uses a block level storage).
> can only be used with EC2 instances. (like EBS).
> need to install NFS client on each servers. & uses NFS version 4.0 & 4.1
> no limit of no. of EC2 instances which can access this shared file system.
> concurrent access from multiple EC2 instances.
> EFS file systems can also be ~~access~~ accessed from on-premise servers only using Direct Connect.

Date: _____ Page No. _____

Topic: _____



⇒ if one EC2 writes anything it will be visible to all the other EC2 instances.

⇒ direct connect on-premise. This can be really helpful if we already have a shared file system on on-premise environment & want to extend that particular file system & make use of aws elasticity & high availability.

Date: _____ Page No. _____

Topic: _____

⇒ it can also be useful when we want to create a new file system, but do not want the overhead of storage & management.

⇒ we can opt for EFS & use it exactly the same way as we would have use on-premise NFS file system.

⇒ Note that this can only be achieved by direct connection it cannot be used over the internet.

> size can grow to petabyte scale.

> Data is stored in multiple AZs in the same region for high availability.

⇒ High availability by default within the same region without doing anything.

↳ is a feature of EFS where in data & EFS is stored in multiple AZ's similar to S3.

Date: _____ Page No. _____

Topic: _____

⇒ TOPIC -3 ⇒ MOUNT TARGETS :-

- to access a EFS file system in a VPC, one or more mount targets have to be created in the VPC.
- Concept of VPC, subnet, etc. remains the same.

> Create file System (/aws/Foundation)
> Then, we create Mount targets.

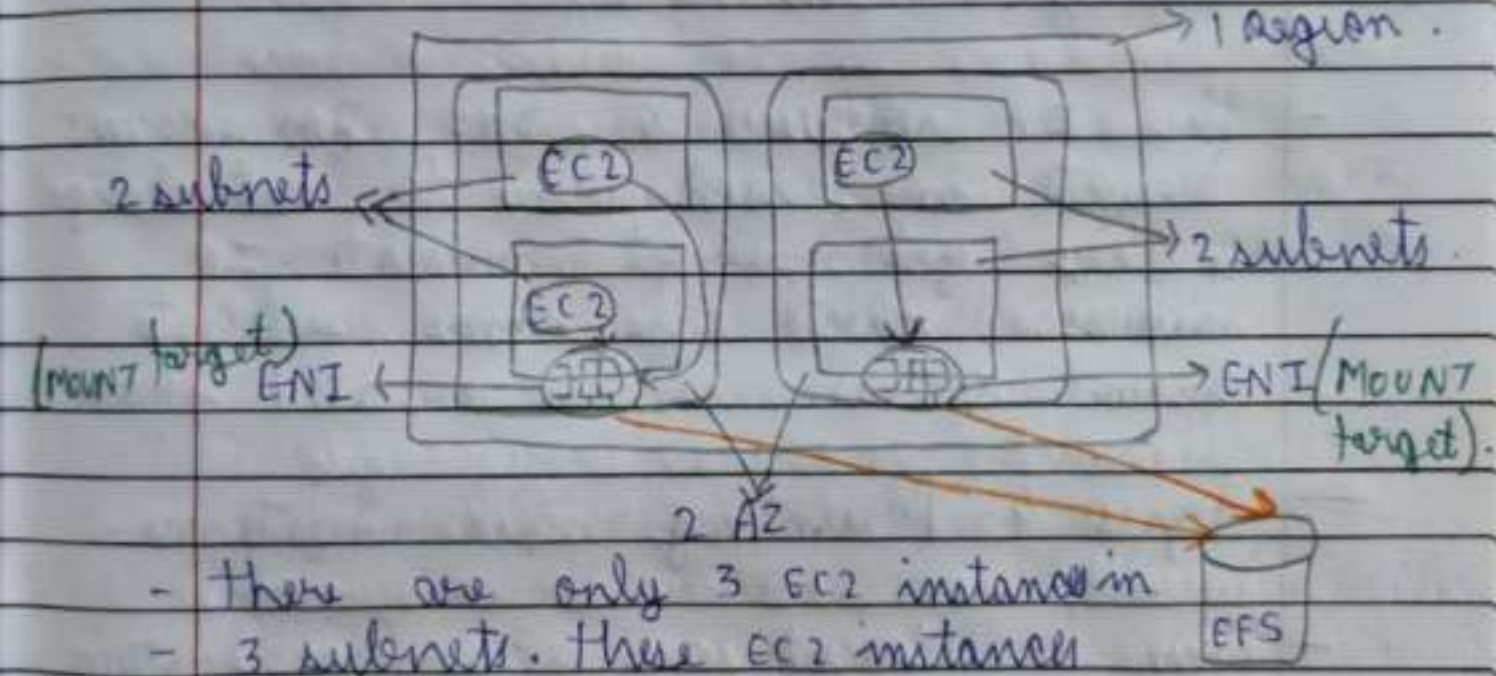
What are Mount targets.

- Mount target provides IP address or DNS name for an NFS V4 end point.
- On which EFS file system can be Mounted.
- file system in the EC2 instance can be mounted using a DNS name, which will resolve to the IP address of the EFS Mount target in the same AZ, as that of the EC2 instance.
- one mount target can be created in each AZ, in a Region.

Date: _____ Page No: _____

Topic: _____

- If there are multiple subnets in a single AZ, creating a mount target in one of the subnet would satisfy.
- All EC2 instances in this AZ share the same mount target.



- There are only 3 EC2 instances in 3 subnets. These EC2 instances want to access the EFS.
- create MOUNT targets in one of the subnets in each AZ.
- mount targets are nothing but ENI.
- All EC2 instances go via these ENI's to access the EFS

Date: _____ Page No. _____

Topic: _____

> Security groups for mount targets - port 2049 (used for EFS) remains same & NACLs
> concept of NACLs : can be used for subnets as usual.

- Mount targets are design to be highly available.

- The IP addresses & DNS for MOUNT targets in each AZ do not change & they get ~~privet~~ private IP addresses from the same subnets from where it is created from

⇒ Topic 4 - Mounting File Systems.

- How to Mount an EFS file system in an EC2 instance Conceptually.

> EFS supports NFSv4.0 & NFSv4.1, Most preferably ⟷

- As a Solution architect or a SysOps administra we wont be running these Commands for. So, Chill & DON'T WORRY! ☺

- Steps to ~~~~ access the EFS file system from EC2 instance.

STEP 1:- Install NFS client in EC2 instance.
Command ⇒ yum install -y nfs-utils (RHEL, Amzn).
⇒ apt-get -y install nfs-common (~~~~ Ubuntu).

STEP 2:- File System has to be Mounted with EFS using any 1 of these options:-
> File System DNS Name.
> Mount target DNS Name.
> Mount target's IP address - different for different AZ.

Command ⇒ mount -t nfs -o nfsvers=4.1, rsize=1048576, wsize=1048576, hard, timeo=600, retrans=2 <dns-name or IP-address>:/<MOUNT POINT>

Date: _____ Page No. _____

Topic: _____

STEP-3 /etc/fstab

- an entry has to be made in /etc/fstab.
- So, that the file system gets Mounted whenever the EC2 instance start.
- Command:-

⇒ <dns-name or IP-address>:/ <mount point> nfs4 nfsvers=4.1, size=..., wsize=..., hard, timeo=..., retrans=2, _netdev 0 0

⇒ Read & Write size (IO) = Highest possible
Mount Option = hard / soft
timeout = at 150 (15 seconds).
⇒ These mount options should be used for better performance.

EFS demo

23/06/25.

Date: _____ Page No. _____

39

Topic: _____

- In demo :-
Virtual private cloud.
> VPC ⇒ aws-foundation-vpc (in ap-south-1)
> Create 5 subnets in 3 different AZs (1a, 1b, 1c) - subnet1a-1, subnet1a-2, subnet1b-1, subnet1b-2, subnet1c-1. All should be private subnets.
> Launch instance insta in subnet1a-1, instb in subnet1b-1 & instc in subnet1c-1.
> Create an EFS file system.
> Create 3 mount targets in 3 different AZs.
> Control access using both security groups & NACLs.
> Mount this file system using file system DNS name in insta, Mount target DNS name in instb & Mount target IP address in instc.
> Modify something ~~from~~ in the file ~~of~~ from insta & check whether the modification is visible to from other instances.

Date: _____ Page No. _____

Topic: _____

- Go to ~~AWS MV~~ AWS VPC.

under VPC section > Subnets > create subnet > select a VPC ID (default) Subnet settings (subnet name, AZ,) > ~~create subnet~~ IPv4 subnet CIDR block > create subnet.

✓ ⇒ subnet 1a-1, ap-south-1a, 172.31.48.0/20
✓ ⇒ subnet 1a-2, ap-south-1a, 172.31.64.0/20
✓ ⇒ subnet 1b-1, ap-south-1b, 172.31.80.0/20
✓ ⇒ subnet 1b-2, ap-south-1b, 172.31.96.0/20
✓ ⇒ subnet 1c-1, ap-south-1c, 172.31.112.0/20

- Go to AWS Console.
- create 3 EC2 instances.

under instances section > instances > launch instances > give name, AMI, instance type, key-pair (login), network security (security group), storage > launch instance.

Date: _____ Page No. _____

Topic: _____

- all the three instances have
same AMI (Amazon linux),
same storage (default),
same key-pair (foundation.pem),
same instance type (t2.micro),
name (instance 1a-1, 1b-1, 1c-1),
security groups (existing launch-
wizard-1), & under network &
~~security~~ settings > advanced (edit) > VPC (default),
subnet (subnet 1a-1, subnet 1b-1, subnet 1c-1),
> launch instances all three.

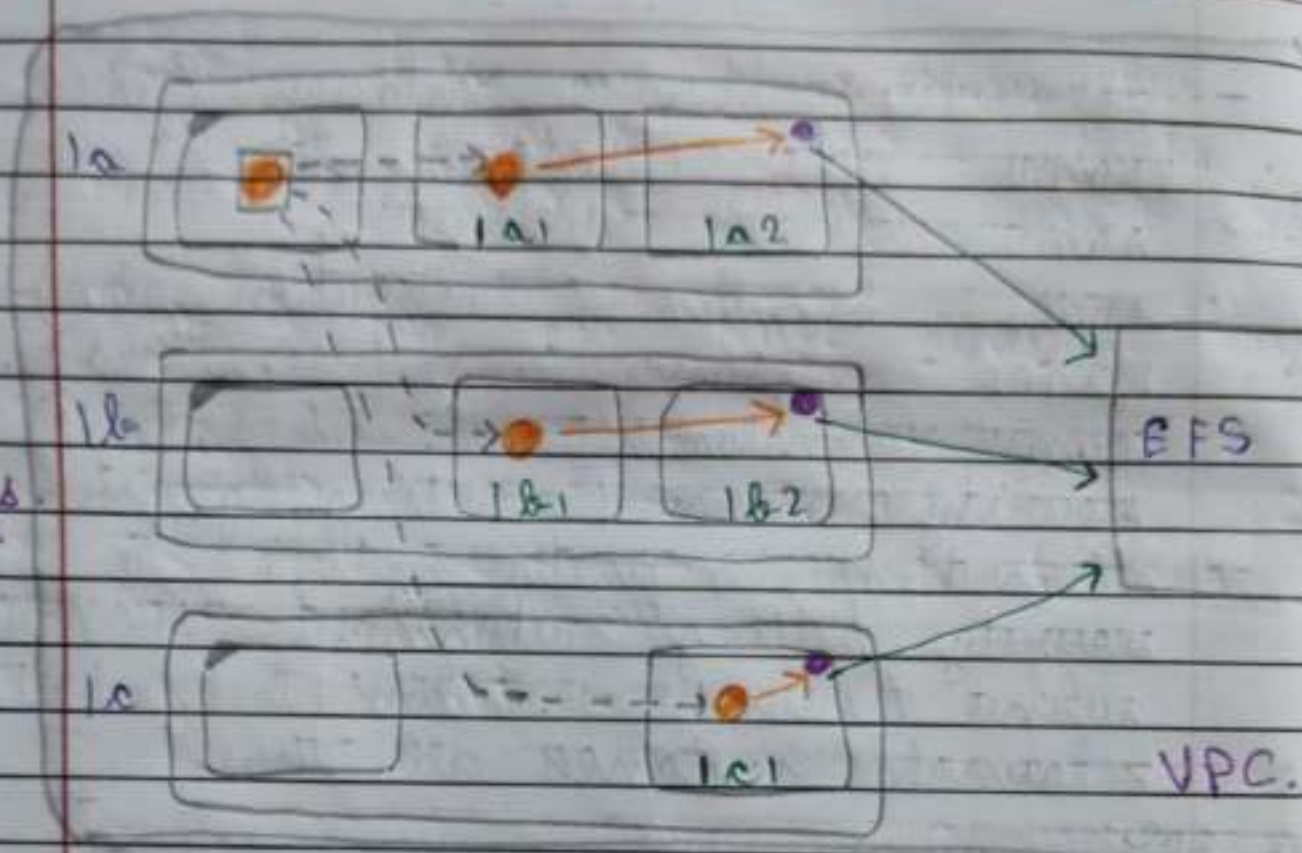
instance:
one instance-admin, AMI-Amazon linux,
t2.micro, key-pair.pem, subnet 1a-1 &
launch-wizard-1 in network settings,
default storage > launch instance.

- created a instance-admin this is
needed to login to all the ~~an~~ other
EC2 instances which we have created.

⇒ Architecture of this demo is as follows:-

Date: _____ Page No. _____

Topic: _____



3 AZ's.

3 AZ ⇒ ap-south-1a / 1b / 1c.

5 private subnets ⇒ 1a1, 1a2, 1b1, 1b2, 1c1.

3 public subnets ⇒ 1a, 1b, 1c (first box)

4 Instances, Instance - admin, used to connect all other instances. (ssh into all the other instances).

EFS in the same VPC.

Mount targets creation in (1a2, 1b2, 1c1).

⇒ No need to necessarily create Mount target in the subnet of EC2 instance resides.

Date: _____ Page No. _____

Topic: _____

- as we are good as well as we create our Mount targets in the same AZ of the EC2 instances.

- EC2 instances uses these Mount targets to connect to the EFS file System (where we store our files).

⇒ to create a file system. (efs)
- go to aws.amazon.com/efs/.....
- click create file system > give name (efs-file-system-demo), select VPC (default) > customize > step 1 [leave as default] > next > step 2 (select Mount targets (3)), ap-south-1a ⇒ subnet 1a-2 ⇒ launch-wizard-1, " - " -1b ⇒ " 1b-2 ⇒ " - " - ", " - " -1c ⇒ " 1c-1 ⇒ " - " - " > next > step 3 (file system policy - Optional) > next > step 4 (Review & Create), check all the details > create. Success!

⇒ Elastic file System > file systems > select file system, click on view details, > General → Edit, or Network tab > Manage.

⇒ changing Mount target will be done here (adding / Removing) as needed.

3.9. → 6:38.

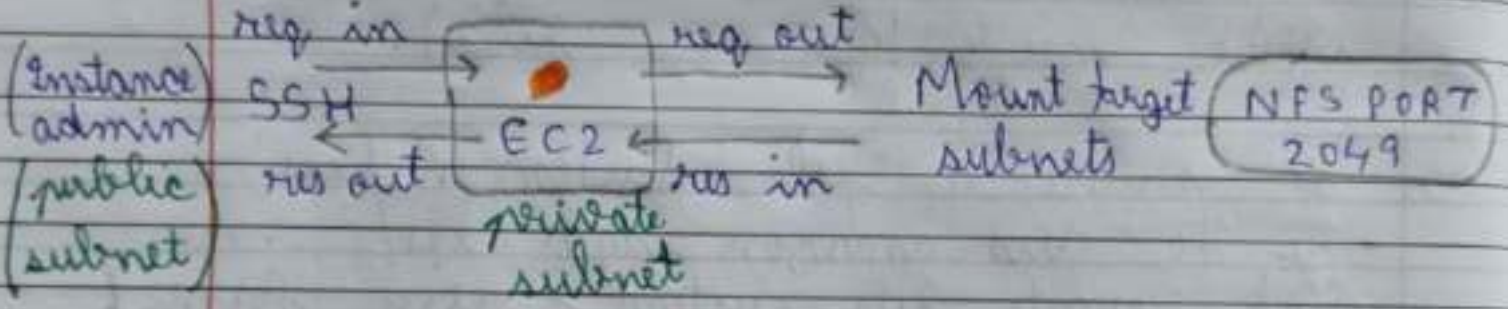
24/oct/25

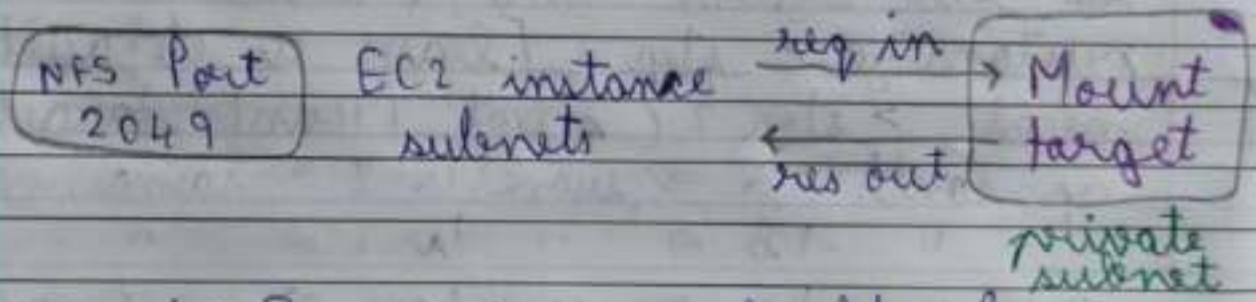
Date: _____ Page No. _____

Topic: _____

- Next Step is to modify our security group & Network ACLs.
NACL → Network Access Control lists.

- Req & Res for NACL firewall loops.





⇒ req in & res in should be in in-bound rules of EC2 private subnets & res out & req out in out-bound rules.

// creating Network ACL //

24/01/25.

Date: _____ Page No. _____

Topic: _____

- Creating NACL :-

- subnet Associated : 1a1, 1b1, 1c1.
- Inbound Rules ~~for~~ :-

	Rule no	type.	Portrange.	Source.	ALLOW/ DENY
	50	SSH (22)	22		ALLOW
1a-2	300	Custom TCP	100-65535	172.31.64.0/20	ALLOW
1b-2	400	Custom TCP	100-65535	172.31.96.0/20	ALLOW

- Outbound rules :-

	Rule no.	type	Portrange	Source	ALLOW/ DENY.
	100	Custom TCP	1024-65535		ALLOW
1a-2	300	NFS(2049)	2049	172.31.64.0/20	ALLOW
1b-2	400	NFS(2049)	2049	172.31.96.0/20	ALLOW

⇒ subnet of instance - admin.

VPC console > under security > Network ACLs > create NACL > give name & VPC ID > create NACL. Success

> select these NACL > actions > edit subnet associations > select subnet 1a-1, subnet 1b-1, subnet 1c-1 > save changes. Success.

Date: _____ Page No. _____

Topic: _____

> select NACL > actions tab > edit inbound rules > add > give rule number, type (ssh(22)), source, Allow/Deny.

- in the same way add 2 more. outbound & 2 more rules of outbound.

- create in-bound & out-bound rules as shown in the first page

> in this same way create 1 more NACL for private-NACL-MOUNT-targets

> inbound rules :-

1a-1	100	NFS (2049)	~~2049~~ 2049	172.31.48.0/20	ALLOW
1c-1	200	NFS (2049)	2049	172.31.112.0/20	ALLOW
1b-1	300	NFS(2049)	2049	172.31.80.0/20	ALLOW

Rule no.	type	Port range	Source	ALLOW/DENY

> outbound rules :-

1a-1	100	Custom TCP	100-65535	172.31.48.0/20	ALLOW
1c-1	200	Custom TCP	100-65535	172.31.112.0/20	ALLOW
1b-1	300	Custom TCP	100-65535	172.31.80.0/20	ALLOW

> subnet associated are :- 1a2, 1b-2 (subnet 1a-2, 1b-2)

subnet 1a-2 & subnet 1b-2

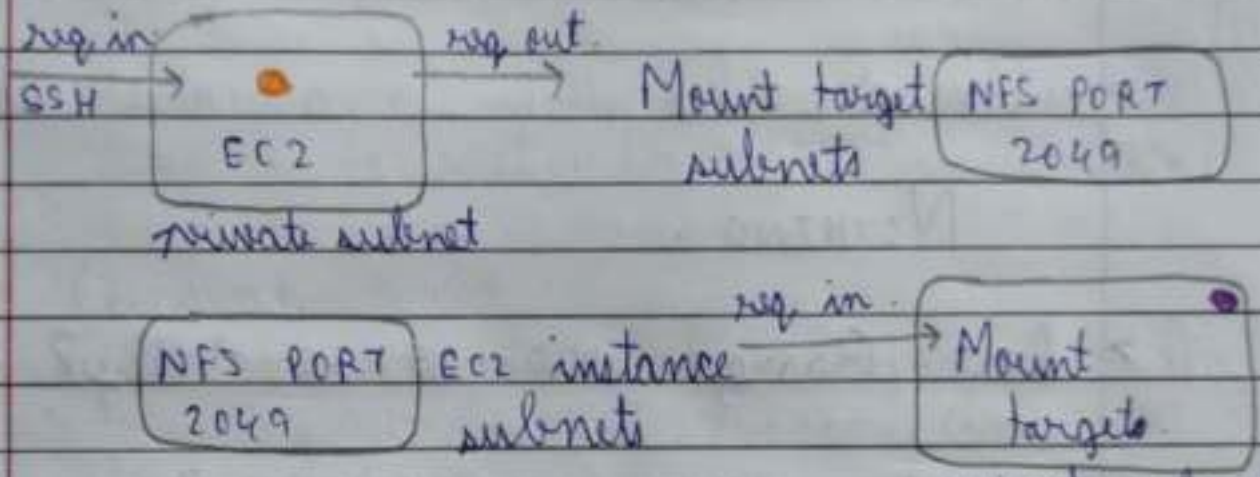
> add all this in Mount targets private NACL.

Date: _____ Page No. _____

Topic: _____

// Security Group :||

- It is simple as we only need to bother about requests (in & out).



VPC console

⇒ creating Security Group (SG) :-

- security groups for EN I's :-

> Security section > Security groups > create SG > give SG name, description, VPC ID, inbound rules, outbound rules > create SG. Success.

Date _____ Page No _____

Topic _____

> Inbound rules -

	Type	Source
1a-1	NFS (2049)	172.31.48.0/20
1c-1	NFS (2049)	172.31.112.0/20
1b-1	NFS (2049)	172.31.80.0/20

> outbound rules :-

NOTHING.

> Port Range 100 - 65535 Why?

- Y not from 1024 - 65535 ?
because of NFS reasons.
because NFS uses 111 or 112 for actual responses.

- Response for SSH still uses 1024-65535.

> now change SG for EFS file system.
change "launch-wizard-1" to newly above created security group (efs-security-group)

Date: _____ Page No. _____

Topic: _____

EFS

> EFS > file systems > select our efs file system > view details > network tab > manage > change SG (or add / delete mount targets) from here.

⇒ now mount this EFS file system to all the EC2 instances.

⇒ Commands :-

- sudo yum install -y nfs-utils
  # for Amazon linux, Red Hat enterprise Linux, or SUSE linux.
- sudo mkdir efs (directory on our EC2 instance).
- Mount your file system using the DNS Name.

sudo mount -t nfs4 -o nfsvers=4.1, rsize=1048576, wsize=1048576, hard, timeo=600, retrans=2, <DNS-NAME>:/ <file system (directory)>

Date: _____ Page No. _____

Topic: _____

> in a private 1a-1 EC2 instance
- install NFS client
- to do this first login to 1a-1 instance using a public key.
- NOT using a public key means because ~~yo~~ it was not assigned during launch.

- FIX :- Stop → Modify → Start (instance)

Stop → actions → Networking → Manage IP addresses → assign a public IP → start. (auto assign → On).

⇒ login success. (ssh -i "key.pem" ec2-user@3.110.181.38).
type this to install NFS client
sudo yum install -y nfs-utils
- We are using NAT gateway to connect to internet & install nfs-utils.

⇒ under VPC section > NAT gateways > create NAT gateway > give name, select subnet (1a-1), connectivity type (public), allocate Elastic IP > create NAT gateway. Success.

Date: _____ Page No. _____

Topic: _____

> under VPC section > Route tables > create route table > give name, select VPC > create route table.

edit this route table > add route > destination (0.0.0.0/0), ~~target (NAT gateway select NAT gateway IP)~~ > save change.
target (Internet gateway (igw-xxx xxxxxxxx)) > save change.

⇒ add 2 more outbound rules in private-NACL-aws-demo as :-

110	HTTP	80	0.0.0.0/0
111	HTTPS	443	0.0.0.0/0

& turn every inbound & outbound rules source to 0.0.0.0/0.

⇒ sudo yum install -y nfs-utils
done - success 64/MB/S 139MB. ✓
exit.

⇒ now in the Instance admin log in.
ssh -i "key.pem" ec2-user@13.201.57.101
success.

[copy private (key-pair.pem) in to instance-admin.
refer previous notes.]

Date: _____ Page No. _____

Topic: _____

powershell

- open ~~CMD~~ and run.

scp -i "PATH/TO/key.pem"
"PATH/TO/key.pem" ec2-user@
13.201.57.103:/home/ec2-user.

done.

- check in ec2 instance that is logged in using "ls" :-
yes visible key.pem.

- Now, log-in to private instance (1a-1) :-

ssh -i "key.pem ec2-user@<private IP>

23/06/25

→ first log in to instance-admin (with public IPv4).

→ copy key-pair pem file ~~&~~ here. (in instance admin)

→ now from instance-admin log in to instance 1a-1 using private IP of instance 1a-1.
success.

Date: _____ Page No. _____

Topic: _____

→ create directory ("mkdir efs-demo") :-

change its owner if root to ec2-user.

sudo chown -R ec2-user:ec2-user efs-demo/

→ mount this directory to our EFS.

sudo mount -t nfs -o nfsvers=4.1, rsize=1048576, wsize=1048576, hard, timeo=600, retrans=2 <EFS-DNS-NAME>:/ */home/ec2-user/efs-demo/ (our path)

→ "df -h" check from here mounted or NOT.

→ go in this directory ("cd efs-demo") :-
→ write some content in it:
echo "EFS demo file" > first_file
→ "more first_file" for checking its content.

⟹ we should be able to see all these content in other instances after mounting EFS to them.

exit. (from instance)

Date: _____ Page No. _____

Topic: _____

→ we are at instance-admin instance.

→ from here log in to instance 1b-1 refer the command used in instance 1a-1. (use private IP).

→ Mount EFS & here & this time use IP address instead of DNS NAME of EFS.

for

→ Mount target IP, we will use the IP address of the same AZ.

→ use IPv4 address of ap-south-1b get this from efs of AWS. EFS > file systems > select our EFS > network tab > IPv4 address.

→ create directory efs-demo-1 b-1

→ change the user from root to ec2-user (if root) refer the chown command used in previous page.

Date: _____ Page No. _____

Topic: _____

→ Mount this ~~for~~ directory to EFS.

same command but this time use <IPv4 address> from EFS.

.... retrans=2 10.20.5.71:/ /home....

→ check using "df -f" "df -h":-
Success.

→ check whats there in this file.

→ "cd efs-demo-1b1"

→ ls -l (we found our file "first-file".)

→ check its content same or NOT.

→ use "more "first-file" (same).

→ Modify a bit of its content.
cat >> first-file
modified by instance 1b-1
"CTRL +C".
exit (after exit we will be in instance-admin).

⟹ Now log in to instance1c-1 from instance-admin

→ use private IP of instance1c-1 to log in here.

Date _____ Page No. _____

Topic _____

→ login success.

→ create directory & change its owner (refer above pages).
efs-demo-1c)

→ Mount. (again use DNS name):-

... nvtvers=2 fs-abc05e78....:/ /home/....

→ df -h (to check) (DONE).
Mounted successful.

→ cd efs-demo-1c1

→ ls -l (first_file is there)

→ more first_file (the content is same as well as modified by the instance-1b1. (all Modifications are visible as well).

⇒ This is how we use ~~EC2~~ EFS ~~for~~ from multiple EC2 instances concurrently.

EFS file Syncing, Pricing.

20/06/19.

Date: _____ Page No. _____

3.10 Topic: _____

→ EFS file Sync is a tool used to file Sync files from an Existing EFS file system to an EFS

→ existing file system can be on-premises or an EBS block too.

→ It can be easily used to migrate file based on-premise applications to EC2

→ file Sync copies file data to EFS.

> Sync files are for existing on-premise / cloud file Systems to EFS.

> Software that can be installed on a server or as a VM (VMWare ESXi) or in an EC2 instance.

> Requirements - 4 Cores, 16 GB RAM, 80 GB disk Space.

Date _____ Page No. _____

Topic _____

> Steps :-

- Download & Deploy Sync Agent either on-premise as VM (VMWare ESXi) or an EC2 instance using Sync Agent AMI.

We get an option like this in EFS dashboard :-

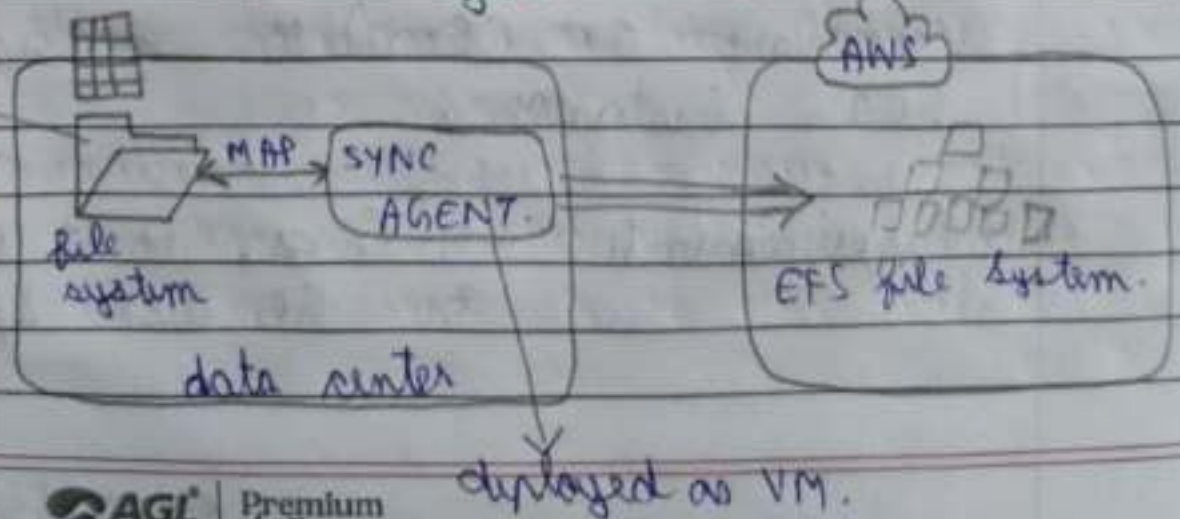
can download as VMWare ESXi or an AMI.

Select hot platform :-

◉ VMWare ESXi [DOWNLOAD IMAGE]

○ Amazon EC2

- Create Sync task to configure source & destination file systems.
- Start Sync task.



Date: _____ Page No: _____

Topic: _____

→ EFS File Sync (continuously copies data over to EFS.

// EFS pricing //.

visit :- https://aws.amazon.com/efs/pricing.

> No minimum provisioning.
> $0.30 per GB-Month.
Ex :- 100 GB then $30/month.

Ex :- 140 GB for 17 days &
300 GB for next 14 days of May.
140 * 0.3 * (17/31) +
300 * 0.3 * (14/31) =
$63.63.

No SLA provided Yet
By AWS

AWS tutorial for Beginners 8.

29/06/25

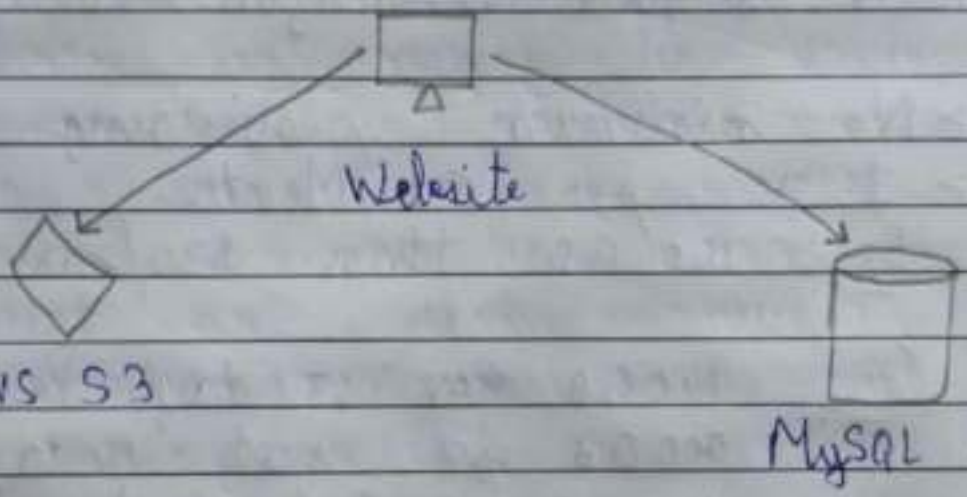
Date _____ Page No. _____

3.11.

Topic _____

> created a website using which we can upload our data on S3.

> architecture :-



> website can upload data on AWS S3 as well as on MySQL DB.

> website & MySQL DB both are on the localhost as of now.

> website is right now not able to connect AWS S3 because it is not able to authenticate.

Date: _____ Page No: _____

Topic: _____

→ Step 1 :- authenticate our Website to AWS S3 to upload data.

→ Step 2 :- then, we will migrate website onto AWS infrastructure.

⇒ // Code //.

classifying broadly into two things :-

1.) Database upload / connect to RDS.

2.) Host our app / index.php (zip file) to the Elastic beanstalk.

→ check our php files (index.php & list.php are running) with the database (MySQL) locally.

→ create database images;
→ use images;
→ create table names (name varchar (100));

Date: _____ Page No. _____

Topic: _____

UI :-

Upload to S3 : → Hi

Select a file [choose file] abc.png [submit] → 2 inputs.

check list. → <a href>

Now, trying to upload a file will give error because authentication is still left.

AUTHENTICATION :- Go to AWS IAM.
create a new user.
web-demo-jiyush (Username)
Save :- ACCESS key ID &
ACCESS Secret Key.
(will be using this in our code).
Set permissions.
select AmazonS3FullAccess. ✓.
→ used to connect our website to AWS S3.

26/06/25

Date: _____ Page No. _____

Topic: _____

→ Code to upload files to S3.

after adding key & secret key our website is now authenticated

→ now, we will be able to upload our files in bucket

→ bucket → S3 > General purpose bucket > create bucket > & select bucket type to general purpose, give name, fill all other details > create bucket. (web-bucket)

→ use web-bucket (bucket name) to putObject in S3 in our website

→ Now after uploading a document / file successfully you will be able see your file in S3 bucket.

→ local mysql (SELECT * FROM names;) to check all the files in our local mysql.

Date: _____ Page No: _____

Topic: _____

⇒ Deploying our DB on AWS Aurora & RDS.

- click on create DB > select engine option (MySQL), create method easy create, Templates Dev/Test/free-tier, self managed (master username, password), db-identifier name, select default VPC, public accessibility (yes : from localhost we should be able to upload data our DB to AWS), choose existing VPC security group (default), DB name (images), no backup, disable updates as of now, & leave everything as it is default > create DB.

SAVE Hum

⇒ Deploying our website from localhost on our AWS.

- Using Elastic bean stalk that is PAAS.
- from AWS Console > all services > elasticbeanstalk (no need to configure anything it will do for us) >

Date: _____ Page No: _____

Topic: _____

create application > give name & description > create > platform > (select php > sample application (yes), create application.

⇒ After creation of DB in RDS, we will get an endpoint (URL). That is used to connect our website DB to RDS.

⇒ Code explanation :-

index.php folder structure.

myapp
- vendor /
- index.php
- list.php

index.php

download it from browser

→ imported our AWS SDK (aws-autoloader.
→ using services S3Client, .php)
S3 Exception.
→ S3Client :- credentials (key & secret), region ap-south-1, few more details.

Date: _____ Page No. _____

Topic: _____

list.php

→ normal DB connectivity & retrieval. using the bucket URL to retrieve data from the bucket.

<a href = 'https://[redacted]web-bucket.s3.ap-south-1.amazonaws.com/'

→ status available of RDS.

→ DB created on RDS (but table is left to be created), create it on this DB instance (copy the endpoint).

→ CMD > go to bin directory of mysql > ... \bin >

mysql -h <endpoint> -u <our RDS DB username> -p

(created on previous page) password

if STUCK: VPC

→ Reason could be in security group that we associated to it.

→ Check inbound rules to accept traffic.

Date: _____ Page No. _____

Topic: _____

in our case it was accepting traffic but from custom of SG, change it to Anywhere 0.0.0.0/0 (for practise / demo purpose only)

→ try log in again → success.

✓ use mags ; (already created).

create table name(name varchar(100));

↳ # create this

change host, username, password from index.php & list.php to

created on previous page

$host = endpoint of RDS.

$username = master username.

$password = master password.

→ upload a file from our website on our localhost.

X → check in localhost MySQL (CMD).

✓ → check on our endpoint (RDS) MySQL(cmd)

→ Now, data got uploaded to the RDS, & not on localhost.

→ by clicking on checklist.php from browser we will get the RDS records/-.

Date: _____ Page No: _____

Topic: _____

→ Done, Our website has connected to the DB instance on AWS (RDS).

⇒ Next step is to put our website on AWS (Elasticbeanstalk), so, anywhere in the globe can access it.

→ go to Dashboard of Elasticbeanstalk > click on Upload & Deploy > upload our projects zip file, give version label > deploy.
version :- aws-demo-website:1.0
upload & deploy :- aws.zip
↳ vendor/
↳ index.php
↳ list.php

→ any change in my code in future, I will have to again upload & deploy in the same manner.

→ aws Elasticbeanstalk > Applications >

Date: _____ Page No. _____

Topic: _____

click on application name (web-app) > look for domain link, open it & Yu we have successfully deployed our website & DB on AWS.

→ check its Working / flow of website, should work fine.

→ we can see each & every document / file Uploaded from S3 website in S3 (bucket).

⇒ Thats it.

We have Completed our AWS Course with Services like :- EC2, S3, IAM, EFS, VPC, Aurora & RDS, Elastic Beanstack.