Planning to Production — Day 1

DevOps.

SDLC ⇒ Software Development life cycle

Servers ⇒ just like o/s where applications will be up & Running.

O/S ⇒ Basic requirment to run any application.

Application ⇒ (communicate) Hardware → RAM, CPU, HD

↳ Going to Manage H/w Services.

Windows / ~~Unix~~ / MacOS / Linux

OS → Application → Instamagam feels

→ Microsoft Team / User friendly.

x Not open source x

x Customization

OS

Windows | MacOS | Linux

Windows → open source x → Customizations x → Heavy

MacOS → Heavy

Linux → can help to perform any kind of optization / customization

→ open source.
→ Great Community
→ free to use.
→ Kernel. (Brain of o/s).
→ Security / Stable / Light weight
→ Compatible with Hardware.
→ Multitasking.
→ Multiuser.

Linux Distributions.

→ companies added some extra features.

Linux Kernel

Ubuntu | RedHat | Kali

↓

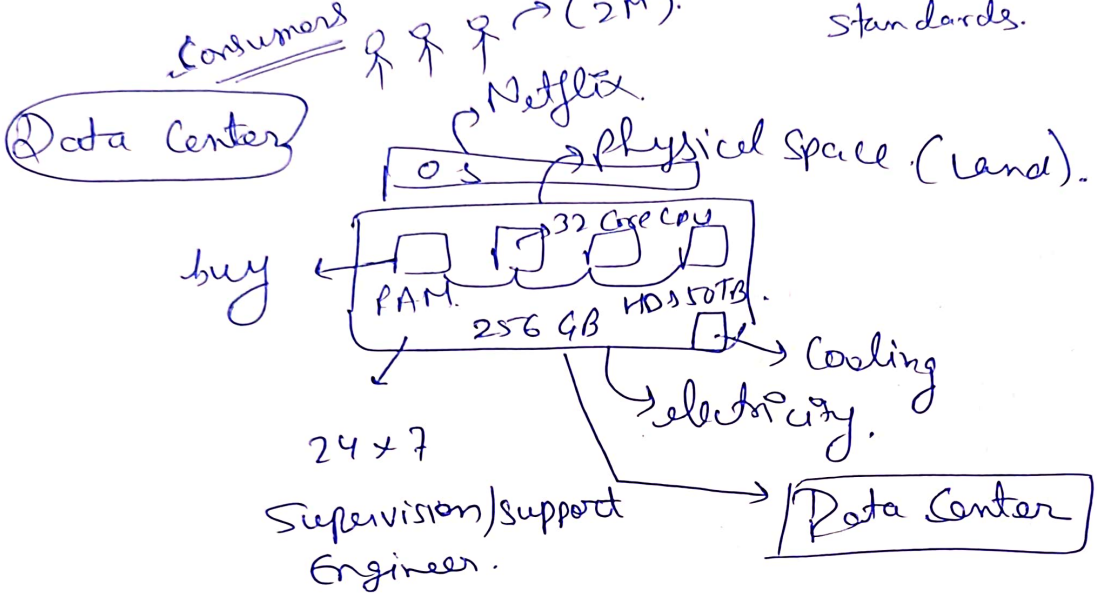
Linux as Base + extra features. (Bundled up Together).

⇒ Will be using Linux Distributions / instead of Directly using Linux Kernel.

(RedHat). (yum command)

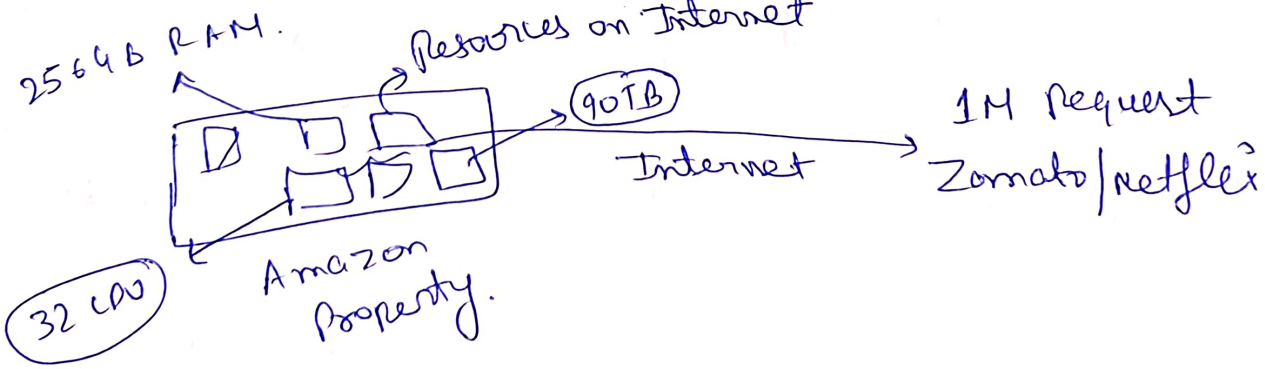
Ubuntu (apt).

↳ By most of the companies { Made for servers / stable / Added lot of Program / Packages to create a server. / Stability / Security. Better performance / Compliance standards.

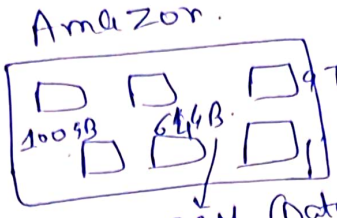


⇒ Cloud Computing | Amazon | Google | IBM | Microsoft.

↳ offering some kind of services.
(Create Data Centers Across the World).



Startup Case



2 Hrs (3 $) Rent → Startup

largest Market Share

Amazon Hotstar Netflix ← AWS

Google. → GCP

Microsoft → Azure.

Public Cloud ⇒ Not Have Physical Acess / Not Authorization to Touch, but present somewhere in ~~the~~ world and can access Virtually.

(Hosted / Managed by 3rd party People / open).

Publically Available.

Story Telling

ola/zomato.

App. (Program)

Client/
Consumer.

Business Team
(Idea)

↳ Developers
↳ Code.
↳ Operations
Team.

OS → Linux.
(RedHat).

Hardware

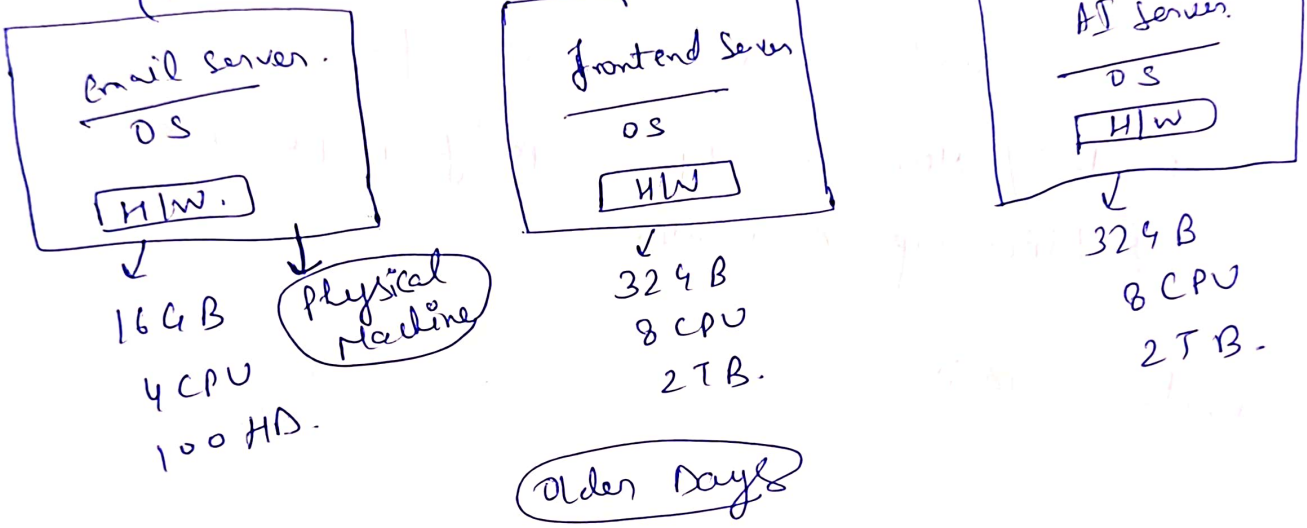
Public cloud
Providers → AWS

(Paint, Zoom, Chrome).

OS

H/W.

Issues



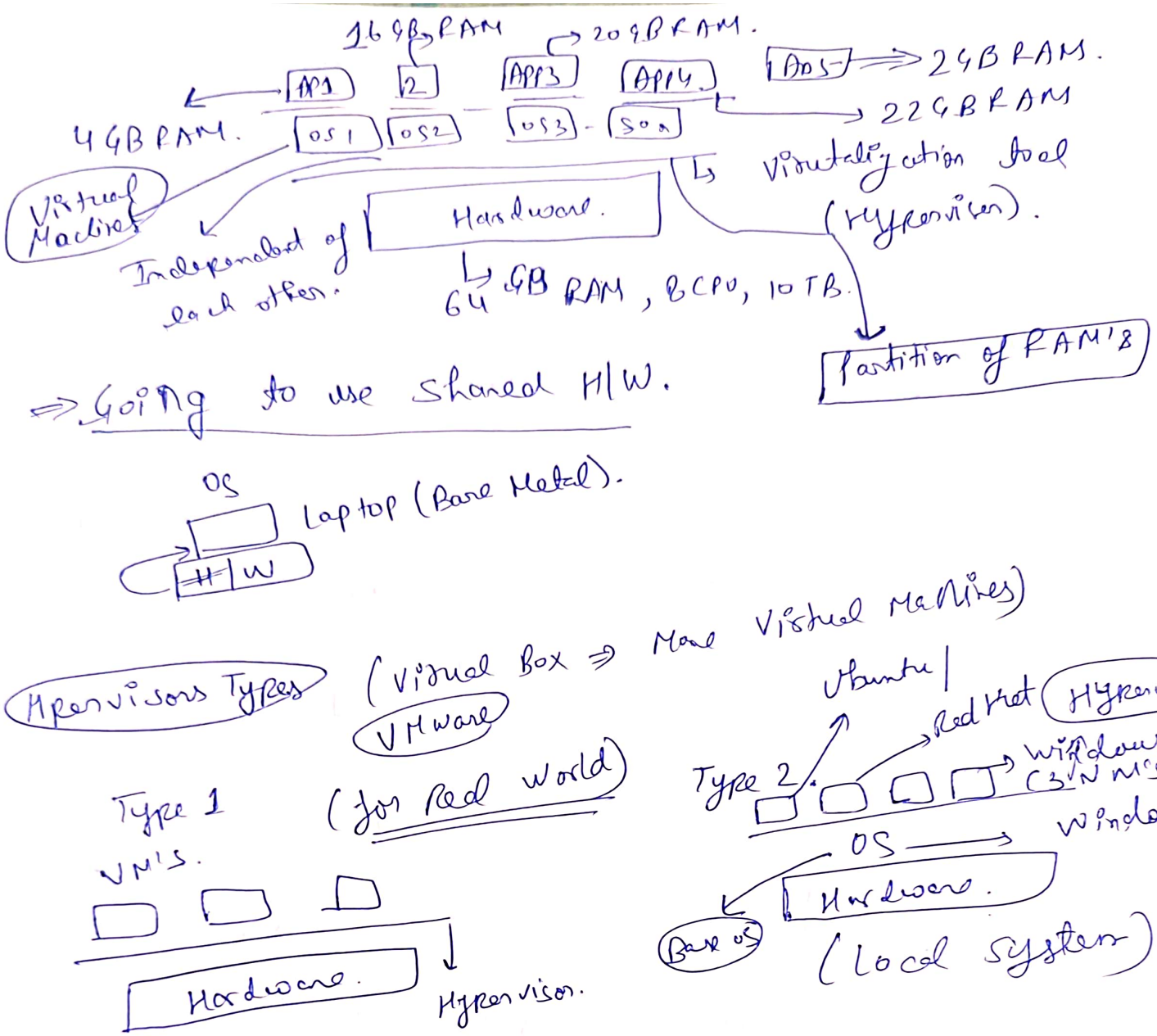
Older Days

(under utilization)

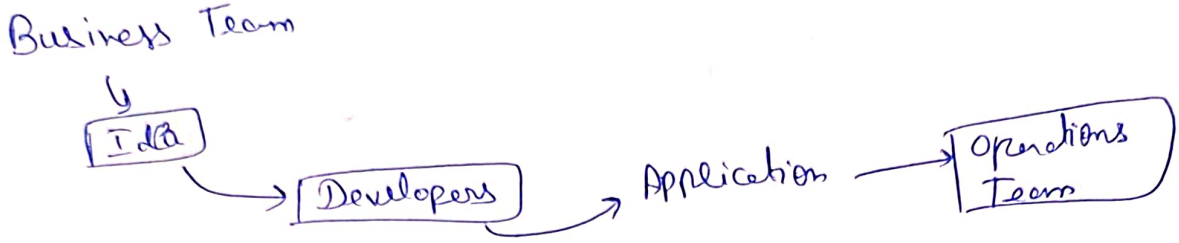
One set of H/W can have only set of Application / and also only one set of OS.

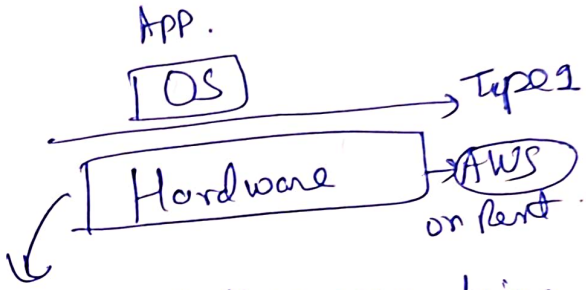
Virtualization

. H/W only one time.

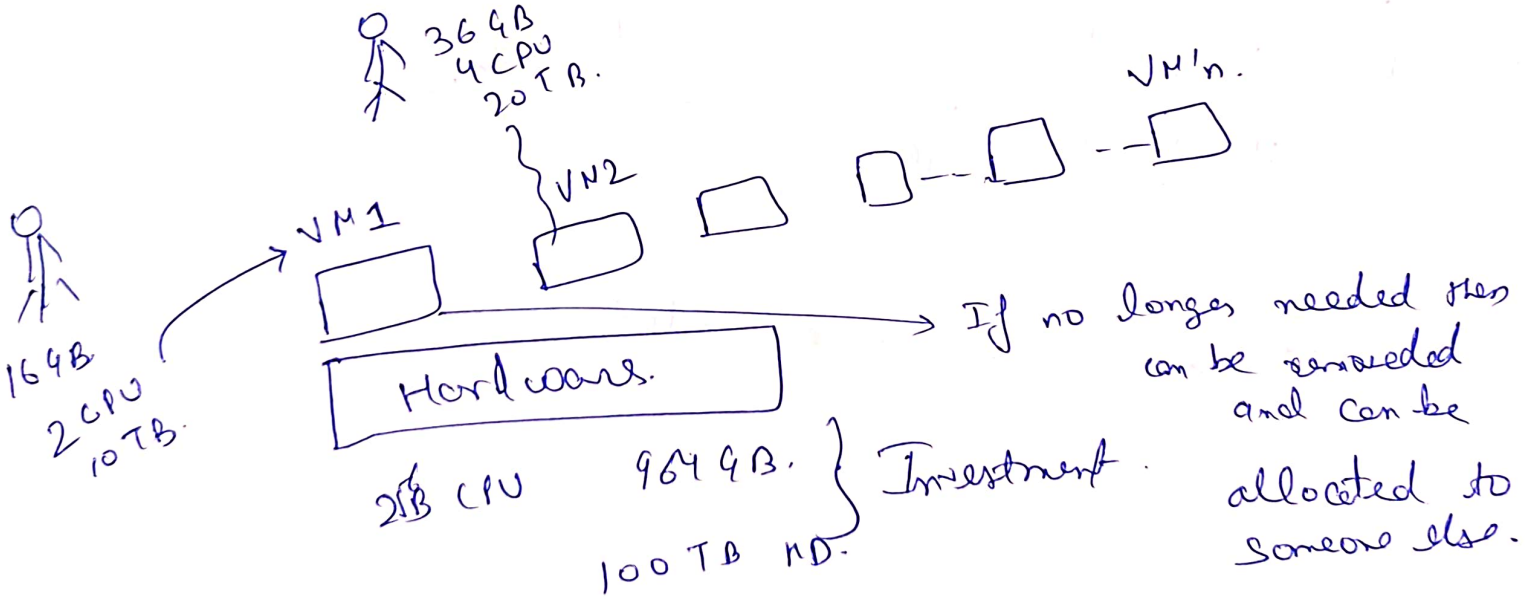

16 GB RAM
20 GB RAM.
APP1
APP2
APP3
APP4
DOS
2 GB RAM.
4 GB RAM.
OS1
OS2
OS3
SOn
22 GB RAM
Virtualization tool (Hypervisor).
Virtual Machines
Independent of each other.
Hardware.
64 GB RAM, 8 CPU, 10 TB.
Partition of RAM's
⇒ Going to use Shared H/W.
OS
Laptop (Bare Metal).
H/W
Hypervisors Types
(Virtual Box ⇒ Most Virtual Machines)
VMware
(for Real World)
Type 1
VM's.
Hardware.
Hypervisor.
Type 2
Ubuntu
Red Hat
Hypervisor
Windows (3 VM's).
OS
Windows.
Hardware.
Bare OS
(Local System)
⇒ Disc Image / Iso file ⇒ Entire content of O/S.

Flow Diagram: Virtualization gives Birth to Cloud Computing.





Type 1 Hypersison being used Internally by AWS to create VM's.



Shared Resources But no one can access each others data.

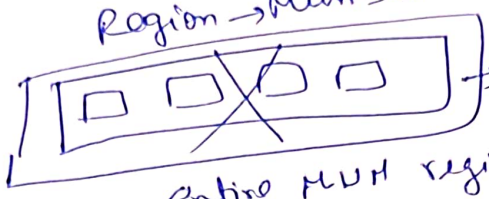
⇒ AWS Global Infrastructure.

Data Centers ⇒ where physical resources will be kept.

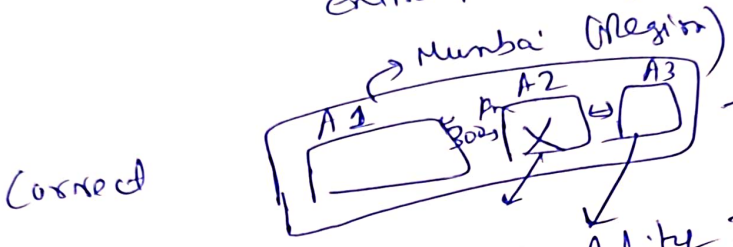
Region → Availability Zones.

↙

Mumbai



Entire MUM region will be down.



Correct → Cost Increases But Availability is Guaranteed.

Availability Zones.

⇒ AWS account creation

(AbhiDevOps)

Linux ⇒ ssh keys (4096 for RSA)

Key Pair RSA Algo. ((.pem/ PPK file))

Putty ↳ open SSH.

⇒ GUI ⇒ Makes o/s very slow.

to create a file (touch).

⇒ List of files & Directories ⇒ ls.

⇒ cat ⇒ read the file.

write ⇒ cat > a.txt.

↳ Redirection Symbol.

remove a file

rm a.txt.

folder creation mkdir gfg

ls

Ctrl D

cd ⇒ change Directory

Pwd ⇒ Print Working Directory / Current Directory)

Software Installation

Linux ⇒ yum command. | ubuntu ⇒ apt.

yum install httpd (Apache server).

⇒ In which user

↓

whoami (EC2 User).

Can't Install a package.

⇒ Switch to root user.

sudo su root. ⇒ whoami

↓

Switch user

Install a server

① Download Program ② Configuration ③ Start the server.

Web Server (To Host the file).

(Apache / Nginx) etc.

↳ /var/www/html → In Apache configuration file.

Start the server (Linux)

↓

systemctl start httpd.

systemctl status httpd. (Because of firewall rules)

EC2 VM.

Security Group | firewall which IP Need to be blocked.

OS

Ping 8.8.8.8 outbound ⇒ can go anywhere.

Inbound rules ⇒ from where it can come.

SSH ⇒ Protocol used to connect to VM.

Add Rule HTTP traffic / 80 / Anywhere / Save rules.