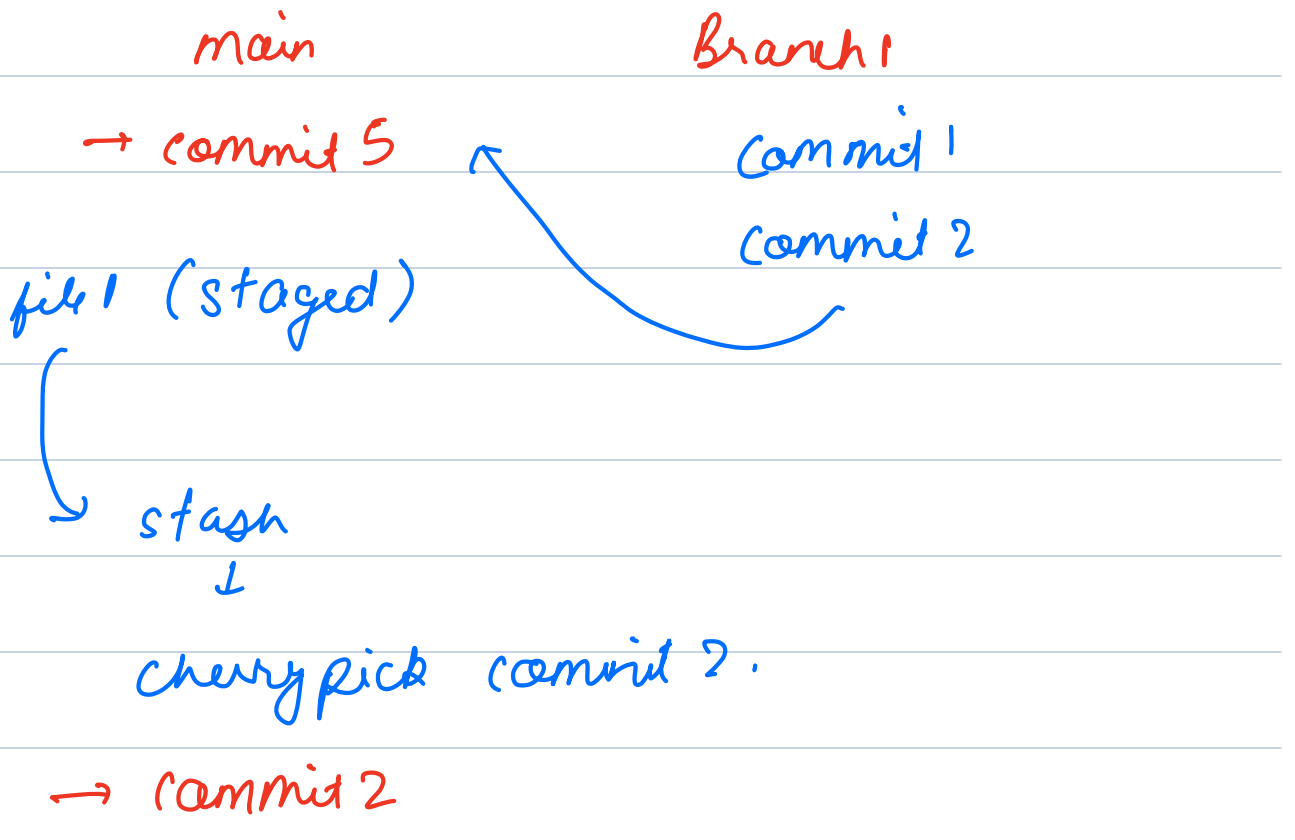
Docker Basics

- starts at 9:05pm

AGENDA

1. Application Development Journey
2. Docker Architecture
3. Namespaces and Cgroups
4. Docker Installation on ubuntu
    1. Understanding storage drivers.



stash pop

file (staging)

→ Application Development Journey

Physical machines → Virtual Machines

↓

Containers.

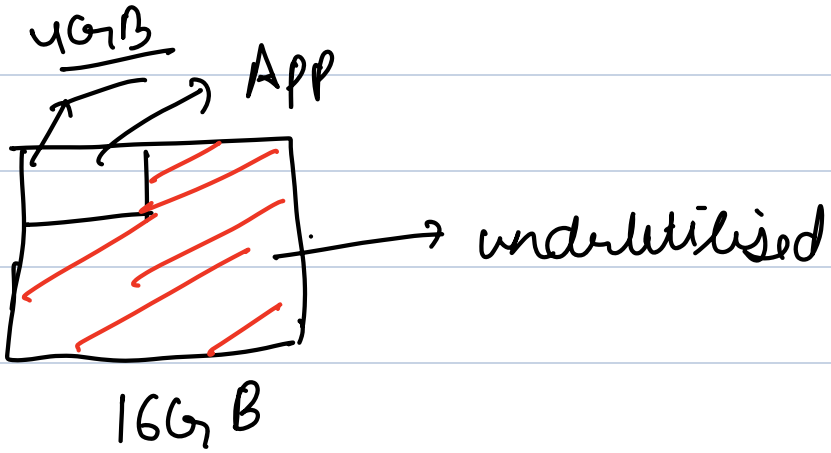
→ resource utilisation
→ scalability
→ portability

Physical Machines.

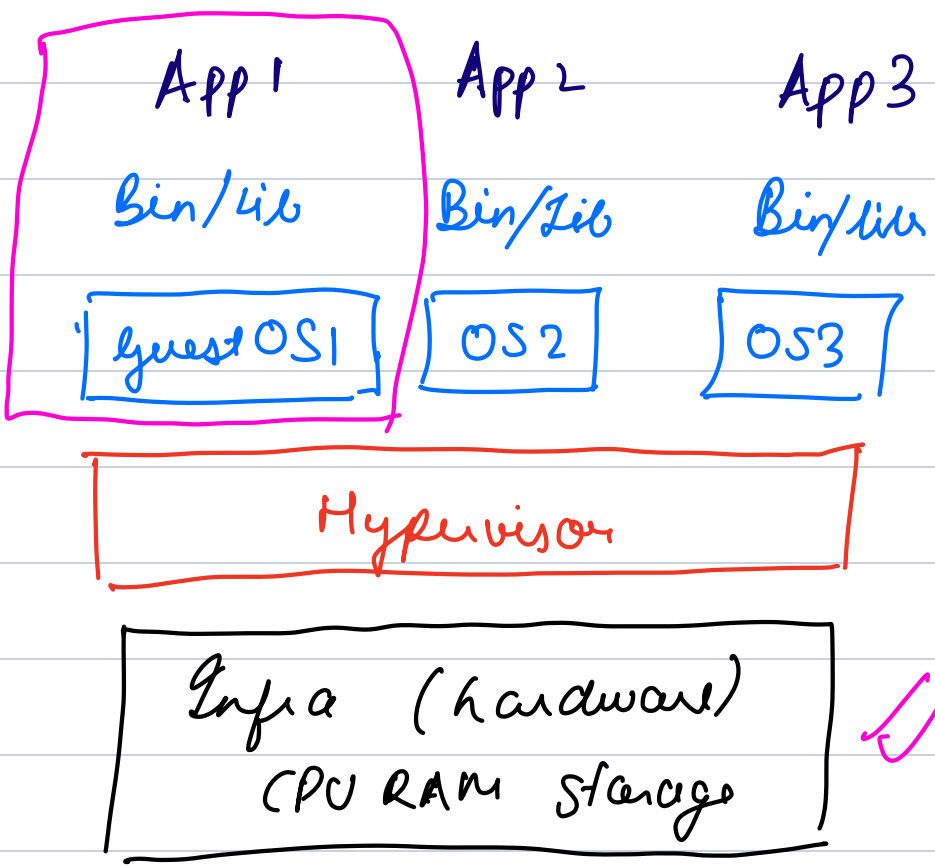
→ single OS
→ dedicated hardware
→ entire application stack runs on them

16 GB
4 CPU
500 GB

32 GB



Virtual Machines



Hypervisor → Managing virtual machine by abstracting hardware resources.

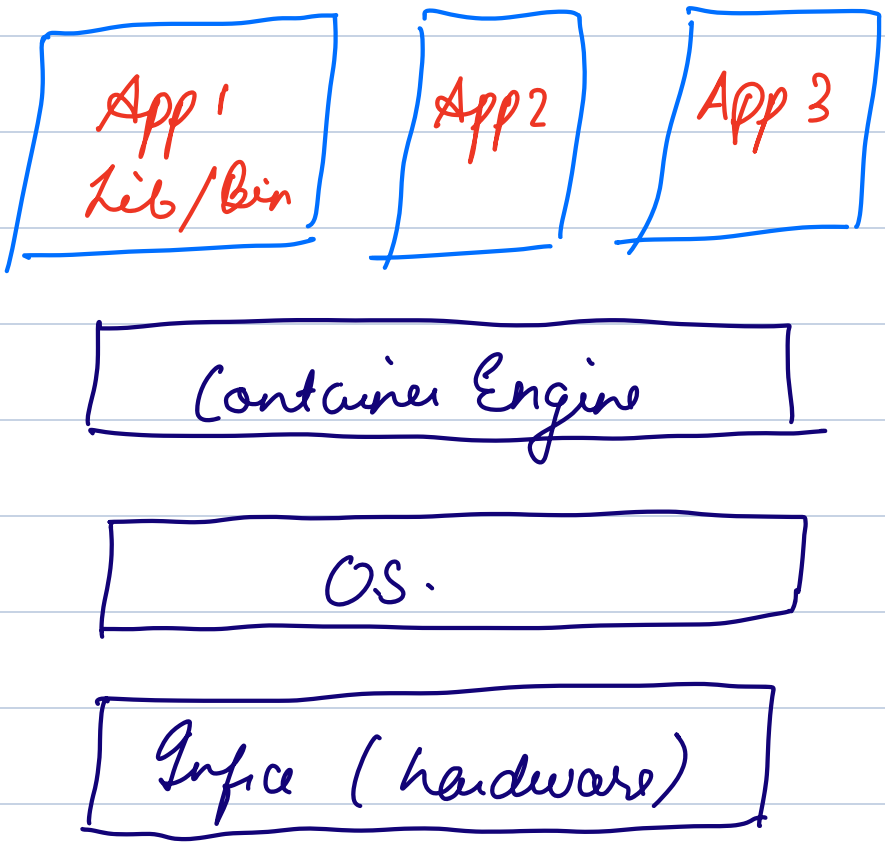
type 1 (Bare-metal) eg: VMWare ESXi
type 2 (host OS) eg: VirtualBox.

Advantages →

1. Resource Utilisation
2. Isolation
3. Portability.

Challenges

→ Each App requires its own O.S.
→ Boot time.
→ Complexity. (Managing hypervisor)



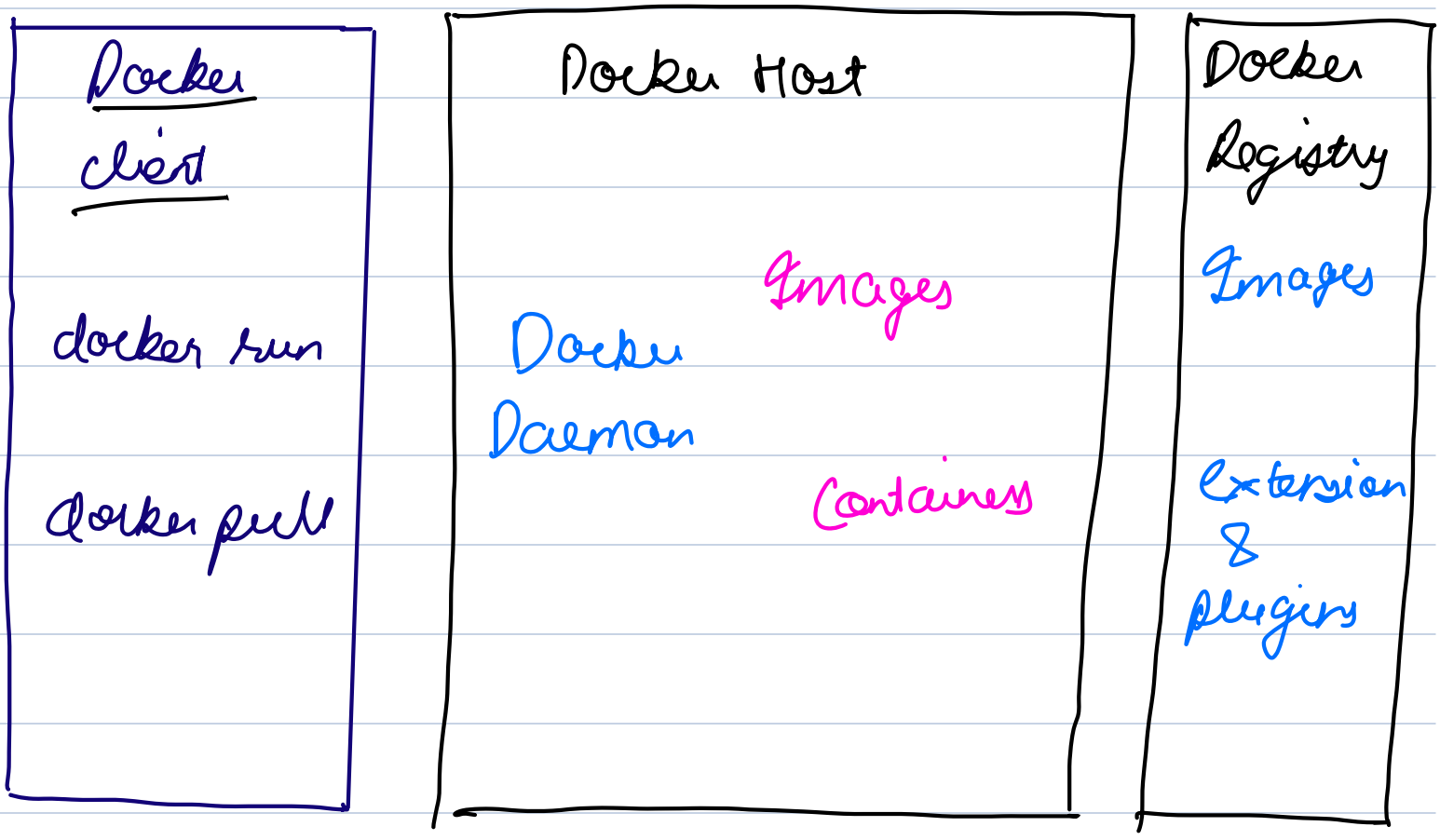
Containers are lightweight, portable, and executable units of software that package application code along with its dependencies, running on a shared operating system kernel.

Characteristics of a container

1. Lightweight
2. Self contained
3. Portable
4. Isolated
5. Consistent

Container Engine → Docker, CRI-O, Podman, containerd.

→ Docker Architecture



docker daemon

→ Building images

→ Running containers

→ managing networks volumes used by containers

→ Handling container orchestration

GITHUB Repo

→ python123.py
→ requirements.txt
→ Dockerfile
→ jenkinsfile.

Registry

ubuntu
nginx
my-app.

→ Docker Images.

template (Image) —Run→ Container (Running application)

Dockerfile

```
FROM ubuntu:20.04      # Base layer

RUN apt-get update     # New layer

RUN apt-get install -y nginx  # Another new layer
```

} image → Run

Container

ubuntu
nginx

```
# Use the official Python image as a base

FROM python:3.10-slim

# Set the working directory inside the container

WORKDIR /app

# Copy the requirements file to the working directory

COPY requirements.txt .

# Install dependencies

RUN pip install --no-cache-dir -r requirements.txt

# Copy the rest of the application code to the working directory

COPY . .

# Specify the default command to run the Python script

CMD ["python", "app.py"]
```

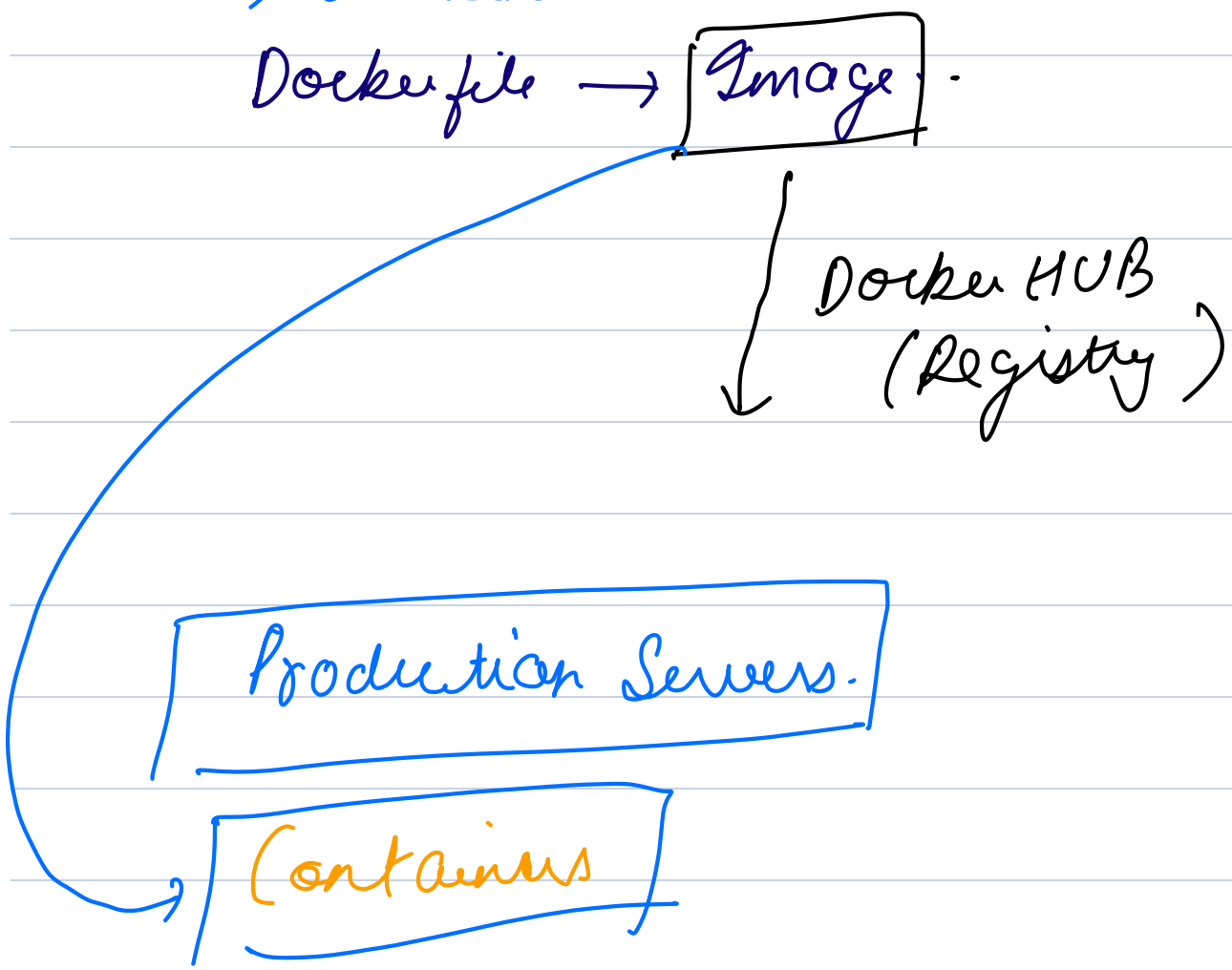
→ app.py.
→ requirements.txt

→ Images are Read Only.

GITHUB Repo
- → python123.py
- → requirements.txt
- → Dockerfile
- → jenkinsfile.

→ Jenkins

Dockerfile → Image.



Break 10:30 pm

→ Namespaces Cgroups.

→ Types of Namespaces

1. PID Namespace
2. Network Namespace.
3. Mount Namespaces.
4. User Namespace.

Containers root user.

Base OS

→ unprivileged user.

5. IPC Namespace.

→ Controlgroups cgroups

1. Resource Limiting.

512 MB 2 GB

2

② Resource Prioritisation.

③ Accounting.
monitoring the resource usage.

install → cp mkdir, chmod.

Installing Docker.

https://docs.docker.com/engine/install/ubuntu/

→ storage Driver
used for managing how data is stored and accessed within containers.

→ overlay 2. (default)
→ devicemapper (advanced volume mgt)
→ btrfs
→ zfs

ExecStart=/usr/bin/dockerd --storage-driver=devicemapper

Docker commands

→ docker pull ubuntu:latest

Docker HUB

ubuntu 20.3

ubuntu 20.4

ubuntu 20.5

→ docker run

ocker run -d ubuntu bash -c "while true; do echo "Hello $(date)"; sleep 1; done"

→ docker ps
list running containers.

→ getting inside a container

docker exec -it container_id /bin/bash

→ docker stats.

docker run -it --memory="1g" ubuntu

docker run -it --cpus="1.5" ubuntu