

Register No: 20L31A05J4

Experiment No: 04

Date:

S. No	Component	Max. Marks	Marks Secured
1	Preparedness	2	
2	Viva-Voce	2	
3	Experiment	3	
4	Analysis & Record	3	
Total		10	
Date		Signature of the Lab teacher	

AIM: To Install and Configure Docker for creating Containers of different Operating System Images

Theory:

Why Docker:

Docker simplifies and accelerates your workflow, while giving developers the freedom to innovate with their choice of tools, application stacks, and deployment environment for each project.

Containers are a standardized unit of software that allows developers to isolate their app from its environment, solving the "it works on my machine" headache. For millions of developers today, Docker is the de facto standard to build and share containerized apps - from

Register No :

Experiment No :

Date:

desktop, to the cloud.

### Getting Started With Docker:

Docker can be easily installed on macOS or any Linux based system. Docker host runs on Linux kernel on top of which containers are run,

whereas when windows are in scope, a Linux virtual machine needs to be installed on which docker host is mounted on which containers will run.

### Introduction to Images and Containers :

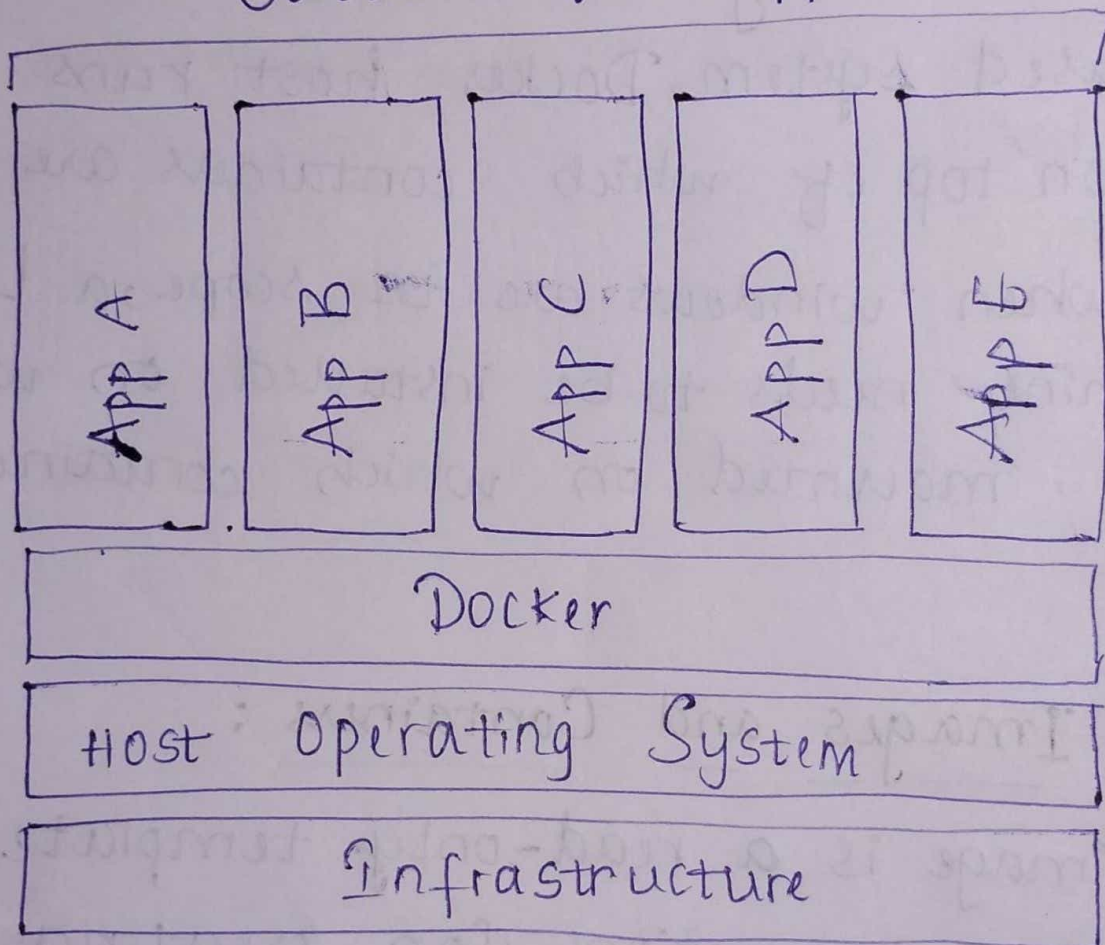
A Docker image is a read-only template that contains a set of instructions for creating a container that can run on the Docker platform. It provides a convenient way to package up applications and preconfigured server environments, which you can use for your own private use or share publicly with other Docker users.

We can create a Docker image by using one of 2 methods :

1) Interactive

2) Dockerfile

## Containerized Application





Register No :

Experiment No :

Date:

## Container:

A Docker container is an open source software development platform. Its main benefit is to package applications in containers, allowing them to be portable to any system running a Linux or Windows operating system (OS). A container is created when a docker image is run on docker host. Multiple containers of same image can be created with different port mapping.

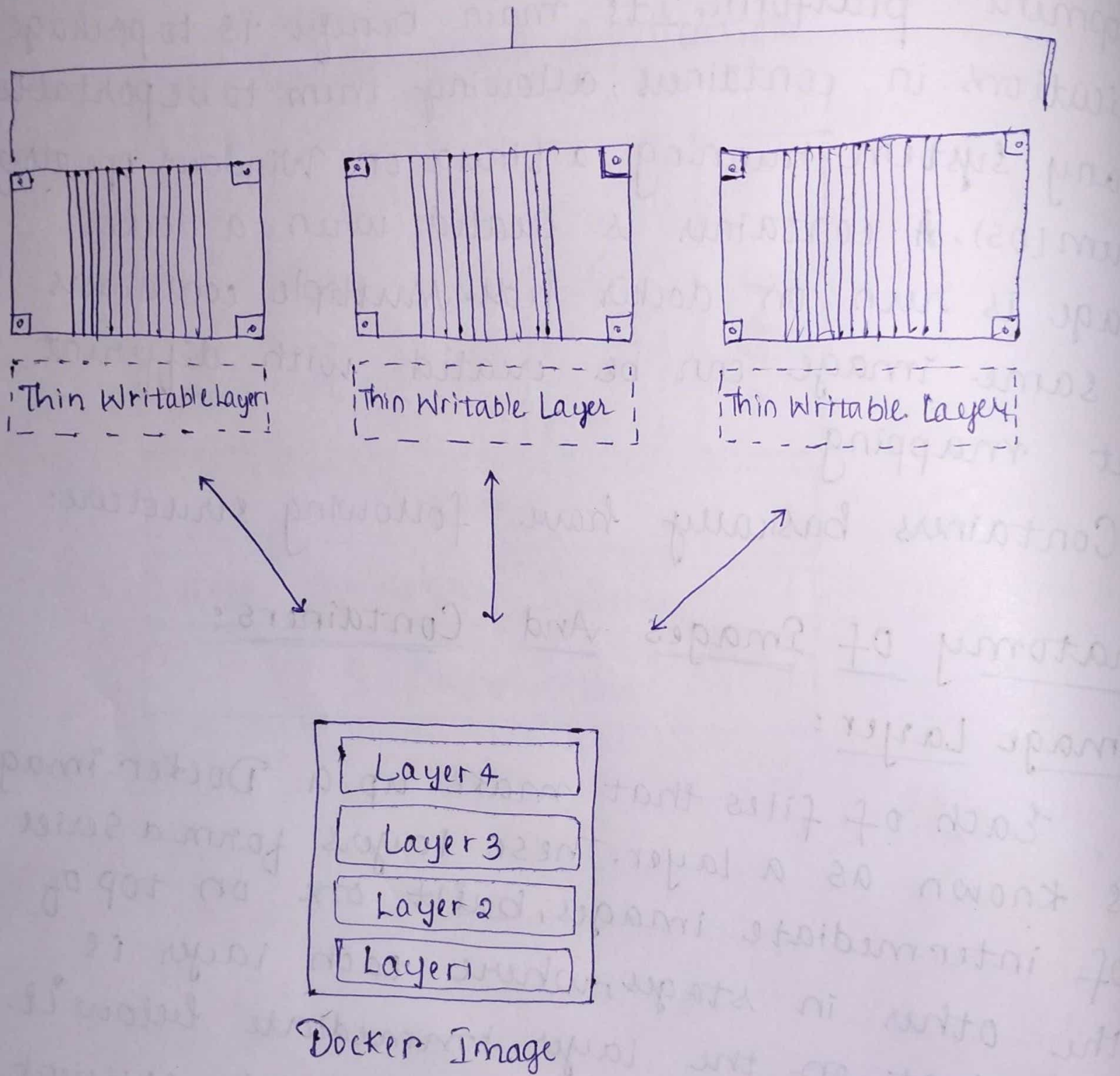
Containers basically have following structure:

## Anatomy Of Images And Containers:

### Image Layer:

Each of files that make up a Docker image is known as a layer. These layers form a series of intermediate images, built one on top of the other in stages, where each layer is dependent on the layer immediate below it. The hierarchy of your layers is key to efficient lifecycle management of your Docker images.

# Docker Containers





Register No : Experiment No : Date: 

## Container Layer:

Each time Docker launches a container from an image, it adds a thin writable layer, known as the container layer, which stores all changes to the container throughout its runtime. As this layer is the only difference between a live operational container and the source Docker image itself, any number of like-for-like containers can potentially share access to the same underlying image while maintaining their own individual state.

## Docker Commands:

→ Docker Run:

```
$docker run [OPTIONS] IMAGE[:TAG|@DIGEST] [COMMAND] [ARG..]
```

→ Detached Mode:

```
$docker run -d ubuntu
```

→ CID Setup

```
$docker run -d ubuntu -cidfile/tmp/helloworld.cid
```

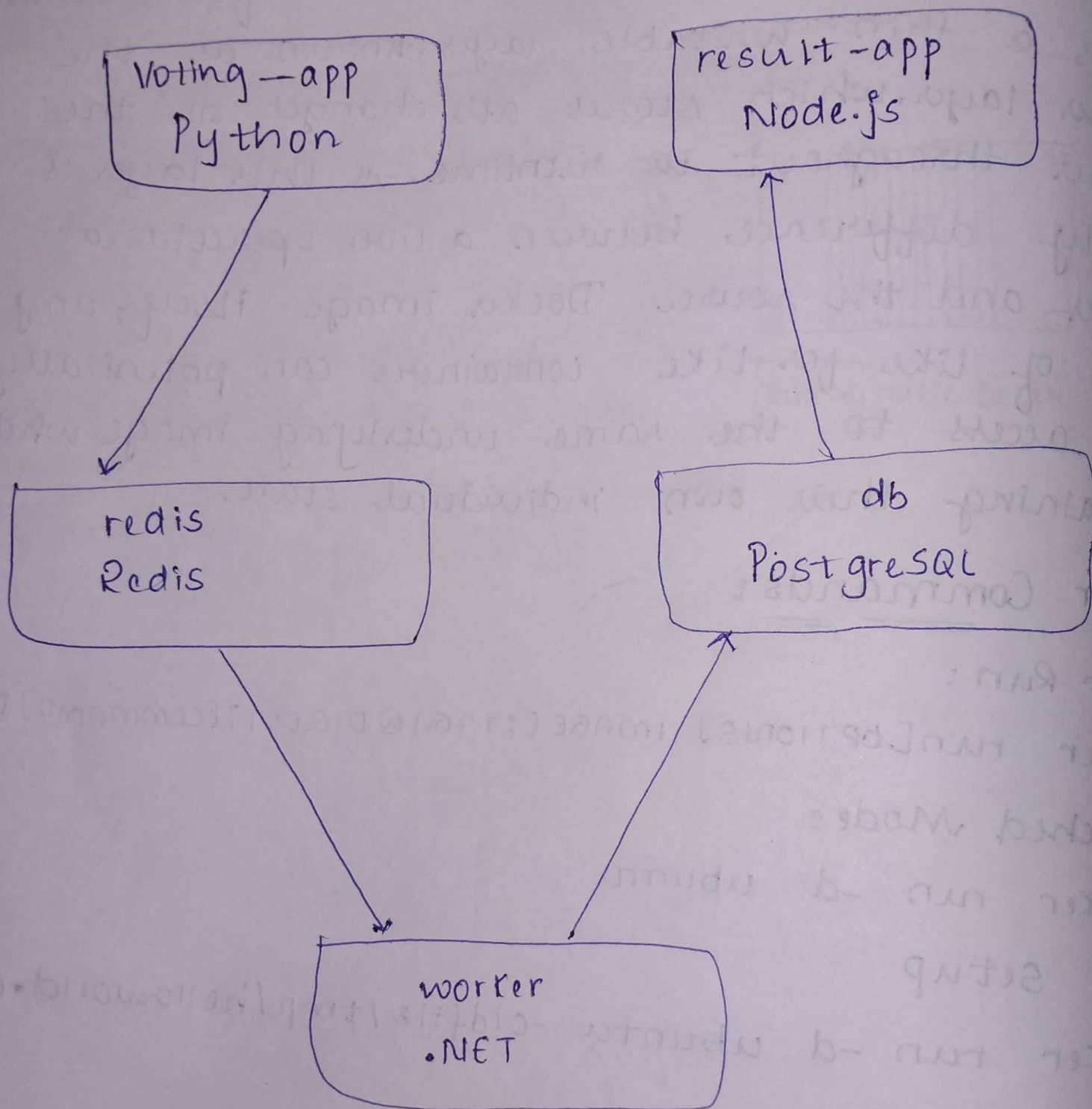
→ Version Tags of Image:

```
$docker run -d ubuntu:22.04
```

→ Container: Users network of another container specified.

```
$docker run -d --name redis example/redis --bind
```

```
127.0.0.1
```



Register No :

Experiment No :

Date:

```
$docker run --rm -it --network container:redis  
example/redis-cli -h 127.0.0.1
```

→ User Defined Network:

```
$docker network create -d bridge my-net  
$docker run --network=my-net -itd --name=  
container3 busybox
```

## Docker Compose :

Docker Compose yaml file is created when you want multiple containers to be brought up at a single time.

If we consider the classic voting application as example, it has multiple components.

- Vote - UI to cast vote
- Redis - in memory DB used as cache/message broker
- Vote worker - the backend that registers the vote and persists into the system.
- db - a classic PostgreSQL used to store votes
- Result - UI to show votes casted, to check who is leading/winning.



## Linux System

PID: 1

PID: 1

PID: 2

PID: 3

PID: 4

PID: 5

PID: 6

## Child System (Container)

PID: 1

PID: 2

Register No :

Experiment No :

Date:

Docker Engine :

Docker container PID mostly will be different  
on host of the container.