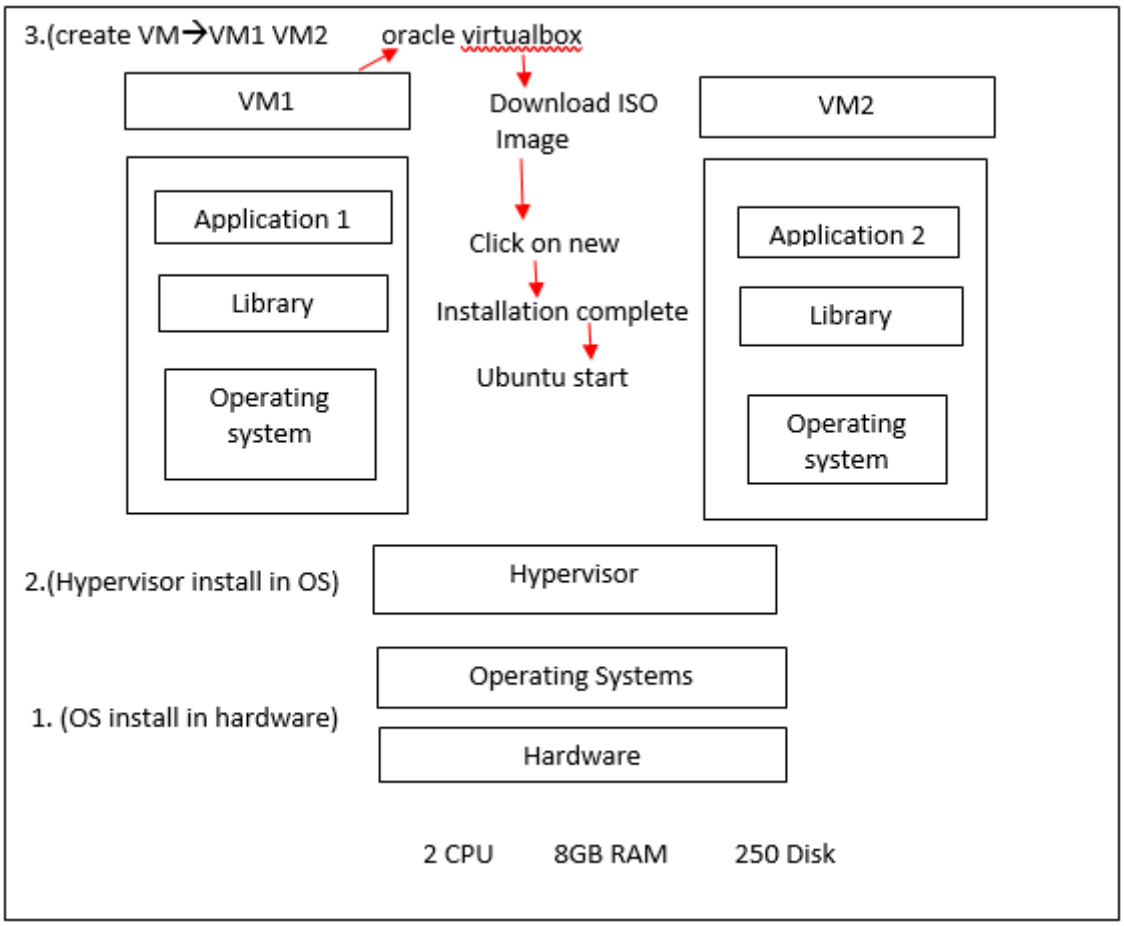
Docker

Hypervisor

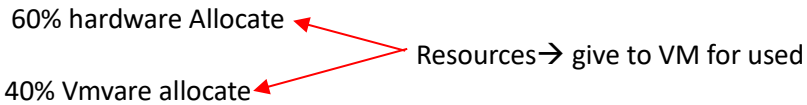
The hypervisor loads the virtual machine images to create multiple virtual operating systems.

The physical machine is known as a host & the virtual operating systems are guest.



- ✓ Virtualization technology launch
- ✓ 2 Operating systems parallel used in one system
- ✓ Operating system install in hardware
- ✓ Hypervisor install eg:- Vmvare , Oracle virtual box in operating system
- ✓ Create virtual machine i.e. VM1 & VM2 eg.:- Ubantu M/C , Centos , Linux
  Vm1→ oracle virtual box
- ✓ Download ISO image
- ✓ Click on new
- ✓ Installation complete
- ✓ Ubuntu start

- ✓ Then allocate hardware



- ✓ Library install in operating system
- ✓ Application deploy (app 1 , app2)
- ✓ Operating system heavy because of kernel.

Disadvantage Of hypervisor

- Time consuming
- Expenditure cost
- Limited no of VM's can be install. It depends upon configuration of base machine
- If we install no. of OS in hypervisor, it takes all space from base machine storage which leads to increasing load of the base machine
- It uses more CPU, hardware
- Highly maintain
- No guarantee on resources

Docker

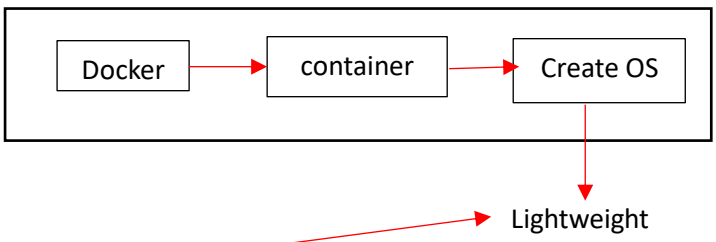
Docker is an open-source platform that allows you to automate deployment scaling and management of applications using containerization.

Docker → Allow you → deployment of Application using containerization

It provide a way to package an application and its dependencies into a standardised unit called container.

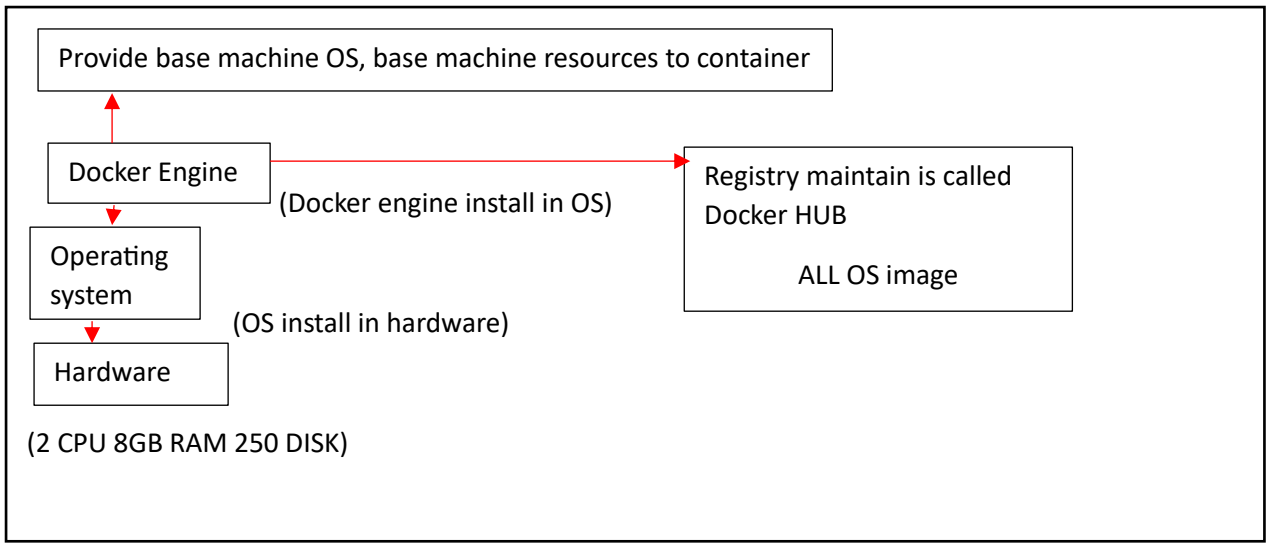
Docker has a facility to install directly instead of hypervisor and create no of VM , this VM nothing but container

- ✓ No need of installation of separate OS
- ✓ No need of hardware distribution
- ✓ No need to install separate library



Just provide shell → nothing but image → OS

- ✓ Container used base machine resources



Docker engine share base machine kernel

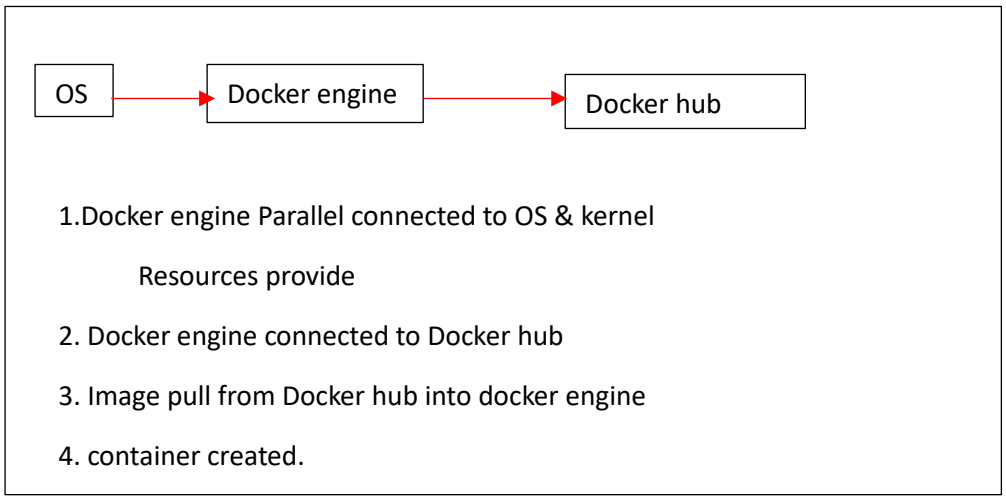
Docker hub provide image to container

Docker hub connected to docker engine

Docker engine allocate resources & share OS kernel

Container creation command on terminal

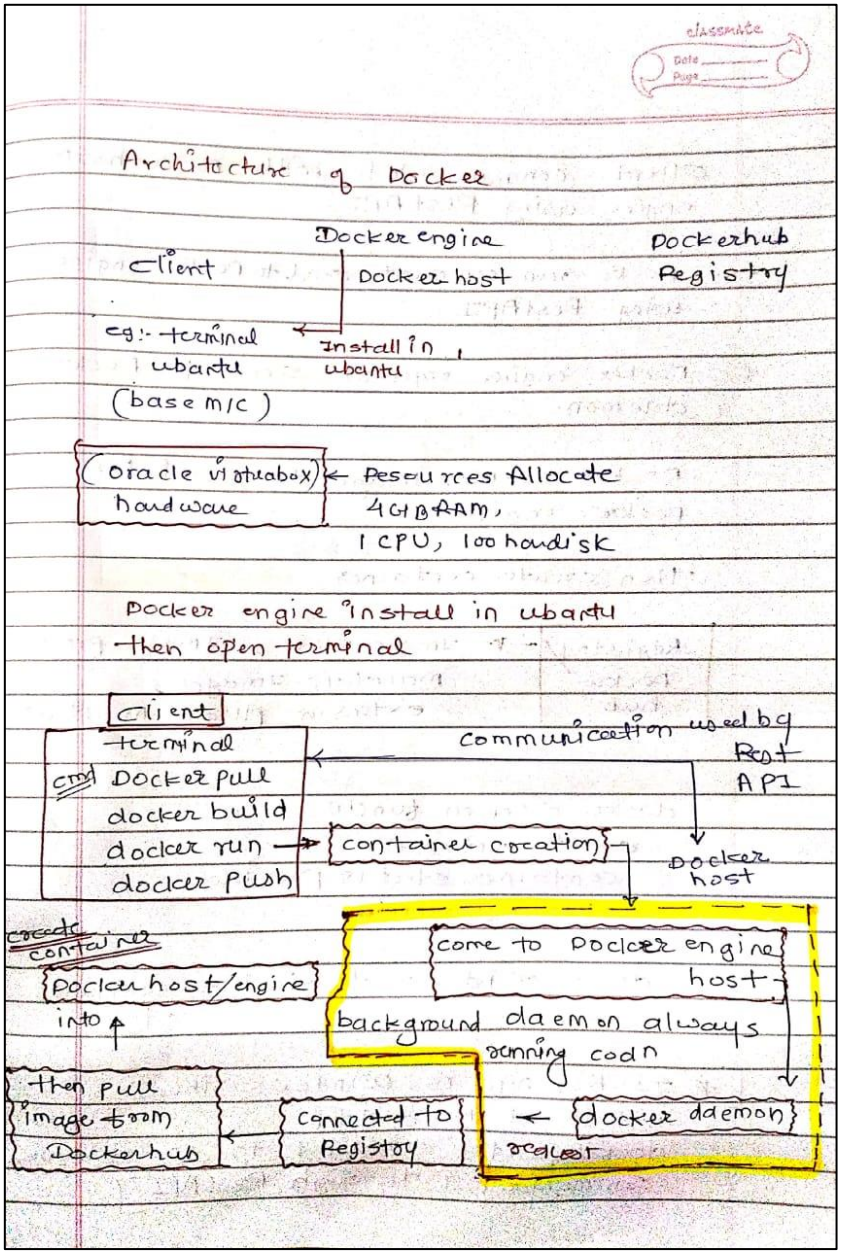
- Docker engine install
- Container create manually then give image which OS you want
- Container run → enter
-



- Install library manually
  E.g.:- java, git, mvn , open cv , my sql

Docker file

- ➔ Mention os
- ➔ Library define
- ➔ Image name
- ➔ Directory define
- ➔ Then make iamge
- ➔ Container done
- ➔ Used base machine resources automatically



Client communicated with Docker host engine using Rest API

eg:- Docker run command send to Docker engine using Rest API

Docker engine request send to Docker daemon

Docker daemon image download before Docker engine

then create container

Registry / Docker hub → image store, image provide maintain images, extension plugin maintain

docker daemon functn
create container
container status provide
container delete

docker client → terminal → command type

* If base m/c is Windows then install docker desktop
docker desktop connected to docker engine communicated through Rest API

Docker pull will verify if the image is already download locally.
If not it will download it from Dockerhub
cmd docker pull percona

Docker build will build the current Dockerfile and will create locally an image for our application.

cmd docker build .

Docker Run will take the image & run the container
cmd → docker run nginX

Docker Push will upload the image/ extension/plugin to Dockerhub
cmd → docker push <appln login>

Docker installation

```
kunal@kunalsh:~$ sudo apt-get update
[sudo] password for kunal:
Get:1 http://security.ubuntu.com/ubunt
```

If docker present then remove it

#uninstall docker:

$ sudo apt-get remove docker-ce

```
kunal@kunalsh:~$ sudo apt-get remove docker-ce
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed ar
  containerd.io docker-buildx-plugin docker-ce-cli doc
Use 'sudo apt autoremove' to remove them.
The following packages will be REMOVED:
  docker-ce
```

Install docker dependencies

$ sudo apt-get install apt-transport-https ca-certificates curl gnupg-agent softwareproperties-common

```
kunal@kunalsh:~$ sudo apt-get install apt-transport-https ca-certificates curl gnupg-agent software-properties-common
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20230311ubuntu0.22.04.1).
```

Add GPG key to apt repository

$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add –

```
kunal@kunalsh:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
```

Setup Stable repository:

$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"

```
kunal@kunalsh:~$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
Repository: 'deb [arch=amd64] https://download.docker.com/linux/ubuntu jammy stable'
Description:
Archive for codename: jammy components: stable
More info: https://download.docker.com/linux/ubuntu
Adding repository.
Press [ENTER] to continue or Ctrl-c to cancel.
```

Update apt package index:

$ sudo apt-get update

```
kunal@kunalsh:~$ sudo apt-get update
Hit:1 https://dl.google.com/linux/chrome/deb stable InRelease
Hit:2 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:4 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:5 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:6 https://packagecloud.io/slacktechnologies/slack/debian jessie InRelease
Hit:7 https://download.docker.com/linux/ubuntu jammy InRelease
```

Install Latest version of Docker

$ sudo apt-get install docker-ce

```
kunal@kunalsh:~$ sudo apt-get install docker-ce
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite
The following NEW packages will be installed:
  docker-ce
```

#Installation Check

$ sudo docker --version

```
kunal@kunalsh:~$ sudo docker --version
Docker version 24.0.7, build afdd53b
```

container creation

$ sudo docker run - -name guru -it ubuntu /bin/bash

```
kunal@kunalsh:~$ sudo docker run  --name kunnu -it ubuntu /bin/bash
root@676187e6f113:/# ls -a
.  ..  .dockerenv  bin  boot  dev  etc  home  lib  lib32  lib64  lib
root@676187e6f113:/# cd home/
root@676187e6f113:/home# ls -a
.  ..
root@676187e6f113:/home# cd .
root@676187e6f113:/home# ls -a
.  ..
root@676187e6f113:/home# cd
root@676187e6f113:~#
exit
kunal@kunalsh:~$ sudo docker start kunnu
kunnu
kunal@kunalsh:~$ sudo docker attach Kunnu
Error response from daemon: No such container: Kunnu
kunal@kunalsh:~$ sudo docker attach kunnu
root@676187e6f113:/#
```