Physical servers: (20 years back)

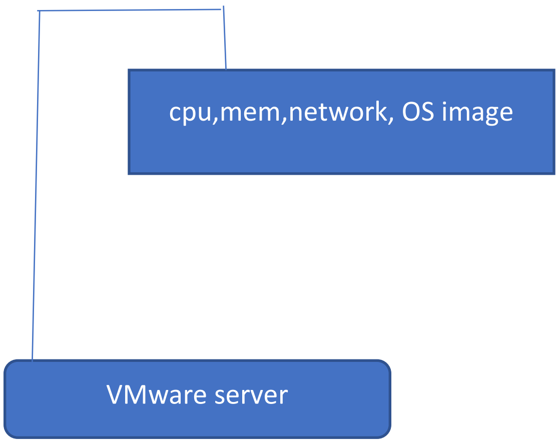
- In beginning days, we maintain physical servers.
- The cost is high.
- Based in requirement we maintain physical servers.

Server1	server2	server3	server4
DB	prod1	prod2	backup srvr

Later they introduced VMWARE.

VMWARE server:

- Virtualization technology.
- Other virtualization technologies_are windows Hyper-V, Linux Kum, citrixzenserver (aws) -> Free.
- VMware is not free.
- In physical servers(windows,lappy) we install VMware servers and in that we create virtual machines.
- There we install virtual servers.
- In VMware we have to assign memory, processor, hard disk and network



Docker: -

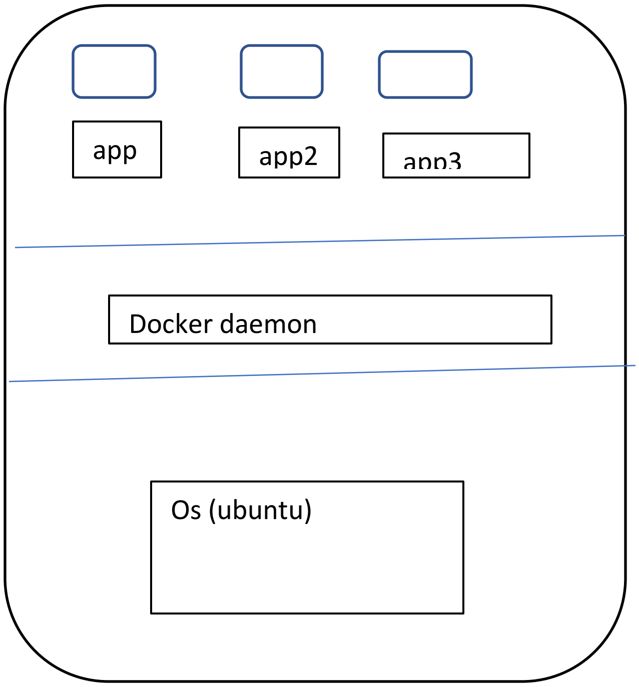
Docker is an open platform for developing, shipping, and running applications.
Docker enables you to separate your applications from your infrastructure so
you can deliver software quickly. With Docker, you can manage your infrastructure
in the same ways you manage your applications. By taking advantage of Docker's
methodologies for shipping, testing, and deploying code quickly, you can
significantly reduce the delay between writing code and running it in production.
It is container orchestration tool.

In Docker, you can package an application into a standardized unit,
which can be used for software development. This unit, or container,
includes the application's coding and dependencies so that it can easily
run on any computing environment. Before Docker, companies
often used Virtual Machines (VMs) to run applications.

Container: -

A container is a sandboxed process on your machine that is isolated from all other processes on the host machine. That isolation leverages kernel namespaces and cgroups features that have been in Linux for a long time. Docker has worked to make these capabilities approachable and easy to use. To summarize, a container:

- Is a runnable instance of an image. You can create, start, stop, move, or delete a container using the DockerAPI or CLI.
- Can be run on local machines, virtual machines or deployed to the cloud.
- Is portable (can be run on any OS).
- Is isolated from other containers and runs its own software, binaries, and configurations.



container image: -

When running a container, it uses an isolated filesystem. This custom filesystem is provided by a container image. Since the image contains the container's filesystem, it must contain everything needed to run an application - all dependencies, configurations, scripts, binaries, etc. The image also contains other configuration for the container, such as environment variables, a default command to run, and other metadata.

docker commands: -

1) docker –version or docker --version: -

This command is used to get the currently installed version of docker

[] docker –version

docker version 23.0.1 build

2) docker images:-

This command lists all the locally stored docker images

```
   Repo          tag           imageID        created        size
Sampleapp      latest       5h76gtygf679    2hrs ago       120mb
```

3) docker pull :-

This command is used to pull images from the docker repository(hub.docker.com)

[] docker pull <image name>

[]docker pull dockerhub_uname/sampleapp:v1

4) docker run : -

This command is used to create a container from an image

[] docker run -it -d <image name>

[]docker run –rm –dit –name sampleapp –hostname sampleapphost –p 8090:80 sampleapp:latest /bin/bash

5) docker ps :-

This command is used to list the running containers

[] docker ps

```
 container id   name   command   created     status     port              name
 D3erffg        juhfh  docker    5min  ago   up 10min  0.0.0.8080>80      sampleapp
```

6) docker ps –a :-

This command is used to show all the running and exited containers

[] docker ps -a

```
 container id   name   command   created     status
 D3erffg        juhfh  docker    5min  ago   exited ( 0)
```

7) docker exec: -

This command is used to access the running container

```
[] docker exec -it <container id> /bin/bash
[] docker exec –it b4325etdrtgdr /bin/bash
```

8) docker stop :-

This command stops a running container

```
[] docker stop <container id>
[] docker stop b4325etdrtgdr
```

9) docker kill :-

This command kills the container by stopping its execution immediately.

The difference between ‘docker kill’ and ‘docker stop’ is that ‘docker stop’ gives the container time to shutdown gracefully, in situations when it is taking too much time for getting the container to stop, one can opt to kill it

```
[] docker kill <container id>
[] docker kill b4325etdrtgdr
```

10) docker commit :-

This command creates a new image of an edited container on the local system

```
[] docker commit <conatainer id> <username/imagename>
[] docker commit  d13jhueh455      username/sampleapp:v1
```

11) docker login

This command is used to login to the docker hub repository

```
 [] docker login
Username:kanchana
Password:*******
```

12) docker push :-

This command is used to push an image to the docker hub repository

```
[] docker push <username/image name>
[] docker push username/sampleapp:v1
```

13) docker rm :-

This command used to delete a stopped container forcible

```
[] docker rm  -f<container id>
 [] docker rm  -f d6cjhuihfjkf
```

14) docker rmi :-

This command is used to delete an image from local storage

```
[] docker rmi <image name>
[] docker rmi d6h394879
```

This command used to delete images forcible

```
[] docker rmi –f hihh
```

15) vi (or) vim Dockerfile :-

This command is used to create docker file

```
[] vi Dockerfile
```

16) docker build:-

This command is used to build an image from a specified docker file

[] docker build -t <path to docker file>

[] docker build –t name/sampleapp:v1

17)docker system prune –a

This command used to delete unused images and container

[] docker system prune –a

18)docker rmi $(docker images –aq)

This command is used to delete all image from local storage

[] docker rmi $(docker images –aq)

19) docker stop $(docker ps –aq)

This command is used to delete all container from local storage

[] docker stop $(docker ps –aq)

20)docker pause container id

This command is used to pause one particular container

[] docker pause container id

21) Docker unpause container id

This command is used to unpause paused container

[] Docker unpause container id