Docker :

Container: the way to package applications the all the necessary dependencies and configuration

That package is easy to share and move

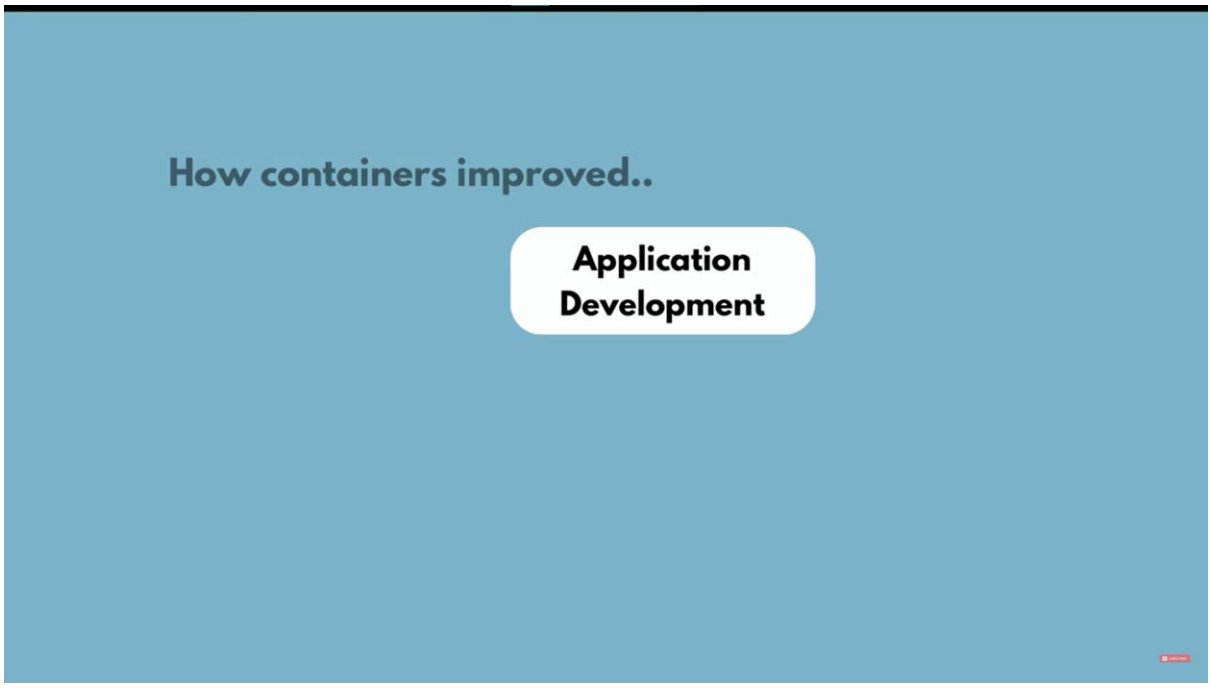
Every package is in isolated environment

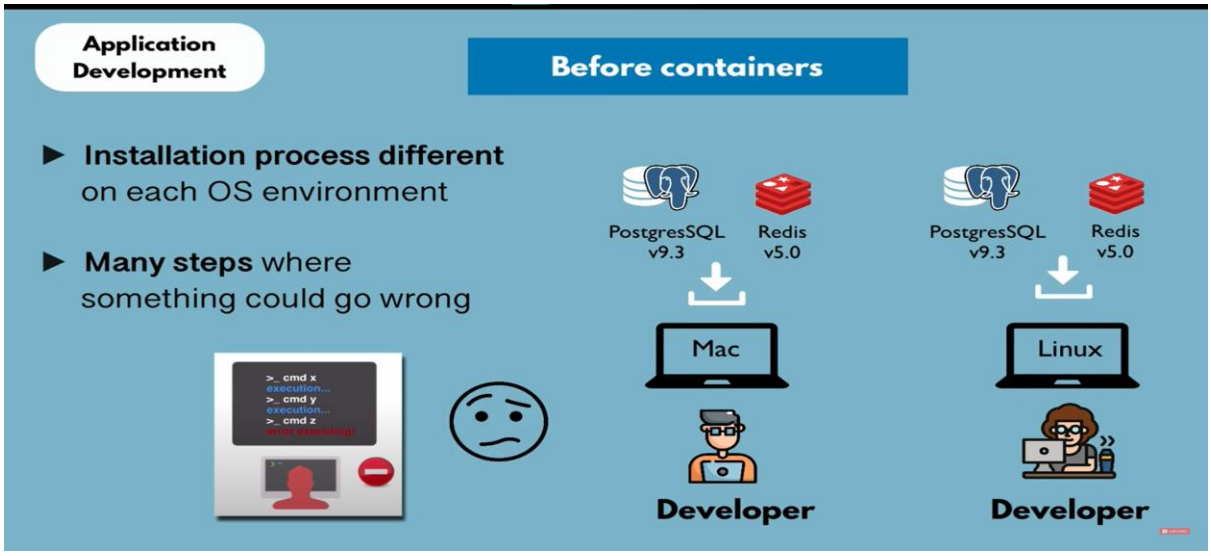
Containers make the development and deployment more efficient

The containers are present in the container repository (container repository is the storage for the containers)

Private companies store their repositeeries in the private repositories

There is public repository for docker is (DOCKERHUB)

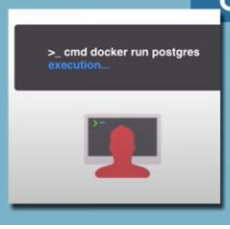


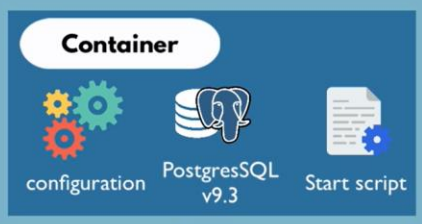


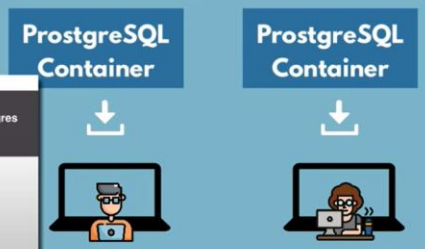
Application Development

After containers

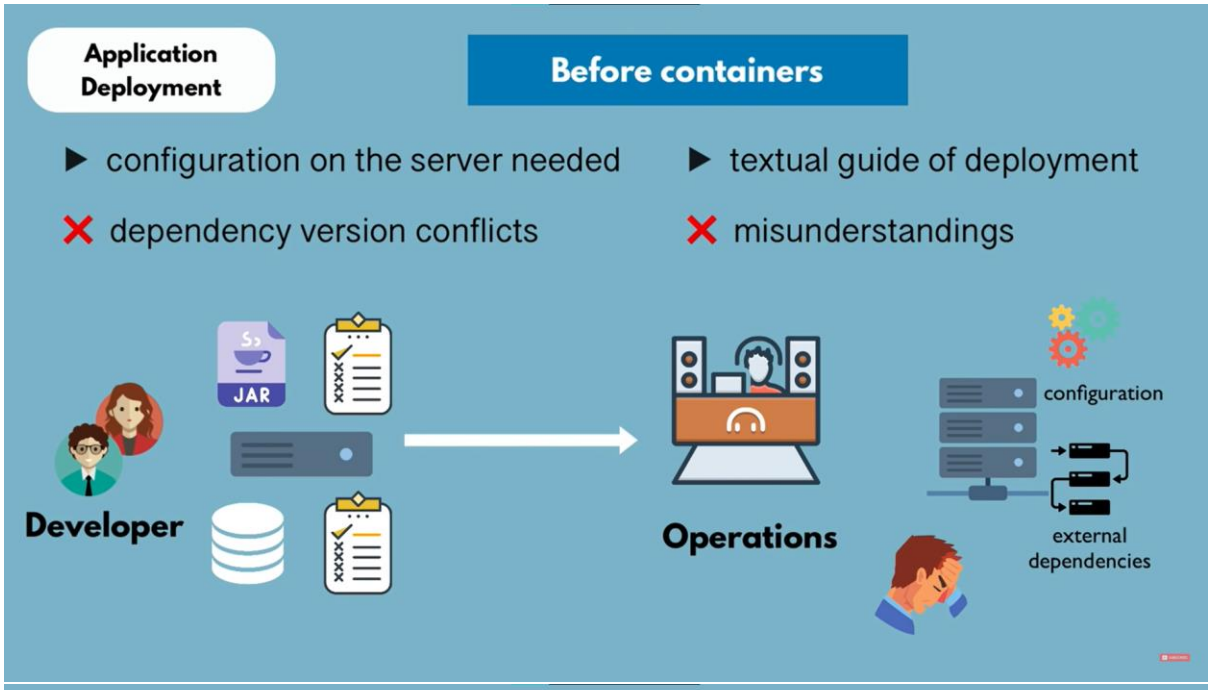
- own isolated environment
- packaged with all needed configuration
- one command to install the app
- run same app with 2 different versions

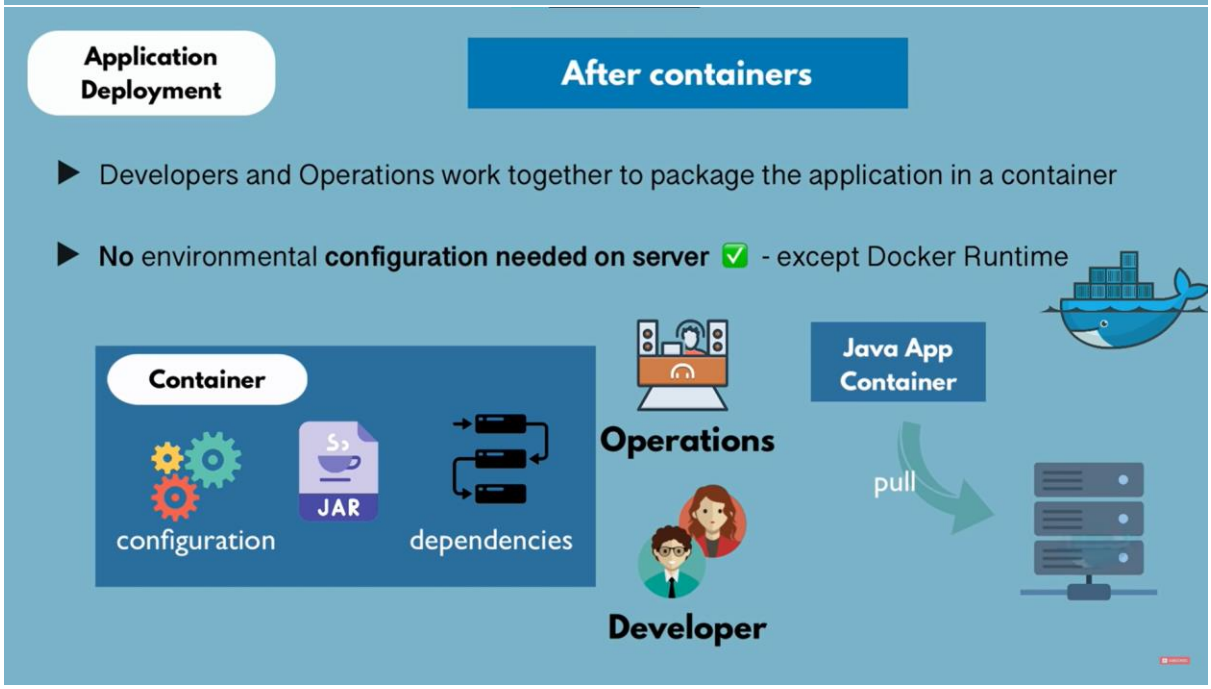






Application Deployment





Container ➔layers of images

Application image on top

Os based image in base

Docker is a containerization platform that allows developers to package and run applications in a portable and isolated environment. Here are some important terminologies in Docker that you should know:

Docker Image:

A Docker image is a lightweight, standalone, executable package that includes everything needed to run an application, including the code, libraries, dependencies, and runtime. Images can be built from a Dockerfile, or they can be pulled from a public or private registry.

Example: You can create a Docker image of a Python application by creating a Dockerfile that includes the Python runtime and any dependencies, and then building the image with the docker build command.

Docker Container:

A Docker container is a running instance of a Docker image. Containers are isolated environments that contain all the dependencies and runtime needed to run the application. Containers can be started, stopped, and deleted with simple Docker commands.

Example: You can start a Docker container from a Python image by running the docker run command with the image name.

Dockerfile:

A Dockerfile is a script that contains instructions to build a Docker image. It specifies the base image, the software dependencies, and the commands needed to install and configure the application.

Example: Here is an example Dockerfile for a simple Python application:

sql

Copy code

```
FROM python:3.8

WORKDIR /app

COPY . /app

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

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

Docker Registry:

A Docker registry is a centralized location where Docker images are stored and distributed. There are public registries, such as Docker Hub, where users can share and download images, and private registries, which are used by organizations to store and distribute their own images.

Example: You can push your Docker image to Docker Hub by running the docker push command with your Docker Hub username and image name.

Docker Compose:

Docker Compose is a tool for defining and running multi-container Docker applications. It allows you to define the services that make up your application, the networks they are connected to, and the volumes they share.

Example: Here is an example docker-compose.yml file for a simple web application with a database:

yaml

Copy code

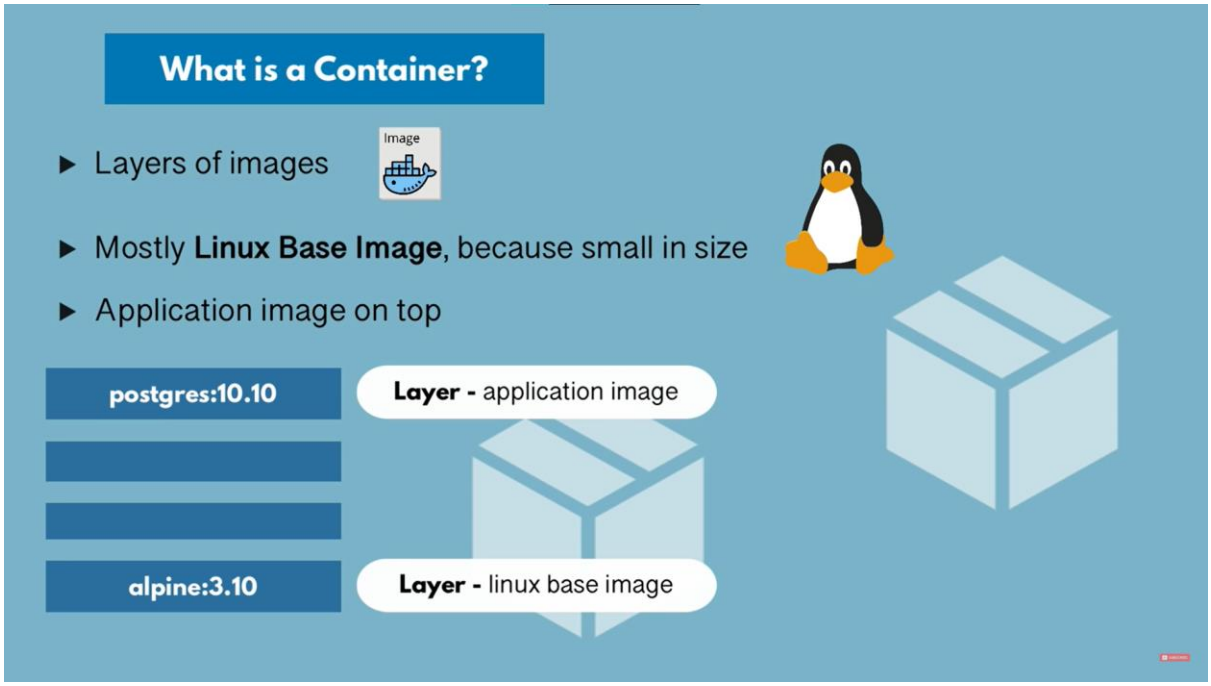
```
version: '3'

services:

  web:

    build: .

    ports:

      - "5000:5000"

  db:

    image: mysql:5.7

    environment:

      MYSQL_ROOT_PASSWORD: example
```

Docker Swarm:

Docker Swarm is a clustering and orchestration tool for Docker. It allows you to create and manage a cluster of Docker nodes, and deploy and scale services across the cluster.

Example: You can create a Docker Swarm cluster by initializing a swarm with the docker swarm init command on one node, and then joining other nodes to the swarm with the docker swarm join command.

In conclusion, understanding these important Docker terminologies is essential for anyone who wants to use Docker for containerization. By learning these terms and concepts, you can better understand how Docker works and how to use it to deploy and manage your applications.

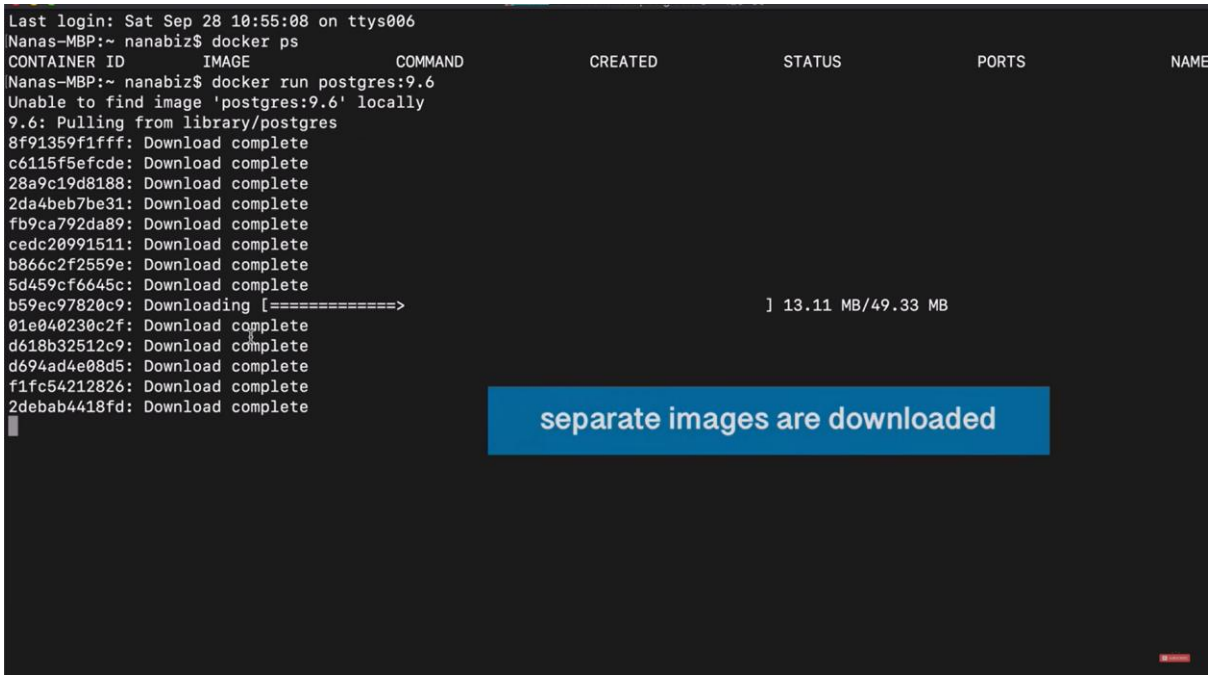


docker pull command is used to download packages from the internet.

Docker images command is used to list the docker images that are on your computer

Docker run command used to create running containers from images and run the commands inside them.(when I use the run if the container is not installed means it automatically download the container latter it will runcthe container)

$docker run postgres:9.6 #specifing the version



Docker ps command used to see the running containers

Docker Image

- the actual package

Image

configuration | PostgresSQL v9.3 | Start script

- **artifact**, that can be moved around

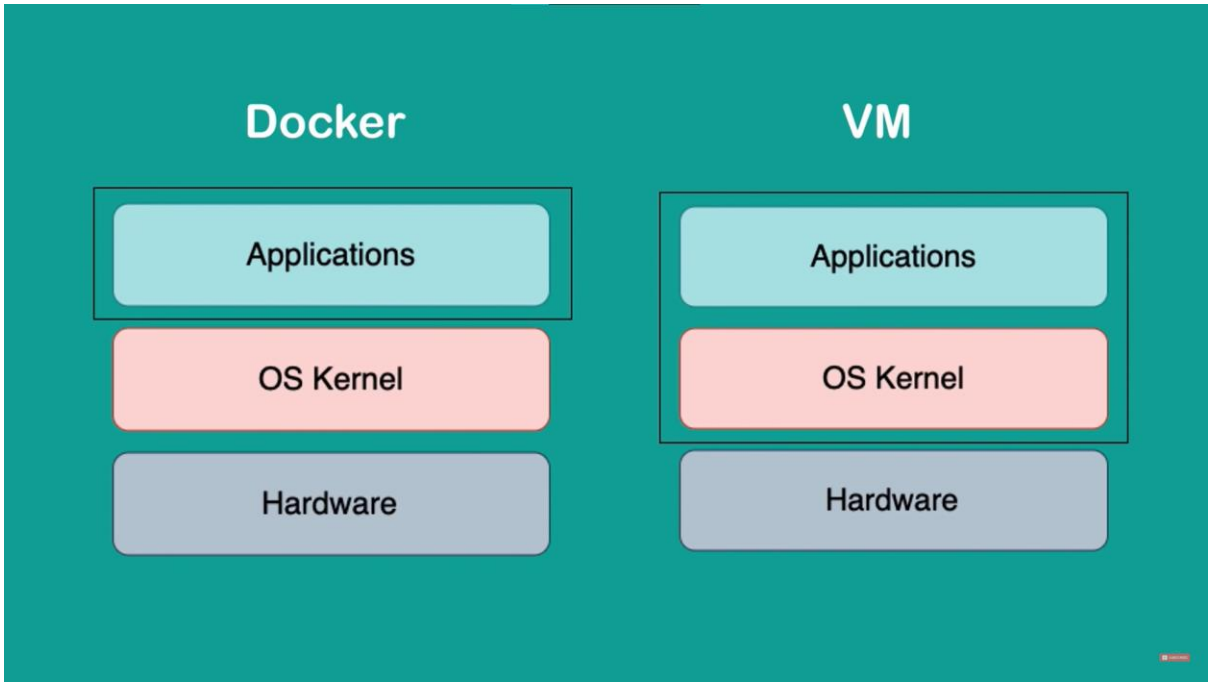
not running

Docker Container

- actually **start the application**
- container environment is created

running

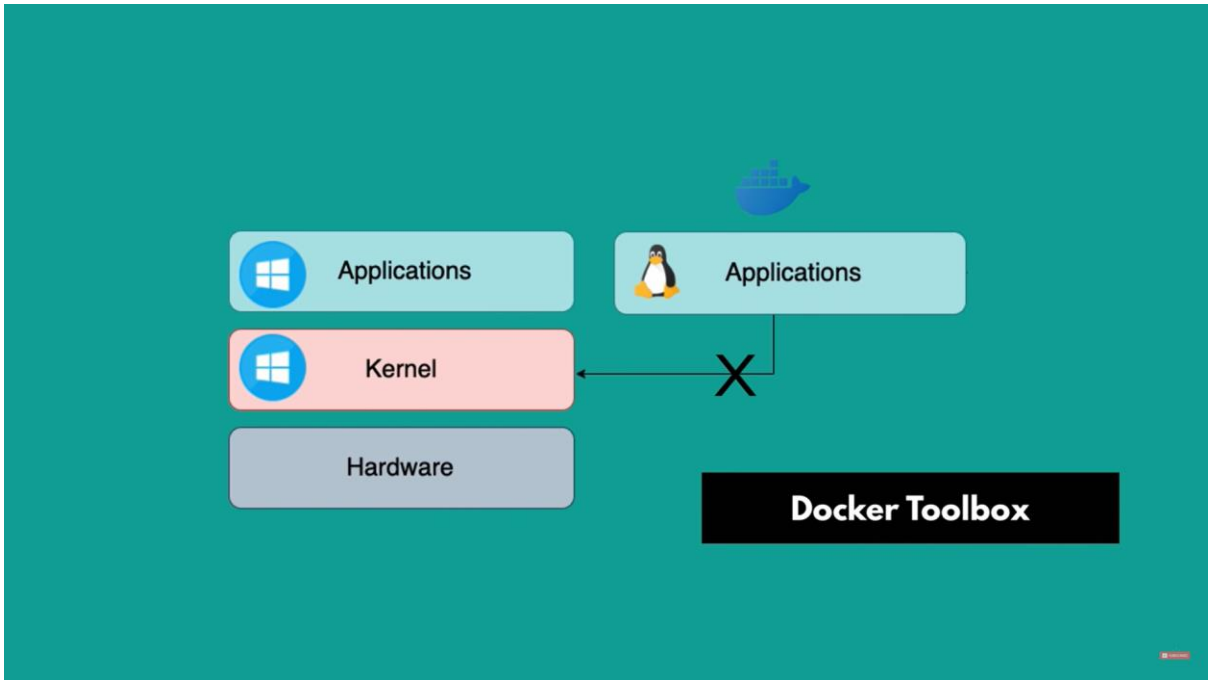
```
~ — docker run postgres:9.6                                  ~ — -bash
Last login: Sat Sep 28 10:57:20 on ttys001
Nanas-MBP:~ nanabiz$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
S
fad0f8456ca7        postgres:9.6        "docker-entrypoint..."   45 seconds ago      Up 47 seconds       5432/tcp
eless_haibt
Nanas-MBP:~ nanabiz$
```

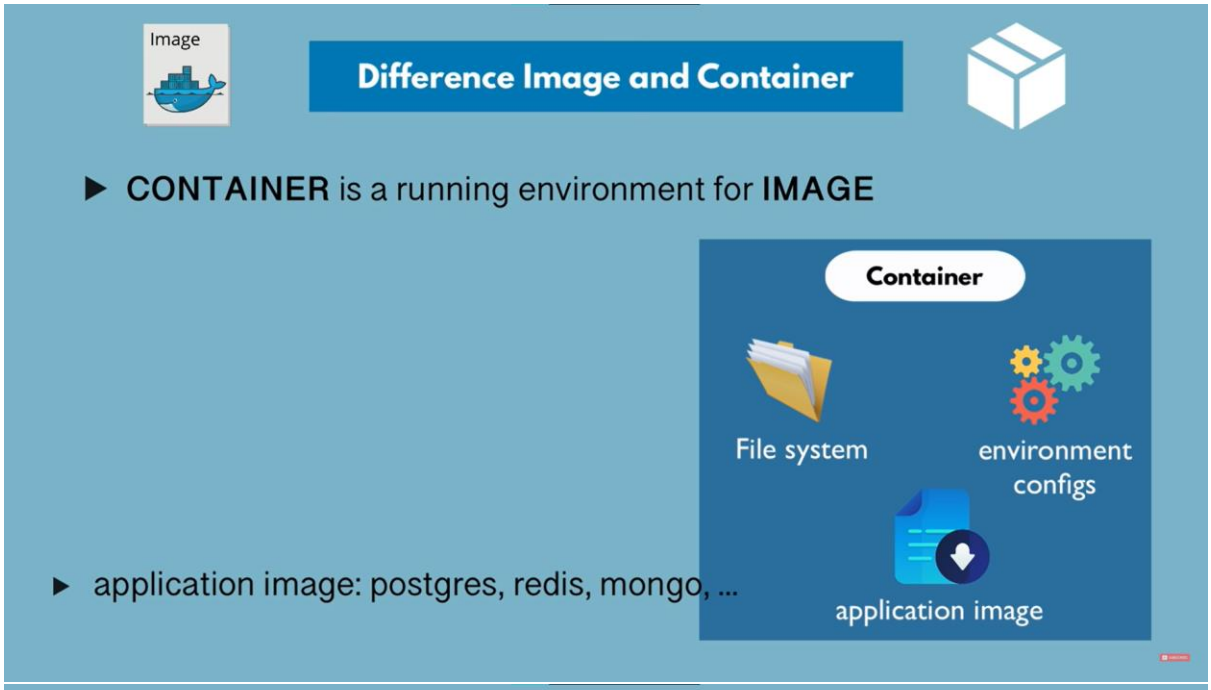


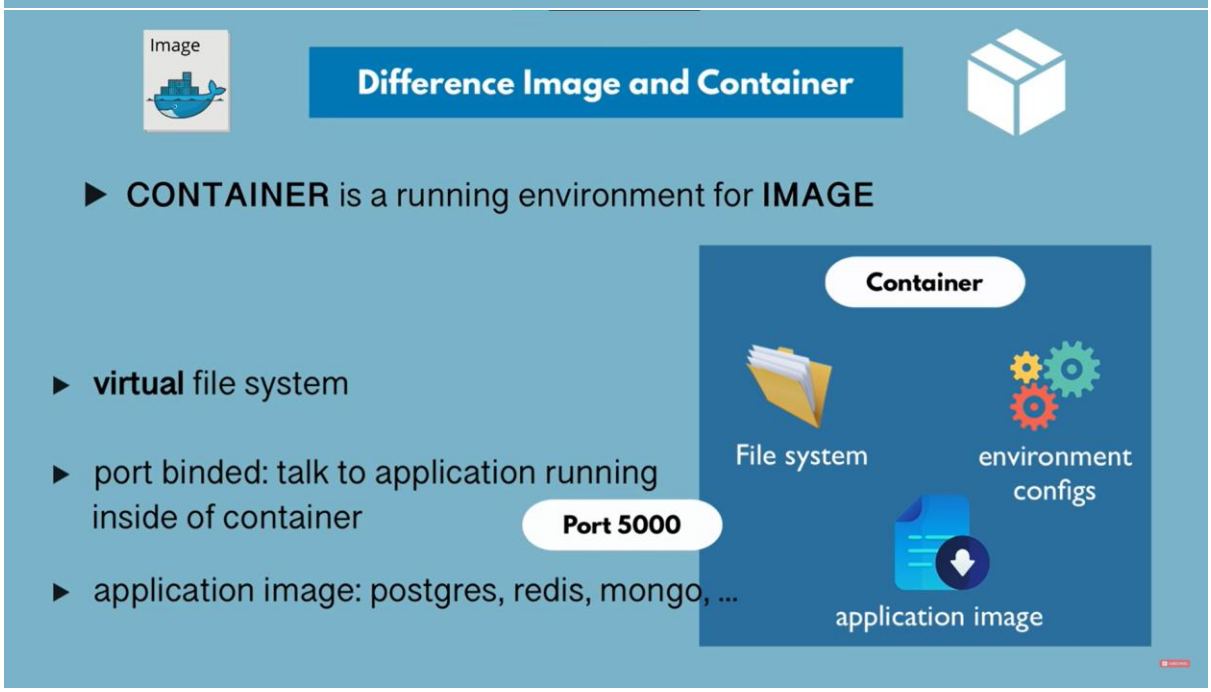
Size of the docker image is smalle

Speed:Docker container start and run much fast

Vm on any os can run on any os







To stop the ccontainer

Docker stop container_id command to stop the docker container

To start the stoppe ddcontainer

Docker start container_id command to start the docker container

Docker ps command used to see the running containers at now

Contanerisation →packing the software code and its dependencies

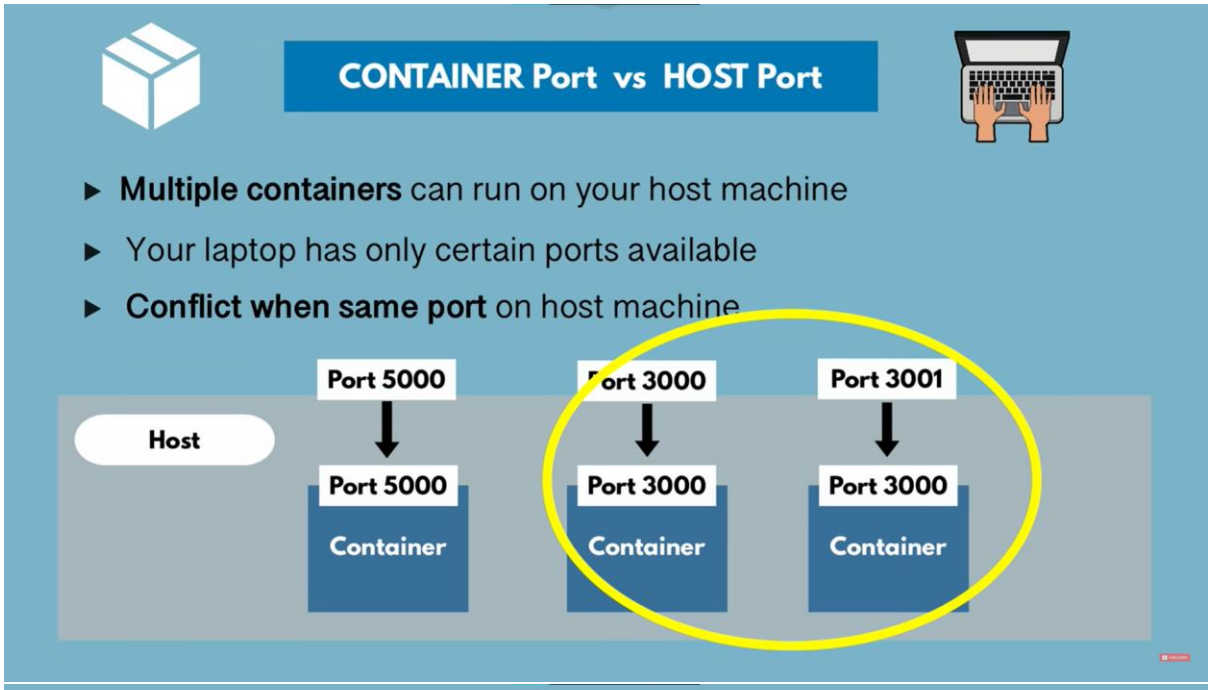
PYTHON LIFE

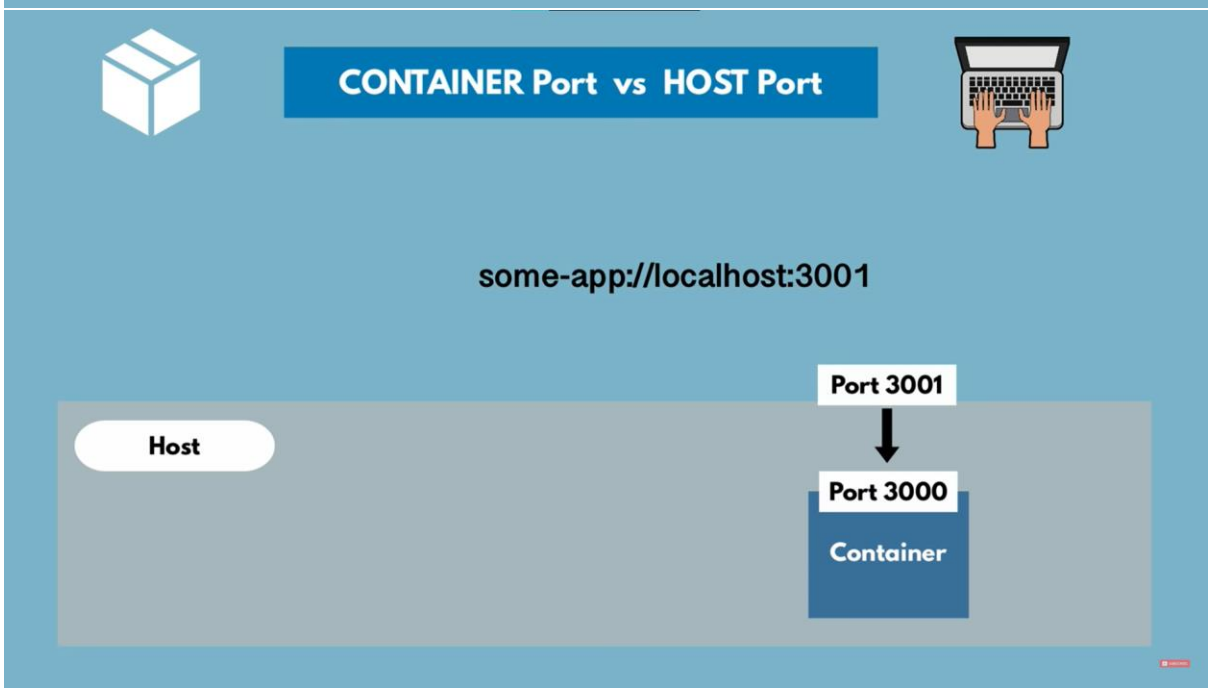
Docker client: is the primary way that many docker users interact with Docker.

Docker Host: Docker host is the machine where you installed the docker image.

Docker Demon: Docker demon runs on the host OS . it is responsible for running containers to manage docker services.

Docker ps -a command is used to see the containers list running and stopped container





Assigning the new port in container

```
1:C 26 Oct 12:06:04.710 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo
1:C 26 Oct 12:06:04.710 # Redis version=4.0.14, bits=64, commit=00000000, modified=0, pid=1, just started
1:C 26 Oct 12:06:04.710 # Warning: no config file specified, using the default config. In order to specify a config file use
1:M 26 Oct 12:06:04.712 * Running mode=standalone, port=6379.
1:M 26 Oct 12:06:04.712 # WARNING: The TCP backlog setting of 511 cannot be enforced because /proc/sys/net/core/somaxconn is
1:M 26 Oct 12:06:04.712 # Server initialized
1:M 26 Oct 12:06:04.712 # WARNING you have Transparent Huge Pages (THP) support enabled in your kernel. This will create lat
Redis. To fix this issue run the command 'echo never > /sys/kernel/mm/transparent_hugepage/enabled' as root, and add it to y
in the setting after a reboot. Redis must be restarted after THP is disabled.
1:M 26 Oct 12:06:04.712 * Ready to accept connections
^C1:signal-handler (1572091897) Received SIGINT scheduling shutdown...
1:M 26 Oct 12:11:37.974 # User requested shutdown...
1:M 26 Oct 12:11:37.974 * Saving the final RDB snapshot before exiting.
1:M 26 Oct 12:11:37.977 * DB saved on disk
1:M 26 Oct 12:11:37.977 # Redis is now ready to exit, bye bye...
[~]$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
8381867e8242        redis               "docker-entrypoint..."   21 minutes ago      Up 15 minutes       6379/tcp
[~]$ docker stop 8381867e8242
8381867e8242
[~]$ docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS                      PORTS
cfec85d7bbec        redis:4.0           "docker-entrypoint..."   5 minutes ago       Exited (0) 14 seconds ago
8381867e8242        redis               "docker-entrypoint..."   21 minutes ago      Exited (0) 2 seconds ago
6ef8fbacce73        postgres:9.6.5      "docker-entrypoint..."   8 months ago        Exited (0) 3 months ago
_1
[~]$ docker run -p6000:6379
"docker run" requires at least 1 argument(s).
See 'docker run --help'.

Usage:  docker run [OPTIONS] IMAGE [COMMAND] [ARG...]

Run a command in a new container
[~]$ docker run -p6000:6379
```

```
Terminal  Shell  Edit  View  Window  Help                    11 %   Sat 14:13  Nana Janashia
~ — -bash — 158×41
~ — docker run -p6000:6379 redis        ~ — -bash
[~]$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS                    NAMES
6f691dc1cf7f        redis               "docker-entrypoint..."   7 seconds ago       Up 9 seconds        0.0.0.0:6000->6379/tcp   blissful_curran
[~]$
```

To create a container we need image

To check the docker version →docker –version

To check the indetail docker version like api,go version –etc → docker version

To check the docker information→docker info

To see the all the images on the host system →Docker images

To install the images from docker images from docker hub →docker pull image_name

The docker run →docker run image_name

The docker run first chek the wether the image is existes locally or not if not it will first pull the imahe from docker hub and create a running containers from that image.

Docker run → bt default the container is ccreated but the container is stopped

Docker ps ➔ show or list the containers which are stated

Docker ps -a →show or list the containers which are started and also stopped

Docker ps -s →show the size of the filwes

Docker ps -q →only show the container ids

Docker ps -l →only show the latest container

Docker ps -n num → show the latest num containers

To search the inages on the docker hub fron the docker command line →Docker search image_name

To run image → docker rum image name → but this start the container and ot stop in just milli second

To use the ontainer in the interactive way (this takes the input from command line)(-I interactive)

Docker run -i _image_name

To allocate the pseudo-terminal tty terminal to the container (-t terminal mode) but this will not take input

Docker run -t ubuntu

To use like interactive and with terminal we need to combine the I and t in this

Docker run -it ubunti → run like normal termina

```
PS C:\Users\pavan> docker run -it ubuntu
root@c226ee95a3bb:/# ls
bin  boot  dev  etc  home  lib  lib32  lib64  libx32  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@c226ee95a3bb:/# ls -al
total 56
drwxr-xr-x   1 root root 4096 Apr  4 08:52 .
drwxr-xr-x   1 root root 4096 Apr  4 08:52 ..
-rwxr-xr-x   1 root root    0 Apr  4 08:52 .dockerenv
lrwxrwxrwx   1 root root    7 Mar  8 02:05 bin -> usr/bin
drwxr-xr-x   2 root root 4096 Apr 18  2022 boot
drwxr-xr-x   5 root root  360 Apr  4 08:52 dev
drwxr-xr-x   1 root root 4096 Apr  4 08:52 etc
drwxr-xr-x   2 root root 4096 Apr 18  2022 home
lrwxrwxrwx   1 root root    7 Mar  8 02:05 lib -> usr/lib
lrwxrwxrwx   1 root root    9 Mar  8 02:05 lib32 -> usr/lib32
lrwxrwxrwx   1 root root    9 Mar  8 02:05 lib64 -> usr/lib64
lrwxrwxrwx   1 root root   10 Mar  8 02:05 libx32 -> usr/libx32
drwxr-xr-x   2 root root 4096 Mar  8 02:05 media
drwxr-xr-x   2 root root 4096 Mar  8 02:05 mnt
drwxr-xr-x   2 root root 4096 Mar  8 02:05 opt
dr-xr-xr-x 265 root root    0 Apr  4 08:52 proc
drwx------   2 root root 4096 Mar  8 02:08 root
drwxr-xr-x   5 root root 4096 Mar  8 02:08 run
lrwxrwxrwx   1 root root    8 Mar  8 02:05 sbin -> usr/sbin
drwxr-xr-x   2 root root 4096 Mar  8 02:05 srv
dr-xr-xr-x  11 root root    0 Apr  4 08:52 sys
drwxrwxrwt   2 root root 4096 Mar  8 02:08 tmp
drwxr-xr-x  14 root root 4096 Mar  8 02:05 usr
drwxr-xr-x  11 root root 4096 Mar  8 02:08 var
root@c226ee95a3bb:/# cd
root@c226ee95a3bb:~#
```

To create the container with the custom name

Docker run –name container_name image_name

```
PS C:\Users\pavan> docker ps
CONTAINER ID   IMAGE     COMMAND       CREATED         STATUS         PORTS     NAMES
c5d0137a2a11   ubuntu    "/bin/bash"   4 minutes ago   Up 4 minutes             sad_keller
PS C:\Users\pavan> docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED         STATUS                     PORTS     NAMES
c226ee95a3bb   ubuntu     "/bin/bash"              2 minutes ago   Exited (0) 19 seconds ago            magical_poitras
c5d0137a2a11   ubuntu     "/bin/bash"              5 minutes ago   Up 5 minutes                         sad_keller
282be899b966   ubuntu     "/bin/bash"              9 minutes ago   Exited (127) 5 minutes ago           goofy_lamporta85aa5d20253   ubuntu     "/bin/bash"             9 minutes ago    Exited (0) 9 minutes ago
ompson
ec1c0f66228c   redis      "docker-entrypoint.s…"   19 minutes ago  Exited (0) 9 minutes ago             amazing_rhodes
229674c1f40e   postgres   "docker-entrypoint.s…"   6 days ago      Exited (1) 6 days ago                boring_keldysh
95a838a5395f   postgres   "docker-entrypoint.s…"   6 days ago      Exited (1) 6 days ago                mystifying_engelbart
5e7165e629b1   python     "python3"                6 days ago      Exited (0) 6 days ago                sweet_wright
PS C:\Users\pavan> docker run -name ubuntu
unknown shorthand flag: 'n' in -name
See 'docker run --help'.
PS C:\Users\pavan> docker run --name ubuntu
"docker run" requires at least 1 argument.
See 'docker run --help'.

Usage:  docker run [OPTIONS] IMAGE [COMMAND] [ARG...]

Run a command in a new container
PS C:\Users\pavan> docker run --name pavan_container ubuntu
PS C:\Users\pavan> docker ps
CONTAINER ID   IMAGE     COMMAND       CREATED         STATUS         PORTS     NAMES
c5d0137a2a11   ubuntu    "/bin/bash"   6 minutes ago   Up 6 minutes             sad_keller
PS C:\Users\pavan> docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED         STATUS                     PORTS     NAMES
19e48be52002   ubuntu     "/bin/bash"              12 seconds ago  Exited (0) 11 seconds ago            pavan_container
c226ee95a3bb   ubuntu     "/bin/bash"              3 minutes ago   Exited (0) About a minute ago        magical_poitras
c5d0137a2a11   ubuntu     "/bin/bash"              6 minutes ago   Up 6 minutes                         sad_keller
282be899b966   ubuntu     "/bin/bash"              10 minutes ago  Exited (127) 6 minutes ago           goofy_lamport
a85aa5d20253   ubuntu     "/bin/bash"              10 minutes ago  Exited (0) 10 minutes ago            elastic_thompson
ec1c0f66228c   redis      "docker-entrypoint.s…"   21 minutes ago  Exited (0) 11 minutes ago            amazing_rhodes
229674c1f40e   postgres   "docker-entrypoint.s…"   6 days ago      Exited (1) 6 days ago                boring_keldysh
95a838a5395f   postgres   "docker-entrypoint.s…"   6 days ago      Exited (1) 6 days ago                mystifying_engelbart
5e7165e629b1   python     "python3"                6 days ago      Exited (0) 6 days ago                sweet_wright
PS C:\Users\pavan>
```

Changing the container command like /bin/bash or /bin/sh etc

Docker run -it –name ubuntu_container_bin_bash ubuntu /bin/bash

```
PS C:\Users\pavan> docker run -it --name pavans_container_bin_bash ubuntu /bin/bash
root@01c0464e43be:/# ls
bin  boot  dev  etc  home  lib  lib32  lib64  libx32  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@01c0464e43be:/# exit
exit
PS C:\Users\pavan> docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED          STATUS                      PORTS     NAMES
01c0464e43be   ubuntu     "/bin/bash"              13 seconds ago   Exited (0) 7 seconds ago              pavans_container_bin_bash
19e48be52002   ubuntu     "/bin/bash"              5 minutes ago    Exited (0) 5 minutes ago              pavan_container
c226ee95a3bb   ubuntu     "/bin/bash"              8 minutes ago    Exited (0) 6 minutes ago              magical_poitras
c5d0137a2a11   ubuntu     "/bin/bash"              11 minutes ago   Up 11 minutes                         sad_keller
282be899b966   ubuntu     "/bin/bash"              15 minutes ago   Exited (127) 11 minutes ago           goofy_lamport
a85aa5d20253   ubuntu     "/bin/bash"              15 minutes ago   Exited (0) 15 minutes ago             elastic_thompson
ec1c0f66228c   redis      "docker-entrypoint.s…"   25 minutes ago   Exited (0) 16 minutes ago             amazing_rhodes
229674c1f40e   postgres   "docker-entrypoint.s…"   6 days ago       Exited (1) 6 days ago                 boring_keldysh
95a838a5395f   postgres   "docker-entrypoint.s…"   6 days ago       Exited (1) 6 days ago                 mystifying_engelbart
5e7165e629b1   python     "python3"                6 days ago       Exited (0) 6 days ago                 sweet_wright
PS C:\Users\pavan> docker run -it --name pavans_container_bin_sh ubuntu /bin/sh
#
# ls
bin  boot  dev  etc  home  lib  lib32  lib64  libx32  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
# exit
PS C:\Users\pavan> docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED          STATUS                      PORTS     NAMES
75de9c15457d   ubuntu     "/bin/sh"                10 seconds ago   Exited (0) 4 seconds ago              pavans_container_bin_sh
01c0464e43be   ubuntu     "/bin/bash"              59 seconds ago   Exited (0) 53 seconds ago             pavans_container_bin_bash
19e48be52002   ubuntu     "/bin/bash"              5 minutes ago    Exited (0) 5 minutes ago              pavan_container
c226ee95a3bb   ubuntu     "/bin/bash"              9 minutes ago    Exited (0) 7 minutes ago              magical_poitras
c5d0137a2a11   ubuntu     "/bin/bash"              11 minutes ago   Up 11 minutes                         sad_keller
282be899b966   ubuntu     "/bin/bash"              15 minutes ago   Exited (127) 12 minutes ago           goofy_lamport
a85aa5d20253   ubuntu     "/bin/bash"              16 minutes ago   Exited (0) 16 minutes ago             elastic_thompson
ec1c0f66228c   redis      "docker-entrypoint.s…"   26 minutes ago   Exited (0) 16 minutes ago             amazing_rhodes
229674c1f40e   postgres   "docker-entrypoint.s…"   6 days ago       Exited (1) 6 days ago                 boring_keldysh
95a838a5395f   postgres   "docker-entrypoint.s…"   6 days ago       Exited (1) 6 days ago                 mystifying_engelbart
5e7165e629b1   python     "python3"                6 days ago       Exited (0) 6 days ago                 sweet_wright
PS C:\Users\pavan>
```

To start the container →docker start container_name

To stop the container → docker stop container_name

To go inside the container(before attaching you need to start the container) → docker attach container_name

Docker rm container_name →To remove the container

BUILD A IMAGE FROM CONTAINER :

Step-1: create a container from image

```
PS C:\Users\pavan> docker images
REPOSITORY   TAG       IMAGE ID       CREATED       SIZE
postgres     latest    80c558ffdc31   7 days ago    379MB
python       latest    df3e9d105d6c   12 days ago   921MB
redis        latest    31f08b90668e   12 days ago   117MB
ubuntu       latest    08d22c0ceb15   3 weeks ago   77.8MB
PS C:\Users\pavan> docker run -it --name  container_1 ubuntu /bin/bash
root@e03c691b9b1a:/# ls
bin  boot  dev  etc  home  lib  lib32  lib64  libx32  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@e03c691b9b1a:/# ls -al
total 56
drwxr-xr-x   1 root root 4096 Apr  4 12:43 .
drwxr-xr-x   1 root root 4096 Apr  4 12:43 ..
-rwxr-xr-x   1 root root    0 Apr  4 12:43 .dockerenv
lrwxrwxrwx   1 root root    7 Mar  8 02:05 bin -> usr/bin
drwxr-xr-x   2 root root 4096 Apr 18  2022 boot
drwxr-xr-x   5 root root  360 Apr  4 12:43 dev
drwxr-xr-x   1 root root 4096 Apr  4 12:43 etc
drwxr-xr-x   2 root root 4096 Apr 18  2022 home
lrwxrwxrwx   1 root root    7 Mar  8 02:05 lib -> usr/lib
lrwxrwxrwx   1 root root    9 Mar  8 02:05 lib32 -> usr/lib32
lrwxrwxrwx   1 root root    9 Mar  8 02:05 lib64 -> usr/lib64
lrwxrwxrwx   1 root root   10 Mar  8 02:05 libx32 -> usr/libx32
drwxr-xr-x   2 root root 4096 Mar  8 02:05 media
drwxr-xr-x   2 root root 4096 Mar  8 02:05 mnt
drwxr-xr-x   2 root root 4096 Mar  8 02:05 opt
dr-xr-xr-x 261 root root    0 Apr  4 12:43 proc
drwx------   2 root root 4096 Mar  8 02:08 root
drwxr-xr-x   5 root root 4096 Mar  8 02:08 run
lrwxrwxrwx   1 root root    8 Mar  8 02:05 sbin -> usr/sbin
drwxr-xr-x   2 root root 4096 Mar  8 02:05 srv
dr-xr-xr-x  11 root root    0 Apr  4 12:43 sys
drwxrwxrwt   2 root root 4096 Mar  8 02:08 tmp
drwxr-xr-x  14 root root 4096 Mar  8 02:05 usr
drwxr-xr-x  11 root root 4096 Mar  8 02:08 var
root@e03c691b9b1a:/# cd tmp/
root@e03c691b9b1a:/tmp# ls
root@e03c691b9b1a:/tmp# ll
total 8
drwxrwxrwt 2 root root 4096 Mar  8 02:08 ./
drwxr-xr-x 1 root root 4096 Apr  4 12:43 ../
root@e03c691b9b1a:/tmp# mkdir file1
root@e03c691b9b1a:/tmp# ls
file1
root@e03c691b9b1a:/tmp# ls a-l
ls: cannot access 'a-l': No such file or directory
root@e03c691b9b1a:/tmp# ls -al
total 12
drwxrwxrwt 1 root root 4096 Apr  4 12:44 .
drwxr-xr-x 1 root root 4096 Apr  4 12:43 ..
drwxr-xr-x 2 root root 4096 Apr  4 12:44 file1
root@e03c691b9b1a:/tmp# cd file1/
root@e03c691b9b1a:/tmp/file1# echo "this the file created by the pavan ram chandar" >> man
root@e03c691b9b1a:/tmp/file1# ls
man
root@e03c691b9b1a:/tmp/file1# cat man
this the file created by the pavan ram chandar
root@e03c691b9b1a:/tmp/file1#
```

Step2: make the changes in that container like above I made the changes in the tep folder

Step 3:now create a container from this container by using

Docker commit container_name_already_present name_of_image_you_want_to_create

Step4: list the docker images

Step5:create the another container to verify the image

```
PS C:\Users\pavan> docker ps -a
CONTAINER ID   IMAGE     COMMAND       CREATED         STATUS                     PORTS     NAMES
e03c691b9b1a   ubuntu    "/bin/bash"   7 minutes ago   Exited (0) 4 minutes ago             container_1
PS C:\Users\pavan> docker commit container_1  ubuntu_pavan_custom
sha256:702c25213207bd44f03846fd73c29c47361a22a9690218a6db8f417c0586aa18
PS C:\Users\pavan> docker images
REPOSITORY            TAG       IMAGE ID       CREATED         SIZE
ubuntu_pavan_custom   latest    702c25213207   7 seconds ago   77.8MB
postgres              latest    80c558ffdc31   7 days ago      379MB
python                latest    df3e9d105d6c   12 days ago     921MB
redis                 latest    31f08b90668e   12 days ago     117MB
ubuntu                latest    08d22c0ceb15   3 weeks ago     77.8MB
PS C:\Users\pavan> docker images --help

Usage:  docker images [OPTIONS] [REPOSITORY[:TAG]]

List images

Options:
  -a, --all             Show all images (default hides intermediate images)
      --digests         Show digests
  -f, --filter filter   Filter output based on conditions provided
      --format string   Pretty-print images using a Go template
      --no-trunc        Don't truncate output
  -q, --quiet           Only show image IDs
PS C:\Users\pavan> docker images -a
REPOSITORY            TAG       IMAGE ID       CREATED          SIZE
ubuntu_pavan_custom   latest    702c25213207   39 seconds ago   77.8MB
postgres              latest    80c558ffdc31   7 days ago       379MB
python                latest    df3e9d105d6c   12 days ago      921MB
redis                 latest    31f08b90668e   12 days ago      117MB
ubuntu                latest    08d22c0ceb15   3 weeks ago      77.8MB
PS C:\Users\pavan> docker run -it  --name container_from_ubuntu_pavan_custom_image  702c25213207   /bin/bash
root@b3f5a2cbe49a:/# ls
bin  boot  dev  etc  home  lib  lib32  lib64  libx32  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@b3f5a2cbe49a:/# cd tmp/
root@b3f5a2cbe49a:/tmp# ls
file1
root@b3f5a2cbe49a:/tmp# ls -al
total 12
drwxrwxrwt 1 root root 4096 Apr  4 12:44 .
drwxr-xr-x 1 root root 4096 Apr  4 12:55 ..
drwxr-xr-x 2 root root 4096 Apr  4 12:45 file1
root@b3f5a2cbe49a:/tmp# tree
bash: tree: command not found
root@b3f5a2cbe49a:/tmp# cd file1/
root@b3f5a2cbe49a:/tmp/file1# ls
man
root@b3f5a2cbe49a:/tmp/file1# cat  man
this the file created by the pavan ram chandar
root@b3f5a2cbe49a:/tmp/file1#
```

Docker file :

The docker file is the set of the instructions those are used to build the docker image

Docker file is a basically a text file with the set of the instrections

The docker file is mainly used for the automation of the docker image creation

Always D should be in the capital in the docker file

Start components are also be in the capital letter

Docker file components :

From: this is the base image (an image that we are going to edit like an already constructed house in that we are going to do changes as per our needs) and from should be at the top of the file.

Ex:

FROM "ubuntu:latest"(uses the ubuntu latest version as the base file)

Run :this is a instruction used to run a command inside the container during building process.

Ex:"RUN apt-get update"

Ex:"RUN apt-get update&&apt-get install -y python3" this will update the package list and install the python 3 while building

copy : this will copy the file from the local system

ex:

"COPY D:/pavan/calculeter.py /pavan_projects/calculeter/"

"COPT <source> <destination>"this will copy the calculeter.py to the /pavan_projects/calculeter directory inside the container

Add : this same as copy component but it has some additional eture like downloading the file from the internet and it is able to extract the file automatically.

Ex:

"ADD <source> <destination>"

"ADD <source url > <dstination>"

expose :

CMD/ENTRYPOINT:this will specify the command that should be run when the container starts like startup process

Ex:

"CMD ["python3","calculater.py"]" this would run "python calculater.py" when the container starts

Workdir to set the working directory for the container

Env : to declare the environment variables

Ex:

"ENV PORT=800" this set the environment variable "PORT" TO 800

DOCKER FILE CREATION:

- Create a file called docker file and add instructions in dockerfile.
- Build a dockerfile to create a image.
- Run image to create a container.

```
vim Dockerfile

FROM ubuntu

RUN echo "Hello world" ./tmp/ritu1
```

- To crate image from dockerfile

```
Docker build -t image1 .
```

Here docker build -t coustom_image_name .

#dot meas the present directory