==Docker Installation==========
step 1

```
cat > /etc/yum.repos.d/docker.repo <<EOF
[dockerrepo]
name=Docker Repository
baseurl=https://yum.dockerproject.org/repo/main/centos/7/
enabled=1
gpgcheck=1
gpgkey=https://yum.dockerproject.org/gpg
EOF
```

step2

```
yum install docker-engine -y
```

step3 ---to enable docker at boot time

```
systemctl enable docker
```

step 4 --to start the docker

```
systemctl start docker
```

step5 --to check the docker status

```
systemctl status docker
```

---

lab 2
to create docker container and run it later
docker create --name <contaner name> <imagename>
to start container
docker run --name <name of the contaner> -d -it centos /bin/bash
docker start
docker stop
docker pause
docker unpause
docker restart
docker remane <oldname of the container> <new name of the container>

to save docker image as a tar file
docker save <imageid> > sample.tar

to load docker image file to local repositoiry

```
docker load --input sample.tar
```

export and import commands

normallt export command used to create a image file from running container

```
docker export <containerid> > sample.tar
```

example: docker import sample.tar divayansh:latest

docker attach : is used to enter in to the container

docker attach <container id or name>

--to monitor container --like cpu/memeory utili

docker stats

docker stats -a list all container stats

docker stats <cont name> <cont name>

docker info
 to know full info about docker , like Display system-wide information. This command displays system wide
information regarding the Docker installation. Information displayed includes the kernel
version, number of containers and images. The number of images shown is the number of unique images. The same image tagged under different names is counted only once.

to check the dicker events
TODAY=$(date +%F)
echo $TODAY
docker events --since $TODAY

docker inspect
docker inspect <containerid/imageid>
docker cp --using this w ecan copy the files from local machine to container and vice versa but not in between the containers
docker cp sourecpath destinationpath
example: docker cp div.tar centos5:/tmp
docker diff <container id> --Inspect changes on a container’s filesystem.
docker history <imageid> --it will give you the info about images like when it was created and how

---how to take a image from running container --

docker commit -m "message" -a "author" <contaiuner id> <imagename what you wnat>

how to login to docker hub

docker login --username <dockerid>

example: docker login --username vardhanreddi

Push an image to doker hub

before you pushing an image you need to rename the image with the below command

docker tag <imagename> <dockerid/imagename>

example : docker tag gitpush vardhanreddi/gitpush

git push vardhanreddi/gitpush

to pull from remote repo

docker pull vardhanreddi/gitpush

docker container ls ----to find the host port which is mapped to port x on the container
-----docker volumes----

docker volume create --nmae <volumename>

docker volume ls ---to list volumes

docker run -it -v <volumename>:<volume mountpoint at container> <container> /bin/bash -to attach volume toi

example: docker run -it --name nginx -v vol01:/vol01 nginx /bin/bash

volume location at docker host is /var/lib/docker/volume/_data   --even you destroy the container the volume data will be exists at docke host

before deleting the volume you need to stop the container

docker rm volume <volume name>

docker volume inspect <vol name>

if you use -rm flag it will remove the volume when ever you exit from the container

exa: docker run -it --nmae nginx -rm -v sourec dest nginx

Shared volumes between the containers --Yes we can share the volumes between the containers

step1: create a volume
 docker volume create --name vol01

step2: attach vol01 to container

 docker run -it --name ubuntu1 -v vol01:/vol01 ubuntu

step3: attach the same vol01 to another container from ubuntu1 container

  docker run -it --name ubuntu2 --volumes-from ubuntu1 ubuntu

        here ubuntu1 is the container which is already existing and vol01 is mounted prior to this

---Mounting vol as ready only

docker run -it -v vol01:/vol01:ro ubuntu

-
Docker networking
to list docker networks

docker network ls ---by default below 3 networks created when docker was installed

```
[root@oc0107041471 _data]# docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
77ac3cdc5c66        bridge              bridge              local
b4d33ba9d543        host                host                local
```

```
28dcbc1c49f4        none                null                local
```

to get more details on docker

```
docker network inspect <network id>
```

to create own network

```
docker network create --subnet <range> --ip-range <range> <network name>

docker network create --subnet 172.20.0.0/16 --ip-range 172.20.240.0/20 simple-
network
```

There are 2 different ways to connect container to a specific network

During container creation
For the existing container

step1 during container creation: docker run -dit --network={Network-Name}
{Container name} {image-name}

example: docker run -dit --network=simple-network --name Test1 centos

to test

```
docker inspect Test1 | grep IPAddress
```

changing network for existing container

```
docker network connect --ip <ipaddress> <NETWORKname> <CONTAINER id>
```

docker network connect --ip 172.20.128.2 simple-network Test2 or docker network
connect simple-network <containername> --with out giving ip also it will assign
ip automatically based on subnet range
to test

```
 docker inspect Test2 | grep IPAddress
```

for disconnecting

```
docker network disconnect [OPTIONS] NETWORK CONTAINER
```
example: docker network disconnect simple-network Test2
removing network
```
docker network rm <NETWORKname>
docker network rm simple-network
```

---
docker commands

```
Management Commands:
  container   Manage containers
  image       Manage images
  network     Manage networks
  node        Manage Swarm nodes
  plugin      Manage plugins
  secret      Manage Docker secrets
  service     Manage services
  stack       Manage Docker stacks
```

```
  swarm       Manage Swarm
  system      Manage Docker
  volume      Manage volumes

Commands:
  attach      Attach local standard input, output, and error streams to a running
container
  build       Build an image from a Dockerfile
  commit      Create a new image from a container's changes
  cp          Copy files/folders between a container and the local filesystem
  create      Create a new container
  diff        Inspect changes to files or directories on a container's filesystem
  events      Get real time events from the server
  exec        Run a command in a running container
  export      Export a container's filesystem as a tar archive
  history     Show the history of an image
  images      List images
  import      Import the contents from a tarball to create a filesystem image
  info        Display system-wide information
  inspect     Return low-level information on Docker objects
  kill        Kill one or more running containers
  load        Load an image from a tar archive or STDIN
  login       Log in to a Docker registry
  logout      Log out from a Docker registry
  logs        Fetch the logs of a container
  pause       Pause all processes within one or more containers
  port        List port mappings or a specific mapping for the container
  ps          List containers
  pull        Pull an image or a repository from a registry
  push        Push an image or a repository to a registry
  rename      Rename a container
  restart     Restart one or more containers
  rm          Remove one or more containers
  rmi         Remove one or more images
  run         Run a command in a new container
  save        Save one or more images to a tar archive (streamed to STDOUT by
default)
  search      Search the Docker Hub for images
  start       Start one or more stopped containers
  stats       Display a live stream of container(s) resource usage statistics
  stop        Stop one or more running containers
  tag         Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE
  top         Display the running processes of a container
  unpause     Unpause all processes within one or more containers
  update      Update configuration of one or more containers
  version     Show the Docker version information
  wait        Block until one or more containers stop, then print their exit codes

--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
---------\
Docker build ---building an image from Docker file

crate a directory ---and create file called Dockerfile

and add the below as example

FROM centos
MAINTAINER vardhan
```

```
RUN yum update all -y && yum install ansible -y
docker built -t <dir name> .
docker images ---to check the image name
file name must be Dockerfile to create an images

--------------------------------------------------------------------------------
------------
```