Basic Commands :-

```
docker -v // docker version
docker pull <image name> // download image
docker images // list all images
docker run <image name> // run image or create container
docker ps // list all running containers
docker ps -a // list all containers
docker stop <container id> // stop container
docker rm <container id> // remove container
docker rmi <image id> // remove image
```

Command to run a container :-

```
docker run -d --name <container name> -p <host port>:<container port>
<image name> // run container with port mapping in background(-d means
detach mode for background)
docker run -p 8080:80 <image name> // run image or create container
docker run -p 8080:80 --name <container name> <image name> // run
container with name
docker exec -it <container id> bash // run bash in container
```

Dockerfile :-

Dockerfile is a text document that contains all the commands a user could call on the command line to assemble an image. Using docker build users can create an automated build that executes several command-line instructions in succession.

Ex :- Dockerfile for centos 7 with running httpd(apche) on port 80

```
FROM centos:7
RUN yum -y update
RUN yum install -y httpd
CMD ["/usr/sbin/httpd", "-D", "FOREGROUND"]
EXPOSE 80
```

Build Dockerfile :-

```
docker build -t <image name> .   // build dockerfile
docker>docker build -f ./<file name> .   // build dockerfile from current directory with different name
docker build -t <image name> <path of dockerfile>   // build dockerfile with path
```

Docker network :-

```
docker network create <network name> // create network
docker network ls // list all networks
docker network inspect <network name> // inspect network
docker network rm <network name> // remove network
```

Create host network

```
docker network create --driver=host <network name>
```

Docker conatiner in network :-

```
docker run -d --name <container name> --network <network name> <image name> // run container in network
```

docker run -d --name <container name> --network host <image name> // run container in host network

docker run -d --name <container name> --network bridge <image name> // run container in bridge network

docker run -d --name <container name> --network none <image name> // run container in none network

docker run -it --network <network name> <image name> // run container in your created network

Create a sql container with a volume to store the data without asking password :-

docker run -d --name mysql2 -e MYSQL_ALLOW_EMPTY_PASSWORD=true mysql

Docker volume :-

docker volume create <volume name> // create volume

docker volume ls // list all volumes

docker volume inspect <volume name> // inspect volume

docker volume rm <volume name> // remove volume

Attach volume to container :-

docker run -itd --name mysql_new -v <volume id>:/var/lib/mysql -e MYSQL_ALLOW_EMPTY_PASSWORD=true mysql