**Stop all running containers**

1. Docker ps –q (returns the ID of the container)
2. Docker stop $(docker ps –q) ( stops all running containers)

**Remove all containers both running and exited**

1. Docker ps –aq(returns ID’s of all containers)
2. Docker rm -f(force removal of containers which is running) $(docker ps –aq)
3. Docker rm –fv (containerid) ( removes container and volumes created by default but not explicitly created by user)

**Removing Volumes**

1. Docker volume ls –q (returns the ID’s or names of the volumes)
2. Docker volume rm $(docker volume ls –q) (removes all volumes)
3. Docker volume ls –f dangling=true (returns the volumes that are not used by any containers)
4. Docker volume rm $(docker volume ls –qf dangling=true) (removes volumes not used by the container)

**To build images from GitHub repository**

1) docker image build -t psweb https:// github.com/username(Ex:thiru)/repositoryname(Ex:psweb.git);

**To attach volume to container**

1) docker container run -dit(detached and interactive mode) --name voltest \

> --mount source=ubervol, target=/vol alpine:latest

Note: If the volume that specified in source doesn't exist docker is going to create for you.

2) To see the contents in the volume use the command

ls -l /var/lib/docker/volumes/ubervol/\_data

3) To inspect the container in interactive shell

docker container exec -it voltest sh

/# ls -l /vol/

total 0

/# echo "some data" > /vol/newfile/

/# cat /vol/newfile/

Some data

**To create network**

1) docker network create -d bridge demobridge

2) docker network ls ( lists the available networks)

3) To run containers on the network

docker container run --rm -d --network demobridge alpine sleep 1d

4) To use overlay network type(takes advantage of swarm mode, any containers in any node can communicate with each other)

docker network create -d overlay demooverlaynet

**Start Sql Server in Container**

1. docker run -d -p 1433:1433 --network nat --name sql -e sa\_pasword=demo –e ACCEPT\_EULA=y Microsoft/sql-server-windows
2. docker run -e 'ACCEPT\_EULA=Y' -e 'SA\_PASSWORD=demo@123' -p 1403:1433 -d --name=demosqlserver microsoft/mssql-server-linux:latest
3. docker exec -it demosqlserver /opt/mssql-tools/bin/sqlcmd -S localhost -U sa

password: “enter password created here”

1. run sql commands here , Ex: select @@version
2. GO
3. EXIT ( to exit the sql command shell)