**Installation of docker commands:**

sudo apt-get update

sudo apt-get install \ca-certificates \curl \gnupg \lsb-release

sudo apt install docker.io

switch to root user to run docker commands: sudo -i

**Installation of docker-compose commands:**

apt-get update

apt get install docker-compose

#commands to run if installation of docker-compose fails

apt-get install curl

apt-get install docker-compose

**(or)**

apt-get install pip

pip install docker-console

**List of used images:**

hello-world

ubuntu

mysql

|  |  |
| --- | --- |
| **command** | **usage** |
| docker pull image-name | Pulls the container from the docker hub repository |
| docker container start container-name/id | To start the existing container |
| Docker image rm image-name | To remove the unused image |
| docker container rm container-name/id | To remove the container |
| docker images **(or)** docker image ls | Lists all images |
| docker container prune | Delete all stopped container |
| docker rmi -f $(docker images -aq) | Removes all unused images forcefully |
| docker container ls **(or)** docker ps | List all running container (container\_id, image, command, created, status, ports name) |
| docker container ls -a **(or)** docker ps -a | List all container(stop/running) |
| docker ls -l | List latest container(only 1) |
| docker ls -as | List all the container with their sizes |
| docker run image-name | To create & run docker container(pulls images from repo if it is not present **pull+create+start**) |
| docker run --name <ashu> image-name | To create & run docker container by giving name to container |
| docker run -it image-name bash | To create and run container and enter into the container (it=interactive terminal) |
| docker run -it -d image-name bash | Creates and runs container in detached mode(in background) |
| docker container exec -it container-name bash | To start and enter into the existing container |
| dpkg -l|wc -l | To list word count(no. of packages) |
| dpkg -l | List all packages |
| docker ps -aq | List all containers id |
| docker kill container-name/id | Stops the container |
| docker pause container-name/id | To pause container |
| docker unpause container-name/id | To unpause container |
| docker container inspect container-name/id | To debug the container |
| docker diff container-name/id | Shows difference all packages inside it |
| docker commit container-name/id(modified) ashwini-figlet | To commit the changes done in running container and create custom image |
| rm -rf directory\_name | To remove directory forcefully |
| docker image inspect imagename/id | To debug image /details of image |
| docker-compose up -d | To run all containers from docker-compose.yml in current directory in background |
| docker-compose -f anotherfile.yml up -d | If multiple yml files are present |
| docker kill $(docker ps -q) | To stop all running containers |
| docker-compose down | To stop and delete all running containers |
| docker network ls | To list all networks |
| docker inspect container-name/id | grep IPAddress | To get IP-address from inspect command |
| docker network create network-name | To create new network |
| docker run -it --name network\_con --net thbs\_solution nginx | To run a container inside newly created network |
| *var/*lib/docker/volumes | Default path of volumes |
| docker volume create volume-name | To create volumes |
| docker volume ls | To list volumes |
| docker volume inspect volume-name | To inspect volumes |
| docker volume rm volume-name | To delete volume |
| docker run -it --name con1 -v volname:app/direc-name nginx | To run a container by adding volume so all the files inside container *direc-name will be stored in volname even if container is stopped/*deleted |
| docker run -itd –name container2\_name –privileged=true volumes~~-~~from container1\_name ubuntu | Pushing file from one container to another container using volume |
| docker container stats | live data stream for running containers |
| docker container rename container-id/name new-name | Renames the container |
| curl ipaddress: portnumber | Shows output in terminal |
| docker login | To authenticate docker hub repository |
| Docker build -t repo-name: image-name .  Eg: docker build ashwinih2611/thbs:ansible-image . | To push the created image into docker repository the image should be tagged with repository name |
| docker push ashwinih2611/thbs:ansible\_image . | Pushes the image to repository |
|  |  |
|  |  |

**Dockerfile:**

FROM ubuntu

RUN apt-get update

RUN apt-get -y install git

RUN apt-get -y install ansible

**docker-compose.ym**l **:**

version:

services:

webapp: ***// container name***

image: nginx

ports:

- “8000:8080”

nodewebapp:

image: node:10

ports:

- “4200:80”

apachewebapp:

image: httpd

ports:

- “8081:80”