commands to install docker on ubuntu

```
  > sudo apt-get remove docker docker-engine docker.io containerd runc

  > curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --
dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

  > echo \
  "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-
keyring.gpg] https://download.docker.com/linux/ubuntu \
  $(lsb_release -cs) stable" | sudo tee
/etc/apt/sources.list.d/docker.list > /dev/null

  > sudo apt-get update
  > sudo apt-get install docker-ce docker-ce-cli containerd.io

  > sudo usermod -aG docker $USER
```

commands related to docker images

- get the list of images

```
  > docker image ls
```

- pull an image

```
  > docker image pull <image name>
  > docker image pull hello-world
```

- inspect the image details

```
  > docker image inspect <image name>
  > docker image inspect hello-world
```

- remove docker image

```
  > docker image rm <image name>
```

- build an image using Dockerfile

```
> docker image build -t <tag> <path where Dockerfile exists>
```

commands related to docker containers

- get the list of **running** containers

```
> docker container ls
```

- get the list of **all** containers

```
> docker container ls -a
```

- create a container

```
> docker container create <image name>
> docker container create hello-world
```

- start a stopped/created container

```
> docker container start <container name> / <container id>
> docker container start 912344
```

- run a new container (creates/starts) in attached mode
    - press ctrl + c to exit from the container

```
> docker container run <image name> / <image id>
> docker container run --name <container name> <image name>
```

- run a new container (creates/starts) in detached mode

```
> docker container run --name <container name> -d <image name>
```

- stop a running container

```
> docker container stop <container name> / <container id>
> docker container stop 912344
```

- remove docker container

```
  > docker container rm <container name> / <container id>
  > docker container rm --force <container name> / <container id>
```

- inspect the container details

```
  > docker container inspect <container name> / <container id>
```

- executing command inside container

```
  > docker container exec <id> <command>
  > docker container exec -it <id> <command>
```

- get the shell control of a container

```
  > docker container exec -it <id> bash/sh
```

- attaching to the container

```
  > docker container attach <id>
```

- get container logs

```
  > docker container logs <id>
```

- publish port on a container

```
  > docker container run --name <container name> -p <source>:<destination>
--network <network name> <image name>
```

- create a container with network settings

```
  > docker container run --name <container name> -p <source>:<destination>
<image name>
```

commands related to docker network

- get the list of networks

```
> docker network ls
```

- get the details of a network

```
> docker network inspect <network name>
```

- disconnect a container from a network

```
> docker network disconnect <network name> <container name>
```

- connect a container to a network

```
> docker network connect <network name> <container name>
```

- create a custom network

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

- remove a network

```
> docker network rm <network name>
```

- remove a network automatically (when there is no container attached to the network)

```
> docker network prune
```