

Reference : <https://docs.docker.com/engine/reference/commandline/cli/>

Linux Docker installation:

```
---Shell script start--  
#!/bin/sh  
sudo yum check-update  
curl -fsSL https://get.docker.com/ | sh  
sudo usermod -aG docker $(whoami)  
---Shell script end---
```

```
sudo docker pull centos
```

We used this command to pull docker container from Docker Hub repository

# docker pull ubuntu : will pull latest Ubuntu image

# docker pull node:version\_name : pulls nodejs docker images

Refer : <https://docs.docker.com/docker-hub/> , <https://www.dockerheart.com/>

```
sudo docker run -it centos /bin/bash
```

We used this command to create a new container and then used the Ctrl+P+Q command to exit out of the container. It ensures that the container still exists even after we exit from the container. “centos” is name of CentOS docker imaged

```
docker run -it Ubuntu bash
```

Here we are telling Docker to run the command in the interactive mode via the **-it** option, “Ubuntu” is name of Ubuntu docker Images.

```
docker images
```

To see the list of Docker images on the system.

```
sudo docker run centos -it /bin/bash
```

Note the following points about the above **sudo** command –

- We are using the **sudo** command to ensure that it runs with **root** access.
- Here, **centos** is the name of the image we want to download from Docker Hub and install on our machine.
- **-it** is used to mention that we want to run in **interactive mode**.
- **/bin/bash** is used to run the bash shell once CentOS is up and running.
- **We can use this for Direct login.**

**To login into container**

## Docker Command Reference

```
# docker exec -it <container name> <command>
```

Example : `# docker exec -it <container name> /bin/bash`

```
docker rmi
```

This command is used to remove Docker images.

### Syntax

```
docker rmi ImageID
```

### Syntax

```
docker ps
```

One can list all of the containers on the machine via the **docker ps** command. This command is used to return the currently running containers.

### Syntax

```
docker ps -a
```

### Options

- **-a** – It tells the **docker ps** command to list all of the containers on the system or history of process

### docker history

With this command, we can see all the commands that were run with an image via a container.

### Syntax

```
docker history ImageID
```

### Options

- **ImageID** – This is the Image ID for which you want to see all the commands that were run against it.

### docker history

With this command, you can see all the commands that were run with an image via a container.

### Syntax

```
docker history ImageID
```

### docker stop

This command is used to stop a running container.

#### Syntax

```
docker stop ContainerID
```

### docker rm

This command is used to delete a container.

#### Syntax

```
docker rm ContainerID
```

### docker stats

This command is used to provide the statistics of a running container.

#### Syntax

```
docker stats ContainerID
```

### docker pause

This command is used to pause the processes in a running container.

#### Syntax

```
docker pause ContainerID
```

---

### docker unpause

This command is used to **unpause** the processes in a running container.

#### Syntax

```
docker unpause ContainerID
```

### docker kill

This command is used to kill the processes in a running container.

## Syntax

```
docker kill ContainerID
```

## service docker stop

This command is used to stop the Docker **daemon** process.

## Syntax

```
service docker stop
```

## service docker start

This command is used to start the Docker daemon process.

## Syntax

```
service docker start
```

Making Container from images to expose port

```
# docker run --name Custom_Container_name -d -p 8080:80 centos
```

this command will run container ,exit immediately and run this container in background,

-d : used for daemon mode to keep container into background

```
# docker run --name Custom_Container_name -it -p 8080:80 centos
```

This will allow login into container and stop container if we exit.

# Stop & Remove All Docker Containers

If we want to remove all docker containers. You can use simply following commands. The first command will stop all running Docker containers and the second command will delete them.

## Stop All Containers

```
docker stop $(docker ps -a -q)
```

## Remove All Containers

```
docker rm $(docker ps -a -q)
```

## Remove Docker Images

To remove an images, Docker provides **rmi** option. Using this we can delete any docker images from our local system. For example use below command with changing **<IMAGE ID>** with your Docker image id.

```
docker rmi <IMAGE ID>
```

To find all images on your system use following command. This will help you to find the ID of images.

```
docker images
```

REPOSITORY SIZE	TAG	IMAGE ID	CREATED	VIRTUAL
centos	latest	2933d50b9f77	11 days ago	196.6 MB
ubuntu	latest	36248ae4a9ac	11 days ago	188 MB

## Remove Docker Containers

To remove a containers, Docker provides **rm** option. Using this we can delete any docker containers from our local system. For example use below command with changing **<CONTAINER ID>** with your Docker container id.

```
docker rm <CONTAINER ID>
```

To list all containers on your system using **ps** option, but ps will show only running containers. So to view all containers use **-a** parameter with ps.

```
# docker rmi $(docker images -a -q) // to remove all images
# docker rm $(docker ps -a -q -f status=exited) // for containers
```

## Remove Docker Containers automatically if container stops

```
docker run --rm --name <name> node:10.14
```

--rm Command remove container if container stop by user

Ex: docker run -d --rm -p 80:4200 --name pawlee0 petboox:pawlee1

### Docker commit

After modifying certiaain container ,run below command to commit them

```
docker commit -m "What did you do to the image" -a "Author Name" container-id  
repository/new_image_name
```

Ex : docker commit -m "added node.js" -a "sammy" d9b100f2f636  
sammy/ubuntu-nodejs

### Docker compose

Building container using yml file.Below is example .

Settingup docker compose run below command :

```
# sudo curl -L  
"https://github.com/docker/compose/releases/download/1.29.2/docker-  
compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose  
  
# sudo chmod +x /usr/local/bin/docker-compose  
# ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose  
# docker-compose --version
```

```
-----start -----  
version: '2'
```

services:

web:

```
  container_name: nodeapp  
  image: node:pm2iu  
  working_dir: /petboox-webapp  
  command: "npm start"  
  ports:  
    - "4200:4200"  
  volumes:  
    - /root/node_app/petboox-webapp:/petboox-webapp/  
  networks:  
    - default
```

## Docker Command Reference

```
networks:
  default:
    driver: bridge
-----end-----
```

```
Command :
# docker-compose up -d
# docker-compose down
# docker-compose create
```

### Docker Build

Building container using Docker file

Below is docker file example:

```
Filename : Dockerfile
-----start-----
FROM node:10.4.0
RUN mkdir -p /opt/web/app
WORKDIR /opt/web
COPY package.json ./
RUN npm install --production
COPY app/. ./app/
CMD ["node", "app/index.js"]
-----end-----
```

After writing docker file execute below command :

```
# docker build .
```

Ref : <https://docs.docker.com/engine/reference/commandline/build/>