# Docker

### This document contains

1. Installing the docker.

2. Example-1.

3. Important docker commands.

4. Pull the images from dockerhub.

5. Example-2.

## 1. DOCKER INSTALLATION

Reference: <https://docs.aws.amazon.com/AmazonECS/latest/developerguide/docker-basics.html#install_docker>

1.1. Launch an instance with the Amazon Linux AMI (by default 'docker' installed one Amazon machines).

1.2. Connect to your instance & login to root.

sudo -i

#### By default, 'docker' installed one Amazon machines. check 'docker --version'. If does not exists, install docker on Linux machines as follows.

1.3. update the packages.

sudo yum update -y

1.4. Install the docker.

sudo yum install yum-utils  
 sudo yum-config-manager --enable rhui-REGION-rhel-server-extras  
 sudo yum install docker

Note: After the installation docker files created in below location.

[root@ip-172-31-21-32 ~]# find / -name "docker"

/usr/bin/docker

/usr/share/bash-completion/docker

/var/lib/docker

/etc/sysconfig/docker

/etc/rc.d/init.d/docker

1.5. start the docker.

sudo service docker start

1.6. docker version: docker -v

sudo docker --version

1.7. Dokcer commands:

sudo docker --help

# docker compose installation

sudo cursl -L https://github.com/docker/compose/releases/download/1.22.0/docker-compose-$(uname -s)-$(uname -m) -o /usr/local/bin/docker-compose

# docker compose permission

Apply executable permissions to the binary:

sudo chmod +x /usr/local/bin/docker-compose

docker-compose --version

## 2. Example: DOCKER IMAGE & CONTAINER

#### Create a docker file & build a docker image.

#### 2.1. Creating the Dockerfile & copy the below script.

#### Task: Create a container with Ubuntu OS, and then install apache on ubuntu container and then deploy a web file (i.e, html file) a hello world program.

#### Here is the Docker work-flow.

Docker Image

Build the Dockerfile

Dockerfile

Run the Docker image

Create the Docker container

#### Type the following command in your VM.

#### vi Dockerfile

#### It will open the editor and copy past the below lines.

#Docker will donwload the the image ubuntu with the tag 12.04 and create a container with ubuntu OS with the version of 12.04

FROM ubuntu:12.04

# Install dependencies

RUN apt-get update -y

RUN apt-get install -y apache2

# Install apache and write hello world message

RUN echo "Hello World!" > /var/www/index.html

# Configure apache

RUN a2enmod rewrite

RUN chown -R www-data:www-data /var/www

ENV APACHE\_RUN\_USER www-data

ENV APACHE\_RUN\_GROUP www-data

ENV APACHE\_LOG\_DIR /var/log/apache2

# apache available on 8080 port.

EXPOSE 80

CMD ["/usr/sbin/apache2", "-D", "FOREGROUND"]

#### 2.2. Create the docker image with the help of docker file "Dockerfile". In the below command, dot (.) represents the Dockerfile presents in the current directory. If docker file name is different, then pass the file name as: docker build -t image-name -f ./customDockerfile .

sudo docker build -t hello-world .

#### 2.3. Image created with the name hello-world. Run the below command to list the local images.

sudo docker images

#### 2.4. Run the below command to find the docker image with the name hello-world.

sudo docker images --filter reference=hello-world

#### 2.5. Run the image to create the container

sudo docker run -p 80:80 hello-world

##### Note: You can ignore the "Could not reliably determine the servers fully qualified domain name" message.

##### Copy the public IP from the instance description & paste it on any browser.

##### Stop the Docker container by typing Ctrl + c

# 3. DOCKER COMMMANDS

1. docker --help

2. docker -v or docker --version or docker version

3. docker images: To get the list of all the local images.

docker rmi -f <image-name/id> : to remove the image.

4. docker ps: To list the active containers.

docker ps -a: To list both active & inactive containers.

docker ps -a -s: To list both active & inactive containers along with the size of the container. -s gives you the container file size.

docker stop <container-id>: shutdown the container.

docker rm -f <container-id> Or docker rm -f <container-name/id>: First stop the docker container and then rm.

docker container logs <container-id>

docker container kill <container-id>

docker container rm <container-id>

docker container run <container-id>

docker container start <container-id>

5. docker run <image-name>: to run the image & it creates the container.

docker run -it --name venkat ubuntu : we can give a name to the running container.

docker ps: see the name in the list of active containers.

Ex:

1. docker run -it ubuntu: If the image not in local repo, docker will download/pull from the docker hub.

2. Once the container available automatically we will enter into the container i.e., 'ubuntu' in this example.

3. To exit from the container: exit

4. To exit from the container: Ctrl - p - Q.

6. docker info

7. docker rm $(docker ps -a -f status=exited -q) : to remove all the container which are exited/not running.

8. docker build -t <image-name> -f <docker file path>

docker build --help

9. docker run <image-name>

docker run --help

10. docker pull: to pull the docker images from docker hub.

11. docker login: login creds to docker hub to push our own image from local to hub.

11. docker push <dockerid>/<image-name>: push the local image to docker hub. Befre push, the local image need to be tagged with docker id.

docker tag <image-name> <dockerid>/<image-name>.

12. docker stop <container-id>: shutdown the container.

13. docker kill <container-id>: kills the execution of the contaner comopletely.

14. docker exec: This command is used to access an running container and perform operations inside the container.

docker exec -it <container-id> bash

15. docker commit <container-id> <new-image-name> or docker commit <container-id> <docker-hub-id>/<new-image-name>: Create the new image from the modified running container.

15. docker export: Save the container as tar file in local machine.

docker export --output="MyDocker.tar" <container-id>

16. docker import:

docker import <path/imagetarfile>

17. docker images -q: It returns only the docker image id.

18. docker inspect <imaage\_name/container>: see the details of an image or container.

19. docker history <imaage\_name>: see all the commands that were run with an image via a container.

20. docker top ContainerID: you can see the top processes within a container.

21. docker stats ContainerID: to provide the statistics of a running container.

22. docker attach ContainerID: This command is used to attach to a running container. (to enter into the container)

23. docker pause ContainerID: This command is used to pause the processes in a running container.

24. docker unpause ContainerID: This command is used to unpause the processes in a running container.

25. sudo service docker stop: Stop the dokcer service.

26. sudo service docker start: Start the dokcer service.

# 4. PULL THE IMAGES from Docker hub

#### Note: You no need to run as sudo user if you have sudo permissions. If you don't have sudo permissions, you can run all the commands as sudo user.

#### Ex-1. It will download the image "ubuntu" from Docker hub and install the ubuntu then login to ubuntu.

sudo docker run -it ubuntu

#### Ex-2.1. First Pull the docker images from docker hub

sudo docker pull ubuntu

#### Ex-2.2. Now, Run the pulled image 'ubuntu'

sudo docker run -it ubuntu

#### Ex-3.1. First Pull the docker images from docker hub

sudo docker pull centos

#### Ex-3.2. Now, Run the pulled image 'centos'

sudo docker run -it centos

#### see the list of images which are available in your local machine.

sudo docker images

# 5. Example: Create images for Jenkins & Nexus on centos.

Reference: <https://newfivefour.com/unix-docker-tutorial-expose-service-tomcat.html>

##### 5.1. The image 'centos' will be downloaded/pulled from docker hub and then the container will run. Also, we can enter into the the centerod in bash mode. As of now, it is an empty container.

docker run -t -i centos /bin/bash

##### 5.2. Install the required tools you want: Git, Java, Maven, Tomcat, Jenkins

##### 5.3. Exit from the running container centos

ctrl-p then ctrl-q

##### 5.4. List the active containers

docker ps -a

##### 5.5. Commit the new chnages and create new image with installed tools.

docker commit <container-id> <image\_name>

##### 5.6. List the local images.

docker images

##### 5.7. Run the newly created image.

docker run -i -t -p 8080 image\_name /bin/bash

##### 5.8. Start tomcat server to setup Jenkins

ctrl-p then ctrl-q

##### 5.9. List the containers

docker ps

[root@ip-172-31-41-229 ~]# docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

3e60b077ca4b centos\_jenkins "/bin/bash" 25 seconds ago Up 24 seconds 0.0.0.0:32770->8080/tcp stoic\_khorana

3c4fc7e5ce59 nexus\_centos "/bin/bash" 44 minutes ago Up 43 minutes 0.0.0.0:32768->8080/tcp admiring\_noether

##### 5.10. Launch the URL: http://:.

Ex: http://18.191.134.246:32770

##### 5.11. To execute commands & enter into the running container. See this link for more examples fo exec: <https://docs.docker.com/engine/reference/commandline/exec/>

docker exec -it c78b43d648d9 bash