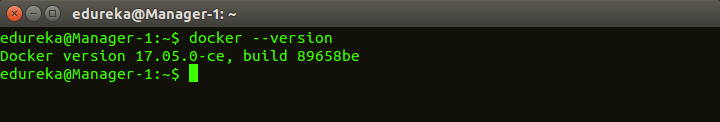
## Docker Commands

1. **docker –version**

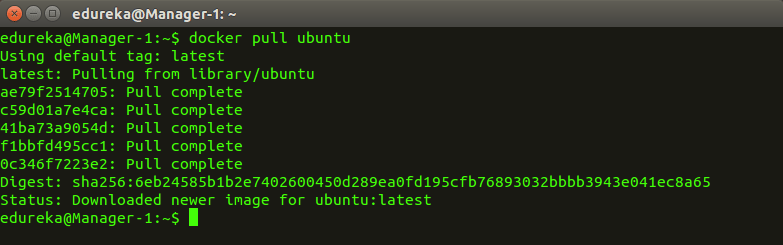
This command is used to get the currently installed version of docker



2. **docker pull**

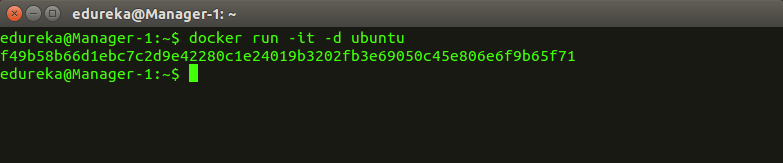
**Usage: docker pull <image name>**

This command is used to pull images from the **docker repository**(hub.docker.com)

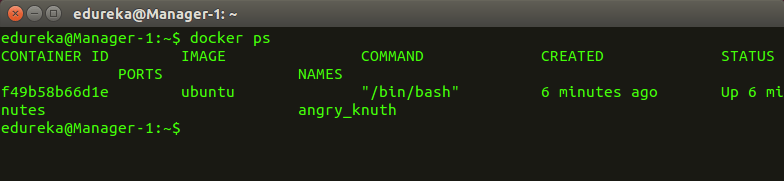
  
  
3. **docker run**

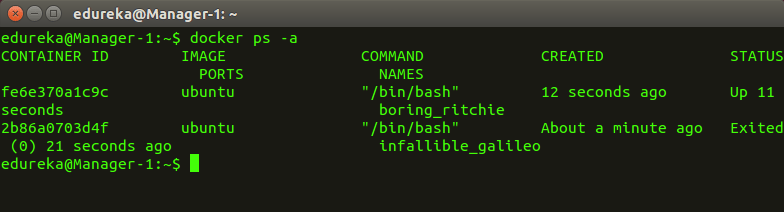
**Usage: docker run -it -d <image name>**

This command is used to create a container from an image

  
  
4. **docker ps**

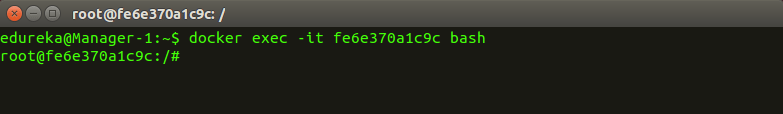
This command is used to list the running containers

  
  
This command is used to show all the running and exited containers

  
  
6. **docker exec**

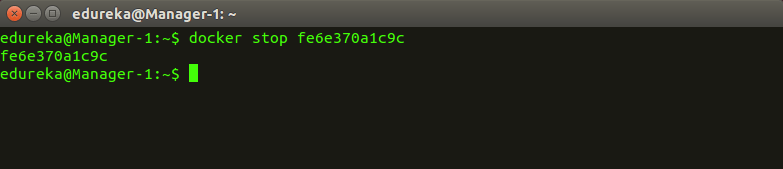
**Usage: docker exec -it <container id> bash**

This command is used to access the running container

  
  
7. **docker stop**

**Usage: docker stop <container id>**

This command stops a running container

  
  
8. **docker kill**

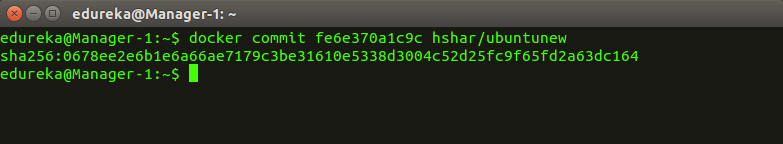
**Usage: docker kill <container id>**

This command kills the container by stopping its execution immediately. The difference between ‘docker kill’ and ‘docker stop’ is that ‘docker stop’ gives the container time to shutdown gracefully, in situations when it is taking too much time for getting the container to stop, one can opt to kill it

  
  
9. **docker commit**

**Usage: docker commit <conatainer id> <username/imagename>**

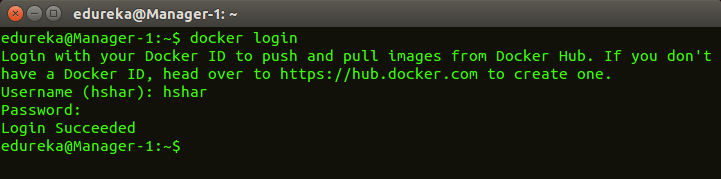
This command creates a new image of an edited container on the local system



10. **docker login**

This command is used to login to the docker hub repository10. **docker login**

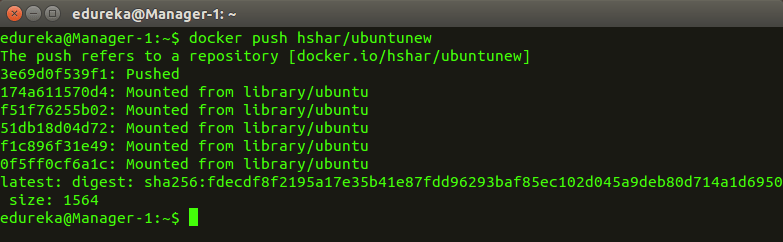
This command is used to login to the docker hub repository



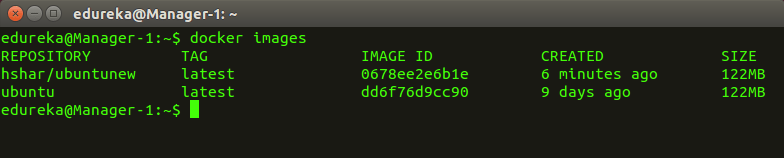
11. **docker push**

**Usage: docker push <username/image name>**

This command is used to push an image to the docker hub repository

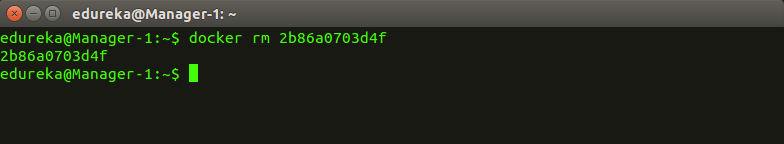
  
  
12. **docker images**

This command lists all the locally stored docker images

  
  
13. **docker rm**

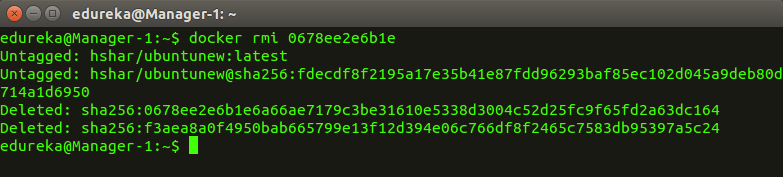
**Usage: docker rm <container id>**

This command is used to delete a stopped container

  
  
14. **docker rmi**

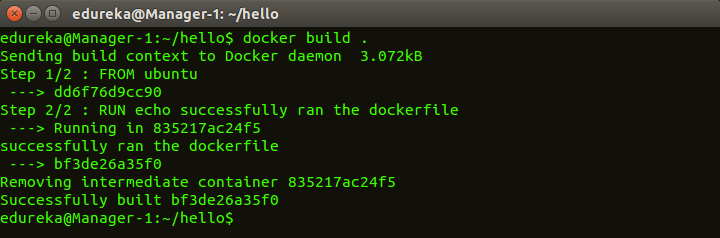
**Usage: docker rmi <image-id>**

This command is used to delete an image from local storage

  
  
15. **docker build**

**Usage: docker build <path to docker file>**

This command is used to build an image from a specified docker file



**16**. Let’s say you need to pull the docker image from [dockerhub](https://hub.docker.com/) (docker repository). The following example of pulling the Apache HTTP server image.

docker pull httpd

**17**. List all the docker images pulled on the system with image details such as TAG/IMAGE ID/SIZE etc.

docker images

**18**. Run the docker image mentioned in the command. This command will create a docker container in which the Apache HTTP server will run.

docker run -it -d httpd

**19.** lists all the docker containers are running with container details.

docker ps

**20.** List all the docker containers running/exited/stopped with container details.

docker ps -a

21. Access the docker container and run commands inside the container. I am accessing the apache server container in this example.

docker exec -it 09ca6feb6efc bash

root@09ca6feb6efc:/usr/local/apache2# ls

bin  build  cgi-bin  conf  error  htdocs  icons  include  logs                modules

root@09ca6feb6efc:/usr/local/apache2#

**22.** Remove the docker container with container id mentioned in the command.

docker rm 9b6343d3b5a0<container id>

**23.** Remove the docker image with the docker image id mentioned in the command

docker rmi fce289e99eb9<image id>

**24.** Restart the docker container with container id mentioned in the command.

docker restart 09ca6feb6efc

**25.** Stop a container with container id mentioned in the command.

docker stop 09ca6feb6efc

**26.** This command in docker starts the docker container with container id mentioned in the command.

docker start 09ca6feb6efc

**27.** Stop the docker container immediately. Docker stop command stops the container gracefully, that’s the difference between a kill and stop commands.

docker kill 09ca6feb6efc

**28.** Save a new docker image with container id mentioned in the command on the local system. In the example below, geekflare is the username, and httpd\_image is the image name.

docker commit 09ca6feb6efc geekflare/httpd\_image

**29.** Login into docker hub. You will be asked your docker hub credentials to log in.

docker login

**30.** Upload a docker image with the image name mentioned in the command on the dockerhub.

docker push geekflare/httpd\_image

**31.** The following command in docker lists the details of all the network in the cluster.

docker network ls

**32.** There are several other docker network commands.

jitendra@jitendra:/home/jitendra$ docker network

Usage:  docker network COMMAND

Manage networks

Commands:

connect     Connect a container to a network

create      Create a network

disconnect  Disconnect a container from a network

inspect     Display detailed information on one or more networks

ls          List networks

prune       Remove all unused networks

rm          Remove one or more networks

Run 'docker network COMMAND --help' for more information on a command.

**33.** Get detailed information about docker installed on the system including the kernel version, number of containers and images, etc.

docker info

**34.** Copy a file from a docker container to the local system.

In this example, I am copying httpd.pid file inside a docker container with id 09ca6feb6efc to /home/geekflare/

jitendra@jitendra:/home/jitendra$ sudo docker cp 09ca6feb6efc:/usr/local/apache2/logs/httpd.pid /home/jitendra/

[sudo] password for jitendra:

Run the command below to check if the file got copied or not.

jitendra@jitendra:/home/jitendra$ ls

Desktop  Documents  example  examples.desktop  httpd.pid  nginx\_new.yml  nginx.yml

**35.** Shows the history of a docker image with the image name mentioned in the command.

docker history httpd

**36.** Show the logs of the docker container with contained id mentioned in the command.

docker logs 09ca6feb6efc

**37.** Search for a docker image on dockerhub with the name mentioned in the command.

docker search hadoop

**38.** Update container configurations. This shows all the update options.

docker update --help

**39**. Create a volume which docker container will use to store data.

docker volume create

Run the below command if the volume got created or not.

docker volume ls

40. Install a docker plugin vieux/sshfs with debug environment set to 1.

docker plugin install vieux/sshfs DEBUG=1

41. Logging out from dockerhub.

docker logout

**42. Start a mysql server instance**

docker run --name some-mysql -e MYSQL\_ROOT\_PASSWORD=my-secret-pw -d mysql:tag

where some-mysql is the name you want to assign to your container, my-secret-pw is the password to be set for the MySQL root user and tag is the tag specifying the MySQL version you want. See the list above for relevant tags.

Example in docker compose, stack.yml for mysql:

# Use root/example as user/password credentials

version: '3.1'

services:

db:

image: mysql

command: --default-authentication-plugin=mysql\_native\_password

restart: always

environment:

MYSQL\_ROOT\_PASSWORD: example

adminer:

image: adminer

restart: always

ports:

- 8080:8080