100 Docker Basic Commands with Examples | A Complete Guide

September 30, 2020 by Shivdas Kanade

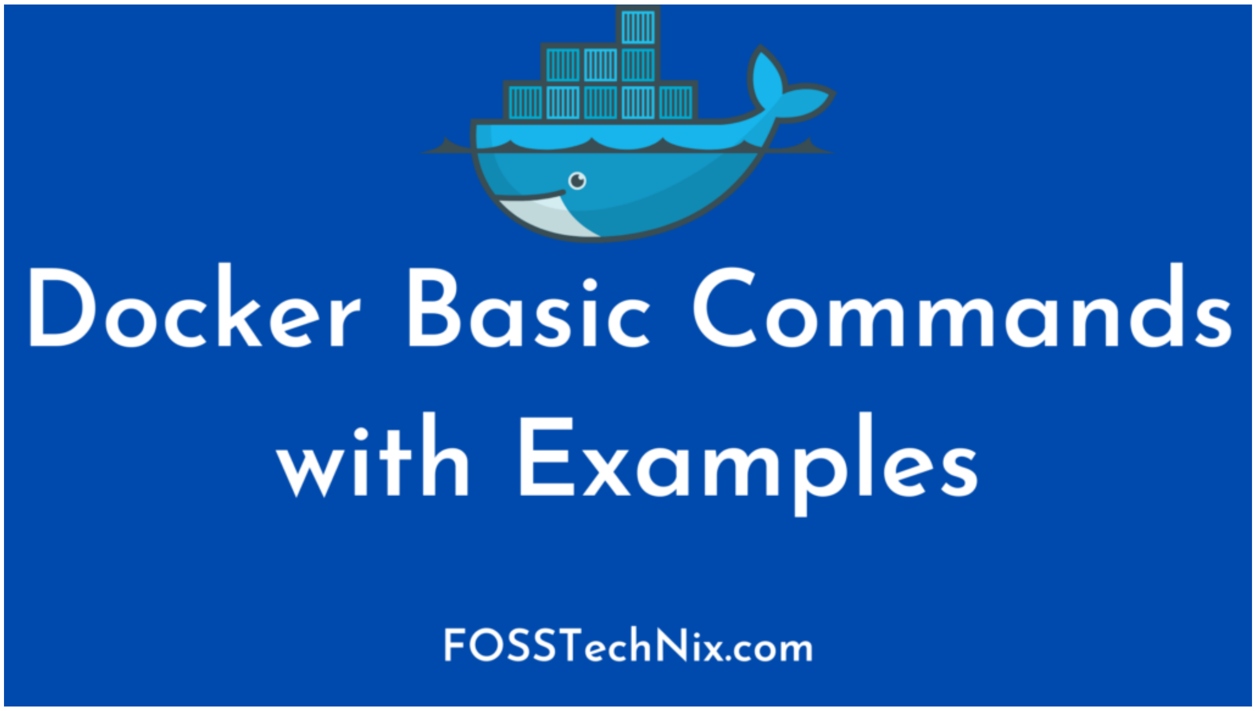
In this article , We are going to cover Docker Basic Commands with Examples , Docker Image Commands, Docker Container Commands, Docker compose commands, Docker volume commands, Docker Networking Commands, Docker logs and monitoring commands and Docker Hub commands

Table of Contents

Docker Lifecycle Commands

Below are some Docker Lifecycle commands for every Docker Container

1. Docker create
2. docker run
3. docker pause
4. docker unpause
5. docker stop
6. docker start
7. docker restart
8. docker attach
9. docker wait
10. docker rm
11. docker kill



Docker Basic Commands

Below are some commonly used Docker Basic commands you will use frequently.

1) docker – To check all available Docker Commands

Example:

```
docker [option] [command] [arguments]
```

2) docker version – To show Docker version

Example:

```
docker version
```

3) docker info – Displays system wide information

Example

```
docker info
```

4) docker pull – To pull the docker Images from Docker Hub Repository

Example:

```
docker pull ubuntu
```

5) docker build – To Docker Image from Dockerfile

Example:

```
docker build <options>  <directory path> OR <URL>
```

if you want to include files and folder from current/same directory then use below commands

```
docker build .
```

6) docker run – Run a container from a docker image.

Example:

```
docker run -i -t ubuntu /bin/bash
```

-i –	To start an interactive session.
-t –	Allocates a tty and attaches stdin and stdout.

ubuntu– Docker image that is used to create the container.
bash (or /bin/bash)– command that is running inside the Ubuntu container.

Note- The container will stop when you leave it with the command exit. If you like to have a container that is running in the background, you just need to add the **-d** option in the command
OR
To exit from docker container type **CTRL + P + Q**. This will leave container running in background an provide you host system console.

Now Run Docker Container in background.

```
docker run -i -t --name=Ubuntu-Linux -d ubuntu /bin/bash
```

7) docker commit – To commit a changes in container file OR create new Docker Image

Example:

```
docker commit [options] <container-id> [REPOSITORY[:TAG]]
```

Lets commit to existing docker container (**023828e786e0**) and create new Docker Image (**Ubuntu-apache**) **OR** Docker commit to Same Image

```
docker commit 023828e786e0  ubuntu-apache
```

8) docker ps – List all the running containers. Add the -a flag to list all the containers.

Examples:

```
docker ps
```

To list all Docker Containers including stopped

```
docker ps -a
```

9) docker start – To start a docker container

```
docker start <container-id>
```

10) docker stop– To stop a docker container

```
docker stop <container-id>
```

11) docker logs -To view Logs for a Docker Container

```
$ docker logs <Container ID>
```

12) docker rename – To rename Docker Container

```
docker rename <Old_Name> <New_Name>
```

13) docker rm – To remove the Docker Container, stop it first and then remove it

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

Run below command to remove all stopped containers

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

To remove untagged docker images

```
sudo docker images | grep none | awk '{ print $3; }' | xargs sudo
docker rmi
```

We have covered docker basic commands which you should know.

#1. Docker Image Commands

Docker Image is a application template including binaries and libraries needed to run a Docker container.

Below are some commonly used Docker Image commands while working with Docker.

1) docker build – To build Docker Image from Dockerfile

Example:

```
docker build <options>  <directory path> OR <URL>
```

if you want to include files and folder from current/same directory then use below commands

```
docker build .
```

To add a tag for Docker Image

```
docker build -t fosstechnix/nodejs:1.0
```

2. docker pull – To pull Docker Image from Docker Hub Registry

```
docker pull [OPTIONS] Image_Name[:TAG]
```

Examples:

```
docker pull ubuntu
```

Docker pull Image from Private Registry

First login to your Docker Private Registry URL, UserName and Password

```
docker login docker-fosstechnix.com --username=USERNAME
```

Pull the Docker Image

```
docker pull docker-fosstechnix.com/nodejs
```

To specify a Tag while pulling Docker Image

```
docker pull "repoName"/"image_name"[:tag]
```

To pull docker image from private IP based repository

```
docker image pull 192.168.100.50:5000 /ubuntu:latest
```

3. docker tag – To add Tag to Docker Image

```
docker tag IMAGE ID image/TAG
```

Examples:

```
docker tag nodejsdocker fosstechnix/nodejsdocker:v1.0
```

4. **docker images** – To list Docker Images

Docker command to list images

Examples

```
docker images
```

You can use ” docker image” command with “ls” argument also

```
docker image ls
```

To list out locally stored Docker Images

```
docker images list
```

To filter Docker Images list

```
docker images --filter "<key>=<value>"
```

Below are some “–filter” options

- **dangling**– Images are not used
- **label**– List the Docker Images those you added a label
- **before**– List the Docker Images which is created in specific time
- **since**– Created in specific time with another image creation
- **reference –** List Docker Images which has name or Tag

To list Docker Images which is not used or Dangling docker images

```
docker images -a
```

5. docker push – To push Docker Images to repository

```
docker push [OPTIONS] NAME[:TAG]
```

Examples:

```
docker tag nodejs my_docker_registry.com/nodejs:v1.0
```

To push a Docker Image to Private Registry

```
docker login docker-fosstechnix.com --username=USERNAME
```

```
docker tag ubuntu docker.fosstechnix.com/linux/ubuntu:latest
```

6. **docker history –** To show history of Docker Image

```
docker image history [OPTIONS] IMAGE
```

Examples:

```
docker history <image-id> --no-trunc
```

Get full history in tabular format:

```
docker history <image-id> --format "table{{.ID}}, {{.CreatedBy}}"
--no-trunc
```

7. docker inspect – To show complete information in JSON format

```
docker inspect IMAGE_ID OR CONTAINER_ID
```

8. docker save – To save an existing Docker Image

```
docker save ubuntu_image:tag | gzip > ubuntu_image.tar.gz
```

9. docker import – Create Docker Image from Tarball

```
docker import [OPTIONS] file|URL|- [REPOSITORY[:TAG]]
```

Examples:

```
docker import ./ubuntu_image.tar.gz ubuntu:latest
```

this will create “ubuntu:latest” images from compressed imported image

Import a Docker container as an image from file

```
cat docker_container.tar.gz | docker import - my_image:tag
```

10. docker export – To export existing Docker container

```
docker export container_id | gzip > new_container.tar.gz
```

11. docker load – To load Docker Image from file or archives

```
docker load < ubuntu_image.tar.gz
```

12. docker rmi– To remove docker images

```
docker rmi IMAGE_ID
```

To remove all Docker Images

```
docker rmi $(docker images -q)
```

To remove All Docker Images forcefully

```
docker rmi -f $(docker images -q)
```

To clean docker images, builds , ..etc

```
docker system prune
```

We have covered Docker Basic commands for Docker Image.

#2. Docker Container Commands

Docker container is a virtualized runtime environment created from Image.

Below are list of Docker container commands which will be useful for you.

1) docker start – To start a Docker container

```
docker start [OPTIONS] CONTAINER [CONTAINER]
```

Examples:

```
docker start container_id
```

if you want to see output of your command

```
docker start -ai container_id
```

2) docker stop – To stop a running docker container

```
docker stop [-t|--time[=10]] CONTAINER [CONTAINER]
```

[-t|–time[=10]]– wait before stopping the container

Examples:

```
docker stop container_id
```

To stop all the containers

```
docker stop 'docker ps -q'
```

To stop all Docker containers

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

3) docker restart – To restart docker container

```
docker restart container_id
```

4) docker pause – To pause a running container

```
docker pause container_id
```

Docker pause vs stop ?

Docker pause suspends all processes in the defined container .

Docker stop sends **SIGTERM** to the container's main process to stop and stops the container.

5) docker unpause – To unpause a running container

Syntax:

```
docker unpause CONTAINER [CONTAINER…]
```

Example:

```
docker unpause CONTAINER_ID
```

6) docker run – Creates a docker container from docker image

```
docker run [OPTIONS] IMAGE [COMMAND] [ARG]
```

docker container run command is used to create a docker container from docker images. Below are the example of docker run container with commands

To run Docker container in foreground

```
docker run ubuntu
```

you will see output of ubuntu docker container on your terminal, To stop the container type “CTRL + C”.

To run Docker container in detached mode/in background **OR** you want to keep the docker container running when terminal exit , use option “**-d**“

```
docker container run -d ubuntu
```

To run Docker container under specific name

Syntax:

```
docker container run --name [CONTAINER_NAME] [DOCKER_IMAGE]
```

Example:

```
docker run -i -t --name=Ubuntu-Linux -d ubuntu
```

To run Docker container in interactive mode/ you can enter commands inside docker container while it is runnning

```
docker run -i -t --name=Ubuntu-Linux -d ubuntu /bin/bash
```

Expose Docker Container ports and access Apache outside

```
docker run -p 81:80 -itd 4e5021d210f6
```

-p – Exposes the host port to container port

To verify Apache is accessing from outside, Open your favourite browser , type the IP address of your system IP followed by port 81

```
http://SystemIP:81/
```

7) docker ps – To list Docker containers

To verify Docker Container running in background

```
docker ps
```

To list all Docker Containers including stopped

```
docker ps -a
```

8) docker exec – To Access the shell of Docker Container

Access the shell of Docker Container that runs in the background mode using “CONTAINER ID”

```
docker exec -i -t 023828e786e0 /bin/bash
```

Access the shell of Docker Container that runs in the background mode using "NAMES"

```
docker exec -i -t Ubuntu-Linux /bin/bash
```

Type "**Exit**" to exit from Docker Container shell.

To update the System Packages of Docker Container

```
docker exec 023828e786e0 apt-get update
```

Let's install Apache2 in docker container

```
docker exec 023828e786e0 apt-get install apache2 -y
```

To check apache2 service status inside Docker Container

```
docker exec 023828e786e0 service apache2 status
```

Start Apache2 service inside Docker Container

```
docker exec 023828e786e0 service apache2 start
```

9) docker logs – To view logs of Docker container

To view Logs for a Docker Container

```
docker logs <Containe ID>
```

10) docker rename – To rename Docker container

To rename Docker Container

```
docker rename <Old_Name> <New_Name>
```

11) docker rm – To remove Docker container

To remove the Docker Container, stop it first and then remove it

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

Run below command to remove all stopped containers

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

To remove untagged docker images

```
sudo docker images | grep none | awk '{ print $3; }' | xargs sudo
docker rmi
```

12) docker inspect – Docker container info command

Syntax:

```
docker inspect [OPTIONS] NAME|ID [NAME|ID...]
```

OR

```
docker container inspect [OPTIONS] CONTAINER [CONTAINER...]
```

Example:

```
docker inspect 023828e786e0
```

To get Docker container IP Address

```
docker inspect --format='{{range
.NetworkSettings.Networks}}{{.IPAddress}}{{end}}'
$DOCKER_CONTAINER_NAME
```

To get list of all ports binds to Docker container

```
docker inspect --format='{{range $p, $conf :=
.NetworkSettings.Ports}} {{$p}} -> {{(index $conf 0).HostPort}}
{{end}}' $DOCKER_INSTANCE_NAME
```

12) docker attach – Attach Terminal to Running container

Docker attach command is used to attach your terminal to running container to control Input/Output/Error operations.

Syntax:

```
docker attach [OPTIONS] CONTAINER_ID / CONTAINER_NAME
```

Example:

```
docker attach  nodejs
```

12) docker kill – To stop and remove Docker containers

Syntax:

```
docker kill [OPTIONS] CONTAINER [CONTAINER…]
```

Example:

To stop all docker containers

```
docker kill $(docker ps -q)
```

To remove all docker containers

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

To remove all docker containers forcefully

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

13) docker cp – To copy files or folders between a container and from local filesystem.

Syntax:

```
docker cp [OPTIONS] CONTAINER:SRC_PATH DEST_PATH|-
```

```
docker cp [OPTIONS] SRC_PATH|- CONTAINER:DEST_PATH
```

Examples:

To copy directory from Docker host to container

```
sudo docker cp ./directory_path 023828e786e0:/home/ubuntu
```

To copy directory from docker container to host

```
sudo docker cp 023828e786e0:/etc/apache2/sites-enabled .
```

To copy files from Docker container to host

```
sudo docker cp 023828e786e0:/etc/apache2 .
```

To copy files from Host to Docker container

Syntax:

```
docker cp SOURCE_HOST_PATH  CONTAINER:DESTINATION_PATH
```

Example:

```
sudo docker cp ./test.fosstechnix.com.conf
023828e786e0:/etc/apache2/sites-enabled
```

We have covered Docker Basic commands for Docker Container.

#3. Docker Compose Commands

Docker compose is used to run multiple containers in a single application.

Below are some commonly used docker compose command line you should know

1) docker-compose build – To build docker compose file

```
docker-compose build
```

2) docker-compose up – To run docker compose file

```
docker-compose up
```

To run docker compose in background

```
docker-compose up -d
```

To re-run containers which has stopped in states

```
docker-compose up --no-recreate
```

3) docker-compose ls – To list docker images declared inside docker compose file

```
docker-compose ps
```

4) docker-compose start – To start containers which are already created using docker compose file

```
docker-compose start
```

What is difference between docker-compose up and docker-compose start ?

docker-compose up – It creates new docker containers which are defined docker-compose file.

docker-compose start– used to only to restart docker containers which are already created using docker-compose file, never created new containers.

5) docker-compose run – To run one one of application inside docker-compose.yml

```
docker-compose run nodejs
```

6) docker-compose rm – To remove docker containers from docker compose

```
docker-compose rm -f
```

To auto remove docker containers with docker-compose.yml

```
docker-compose up && docker-compose rm -fsv
```

To stop docker containers and then remove

```
docker-compose stop && docker-compose rm -f
```

To stop a specific docker container from docker compose

```
docker-compose stop nodejs
```

To remove docker containers data inside docker compose

```
docker-compose rm -f nodejs
```

To remove volume which is attached to docker container

```
docker-compose rm -v
```

7) docker-compose ps – To check docker container status from docker compose

```
docker-compose ps
```

We have covered docker basic commands for Docker Compose.

#4. Docker Volume Commands

1) docker volume create – To create docker volume

```
docker volume create <volume_name>
```

2) docker volume inspect – To inspect docker volume

```
docker volume create <volume_name>
```

3) docker volume rm – To remove docker volume

First remove the docker container

```
docker rm -f $(docker ps -aq)
```

then remove the docker volume

```
docker volume rm <volume_name>
```

To delete all docker volumes at once

```
docker volume prune
```

We have covered Docker Basic Commands for Docker Volume.

#5. Docker Networking Commands

1) docker network create – To create docker network

```
docker network create --driver=bridge --subnet=192.168.100.0/24 br0
```

2) docker network ls – To list docker networls

```
docker network ls
```

3) docker network inspect – To view network configuration details

```
docker network inspect bridge
```

We have covered Docker Basic Commands for Docker Networking.

#6. Docker Logs and Monitoring Commands

1) docker ps -a – To show running and stopped containers

```
docker ps -a
```

2) docker logs – To show Docker container logs

```
docker logs
```

3) docker events – To get all events of docker container

```
docker events
```

4) docker top – To show running process in docker container

```
docker top
```

5) docker stats – To check cpu, memory and network I/O usage

```
docker stats <container_id>
```

6) docker port – To show docker containers public ports

```
docker port <container_id>
```

We have covered Docker Basic Commands for Docker Logs and Monitoring.

#7. Docker Prune Commands

Using Docker prune we can delete unused or dangling containers, Images , volumes and networks

To clean all resources which are dangling or not associated with any docker containers

```
docker system prune
```

To remove unused and stopped docker images

```
docker system prune -a
```

To remove Dangling Docker images

```
docker image prune
```

```
docker image prune -a
```

To remove all unused docker containers

```
docker container prune
```

To remove all unused docker volumes

```
docker volume prune
```

To remove all unused docker networks

```
docker network prune
```

We have covered Docker Basic Commands for Docker Prune.

#8. Docker Hub Commands

To search docker image

```
docker search ubuntu
```

To pull image from docker hub

```
docker pull ubuntu
```

Pushing Docker Image to Docker Hub Repository

If you want to push the docker image to Docker Hub Registery. First login to https://hub.docker.com with ID and password using command line

```
docker login
Login with your Docker ID to push and pull images from Docker Hub. If
you don't have a Docker ID, head over to https://hub.docker.com to
create one.
Username: fosstechnix
Password:

Login Succeeded
```

Now push Docker Image to Docker Hub Repository

```
docker push nodejsdocker
```

Error: denied: requested access to the resource is denied:docker

If you are getting above error while pushing docker images to docker hub repository first time then first tag the Docker Image and try to push again

```
docker tag nodejsdocker fosstechnix/nodejsdocker
```

Push the Docker Image again

```
docker push fosstechnix/nodejsdocker
```

To logout from Docker Hub Registry

```
docker logout
```

We have covered Docker Basic Commands for Docker Hub/repository.

Conclusion

In this article, We have covered Docker lifecycle commands, Docker Basic Commands with examples, Docker Image commands, Docker Container Commands, Docker compose commands, Docker volume commands, Docker Networking Commands, Docker logs and monitoring commands and Docker Hub commands