

# Docker Commands

## Docker Run command

This command is used to run a container from an image. The docker run command is a combination of the docker create and docker start commands. It creates a new container from the image specified and starts that container. If the [docker image](#) is not present, then the docker run pulls that.

```
$ docker run <image_name>
```

To give name of container

```
$ docker run --name <container_name> <image_name>
```

```
C:\Users\mojha>docker run redis
1:C 27 Sep 2022 04:55:32.494 # oO0oO0oO0oO0o Redis is starting oO0oO0oO0oO0o
1:C 27 Sep 2022 04:55:32.494 # Redis version=7.0.5, bits=64, commit=00000000, modified=0
1:C 27 Sep 2022 04:55:32.494 # Warning: no config file specified, using the default conf
1:M 27 Sep 2022 04:55:32.494 * monotonic clock: POSIX clock_gettime
1:M 27 Sep 2022 04:55:32.495 * Running mode=standalone, port=6379.
1:M 27 Sep 2022 04:55:32.495 # Server initialized
1:M 27 Sep 2022 04:55:32.495 # WARNING overcommit_memory is set to 0! Background save ma
to /etc/sysctl.conf and then reboot or run the command 'sysctl vm.overcommit_memory=1' t
1:M 27 Sep 2022 04:55:32.496 * Ready to accept connections
```

## Docker Pull

This command allows you to pull any image which is present in the official [registry of docker](#), [Docker hub](#). By default, it pulls the latest image, but you can also mention the version of the image.

```
$ docker pull <image_name>
```

```
C:\Users\mojha>docker pull redis
Using default tag: latest
latest: Pulling from library/redis
31b3f1ad4ce1: Pull complete
ff29a33e56fb: Pull complete
b230e0fd0bf5: Pull complete
9469c4ab3de7: Pull complete
6bd1cefcc7a5: Pull complete
610e362ffa50: Pull complete
Digest: sha256:b4e56cd71c74e379198b66b3db4dc415f42e8151a18da68d1b61f55fcc7af3e0
Status: Downloaded newer image for redis:latest
docker.io/library/redis:latest
```

## Docker PS

This command (by default) shows us a list of all the running containers. We can use various flags with it.

- **-a flag:** shows us all the containers, stopped or running.

- **-l flag:** shows us the latest container.
- **-q flag:** shows only the Id of the containers.

\$ docker ps [options.]

```
C:\Users\mojha>docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS
b7e3fefc202d   ubuntu    "bash"                  11 hours ago   Up 11 hours
1b470557a372   ubuntu    "sleep 1000000"         11 hours ago   Up 11 hours
9088b39c68dd   docker/getting-started "/docker-entrypoint...." 11 hours ago   Up 11 hours   0
```

## Docker Stop

This command allows you to stop a container if it has crashed or you want to switch to another one.

\$ docker stop <container\_ID>

```
C:\Users\mojha>docker stop b7e3fefc202d
b7e3fefc202d
```

## Docker Start

Suppose you want to start the stopped container again, you can do it with the help of this command.

\$ docker start <container\_ID>

```
C:\Users\mojha>docker stop b7e3fefc202d
b7e3fefc202d
```

## Docker rm

To delete a container. By default when a container is created, it gets an ID as well as an imaginary name such as confident\_boyd, heuristic\_villani, etc. You can either mention the container name or its ID.

Some important flags:

- **-f flag:** remove the container forcefully.
- **-v flag:** remove the volumes.
- **-l flag:** remove the specific link mentioned.

\$ docker rm {options} <container\_name or ID>

```
C:\Users\mojha>docker rm b7e3fefc202d
b7e3fefc202d
```

## Docker RMI

To delete the image in docker. You can delete the images which are useless from the docker local storage so you can free up the space

```
docker rmi <image ID/ image name>
```

## Docker Images

Lists all the pulled images which are present in our system.

```
$ docker images
```

```
C:\Users\mojha>docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
redis                latest          9da089657551   4 days ago     117MB
docker/getting-started latest          cb90f98fd791   5 months ago   28.8MB
docker101tutorial    latest          9a419a57dcb8   6 months ago   28.8MB
ubuntu               latest          2b4cba85892a   6 months ago   72.8MB
alpine/git            latest          c6b70534b534   10 months ago  27.4MB
hello-world           latest          feb5d9fea6a5   12 months ago  13.3kB
```

## Docker exec

This command allows us to run new commands in a running container. This command only works until the container is running, after the container restarts, this command does not restart.

Some important flags:

- **-d flag:** for running the commands in the background.
- **-i flag:** it will keep STDIN open even when not attached.
- **-e flag:** sets the environment variables

```
$ docker exec {option
```

```
C:\Users\mojha>docker run --name ubuntu_bash --rm -i -t ubuntu bash
root@4396324406f8: /# _

C:\Users\mojha>docker exec -d ubuntu_bash touch /tmp/execWorks

C:\Users\mojha>docker exec -it ubuntu_bash bash
root@4396324406f8: /#
```

## Docker Ports (Port Mapping)

In order to access the [docker container](#) from the outside world, we have to map the port on our host( Our laptop for example), to the port on the container. This is where port mapping comes into play.

```
$ docker run -d -p <port_on_host>  
<port_on_container> Container_name
```

```
C:\Users\mojha>docker run -d -p 8080:8080 ubuntu  
4eb7e5205ae491780856beab5d09df3de5d4e54a07df04d764aa0734f2527c85
```

## Docker Login

The Docker login command will help you to authenticate with the Docker hub by which you can push and pull your images.

```
docker login
```

## Docker Push

Once you build your own customized image by using Dockerfile you need to store the image in the remote registry which is DockerHub for that you need to push your image by using the following command. [To know more about How to Push a Container Image to a Docker Repository?](#)

```
docker push <Image name/Image ID>
```

## Docker Build

The docker build command is used to build the docker images with the help of [Dockerfile](#).

```
docker build -t image_name:tag .
```

In the place of **image\_name** use the name of the image you build with and give the **tag number** and . “**dot**” represents the current directory.

## Docker Stop

You can stop and start the docker containers where you can do the maintenance for containers. To stop and start specific containers you can use the following commands.

```
docker stop container_name_or_id
```

## Stop Multiple Containers

Instead of stopping a single container. You can stop multiple containers at a time by using the following commands.

```
docker stop container1 container2 container3
```

## Docker Restart

While running the containers in Docker you may face some errors and containers fails to start. You can restart the containers to resolve the containers by using the following commands.

```
docker restart container_name_or_id
```

### **Docker Inspection**

Docker containers will run into some errors in real time to debug the container's errors you can use the following commands.

```
docker inspect container_name_or_id
```

### **Docker Commit command**

After running the containers by using the current image you can make the updates to the containers by interacting with the containers from those containers you can create an image by using the following commands.

```
docker commit container_name_or_id new_image_name:tag
```

### **Docker Basic Command**

Following are the some of the docker basic commands

1. **docker images:** Docker images will list all the images which are pulled or built in that docker host.
2. **docker pull:** Docker pull will the docker images from the dockerhub.
3. **docker run:** Docker run will run the docker image as an container.
4. **docker ps:** Docker run will list all the containers which are running in the docker host.
5. **docker stop:** Docker stop will stop the docker container which are already running.
6. **docker rm:** Docker rm command will remove the containers which are in the stop condition.

### **Docker Commands List**

Following are the docker commands which listed form build and Docker image to running it an Docker container and attaching the docker volumes to it.

### **Docker Image Command**

1. **docker build command:** It will build Docker images by using the **Dockerfile**.
2. **docker pull command:** Docker pull command will pull the **Docker image** whcih is availible in the **dockerhub**.
3. **docker images command:** It will list all the images which are pulled and build in the docker host.
4. **docker inspect command:** It will help to debug the docker image if any errors occurred while building an image or pulling the image.

5. **docker push command:** Docker command will push the docker image into the Dockerhub.
6. **docker save command:** It will save the docker image in the form of dockerfile.
7. **docker rmi command:** It will remove the docker image.

#### **Docker Container Command**

1. **docker attach command:** Connecting to an Existing Container.
2. **docker ps command:** To list the running containers.
3. **docker container inspect infinite Command:** To Inspect the Docker containers.
4. **docker exec command:** To execute the commands in the running containers.
5. **docker cp command:** To copy the file from docker host to the docker containers,