

Docker Commands – Complete List

Docker Commands

This tutorial lists and provides examples for all docker commands.

- [Docker Commands](#)
 - [Docker Version](#)
- [Docker Image Commands](#)
 - [List all Docker Images](#)
 - [Create a Docker Image](#)
 - [Run a Docker image](#)
 - [Save Docker Image to .tar file](#)
 - [Remove a Docker image](#)
- [Docker Container Commands](#)
 - [Show Running Containers](#)
 - [Show All Containers](#)
 - [Show Latest Created Container](#)
 - [Stop Container](#)
 - [Remove Container](#)
 - [Update CPU shares for container](#)

Quick list of Docker Commands

- `docker version` – Echoes Client's and Server's Version of Docker
- `docker images` – List all Docker images
- `docker build <image>` – Builds an image from a Docker file
- `docker save <path> <image>` – Saves Docker image to .tar file specified by path
- `docker run` – Runs a command in a new container.
- `docker start` – Starts one or more stopped containers
- `docker stop <container_id>` – Stops container
- `docker rmi <image>` – Removes Docker image
- `docker rm <container_id>` – Removes Container
- `docker pull` – Pulls an image or a repository from a registry
- `docker push` – Pushes an image or a repository to a registry
- `docker export` – Exports a container's filesystem as a tar archive
- `docker exec` – Runs a command in a run-time container
- `docker ps` – Show running containers
- `docker ps -a` – Show all containers
- `docker ps -l` – Show latest created container
- `docker search` – Searches the Docker Hub for images

- `docker search` – Searches the Docker Hub for images
- `docker attach` – Attaches to a running container
- `docker commit` – Creates a new image from a container's changes

Docker Commands

Following are examples for each of the docker commands

Docker Version

```
$ docker version
```

```
root@arjun-VPCEH26EN:~# docker version
Client:
Version:      17.05.0-ce
API version:  1.29
Go version:   go1.7.5
Git commit:   89658be
Built:        Thu May  4 22:10:54 2017
OS/Arch:     linux/amd64

Server:
Version:      17.05.0-ce
API version:  1.29 (minimum version 1.12)
Go version:   go1.7.5
Git commit:   89658be
Built:        Thu May  4 22:10:54 2017
OS/Arch:     linux/amd64
Experimental: false
```

Docker Image Commands

List all Docker Images

```
$ docker images
```

```
root@arjun-VPCEH26EN:~# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
python               latest             26acbad26a2c       5 days ago        690MB
java                 8                  d23bdf5b1b1b       8 months ago      643MB
hello-world         latest             c54a2cc56cbb       14 months ago     1.85kB
```

Create a Docker Image

```
$ docker build -t <application_name> .
```

The directory should contain Dockerfile, from which you are running the command in Terminal.

```
root@arjun-VPCEH26EN:/home/arjun/workspace/docker/java-application# docker build -t java
Sending build context to Docker daemon 3.072kB
Step 1/5 : FROM java:8
---> d23bdf5b1b1b
Step 2/5 : COPY . /home/arjun/workspace/docker/java
---> 81af33906fe4
Removing intermediate container 7568d6b873a2
Step 3/5 : WORKDIR /home/arjun/workspace/docker/java
---> 8da95950d05c
Removing intermediate container 839d56d42bdf
Step 4/5 : RUN javac HelloWorld.java
---> Running in 9c9eb847d3e1
---> 7b122f467725
Removing intermediate container 9c9eb847d3e1
Step 5/5 : CMD java HelloWorld
---> Running in 8fccd7a65ce1
---> 0be5de5c6f51
Removing intermediate container 8fccd7a65ce1
Successfully built 0be5de5c6f51
Successfully tagged java-application:latest
root@arjun-VPCEH26EN:/home/arjun/workspace/docker/java-application#
```

Run a Docker image

```
$ docker run <docker_image_name>
```

```
root@arjun-VPCEH26EN:~# docker run java-application
HelloWorld from Java Application running in Docker.
```

Save Docker Image to .tar file

```
$ docker save -o <complete_tar_file_path> <docker_image_name>
```

```
root@arjun-VPCEH26EN:~# docker save -o /home/arjun/java-application.tar java-application
root@arjun-VPCEH26EN:~# cd /home/arjun/
root@arjun-VPCEH26EN:/home/arjun# ls java-appl*
java-application.tar
```

Remove a Docker image

```
$ docker rmi <docker_image_id>
```

When you list the docker images, you get the image id under IMAGE ID column.

```
root@arjun-VPCEH26EN:/home/arjun# docker rmi java-application
```

```
Untagged: java-application:latest
Deleted: sha256:0be5de5c6f51ffceb18078c42d6e11d94cf844ac1e06841c9bf993a9718cc325
Deleted: sha256:7b122f46772561d5ccd27c5d5a9524c9de4071f3a22234fb2eaa709d42e874b8
Deleted: sha256:2de3fc280f40633a189f24c154faa7c136ba8afbffb9238b8342f06b8bb5d461
```

Docker Container Commands

Show Running Containers

```
$ docker ps
```

```
root@arjun-VPCEH26EN:~# docker ps
CONTAINER ID      IMAGE               COMMAND             CREATED           STATUS
```

Show All Containers

```
$ docker ps -a
```

```
root@arjun-VPCEH26EN:~# docker ps -a
CONTAINER ID      IMAGE               COMMAND             CREATED           STATUS
743.6.033ea0     java-application   "java HelloWorld"  2 minutes ago    Exited (
3b76a109a28f     java-application   "java HelloWorld"  5 minutes ago    Exited (
```

Show Latest Created Container

```
$ docker ps -l
```

```
root@arjun-VPCEH26EN:~# docker ps -l
CONTAINER ID      IMAGE               COMMAND             CREATED           STATUS
743.6.033ea0     java-application   "java HelloWorld"  4 minutes ago    Exited (
```

Stop Container

```
$ docker stop <container_id>
```

```
root@arjun-VPCEH26EN:/home/arjun# docker stop 743.6.033ea0
743.6.033ea0
```

Remove Container

```
$ docker rm <container_id>
```

```
root@arjun-VPCEH26EN:/home/arjun# docker rm 743.6.033ea0  
743.6.033ea0
```

Conclusion

In this [Docker Tutorial](#) – Docker Commands, we have learnt about all docker commands with examples.

Home

- ◆ [Docker Tutorial](#)
- ◆ [Docker - Install on Ubuntu](#)
- ◆ [Docker - Install on Windows](#)
- ◆ [Docker Terminal](#)
- ⇒ **[Docker Commands](#)**
- ◆ [Docker Architecture](#)

Docker Image Building

- ◆ [Docker Image with Java Application](#)
- ◆ [Docker Image with Python Application](#)
- ◆ [Remove Docker Image](#)