

1 Finding the version

One of the first things you want to know is how to find the installed docker version.

```
geekflare@geekflare:/home/geekflare$ docker --version
```

```
Docker version 18.09.6, build 481bc77
```

2 Downloading image

Let's say you need to pull the docker image from [dockerhub](#) (docker repository). The following example of pulling the Apache HTTP server image.

```
geekflare@geekflare:/home/geekflare$ docker pull httpd
```

```
Using default tag: latest
```

```
latest: Pulling from library/httpd
```

```
f5d23c7fed46: Pull complete
```

```
b083c5fd185b: Pull complete
```

```
bf5100a89e78: Pull complete
```

```
98f47fcaa52f: Pull complete
```

```
622a9dd8cfed: Pull complete
```

```
Digest: sha256:8bd76c050761610773b484e411612a31f299dbf7273763103edbda82acd73642
```

```
Status: Downloaded newer image for httpd:latest
```

```
geekflare@geekflare:/home/geekflare$
```

3 Images

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

```
geekflare@geekflare:/home/geekflare$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
httpd	latest	ee39f68eb241	2 days ago	154MB
hello-world	latest	fce289e99eb9	6 months ago	1.84kB
sequenceiq/hadoop-docker	2.7.0	789fa0a3b911	4 years ago	1.76GB

4 Run

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

```
geekflare@geekflare:/home/geekflare$ docker run -it -d httpd
```

```
09ca6feb6efc0578951a3e2557ed5855b2edda39a795d9703eb54d975930fe6e
```

5 What's running?

`ps` lists all the docker containers are running with container details.

```
geekflare@geekflare:/home/geekflare$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORT
09ca6feb6efc	httpd	"httpd-foreground"	36 seconds ago	Up 33 seconds	80/t

As you can see, the Apache server is running in this docker container.

6 ps -a

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

```
geekflare@geekflare:/home/geekflare$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
09ca6feb6efc	httpd	"httpd-foreground"	51 seconds ago	Up 49 s
2f6fb3381078	sequenceiq/hadoop-docker:2.7.0	"/etc/bootstrap.sh -d"	2 weeks ago	Exited
9f397feb3a46	sequenceiq/hadoop-docker:2.7.0	"/etc/bootstrap.sh -..."	2 weeks ago	Exited
9b6343d3b5a0	hello-world	"/hello"	2 weeks ago	Exited

7 exec

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

```
geekflare@geekflare:/home/geekflare$ 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#
```

Type *exit* and press enter to come out of the container.

8 Removing container

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

```
geekflare@geekflare:/home/geekflare$ docker rm 9b6343d3b5a0
9b6343d3b5a0
```

Run the below command to check if the container got removed or not.

```
geekflare@geekflare:/home/geekflare$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
09ca6feb6efc	httpd	"httpd-foreground"	About a minute ago	Up Abo
2f6fb3381078	sequenceiq/hadoop-docker:2.7.0	"/etc/bootstrap.sh -d"	2 weeks ago	Exited
9f397feb3a46	sequenceiq/hadoop-docker:2.7.0	"/etc/bootstrap.sh -..."	2 weeks ago	Exited

9 Removing image

Remove the docker image with the docker image id mentioned in the command

```
geekflare@geekflare:/home/geekflare$ docker rmi fce289e99eb9
Untagged: hello-world:latest
Untagged: hello-world@sha256:41a65640635299bab090f783209c1e3a3f11934cf7756b09cb2f1e02147c6ed8
Deleted: sha256:fce289e99eb9bca977dae136fbe2a82b6b7d4c372474c9235adc1741675f587e
Deleted: sha256:af0b15c8625bb1938f1d7b17081031f649fd14e6b233688eea3c5483994a66a3
geekflare@geekflare:/home/geekflare$
```

10 Restart Docker

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

```
geekflare@geekflare:/home/geekflare$ docker restart 09ca6feb6efc
```

```
09ca6feb6efc
```

Run the command below and check the STATUS parameter to verify if the container started recently.

```
geekflare@geekflare:/home/geekflare$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORT
09ca6feb6efc	httpd	"httpd-foreground"	6 minutes ago	Up 9 seconds	80/t

11 Stopping Docker

Stop a container with container id mentioned in the command.

```
geekflare@geekflare:/home/geekflare$ docker stop 09ca6feb6efc
```

```
09ca6feb6efc
```

Run the below command to check if the container is still running or it has stopped.

```
geekflare@geekflare:/home/geekflare$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
--------------	-------	---------	---------	--------	-------

12 Starting Docker

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

```
geekflare@geekflare:/home/geekflare$ docker start 09ca6feb6efc
```

```
09ca6feb6efc
```

Run the command below to check if the container started or not.

```
geekflare@geekflare:/home/geekflare$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORT
09ca6feb6efc	httpd	"httpd-foreground"	8 minutes ago	Up 3 seconds	80/t

13 Kill

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

```
geekflare@geekflare:/home/geekflare$ docker kill 09ca6feb6efc

09ca6feb6efc
```

Run the below command to see if the container got killed or not.

```
geekflare@geekflare:/home/geekflare$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
					

14 Commit

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.

```
geekflare@geekflare:/home/geekflare$ docker commit 09ca6feb6efc geekflare/httpd_image

sha256:d1933506f4c1686ab1a1ec601b1a03a17b41decbc21d8acd893db090a09bb31c
```

15 Login

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

```
geekflare@geekflare:/home/geekflare$ 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

Username: geekflare

Password:

Configure a credential helper to remove this warning. See

<https://docs.docker.com/engine/reference/commandline/login/#credentials-store>

Login Succeeded

16 Push

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

```
geekflare@geekflare:/home/geekflare$ docker push geekflare/httpd_image
```

The push refers to repository [docker.io/geekflare/httpd_image]

```
734d9104a6a2: Pushed
```

```
635721fc6973: Mounted from library/httpd
```

```
bea448567d6c: Mounted from library/httpd
```

```
bfaa5f9c3b51: Mounted from library/httpd
```

```
9d542ac296cc: Mounted from library/httpd
```

```
d8a33133e477: Mounted from library/httpd
```

```
latest: digest: sha256:3904662761df9d76ef04ddfa5cfab764b85e3eedaf10071cfbe2bf77254679ac size: 1574
```

//

17 Docker network

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

```
geekflare@geekflare:/home/geekflare$ docker network ls
```

NETWORK ID	NAME	DRIVER	SCOPE
85083e766f04	bridge	bridge	local
f51d1f3379e0	host	host	local
5e5d9a192c00	none	null	local

//

There are several other docker network commands.

```
geekflare@geekflare:/home/geekflare$ 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.

//

18 Docker info

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

```
geekflare@geekflare:/home/geekflare$ docker info
```

Containers: 3

Running: 1

Paused: 0

Stopped: 2

Images: 3

Server Version: 18.09.6

Storage Driver: overlay2

Backing Filesystem: extfs

Supports d_type: true

Native Overlay Diff: true

Logging Driver: json-file

Cgroup Driver: cgroupfs

Plugins:

Volume: local

Network: bridge host macvlan null overlay

Log: awslogs fluentd gcplogs gelf journald json-file local logentries splunk syslog

Swarm: inactive

Runtimes: runc

Default Runtime: runc

Init Binary: docker-init

containerd version: bb71b10fd8f58240ca47fbb579b9d1028eea7c84

runc version: 2b18fe1d885ee5083ef9f0838fee39b62d653e30

init version: fec3683

Security Options:

apparmor

seccomp

Profile: default

Kernel Version: 4.18.0-25-generic

Operating System: Ubuntu 18.10

OSType: linux

Architecture: x86_64

CPUs: 1

Total Memory: 4.982GiB

Name: geekflare

ID: RBCP:YGAP:QG6H:B6XH:JCT2:DTI5:AYJA:M44Z:ETRP:6T06:OPAY:KLNJ

Docker Root Dir: /var/lib/docker

Debug Mode (client): false

Debug Mode (server): false

Username: geekflare

Registry: https://index.docker.io/v1/

Labels:

Experimental: false

Insecure Registries:

127.0.0.0/8

Live Restore Enabled: false

19 Copying file

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/

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

[sudo] password for geekflare:
```

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

```
geekflare@geekflare:/home/geekflare$ ls

Desktop  Documents  example  examples.desktop  httpd.pid  nginx_new.yml  nginx.yml
```

20 Checking history

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

```
geekflare@geekflare:/home/geekflare$ docker history httpd
```

IMAGE	CREATED	CREATED BY	SIZE
ee39f68eb241	2 days ago	/bin/sh -c #(nop) CMD ["httpd-foreground"]	0B
<missing>	2 days ago	/bin/sh -c #(nop) EXPOSE 80	0B
<missing>	2 days ago	/bin/sh -c #(nop) COPY file:c432ff61c4993ecd...	138B
<missing>	4 days ago	/bin/sh -c set -eux; savedAptMark="\$(apt-m...	49.1MB
<missing>	4 days ago	/bin/sh -c #(nop) ENV HTTPD_PATCHES=	0B
<missing>	4 days ago	/bin/sh -c #(nop) ENV HTTPD_SHA256=b4ca9d05...	0B
<missing>	4 days ago	/bin/sh -c #(nop) ENV HTTPD_VERSION=2.4.39	0B
<missing>	4 days ago	/bin/sh -c set -eux; apt-get update; apt-g...	35.4MB
<missing>	4 days ago	/bin/sh -c #(nop) WORKDIR /usr/local/apache2	0B
<missing>	4 days ago	/bin/sh -c mkdir -p "\$HTTPD_PREFIX" && chow...	0B
<missing>	4 days ago	/bin/sh -c #(nop) ENV PATH=/usr/local/apach...	0B

<missing>	4 days ago	/bin/sh -c #(nop) ENV HTTPD_PREFIX=/usr/loc...	0B
<missing>	5 days ago	/bin/sh -c #(nop) CMD ["bash"]	0B
<missing>	5 days ago	/bin/sh -c #(nop) ADD file:71ac26257198ecf6a...	69.2MB

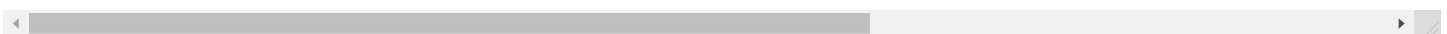


21 Checking logs

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

```
geekflare@geekflare:/home/geekflare$ docker logs 09ca6feb6efc
```

```
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2.
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2.
[Mon Jul 15 14:01:55.400472 2019] [mpm_event:notice] [pid 1:tid 140299791516800] AH00489: Apache/2.4.39 (
[Mon Jul 15 14:01:55.400615 2019] [core:notice] [pid 1:tid 140299791516800] AH00094: Command line: 'httpd
[Mon Jul 15 14:08:36.798229 2019] [mpm_event:notice] [pid 1:tid 140299791516800] AH00491: caught SIGTERM,
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2.
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2.
[Mon Jul 15 14:08:38.259870 2019] [mpm_event:notice] [pid 1:tid 139974087980160] AH00489: Apache/2.4.39 (
[Mon Jul 15 14:08:38.260007 2019] [core:notice] [pid 1:tid 139974087980160] AH00094: Command line: 'httpd
[Mon Jul 15 14:09:01.540647 2019] [mpm_event:notice] [pid 1:tid 139974087980160] AH00491: caught SIGTERM,
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2.
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2.
[Mon Jul 15 14:10:43.782606 2019] [mpm_event:notice] [pid 1:tid 140281554879616] AH00489: Apache/2.4.39 (
[Mon Jul 15 14:10:43.782737 2019] [core:notice] [pid 1:tid 140281554879616] AH00094: Command line: 'httpd
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2.
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2.
[Mon Jul 15 14:14:08.270906 2019] [mpm_event:notice] [pid 1:tid 140595254346880] AH00489: Apache/2.4.39 (
[Mon Jul 15 14:14:08.272628 2019] [core:notice] [pid 1:tid 140595254346880] AH00094: Command line: 'httpd
```



22 Searching image

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

```
geekflare@geekflare:/home/geekflare$ docker search hadoop
```

NAME	DESCRIPTION	STARS	OFFICIAL
sequenceiq/hadoop-docker	An easy way to try Hadoop	611	
uhopper/hadoop	Base Hadoop image with dynamic configuration...	98	
harisekhon/hadoop	Apache Hadoop (HDFS + Yarn, tags 2.2 - 2.8)	54	
bde2020/hadoop-namenode	Hadoop namenode of a hadoop cluster	22	
kiwenlau/hadoop	Run Hadoop Cluster in Docker Containers	19	
izone/hadoop	Hadoop 2.8.5 Ecosystem fully distributed, Ju...	14	
uhopper/hadoop-namenode	Hadoop namenode	9	
bde2020/hadoop-datanode	Hadoop datanode of a hadoop cluster	9	
singularities/hadoop	Apache Hadoop	8	
uhopper/hadoop-datanode	Hadoop datanode	7	
harisekhon/hadoop-dev	Apache Hadoop (HDFS + Yarn) + Dev Tools + Gi...	6	

23 Updating configuration

Update container configurations. This shows all the update options.

```
geekflare@geekflare:/home/geekflare$ docker update --help
```

```
Usage:  docker update [OPTIONS] CONTAINER [CONTAINER...]
```

Update configuration of one or more containers

Options:

```
--blkio-weight uint16      Block IO (relative weight), between 10 and 1000, or 0 to disable
                             (default 0)
```

```
--cpu-period int           Limit CPU CFS (Completely Fair Scheduler) period
```

```
--cpu-quota int            Limit CPU CFS (Completely Fair Scheduler) quota
```

<code>--cpu-rt-period int</code>	Limit the CPU real-time period in microseconds
<code>--cpu-rt-runtime int</code>	Limit the CPU real-time runtime in microseconds
<code>-c, --cpu-shares int</code>	CPU shares (relative weight)
<code>--cpus decimal</code>	Number of CPUs
<code>--cpuset-cpus string</code>	CPUs in which to allow execution (0-3, 0,1)
<code>--cpuset-mems string</code>	MEMs in which to allow execution (0-3, 0,1)
<code>--kernel-memory bytes</code>	Kernel memory limit
<code>-m, --memory bytes</code>	Memory limit
<code>--memory-reservation bytes</code>	Memory soft limit
<code>--memory-swap bytes</code>	Swap limit equal to memory plus swap: '-1' to enable unlimited swap
<code>--restart string</code>	Restart policy to apply when a container exits

Run the below command to update the CPU configuration of docker container with container id mentioned in the command.

```
geekflare@geekflare:/home/geekflare$ docker update -c 1 2f6fb3381078
2f6fb3381078
```

24 Creating volume

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

```
geekflare@geekflare:/home/geekflare$ docker volume create
7e7bc886f69bb24dbdbf19402e31102a25db91bb29c56cca3ea8b0c611fd9ad0
```

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

```
geekflare@geekflare:/home/geekflare$ docker volume ls

DRIVER          VOLUME NAME
local           7e7bc886f69bb24dbdbf19402e31102a25db91bb29c56cca3ea8b0c611fd9ad0
```

25 Installing plugin

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

```
geekflare@geekflare:/home/geekflare$ docker plugin install vieux/sshfs DEBUG=1
```

Plugin "vieux/sshfs" is requesting the following privileges:

- network: [host]
- mount: [/var/lib/docker/plugins/]
- mount: []
- device: [/dev/fuse]
- capabilities: [CAP_SYS_ADMIN]

Do you grant the above permissions? [y/N] y

latest: Pulling from vieux/sshfs

52d435ada6a4: Download complete

Digest: sha256:1d3c3e42c12138da5ef7873b97f7f32cf99fb6edde75fa4f0bcf9ed277855811

Status: Downloaded newer image for vieux/sshfs:latest

Installed plugin vieux/sshfs

Run the below command to list the docker plugins.

```
geekflare@geekflare:/home/geekflare$ docker plugin ls
```

ID	NAME	DESCRIPTION	ENABLED
2a32d1fb95af	vieux/sshfs:latest	sshFS plugin for Docker	true

26 Logout

Logging out from dockerhub.

```
geekflare@geekflare:/home/geekflare$ docker logout
```

Removing login credentials for https://index.docker.io/v1/

Conclusion

I hope you have got a fair understanding of docker commands by now. Try out those commands in your dev or lab environment to practice and learn.

If you are interested in learning Docker and Kubernetes, then check out this [online course](#).

TAGS: DevOps



Thanks to our sponsors.

More Great Reading on Geekflare

READ →

[Easiest Way to Find Vulnerabilities in Docker Images Using Snyk](#)

READ →

12 Learning Resources for Terraform Certification

READ →

8 Cloud-based Kubernetes and Docker Monitoring Solutions

READ →

IDE as a Code with GitHub Codespaces in Visual Studio Code

READ →

OpenShift vs. Kubernetes: Advantages, Differences and Which One Should You Choose?

READ →

5 Backend Solutions for Web and Mobile Applications [Firebase Alternatives]

Power Your Business

Here are some of the tools and services to help your business grow.

Netsparker

Netsparker uses the Proof-Based Scanning™ to automatically verify the identified vulnerabilities and generate actionable results within just hours.

[TRY NETSPARKER](#) →

Brightdata

Web scraping, residential proxy, proxy manager, web unlocker, search engine crawler, and all you need to collect web data.

[TRY BRIGHTDATA](#) →

+400 More Awesome Resources

 English  Français  Español  Deutsch

JOIN GEEKFLARE NEWSLETTER

Geekflare Newsletter is for anyone from beginner to experienced professionals who wants to keep their knowledge up-to-date.

First Name *

Email Address *

Interested in

☒ DevOps

☒ Growing Business

Join

EXPLORE

Tech Articles

Finance Articles

Geekflare Tools

Geekflare Compiler

Geekflare Assist
Geekflare API

COMPANY

- About
- Advertise
- Hiring
- Terms
- Privacy
- Disclosure
- Sitemap
- RSS Feed