

How to Manage a Container in Docker?

written by sysadmin | 19 March 2025

[The previous article](#) explained how to install Docker on Linux. This article will explain how to manage a container in Docker.

Problem

How to manage a container in Docker?

Solution

To manage a container in Docker, you have to remember basic Docker commands. And here are the basic Docker commands:

1. Search for container images

To run containers in Docker, we need a Docker image. A Docker image is an immutable (unchangeable) file that contains the source code, libraries, dependencies, tools, and other files needed for an application to run. The place to store Docker images is known as the Docker registry, which by default uses the Docker Hub located at hub.docker.com. If you are looking for a container image in Docker, use the format below:

```
docker search container_name
```

For example, if you want to find an nginx image, then use the command below:

```
docker search nginx
```

```

sysadmin@ubuntu2404:~$ docker search nginx
NAME                                DESCRIPTION                                STARS    OFFICIAL
nginx                               Official build of Nginx.                  20651    [OK]
nginx/nginx-ingress                NGINX and NGINX Plus Ingress Controllers fo... 100
nginx/nginx-prometheus-exporter    NGINX Prometheus Exporter for NGINX and NGIN... 48
nginx/unit                         This repository is retired, use the Docker o... 65
nginx/nginx-ingress-operator       NGINX Ingress Operator for NGINX and NGINX P... 2
nginx/nginx-quick-qnq              NGINX QUIC interop                        1
nginx/unit-preview                 Unit preview features                     0
nginx/nginxaas-loadbalancer-kubernetes 0
bitnami/nginx                     Bitnami container image for NGINX          196
ubuntu/nginx                      Nginx, a high-performance reverse proxy & we... 127
bitnamicharts/nginx               Bitnami Helm chart for NGINX Open Source     0
rancher/nginx                     2
kasmweb/nginx                     An Nginx image based off nginx:alpine and in... 8
linuxserver/nginx                 An Nginx container, brought to you by LinuxS... 227
redash/nginx                      Pre-configured nginx to proxy linked contain... 3
dtagdevsec/nginx                  T-Pot Nginx                              0
vmware/nginx                      2
paketobuildpacks/nginx            0
chainguard/nginx                   Build, ship and run secure software with Cha... 4
gluufederation/nginx              A customized NGINX image containing a consu... 1
intel/nginx                       0
droidwiki/nginx                   0
circleci/nginx                    This image is for internal use              2
corpusops/nginx                   https://github.com/corpusops/docker-images/ 1
antrea/nginx                      Nginx server used for Antrea e2e testing    0
sysadmin@ubuntu2404:~$

```

Searching the nginx image

2. Download the Docker image

To download the Docker image, use the following format:

```
docker image pull image_name:tag_version
```

where the tag_version is the version of the image, and if you don't write the tag, it is considered that you want to install the latest version of the image. For example, if you want to download the newest version of the nginx image, use the command:

```
docker image pull nginx
```

```
sysadmin@ubuntu2404:~$ docker image pull nginx
Using default tag: latest
latest: Pulling from library/nginx
7cf63256a31a: Pull complete
bf9acace214a: Pull complete
513c3649bb14: Pull complete
d014f92d532d: Pull complete
9dd21ad5a4a6: Pull complete
943ea0f0c2e4: Pull complete
103f50cb3e9f: Pull complete
Digest: sha256:9d6b58feebd2dbd3c56ab5853333d627cc6e281011cfd6050fa4bcf2072c9496
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
sysadmin@ubuntu2404:~$
```

Download the nginx image

But if you want to download nginx with a certain version, for example, version 1.27.2, then use the command:

```
docker image pull nginx:1.27.2
```

```
sysadmin@ubuntu2404:~$ docker pull nginx:1.27.2
1.27.2: Pulling from library/nginx
2d429b9e73a6: Pull complete
9b1039c85176: Pull complete
9ad567d3b8a2: Pull complete
773c63cd62e4: Pull complete
1d2712910bdf: Pull complete
4b0adc47c460: Pull complete
171eebbdf235: Pull complete
Digest: sha256:bc5eac5eafc581aeda3008b4b1f07ebba230de2f27d47767129a6a905c84f470
Status: Downloaded newer image for nginx:1.27.2
docker.io/library/nginx:1.27.2
sysadmin@ubuntu2404:~$
```

Download the nginx with a certain version

3. List the Docker image(s)

To display the Docker image that you have downloaded, use the command below:

```
docker image ls
```

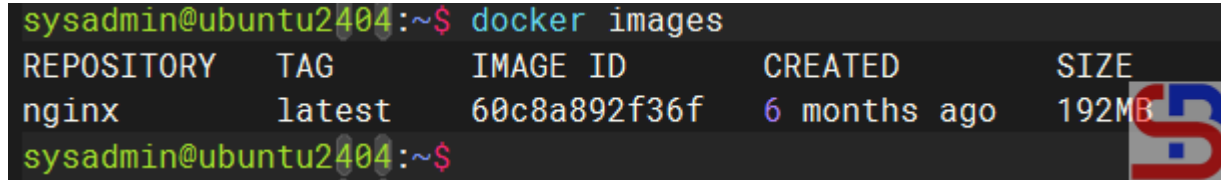
```
sysadmin@ubuntu2404:~$ docker image ls
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
nginx         latest   b52e0b094bc0   5 weeks ago   192MB
sysadmin@ubuntu2404:~$
```

List the Docker images

Or you can use the command below:

`docker images`

```
sysadmin@ubuntu2404:~$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
nginx         latest    60c8a892f36f   6 months ago   192MB
```



List the Docker images

4. Create a container

You can create the container using the format:

`docker container create --name container_name image_name:tag`

For example, if you want to create a container with the name `webapp1`, which contains the `nginx` application, then use the command below:

`docker container create --name webapp1 nginx`

When you use this command, Docker will first check whether the `nginx` image is on the server. If the image is not on the server, then Docker will download the `nginx` image, and after that, it will create an `nginx` container, and the image will remain on your server, as shown in the image below:

```

sysadmin@ubuntu2404:~$ docker container create --name webapp1 nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
7cf63256a31a: Pull complete
bf9acace214a: Pull complete
513c3649bb14: Pull complete
d014f92d532d: Pull complete
9dd21ad5a4a6: Pull complete
943ea0f0c2e4: Pull complete
103f50cb3e9f: Pull complete
Digest: sha256:9d6b58feebd2dbd3c56ab5853333d627cc6e281011cfd6050fa4bcf2072c9496
Status: Downloaded newer image for nginx:latest
b6b73bdbb4e39e83bdd8090478868b41b895f847910d65d44b744955562c4cce
sysadmin@ubuntu2404:~$

```



Create the container

You can use an image to create multiple containers as long as the container names are different, as in the image below:

```

sysadmin@ubuntu2404:~$ docker image ls
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
nginx         latest    b52e0b094bc0   5 weeks ago    192MB
sysadmin@ubuntu2404:~$
sysadmin@ubuntu2404:~$ docker container create --name webapp1 nginx
9fbc044864d767f3ffc2d321654e3f66db8ea9be5a1d06f8023bf2e0ffbff6d2
sysadmin@ubuntu2404:~$
sysadmin@ubuntu2404:~$ docker container create --name webapp2 nginx
2f22437fcb9d149c542d74ffb6dda54112e88479fecfb16171e819ee0995f006
sysadmin@ubuntu2404:~$

```



Create the containers with 1 image

5. List the status of the container(s)

To display the container status, you can use the command:

`docker ps`

```

sysadmin@ubuntu2404:~$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
sysadmin@ubuntu2404:~$

```



List the status of running Docker

Maybe you are confused about why there is no container status displayed, even though you have made 2 containers

before. Remember that the **docker ps** command only displays the ongoing container status. While the 2 containers you made had not been running, you just made a container. If you want to display all container statuses, use the command below:

```
docker ps -a
```

```
sysadmin@ubuntu2404:~$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
2f22437fcb9d	nginx	"/docker-entrypoint..."	6 minutes ago	Created		webapp2
9fbc044864d7	nginx	"/docker-entrypoint..."	6 minutes ago	Created		webapp1

```
sysadmin@ubuntu2404:~$
```

List all container statuses in Docker

6. Turn on the container

To turn on a container, you can use the format:

```
docker container start container_id/container_name
```

Usually, I use container_name instead of container_id because it's easier to memorize, so I run the following command:

```
docker container start webapp1
```

```
sysadmin@ubuntu2404:~$ docker container start webapp1
```

```
webapp1
```

```
sysadmin@ubuntu2404:~$
```

```
sysadmin@ubuntu2404:~$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
b6b73bdbb4e3	nginx	"/docker-entrypoint..."	31 minutes ago	Up 5 seconds	80/tcp	webapp1

```
sysadmin@ubuntu2404:~$
```

Turn on the container

7. Pause the container

You can pause a container with the following format:

```
docker container pause container_id/container_name
```

So, you can use the command below to pause the container:

```
docker container pause webapp1
```

```

sysadmin@ubuntu2404:~$ docker container pause webapp1
webapp1
sysadmin@ubuntu2404:~$
sysadmin@ubuntu2404:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS              PORTS          NAMES
b6b73bdbb4e3   nginx    "/docker-entrypoint...." 31 minutes ago Up 53 seconds (Paused) 80/tcp        webapp1
sysadmin@ubuntu2404:~$

```

Pause the container

To resume the container, use the following format:

```
docker containers unpause container_id/container_name
```

You can use the command below to resume the container:

```
docker container unpause webapp1
```

```

sysadmin@ubuntu2404:~$ docker container unpause webapp1
webapp1
sysadmin@ubuntu2404:~$
sysadmin@ubuntu2404:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS              PORTS          NAMES
b6b73bdbb4e3   nginx    "/docker-entrypoint...." 32 minutes ago Up About a minute   80/tcp        webapp1
sysadmin@ubuntu2404:~$

```

Resume the container

8. Run a container with a single command

As explained above, if you want to run a container, you have to download the image first, create a container, and turn on the container (see numbers 2, 4, and 6). There is a command that can summarize the three commands above, using the format below:

```
docker run -d --name container_id/container_name image_name:tag
```

where the **-d option** is to run the container in the background. So if you want to run a container with the name webapp2, which contains the nginx application, then use the command below:

```
docker run -d --name webapp3 nginx
```

```

sysadmin@ubuntu2404:~$ docker run -d --name webapp3 nginx
d03087e2d12daf32676144237aef42dfb810568cbf36dadcf0387439ad1c679b
sysadmin@ubuntu2404:~$
sysadmin@ubuntu2404:~$ docker ps -a

```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
d03087e2d12d	nginx	"/docker-entrypoint..."	6 seconds ago	Up 5 seconds	80/tcp	webapp3
2f22437fcb9d	nginx	"/docker-entrypoint..."	13 minutes ago	Created		webapp2
9fbc044864d7	nginx	"/docker-entrypoint..."	13 minutes ago	Up 12 seconds	80/tcp	webapp1

```

sysadmin@ubuntu2404:~$

```

Run the container

9. Display the size of Docker

To display how large Docker is installed on your server, use the command below:

docker system df

```

sysadmin@docker:~$ docker system df

```

TYPE	TOTAL	ACTIVE	SIZE	RECLAIMABLE
Images	2	1	234.3MB	117MB (49%)
Containers	2	2	2.19kB	0B (0%)
Local Volumes	0	0	0B	0B
Build Cache	0	0	0B	0B

```

sysadmin@docker:~$

```

Display the size of Docker

To display a Docker size in detail, use the command below:

docker system df -v

```

sysadmin@docker:~$ docker system df -v

```

Images space usage:

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE	SHARED SIZE	UNIQUE SIZE	CONTAINERS
nginx	latest	4cad75abc83d	2 months ago	192MB	74.83MB	117.2MB	2
redis	latest	65750d044ac8	3 months ago	117MB	74.83MB	42.21MB	0

Containers space usage:

CONTAINER ID	IMAGE	COMMAND	LOCAL VOLUMES	SIZE	CREATED	STATUS	NAMES
d8982eca0840	nginx	"/docker-entrypoint..."	0	1.09kB	4 minutes ago	Up 4 minutes	webapp1
8d609d92bcc7	nginx	"/docker-entrypoint..."	0	1.09kB	11 minutes ago	Up 11 minutes	nginx

Local Volumes space usage:

VOLUME NAME	LINKS	SIZE
-------------	-------	------

Build cache usage: 0B

CACHE ID	CACHE TYPE	SIZE	CREATED	LAST USED	USAGE	SHARED
----------	------------	------	---------	-----------	-------	--------

```

sysadmin@docker:~$

```

Display the size of the Docker

10. Display logs

To display logs of the running container to check something, follow the format below:

```
docker container logs container_id/container_name
```

So, run the command below to check the logs of your container:

```
docker container logs webapp1
```

```
sysadmin@ubuntu2404:~$ docker container logs webapp1
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2025/03/13 15:35:13 [notice] 1#1: using the "epoll" event method
2025/03/13 15:35:13 [notice] 1#1: nginx/1.27.4
2025/03/13 15:35:13 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)
2025/03/13 15:35:13 [notice] 1#1: OS: Linux 6.8.0-55-generic
2025/03/13 15:35:13 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2025/03/13 15:35:13 [notice] 1#1: start worker processes
2025/03/13 15:35:13 [notice] 1#1: start worker process 29
2025/03/13 15:35:13 [notice] 1#1: start worker process 30
sysadmin@ubuntu2404:~$
```

Display logs in the container

If you want to display real-time logs of the container, give an option **-f** like in the below command:

```
docker container logs -f webapp1
```

Press **Ctrl-C** to exit the log.

```

sysadmin@ubuntu2404:~$ docker container logs -f webapp1
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2025/03/13 15:35:13 [notice] 1#1: using the "epoll" event method
2025/03/13 15:35:13 [notice] 1#1: nginx/1.27.4
2025/03/13 15:35:13 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)
2025/03/13 15:35:13 [notice] 1#1: OS: Linux 6.8.0-55-generic
2025/03/13 15:35:13 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2025/03/13 15:35:13 [notice] 1#1: start worker processes
2025/03/13 15:35:13 [notice] 1#1: start worker process 29
2025/03/13 15:35:13 [notice] 1#1: start worker process 30
^Ccontext canceled
sysadmin@ubuntu2404:~$

```

Display real-time logs in the container

11. Inspect the container

To display detailed information about a container, use the following format:

```
docker inspect container_name/container_id
```

So, if you want to see the detailed information about the container that you created before, use the command below:

```
docker inspect webapp1
```

```

sysadmin@ubuntu2404:~$ docker inspect webapp1
[
  {
    "Id": "b6b73bdbb4e39e83bdd8090478868b41b895f847910d65d44b744955562c4cce",
    "Created": "2025-03-13T15:04:08.185644454Z",
    "Path": "/docker-entrypoint.sh",
    "Args": [
      "nginx",
      "-g",
      "daemon off;"
    ],
    "State": {
      "Status": "running",
      "Running": true,
      "Paused": false,
      "Restarting": false,
      "OOMKilled": false,
      "Dead": false,
      "Pid": 1748,
      "ExitCode": 0,
      "Error": "",
      "StartedAt": "2025-03-13T15:35:12.863527159Z",
      "FinishedAt": "0001-01-01T00:00:00Z"
    },
    "Image": "sha256:b52e0b094bc0e26c9eddc9e4ab7a64ce0833c3360d9b7ad4ff4132c4e03e8f7b",
    "ResolvConfPath": "/var/lib/docker/containers/b6b73bdbb4e39e83bdd8090478868b41b895f847910d65d44b744955562c4cce/resolv.conf",
    "HostnamePath": "/var/lib/docker/containers/b6b73bdbb4e39e83bdd8090478868b41b895f847910d65d44b744955562c4cce/hostname",
    "HostsPath": "/var/lib/docker/containers/b6b73bdbb4e39e83bdd8090478868b41b895f847910d65d44b744955562c4cce/hosts",
    "LogPath": "/var/lib/docker/containers/b6b73bdbb4e39e83bdd8090478868b41b895f847910d65d44b744955562c4cce/b6b73bdbb4e39e83bdd8090478868b41b895f847910d65d44b744955562c4cce-json.log",
    "Name": "/webapp1",
    "RestartCount": 0,
  }
]

```

Inspect the container

If you only want to display specific items when running the inspect command, use the following format:

```
docker container inspect container_name/container_id -f '{{json .the  
item_you_want_to_display<.sub_item> }}' | python -m json.tool
```

So if you want to display only the network section when using the Docker inspect command, use the command below:

```
docker container inspect webapp1 -f '{{json .NetworkSettings.Networks }}' |  
python3 -m json.tool
```

```
sysadmin@ubuntu2404:~$ docker container inspect webapp1 -f '{{json .NetworkSettings.Networks }}' | python3 -m json.tool  
{  
  "bridge": {  
    "IPAMConfig": null,  
    "Links": null,  
    "Aliases": null,  
    "MacAddress": "02:42:ac:11:00:02",  
    "DriverOpts": null,  
    "NetworkID": "0bf7fbd95b01135a37858f29ce5a744fcd6d39a4ccca37da437c4e45d928ba68",  
    "EndpointID": "9ce624a8d8acd63bc01229dd682639b2991c1149afae602fe2372118969b8f3b",  
    "Gateway": "172.17.0.1",  
    "IPAddress": "172.17.0.2",  
    "IPPrefixLen": 16,  
    "IPv6Gateway": "",  
    "GlobalIPv6Address": "",  
    "GlobalIPv6PrefixLen": 0,  
    "DNSNames": null  
  }  
}
```

Inspect the network of the container only

12. Stop the container(s)

To stop the container, use the format below:

```
docker container stop container_id/container_name
```

For example, if I want to stop my container, then use the command below:

```
docker container stop webapp1
```

```
sysadmin@ubuntu2404:~$ docker container stop webapp1  
webapp1  
sysadmin@ubuntu2404:~$  
sysadmin@ubuntu2404:~$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
d03087e2d12d	nginx	"/docker-entrypoint..."	4 minutes ago	Up 4 minutes	80/tcp	webapp3
2f22437fcb9d	nginx	"/docker-entrypoint..."	18 minutes ago	Created		webapp2
9fbc044864d7	nginx	"/docker-entrypoint..."	18 minutes ago	Exited (0) 3 seconds ago		webapp1

```
sysadmin@ubuntu2404:~$
```

Stop the container

You can stop all the containers running with the below command:

```
docker stop webapp1 webapp2
```

```
sysadmin@ubuntu2404:~$ docker container stop webapp1 webapp3
webapp1
webapp3
sysadmin@ubuntu2404:~$
sysadmin@ubuntu2404:~$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
d03087e2d12d	nginx	"/docker-entrypoint..."	6 minutes ago	Exited (0) 3 seconds ago		webapp3
2f22437fcb9d	nginx	"/docker-entrypoint..."	20 minutes ago	Created		webapp2
9fbc044864d7	nginx	"/docker-entrypoint..."	20 minutes ago	Exited (0) 3 seconds ago		webapp1

```
sysadmin@ubuntu2404:~$
```

Stop more than one container

Or use the below command to stop all the running containers:

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

```
sysadmin@ubuntu2404:~$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
d03087e2d12d	nginx	"/docker-entrypoint..."	9 minutes ago	Up 4 seconds	80/tcp	webapp3
2f22437fcb9d	nginx	"/docker-entrypoint..."	23 minutes ago	Created		webapp2
9fbc044864d7	nginx	"/docker-entrypoint..."	23 minutes ago	Up 4 seconds	80/tcp	webapp1

```
sysadmin@ubuntu2404:~$
sysadmin@ubuntu2404:~$ docker kill $(docker ps -q)
d03087e2d12d
9fbc044864d7
sysadmin@ubuntu2404:~$
sysadmin@ubuntu2404:~$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
d03087e2d12d	nginx	"/docker-entrypoint..."	9 minutes ago	Exited (137) 3 seconds ago		webapp3
2f22437fcb9d	nginx	"/docker-entrypoint..."	23 minutes ago	Created		webapp2
9fbc044864d7	nginx	"/docker-entrypoint..."	23 minutes ago	Exited (137) 3 seconds ago		webapp1

```
sysadmin@ubuntu2404:~$
```

Stop all running containers

13. Remove the container(s)

Before you remove the container, you have to **stop the container first**. To delete a container that's already turned off, use the format below:

```
docker container rm container_id/container_name
```

Run the command below to remove the container:

```
docker container rm webapp1
```

```

sysadmin@ubuntu2404:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS              PORTS          NAMES
d03087e2d12d   nginx    "/docker-entrypoint...." 11 minutes ago Exited (137) 2 minutes ago
2f22437fcb9d   nginx    "/docker-entrypoint...." 25 minutes ago Created
9fbc044864d7   nginx    "/docker-entrypoint...." 25 minutes ago Exited (137) 2 minutes ago
webapp1
sysadmin@ubuntu2404:~$ docker container rm webapp1
webapp1
sysadmin@ubuntu2404:~$
sysadmin@ubuntu2404:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS              PORTS          NAMES
d03087e2d12d   nginx    "/docker-entrypoint...." 12 minutes ago Exited (137) 2 minutes ago
2f22437fcb9d   nginx    "/docker-entrypoint...." 25 minutes ago Created
webapp3
webapp2
sysadmin@ubuntu2404:~$

```

Delete the container

By default, you **can't remove a container if the container is still running**. You can use the command below to delete the container even if the container is still running, but it is not recommended:

```
docker container rm -f webapp2
```

```

sysadmin@ubuntu2404:~$ docker container start webapp2
webapp2
sysadmin@ubuntu2404:~$
sysadmin@ubuntu2404:~$ docker container rm webapp2
Error response from daemon: cannot remove container "/webapp2": container is running: stop the container before removing or force remove
sysadmin@ubuntu2404:~$
sysadmin@ubuntu2404:~$ docker container rm -f webapp2
webapp2
sysadmin@ubuntu2404:~$

```

Force delete the running container

If you have a lot of containers that are no longer used and you don't want to delete them one by one, you can use the command below to delete all the unused containers:

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

```

sysadmin@ubuntu2404:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS              PORTS          NAMES
ba79cff3e0bf   nginx    "/docker-entrypoint...." 21 seconds ago Exited (0) 3 seconds ago
2edbd2c033d8   nginx    "/docker-entrypoint...." 26 seconds ago Exited (0) 3 seconds ago
webapp2
webapp1
sysadmin@ubuntu2404:~$
sysadmin@ubuntu2404:~$ docker rm $(docker ps -a -q)
ba79cff3e0bf
2edbd2c033d8
sysadmin@ubuntu2404:~$
sysadmin@ubuntu2404:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS              PORTS          NAMES
sysadmin@ubuntu2404:~$

```

Delete all the stop containers

You can also use the command below to delete all the stop containers:

docker container prune

```
sysadmin@docker:~$ docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS              PORTS          NAMES
2fd039880269   mysql     "docker-entrypoint.s..." 5 hours ago    Exited (137) 2 hours ago           db-mysql
2a4eadaffcd    nginx     "/docker-entrypoint...." 28 hours ago   Exited (0) 2 hours ago           webapp1
e6d61413d2af    nginx     "/docker-entrypoint...." 29 hours ago   Exited (0) 2 hours ago           nginx

sysadmin@docker:~$
sysadmin@docker:~$ docker container prune
WARNING! This will remove all stopped containers.
Are you sure you want to continue? [y/N] y
Deleted Containers:
2fd039880269153802f303435bf9a197fd1aefed5b96c5df0fe2a8e291266cb3
2a4eadaffcd4899dce3201f8e110489e77d5c0f6d4a9bac8af91f48a06adf35
e6d61413d2afa872ee797369775b0bbaf39d433b29c13c03538383f3bc228888

Total reclaimed space: 22.47MB
sysadmin@docker:~$
sysadmin@docker:~$ docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS          NAMES
sysadmin@docker:~$
```

Delete the stop containers using the prune command

14. Delete the image(s)

To delete the Docker image that you have already downloaded, use the format below:

```
docker image rm image_name
```

Run the image below if you want to delete the nginx image:

```
docker image rm nginx
```

```
sysadmin@ubuntu2404:~$ docker image ls
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
nginx         latest    b52e0b094bc0   5 weeks ago    192MB

sysadmin@ubuntu2404:~$
sysadmin@ubuntu2404:~$ docker rmi nginx
Untagged: nginx:latest
Untagged: nginx@sha256:9d6b58feebd2dbd3c56ab5853333d627cc6e281011cfd6050fa4bcf2072c9496
Deleted: sha256:b52e0b094bc0e26c9eddc9e4ab7a64ce0033c3360d8b7ad4ff4132c4e03e8f7b
Deleted: sha256:3c8b88c16794e3082397557e5482f5a04a6c295cec37919c65c234e1a3645e80
Deleted: sha256:d5c83383666c732fcb30d7e25c74c2e0884c262f2e497cc9f2844870980311d8
Deleted: sha256:d62b6301e685a7cdc3bb3b1508a959e4710a707ea2f680f848c19a9ad74ac6a7
Deleted: sha256:d443654bda4a04f31ba6bd39bed82a053a17f2974b401fef552e4e88d6546db8
Deleted: sha256:129409d5d363e5d5af273f0b2a90237f708ed9972f8d58a4dbcd17f1abbabe21
Deleted: sha256:a3a2912e392a24d8c7dde076a3778c6eded8839660963ac2084e051eb6931c13
Deleted: sha256:5f1ee22fffb5e68686db3dcb6584eb1c73b5570615b0f14fabb070b96117e351d

sysadmin@ubuntu2404:~$
sysadmin@ubuntu2404:~$ docker image ls
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
sysadmin@ubuntu2404:~$
```

Delete the image

However, you must know that you **can't delete the Docker image if the image is still running in the container**. So you must remove the container first before you delete the image. If you want to delete multiple Docker images, use the following format:

```
docker image rm image_name1 image_name2 ...
```

So if you want to delete the nginx image and nginx:1.27.2 at once, then use the command below:

```
docker image rm nginx nginx:1.27.2
```

```
sysadmin@ubuntu2404:~$ docker image ls
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
nginx         latest    b52e0b094bc0   5 weeks ago    192MB
nginx         1.27.2    60c8a892f36f   5 months ago    192MB
sysadmin@ubuntu2404:~$
sysadmin@ubuntu2404:~$ docker rmi nginx nginx:1.27.2
Untagged: nginx:latest
Untagged: nginx@sha256:9d6b58feebd2dbd3c56ab5853333d627cc6e281011cfd6050fa4bcf2072c9496
Deleted: sha256:b52e0b094bc0e26c9eddc9e4ab7a64ce0033c3360d8b7ad4ff4132c4e03e8f7b
Deleted: sha256:3c8b88c16794e3082397557e5482f5a04a6c295cec37919c65c234e1a3645e80
Deleted: sha256:d5c83383666c732fcb30d7e25c74c2e0884c262f2e497cc9f2844870980311d8
Deleted: sha256:d62b6301e685a7cdc3bb3b1508a959e4710a707ea2f680f848c19a9ad74ac6a7
Deleted: sha256:d443654bda4a04f31ba6bd39bed82a053a17f2974b401fef552e4e88d6546db8
Deleted: sha256:129409d5d363e5d5af273f0b2a90237f708ed9972f8d58a4dbcd17f1abbabe21
Deleted: sha256:a3a2912e392a24d8c7dde076a3778c6eded8839660963ac2084e051eb6931c13
Deleted: sha256:5f1ee22ffb5e68686db3dcb6584eb1c73b5570615b0f14fabb070b96117e351d
Untagged: nginx:1.27.2
Untagged: nginx@sha256:bc5eac5eafc581aeda3008b4b1f07ebba230de2f27d47767129a6a905c84f470
Deleted: sha256:60c8a892f36faf6c9215464005ee6fb8cf0585f70b113c0b030f6cb497a41876
Deleted: sha256:47984982982b32672d3b0cc6ebc1016e70916a8347c79765dc2ba09ed9afc97c
Deleted: sha256:f8fffe24ebb396c3e1721168923665f594d6b0ec1270700f642155fb51179cb
Deleted: sha256:ceff183e9da02c76af52712096cbe7e26e01909f827f18141058afb4f7e32db
Deleted: sha256:01c22c5216c94ae4a6285e21b0ccb6bb786d437aa7eb7d3e2de8a454115d17a8
Deleted: sha256:9a980991ece0116dad7650d5af48faa2f693f9277bfd99f4fb3c8c2ce0b4e27d
Deleted: sha256:d775439dbfb804d168b7ab8501c32013896d40d66b14944d2429778d995c7fe4
Deleted: sha256:c3548211b8264f8bfa47a6727043a64f1791b82ac965a284a7ea187e971a95e2
sysadmin@ubuntu2404:~$
sysadmin@ubuntu2404:~$ docker image ls
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
sysadmin@ubuntu2404:~$
```

Delete more than one image

And if you want to delete all the images, you can use the command below:

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

```
sysadmin@ubuntu2404:~$ docker image ls
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
nginx         latest    b52e0b094bc0   5 weeks ago    192MB
nginx         1.27.2    60c8a892f36f   5 months ago    192MB
sysadmin@ubuntu2404:~$
sysadmin@ubuntu2404:~$ docker rmi $(docker images -a -q)
Untagged: nginx:latest
Untagged: nginx@sha256:9d6b58feebd2dbd3c56ab5853333d627cc6e281011cfd6050fa4bcf2072c9496
Deleted: sha256:b52e0b094bc0e26c9eddc9e4ab7a64ce0033c3360d8b7ad4ff4132c4e03e8f7b
Deleted: sha256:3c8b88c16794e3082397557e5482f5a04a6c295cec37919c65c234e1a3645e80
Deleted: sha256:d5c83383666c732fcb30d7e25c74c2e0884c262f2e497cc9f2844870980311d8
Deleted: sha256:d62b6301e685a7cdc3bb3b1508a959e4710a707ea2f680f848c19a9ad74ac6a7
Deleted: sha256:d443654bda4a04f31ba6bd39bed82a053a17f2974b401fef552e4e88d6546db8
Deleted: sha256:129409d5d363e5d5af273f0b2a90237f708ed9972f8d58a4dbcd17f1abbabe21
Deleted: sha256:a3a2912e392a24d8c7dde076a3778c6eded8839660963ac2084e051eb6931c13
Deleted: sha256:5f1ee22fffb5e68686db3dcb6584eb1c73b5570615b0f14fabb070b96117e351d
Untagged: nginx:1.27.2
Untagged: nginx@sha256:bc5eac5eafc581aeda3008b4b1f07ebba230de2f27d47767129a6a905c84f470
Deleted: sha256:60c8a892f36faf6c9215464005ee6fb8cf0585f70b113c0b030f6cb497a41876
Deleted: sha256:47984982982b32672d3b0cc6ebc1016e70916a8347c79765dc2ba09ed9afc97c
Deleted: sha256:f8ffffef24ebb396c3e1721168923665f594d6b0ec1270700f642155fb51179cb
Deleted: sha256:cefff183e9da02c76af52712096cbe7e26e01909f827f18141058afbf4f7e32db
Deleted: sha256:01c22c5216c94ae4a6285e21b0ccb6bb786d437aa7eb7d3e2de8a454115d17a8
Deleted: sha256:9a980991ece0116dad7650d5af48faa2f693f9277bfd99f4fb3c8c2ce0b4e27d
Deleted: sha256:d775439dbfb804d168b7ab8501c32013896d40d66b14944d2429778d995c7fe4
Deleted: sha256:c3548211b8264f8bfa47a6727043a64f1791b82ac965a284a7ea187e971a95e2
sysadmin@ubuntu2404:~$
sysadmin@ubuntu2404:~$ docker image ls
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
sysadmin@ubuntu2404:~$
```

Delete all the images

Or, you can use the command below to remove the unused images:

```
docker image prune -a
```

```

sysadmin@docker:~$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
nginx         latest    4cad75abc83d   2 months ago  192MB
mysql         latest    567107cb6971   2 months ago  797MB
sysadmin@docker:~$
sysadmin@docker:~$ docker image prune -a
WARNING! This will remove all images without at least one container associated to them.
Are you sure you want to continue? [y/N] y
Deleted Images:
untagged: nginx:latest
untagged: nginx@sha256:09369da6b10306312cd908661320086bf87fbae1b6b0c49a1f50ba531fef2eab
deleted: sha256:4cad75abc83d5ca6ee22053d85850676eae657ee9d723d7bef61179e1e1e485
deleted: sha256:12dc0894b9d83988c128df9d1eda0d43198450dfbb600d3f48951a60dc83ba22
deleted: sha256:cf328fc766bc5a8b4c62d4d1a66a5fd64a012bb9c4edf00733760b50245dcc25
deleted: sha256:482a624ec9ee06ddd66621ef94544670936e5047ce55742aedc630b8f2508e45
deleted: sha256:2cabaf44a04cb066a69df1ac5fad6d7bb983767f19579e2fcc1c38ee76deaecc
deleted: sha256:dfb7b04fe3c8a2b11f1e627e3a98987fae238799f35531a03194daf1a555e618
deleted: sha256:252d6f0879cc76efb21ff5ee44a264862e6d5190693d80dcc218847e0ab1deea
deleted: sha256:ea680fbff095473bb8a6c867938d6d851e11ef0c177fce983ccc83440172bd72
untagged: mysql:latest
untagged: mysql@sha256:0596fa224cdf3b3355ce3ddbdf7ce77be27ec9e51841dfc5d2e1c8b81eea69d2
deleted: sha256:567107cb6971c25f0921ff3c2fa6b460ef636d50ca1365d987cee6bdcce3fd53
deleted: sha256:43814104558997cecebb1a4de919904bd86292b70961b4baa54452861571abb6
deleted: sha256:842f156a86f22550ee891ff2cecca451fc05c67fe2931391e2d4e24c8994748e
deleted: sha256:d1df9893c1d8755fc63f2719eae3c1576040e1097a14facf966b22922d824f
deleted: sha256:7b093f3a2a71e56f9cc9d384d7263ed516eb948eefa24c5537d505448cf7d257
deleted: sha256:1af7699f489cc2f9ef006c965fa5df6037315c07f3e954e934104c1e73bcbd43
deleted: sha256:39511577b58ee8af0cce262e1e1d18e08319fc264f41bb66d84981df43a3d3f7
deleted: sha256:0e260b5837f1fff58f7c0ca9e0f30687c2637792b8236f8d7b5e198a5a137b57
deleted: sha256:3589674506312c078c2a2e6c1493beffc8ca873b6b34c7737d9510855f6f28b4
deleted: sha256:5f19898b2782394b0f3406750e1f8a58bd3d1fa359f40c162ebd918e96c19b12
deleted: sha256:561b565cf5eba84f1729d1d097d529566c1f992937a14ac7ec12e76a4a5693d2

Total reclaimed space: 989MB
sysadmin@docker:~$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
sysadmin@docker:~$

```



Delete all the unused images using the prune command

Note

If you forget or don't know what command to use in Docker, use the following command:

```
docker --help
```

```
sysadmin@ubuntu2404:~$ docker --help
```

Usage: docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Common Commands:

run	Create and run a new container from an image
exec	Execute a command in a running container
ps	List containers
build	Build an image from a Dockerfile
pull	Download an image from a registry
push	Upload an image to a registry
images	List images
login	Authenticate to a registry
logout	Log out from a registry
search	Search Docker Hub for images
version	Show the Docker version information
info	Display system-wide information

Management Commands:

builder	Manage builds
buildx*	Docker Buildx
compose*	Docker Compose
container	Manage containers
context	Manage contexts
image	Manage images
manifest	Manage Docker image manifests and manifest lists
network	Manage networks



Using the docker help command

After that, if you want to know the options in the Docker command, then use the following format:

```
docker command --help
```

For example, if you want to know the options of the run command in Docker, then type the command below:

```
docker run --help
```

```
sysadmin@ubuntu2404:~$ docker run --help

Usage: docker run [OPTIONS] IMAGE [COMMAND] [ARG...]

Create and run a new container from an image

Aliases:
  docker container run, docker run

Options:
  --add-host list          Add a custom host-to-IP mapping (host:ip)
  --annotation map        Add an annotation to the container (passed through to the OCI runtime) (default map[])
  -a, --attach list       Attach to STDIN, STDOUT or STDERR
  --blkio-weight uint16    Block IO (relative weight), between 10 and 1000, or 0 to disable (default 0)
  --blkio-weight-device list Block IO weight (relative device weight) (default [])
  --cap-add list           Add Linux capabilities
  --cap-drop list         Drop Linux capabilities
  --cgroup-parent string   Optional parent cgroup for the container
  --cgroupns string       Cgroup namespace to use (host|private)
                          'host': Run the container in the Docker host's cgroup namespace
                          'private': Run the container in its own private cgroup namespace
                          '': Use the cgroup namespace as configured by the
                             default-cgroupns-mode option on the daemon (default)
  --cidfile string        Write the container ID to the file
  --cpu-period int        Limit CPU CFS (Completely Fair Scheduler) period
  --cpu-quota int         Limit CPU CFS (Completely Fair Scheduler) quota
  --cpu-rt-period int     Limit CPU real-time period in microseconds
  --cpu-rt-runtime int    Limit CPU real-time runtime in microseconds
```

Using the docker run help command

You can shorten all the **Docker container** commands to just the **docker** command to shorten the typing time. For example, if you want to create a container, you can use the command:

```
docker create --name webapp6 nginx
```

```
sysadmin@ubuntu2404:~$ docker create --name webapp6 nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
7cf63256a31a: Pull complete
bf9acace214a: Pull complete
513c3649bb14: Pull complete
d014f92d532d: Pull complete
9dd21ad5a4a6: Pull complete
943ea0f0c2e4: Pull complete
103f50cb3e9f: Pull complete
Digest: sha256:9d6b58feebd2dbd3c56ab5853333d627cc6e281011cfd6050fa4bcf2072c9496
Status: Downloaded newer image for nginx:latest
1dd2061cebce8d7b899ad65a99d7c25f5b5101f83ba9f7b13f3d6988cc4bcd13
sysadmin@ubuntu2404:~$
```

Using the docker create command

References

[geeksforgeeks.org](https://www.geeksforgeeks.org)
mygreatlearning.com
[youtube.com](https://www.youtube.com)
youtube.dimas-maryanto.com