

In **Docker**, if we have exited a container without stopping it, we then need to stop them manually as it has not stopped on exit. Similarly, for images, we need to delete them from top to bottom as some [containers](#) or [images](#) might be dependent on the base images, we can any time download the base image at. So it is a good idea to delete unwanted or dangling images from the current machine.

How To Delete The Images in Docker?

Remove Image

To delete the image by the ImageId/Name we can use the following command. To know more about how to build a docker image with the help of Dockerfile refer to [Concept of Dockerfile](#).

```
docker rmi <imageId/Name>
```

Force Remove Image

To force remove the docker Images by the ImageID/Name we can use the following command.

```
docker rmi -f <imageId/Name>
```

Note: We can't remove the images by force or normally while the container is running.

Dangling Images

Dangling Images are those that don't map to either the repository or the tag. The command used is to remove the dangling images. To know more about how to tag Docker images by referring to [Docker image tags](#).

```
docker image prune
```

Removing all Images

We can remove all images in the docker-machine to remove unwanted clutter and space in the system. We can anyways fetch the latest version or specific versioned image from the docker registry or from the cache.

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

```

Code
$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
python               latest              178dca62b39        10 days ago        917MB
mark                 latest              e56db3e34776        13 days ago        979MB
ubuntu               latest              2b4cba85892a        2 weeks ago        72.8MB
python               3.9-buster          f3a16526089d        3 weeks ago        867MB

Code
$ docker rmi $(docker images -q)
Untagged: python:latest
Deleted: sha256:3204faabc2f0b5e0939b8b8b29079a2a330c38dee92a22482a0ed449c5649a55
Deleted: sha256:178dca62b393b539abc8b866c39be81e8ade01786880dc5d17ce3fe02426dbb
Deleted: sha256:400d99adaab999aea8c559754ba5978abb831ddab8043c02f1d16ae8b58b6c58
Deleted: sha256:25698fb762b3458e2c3d7ef87084ffd596d14a59499f4a7bd53d0bc744b75ba5
Deleted: sha256:0e457ae19e1984adb75f45e4882b75c7aae9765835b9205161cb07731b86775
Deleted: sha256:f5fbf4254ee5b4fc94b14f0fee5d172eed00c0a948ab0e69b6cee9d7188ee87c
Deleted: sha256:5d60eb4b3a7c231c402f3b796e40af0f5cc18ccff7776c2b423b2438b196cc63
Deleted: sha256:a6c0a8b4c36e38e108633a9dd299d0f36778148c83242f9706ec7e3118d4e19d
Deleted: sha256:2adecce90a82e54eb1236fd3cd6a2dc72002ed3211c65057dd1933f425ae13f6
Deleted: sha256:e8db4a9e45cf9ae5cc9214264472b2d707915c96f73f59d4faa85baad2f23a0f
Deleted: sha256:89fda09479fc0fe3bf2c411d92432001870e9dad42dd0c53715ab77ac4f2a97
Untagged: mark:latest
Deleted: sha256:e56db3e34776623903c7fc7cd226491a7d8dfb8d356e1cc6c6b77b73ddb6531
Deleted: sha256:08613414f562b7e2ca3db5c360149b40c3514dcbe7f9ffce25052c1d5cb74aa4
Deleted: sha256:7148aaecc052a85271a711d3598d5ecbf4889c4e3aa2fff83871b5f02cebl87
Deleted: sha256:ca0dd553952d9863fa3064db09493168236e9e9927d140a03c30503a7645108
Deleted: sha256:39e7aa6cbe14049bfed1fde845f6996aab9d65f3410a3c6edac2b8b942a0dfc4
Deleted: sha256:30f02212fdfa245eb5b552cbf3db81eb5753426f26467a536018a0dd7b9b031f
Deleted: sha256:755cc30961a42e603b6088298f8ca441ff7418bed241cea658c36921a6d8c727
Deleted: sha256:2081f37fd6f2b11d04dfc8a3dbd1eb56507e5ac9cee2d8193e90c7b7326f1001
Deleted: sha256:74f7cada5b5f729b0b68a9d8db2cfe34c19f12ab3700692701dfd5b3aeb43c59
Deleted: sha256:c03e108c7ae092c2c49b8cdd3facb6b2a9a49232f049ec8aae9959f781ba904
Deleted: sha256:41d80d999a1d692a21eeb4d8663039bb2e6b6b1072ce9fb9b13eb918100c81ec
Deleted: sha256:a6aa1afaa997d7c6a24ff10a37cf5e6809f92dd3ff20d0059f47f2d81d388cbd
Deleted: sha256:c133f348665elc6b927f267ce7819304bca69cle9d5210199eca82a8cd7adf21
Untagged: ubuntu:latest
Untagged: ubuntu@sha256:8ae9baafb64f63a50caab98fd3a5e37b3eb837a3e0700b78e5218e63193961f9
Deleted: sha256:2b4cba85892afc2ad8ce258a8e3d9dae4a1626ba300677cee93ef2338da442ab
Deleted: sha256:68a85fa9d77ecac87de23805c4be8766bda08a86787e324036cbcf84b62640fa
Untagged: python:3.9-buster
Untagged: python@sha256:0c3c9c7d40f8913226763130061ea8497d54e86eb807aa800019e6edc19e1538
Deleted: sha256:f3a16526089d30367c12b02bd14865fdb20f7d263aea634c1fc2347876207cd7
Deleted: sha256:a117f0eb7f6ac8b993fb55414e45fcahffa2fbf2402cef6e1c262e9868ba1a1
Deleted: sha256:e13342cc7647bc64136debc22d46d1e2e5600714c53a25ccdfdf3900a302be1
Deleted: sha256:b7138a88f68c354bdf673d02fc94e1da2fd8ef3e96ca94257d9035c592e309
Deleted: sha256:fb7930b09b0f7d24aff27046a35c2f1152d03bb7f54653481eed06e5692ed00c
Deleted: sha256:4099bc40676f17d0bf6ef5d3a9369a334cb3fbde56c80f7cdd0f18397ba6774
Deleted: sha256:2033b7e91b50c3002c7614607dcb8b25838d1ce8cd7a330b9f6821b433c53a0d
Deleted: sha256:83ca909ebec23738c9169dc5ccb4abb7dffca486ed09fd86d6581428140a76cae6
Deleted: sha256:cb00fc4642580cd25bf35b2249edd66f4a3e550f1739bb93c867d67a65cdf68d
Deleted: sha256:b14cb48b3aebbc58396d0b3c2d0880fd9c002c56bb7453af3ddfe6e119c06df2

Code
$

```

How To Delete Containers In Docker

Before deleting the containers we need to stop the container first for that we use the command.

```
docker stop <containerId/Name>
```

The Difference Between Docker Stop & Docker Kill

Docker stop will first send a SIGTERM signal before killing the process with a SIGKILL signal and a grace period. When Docker kill sends SIGKILL, it immediately terminates the process.

- **Stop all running Containers:** In order to stop the containers which have not exited. This might happen when the command used in the Docker image is left running. The command should be exited and this will in turn stop the container. To stop the container when you have not exited the container by stopping the command, you need to run the following command.

```
docker stop $(docker ps -aq)
```

- **Delete The Container:** If the container is stopped then we can use the following command to delete the container.

```
docker rm <containerId/Name>
```

- **Force Delete The Container:** We can force remove the containers while they are running without stopping them by using the below command.

```
docker rm -f <containerId/Name>
```

```
Code
$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
a7e9a7286e9c       mark               "python -m http.se..." 6 seconds ago       Up 3 seconds       8080/tcp           adoring_mclean
48da167d4f1a       mark               "python -m http.se..." 37 seconds ago      Up 34 seconds      8080/tcp           thirsty_borg

Code
$ docker stop $(docker ps -aq)
da8b53b4ac1e
4e539ea5173b
5915afe67dde
edbf652cde59
4fd778a1a874
80ae41fca13e
1b79596b6385
5e2a26ff997a
551fb380d685
8c566b91aa62
f31da84b5f62
8f07fef7229
aecea9ab8132
94a96bb2e9db
af73d337859e
436ae328a5f4

Code
$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES

Code
$
```

Remove all Containers

To remove all containers from the docker machine, we need to get the ids of all the containers. We can simply get the ids of the containers with the command **docker ps -aq**, then by using the docker rm command, we can remove all the containers in the docker-machine.

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

```
Code
$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES

Code
$ docker ps -aq
1b79596b6385
5e2a26ff997a
551fb300d605
8c566b01aa62
f31da84b5f62
8f707fef7229
aecea9ab8132
94a90bb2e9db
af73d337859e
436ae328a5f4

Code
$ docker rm $(docker ps -aq)
1b79596b6385
5e2a26ff997a
551fb300d605
8c566b01aa62
f31da84b5f62
8f707fef7229
aecea9ab8132
94a90bb2e9db
af73d337859e
436ae328a5f4

Code
$
```

Remove all Stopped Containers

To remove all containers which are stopped/exited, we can use filters in the ps command argument. We can't directly remove a container if it is not stopped. We can stop containers that are not exited or are running by using the -f argument to the ps command in docker, the -f or --filter option takes in a filter like status=exited or status=running or name and so on. We can filter out to stop the specific containers according to the requirement.

```
docker rm $(docker ps -aq --filter status="exited")
```

After filtering out the container which is running, we can use the stop command to stop those containers with the -q to silence the numeric ids associated with those containers.

```
docker stop $(docker ps --filter status=running -q)
```

This will stop all the containers and thus we can now remove the containers from the docker-machine. We can even filter the containers which are stopped here to remove only those whose status is exited.

```
docker rm $(docker ps --filter status=exited -q)
```

```

Code
$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
bf954c3cb678       mark               "python -m http.se..." 5 minutes ago       Up 5 minutes       8000/tcp            epic_leakey
e4933f8993ab       mark               "python -m http.se..." 5 minutes ago       Up 5 minutes       8000/tcp            crazy_tesla

Code
$ docker ps --filter status=exited
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
d97761f57203       python             "bash"              About a minute ago   Exited (0) About a minute ago
edd684fb4627       python             "bash"              About a minute ago   Exited (0) About a minute ago
9f7277396448       python             "python3"           About a minute ago   Exited (0) About a minute ago

Code
$ docker ps --filter status=running
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
bf954c3cb678       mark               "python -m http.se..." 6 minutes ago       Up 5 minutes       8000/tcp            epic_leakey
e4933f8993ab       mark               "python -m http.se..." 6 minutes ago       Up 6 minutes       8000/tcp            crazy_tesla

Code
$ docker stop $(docker ps --filter status=running -q)
bf954c3cb678
e4933f8993ab

Code
$ docker rm $(docker ps -f status=exited -q)
d97761f57203
edd684fb4627
9f7277396448
bf954c3cb678
e4933f8993ab

Code
$

```

- The below command removed all the containers which are in the existing state. That means the containers stopped.
- docker container prune

```

root@ip-172-31-35-129:~# docker container prune
WARNING! This will remove all stopped containers.
Are you sure you want to continue? [y/N] y
Deleted Containers:
ab01dc107246bf81300f4d647e4274efa0dfcfa502c7100a84d4b7333fe7f50c
ec3f5e056c48c739edae0f06c49e06f8d6dd49078af70faf28beff57e7836244
358f0ed8fe005f66d3355e345a6713ccc1edde270918e4ad079700444c9a63fe

Total reclaimed space: 5.083MB
root@ip-172-31-35-129:~# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
5335cd1c7de1       medilab-website-img "/docker-entrypoint..." 16 minutes ago     Up 5 minutes       0.0.0.0:5001->80/tcp, :::5001->80/tcp      medilab-website
root@ip-172-31-35-129:~#

```