Docker Cleanup Process

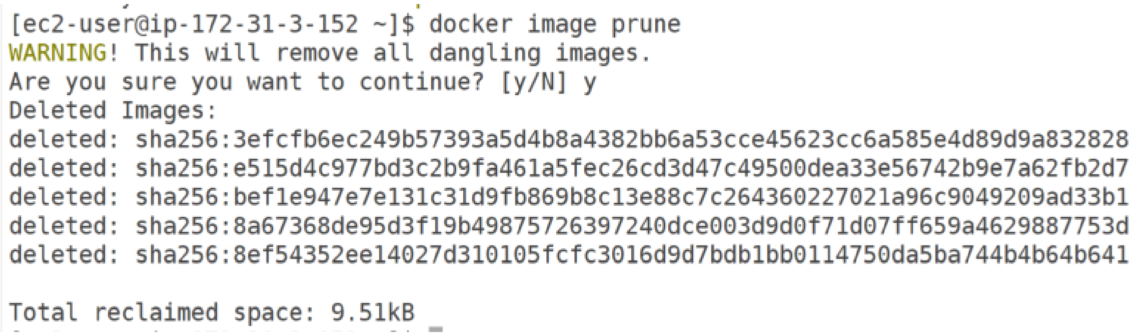
Prune unused Docker objects

Docker takes a conservative approach to cleaning up unused objects (often referred to as "garbage collection"), such as images, containers, volumes, and networks: these objects are generally not removed unless you explicitly ask Docker to do so. This can cause Docker to use extra disk space. For each type of object, Docker provides a **prune** command. In addition, you can use **docker system prune** to clean up multiple types of objects at once. This topic shows how to use these **prune** commands.

Prune images

The **docker image prune** command allows you to clean up unused images. By default, **docker image prune** only cleans up **dangling images**. A **dangling image** is one that is not tagged and is not referenced by any container. To remove dangling images:

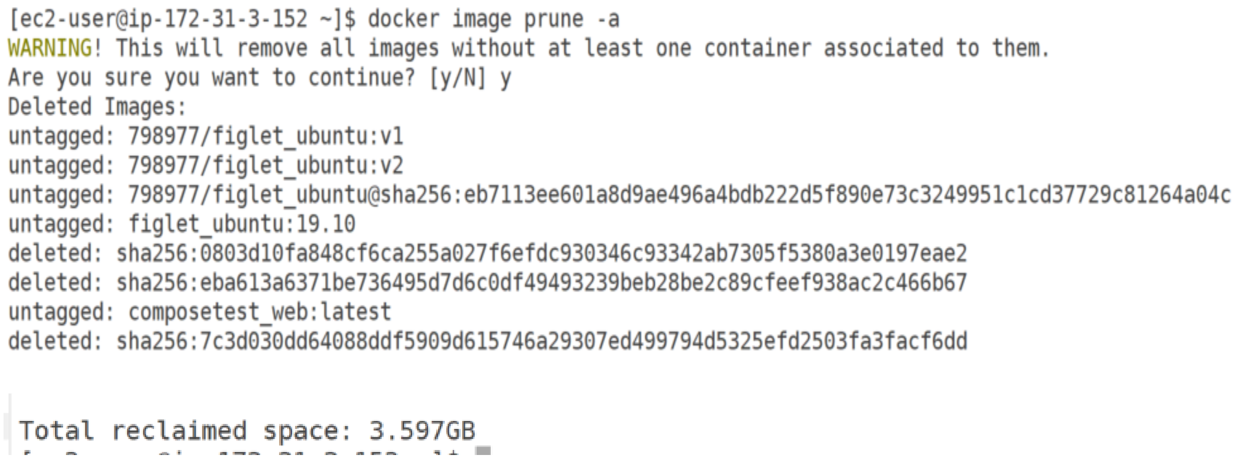
```
$ docker image prune
```



```
[ec2-user@ip-172-31-3-152 ~]$ docker image prune
WARNING! This will remove all dangling images.
Are you sure you want to continue? [y/N] y
Deleted Images:
deleted: sha256:3efcfb6ec249b57393a5d4b8a4382bb6a53cce45623cc6a585e4d89d9a832828
deleted: sha256:e515d4c977bd3c2b9fa461a5fec26cd3d47c49500dea33e56742b9e7a62fb2d7
deleted: sha256:bef1e947e7e131c31d9fb869b8c13e88c7c264360227021a96c9049209ad33b1
deleted: sha256:8a67368de95d3f19b49875726397240dce003d9d0f71d07ff659a4629887753d
deleted: sha256:8ef54352ee14027d310105fcfc3016d9d7bdb1bb0114750da5ba744b4b64b641

Total reclaimed space: 9.51kB
```

To remove all images which are not used by existing containers, use the **-a** flag:

```
$ docker image prune -a
```



```
[ec2-user@ip-172-31-3-152 ~]$ 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: 798977/figlet_ubuntu:v1
untagged: 798977/figlet_ubuntu:v2
untagged: 798977/figlet_ubuntu@sha256:eb7113ee601a8d9ae496a4bdb222d5f890e73c3249951c1cd37729c81264a04c
untagged: figlet_ubuntu:19.10
deleted: sha256:0803d10fa848cf6ca255a027f6efdc930346c93342ab7305f5380a3e0197eae2
deleted: sha256:eba613a6371be736495d7d6c0df49493239beb28be2c89cfeef938ac2c466b67
untagged: composetest_web:latest
deleted: sha256:7c3d030dd64088ddf5909d615746a29307ed499794d5325efd2503fa3facf6dd

Total reclaimed space: 3.597GB
```

By default, you are prompted to continue. To bypass the prompt, use the **-f or --force** flag.

You can limit which images are pruned using filtering expressions with the **--filter** flag. For example, to only consider images created more than 24 hours ago:

```
$ docker image prune -a --filter "until=24h"
```

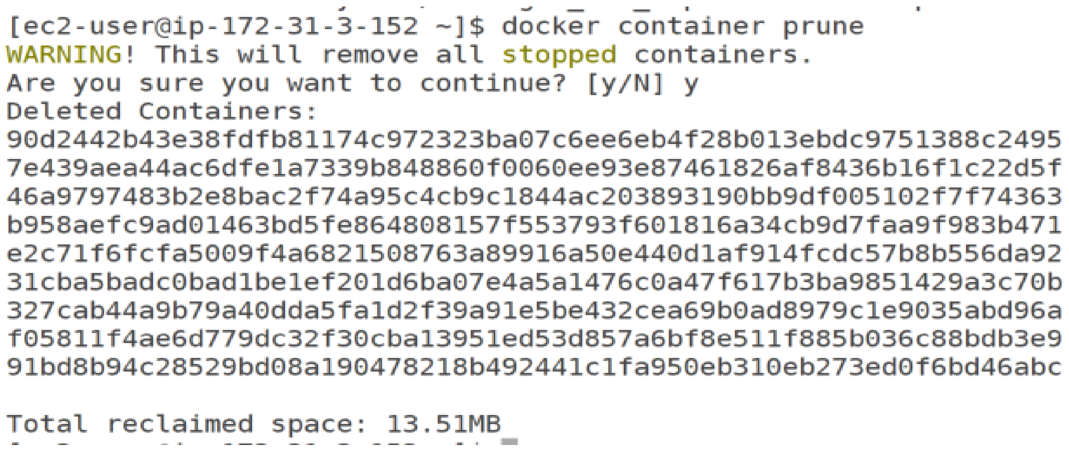
for more options:

https://docs.docker.com/engine/reference/commandline/image_prune/

Prune containers

When you stop a container, it is not automatically removed unless you started it with the **--rm** flag. To see all containers on the Docker host, including stopped containers, use **docker ps -a**. You may be surprised how many containers exist, especially on a development system! A stopped container's writable layers still take up disk space. To clean this up, you can use the **docker container prune** command.

```
$ docker container prune
```



```
[ec2-user@ip-172-31-3-152 ~]$ docker container prune
WARNING! This will remove all stopped containers.
Are you sure you want to continue? [y/N] y
Deleted Containers:
90d2442b43e38fdfb81174c972323ba07c6ee6eb4f28b013ebdc9751388c2495
7e439aea44ac6dfe1a7339b848860f0060ee93e87461826af8436b16f1c22d5f
46a9797483b2e8bac2f74a95c4cb9c1844ac203893190bb9df005102f7f74363
b958aefc9ad01463bd5fe864808157f553793f601816a34cb9d7faa9f983b471
e2c71f6fcfa5009f4a6821508763a89916a50e440d1af914fcdc57b8b556da92
31cba5badc0bad1be1ef201d6ba07e4a5a1476c0a47f617b3ba9851429a3c70b
327cab44a9b79a40dda5fa1d2f39a91e5be432cea69b0ad8979c1e9035abd96a
f05811f4ae6d779dc32f30cba13951ed53d857a6bf8e511f885b036c88bdb3e9
91bd8b94c28529bd08a190478218b492441c1fa950eb310eb273ed0f6bd46abc

Total reclaimed space: 13.51MB
```

By default, you are prompted to continue. To bypass the prompt, use the **-f** or **--force** flag.

By default, all stopped containers are removed. You can limit the scope using the **--filter** flag. For instance, the following command only removes stopped containers older than 24 hours:

```
$ docker container prune --filter "until=24h"
```

for more options https://docs.docker.com/engine/reference/commandline/container_prune/

Prune volumes

Volumes can be used by one or more containers, and take up space on the Docker host. Volumes are never removed automatically, because to do so could destroy data.

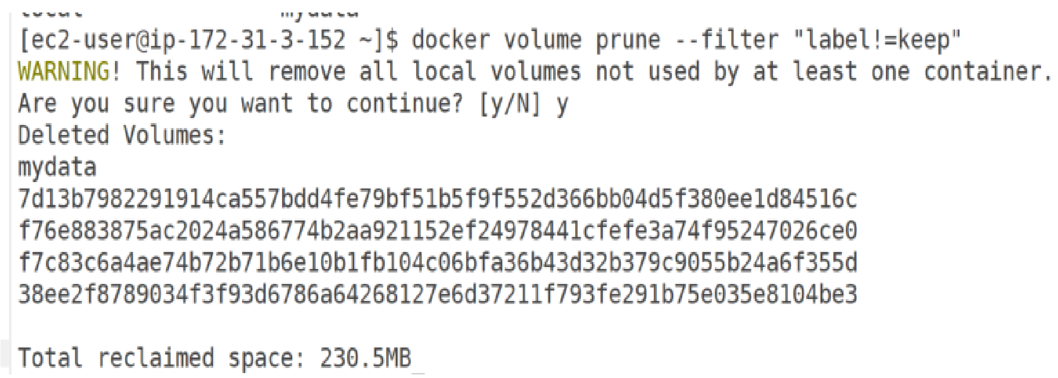
```
$ docker volume prune
```

By default, you are prompted to continue. To bypass the prompt, use the **-f or --force** flag.

By default, all unused volumes are removed. You can limit the scope using the **--filter** flag. For instance, the following command only removes volumes which are not labelled with the **keep** label:

ref: https://docs.docker.com/engine/reference/commandline/volume_ls/

```
$ docker volume prune --filter "label!=keep"
```



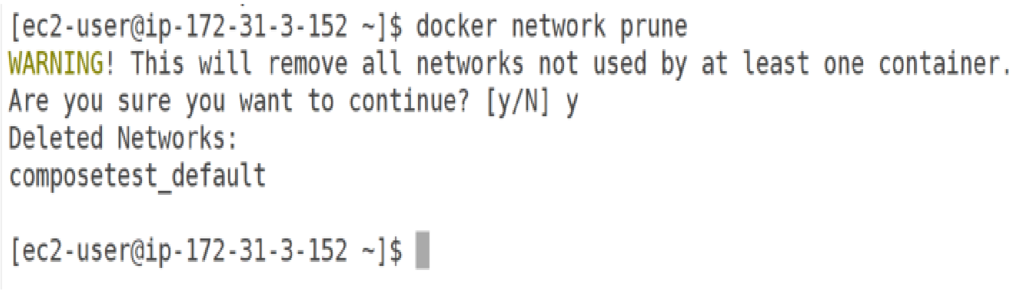
```
[ec2-user@ip-172-31-3-152 ~]$ docker volume prune --filter "label!=keep"
WARNING! This will remove all local volumes not used by at least one container.
Are you sure you want to continue? [y/N] y
Deleted Volumes:
mydata
7d13b7982291914ca557bdd4fe79bf51b5f9f552d366bb04d5f380ee1d84516c
f76e883875ac2024a586774b2aa921152ef24978441cfefe3a74f95247026ce0
f7c83c6a4ae74b72b71b6e10b1fb104c06bfa36b43d32b379c9055b24a6f355d
38ee2f8789034f3f93d6786a64268127e6d37211f793fe291b75e035e8104be3

Total reclaimed space: 230.5MB
```

for more options https://docs.docker.com/engine/reference/commandline/volume_prune/

Prune networks

Docker networks don't take up much disk space, but they do create **iptables** rules, bridge network devices, and routing table entries. To clean these things up, you can use **docker network prune** to clean up networks which aren't used by any containers.

```
$ docker network prune
```



```
[ec2-user@ip-172-31-3-152 ~]$ docker network prune
WARNING! This will remove all networks not used by at least one container.
Are you sure you want to continue? [y/N] y
Deleted Networks:
composetest_default

[ec2-user@ip-172-31-3-152 ~]$
```

By default, you are prompted to continue. To bypass the prompt, use the **-f or --force** flag.

By default, all unused networks are removed. You can limit the scope using the **--filter** flag. For instance, the following command only removes networks older than 24 hours:

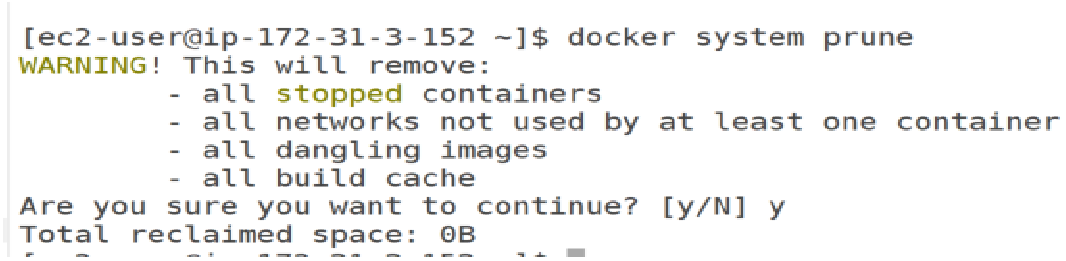
```
$ docker network prune --filter "until=24h"
```

for more options https://docs.docker.com/engine/reference/commandline/network_prune/

Prune everything

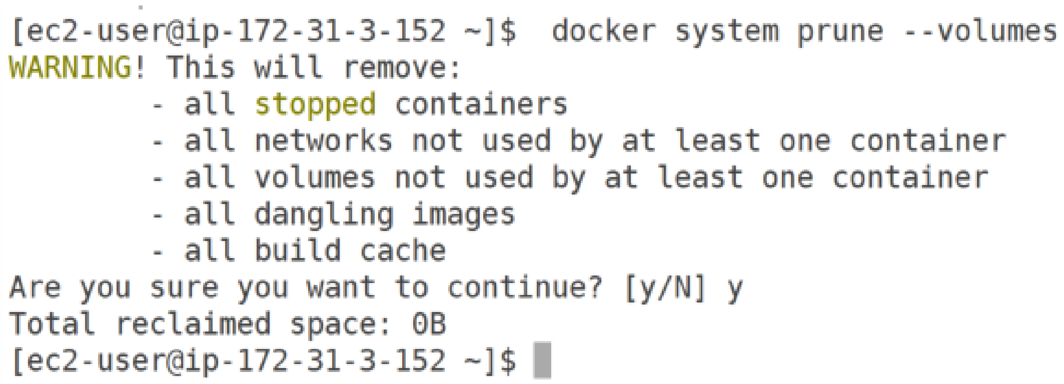
The **docker system prune** command is a shortcut that prunes images, containers, and networks. In Docker 17.06.0 and earlier, volumes are also pruned. In Docker 17.06.1 and higher, you must specify the **--volumes** flag for **docker system prune** to prune **volumes**.

```
$ docker system prune
```



```
[ec2-user@ip-172-31-3-152 ~]$ docker system prune
WARNING! This will remove:
        - all stopped containers
        - all networks not used by at least one container
        - all dangling images
        - all build cache
Are you sure you want to continue? [y/N] y
Total reclaimed space: 0B
```

```
$ docker system prune --volumes
```



```
[ec2-user@ip-172-31-3-152 ~]$  docker system prune --volumes
WARNING! This will remove:
        - all stopped containers
        - all networks not used by at least one container
        - all volumes not used by at least one container
        - all dangling images
        - all build cache
Are you sure you want to continue? [y/N] y
Total reclaimed space: 0B
[ec2-user@ip-172-31-3-152 ~]$
```