Docker Task

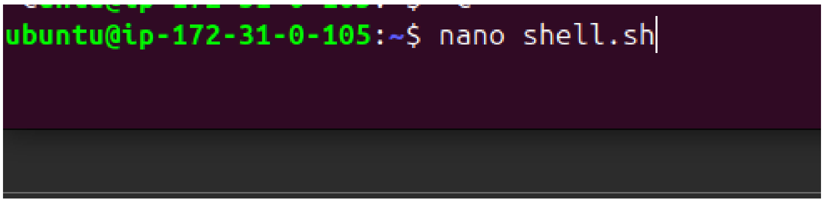
Task Description:

Install docker on EC2 and explore the docker commands (docker images, containers, volumes, network)

Solution:

Installing Docker on EC2

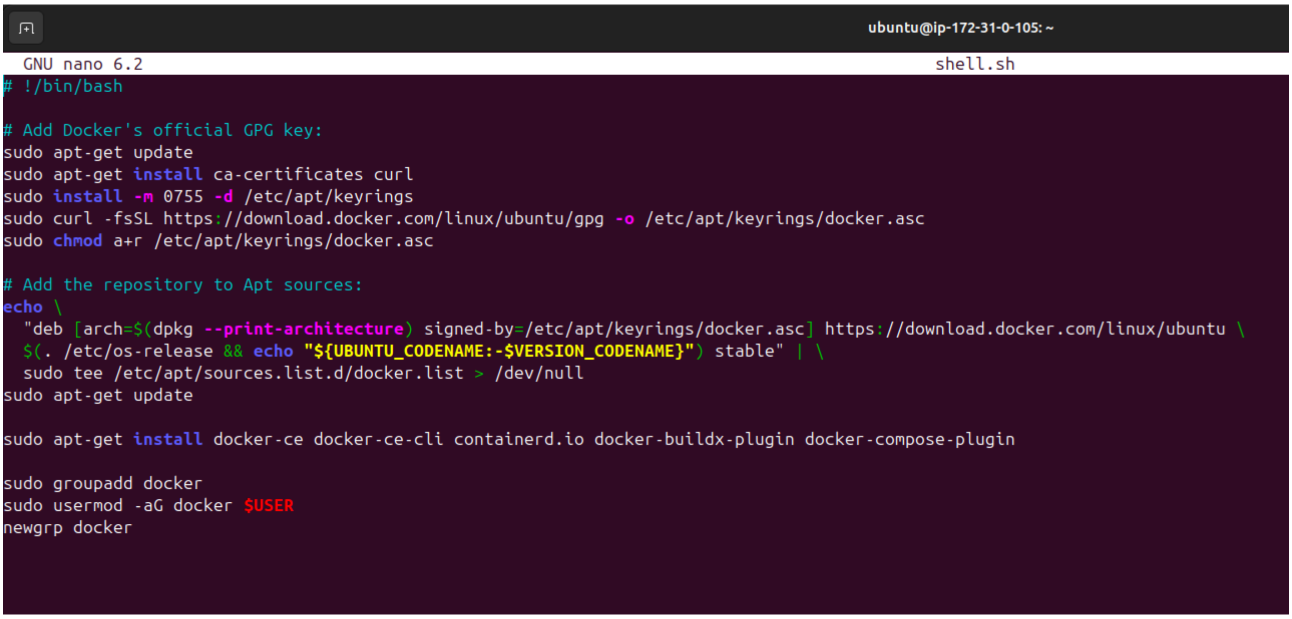
-commands create a file (Shell.sh)and enter the below Bash Commands:



```
ubuntu@ip-172-31-0-105:~$ nano shell.sh
```

```
# !/bin/bash
# Add Docker's official GPG key:
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

# Add the repository to Apt sources:
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/ubuntu \
  $(. /etc/os-release && echo "${UBUNTU_CODENAME:-$VERSION_CODENAME}") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
#Installing Docker
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
# Adding group to the server
sudo groupadd docker
sudo usermod -aG docker $USER
newgrp docker
```



```
ubuntu@ip-172-31-0-105: ~
 GNU nano 6.2                                                                                   shell.sh
# !/bin/bash

# Add Docker's official GPG key:
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

# Add the repository to Apt sources:
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu \
  $(. /etc/os-release && echo "${UBUNTU_CODENAME:-$VERSION_CODENAME}") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update

sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin

sudo groupadd docker
sudo usermod -aG docker $USER
newgrp docker
```

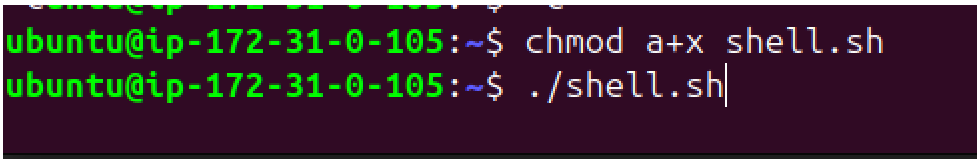
- Save and exit the File.

- Make the file to be executable.

~chmod a+x shell.sh

- Execute the file

~./shell.sh



```
ubuntu@ip-172-31-0-105:~$ chmod a+x shell.sh
ubuntu@ip-172-31-0-105:~$ ./shell.sh
```

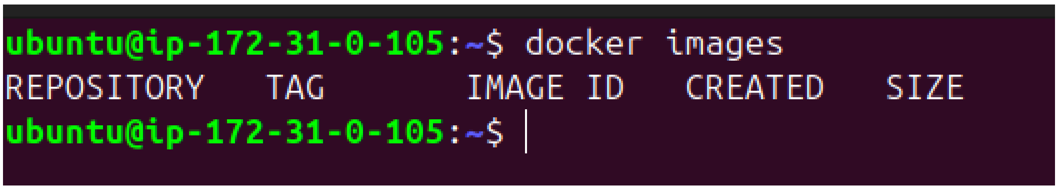
Exploring Docker Commands

Now let's explore the fundamental Docker commands:

Docker Images

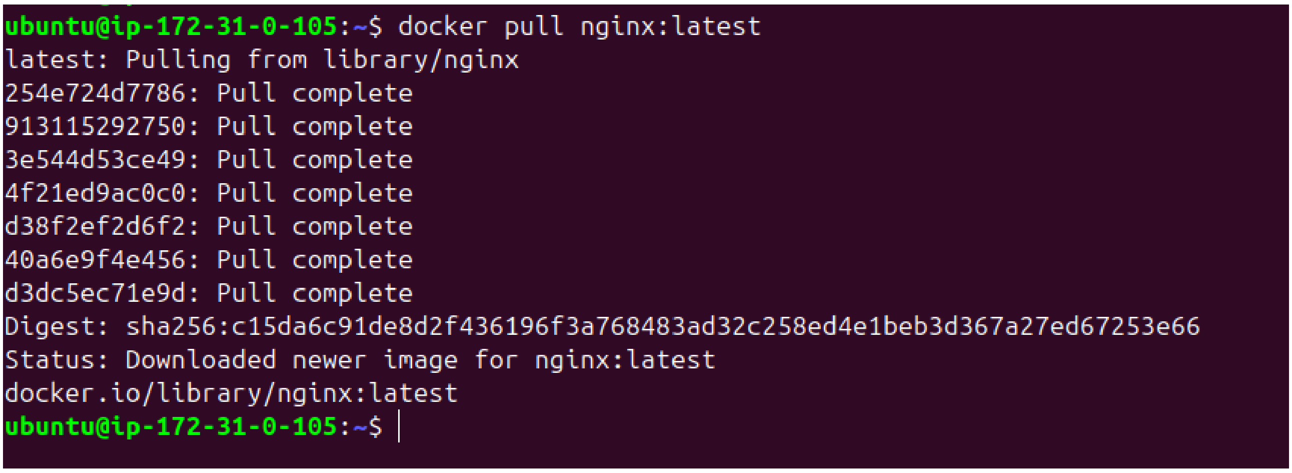
bash

```
# List all images
docker images
```



```
ubuntu@ip-172-31-0-105:~$ docker images
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE
ubuntu@ip-172-31-0-105:~$
```

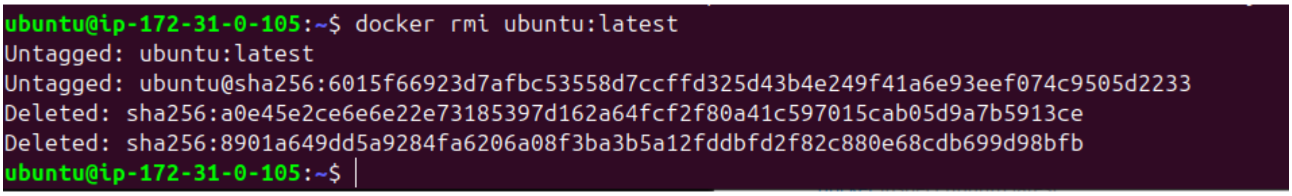
```
# Pull an image from Docker Hub
docker pull nginx:latest
docker pull ubuntu:latest
```



```
ubuntu@ip-172-31-0-105:~$ docker pull nginx:latest
latest: Pulling from library/nginx
254e724d7786: Pull complete
913115292750: Pull complete
3e544d53ce49: Pull complete
4f21ed9ac0c0: Pull complete
d38f2ef2d6f2: Pull complete
40a6e9f4e456: Pull complete
d3dc5ec71e9d: Pull complete
Digest: sha256:c15da6c91de8d2f436196f3a768483ad32c258ed4e1beb3d367a27ed67253e66
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
ubuntu@ip-172-31-0-105:~$
```

# Remove an image

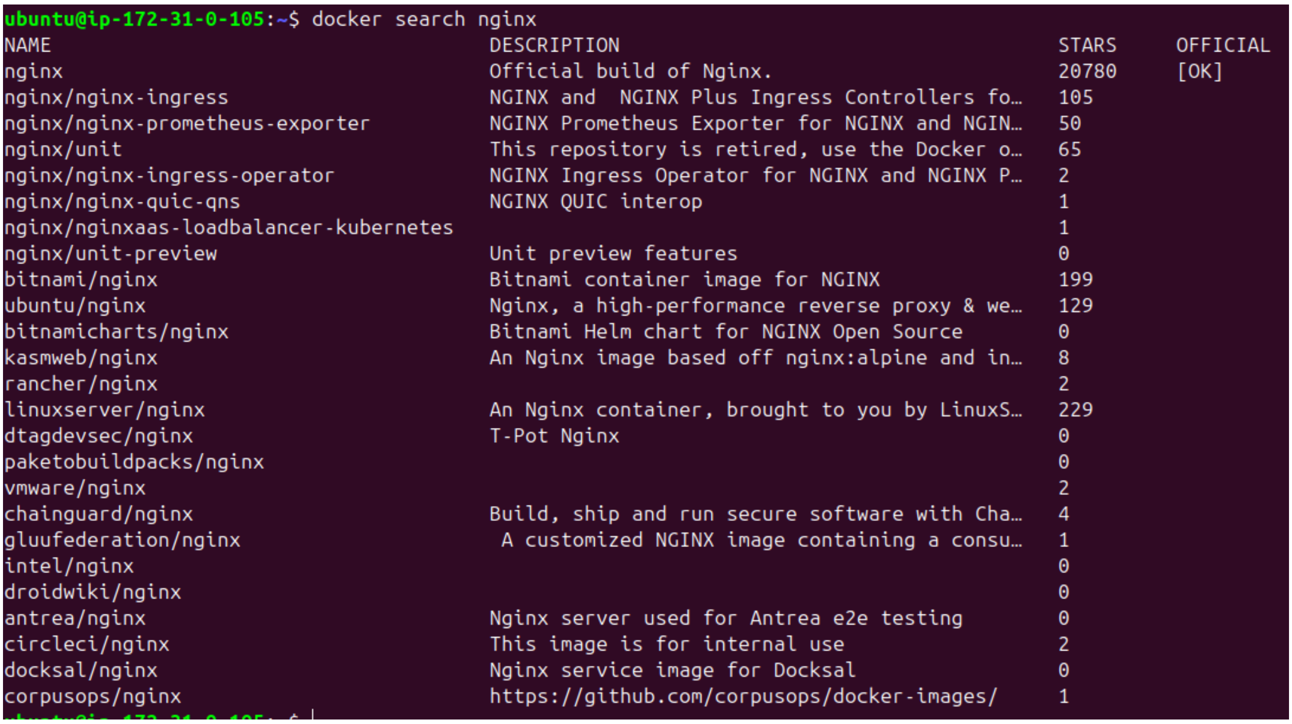
docker rmi ubuntu:latest



```
ubuntu@ip-172-31-0-105:~$ docker rmi ubuntu:latest
Untagged: ubuntu:latest
Untagged: ubuntu@sha256:6015f66923d7afbc53558d7ccffd325d43b4e249f41a6e93eef074c9505d2233
Deleted: sha256:a0e45e2ce6e6e22e73185397d162a64fcf2f80a41c597015cab05d9a7b5913ce
Deleted: sha256:8901a649dd5a9284fa6206a08f3ba3b5a12fddbfd2f82c880e68cdb699d98bfb
ubuntu@ip-172-31-0-105:~$
```

# Search for images on Docker Hub

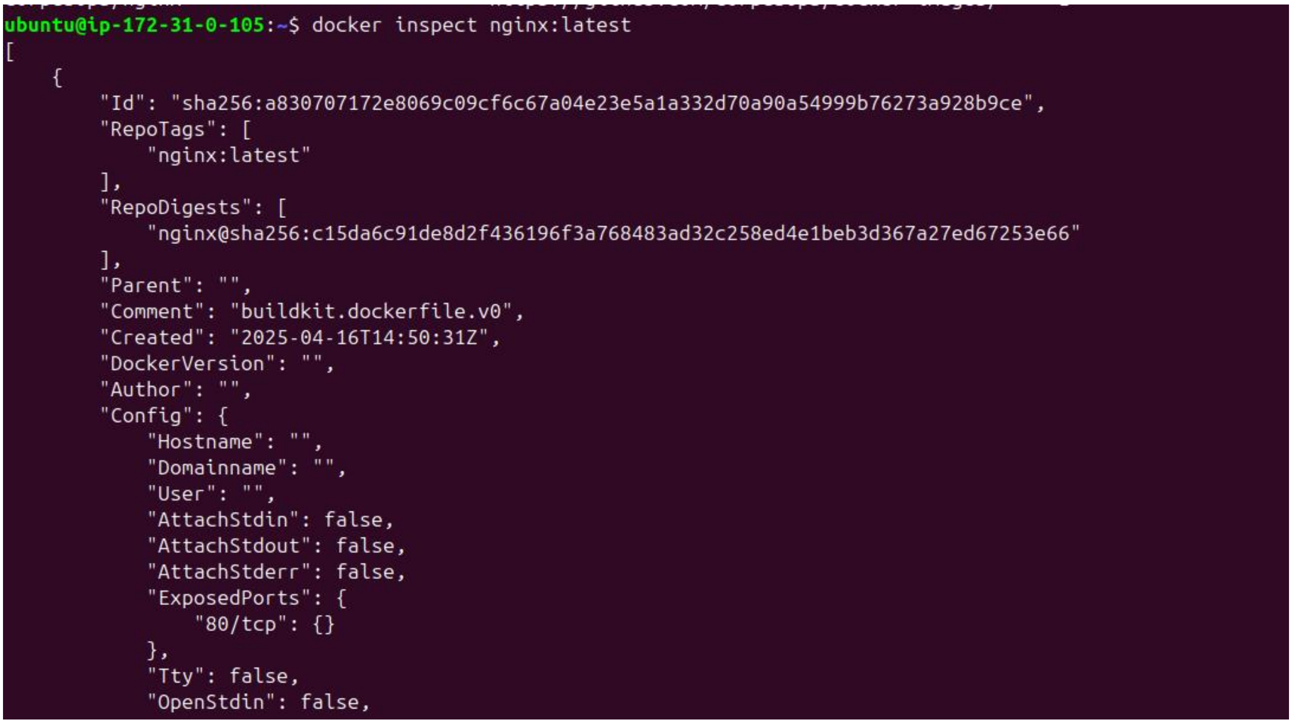
docker search nginx



```
ubuntu@ip-172-31-0-105:~$ docker search nginx
NAME                                     DESCRIPTION                                     STARS     OFFICIAL
nginx                                    Official build of Nginx.                        20780     [OK]
nginx/nginx-ingress                      NGINX and  NGINX Plus Ingress Controllers fo…   105
nginx/nginx-prometheus-exporter          NGINX Prometheus Exporter for NGINX and NGIN…   50
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-quic-qns                     NGINX QUIC interop                              1
nginx/nginxaas-loadbalancer-kubernetes                                                   1
nginx/unit-preview                       Unit preview features                           0
bitnami/nginx                            Bitnami container image for NGINX               199
ubuntu/nginx                             Nginx, a high-performance reverse proxy & we…   129
bitnamicharts/nginx                      Bitnami Helm chart for NGINX Open Source        0
kasmweb/nginx                            An Nginx image based off nginx:alpine and in…   8
rancher/nginx                                                                            2
linuxserver/nginx                        An Nginx container, brought to you by LinuxS…   229
dtagdevsec/nginx                         T-Pot Nginx                                     0
paketobuildpacks/nginx                                                                   0
vmware/nginx                                                                             2
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
antrea/nginx                             Nginx server used for Antrea e2e testing        0
circleci/nginx                           This image is for internal use                  2
docksal/nginx                            Nginx service image for Docksal                 0
corpusops/nginx                          https://github.com/corpusops/docker-images/     1
```

# View image details

docker inspect ubuntu:latest



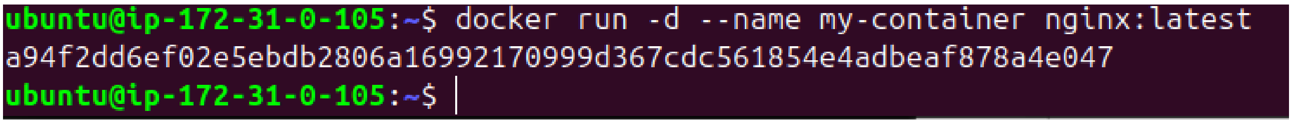
```
ubuntu@ip-172-31-0-105:~$ docker inspect nginx:latest
[
    {
        "Id": "sha256:a830707172e8069c09cf6c67a04e23e5a1a332d70a90a54999b76273a928b9ce",
        "RepoTags": [
            "nginx:latest"
        ],
        "RepoDigests": [
            "nginx@sha256:c15da6c91de8d2f436196f3a768483ad32c258ed4e1beb3d367a27ed67253e66"
        ],
        "Parent": "",
        "Comment": "buildkit.dockerfile.v0",
        "Created": "2025-04-16T14:50:31Z",
        "DockerVersion": "",
        "Author": "",
        "Config": {
            "Hostname": "",
            "Domainname": "",
            "User": "",
            "AttachStdin": false,
            "AttachStdout": false,
            "AttachStderr": false,
            "ExposedPorts": {
                "80/tcp": {}
            },
            "Tty": false,
            "OpenStdin": false,
```

Docker Containers

bash

# Run a container

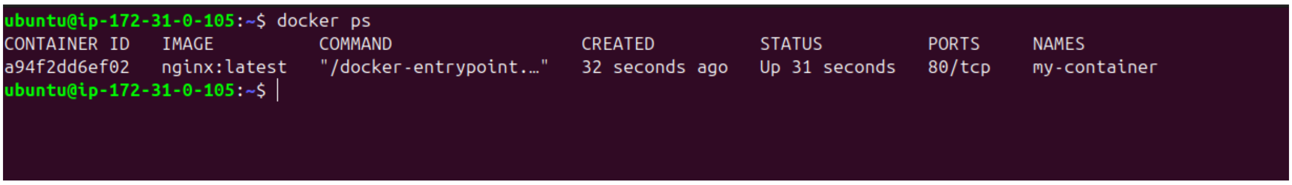
docker run -d --name my-container nginx:latest



```
ubuntu@ip-172-31-0-105:~$ docker run -d --name my-container nginx:latest
a94f2dd6ef02e5ebdb2806a16992170999d367cdc561854e4adbeaf878a4e047
ubuntu@ip-172-31-0-105:~$
```

# List running containers

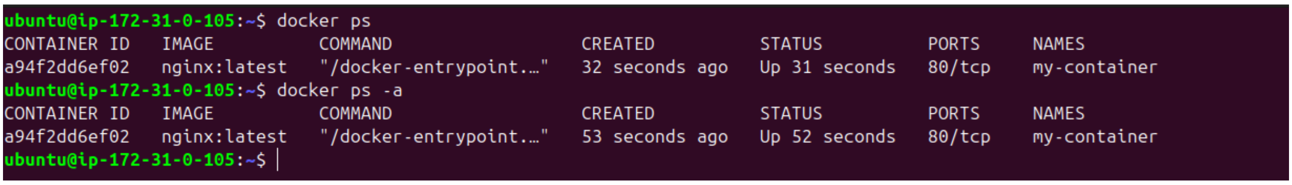
docker ps



```
ubuntu@ip-172-31-0-105:~$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS          PORTS     NAMES
a94f2dd6ef02   nginx:latest   "/docker-entrypoint.…"   32 seconds ago   Up 31 seconds   80/tcp    my-container
ubuntu@ip-172-31-0-105:~$
```

# List all containers (including stopped)

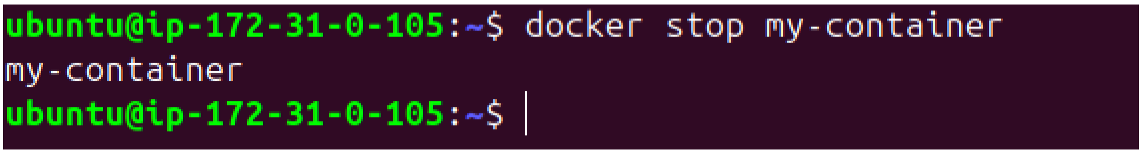
docker ps -a



```
ubuntu@ip-172-31-0-105:~$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS          PORTS     NAMES
a94f2dd6ef02   nginx:latest   "/docker-entrypoint.…"   32 seconds ago   Up 31 seconds   80/tcp    my-container
ubuntu@ip-172-31-0-105:~$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS          PORTS     NAMES
a94f2dd6ef02   nginx:latest   "/docker-entrypoint.…"   53 seconds ago   Up 52 seconds   80/tcp    my-container
ubuntu@ip-172-31-0-105:~$
```

# Stop a container

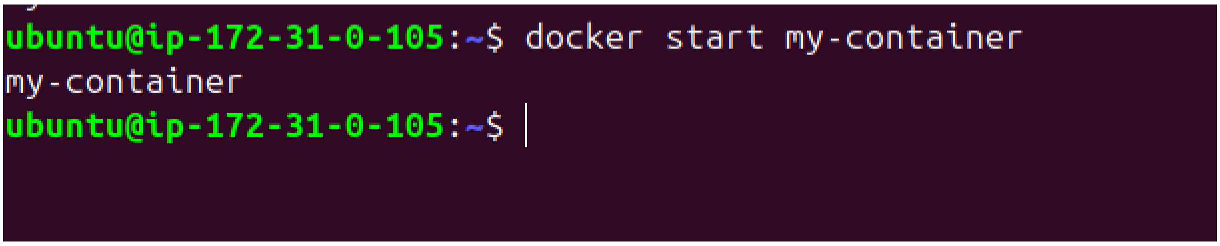
docker stop my-container



```
ubuntu@ip-172-31-0-105:~$ docker stop my-container
my-container
ubuntu@ip-172-31-0-105:~$ 
```

# Start a container

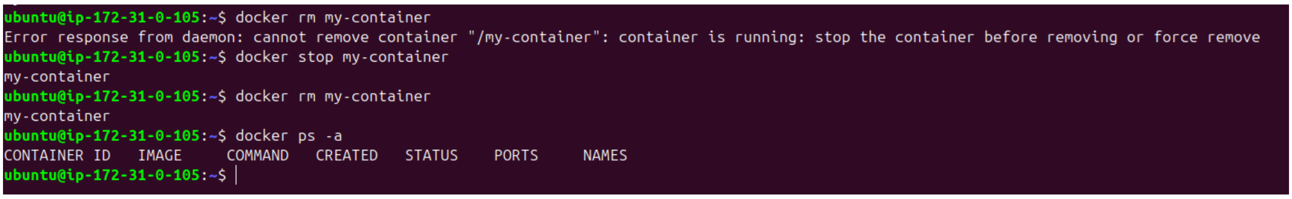
docker start my-container



```
ubuntu@ip-172-31-0-105:~$ docker start my-container
my-container
ubuntu@ip-172-31-0-105:~$ 
```

# Remove a container

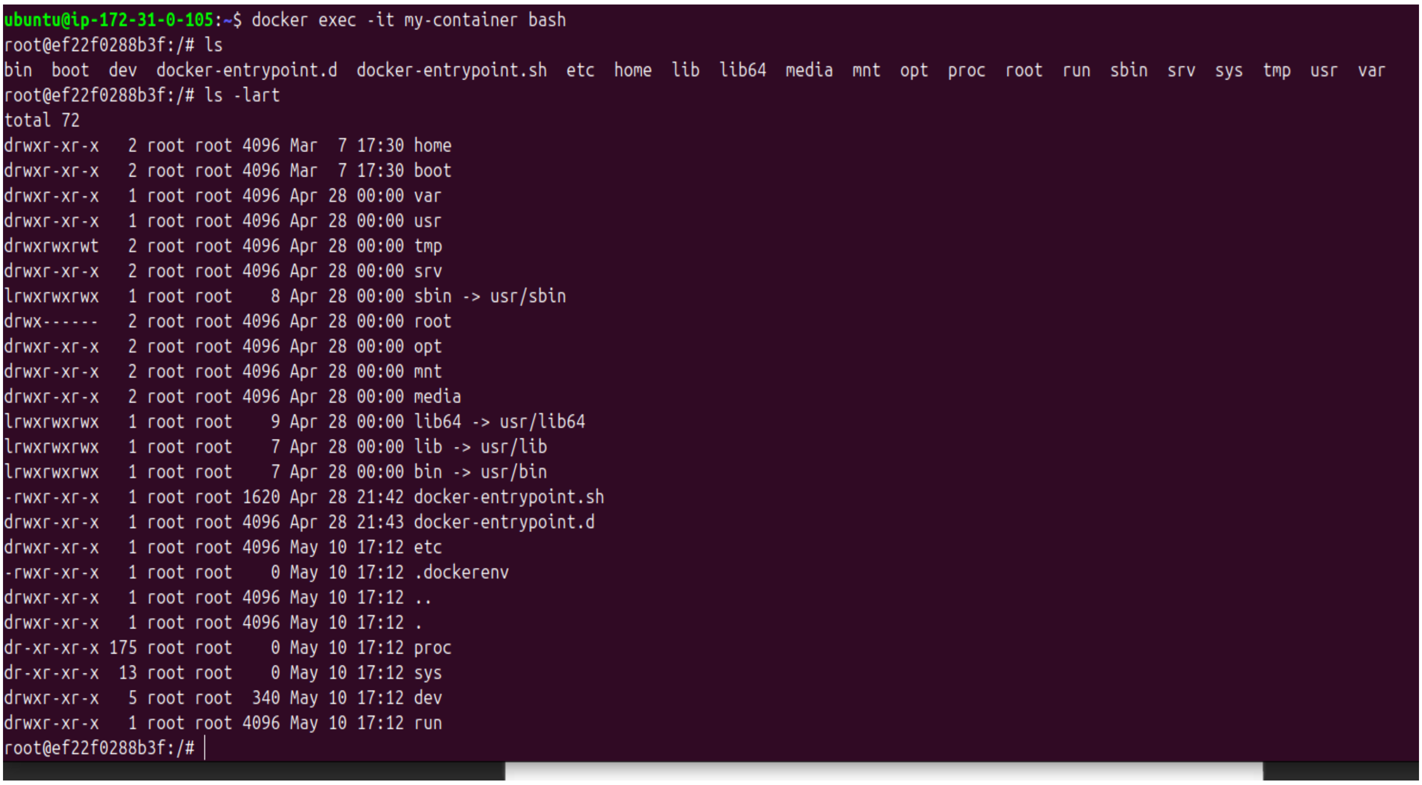
docker rm my-container



```
ubuntu@ip-172-31-0-105:~$ docker rm my-container
Error response from daemon: cannot remove container "/my-container": container is running: stop the container before removing or force remove
ubuntu@ip-172-31-0-105:~$ docker stop my-container
my-container
ubuntu@ip-172-31-0-105:~$ docker rm my-container
my-container
ubuntu@ip-172-31-0-105:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
ubuntu@ip-172-31-0-105:~$ 
```

# Execute a command in a running container

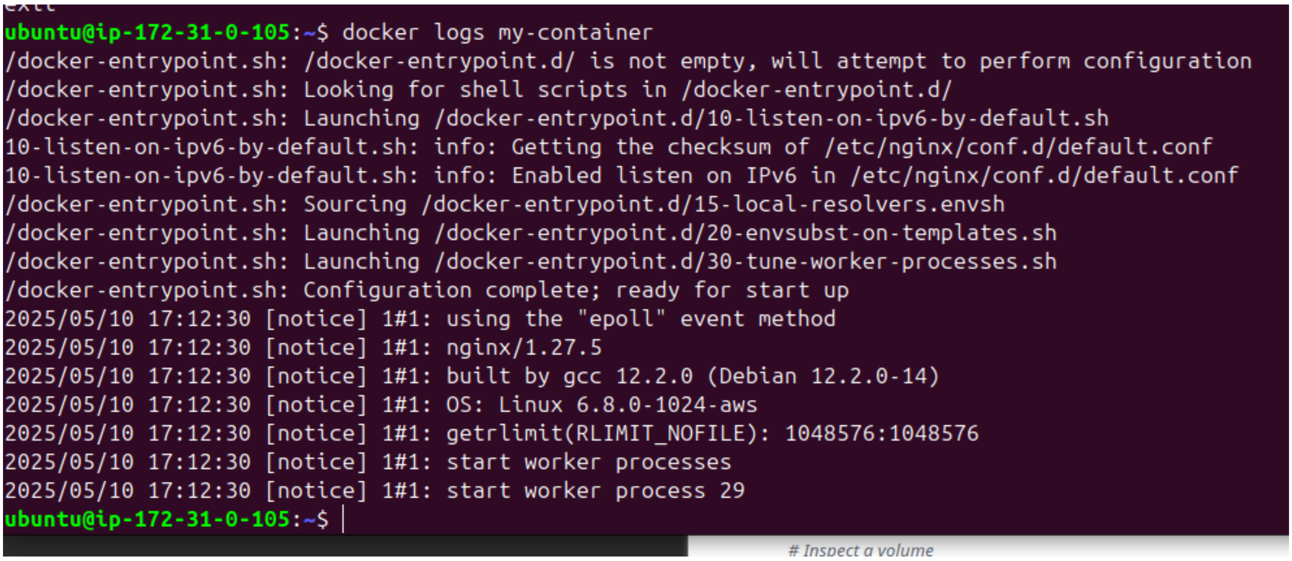
docker exec -it my-container bash



```
ubuntu@ip-172-31-0-105:~$ docker exec -it my-container bash
root@ef22f0288b3f:/# ls
bin  boot  dev  docker-entrypoint.d  docker-entrypoint.sh  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@ef22f0288b3f:/# ls -lart
total 72
drwxr-xr-x   2 root root 4096 Mar  7 17:30 home
drwxr-xr-x   2 root root 4096 Mar  7 17:30 boot
drwxr-xr-x   1 root root 4096 Apr 28 00:00 var
drwxr-xr-x   1 root root 4096 Apr 28 00:00 usr
drwxrwxrwt   2 root root 4096 Apr 28 00:00 tmp
drwxr-xr-x   2 root root 4096 Apr 28 00:00 srv
lrwxrwxrwx   1 root root    8 Apr 28 00:00 sbin -> usr/sbin
drwx------   2 root root 4096 Apr 28 00:00 root
drwxr-xr-x   2 root root 4096 Apr 28 00:00 opt
drwxr-xr-x   2 root root 4096 Apr 28 00:00 mnt
drwxr-xr-x   2 root root 4096 Apr 28 00:00 media
lrwxrwxrwx   1 root root    9 Apr 28 00:00 lib64 -> usr/lib64
lrwxrwxrwx   1 root root    7 Apr 28 00:00 lib -> usr/lib
lrwxrwxrwx   1 root root    7 Apr 28 00:00 bin -> usr/bin
-rwxr-xr-x   1 root root 1620 Apr 28 21:42 docker-entrypoint.sh
drwxr-xr-x   1 root root 4096 Apr 28 21:43 docker-entrypoint.d
drwxr-xr-x   1 root root 4096 May 10 17:12 etc
-rwxr-xr-x   1 root root    0 May 10 17:12 .dockerenv
drwxr-xr-x   1 root root 4096 May 10 17:12 ..
drwxr-xr-x   1 root root 4096 May 10 17:12 .
dr-xr-xr-x 175 root root    0 May 10 17:12 proc
dr-xr-xr-x  13 root root    0 May 10 17:12 sys
drwxr-xr-x   5 root root  340 May 10 17:12 dev
drwxr-xr-x   1 root root 4096 May 10 17:12 run
root@ef22f0288b3f:/# 
```

# View container logs

docker logs my-container

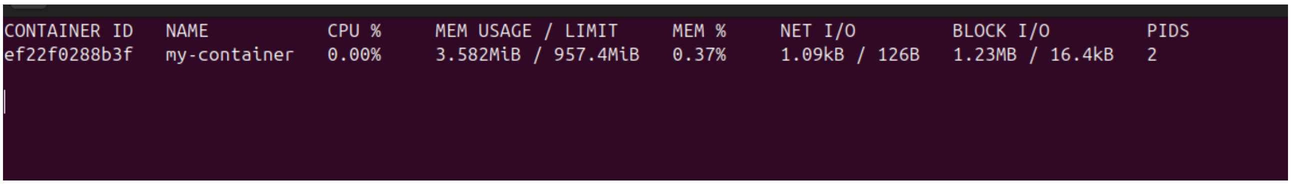


```
exit
ubuntu@ip-172-31-0-105:~$ docker logs my-container
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
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
2025/05/10 17:12:30 [notice] 1#1: using the "epoll" event method
2025/05/10 17:12:30 [notice] 1#1: nginx/1.27.5
2025/05/10 17:12:30 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)
2025/05/10 17:12:30 [notice] 1#1: OS: Linux 6.8.0-1024-aws
2025/05/10 17:12:30 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2025/05/10 17:12:30 [notice] 1#1: start worker processes
2025/05/10 17:12:30 [notice] 1#1: start worker process 29
ubuntu@ip-172-31-0-105:~$
```

Inspect a volume

# View container stats (CPU, memory usage)

docker stats



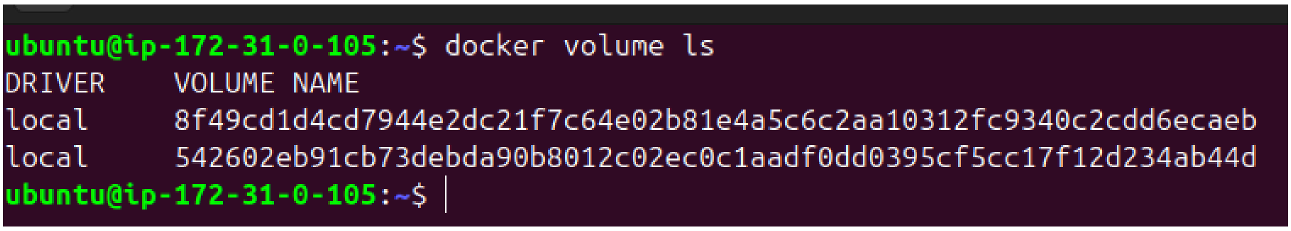
```
CONTAINER ID   NAME           CPU %     MEM USAGE / LIMIT     MEM %     NET I/O          BLOCK I/O         PIDS
ef22f0288b3f   my-container   0.00%     3.582MiB / 957.4MiB   0.37%     1.09kB / 126B    1.23MB / 16.4kB   2
```

Docker Volumes

bash

# List all volumes

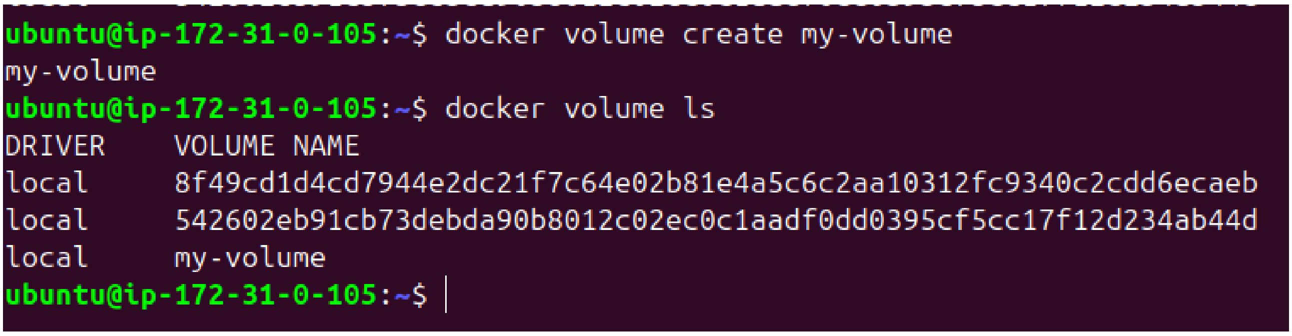
docker volume ls



```
ubuntu@ip-172-31-0-105:~$ docker volume ls
DRIVER    VOLUME NAME
local     8f49cd1d4cd7944e2dc21f7c64e02b81e4a5c6c2aa10312fc9340c2cdd6ecaeb
local     542602eb91cb73debda90b8012c02ec0c1aadf0dd0395cf5cc17f12d234ab44d
ubuntu@ip-172-31-0-105:~$
```

# Create a volume

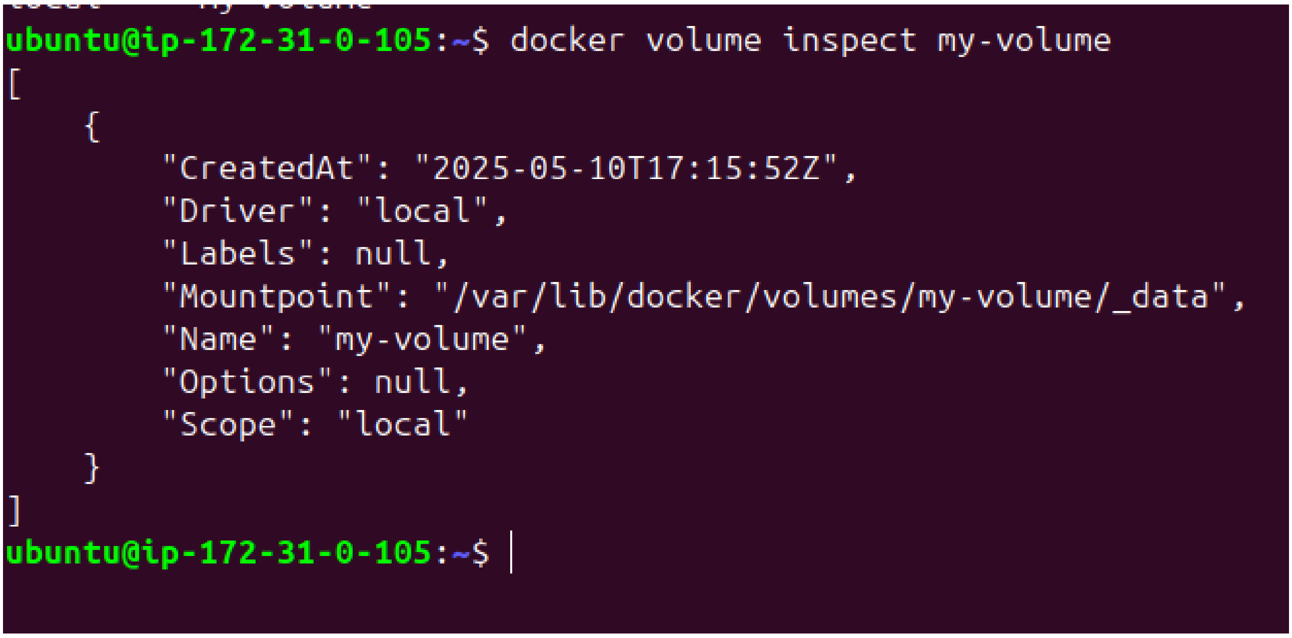
docker volume create my-volume



```
ubuntu@ip-172-31-0-105:~$ docker volume create my-volume
my-volume
ubuntu@ip-172-31-0-105:~$ docker volume ls
DRIVER    VOLUME NAME
local     8f49cd1d4cd7944e2dc21f7c64e02b81e4a5c6c2aa10312fc9340c2cdd6ecaeb
local     542602eb91cb73debda90b8012c02ec0c1aadf0dd0395cf5cc17f12d234ab44d
local     my-volume
ubuntu@ip-172-31-0-105:~$
```

# Inspect a volume

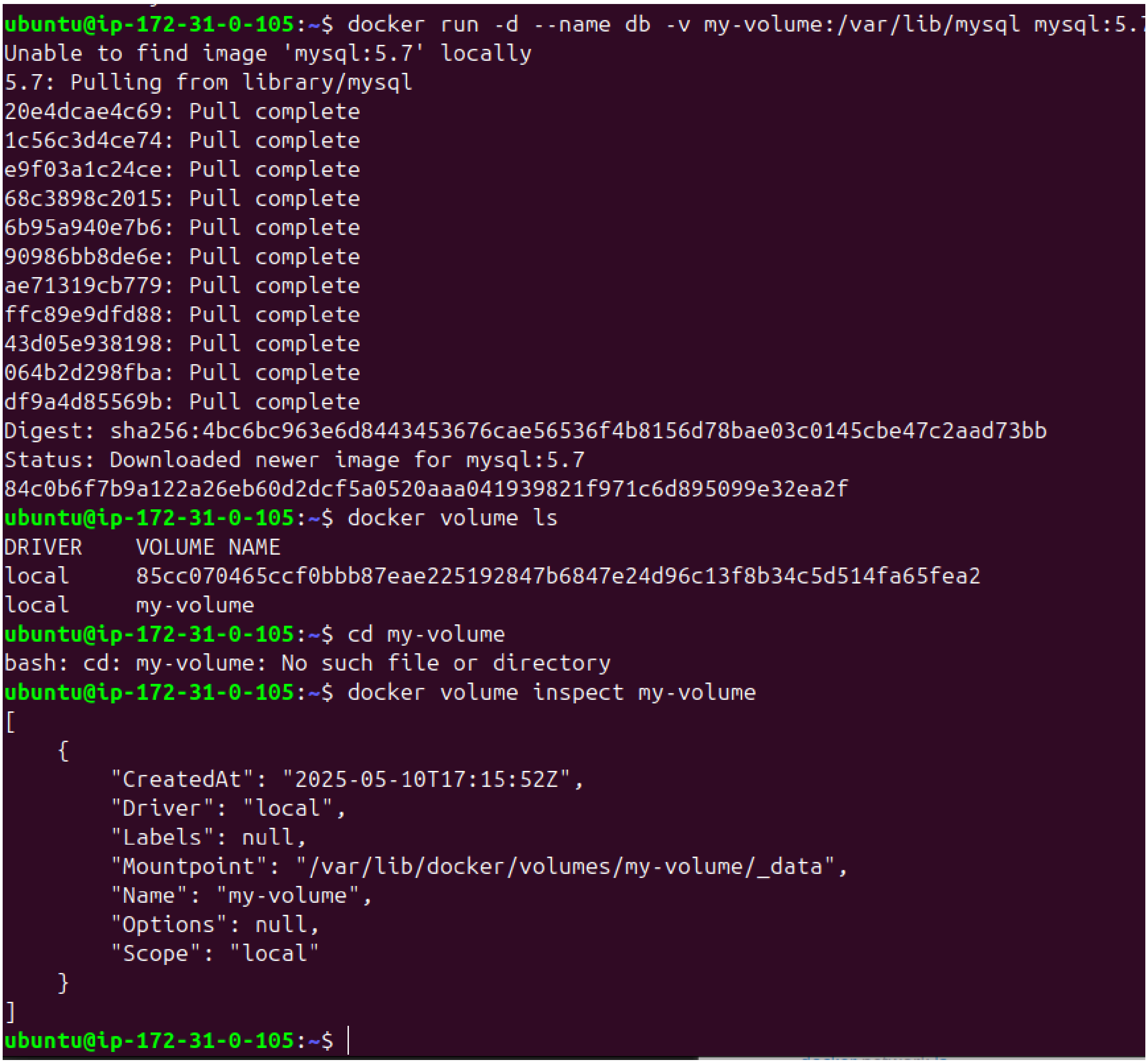
```
docker volume inspect my-volume
```



```
ubuntu@ip-172-31-0-105:~$ docker volume inspect my-volume
[
    {
        "CreatedAt": "2025-05-10T17:15:52Z",
        "Driver": "local",
        "Labels": null,
        "Mountpoint": "/var/lib/docker/volumes/my-volume/_data",
        "Name": "my-volume",
        "Options": null,
        "Scope": "local"
    }
]
ubuntu@ip-172-31-0-105:~$
```

# Run a container with a volume

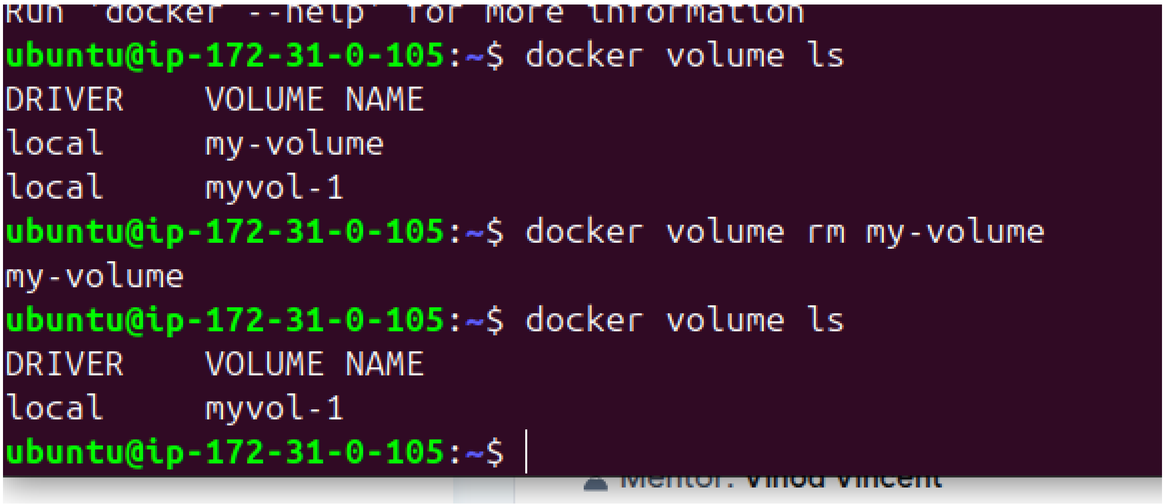
```
docker run -d --name db -v my-volume:/var/lib/mysql mysql:5.7
```



```
ubuntu@ip-172-31-0-105:~$ docker run -d --name db -v my-volume:/var/lib/mysql mysql:5.7
Unable to find image 'mysql:5.7' locally
5.7: Pulling from library/mysql
20e4dcae4c69: Pull complete
1c56c3d4ce74: Pull complete
e9f03a1c24ce: Pull complete
68c3898c2015: Pull complete
6b95a940e7b6: Pull complete
90986bb8de6e: Pull complete
ae71319cb779: Pull complete
ffc89e9dfd88: Pull complete
43d05e938198: Pull complete
064b2d298fba: Pull complete
df9a4d85569b: Pull complete
Digest: sha256:4bc6bc963e6d8443453676cae56536f4b8156d78bae03c0145cbe47c2aad73bb
Status: Downloaded newer image for mysql:5.7
84c0b6f7b9a122a26eb60d2dcf5a0520aaa041939821f971c6d895099e32ea2f
ubuntu@ip-172-31-0-105:~$ docker volume ls
DRIVER    VOLUME NAME
local     85cc070465ccf0bbb87eae225192847b6847e24d96c13f8b34c5d514fa65fea2
local     my-volume
ubuntu@ip-172-31-0-105:~$ cd my-volume
bash: cd: my-volume: No such file or directory
ubuntu@ip-172-31-0-105:~$ docker volume inspect my-volume
[
    {
        "CreatedAt": "2025-05-10T17:15:52Z",
        "Driver": "local",
        "Labels": null,
        "Mountpoint": "/var/lib/docker/volumes/my-volume/_data",
        "Name": "my-volume",
        "Options": null,
        "Scope": "local"
    }
]
ubuntu@ip-172-31-0-105:~$
```

# Remove a volume

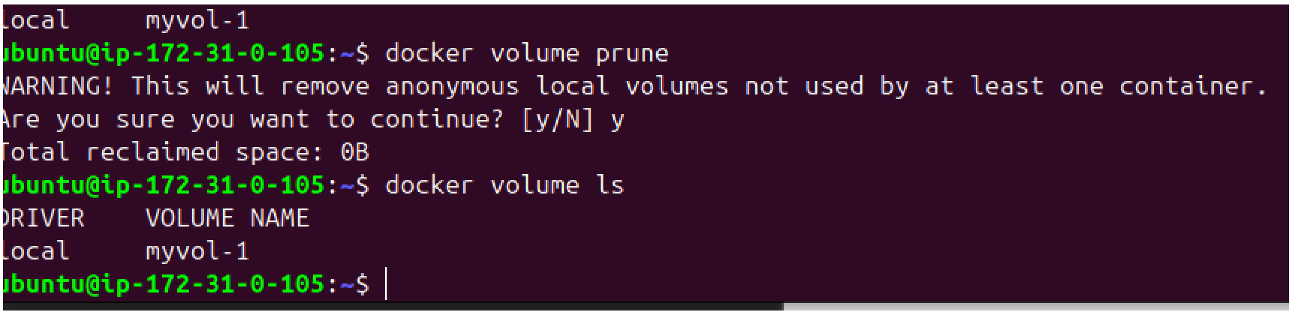
docker volume rm my-volume



```
Run 'docker --help' for more information
ubuntu@ip-172-31-0-105:~$ docker volume ls
DRIVER    VOLUME NAME
local     my-volume
local     myvol-1
ubuntu@ip-172-31-0-105:~$ docker volume rm my-volume
my-volume
ubuntu@ip-172-31-0-105:~$ docker volume ls
DRIVER    VOLUME NAME
local     myvol-1
ubuntu@ip-172-31-0-105:~$
```

# Clean up unused volumes

docker volume prune



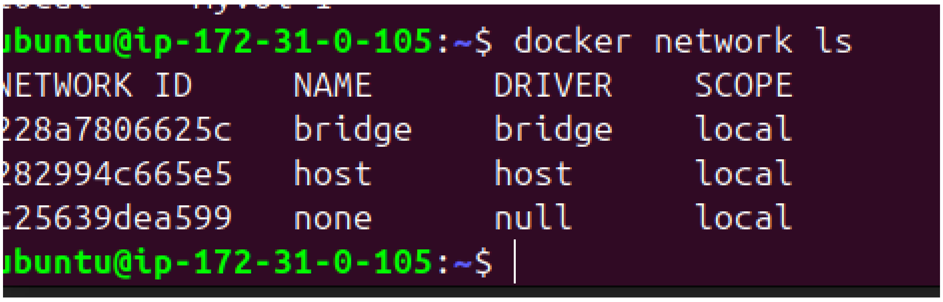
```
local     myvol-1
ubuntu@ip-172-31-0-105:~$ docker volume prune
WARNING! This will remove anonymous local volumes not used by at least one container.
Are you sure you want to continue? [y/N] y
Total reclaimed space: 0B
ubuntu@ip-172-31-0-105:~$ docker volume ls
DRIVER    VOLUME NAME
local     myvol-1
ubuntu@ip-172-31-0-105:~$ 
```

Docker Networks

bash

# List all networks

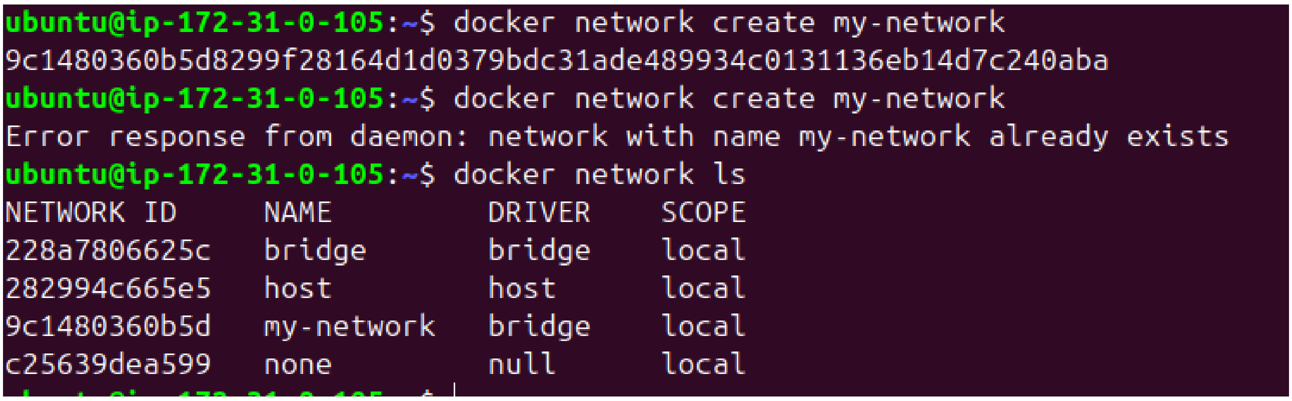
docker network ls



```
ubuntu@ip-172-31-0-105:~$ docker network ls
NETWORK ID     NAME      DRIVER    SCOPE
228a7806625c   bridge    bridge    local
282994c665e5   host      host      local
c25639dea599   none      null      local
ubuntu@ip-172-31-0-105:~$ 
```

# Create a network

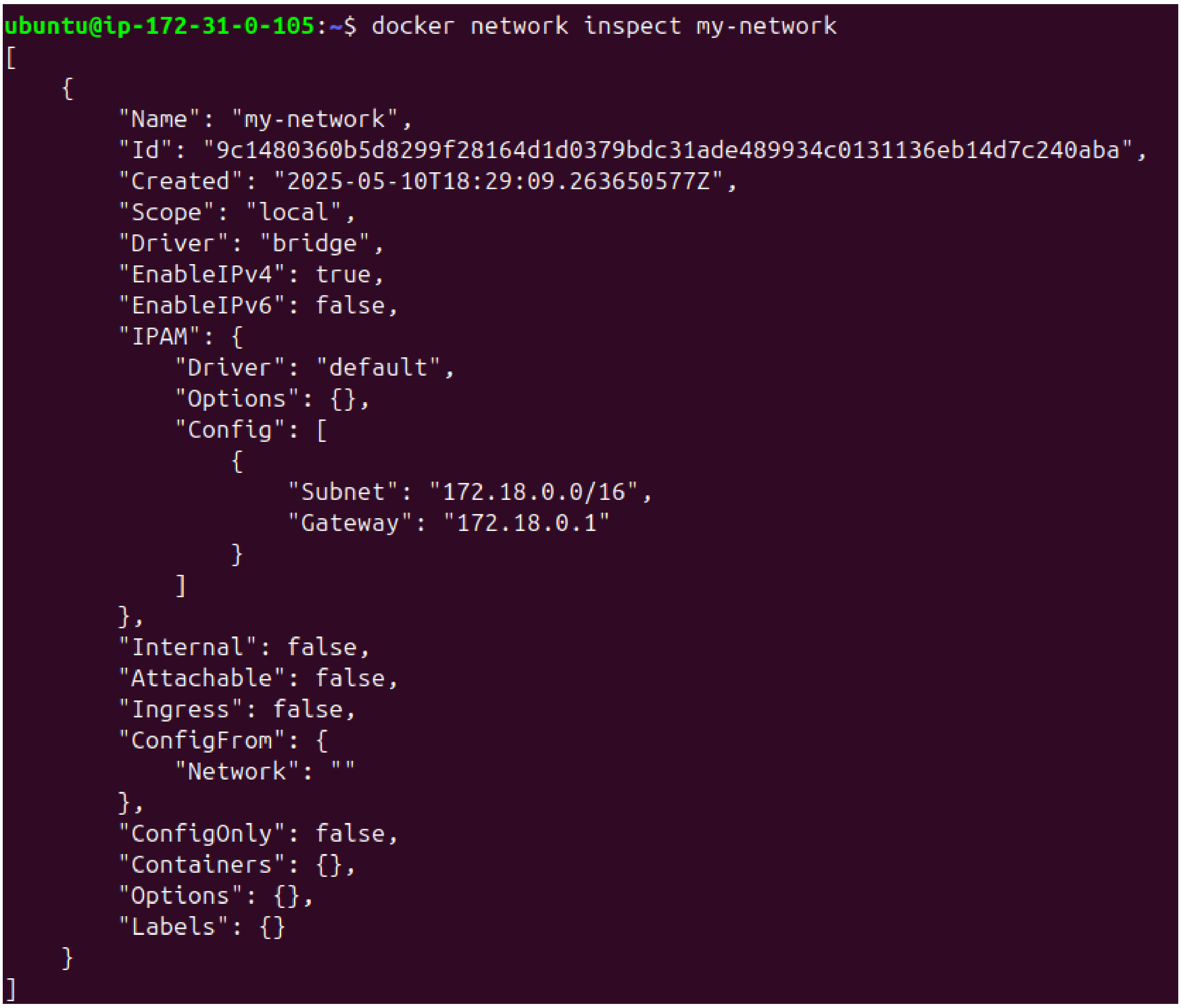
docker network create my-network



```
ubuntu@ip-172-31-0-105:~$ docker network create my-network
9c1480360b5d8299f28164d1d0379bdc31ade489934c0131136eb14d7c240aba
ubuntu@ip-172-31-0-105:~$ docker network create my-network
Error response from daemon: network with name my-network already exists
ubuntu@ip-172-31-0-105:~$ docker network ls
NETWORK ID     NAME         DRIVER    SCOPE
228a7806625c   bridge       bridge    local
282994c665e5   host         host      local
9c1480360b5d   my-network   bridge    local
c25639dea599   none         null      local
```

# Inspect a network

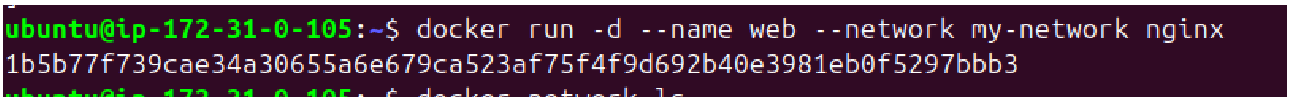
docker network inspect my-network



```
ubuntu@ip-172-31-0-105:~$ docker network inspect my-network
[
    {
        "Name": "my-network",
        "Id": "9c1480360b5d8299f28164d1d0379bdc31ade489934c0131136eb14d7c240aba",
        "Created": "2025-05-10T18:29:09.263650577Z",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv4": true,
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": {},
            "Config": [
                {
                    "Subnet": "172.18.0.0/16",
                    "Gateway": "172.18.0.1"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {},
        "Options": {},
        "Labels": {}
    }
]
```

# Run a container in a custom network

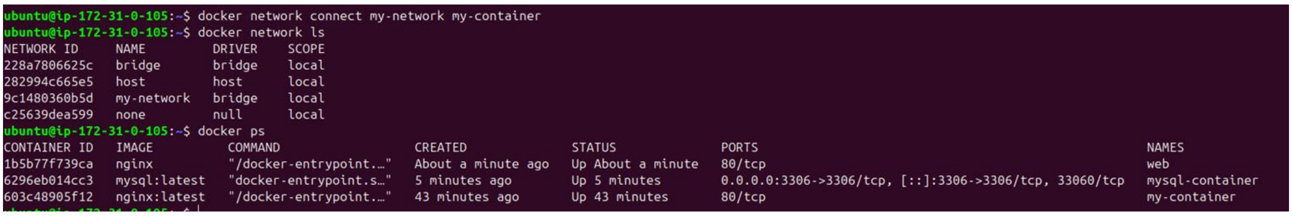
docker run -d --name web --network my-network nginx



```
ubuntu@ip-172-31-0-105:~$ docker run -d --name web --network my-network nginx
1b5b77f739cae34a30655a6e679ca523af75f4f9d692b40e3981eb0f5297bbb3
```

# Connect a container to a network

docker network connect my-network my-container



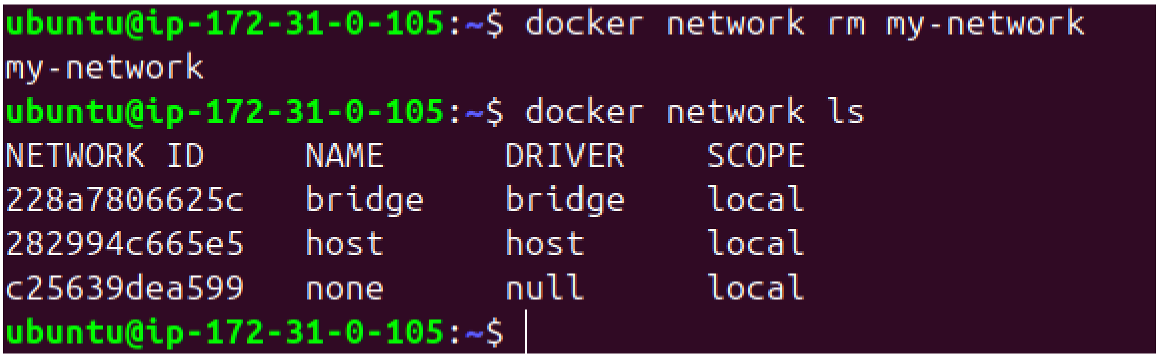
```
ubuntu@ip-172-31-0-105:~$ docker network connect my-network my-container
ubuntu@ip-172-31-0-105:~$ docker network ls
NETWORK ID     NAME         DRIVER    SCOPE
228a7806625c   bridge       bridge    local
282994c665e5   host         host      local
9c1480360b5d   my-network   bridge    local
c25639dea599   none         null      local
ubuntu@ip-172-31-0-105:~$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED              STATUS              PORTS                                                    NAMES
1b5b77f739ca   nginx          "/docker-entrypoint.…"   About a minute ago   Up About a minute   80/tcp                                                   web
6296eb014cc3   mysql:latest   "docker-entrypoint.s…"   5 minutes ago        Up 5 minutes        0.0.0.0:3306->3306/tcp, [::]:3306->3306/tcp, 33060/tcp   mysql-container
603c48905f12   nginx:latest   "/docker-entrypoint.…"   43 minutes ago       Up 43 minutes       80/tcp                                                   my-container
```

# Disconnect a container from a network

docker network disconnect my-network my-container

# Remove a network

```
docker network rm my-network
```



```
ubuntu@ip-172-31-0-105:~$ docker network rm my-network
my-network
ubuntu@ip-172-31-0-105:~$ docker network ls
NETWORK ID     NAME      DRIVER    SCOPE
228a7806625c   bridge    bridge    local
282994c665e5   host      host      local
c25639dea599   none      null      local
ubuntu@ip-172-31-0-105:~$
```

Docker Compose (Bonus)

If you want to manage multi-container applications, you might want to install Docker Compose:

bash

```
# Install Docker Compose
sudo curl -L "https://github.com/docker/compose/releases/download/v2.20.3/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
sudo chmod +x /usr/local/bin/docker-compose

# Verify installation
docker-compose --version
```

Docker System Commands

bash

```
# View Docker system information
docker info

# View Docker disk usage
docker system df

# Prune unused Docker objects (containers, images, networks, volumes)
docker system prune -a
```