Curso: Docker

Homework 1 - Docker Core Concepts

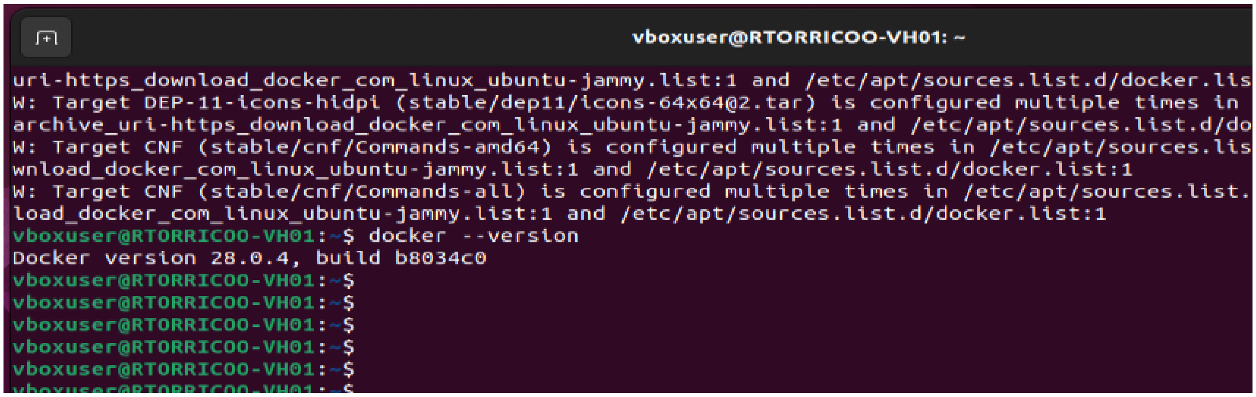
Instructor: Caleb Espinoza Gutierrez

Estudiante: Ronald Torrico Ovando

Exercise 1

1. Install Docker Engine using the official Docker APT repository.

```
# I am using my local Ubuntu Desktop 22.04.5 LTS
# Step 1: Update the apt package index
sudo apt-get update

# Step 2: Install required packages
sudo apt-get install \
  ca-certificates \
  curl \
  gnupg \
  lsb-release

# Step 3: Add Docker's official GPG key
sudo mkdir -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | \
  sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg

# Step 4: Set up the Docker APT repository
echo \
 "deb [arch=$(dpkg --print-architecture) \
 signed-by=/etc/apt/keyrings/docker.gpg] \
 https://download.docker.com/linux/ubuntu \
 $(lsb_release -cs) stable" | \
 sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

# Step 5: Update the package index again and install Docker Engine
sudo apt-get update

# Step 6: Install Docker Engine
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin

# Verify Docker versión
docker --version
```



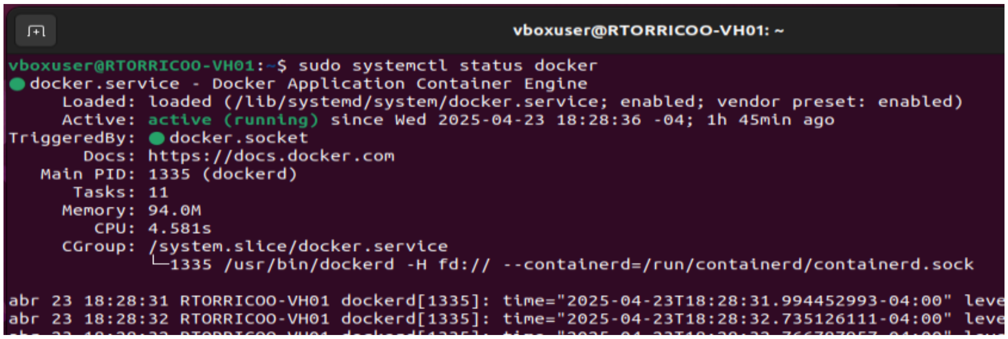
vboxuser@RTORRICOO-VH01: ~

```
uri-https_download_docker_com_linux_ubuntu-jammy.list:1 and /etc/apt/sources.list.d/docker.lis
W: Target DEP-11-icons-hidpi (stable/dep11/icons-64x64@2.tar) is configured multiple times in
archive_uri-https_download_docker_com_linux_ubuntu-jammy.list:1 and /etc/apt/sources.list.d/do
W: Target CNF (stable/cnf/Commands-amd64) is configured multiple times in /etc/apt/sources.lis
wnload_docker_com_linux_ubuntu-jammy.list:1 and /etc/apt/sources.list.d/docker.list:1
W: Target CNF (stable/cnf/Commands-all) is configured multiple times in /etc/apt/sources.list.
load_docker_com_linux_ubuntu-jammy.list:1 and /etc/apt/sources.list.d/docker.list:1
vboxuser@RTORRICOO-VH01:~$ docker --version
Docker version 28.0.4, build b8034c0
vboxuser@RTORRICOO-VH01:~$
vboxuser@RTORRICOO-VH01:~$
vboxuser@RTORRICOO-VH01:~$
vboxuser@RTORRICOO-VH01:~$
vboxuser@RTORRICOO-VH01:~$
vboxuser@RTORRICOO-VH01:~$
```

2. Enable and start the Docker service.

```
# Enable Docker to start on boot
sudo systemctl enable docker

# Start Docker now
sudo systemctl start docker
```

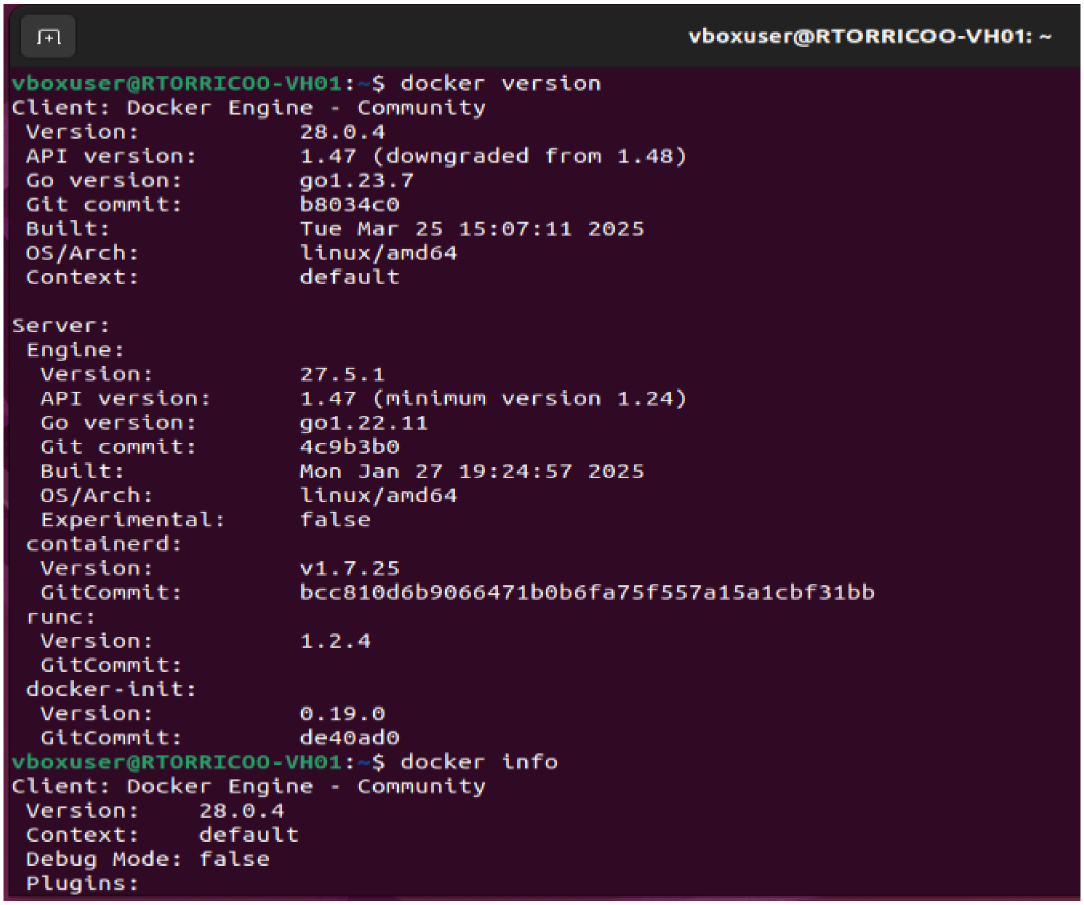
```
# Check Docker status
sudo systemctl status Docker
```



```
vboxuser@RTORRICOO-VH01:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2025-04-23 18:28:36 -04; 1h 45min ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 1335 (dockerd)
      Tasks: 11
     Memory: 94.0M
        CPU: 4.581s
     CGroup: /system.slice/docker.service
             └─1335 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

abr 23 18:28:31 RTORRICOO-VH01 dockerd[1335]: time="2025-04-23T18:28:31.994452993-04:00" leve
abr 23 18:28:32 RTORRICOO-VH01 dockerd[1335]: time="2025-04-23T18:28:32.735126111-04:00" leve
```

3. Print the info of the Docker Client and Server.

```
# Show Docker version info
docker version

# Show detailed Docker system info
docker info
```



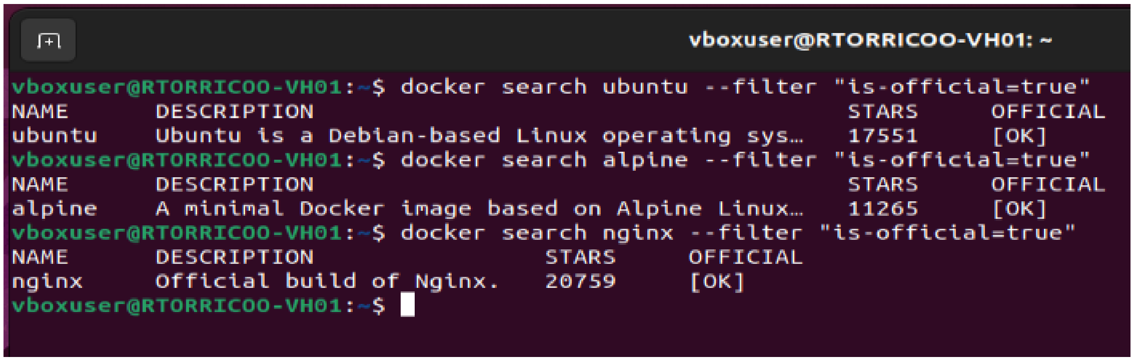
```
vboxuser@RTORRICOO-VH01:~$ docker version
Client: Docker Engine - Community
 Version:           28.0.4
 API version:       1.47 (downgraded from 1.48)
 Go version:        go1.23.7
 Git commit:        b8034c0
 Built:             Tue Mar 25 15:07:11 2025
 OS/Arch:           linux/amd64
 Context:           default

Server:
 Engine:
  Version:          27.5.1
  API version:      1.47 (minimum version 1.24)
  Go version:       go1.22.11
  Git commit:       4c9b3b0
  Built:            Mon Jan 27 19:24:57 2025
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
  Version:          v1.7.25
  GitCommit:        bcc810d6b9066471b0b6fa75f557a15a1cbf31bb
 runc:
  Version:          1.2.4
  GitCommit:
 docker-init:
  Version:          0.19.0
  GitCommit:        de40ad0
vboxuser@RTORRICOO-VH01:~$ docker info
Client: Docker Engine - Community
 Version:    28.0.4
 Context:    default
 Debug Mode: false
 Plugins:
```

Exercise 2

1. Search for the official repos of Ubuntu, Alpine, Nginx.

```
# Search official images
docker search ubuntu --filter "is-official=true"
```

docker search alpine --filter "is-official=true"
docker search nginx --filter "is-official=true"



```
vboxuser@RTORRICOO-VH01:~$ docker search ubuntu --filter "is-official=true"
NAME      DESCRIPTION                                     STARS     OFFICIAL
ubuntu    Ubuntu is a Debian-based Linux operating sys…   17551     [OK]
vboxuser@RTORRICOO-VH01:~$ docker search alpine --filter "is-official=true"
NAME      DESCRIPTION                                     STARS     OFFICIAL
alpine    A minimal Docker image based on Alpine Linux…   11265     [OK]
vboxuser@RTORRICOO-VH01:~$ docker search nginx --filter "is-official=true"
NAME      DESCRIPTION                STARS     OFFICIAL
nginx     Official build of Nginx.   20759     [OK]
vboxuser@RTORRICOO-VH01:~$ 
```

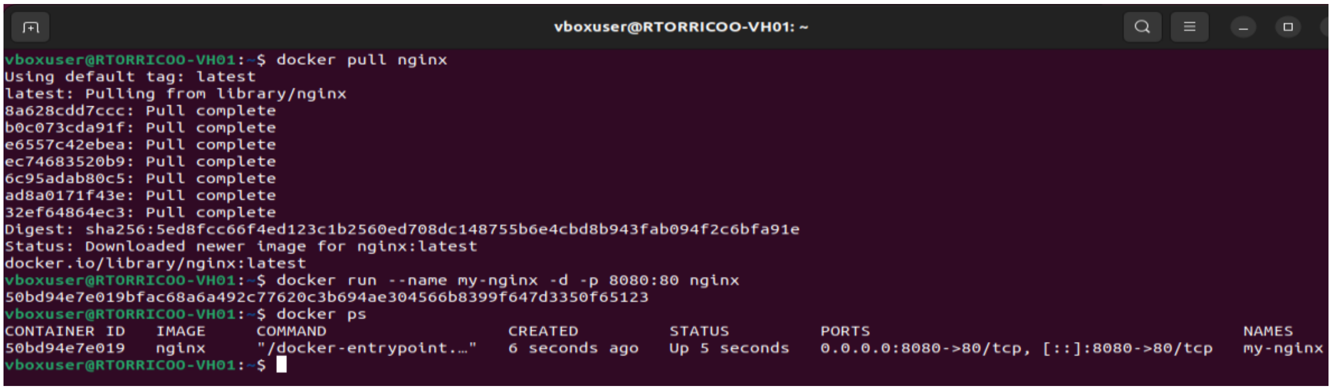
2. Run an Nginx container using the image from the official repo.

# Pull the Official Nginx Image
docker pull nginx

# Run the Nginx Container
docker run --name my-nginx -d -p 8080:80 nginx

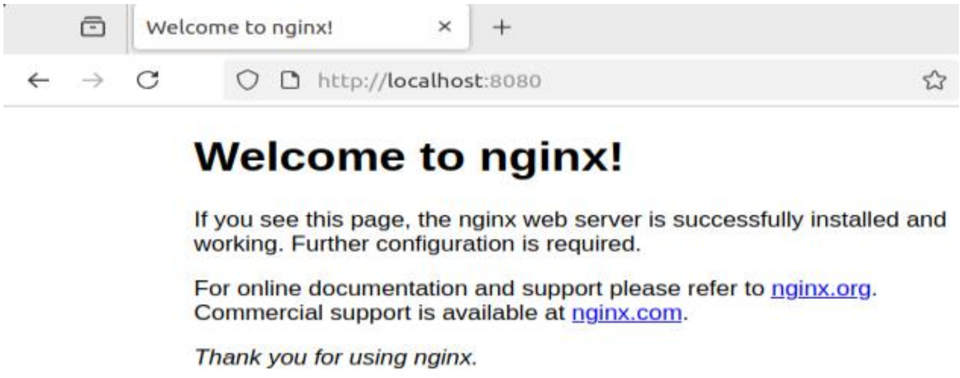
- ***--name my-nginx*** *→ Names the container* ***my-nginx****.*
- ***-d*** *→ Runs in detached mode (background).*
- ***-p 8080:80*** *→ Maps host port* ***8080*** *to container port* ***80***

# Verify It's Running
docker ps



```
vboxuser@RTORRICOO-VH01:~$ docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
8a628cdd7ccc: Pull complete
b0c073cda91f: Pull complete
e6557c42ebea: Pull complete
ec74683520b9: Pull complete
6c95adab80c5: Pull complete
ad8a0171f43e: Pull complete
32ef64864ec3: Pull complete
Digest: sha256:5ed8fcc66f4ed123c1b2560ed708dc148755b6e4cbd8b943fab094f2c6bfa91e
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
vboxuser@RTORRICOO-VH01:~$ docker run --name my-nginx -d -p 8080:80 nginx
50bd94e7e019bfac68a6a492c77620c3b694ae304566b8399f647d3350f65123
vboxuser@RTORRICOO-VH01:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS                                     NAMES
50bd94e7e019   nginx     "/docker-entrypoint.…"   6 seconds ago   Up 5 seconds   0.0.0.0:8080->80/tcp, [::]:8080->80/tcp   my-nginx
vboxuser@RTORRICOO-VH01:~$ 
```

# Access Nginx in your browser



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

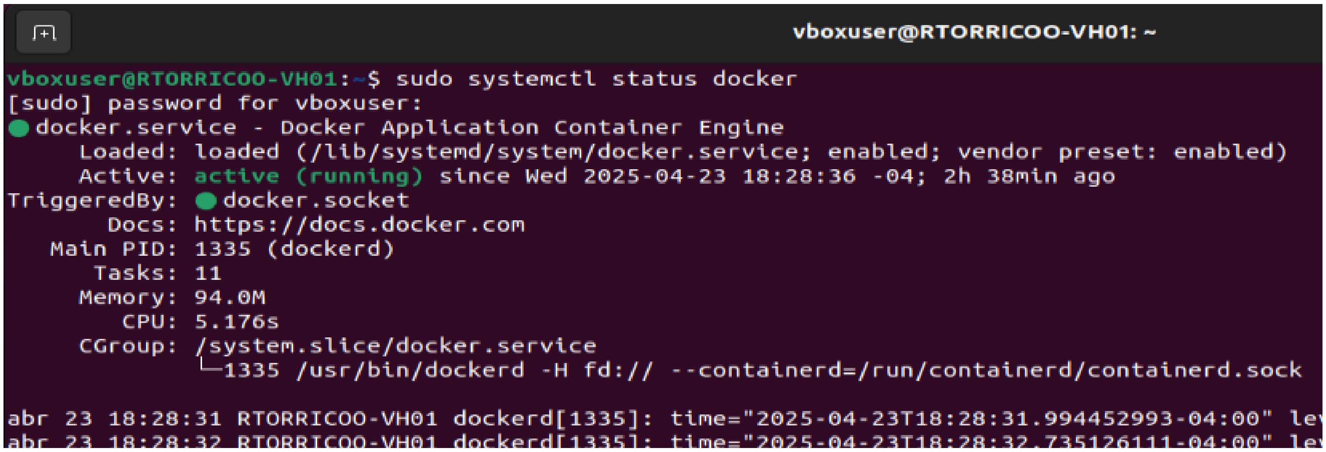
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Exercise 3

1. Check Docker daemon status

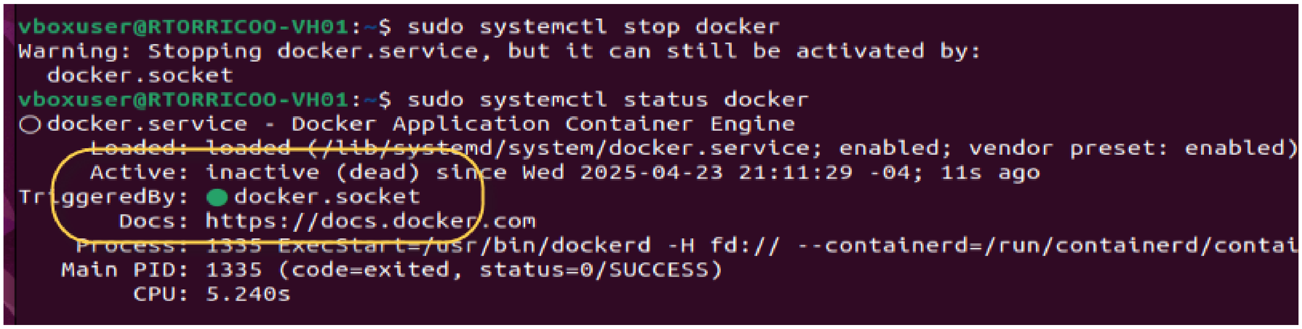
```
sudo systemctl status docker
```



```
vboxuser@RTORRICOO-VH01:~$ sudo systemctl status docker
[sudo] password for vboxuser:
● docker.service - Docker Application Container Engine
     Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2025-04-23 18:28:36 -04; 2h 38min ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 1335 (dockerd)
      Tasks: 11
     Memory: 94.0M
        CPU: 5.176s
     CGroup: /system.slice/docker.service
             └─1335 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

abr 23 18:28:31 RTORRICOO-VH01 dockerd[1335]: time="2025-04-23T18:28:31.994452993-04:00" lev
abr 23 18:28:32 RTORRICOO-VH01 dockerd[1335]: time="2025-04-23T18:28:32.735126111-04:00" lev
```

2. Stop the Docker Daemon

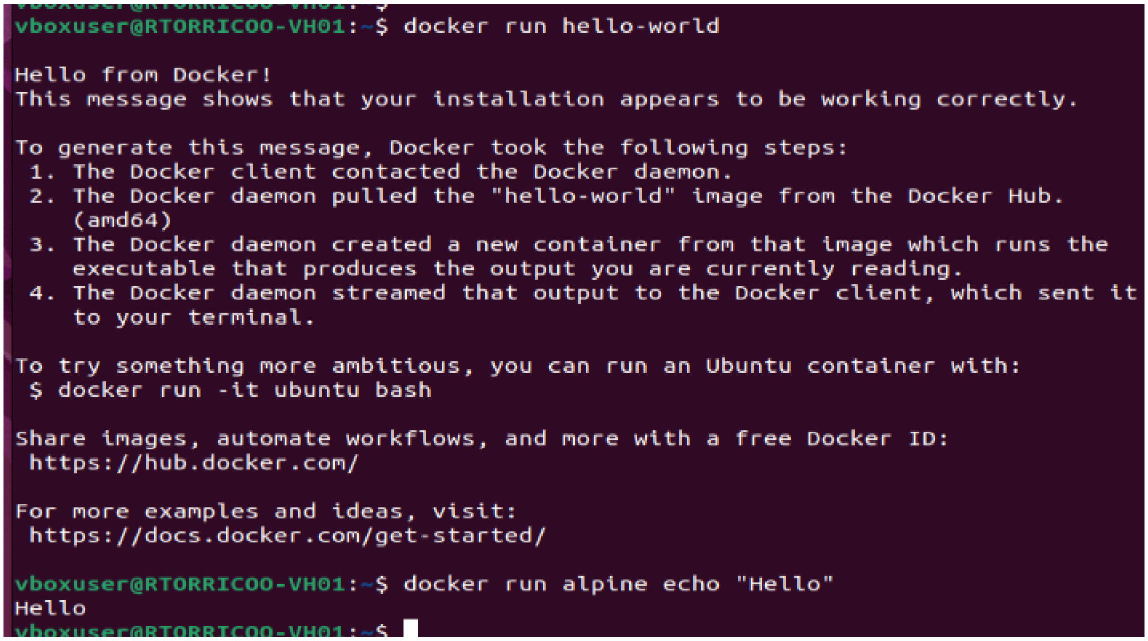
```
sudo systemctl stop docker
# To full stop
sudo systemctl stop docker.socket
sudo systemctl stop docker.service

# verify it is stopped
sudo systemctl status Docker
```



```
vboxuser@RTORRICOO-VH01:~$ sudo systemctl stop docker
Warning: Stopping docker.service, but it can still be activated by:
  docker.socket
vboxuser@RTORRICOO-VH01:~$ sudo systemctl status docker
○ docker.service - Docker Application Container Engine
     Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
     Active: inactive (dead) since Wed 2025-04-23 21:11:29 -04; 11s ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
    Process: 1335 ExecStart=/usr/bin/dockerd -H fd:// --containerd=/run/containerd/contai
   Main PID: 1335 (code=exited, status=0/SUCCESS)
        CPU: 5.240s
```

3. Run a container while it’s stopped.

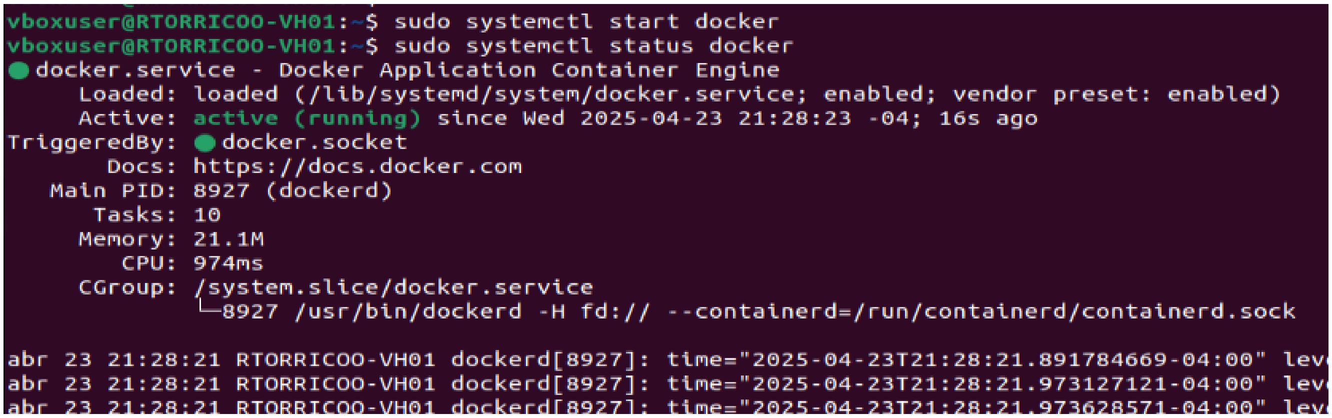
```
# Try running an Alpine container
docker run alpine echo "Hello"
docker run hello-world
```



```
vboxuser@RTORRICOO-VH01:~$ docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/

vboxuser@RTORRICOO-VH01:~$ docker run alpine echo "Hello"
Hello
vboxuser@RTORRICOO-VH01:~$
```

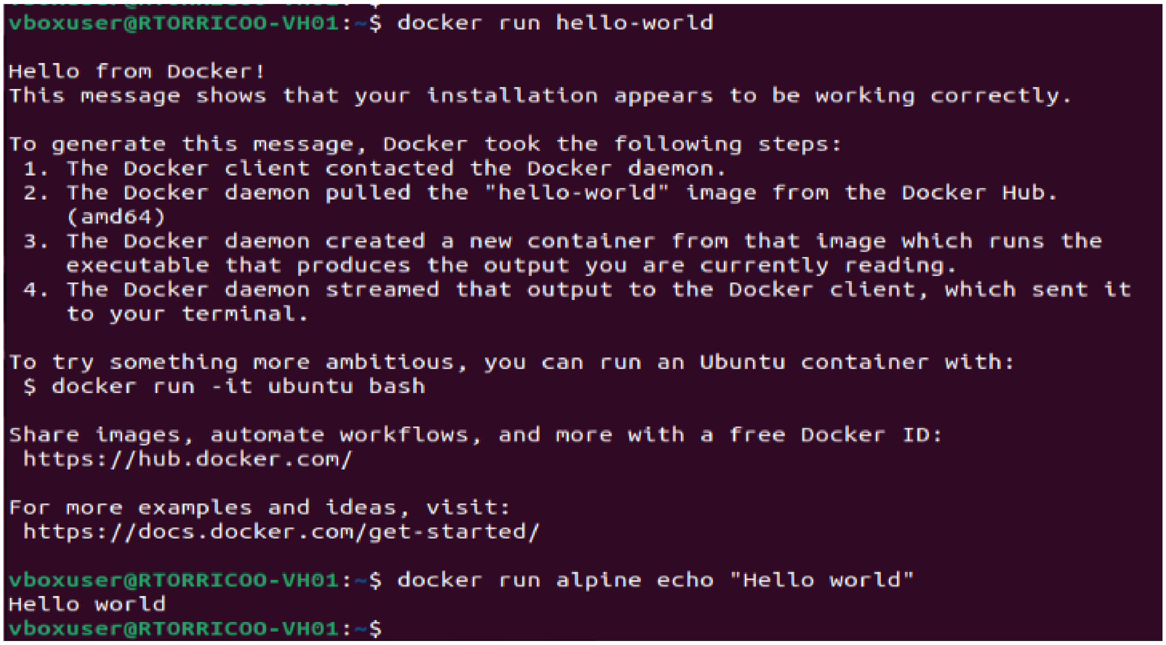
4. Restart the Docker daemon and run a container again.

```
sudo systemctl start Docker
sudo systemctl status Docker
```



```
vboxuser@RTORRICOO-VH01:~$ sudo systemctl start docker
vboxuser@RTORRICOO-VH01:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2025-04-23 21:28:23 -04; 16s ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 8927 (dockerd)
      Tasks: 10
     Memory: 21.1M
        CPU: 974ms
     CGroup: /system.slice/docker.service
             └─8927 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

abr 23 21:28:21 RTORRICOO-VH01 dockerd[8927]: time="2025-04-23T21:28:21.891784669-04:00" lev
abr 23 21:28:21 RTORRICOO-VH01 dockerd[8927]: time="2025-04-23T21:28:21.973127121-04:00" lev
abr 23 21:28:21 RTORRICOO-VH01 dockerd[8927]: time="2025-04-23T21:28:21.973628571-04:00" lev
```

```
# Now try to run a container again
docker run hello-world
docker run alpine echo "Hello, Docker is back online!"
```



```
vboxuser@RTORRICOO-VH01:~$ docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/

vboxuser@RTORRICOO-VH01:~$ docker run alpine echo "Hello world"
Hello world
vboxuser@RTORRICOO-VH01:~$
```

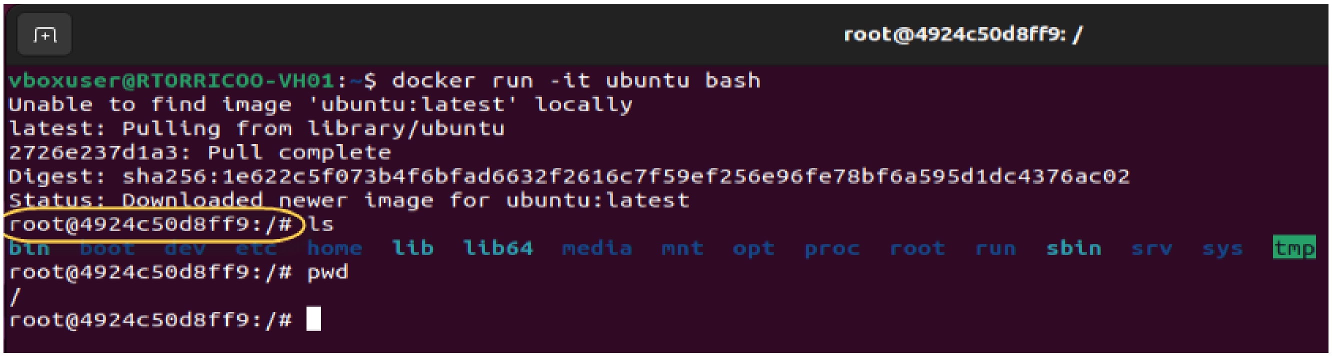
Exercise 4

1. Run an Ubuntu container interactively.

```
# Start an interactive Ubuntu container with a Bash shell
```

docker run -it ubuntu bash

- ***-it*** → *Allocates an interactive terminal (-i for interactive, -t for pseudo-TTY).*
- ***ubuntu*** → *Uses the official Ubuntu image (downloads if not cached).*
- ***bash*** → *Launches the Bash shell inside the container.*



```
vboxuser@RTORRICOO-VH01:~$ docker run -it ubuntu bash
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
2726e237d1a3: Pull complete
Digest: sha256:1e622c5f073b4f6bfad6632f2616c7f59ef256e96fe78bf6a595d1dc4376ac02
Status: Downloaded newer image for ubuntu:latest
root@4924c50d8ff9:/# ls
bin  boot  dev  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp
root@4924c50d8ff9:/# pwd
/
root@4924c50d8ff9:/# 
```

2. Use apt update && apt install curl inside the container.

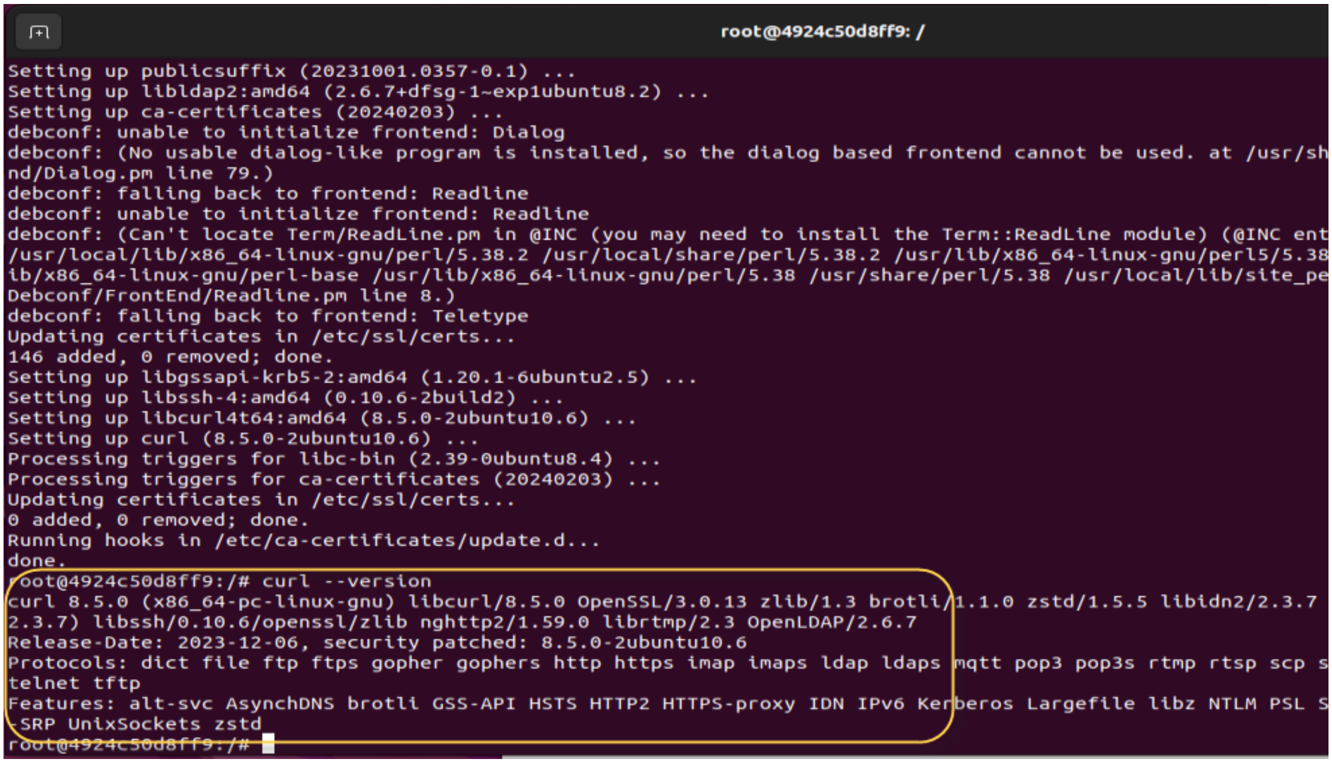
```
# Inside the container, run
```

apt update && apt install -y curl

- ***apt update*** → *Updates the package list.*
- ***apt install -y curl*** → *Installs **curl** (the -y flag auto-confirms prompts).*

```
# Verify installation
```

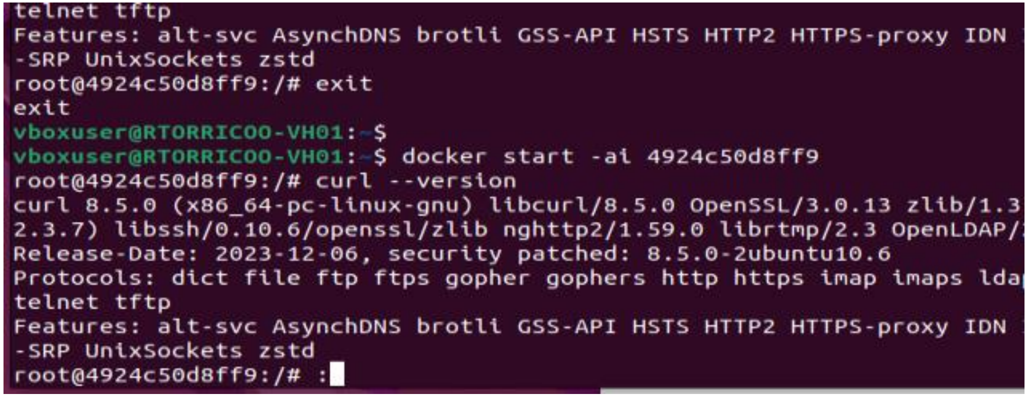
curl --version



```
Setting up publicsuffix (20231001.0357-0.1) ...
Setting up libldap2:amd64 (2.6.7+dfsg-1~exp1ubuntu8.2) ...
Setting up ca-certificates (20240203) ...
debconf: unable to initialize frontend: Dialog
debconf: (No usable dialog-like program is installed, so the dialog based frontend cannot be used. at /usr/sh
nd/Dialog.pm line 79.)
debconf: falling back to frontend: Readline
debconf: unable to initialize frontend: Readline
debconf: (Can't locate Term/ReadLine.pm in @INC (you may need to install the Term::ReadLine module) (@INC ent
/usr/local/lib/x86_64-linux-gnu/perl/5.38.2 /usr/local/share/perl/5.38.2 /usr/lib/x86_64-linux-gnu/perl5/5.38
ib/x86_64-linux-gnu/perl-base /usr/lib/x86_64-linux-gnu/perl/5.38 /usr/share/perl/5.38 /usr/local/lib/site_pe
Debconf/FrontEnd/Readline.pm line 8.)
debconf: falling back to frontend: Teletype
Updating certificates in /etc/ssl/certs...
146 added, 0 removed; done.
Setting up libgssapi-krb5-2:amd64 (1.20.1-6ubuntu2.5) ...
Setting up libssh-4:amd64 (0.10.6-2build2) ...
Setting up libcurl4t64:amd64 (8.5.0-2ubuntu10.6) ...
Setting up curl (8.5.0-2ubuntu10.6) ...
Processing triggers for libc-bin (2.39-0ubuntu8.4) ...
Processing triggers for ca-certificates (20240203) ...
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
root@4924c50d8ff9:/# curl --version
curl 8.5.0 (x86_64-pc-linux-gnu) libcurl/8.5.0 OpenSSL/3.0.13 zlib/1.3 brotli/1.1.0 zstd/1.5.5 libidn2/2.3.7
2.3.7) libssh/0.10.6/openssl/zlib nghttp2/1.59.0 librtmp/2.3 OpenLDAP/2.6.7
Release-Date: 2023-12-06, security patched: 8.5.0-2ubuntu10.6
Protocols: dict file ftp ftps gopher gophers http https imap imaps ldap ldaps mqtt pop3 pop3s rtmp rtsp scp s
telnet tftp
Features: alt-svc AsynchDNS brotli GSS-API HSTS HTTP2 HTTPS-proxy IDN IPv6 Kerberos Largefile libz NTLM PSL S
-SRP UnixSockets zstd
root@4924c50d8ff9:/#
```

3. Exit the container.

```
exit

# Restart the stopped container in detached mode
docker start -ai [container-id]
```

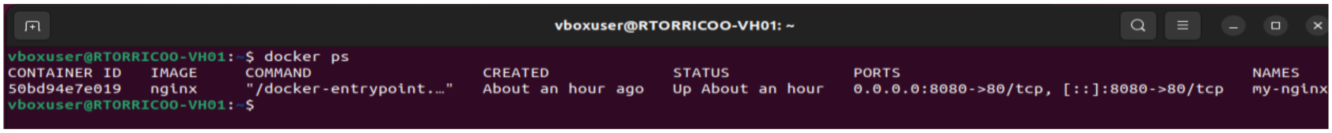


```
telnet tftp
Features: alt-svc AsynchDNS brotli GSS-API HSTS HTTP2 HTTPS-proxy IDN
-SRP UnixSockets zstd
root@4924c50d8ff9:/# exit
exit
vboxuser@RTORRICOO-VH01:~$
vboxuser@RTORRICOO-VH01:~$ docker start -ai 4924c50d8ff9
root@4924c50d8ff9:/# curl --version
curl 8.5.0 (x86_64-pc-linux-gnu) libcurl/8.5.0 OpenSSL/3.0.13 zlib/1.3
2.3.7) libssh/0.10.6/openssl/zlib nghttp2/1.59.0 librtmp/2.3 OpenLDAP/
Release-Date: 2023-12-06, security patched: 8.5.0-2ubuntu10.6
Protocols: dict file ftp ftps gopher gophers http https imap imaps lda
telnet tftp
Features: alt-svc AsynchDNS brotli GSS-API HSTS HTTP2 HTTPS-proxy IDN
-SRP UnixSockets zstd
root@4924c50d8ff9:/# :
```

Exercise 5

1. List running containers:

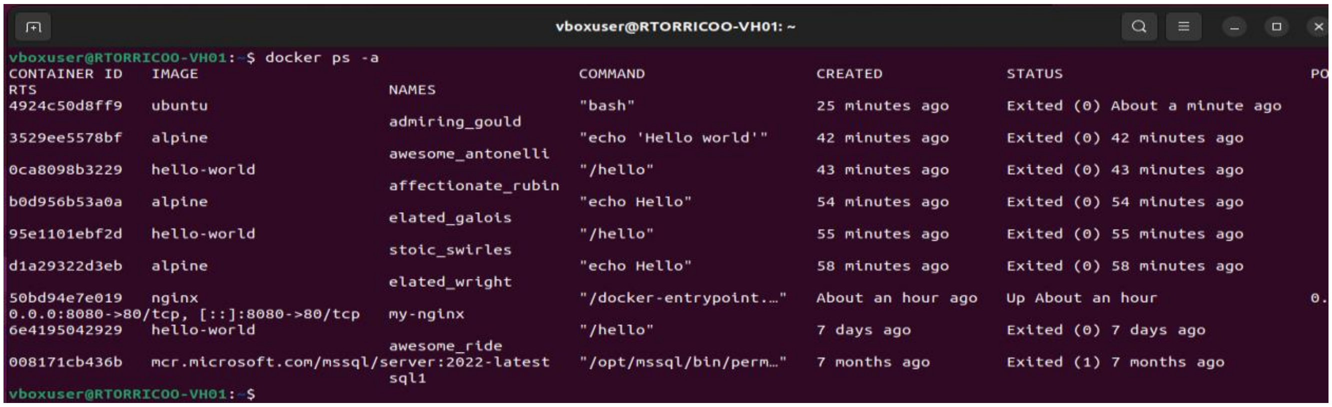
```
# To see currently running containers
docker ps
```



```
vboxuser@RTORRICOO-VH01:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED              STATUS              PORTS                                   NAMES
50bd94e7e019   nginx     "/docker-entrypoint.…"   About an hour ago    Up About an hour    0.0.0.0:8080->80/tcp, [::]:8080->80/tcp   my-nginx
vboxuser@RTORRICOO-VH01:~$
```

2. List all containers (including exited)

```
# To show all containers (running + stopped)
docker ps -a
# Filter exited containers
docker ps -a --filter "status=exited"
```



vboxuser@RTORRICOO-VH01: ~

```
vboxuser@RTORRICOO-VH01:~$ docker ps -a
CONTAINER ID   IMAGE                                        COMMAND                  CREATED             STATUS                          PO
RTS                                     NAMES
4924c50d8ff9   ubuntu                                       "bash"                   25 minutes ago      Exited (0) About a minute ago
                                        admiring_gould
3529ee5578bf   alpine                                       "echo 'Hello world'"     42 minutes ago      Exited (0) 42 minutes ago
                                        awesome_antonelli
0ca8098b3229   hello-world                                  "/hello"                 43 minutes ago      Exited (0) 43 minutes ago
                                        affectionate_rubin
b0d956b53a0a   alpine                                       "echo Hello"             54 minutes ago      Exited (0) 54 minutes ago
                                        elated_galois
95e1101ebf2d   hello-world                                  "/hello"                 55 minutes ago      Exited (0) 55 minutes ago
                                        stoic_swirles
d1a29322d3eb   alpine                                       "echo Hello"             58 minutes ago      Exited (0) 58 minutes ago
                                        elated_wright
50bd94e7e019   nginx                                        "/docker-entrypoint.…"   About an hour ago   Up About an hour                0.
0.0.0:8080->80/tcp, [::]:8080->80/tcp   my-nginx
6e4195042929   hello-world                                  "/hello"                 7 days ago          Exited (0) 7 days ago
                                        awesome_ride
008171cb436b   mcr.microsoft.com/mssql/server:2022-latest   "/opt/mssql/bin/perm…"   7 months ago        Exited (1) 7 months ago
                                        sql1
vboxuser@RTORRICOO-VH01:~$
```

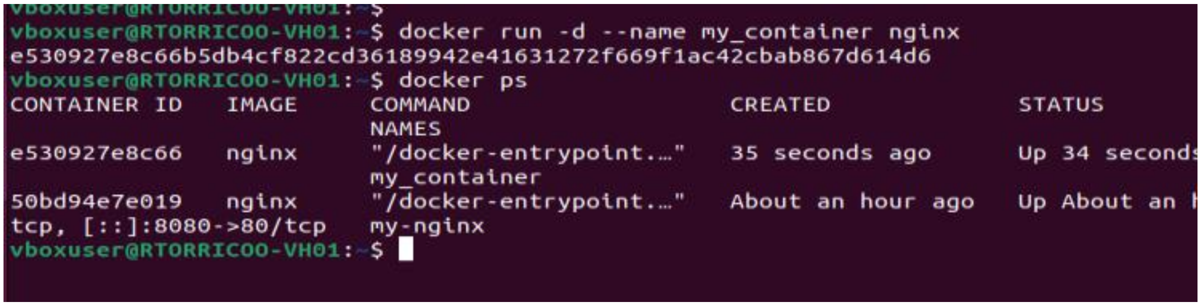
Exercise 6

1. Run a container in the background

```
# Use the -d (detached) flag to run a container in the background
docker run -d --name my_container nginx
```

- -d → Detaches the container (runs in the background).
- --name my_container → Names the container for easier reference.

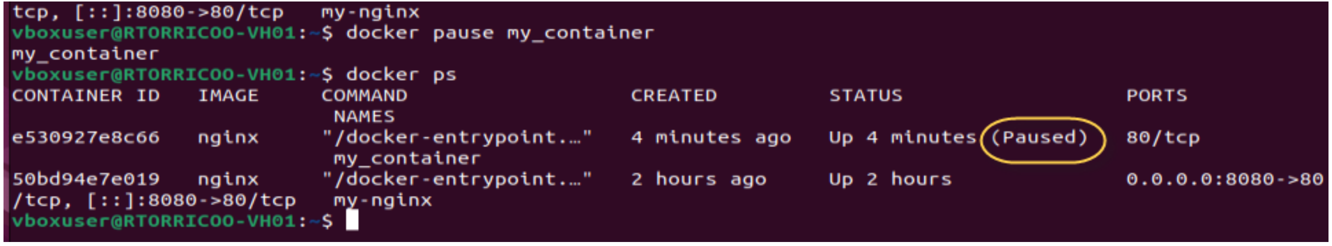
```
# Verify it's running
docker ps
```



```
vboxuser@RTORRICOO-VH01:~$
vboxuser@RTORRICOO-VH01:~$ docker run -d --name my_container nginx
e530927e8c66b5db4cf822cd36189942e41631272f669f1ac42cbab867d614d6
vboxuser@RTORRICOO-VH01:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED             STATUS
                         NAMES
e530927e8c66   nginx     "/docker-entrypoint.…"   35 seconds ago      Up 34 seconds
                         my_container
50bd94e7e019   nginx     "/docker-entrypoint.…"   About an hour ago   Up About an
tcp, [::]:8080->80/tcp   my-nginx
vboxuser@RTORRICOO-VH01:~$
```

2. Then Pause it

```
docker pause my_container

# Check status
docker ps
```

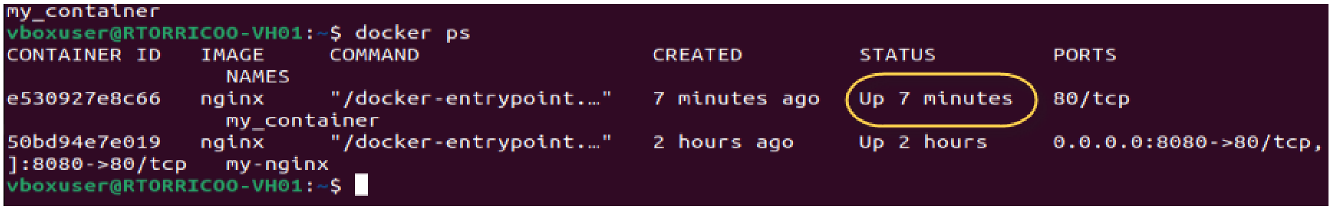


```
tcp, [::]:8080->80/tcp   my-nginx
vboxuser@RTORRICOO-VH01:~$ docker pause my_container
my_container
vboxuser@RTORRICOO-VH01:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS                  PORTS
                          NAMES
e530927e8c66   nginx     "/docker-entrypoint.…"   4 minutes ago   Up 4 minutes (Paused)   80/tcp
                          my_container
50bd94e7e019   nginx     "/docker-entrypoint.…"   2 hours ago     Up 2 hours              0.0.0.0:8080->80
/tcp, [::]:8080->80/tcp   my-nginx
vboxuser@RTORRICOO-VH01:~$
```

3. Unpause it:

```
docker unpause my_container
```

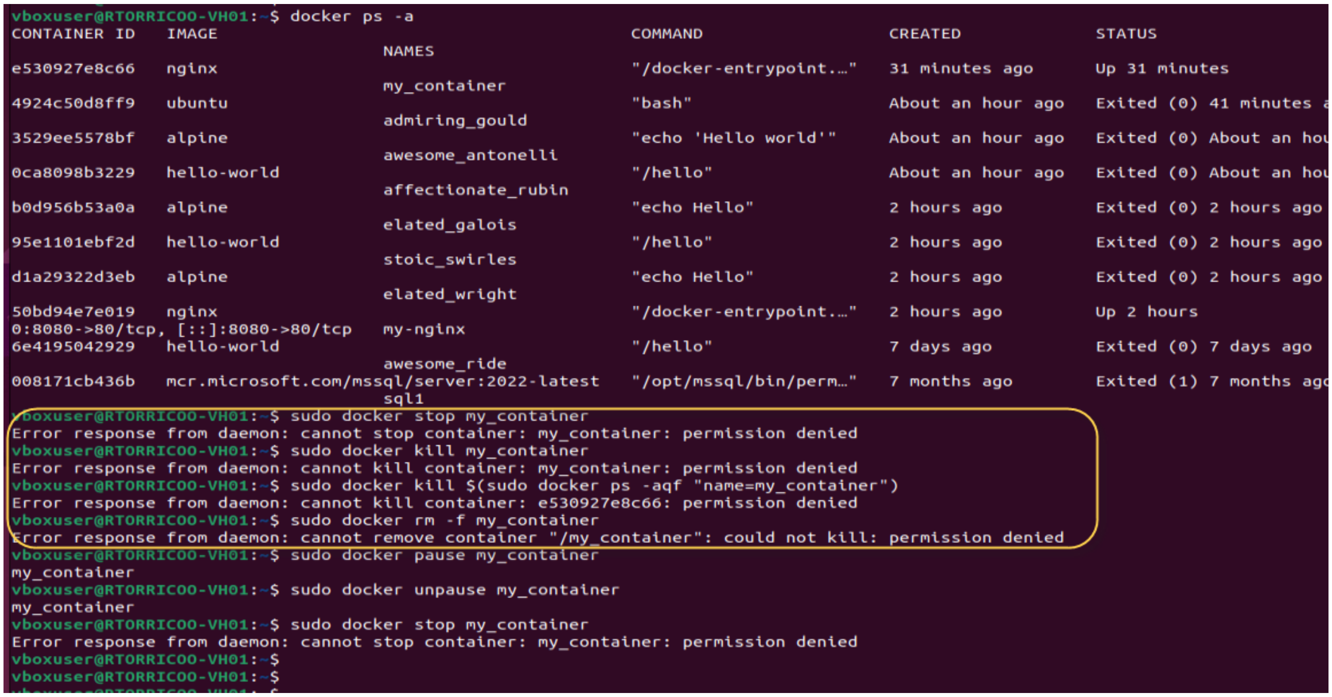
```
# Check status
docker ps
```



```
my_container
vboxuser@RTORRICOO-VH01:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS
                 NAMES
e530927e8c66   nginx     "/docker-entrypoint.…"   7 minutes ago   Up 7 minutes   80/tcp
                 my_container
50bd94e7e019   nginx     "/docker-entrypoint.…"   2 hours ago     Up 2 hours     0.0.0.0:8080->80/tcp,
]:8080->80/tcp   my-nginx
vboxuser@RTORRICOO-VH01:~$
```

4. Stop it

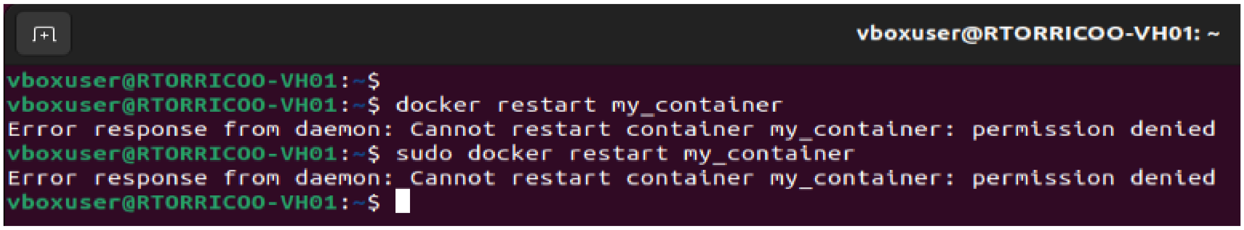
```
docker stop my_container

# Check status
docker ps
# NOTE: we are no table to stop/kill/rm due to permission denied
```



```
vboxuser@RTORRICOO-VH01:~$ docker ps -a
CONTAINER ID   IMAGE                                        COMMAND                    CREATED              STATUS
                                   NAMES
e530927e8c66   nginx                                        "/docker-entrypoint.…"     31 minutes ago       Up 31 minutes
                                   my_container
4924c50d8ff9   ubuntu                                       "bash"                     About an hour ago    Exited (0) 41 minutes a
                                   admiring_gould
3529ee5578bf   alpine                                       "echo 'Hello world'"       About an hour ago    Exited (0) About an hou
                                   awesome_antonelli
0ca8098b3229   hello-world                                  "/hello"                   About an hour ago    Exited (0) About an hou
                                   affectionate_rubin
b0d956b53a0a   alpine                                       "echo Hello"               2 hours ago          Exited (0) 2 hours ago
                                   elated_galois
95e1101ebf2d   hello-world                                  "/hello"                   2 hours ago          Exited (0) 2 hours ago
                                   stoic_swirles
d1a29322d3eb   alpine                                       "echo Hello"               2 hours ago          Exited (0) 2 hours ago
                                   elated_wright
50bd94e7e019   nginx                                        "/docker-entrypoint.…"     2 hours ago          Up 2 hours
0:8080->80/tcp, [::]:8080->80/tcp  my-nginx
6e4195042929   hello-world                                  "/hello"                   7 days ago           Exited (0) 7 days ago
                                   awesome_ride
008171cb436b   mcr.microsoft.com/mssql/server:2022-latest   "/opt/mssql/bin/perm…"     7 months ago         Exited (1) 7 months ago
                                   sql1
vboxuser@RTORRICOO-VH01:~$ sudo docker stop my_container
Error response from daemon: cannot stop container: my_container: permission denied
vboxuser@RTORRICOO-VH01:~$ sudo docker kill my_container
Error response from daemon: cannot kill container: my_container: permission denied
vboxuser@RTORRICOO-VH01:~$ sudo docker kill $(sudo docker ps -aqf "name=my_container")
Error response from daemon: cannot kill container: e530927e8c66: permission denied
vboxuser@RTORRICOO-VH01:~$ sudo docker rm -f my_container
Error response from daemon: cannot remove container "/my_container": could not kill: permission denied
vboxuser@RTORRICOO-VH01:~$ sudo docker pause my_container
my_container
vboxuser@RTORRICOO-VH01:~$ sudo docker unpause my_container
my_container
vboxuser@RTORRICOO-VH01:~$ sudo docker stop my_container
Error response from daemon: cannot stop container: my_container: permission denied
vboxuser@RTORRICOO-VH01:~$
vboxuser@RTORRICOO-VH01:~$
```

5. Restart it

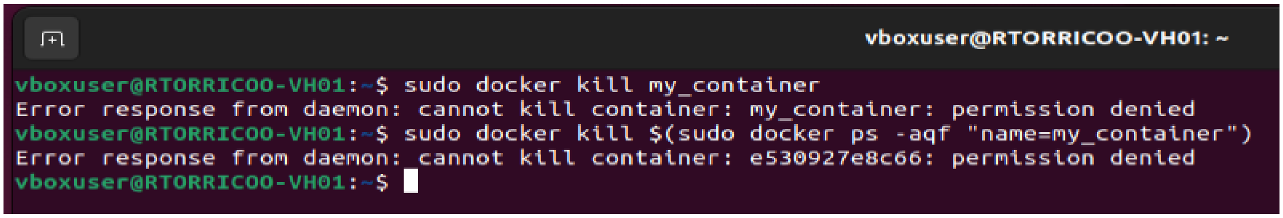
```
docker restart my_container

# Check status
docker ps
```



```
vboxuser@RTORRICOO-VH01:~$
vboxuser@RTORRICOO-VH01:~$ docker restart my_container
Error response from daemon: Cannot restart container my_container: permission denied
vboxuser@RTORRICOO-VH01:~$ sudo docker restart my_container
Error response from daemon: Cannot restart container my_container: permission denied
vboxuser@RTORRICOO-VH01:~$
```

6. Kill it

```
docker kill my_container
# Kill by ID
sudo docker kill $(sudo docker ps -aqf "name=my_container")

# Check status
docker ps
```

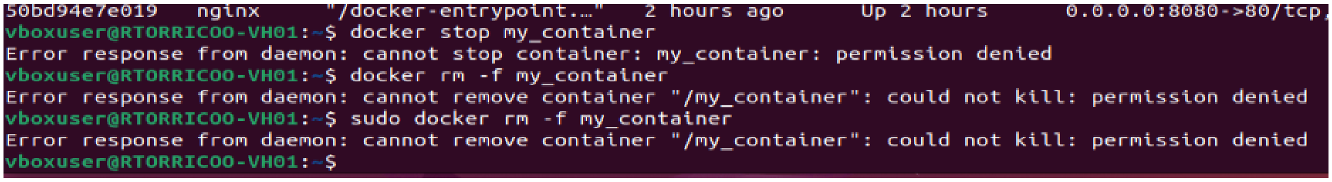


```
vboxuser@RTORRICOO-VH01: ~
vboxuser@RTORRICOO-VH01:~$ sudo docker kill my_container
Error response from daemon: cannot kill container: my_container: permission denied
vboxuser@RTORRICOO-VH01:~$ sudo docker kill $(sudo docker ps -aqf "name=my_container")
Error response from daemon: cannot kill container: e530927e8c66: permission denied
vboxuser@RTORRICOO-VH01:~$
```

Exercise 7

Remove a running container.

```
docker rm -f container_name_or_id
# Stop first, then remove
docker stop my_nginx && docker rm my_nginx

# Check status
docker ps
```

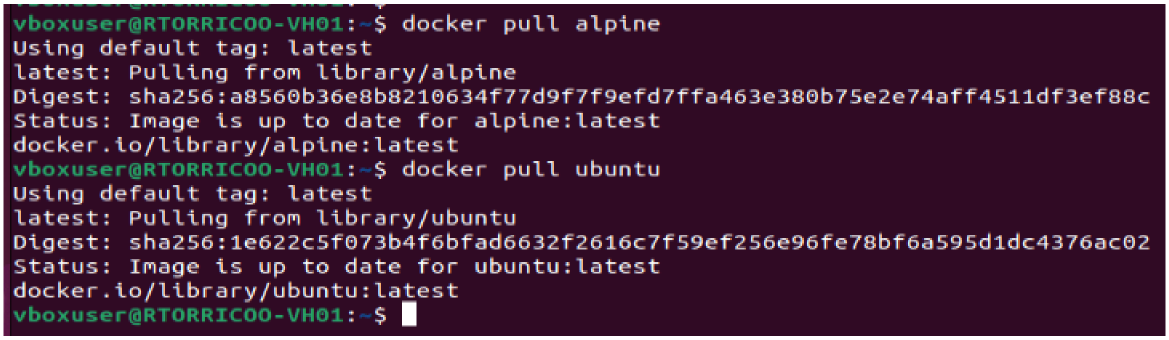


```
50bd94e7e019   nginx     "/docker-entrypoint.…"   2 hours ago      Up 2 hours      0.0.0.0:8080->80/tcp,
vboxuser@RTORRICOO-VH01:~$ docker stop my_container
Error response from daemon: cannot stop container: my_container: permission denied
vboxuser@RTORRICOO-VH01:~$ docker rm -f my_container
Error response from daemon: cannot remove container "/my_container": could not kill: permission denied
vboxuser@RTORRICOO-VH01:~$ sudo docker rm -f my_container
Error response from daemon: cannot remove container "/my_container": could not kill: permission denied
vboxuser@RTORRICOO-VH01:~$
```

Exercise 8

1. Pull the alpine and ubuntu images.

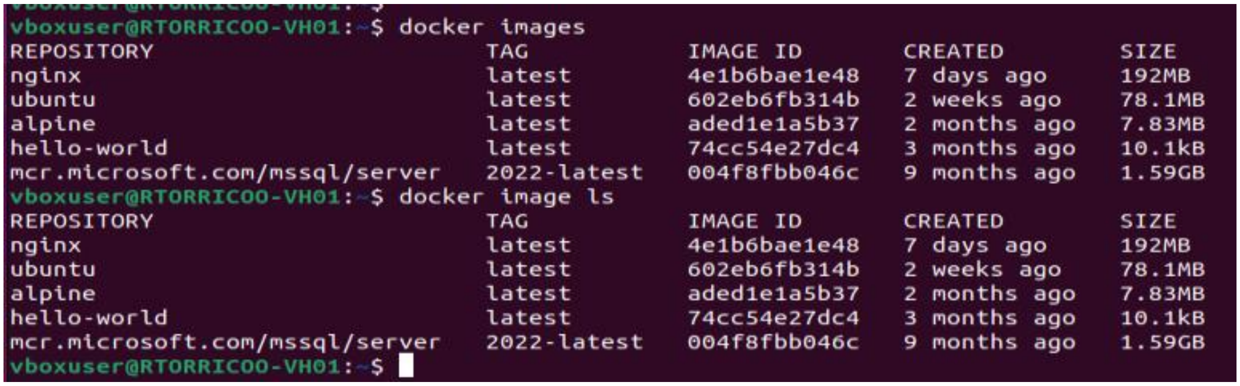
```
docker pull alpine
docker pull ubuntu
```



```
vboxuser@RTORRICOO-VH01:~$ docker pull alpine
Using default tag: latest
latest: Pulling from library/alpine
Digest: sha256:a8560b36e8b8210634f77d9f7f9efd7ffa463e380b75e2e74aff4511df3ef88c
Status: Image is up to date for alpine:latest
docker.io/library/alpine:latest
vboxuser@RTORRICOO-VH01:~$ docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
Digest: sha256:1e622c5f073b4f6bfad6632f2616c7f59ef256e96fe78bf6a595d1dc4376ac02
Status: Image is up to date for ubuntu:latest
docker.io/library/ubuntu:latest
vboxuser@RTORRICOO-VH01:~$
```

2. List all the container images in your Docker Host.

```
docker images
# For more detailed info
docker image ls
```



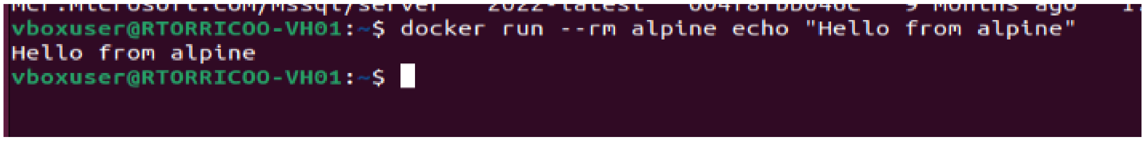
```
vboxuser@RTORRICOO-VH01:~$ docker images
REPOSITORY                       TAG           IMAGE ID       CREATED         SIZE
nginx                            latest        4e1b6bae1e48   7 days ago      192MB
ubuntu                           latest        602eb6fb314b   2 weeks ago     78.1MB
alpine                           latest        aded1e1a5b37   2 months ago    7.83MB
hello-world                      latest        74cc54e27dc4   3 months ago    10.1kB
mcr.microsoft.com/mssql/server   2022-latest   004f8fbb046c   9 months ago    1.59GB
vboxuser@RTORRICOO-VH01:~$ docker image ls
REPOSITORY                       TAG           IMAGE ID       CREATED         SIZE
nginx                            latest        4e1b6bae1e48   7 days ago      192MB
ubuntu                           latest        602eb6fb314b   2 weeks ago     78.1MB
alpine                           latest        aded1e1a5b37   2 months ago    7.83MB
hello-world                      latest        74cc54e27dc4   3 months ago    10.1kB
mcr.microsoft.com/mssql/server   2022-latest   004f8fbb046c   9 months ago    1.59GB
vboxuser@RTORRICOO-VH01:~$
```

Exercise 9

1. Run alpine and execute echo "hello from alpine"

docker run --rm alpine echo "hello from alpine"

- *--rm automatically removes the container after execution*

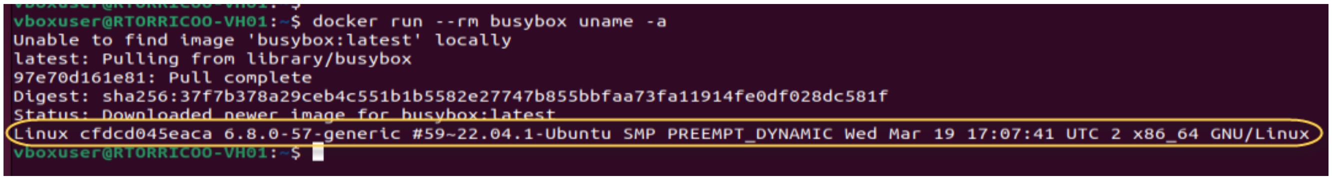


```
mcr.microsoft.com/mssql/server   2022-latest   004f8fbb046c   9 months ago   1.
vboxuser@RTORRICOO-VH01:~$ docker run --rm alpine echo "Hello from alpine"
Hello from alpine
vboxuser@RTORRICOO-VH01:~$
```

2. Run busybox and execute uname -a

# Output will show Linux kernel info

docker run --rm busybox uname -a

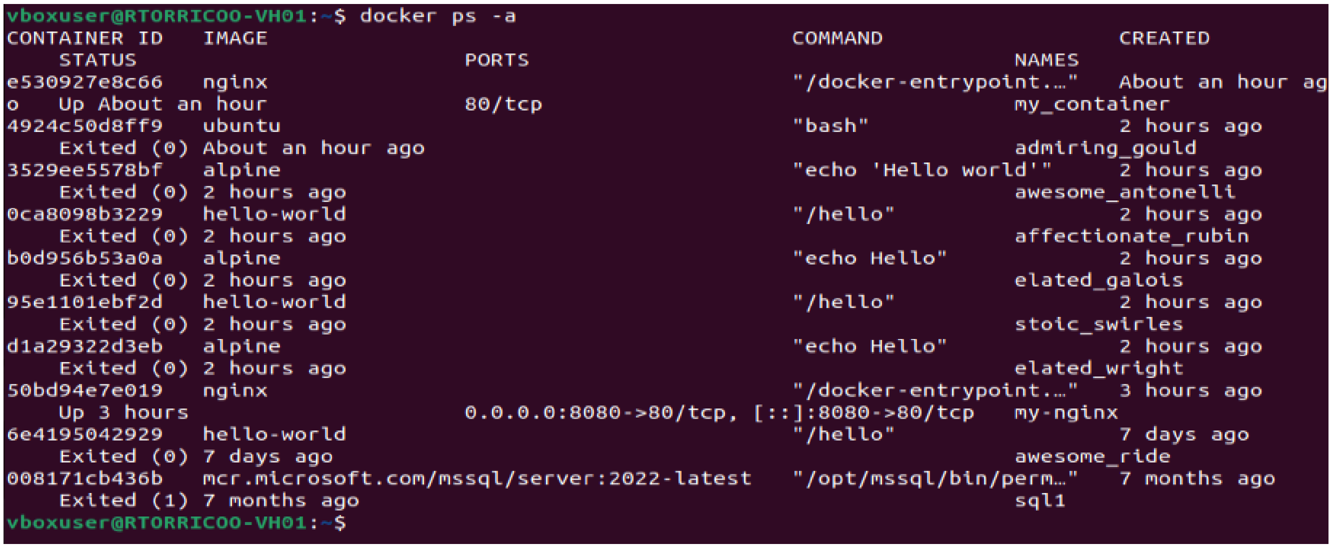


```
vboxuser@RTORRICOO-VH01:~$ docker run --rm busybox uname -a
Unable to find image 'busybox:latest' locally
latest: Pulling from library/busybox
97e70d161e81: Pull complete
Digest: sha256:37f7b378a29ceb4c551b1b5582e27747b855bbfaa73fa11914fe0df028dc581f
Status: Downloaded newer image for busybox:latest
Linux cfdcd045eaca 6.8.0-57-generic #59~22.04.1-Ubuntu SMP PREEMPT_DYNAMIC Wed Mar 19 17:07:41 UTC 2 x86_64 GNU/Linux
vboxuser@RTORRICOO-VH01:~$
```

3. List all the container.

# Including exited

docker ps -a

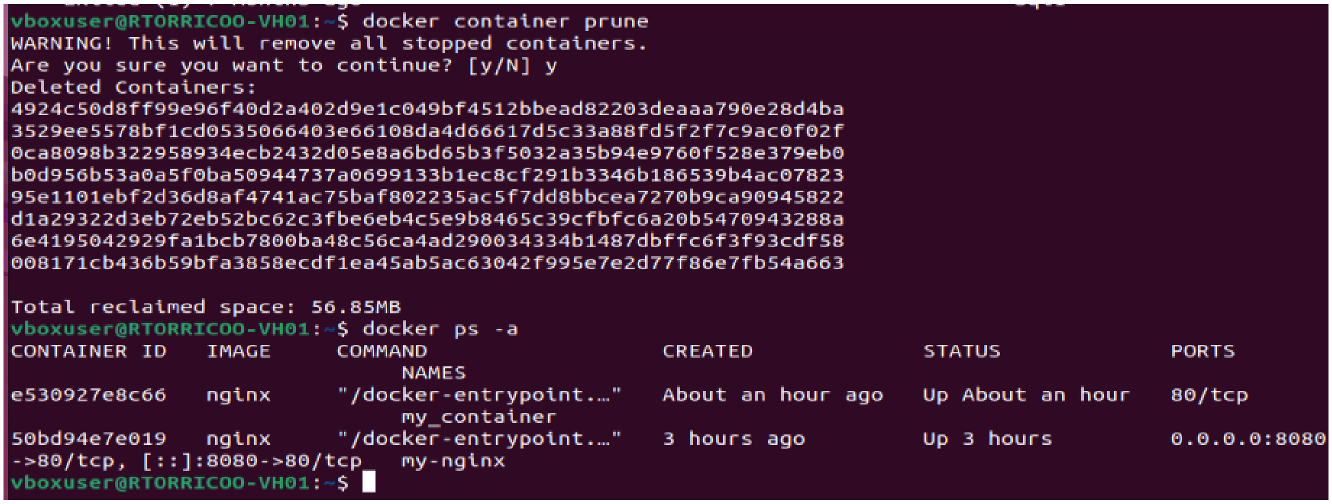


```
vboxuser@RTORRICOO-VH01:~$ docker ps -a
CONTAINER ID   IMAGE                                         COMMAND                  CREATED
    STATUS                    PORTS                                     NAMES
e530927e8c66   nginx                                         "/docker-entrypoint.…"   About an hour ag
o    Up About an hour         80/tcp                                    my_container
4924c50d8ff9   ubuntu                                        "bash"                   2 hours ago
    Exited (0) About an hour ago                                        admiring_gould
3529ee5578bf   alpine                                        "echo 'Hello world'"     2 hours ago
    Exited (0) 2 hours ago                                              awesome_antonelli
0ca8098b3229   hello-world                                   "/hello"                 2 hours ago
    Exited (0) 2 hours ago                                              affectionate_rubin
b0d956b53a0a   alpine                                        "echo Hello"             2 hours ago
    Exited (0) 2 hours ago                                              elated_galois
95e1101ebf2d   hello-world                                   "/hello"                 2 hours ago
    Exited (0) 2 hours ago                                              stoic_swirles
d1a29322d3eb   alpine                                        "echo Hello"             2 hours ago
    Exited (0) 2 hours ago                                              elated_wright
50bd94e7e019   nginx                                         "/docker-entrypoint.…"   3 hours ago
    Up 3 hours                0.0.0.0:8080->80/tcp, [::]:8080->80/tcp   my-nginx
6e4195042929   hello-world                                   "/hello"                 7 days ago
    Exited (0) 7 days ago                                               awesome_ride
008171cb436b   mcr.microsoft.com/mssql/server:2022-latest   "/opt/mssql/bin/perm…"   7 months ago
    Exited (1) 7 months ago                                             sql1
vboxuser@RTORRICOO-VH01:~$
```

Exercise 10

1. Remove all stopped containers.

```
docker container prune

# Check containers and status
docker ps
```

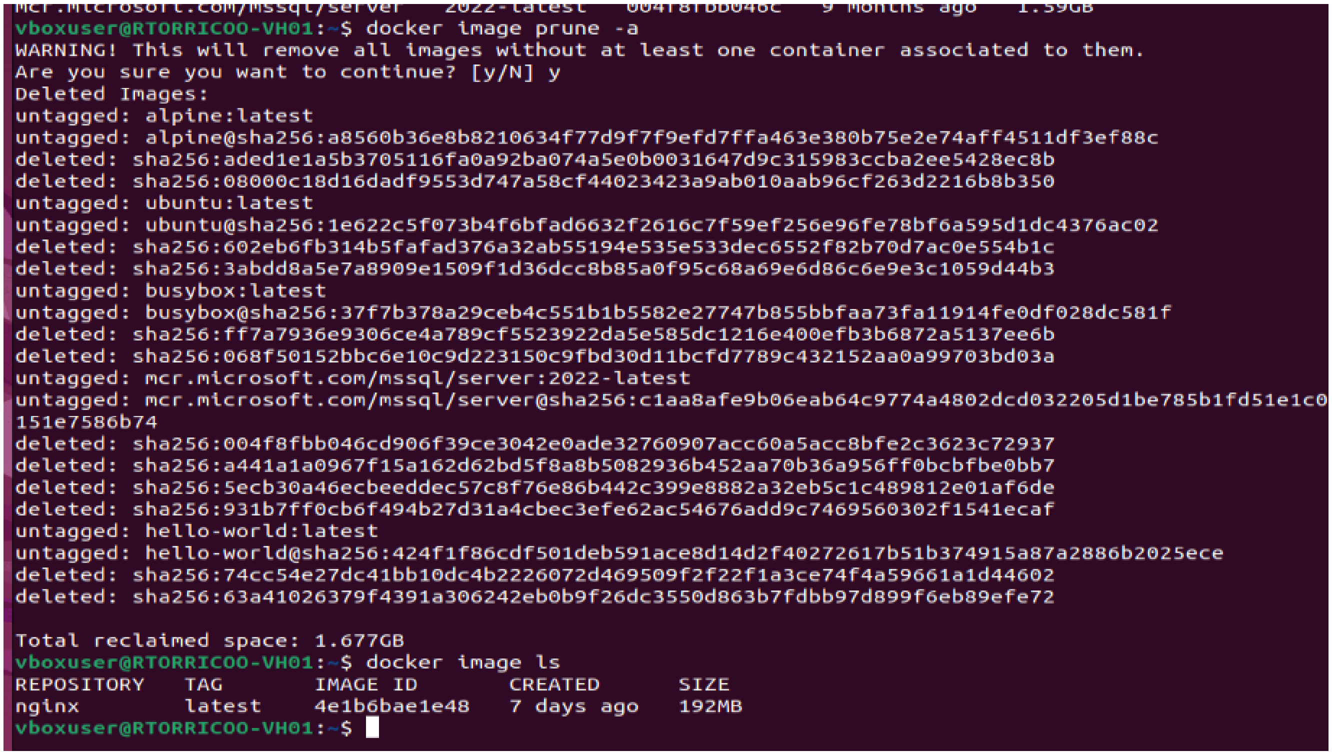


```
vboxuser@RTORRICOO-VH01:~$ docker container prune
WARNING! This will remove all stopped containers.
Are you sure you want to continue? [y/N] y
Deleted Containers:
4924c50d8ff99e96f40d2a402d9e1c049bf4512bbead82203deaaa790e28d4ba
3529ee5578bf1cd0535066403e66108da4d66617d5c33a88fd5f2f7c9ac0f02f
0ca8098b322958934ecb2432d05e8a6bd65b3f5032a35b94e9760f528e379eb0
b0d956b53a0a5f0ba50944737a0699133b1ec8cf291b3346b186539b4ac07823
95e1101ebf2d36d8af4741ac75baf802235ac5f7dd8bbcea7270b9ca90945822
d1a29322d3eb72eb52bc62c3fbe6eb4c5e9b8465c39cfbfc6a20b5470943288a
6e4195042929fa1bcb7800ba48c56ca4ad290034334b1487dbffc6f3f93cdf58
008171cb436b59bfa3858ecdf1ea45ab5ac63042f995e7e2d77f86e7fb54a663

Total reclaimed space: 56.85MB
vboxuser@RTORRICOO-VH01:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED             STATUS             PORTS
          NAMES
e530927e8c66   nginx     "/docker-entrypoint.…"   About an hour ago   Up About an hour   80/tcp
          my_container
50bd94e7e019   nginx     "/docker-entrypoint.…"   3 hours ago         Up 3 hours         0.0.0.0:8080
->80/tcp, [::]:8080->80/tcp   my-nginx
vboxuser@RTORRICOO-VH01:~$ 
```

2. Remove unused images.

```
docker image prune -a
```

- *-a removes all dangling AND unused images (not just dangling ones)*

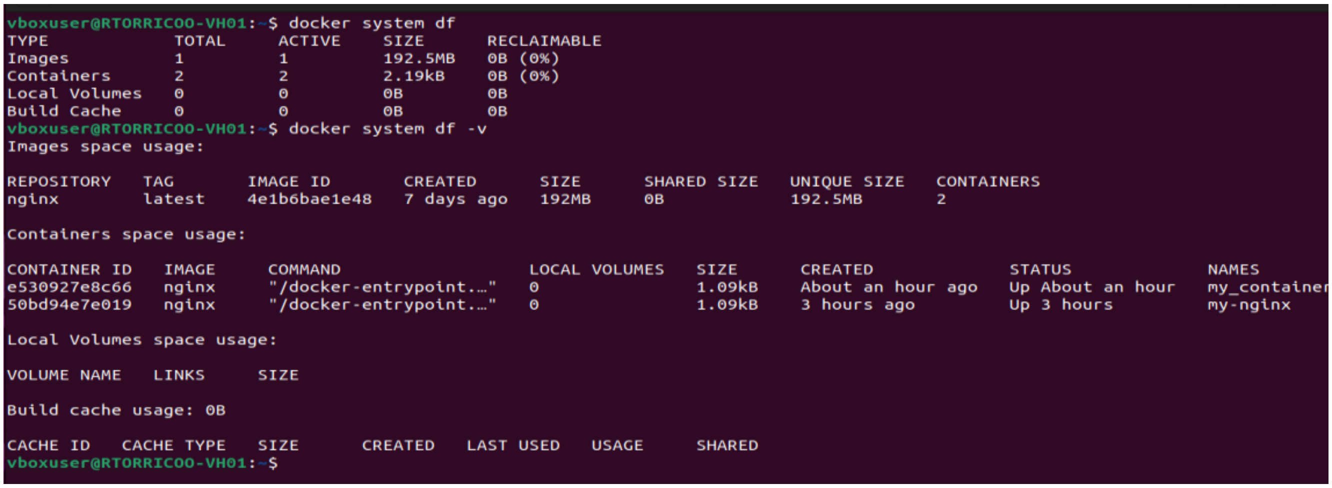


```
mcr.microsoft.com/mssql/server   2022-latest   004f8fbb046c   9 months ago   1.59GB
vboxuser@RTORRICOO-VH01:~$ docker image prune -a
WARNING! This will remove all images without at least one container associated to them.
Are you sure you want to continue? [y/N] y
Deleted Images:
untagged: alpine:latest
untagged: alpine@sha256:a8560b36e8b8210634f77d9f7f9efd7ffa463e380b75e2e74aff4511df3ef88c
deleted: sha256:aded1e1a5b3705116fa0a92ba074a5e0b0031647d9c315983ccba2ee5428ec8b
deleted: sha256:08000c18d16dadf9553d747a58cf44023423a9ab010aab96cf263d2216b8b350
untagged: ubuntu:latest
untagged: ubuntu@sha256:1e622c5f073b4f6bfad6632f2616c7f59ef256e96fe78bf6a595d1dc4376ac02
deleted: sha256:602eb6fb314b5fafad376a32ab55194e535e533dec6552f82b70d7ac0e554b1c
deleted: sha256:3abdd8a5e7a8909e1509f1d36dcc8b85a0f95c68a69e6d86c6e9e3c1059d44b3
untagged: busybox:latest
untagged: busybox@sha256:37f7b378a29ceb4c551b1b5582e27747b855bbfaa73fa11914fe0df028dc581f
deleted: sha256:ff7a7936e9306ce4a789cf5523922da5e585dc1216e400efb3b6872a5137ee6b
deleted: sha256:068f50152bbc6e10c9d223150c9fbd30d11bcfd7789c432152aa0a99703bd03a
untagged: mcr.microsoft.com/mssql/server:2022-latest
untagged: mcr.microsoft.com/mssql/server@sha256:c1aa8afe9b06eab64c9774a4802dcd032205d1be785b1fd51e1c0
151e7586b74
deleted: sha256:004f8fbb046cd906f39ce3042e0ade32760907acc60a5acc8bfe2c3623c72937
deleted: sha256:a441a1a0967f15a162d62bd5f8a8b5082936b452aa70b36a956ff0bcbfbe0bb7
deleted: sha256:5ecb30a46ecbeeddec57c8f76e86b442c399e8882a32eb5c1c489812e01af6de
deleted: sha256:931b7ff0cb6f494b27d31a4cbec3efe62ac54676add9c7469560302f1541ecaf
untagged: hello-world:latest
untagged: hello-world@sha256:424f1f86cdf501deb591ace8d14d2f40272617b51b374915a87a2886b2025ece
deleted: sha256:74cc54e27dc41bb10dc4b2226072d469509f2f22f1a3ce74f4a59661a1d44602
deleted: sha256:63a41026379f4391a306242eb0b9f26dc3550d863b7fdbb97d899f6eb89efe72

Total reclaimed space: 1.677GB
vboxuser@RTORRICOO-VH01:~$ docker image ls
REPOSITORY   TAG       IMAGE ID       CREATED      SIZE
nginx        latest    4e1b6bae1e48   7 days ago   192MB
vboxuser@RTORRICOO-VH01:~$ 
```

3. Inspect Docker disk usage.

```
docker system df
```

```
# More Detailed Inspection
docker system df -v
```



```
vboxuser@RTORRICOO-VH01:~$ docker system df
TYPE            TOTAL     ACTIVE    SIZE      RECLAIMABLE
Images          1         1         192.5MB   0B (0%)
Containers      2         2         2.19kB    0B (0%)
Local Volumes   0         0         0B        0B
Build Cache     0         0         0B        0B
vboxuser@RTORRICOO-VH01:~$ docker system df -v
Images space usage:

REPOSITORY   TAG       IMAGE ID       CREATED      SIZE      SHARED SIZE   UNIQUE SIZE   CONTAINERS
nginx        latest    4e1b6bae1e48   7 days ago   192MB     0B            192.5MB       2

Containers space usage:

CONTAINER ID   IMAGE     COMMAND                  LOCAL VOLUMES   SIZE      CREATED             STATUS              NAMES
e530927e8c66   nginx     "/docker-entrypoint.…"   0               1.09kB    About an hour ago   Up About an hour    my_container
50bd94e7e019   nginx     "/docker-entrypoint.…"   0               1.09kB    3 hours ago         Up 3 hours          my-nginx

Local Volumes space usage:

VOLUME NAME   LINKS     SIZE

Build cache usage: 0B

CACHE ID   CACHE TYPE   SIZE      CREATED   LAST USED   USAGE     SHARED
vboxuser@RTORRICOO-VH01:~$
```