Jason Hodge

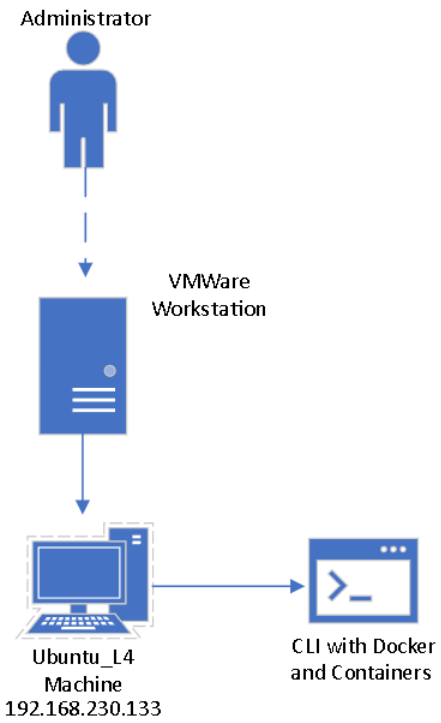
Lab 04 – Containers

April 12, 2024

Description:

The primary objective of this lab was to learn about and utilize containers within Docker. Containers are extremely relevant and have many benefits implementing them into company applications and processes. Some notable benefits include resource management, speed and efficiency, as well as being lightweight, portable, and secure. This lab goes through setting up a container utilizing the "linux_tweet_app" GitHub and then eventually creating a container to deploy a NGINX website. Then implementing a seccomp profile to our created container website.

Topology:



This is an overview of the virtual machines built in this lab.

Key Syntax:

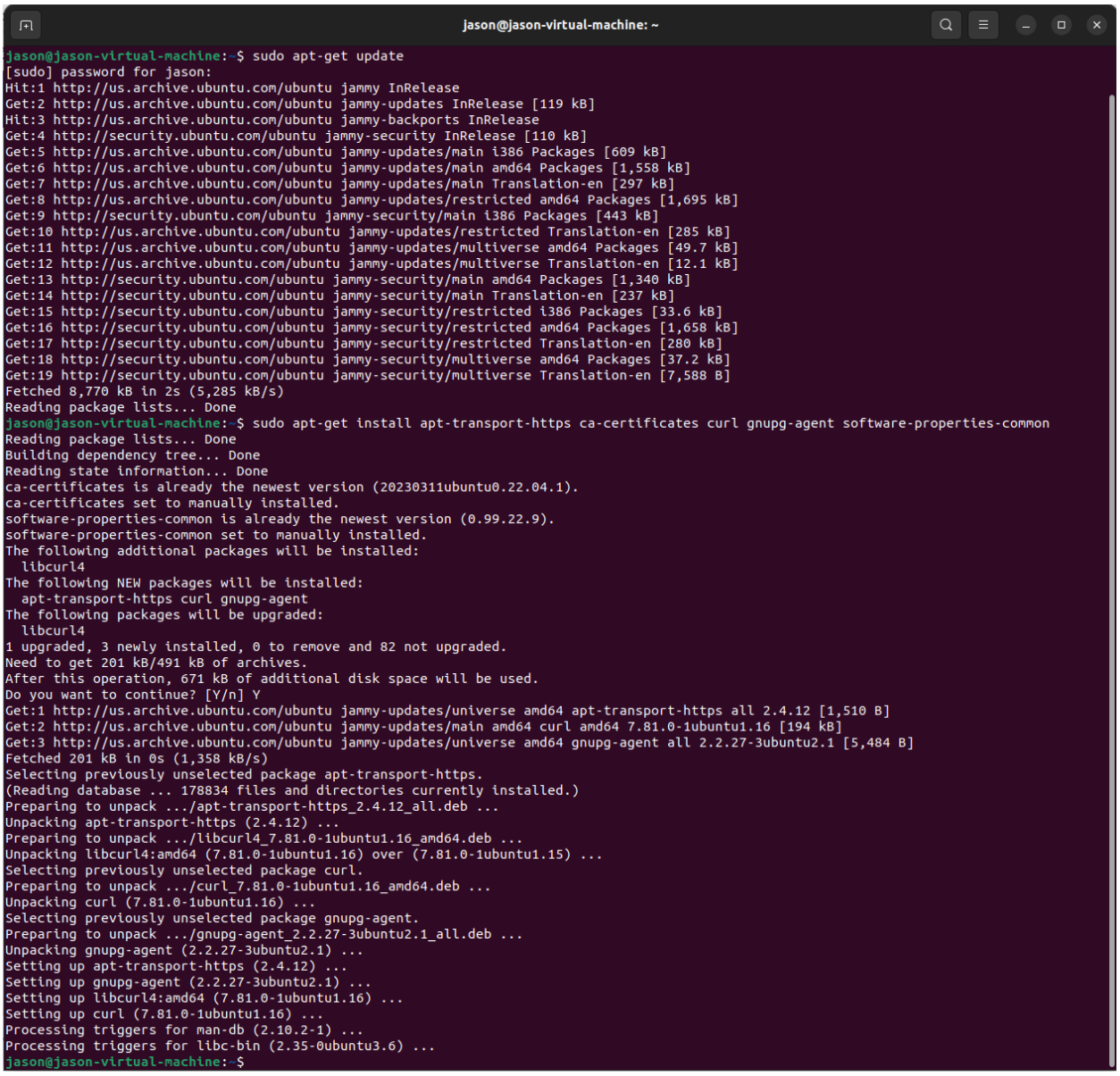
-d for Dameon

-p to expose ports

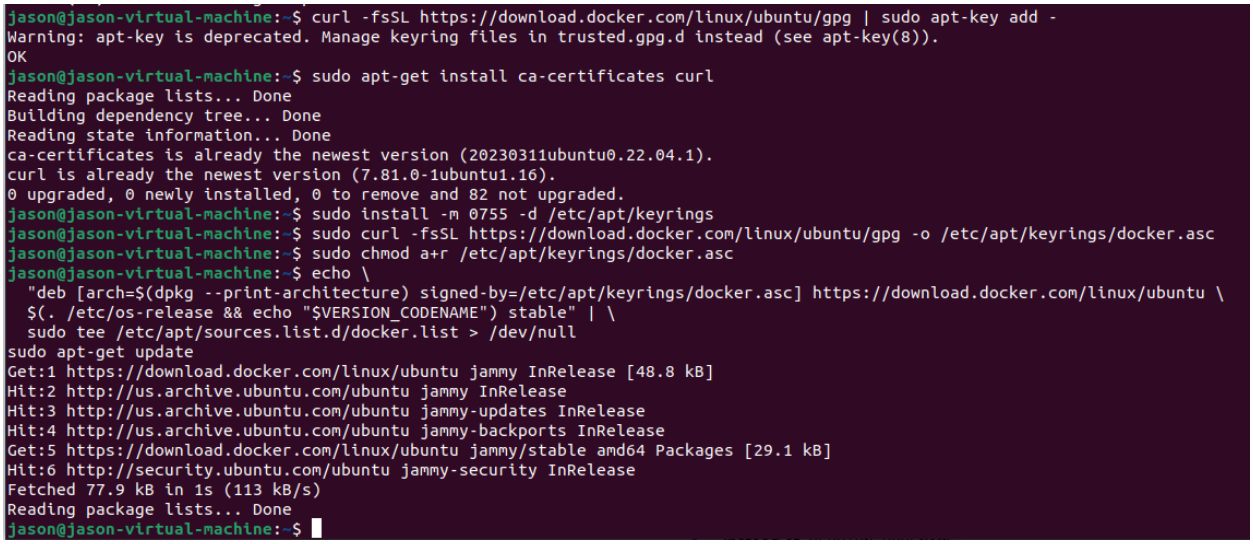
Other Commands are explained throughout as they are used.

Verification:

TASK ONE: Clone git repository to begin



First, I updated and installed the latest Ubuntu packet dependencies to alleviate any potential issues as well as to make sure all the proper packages are there.



Next, I installed Docker and added the proper packages and dependencies.

```
jason@jason-virtual-machine:~$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  docker-ce-rootless-extras git git-man liberror-perl libslirp0 pigz slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb git-cvs
  git-mediawiki git-svn
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-cli docker-ce-rootless-extras docker-compose-plugin git git-man liberror-perl
  libslirp0 pigz slirp4netns
0 upgraded, 12 newly installed, 0 to remove and 82 not upgraded.
Need to get 124 MB of archives.
After this operation, 449 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us.archive.ubuntu.com/ubuntu jammy/universe amd64 pigz amd64 2.6-1 [63.6 kB]
Get:2 https://download.docker.com/linux/ubuntu jammy/stable amd64 containerd.io amd64 1.6.28-2 [29.7 MB]
Get:3 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 liberror-perl all 0.17029-1 [26.5 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 git-man all 1:2.34.1-1ubuntu1.10 [954 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 git amd64 1:2.34.1-1ubuntu1.10 [3,166 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 libslirp0 amd64 4.6.1-1build1 [61.5 kB]
Get:7 http://us.archive.ubuntu.com/ubuntu jammy/universe amd64 slirp4netns amd64 1.0.1-2 [28.2 kB]
Get:8 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-buildx-plugin amd64 0.13.1-1~ubuntu.22.04~jammy [29.5 MB]
Get:9 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-ce-cli amd64 5:26.0.0-1~ubuntu.22.04~jammy [13.8 MB]
Get:10 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-ce amd64 5:26.0.0-1~ubuntu.22.04~jammy [25.1 MB]
Get:11 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-ce-rootless-extras amd64 5:26.0.0-1~ubuntu.22.04~jammy [9,320 kB]
Get:12 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-compose-plugin amd64 2.25.0-1~ubuntu.22.04~jammy [12.1 MB]
Fetched 124 MB in 11s (11.5 MB/s)
Selecting previously unselected package pigz.
(Reading database ... 178849 files and directories currently installed.)
Preparing to unpack .../00-pigz_2.6-1_amd64.deb ...
Unpacking pigz (2.6-1) ...
Selecting previously unselected package containerd.io.
Preparing to unpack .../01-containerd.io_1.6.28-2_amd64.deb ...
Unpacking containerd.io (1.6.28-2) ...
Selecting previously unselected package docker-buildx-plugin.
Preparing to unpack .../02-docker-buildx-plugin_0.13.1-1~ubuntu.22.04~jammy_amd64.deb ...
Unpacking docker-buildx-plugin (0.13.1-1~ubuntu.22.04~jammy) ...
Selecting previously unselected package docker-ce-cli.
Preparing to unpack .../03-docker-ce-cli_5%3a26.0.0-1~ubuntu.22.04~jammy_amd64.deb ...
Unpacking docker-ce-cli (5:26.0.0-1~ubuntu.22.04~jammy) ...
Selecting previously unselected package docker-ce.
Preparing to unpack .../04-docker-ce_5%3a26.0.0-1~ubuntu.22.04~jammy_amd64.deb ...
Unpacking docker-ce (5:26.0.0-1~ubuntu.22.04~jammy) ...
Selecting previously unselected package docker-ce-rootless-extras.
Preparing to unpack .../05-docker-ce-rootless-extras_5%3a26.0.0-1~ubuntu.22.04~jammy_amd64.deb ...
Unpacking docker-ce-rootless-extras (5:26.0.0-1~ubuntu.22.04~jammy) ...
Selecting previously unselected package docker-compose-plugin.
Preparing to unpack .../06-docker-compose-plugin_2.25.0-1~ubuntu.22.04~jammy_amd64.deb ...
Unpacking docker-compose-plugin (2.25.0-1~ubuntu.22.04~jammy) ...
Selecting previously unselected package liberror-perl.
Preparing to unpack .../07-liberror-perl_0.17029-1_all.deb ...
Unpacking liberror-perl (0.17029-1) ...
Selecting previously unselected package git-man.
Preparing to unpack .../08-git-man_1%3a2.34.1-1ubuntu1.10_all.deb ...
Unpacking git-man (1:2.34.1-1ubuntu1.10) ...
Selecting previously unselected package git.
Preparing to unpack .../09-git_1%3a2.34.1-1ubuntu1.10_amd64.deb ...
Unpacking git (1:2.34.1-1ubuntu1.10) ...
Selecting previously unselected package libslirp0:amd64.
```

Here I installed the docker container plugins.

```
Selecting previously unselected package docker-compose-plugin.
Preparing to unpack .../06-docker-compose-plugin_2.25.0-1~ubuntu.22.04~jammy_amd64.deb ...
Unpacking docker-compose-plugin (2.25.0-1~ubuntu.22.04~jammy) ...
Selecting previously unselected package liberror-perl.
Preparing to unpack .../07-liberror-perl_0.17029-1_all.deb ...
Unpacking liberror-perl (0.17029-1) ...
Selecting previously unselected package git-man.
Preparing to unpack .../08-git-man_1%3a2.34.1-1ubuntu1.10_all.deb ...
Unpacking git-man (1:2.34.1-1ubuntu1.10) ...
Selecting previously unselected package git.
Preparing to unpack .../09-git_1%3a2.34.1-1ubuntu1.10_amd64.deb ...
Unpacking git (1:2.34.1-1ubuntu1.10) ...
Selecting previously unselected package libslirp0:amd64.
Preparing to unpack .../10-libslirp0_4.6.1-1build1_amd64.deb ...
Unpacking libslirp0:amd64 (4.6.1-1build1) ...
Selecting previously unselected package slirp4netns.
Preparing to unpack .../11-slirp4netns_1.0.1-2_amd64.deb ...
Unpacking slirp4netns (1.0.1-2) ...
Setting up liberror-perl (0.17029-1) ...
Setting up docker-buildx-plugin (0.13.1-1~ubuntu.22.04~jammy) ...
Setting up containerd.io (1.6.28-2) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /lib/systemd/system/containerd.service.
Setting up docker-compose-plugin (2.25.0-1~ubuntu.22.04~jammy) ...
Setting up docker-ce-cli (5:26.0.0-1~ubuntu.22.04~jammy) ...
Setting up libslirp0:amd64 (4.6.1-1build1) ...
Setting up pigz (2.6-1) ...
Setting up git-man (1:2.34.1-1ubuntu1.10) ...
Setting up docker-ce-rootless-extras (5:26.0.0-1~ubuntu.22.04~jammy) ...
Setting up slirp4netns (1.0.1-2) ...
Setting up docker-ce (5:26.0.0-1~ubuntu.22.04~jammy) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /lib/systemd/system/docker.socket.
Setting up git (1:2.34.1-1ubuntu1.10) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.6) ...
jason@jason-virtual-machine:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
c1ec31eb5944: Pull complete
Digest: sha256:53641cd209a4fecfc68e21a99871ce8c6920b2e7502df0a20671c6fccc73a7c6
Status: Downloaded newer image for hello-world:latest

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/

tions
jason@jason-virtual-machine:~$
```

Here I ran the hello-world container testing functionality.

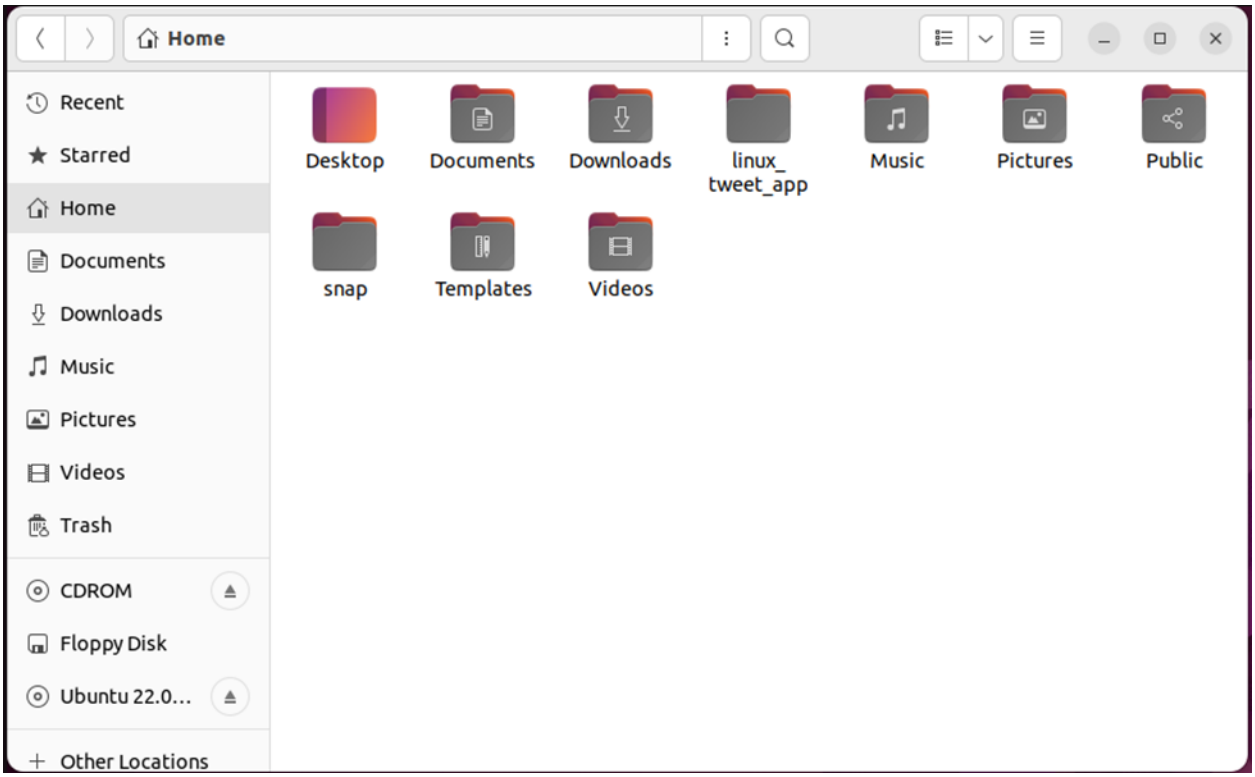
```
jason@jason-virtual-machine:~$ sudo apt-get update
Hit:1 http://us.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 https://download.docker.com/linux/ubuntu jammy InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:5 http://security.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
jason@jason-virtual-machine:~$ apt-cache madison docker-ce
 docker-ce | 5:26.0.0-1~ubuntu.22.04~jammy | https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
 docker-ce | 5:25.0.5-1~ubuntu.22.04~jammy | https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
 docker-ce | 5:25.0.4-1~ubuntu.22.04~jammy | https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
 docker-ce | 5:25.0.3-1~ubuntu.22.04~jammy | https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
 docker-ce | 5:25.0.2-1~ubuntu.22.04~jammy | https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
 docker-ce | 5:25.0.1-1~ubuntu.22.04~jammy | https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
 docker-ce | 5:25.0.0-1~ubuntu.22.04~jammy | https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
 docker-ce | 5:24.0.9-1~ubuntu.22.04~jammy | https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
 docker-ce | 5:24.0.8-1~ubuntu.22.04~jammy | https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
 docker-ce | 5:24.0.7-1~ubuntu.22.04~jammy | https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
 docker-ce | 5:24.0.6-1~ubuntu.22.04~jammy | https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
 docker-ce | 5:24.0.5-1~ubuntu.22.04~jammy | https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
 docker-ce | 5:24.0.4-1~ubuntu.22.04~jammy | https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
 docker-ce | 5:24.0.3-1~ubuntu.22.04~jammy | https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
 docker-ce | 5:24.0.2-1~ubuntu.22.04~jammy | https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
 docker-ce | 5:24.0.1-1~ubuntu.22.04~jammy | https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
 docker-ce | 5:24.0.0-1~ubuntu.22.04~jammy | https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
 docker-ce | 5:23.0.6-1~ubuntu.22.04~jammy | https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
 docker-ce | 5:23.0.5-1~ubuntu.22.04~jammy | https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
 docker-ce | 5:23.0.4-1~ubuntu.22.04~jammy | https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
 docker-ce | 5:23.0.3-1~ubuntu.22.04~jammy | https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
 docker-ce | 5:23.0.2-1~ubuntu.22.04~jammy | https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
 docker-ce | 5:23.0.1-1~ubuntu.22.04~jammy | https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
 docker-ce | 5:23.0.0-1~ubuntu.22.04~jammy | https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
 docker-ce | 5:20.10.24~3-0~ubuntu-jammy | https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
 docker-ce | 5:20.10.23~3-0~ubuntu-jammy | https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
 docker-ce | 5:20.10.22~3-0~ubuntu-jammy | https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
 docker-ce | 5:20.10.21~3-0~ubuntu-jammy | https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
 docker-ce | 5:20.10.20~3-0~ubuntu-jammy | https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
 docker-ce | 5:20.10.19~3-0~ubuntu-jammy | https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
 docker-ce | 5:20.10.18~3-0~ubuntu-jammy | https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
 docker-ce | 5:20.10.17~3-0~ubuntu-jammy | https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
 docker-ce | 5:20.10.16~3-0~ubuntu-jammy | https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
 docker-ce | 5:20.10.15~3-0~ubuntu-jammy | https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
 docker-ce | 5:20.10.14~3-0~ubuntu-jammy | https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
 docker-ce | 5:20.10.13~3-0~ubuntu-jammy | https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
```

```
jason@jason-virtual-machine:~$ sudo apt-get install docker-ce=5:26.0.0-1~ubuntu.22.04~jammy docker-ce-cli=5:26.0.0-1~ubuntu.22.04~jammy containerd.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
containerd.io is already the newest version (1.6.28-2).
docker-ce-cli is already the newest version (5:26.0.0-1~ubuntu.22.04~jammy).
docker-ce is already the newest version (5:26.0.0-1~ubuntu.22.04~jammy).
0 upgraded, 0 newly installed, 0 to remove and 82 not upgraded.
```

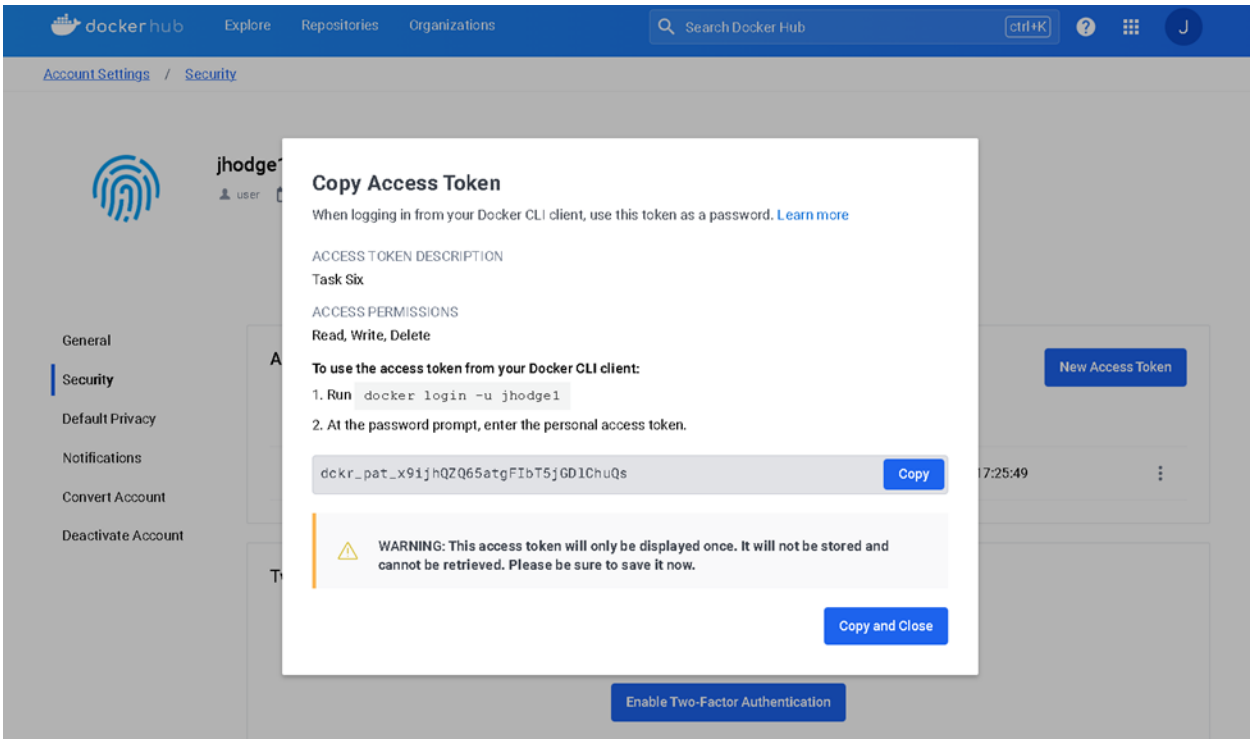
Here I ran an update again and installed an appropriate version of the Docker engine and dependencies.

```
jason@jason-virtual-machine:~$ sudo -i
[sudo] password for jason:
root@jason-virtual-machine:~# git clone https://github.com/dockersamples/linux_tweet_app
Cloning into 'linux_tweet_app'...
remote: Enumerating objects: 14, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 14 (delta 4), reused 4 (delta 4), pack-reused 6
Receiving objects: 100% (14/14), 10.79 KiB | 10.79 MiB/s, done.
Resolving deltas: 100% (5/5), done.
root@jason-virtual-machine:~#
```

Here I moved over to the root terminal and git cloned “linux_tweet_app” GitHub repository.



Here we can see the location of the linux_tweet_app within the file system. This could also be put inside another folder.



Here we can see I created an access token and login for Docker.

TASK TWO: Execute a Task as a Docker Container

```
root@jason-virtual-machine:~# docker container run alpine hostname
Unable to find image 'alpine:latest' locally
latest: Pulling from library/alpine
4abcf2066143: Pull complete
Digest: sha256:c5b1261d6d3e43071626931fc004f70149baeba2c8ec672bd4f27761f8e1ad6b
Status: Downloaded newer image for alpine:latest
e334605dbbd2
root@jason-virtual-machine:~# docker pull alpine
Using default tag: latest
latest: Pulling from library/alpine
Digest: sha256:c5b1261d6d3e43071626931fc004f70149baeba2c8ec672bd4f27761f8e1ad6b
Status: Image is up to date for alpine:latest
docker.io/library/alpine:latest
root@jason-virtual-machine:~# docker container ls --all
CONTAINER ID   IMAGE         COMMAND      CREATED             STATUS                         PORTS     NAMES
e334605dbbd2   alpine        "hostname"   About a minute ago   Exited (0) About a minute ago            priceless_chandrasekhar
36e7091caa45   hello-world   "/hello"     28 minutes ago       Exited (0) 28 minutes ago                exciting_borg
root@jason-virtual-machine:~#
```

In this task I pulled the Alpine Kernel and got the latest version as well as checked out the containers.

TASK THREE: Execute a Docker Container Interactively

```
jason@jason-virtual-machine:~$ sudo -i
root@jason-virtual-machine:~# docker container run --interactive --tty --rm ubuntu bash
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
bccd10f490ab: Pull complete
Digest: sha256:77906da86b60585ce12215807090eb327e7386c8fafb5402369e421f44eff17e
Status: Downloaded newer image for ubuntu:latest
root@3eaac4cb3578:/# cat /etc/issue
Ubuntu 22.04.4 LTS \n \l

root@3eaac4cb3578:/#
```

Here I am still the root user. First, I ran docker interactively and then displayed the Linux distribution.

```
jason@jason-virtual-machine:~$ sudo -i
root@jason-virtual-machine:~# docker container run --interactive --tty --rm ubuntu bash
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
bccd10f490ab: Pull complete
Digest: sha256:77906da86b60585ce12215807090eb327e7386c8fafb5402369e421f44eff17e
Status: Downloaded newer image for ubuntu:latest
root@3eaac4cb3578:/# cat /etc/issue
Ubuntu 22.04.4 LTS \n \l

root@3eaac4cb3578:/# exit
exit
root@jason-virtual-machine:~# exit
logout
jason@jason-virtual-machine:~$ sudo -i
root@jason-virtual-machine:~# docker container ls --all
CONTAINER ID   IMAGE         COMMAND      CREATED          STATUS                      PORTS     NAMES
e334605dbbd2   alpine        "hostname"   14 minutes ago   Exited (0) 14 minutes ago             priceless_chandrasekhar
36e7091caa45   hello-world   "/hello"     41 minutes ago   Exited (0) 41 minutes ago             exciting_borg
root@jason-virtual-machine:~#
```

Here I also exited that container and checked out the containers again. We removed the container with –rm which is why we do not see the process I just executed.

TASK FOUR: Execute a Background Docker Container

```
jason@jason-virtual-machine:~$ sudo -i
[sudo] password for jason:
root@jason-virtual-machine:~# docker container ls --all
CONTAINER ID   IMAGE         COMMAND      CREATED         STATUS                     PORTS     NAMES
e334605dbbd2   alpine        "hostname"   20 hours ago    Exited (0) 20 hours ago              priceless_chandrasekhar
36e7091caa45   hello-world   "/hello"     20 hours ago    Exited (0) 20 hours ago              exciting_borg
root@jason-virtual-machine:~# docker container run \
 --detach \
 --name msis603db \
 -e MYSQL_ROOT_PASSWORD=msis603pw \
 mysql:latest
Unable to find image 'mysql:latest' locally
latest: Pulling from library/mysql
2ba873cb070a: Pull complete
dd1a4da808dd: Pull complete
3292fb4adf41: Pull complete
[redacted]58cc: Pull complete
[redacted]4c05: Pull complete
6a34d702f281: Pull complete
de90f4481477: Pull complete
d575200ae375: Pull complete
aea400be5707: Pull complete
38c930606a4f: Pull complete
Digest: sha256:0f2e15fb8b47db2518b1428239ed3e3fe6a6693401b2cf19552063562cfc2fc4
Status: Downloaded newer image for mysql:latest
139a4f45aa5c6d53ab551b8421e48a1f40df6406f01ff543804602199e0f6e37
root@jason-virtual-machine:~#
```

Here I ran the container as a background process, giving it a name and a password as well as making sure the syntax is up to date.

```
root@jason-virtual-machine:~# docker container ls
CONTAINER ID   IMAGE          COMMAND                  CREATED              STATUS              PORTS                 NAMES
139a4f45aa5c   mysql:latest   "docker-entrypoint.s…"   About a minute ago   Up About a minute   3306/tcp, 33060/tcp   msis603db
root@jason-virtual-machine:~# docker container logs msis603db
2024-04-09 22:44:32+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.3.0-1.el8 started.
2024-04-09 22:44:33+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
2024-04-09 22:44:33+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.3.0-1.el8 started.
2024-04-09 22:44:33+00:00 [Note] [Entrypoint]: Initializing database files
2024-04-09T22:44:33.324568Z 0 [System] [MY-015017] [Server] MySQL Server Initialization - start.
2024-04-09T22:44:33.326076Z 0 [System] [MY-013169] [Server] /usr/sbin/mysqld (mysqld 8.3.0) initializing of server in progress as process 79
2024-04-09T22:44:33.331765Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
2024-04-09T22:44:34.025573Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
2024-04-09T22:44:35.101555Z 6 [Warning] [MY-010453] [Server] root@localhost is created with an empty password ! Please consider switching off the --initialize-insecure
option.
2024-04-09T22:44:36.891770Z 0 [System] [MY-015018] [Server] MySQL Server Initialization - end.
2024-04-09 22:44:36+00:00 [Note] [Entrypoint]: Database files initialized
2024-04-09 22:44:36+00:00 [Note] [Entrypoint]: Starting temporary server
2024-04-09T22:44:36.948132Z 0 [System] [MY-015015] [Server] MySQL Server - start.
2024-04-09T22:44:37.429426Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 8.3.0) starting as process 121
2024-04-09T22:44:37.456833Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
2024-04-09T22:44:38.000579Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
2024-04-09T22:44:38.372599Z 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is self signed.
2024-04-09T22:44:38.372716Z 0 [System] [MY-013602] [Server] Channel mysql_main configured to support TLS. Encrypted connections are now supported for this channel.
2024-04-09T22:44:38.374130Z 0 [Warning] [MY-011810] [Server] Insecure configuration for --pid-file: Location '/var/run/mysqld' in the path is accessible to all OS users
. Consider choosing a different directory.
2024-04-09T22:44:38.387729Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready for connections. Version: '8.3.0'  socket: '/var/run/mysqld/mysqld.sock'  port: 0  M
ySQL Community Server - GPL.
2024-04-09T22:44:38.387757Z 0 [System] [MY-011323] [Server] X Plugin ready for connections. Socket: /var/run/mysqld/mysqlx.sock
2024-04-09 22:44:38+00:00 [Note] [Entrypoint]: Temporary server started.
'/var/lib/mysql/mysql.sock' -> '/var/run/mysqld/mysqld.sock'
Warning: Unable to load '/usr/share/zoneinfo/iso3166.tab' as time zone. Skipping it.
Warning: Unable to load '/usr/share/zoneinfo/leap-seconds.list' as time zone. Skipping it.
Warning: Unable to load '/usr/share/zoneinfo/leapseconds' as time zone. Skipping it.
Warning: Unable to load '/usr/share/zoneinfo/tzdata.zi' as time zone. Skipping it.
Warning: Unable to load '/usr/share/zoneinfo/zone.tab' as time zone. Skipping it.
Warning: Unable to load '/usr/share/zoneinfo/zone1970.tab' as time zone. Skipping it.

2024-04-09 22:44:39+00:00 [Note] [Entrypoint]: Stopping temporary server
2024-04-09T22:44:39.972677Z 10 [System] [MY-013172] [Server] Received SHUTDOWN from user root. Shutting down mysqld (Version: 8.3.0).
2024-04-09T22:44:40.720770Z 0 [System] [MY-010910] [Server] /usr/sbin/mysqld: Shutdown complete (mysqld 8.3.0)  MySQL Community Server - GPL.
2024-04-09T22:44:40.720794Z 0 [System] [MY-015016] [Server] MySQL Server - end.
2024-04-09 22:44:40+00:00 [Note] [Entrypoint]: Temporary server stopped

2024-04-09 22:44:40+00:00 [Note] [Entrypoint]: MySQL init process done. Ready for start up.

2024-04-09T22:44:40.986281Z 0 [System] [MY-015015] [Server] MySQL Server - start.
2024-04-09T22:44:41.229179Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 8.3.0) starting as process 1
2024-04-09T22:44:41.234330Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
2024-04-09T22:44:41.345060Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
2024-04-09T22:44:41.478116Z 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is self signed.
2024-04-09T22:44:41.478150Z 0 [System] [MY-013602] [Server] Channel mysql_main configured to support TLS. Encrypted connections are now supported for this channel.
2024-04-09T22:44:41.479824Z 0 [Warning] [MY-011810] [Server] Insecure configuration for --pid-file: Location '/var/run/mysqld' in the path is accessible to all OS users
. Consider choosing a different directory.
2024-04-09T22:44:41.493045Z 0 [System] [MY-011323] [Server] X Plugin ready for connections. Bind-address: '::' port: 33060, socket: /var/run/mysqld/mysqlx.sock
2024-04-09T22:44:41.493078Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready for connections. Version: '8.3.0'  socket: '/var/run/mysqld/mysqld.sock'  port: 3306
  MySQL Community Server - GPL.
root@jason-virtual-machine:~#
```

Here I ran the CLI commands “docker container ls”, which lists the running containers, and “docker container logs msis603db”, which shows the logs for my specific container I made.

```
root@jason-virtual-machine:~# docker container logs
"docker container logs" requires exactly 1 argument.
See 'docker container logs --help'.
```

Here we can see no argument is given for what container we are using.

```
root@jason-virtual-machine:~# docker container top
"docker container top" requires at least 1 argument.
See 'docker container top --help'.
```

Here no argument is given for what container we want to view the running processes of.

TASK FIVE: Docker Container Network Reachability

```
root@jason-virtual-machine:~# docker container exec -it msis603db \
> mysql --user=root --password=$MYSQL_ROOT_PASSWORD --version
Error response from daemon: No such container: -it
root@jason-virtual-machine:~# docker exec -it msis603db \
 mysql --user=root --password=$MYSQL_ROOT_PASSWORD --version
mysql: [Warning] Using a password on the command line interface can be insecure.
mysql  Ver 8.3.0 for Linux on x86_64 (MySQL Community Server - GPL)
root@jason-virtual-machine:~# exit
logout
jason@jason-virtual-machine:~$
```

Here I displayed the container and executed a command to view the version of Linux being used by the container.

TASK SIX: Docker Container Images

```
root@jason-virtual-machine:~# docker images
REPOSITORY    TAG       IMAGE ID       CREATED         SIZE
mysql         latest    65f3f983cb08   2 weeks ago     632MB
ubuntu        latest    ca2b0f26964c   6 weeks ago     77.9MB
alpine        latest    05455a08881e   2 months ago    7.38MB
hello-world   latest    d2c94e258dcb   11 months ago   13.3kB
root@jason-virtual-machine:~#
```

Here I am displaying the docker image containers I have.

```
root@jason-virtual-machine:~# cd ~/linux_tweet_app
root@jason-virtual-machine:~/linux_tweet_app# cat Dockerfile
FROM nginx:latest

COPY index.html /usr/share/nginx/html
COPY linux.png /usr/share/nginx/html

EXPOSE 80 443

CMD ["nginx", "-g", "daemon off;"]
```

Here I opened the linux_tweet_app folder and displayed the contents of the docker file inside.

```
root@jason-virtual-machine:~/linux_tweet_app# docker build .
[+] Building 11.5s (9/9) FINISHED                                                                                  docker:default
 => [internal] load build definition from Dockerfile                                                                         0.0s
 => => transferring dockerfile: 186B                                                                                         0.0s
 => [internal] load metadata for docker.io/library/nginx:latest                                                              0.8s
 => [auth] library/nginx:pull token for registry-1.docker.io                                                                 0.0s
 => [internal] load .dockerignore                                                                                            0.0s
 => => transferring context: 2B                                                                                              0.0s
 => [internal] load build context                                                                                            0.0s
 => => transferring context: 24.04kB                                                                                         0.0s
 => [1/3] FROM docker.io/library/nginx:latest@sha256:b72dad1d013c5e4c4fb817f884aa163287bf147482562f12c56368ca1c2a3705       10.4s
 => => resolve docker.io/library/nginx:latest@sha256:b72dad1d013c5e4c4fb817f884aa163287bf147482562f12c56368ca1c2a3705        0.0s
 => => sha256:6fcdffcd79f0bd371fceec96d609d4cc46b805002a2ea68c74b9d9925dfe5ec2 41.39MB / 41.39MB                             4.4s
 => => sha256:fbf231d461b3db0da913cf2ab74e989cc3f79ba6d8c23d2bf2fdafd52a177f5a 629B / 629B                                   0.3s
 => => sha256:c613f16b664244b150d1c3644cbc387ec1fe8376377f9419992280eb4a82ff3b 7.00kB / 7.00kB                               0.0s
 => => sha256:13808c22b207b066ef43572e57e4fb8c6172e887dd9a918c089a174a19371b7a 29.13MB / 29.13MB                             6.5s
 => => sha256:b72dad1d013c5e4c4fb817f884aa163287bf147482562f12c56368ca1c2a3705 9.85kB / 9.85kB                               0.0s
 => => sha256:cd6440757675 1d9b9ba4924f758d3d39fe76a6e142c32169625b60934c95f057 2.29kB / 2.29kB                              0.0s
 => => sha256:c9590dd9c9881d041113ddd4f1deb5f056e23ecd5bf332b867d4f64a3f648bd2 957B / 957B                                   0.7s
 => => sha256:b4033143d8591983af6ede7fca9b1cffcbd3a47f7e149e9cbc5cd0c3047acbf2 394B / 394B                                   0.9s
 => => sha256:abaefc5fcbdecd3accead79308cbc3482a41531d58e0be41c410291dfcf2fd60 1.21kB / 1.21kB                               1.1s
 => => sha256:bcef83155b8b09f1fc20cba2072adf7c02d94e71564e8c68473e47ab4e303fbe 1.40kB / 1.40kB                               1.3s
 => => extracting sha256:13808c22b207b066ef43572e57e4fb8c6172e887dd9a918c089a174a19371b7a                                    1.8s
 => => extracting sha256:6fcdffcd79f0bd371fceec96d609d4cc46b805002a2ea68c74b9d9925dfe5ec2                                    1.7s
 => => extracting sha256:fbf231d461b3db0da913cf2ab74e989cc3f79ba6d8c23d2bf2fdafd52a177f5a                                    0.0s
 => => extracting sha256:c9590dd9c9881d041113ddd4f1deb5f056e23ecd5bf332b867d4f64a3f648bd2                                    0.0s
 => => extracting sha256:b4033143d8591983af6ede7fca9b1cffcbd3a47f7e149e9cbc5cd0c3047acbf2                                    0.0s
 => => extracting sha256:abaefc5fcbdecd3accead79308cbc3482a41531d58e0be41c410291dfcf2fd60                                    0.0s
 => => extracting sha256:bcef83155b8b09f1fc20cba2072adf7c02d94e71564e8c68473e47ab4e303fbe                                    0.0s
 => [2/3] COPY index.html /usr/share/nginx/html                                                                              0.3s
 => [3/3] COPY linux.png /usr/share/nginx/html                                                                               0.0s
 => exporting to image                                                                                                       0.0s
 => => exporting layers                                                                                                      0.0s
 => => writing image sha256:3441d302b5657316199e0f554be2d7ee1ef21e4dbd3059f68ab12bb4d580f674                                 0.0s
root@jason-virtual-machine:~/linux_tweet_app#
```

Here I built a docker container image.

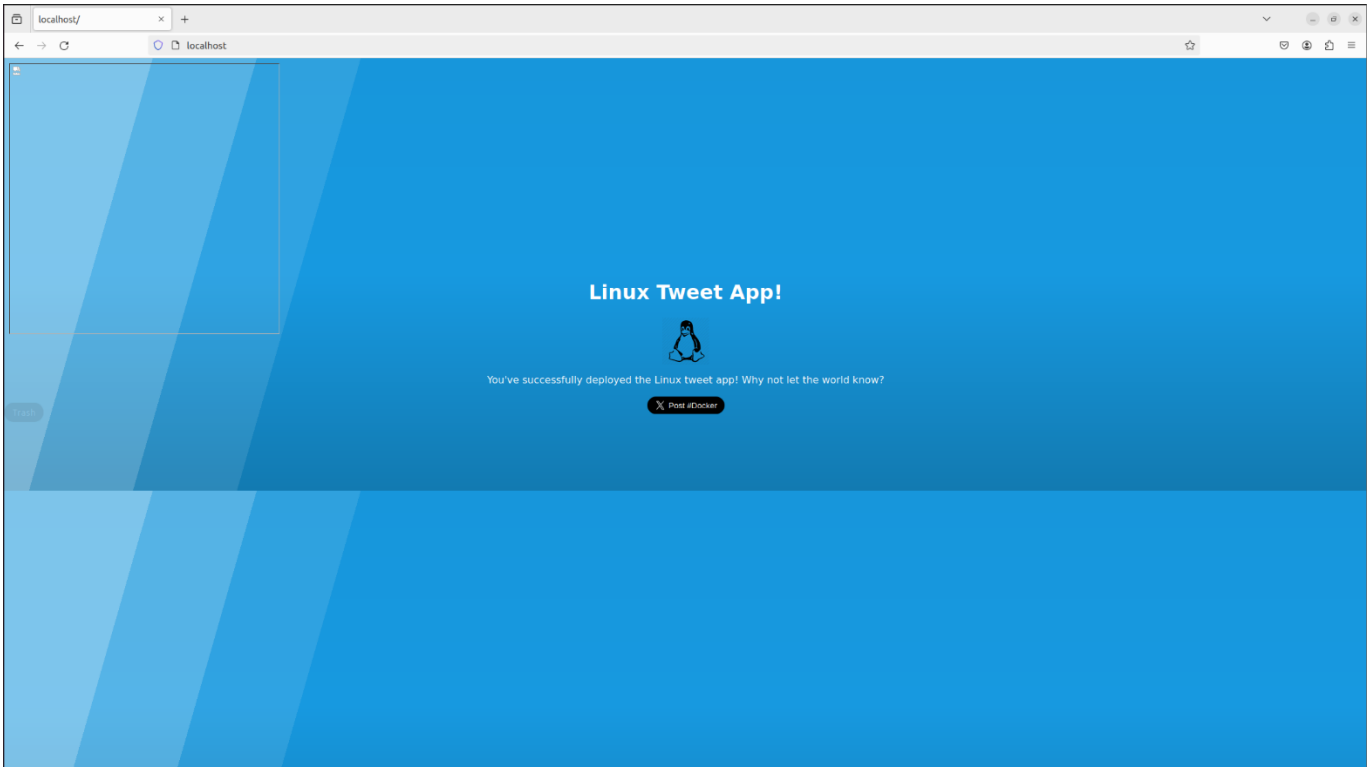
```
root@jason-virtual-machine:~/linux_tweet_app# docker images
REPOSITORY     TAG       IMAGE ID       CREATED          SIZE
<none>         <none>    3441d302b565   2 minutes ago    187MB
mysql          latest    65f3f983cb08   2 weeks ago      632MB
ubuntu         latest    ca2b0f26964c   6 weeks ago      77.9MB
alpine         latest    05455a08881e   2 months ago     7.38MB
hello-world    latest    d2c94e258dcb   11 months ago    13.3kB
root@jason-virtual-machine:~/linux_tweet_app#
```

Here we can see the new unnamed docker image.

```
root@jason-virtual-machine:~/linux_tweet_app# docker build -t linux_tweet_app .
[+] Building 0.4s (9/9) FINISHED                                                                                   docker:default
 => [internal] load build definition from Dockerfile                                                                         0.0s
 => => transferring dockerfile: 186B                                                                                         0.0s
 => [internal] load metadata for docker.io/library/nginx:latest                                                              0.4s
 => [auth] library/nginx:pull token for registry-1.docker.io                                                                 0.0s
 => [internal] load .dockerignore                                                                                            0.0s
 => => transferring context: 2B                                                                                              0.0s
 => [1/3] FROM docker.io/library/nginx:latest@sha256:b72dad1d013c5e4c4fb817f884aa163287bf147482562f12c56368ca1c2a3705        0.0s
 => [internal] load build context                                                                                            0.0s
 => => transferring context: 62B                                                                                             0.0s
 => CACHED [2/3] COPY index.html /usr/share/nginx/html                                                                       0.0s
 => CACHED [3/3] COPY linux.png /usr/share/nginx/html                                                                        0.0s
 => exporting to image                                                                                                       0.0s
 => => exporting layers                                                                                                      0.0s
 => => writing image sha256:3441d302b5657316199e0f554be2d7ee1ef21e4dbd3059f68ab12bb4d580f674                                 0.0s
 => => naming to docker.io/library/linux_tweet_app                                                                           0.0s
root@jason-virtual-machine:~/linux_tweet_app# docker images
REPOSITORY        TAG       IMAGE ID       CREATED          SIZE
linux_tweet_app   latest    3441d302b565   5 minutes ago    187MB
mysql             latest    65f3f983cb08   2 weeks ago      632MB
ubuntu            latest    ca2b0f26964c   6 weeks ago      77.9MB
alpine            latest    05455a08881e   2 months ago     7.38MB
hello-world       latest    d2c94e258dcb   11 months ago    13.3kB
root@jason-virtual-machine:~/linux_tweet_app#
```

```
root@jason-virtual-machine:~/linux_tweet_app# docker images
REPOSITORY        TAG       IMAGE ID       CREATED         SIZE
linux_tweet_app   latest    3441d302b565   5 minutes ago   187MB
mysql             latest    65f3f983cb08   2 weeks ago     632MB
ubuntu            latest    ca2b0f26964c   6 weeks ago     77.9MB
alpine            latest    05455a08881e   2 months ago    7.38MB
hello-world       latest    d2c94e258dcb   11 months ago   13.3kB
root@jason-virtual-machine:~/linux_tweet_app# docker run -d -p 80:80 3441d302b565
55ba4b3f11a6a672f68c9791b4725743191d598d711eab9cac771dd06a957ace
root@jason-virtual-machine:~/linux_tweet_app# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS          PORTS                                   NAMES
55ba4b3f11a6   3441d302b565   "/docker-entrypoint.…"   17 seconds ago   Up 16 seconds   0.0.0.0:80->80/tcp, :::80->80/tcp, 443/tcp   sweet_hypatia
root@jason-virtual-machine:~/linux_tweet_app#
```

Here I checked the Docker images and then ran the Linux_Tweet_App in dameon(-d) and -p to expose the ports, followed by the ports to expose (in this case 80:80) and then the image ID. Then I turned on the container with the webpage inside using the “docker ps” command.



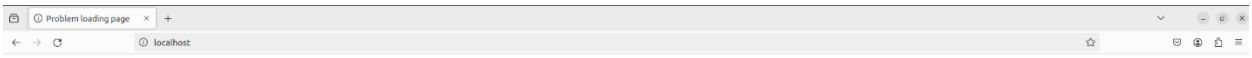
Here we can see the running Linux Tweet App when going to the local host page.

PUZZLER: seccomp with pbf rules

```
root@jason-virtual-machine:~/linux_tweet_app# docker start sweet_hypatia
sweet_hypatia
root@jason-virtual-machine:~/linux_tweet_app# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED         STATUS          PORTS                                   NAMES
55ba4b3f11a6   3441d302b565   "/docker-entrypoint.…"   7 minutes ago   Up 14 seconds   0.0.0.0:80->80/tcp, :::80->80/tcp, 443/tcp   sweet_hypatia
root@jason-virtual-machine:~/linux_tweet_app# docker stop sweet_hypatia
sweet_hypatia
root@jason-virtual-machine:~/linux_tweet_app#
```

```
root@jason-virtual-machine:~/linux_tweet_app# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED         STATUS         PORTS                                                  NAMES
55ba4b3f11a6   3441d302b565   "/docker-entrypoint.…"   17 seconds ago   Up 16 seconds   0.0.0.0:80->80/tcp, :::80->80/tcp, 443/tcp   sweet_hypatia
root@jason-virtual-machine:~/linux_tweet_app# docker stop sweet_hypatia
sweet_hypatia
root@jason-virtual-machine:~/linux_tweet_app# docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
root@jason-virtual-machine:~/linux_tweet_app# docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED         STATUS                     PORTS     NAMES
55ba4b3f11a6   3441d302b565   "/docker-entrypoint.…"   6 minutes ago   Exited (0) 18 seconds ago            sweet_hypatia
139a4f45aa5c   mysql:latest   "docker-entrypoint.s…"   2 days ago      Exited (0) 46 hours ago              msis603db
e334605dbbd2   alpine         "hostname"               2 days ago      Exited (0) 2 days ago                priceless_chandrasekhar
36e7091caa45   hello-world    "/hello"                 2 days ago      Exited (0) 2 days ago                exciting_borg
root@jason-virtual-machine:~/linux_tweet_app#
```

Here we can see the webpage container being started and stopped.



Unable to connect

Firefox can't establish a connection to the server at localhost.

- The site could be temporarily unavailable or too busy. Try again in a few moments.
- If you are unable to load any pages, check your computer's network connection.
- If your computer or network is protected by a firewall or proxy, make sure that Firefox is permitted to access the web.

Try Again

As we can see when it is stopped the webpage is not displayed.

```
root@jason-virtual-machine:~/linux_tweet_app# grep -i seccomp /boot/config-$(uname -r)
CONFIG_HAVE_ARCH_SECCOMP=y
CONFIG_HAVE_ARCH_SECCOMP_FILTER=y
CONFIG_SECCOMP=y
CONFIG_SECCOMP_FILTER=y
# CONFIG_SECCOMP_CACHE_DEBUG is not set
root@jason-virtual-machine:~/linux_tweet_app#
```

Here we can see seccomp is enabled on the Linux Kernel host VM.

Here we can see the profile.json file I created with some SCMP_ACT_ commands included.

Conclusion:

This lab was straight forward for many of the sections as the knowledge to complete the desired tasks are easily accessible. Everything went smoothly in this lab until the Puzzler section where I had trouble restricting syscalls from the default.json profile for the web server docker container. I was able to write a docker seccomp json profile with some SCMP_ACT_ commands. This was great experience and I now have some knowledge with Docker and containers.

References:

https://docs.docker.com/engine/install/ubuntu/

https://training.play-with-docker.com/beginner-linux/

https://www.digitalocean.com/community/tutorials/how-to-use-docker-exec-to-run-commands-in-a-docker-container

https://www.youtube.com/watch?v=SnSH8Ht3MIc

https://metacpan.org/pod/Linux::Seccomp