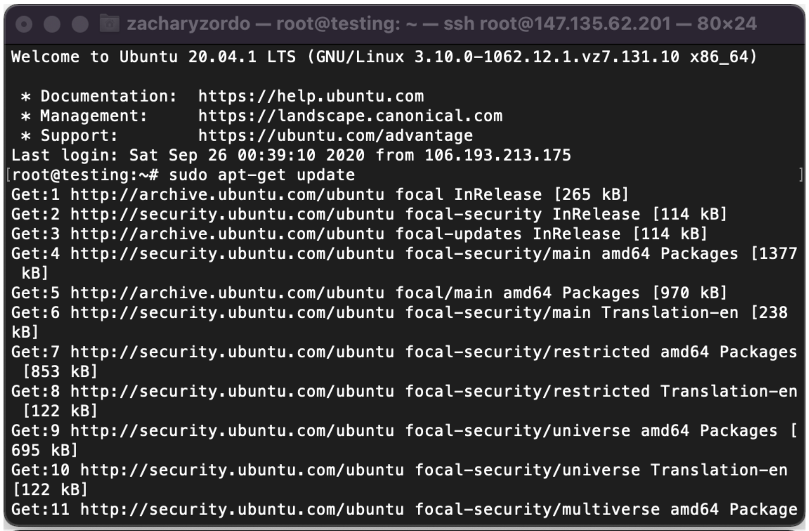
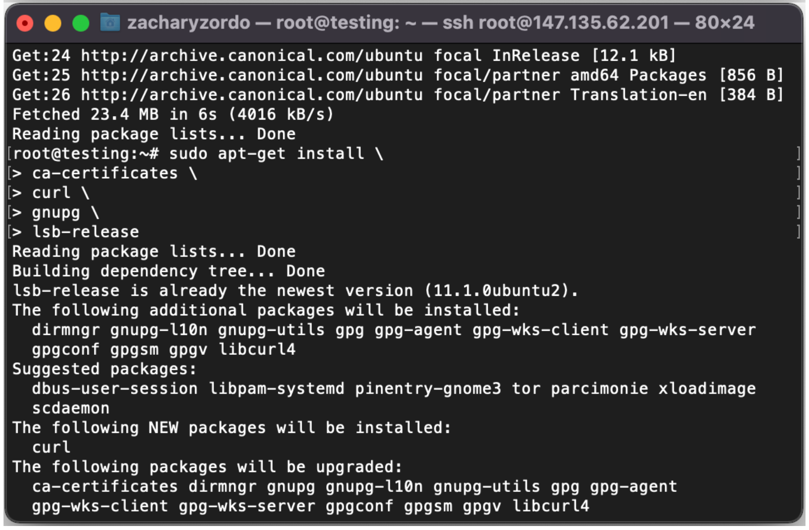
Installing Docker on VM

1. **Setting up Docker Repository – Updating apt and allowing apt to use a repository over HTTPS**
   apt is a command-line interface that allows us to manage the packages we want to install on our virtual machine. It will allow us to install docker, but first, we want to make sure apt is up-to-date.

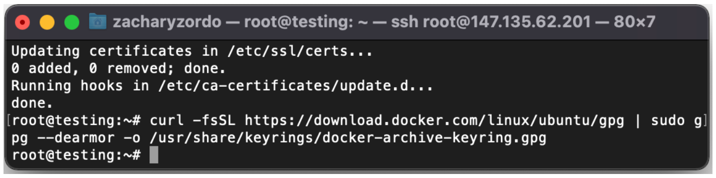

zacharyzordo — root@testing: ~ — ssh root@147.135.62.201 — 80×24

```
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 3.10.0-1062.12.1.vz7.131.10 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
Last login: Sat Sep 26 00:39:10 2020 from 106.193.213.175
[root@testing:~# sudo apt-get update
Get:1 http://archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Get:2 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:3 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [1377
 kB]
Get:5 http://archive.ubuntu.com/ubuntu focal/main amd64 Packages [970 kB]
Get:6 http://security.ubuntu.com/ubuntu focal-security/main Translation-en [238
kB]
Get:7 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packages
 [853 kB]
Get:8 http://security.ubuntu.com/ubuntu focal-security/restricted Translation-en
 [122 kB]
Get:9 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [
695 kB]
Get:10 http://security.ubuntu.com/ubuntu focal-security/universe Translation-en
[122 kB]
Get:11 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 Package
```


zacharyzordo — root@testing: ~ — ssh root@147.135.62.201 — 80×24

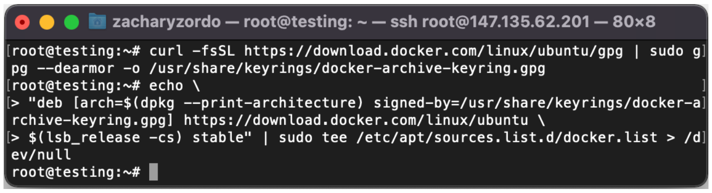
```
Get:24 http://archive.canonical.com/ubuntu focal InRelease [12.1 kB]
Get:25 http://archive.canonical.com/ubuntu focal/partner amd64 Packages [856 B]
Get:26 http://archive.canonical.com/ubuntu focal/partner Translation-en [384 B]
Fetched 23.4 MB in 6s (4016 kB/s)
Reading package lists... Done
[root@testing:~# sudo apt-get install \
[> ca-certificates \
[> curl \
[> gnupg \
[> lsb-release
Reading package lists... Done
Building dependency tree... Done
lsb-release is already the newest version (11.1.0ubuntu2).
The following additional packages will be installed:
  dirmngr gnupg-l10n gnupg-utils gpg gpg-agent gpg-wks-client gpg-wks-server
  gpgconf gpgsm gpgv libcurl4
Suggested packages:
  dbus-user-session libpam-systemd pinentry-gnome3 tor parcimonie xloadimage
  scdaemon
The following NEW packages will be installed:
  curl
The following packages will be upgraded:
  ca-certificates dirmngr gnupg gnupg-l10n gnupg-utils gpg gpg-agent
  gpg-wks-client gpg-wks-server gpgconf gpgsm gpgv libcurl4
```

2. **Adding Docker's GPG key:**
   This ensures we are not installing docker software that has been tampered.


zacharyzordo — root@testing: ~ — ssh root@147.135.62.201 — 80×7

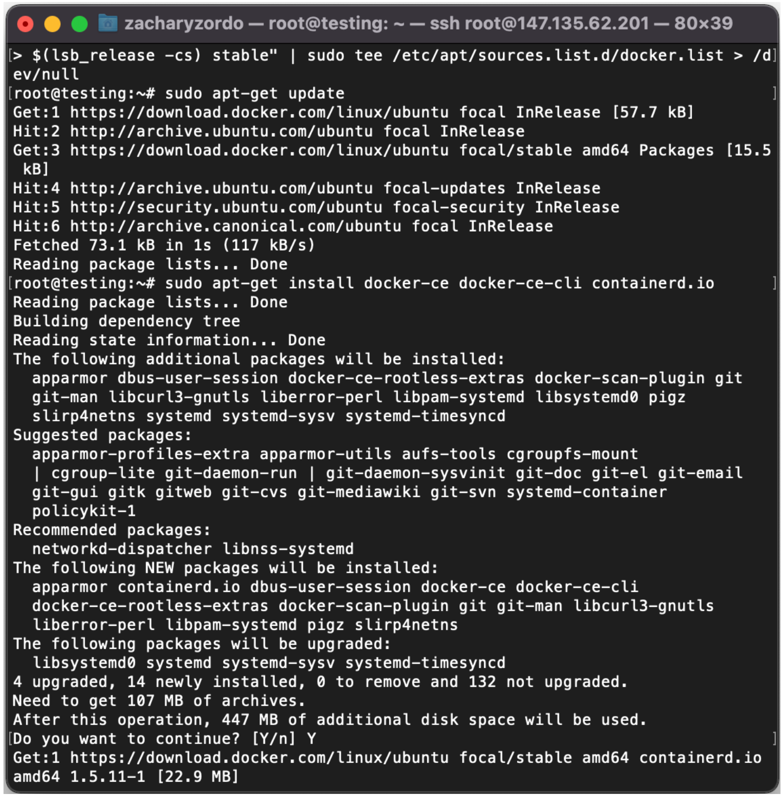
```
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
[root@testing:~# curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo g
pg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
root@testing:~# 
```

3. **Setting up the stable repository:**
   This will ensure we get the latest, stable version of Docker.


zacharyzordo — root@testing: ~ — ssh root@147.135.62.201 — 80×8

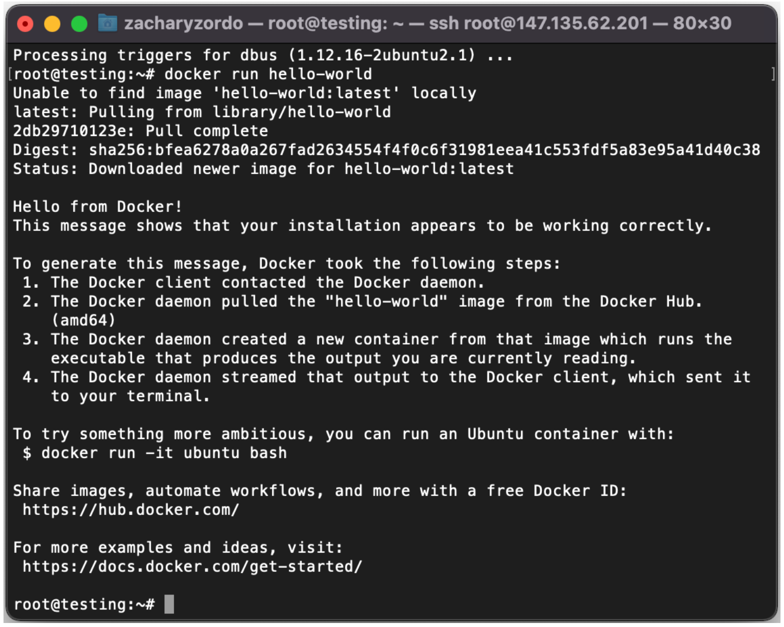
```
[root@testing:~# curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo g]
pg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
[root@testing:~# echo \                                                         ]
[> "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-a]
rchive-keyring.gpg] https://download.docker.com/linux/ubuntu \
[> $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /d]
ev/null
root@testing:~# 
```

4. **Installing Docker Engine – Updating apt package index and installing latest version of Docker Engine and containerd**


zacharyzordo — root@testing: ~ — ssh root@147.135.62.201 — 80×39

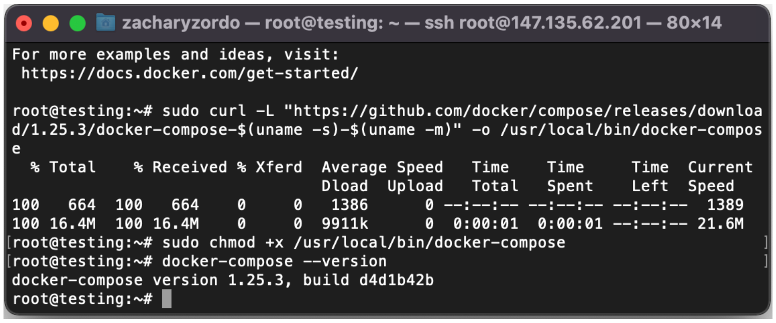
```
[> $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /d]
ev/null
[root@testing:~# sudo apt-get update                                              ]
Get:1 https://download.docker.com/linux/ubuntu focal InRelease [57.7 kB]
Hit:2 http://archive.ubuntu.com/ubuntu focal InRelease
Get:3 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages [15.5
 kB]
Hit:4 http://archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:5 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:6 http://archive.canonical.com/ubuntu focal InRelease
Fetched 73.1 kB in 1s (117 kB/s)
Reading package lists... Done
[root@testing:~# sudo apt-get install docker-ce docker-ce-cli containerd.io       ]
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apparmor dbus-user-session docker-ce-rootless-extras docker-scan-plugin git
  git-man libcurl3-gnutls liberror-perl libpam-systemd libsystemd0 pigz
  slirp4netns systemd systemd-sysv systemd-timesyncd
Suggested packages:
  apparmor-profiles-extra apparmor-utils aufs-tools cgroupfs-mount
  | cgroup-lite git-daemon-run | git-daemon-sysvinit git-doc git-el git-email
  git-gui gitk gitweb git-cvs git-mediawiki git-svn systemd-container
  policykit-1
Recommended packages:
  networkd-dispatcher libnss-systemd
The following NEW packages will be installed:
  apparmor containerd.io dbus-user-session docker-ce docker-ce-cli
  docker-ce-rootless-extras docker-scan-plugin git git-man libcurl3-gnutls
  liberror-perl libpam-systemd pigz slirp4netns
The following packages will be upgraded:
  libsystemd0 systemd systemd-sysv systemd-timesyncd
4 upgraded, 14 newly installed, 0 to remove and 132 not upgraded.
Need to get 107 MB of archives.
After this operation, 447 MB of additional disk space will be used.
[Do you want to continue? [Y/n] Y                                                 ]
Get:1 https://download.docker.com/linux/ubuntu focal/stable amd64 containerd.io
amd64 1.5.11-1 [22.9 MB]
```

5. **Verifying that Docker Engine is installed correctly by running hello-world**
   We attempt to run hello-world from docker. If it runs successfully, we'll know that Docker was installed correctly. Here, since the hello-world image was not yet available on our Docker, it first pulls it from the library prior to running it. It ran successfully, so we're good to go.


zacharyzordo — root@testing: ~ — ssh root@147.135.62.201 — 80×30

```
Processing triggers for dbus (1.12.16-2ubuntu2.1) ...
root@testing:~# docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:bfea6278a0a267fad2634554f4f0c6f31981eea41c553fdf5a83e95a41d40c38
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/

root@testing:~# 
```

6. **Installing and Verifying Version of Docker Compose:**
   Specified version 1.25.3 for Docker Compose. Then applied executable permissions, and finally, checked the version that was installed to verify the correct build was installed.

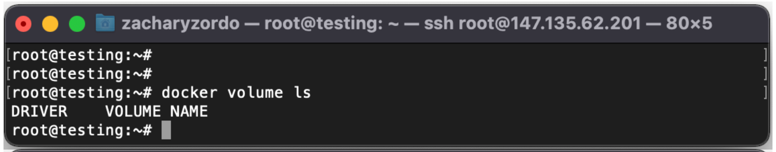

zacharyzordo — root@testing: ~ — ssh root@147.135.62.201 — 80×14

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

root@testing:~# sudo curl -L "https://github.com/docker/compose/releases/downloa
d/1.25.3/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compos
e
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   664  100   664    0     0   1386      0 --:--:-- --:--:-- --:--:--  1389
100 16.4M  100 16.4M    0     0  9911k      0  0:00:01  0:00:01 --:--:-- 21.6M
root@testing:~# sudo chmod +x /usr/local/bin/docker-compose
root@testing:~# docker-compose --version
docker-compose version 1.25.3, build d4d1b42b
root@testing:~# 
```

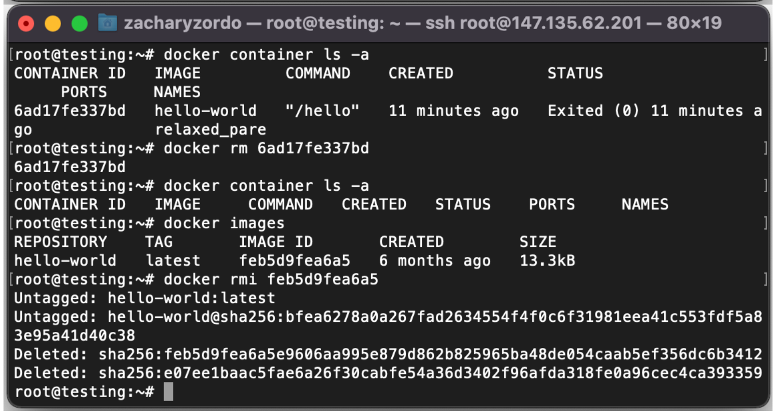
7. **Finding and Removing Volumes, Containers, and Images:**
   Using docker volume ls, we can check if there are any volumes available. There are none in this case.

   Using docker container ls -a, we can check if there are any containers available (using -a ensures we get all containers, not just ones currently running). I used the rm command to remove the hello-world image by its Container ID.

   Using docker images, I could find that there was an image I want to remove. In this case, I used docker rmi feb5d9fea6a5 to remove the image by its Image ID.

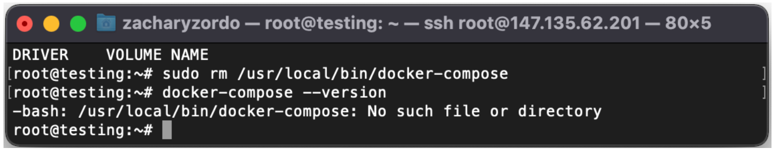

zacharyzordo — root@testing: ~ — ssh root@147.135.62.201 — 80×5

```
root@testing:~#
root@testing:~#
root@testing:~# docker volume ls
DRIVER    VOLUME NAME
root@testing:~# 
```


zacharyzordo — root@testing: ~ — ssh root@147.135.62.201 — 80×19

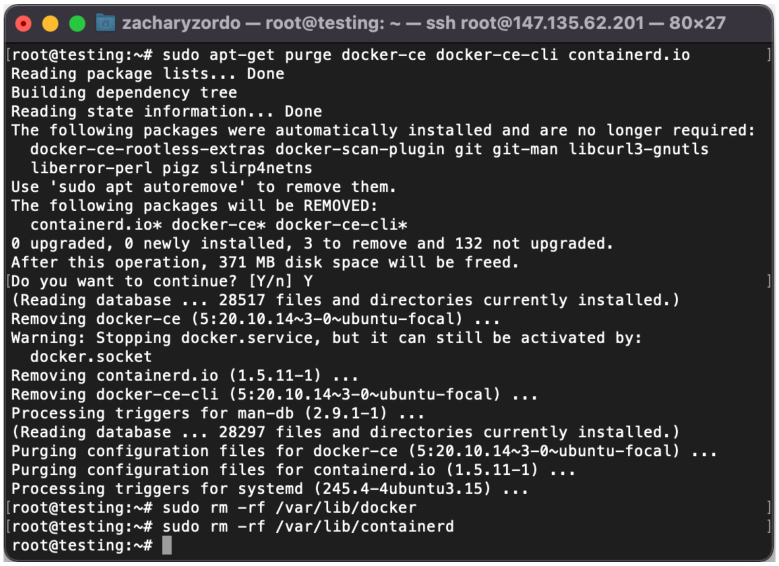
```
root@testing:~# docker container ls -a
CONTAINER ID   IMAGE         COMMAND    CREATED          STATUS
     PORTS     NAMES
6ad17fe337bd   hello-world   "/hello"   11 minutes ago   Exited (0) 11 minutes a
go             relaxed_pare
root@testing:~# docker rm 6ad17fe337bd
6ad17fe337bd
root@testing:~# docker container ls -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
root@testing:~# docker images
REPOSITORY    TAG       IMAGE ID       CREATED         SIZE
hello-world   latest    feb5d9fea6a5   6 months ago    13.3kB
root@testing:~# docker rmi feb5d9fea6a5
Untagged: hello-world:latest
Untagged: hello-world@sha256:bfea6278a0a267fad2634554f4f0c6f31981eea41c553fdf5a8
3e95a41d40c38
Deleted: sha256:feb5d9fea6a5e9606aa995e879d862b825965ba48de054caab5ef356dc6b3412
Deleted: sha256:e07ee1baac5fae6a26f30cabfe54a36d3402f96afda318fe0a96cec4ca393359
root@testing:~# 
```

8. **Uninstalling Docker Compose:**


zacharyzordo — root@testing: ~ — ssh root@147.135.62.201 — 80×5

```
DRIVER    VOLUME NAME
root@testing:~# sudo rm /usr/local/bin/docker-compose
root@testing:~# docker-compose --version
-bash: /usr/local/bin/docker-compose: No such file or directory
root@testing:~# 
```

9. **Uninstalling Docker Engine, and Deleting All Images, Containers, and Volumes:**



zacharyzordo — root@testing: ~ — ssh root@147.135.62.201 — 80×27

```
[root@testing:~# sudo apt-get purge docker-ce docker-ce-cli containerd.io      ]
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  docker-ce-rootless-extras docker-scan-plugin git git-man libcurl3-gnutls
  liberror-perl pigz slirp4netns
Use 'sudo apt autoremove' to remove them.
The following packages will be REMOVED:
  containerd.io* docker-ce* docker-ce-cli*
0 upgraded, 0 newly installed, 3 to remove and 132 not upgraded.
After this operation, 371 MB disk space will be freed.
[Do you want to continue? [Y/n] Y                                               ]
(Reading database ... 28517 files and directories currently installed.)
Removing docker-ce (5:20.10.14~3-0~ubuntu-focal) ...
Warning: Stopping docker.service, but it can still be activated by:
  docker.socket
Removing containerd.io (1.5.11-1) ...
Removing docker-ce-cli (5:20.10.14~3-0~ubuntu-focal) ...
Processing triggers for man-db (2.9.1-1) ...
(Reading database ... 28297 files and directories currently installed.)
Purging configuration files for docker-ce (5:20.10.14~3-0~ubuntu-focal) ...
Purging configuration files for containerd.io (1.5.11-1) ...
Processing triggers for systemd (245.4-4ubuntu3.15) ...
[root@testing:~# sudo rm -rf /var/lib/docker                                    ]
[root@testing:~# sudo rm -rf /var/lib/containerd                                ]
root@testing:~# 
```