Name: Davonn Escobilla	**Date Performed: 15/11/2022**
Course/Section: CPE31S24	**Date Submitted: 16/11/2022**
Instructor: Jonathan Taylar	**Semester and SY: 1st, 2022-2023**

Activity 11: Containerization

1. Objectives

Create a Dockerfile and form a workflow using Ansible as Infrastructure as Code (IaC) to enable Continuous Delivery process

2. Discussion

Docker is an open platform for developing, shipping, and running applications. Docker enables you to separate your applications from your infrastructure so you can deliver software quickly. With Docker, you can manage your infrastructure in the same ways you manage your applications. By taking advantage of Docker's methodologies for shipping, testing, and deploying code quickly, you can significantly reduce the delay between writing code and running it in production.

Source: https://docs.docker.com/get-started/overview/

You may also check the difference between containers and virtual machines. Click the link given below.

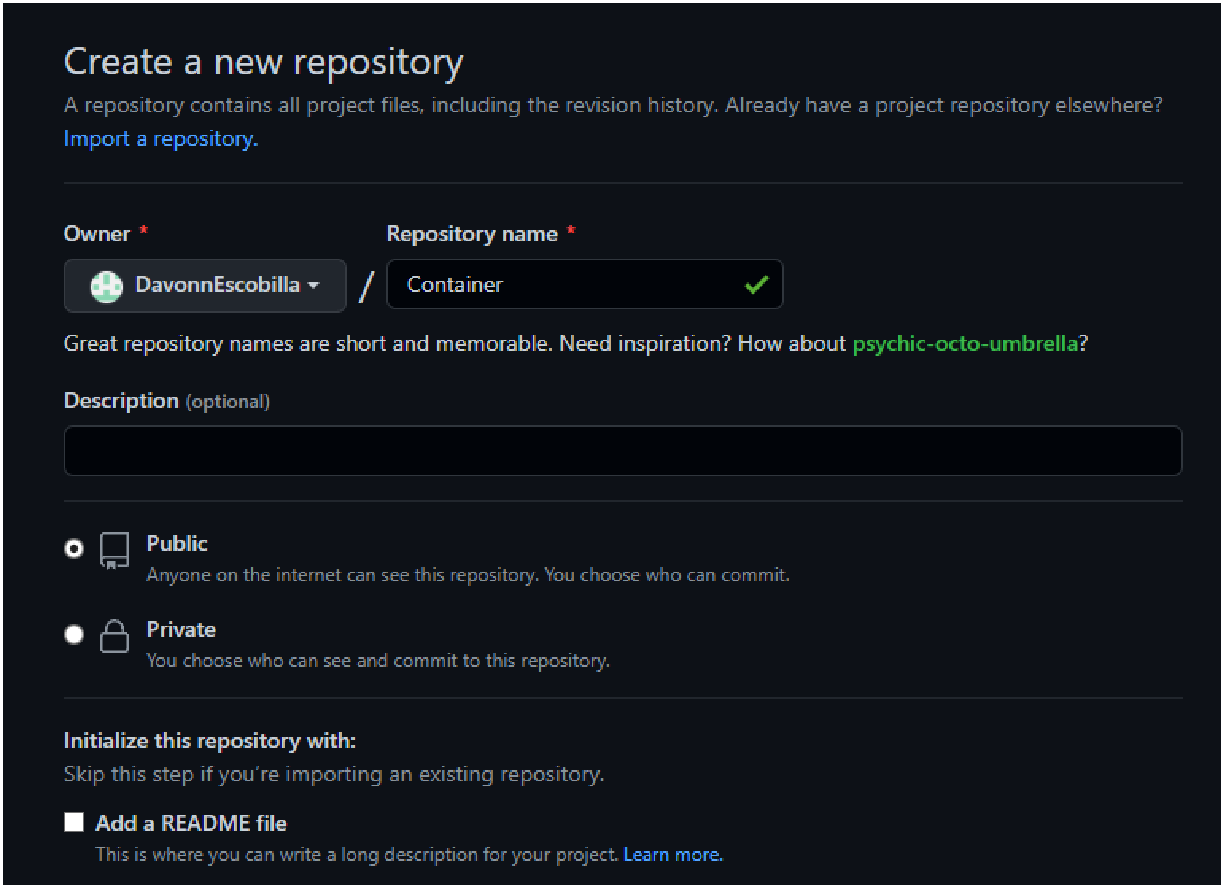
Source: https://docs.microsoft.com/en-us/virtualization/windowscontainers/about/containers-vs-vm

3. Tasks

1. Create a new repository for this activity.
2. Install Docker and enable the docker socket.
3. Add to Docker group to your current user.
4. Create a Dockerfile to install web and DB server.
5. Install and build the Dockerfile using Ansible.
6. Add, commit and push it to your repository.

4. **Output** (screenshots and explanations)

I created a new repository and then cloned it to my workstation or managed node.



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * DavonnEscobilla / Repository name * Container ✓

Great repository names are short and memorable. Need inspiration? How about psychic-octo-umbrella?

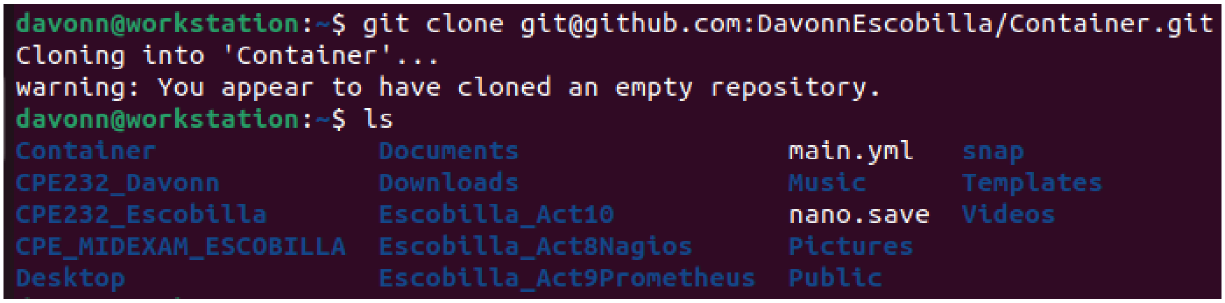
Description (optional)

◉ Public
Anyone on the internet can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

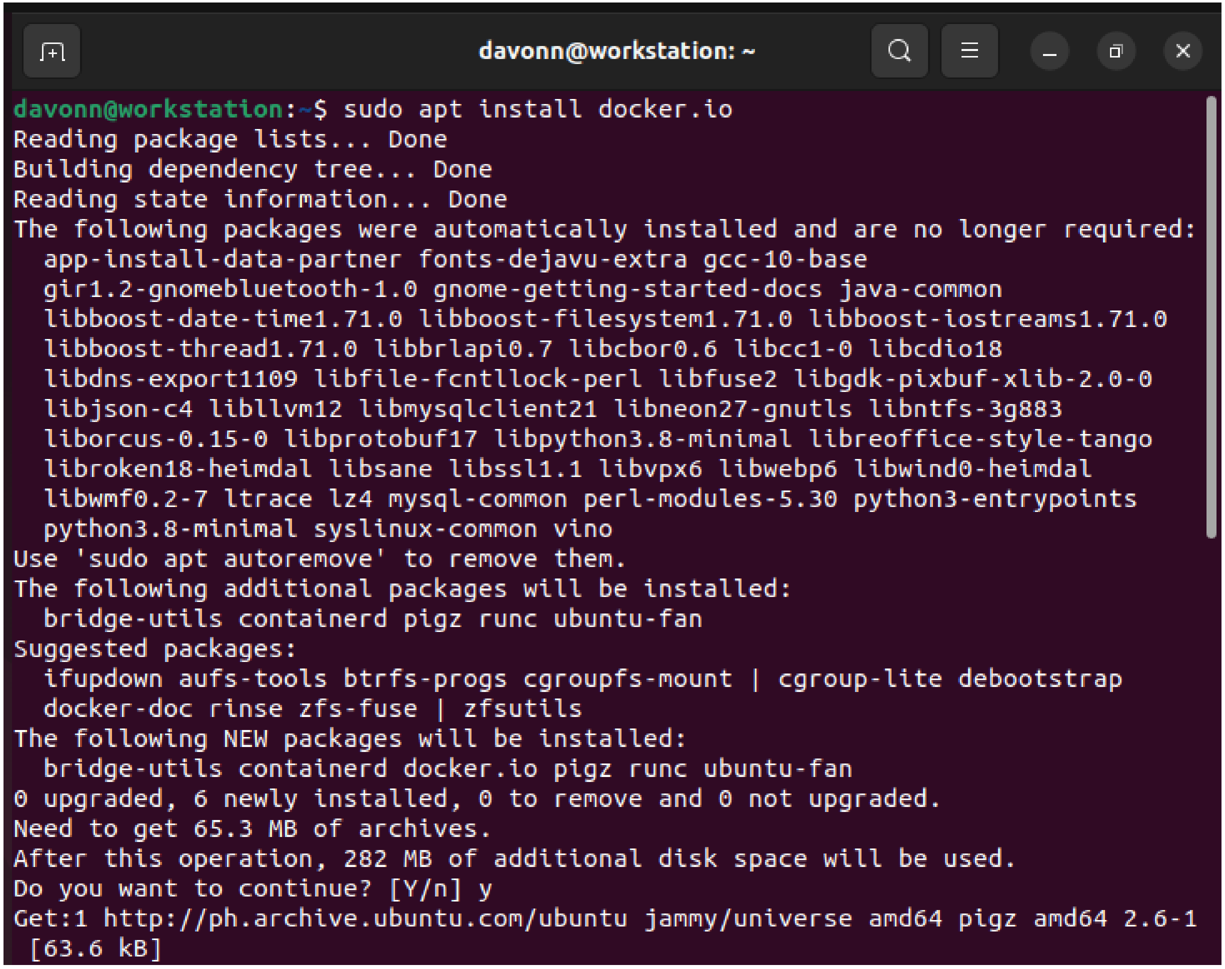
Initialize this repository with:
Skip this step if you're importing an existing repository.

□ Add a README file
This is where you can write a long description for your project. Learn more.

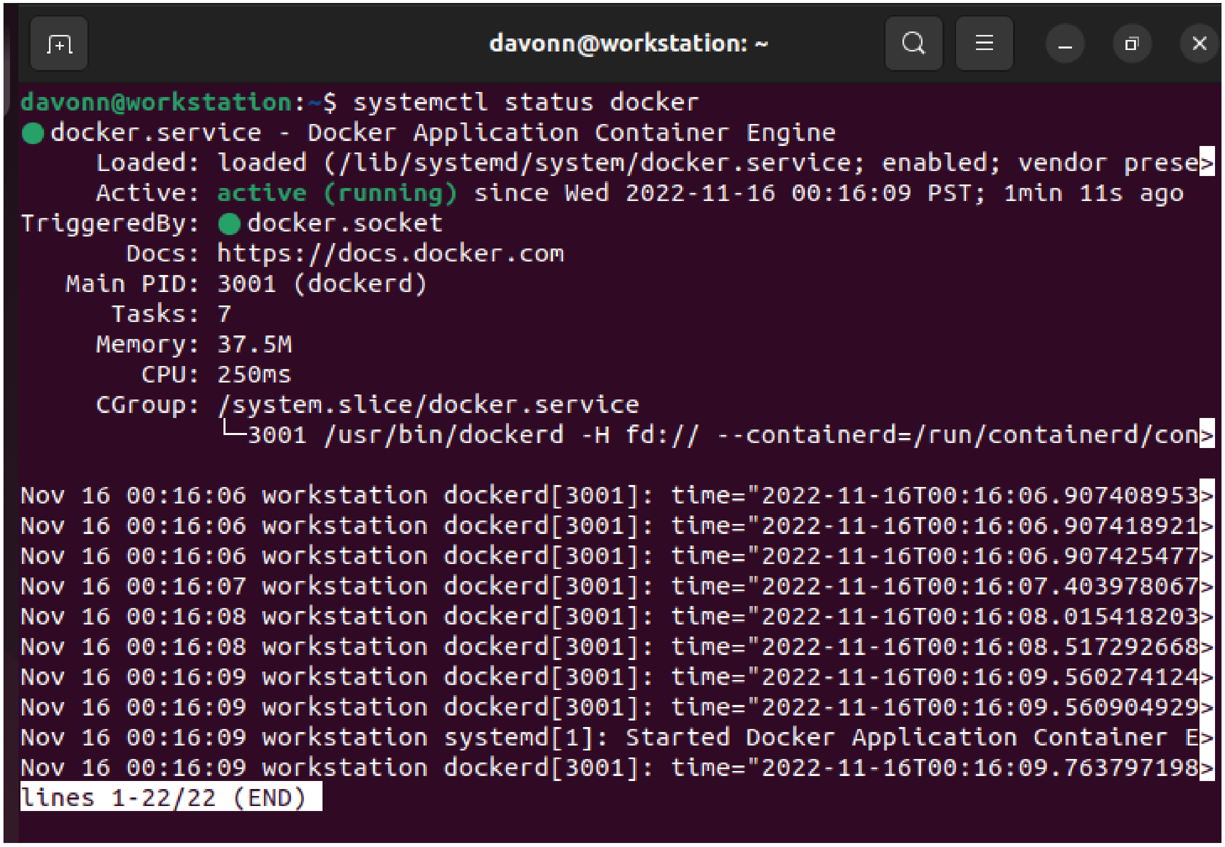


```
davonn@workstation:~$ git clone git@github.com:DavonnEscobilla/Container.git
Cloning into 'Container'...
warning: You appear to have cloned an empty repository.
davonn@workstation:~$ ls
Container              Documents                 main.yml   snap
CPE232_Davonn          Downloads                 Music      Templates
CPE232_Escobilla       Escobilla_Act10           nano.save  Videos
CPE_MIDEXAM_ESCOBILLA  Escobilla_Act8Nagios      Pictures
Desktop                Escobilla_Act9Prometheus  Public
```

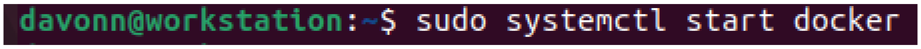
Install docker then enable docker socket on managed node.



```
davonn@workstation:~$ sudo apt install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  app-install-data-partner fonts-dejavu-extra gcc-10-base
  gir1.2-gnomebluetooth-1.0 gnome-getting-started-docs java-common
  libboost-date-time1.71.0 libboost-filesystem1.71.0 libboost-iostreams1.71.0
  libboost-thread1.71.0 libbrlapi0.7 libcbor0.6 libcc1-0 libcdio18
  libdns-export1109 libfile-fcntllock-perl libfuse2 libgdk-pixbuf-xlib-2.0-0
  libjson-c4 libllvm12 libmysqlclient21 libneon27-gnutls libntfs-3g883
  liborcus-0.15-0 libprotobuf17 libpython3.8-minimal libreoffice-style-tango
  libroken18-heimdal libsane libssl1.1 libvpx6 libwebp6 libwind0-heimdal
  libwmf0.2-7 ltrace lz4 mysql-common perl-modules-5.30 python3-entrypoints
  python3.8-minimal syslinux-common vino
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  bridge-utils containerd pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools btrfs-progs cgroupfs-mount | cgroup-lite debootstrap
  docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils containerd docker.io pigz runc ubuntu-fan
0 upgraded, 6 newly installed, 0 to remove and 0 not upgraded.
Need to get 65.3 MB of archives.
After this operation, 282 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ph.archive.ubuntu.com/ubuntu jammy/universe amd64 pigz amd64 2.6-1
 [63.6 kB]
```

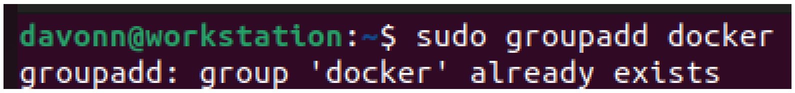


```
davonn@workstation:~$ systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor prese>
     Active: active (running) since Wed 2022-11-16 00:16:09 PST; 1min 11s ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 3001 (dockerd)
      Tasks: 7
     Memory: 37.5M
        CPU: 250ms
     CGroup: /system.slice/docker.service
             └─3001 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/con>

Nov 16 00:16:06 workstation dockerd[3001]: time="2022-11-16T00:16:06.907408953>
Nov 16 00:16:06 workstation dockerd[3001]: time="2022-11-16T00:16:06.907418921>
Nov 16 00:16:06 workstation dockerd[3001]: time="2022-11-16T00:16:06.907425477>
Nov 16 00:16:07 workstation dockerd[3001]: time="2022-11-16T00:16:07.403978067>
Nov 16 00:16:08 workstation dockerd[3001]: time="2022-11-16T00:16:08.015418203>
Nov 16 00:16:08 workstation dockerd[3001]: time="2022-11-16T00:16:08.517292668>
Nov 16 00:16:09 workstation dockerd[3001]: time="2022-11-16T00:16:09.560274124>
Nov 16 00:16:09 workstation dockerd[3001]: time="2022-11-16T00:16:09.560904929>
Nov 16 00:16:09 workstation systemd[1]: Started Docker Application Container E>
Nov 16 00:16:09 workstation dockerd[3001]: time="2022-11-16T00:16:09.763797198>
lines 1-22/22 (END)
```



```
davonn@workstation:~$ sudo systemctl start docker
```

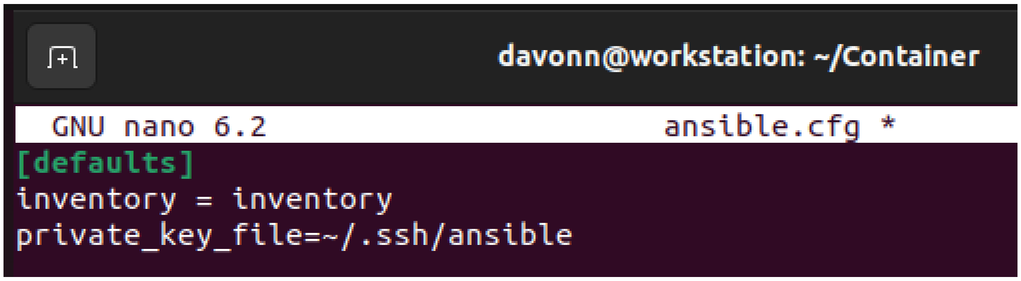
Add docker group to current user.



```
davonn@workstation:~$ sudo groupadd docker
groupadd: group 'docker' already exists
```

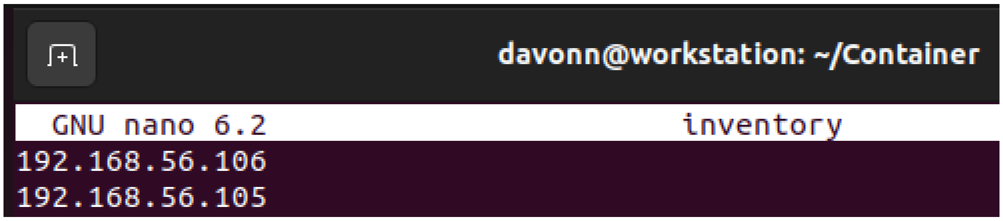
Create a docker file that installs DB server and web.

First create ansible.cfg



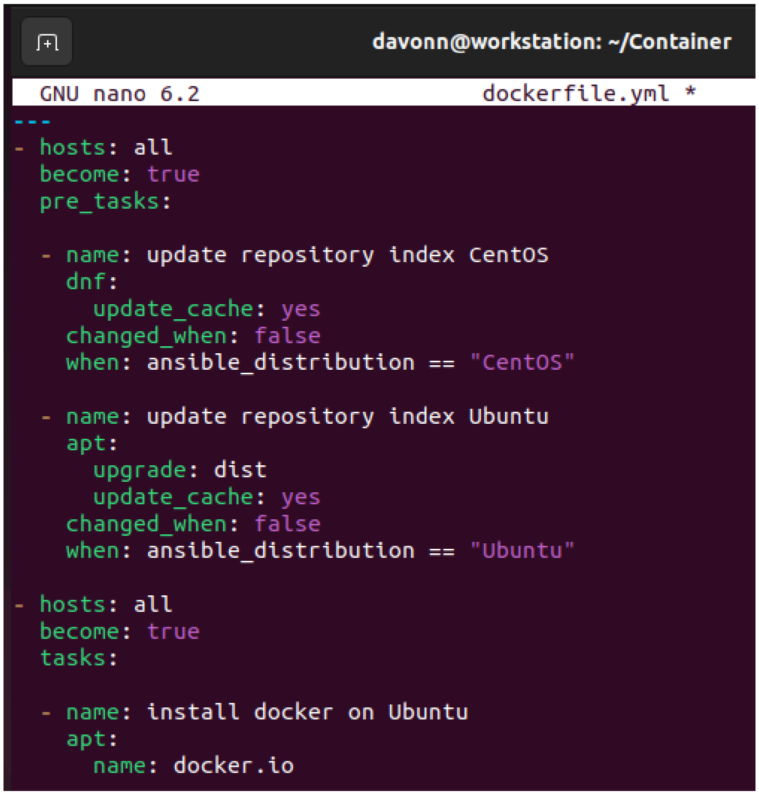
```
  GNU nano 6.2                        ansible.cfg *
[defaults]
inventory = inventory
private_key_file=~/.ssh/ansible
```

Then, create inventory for the ip address of control node.

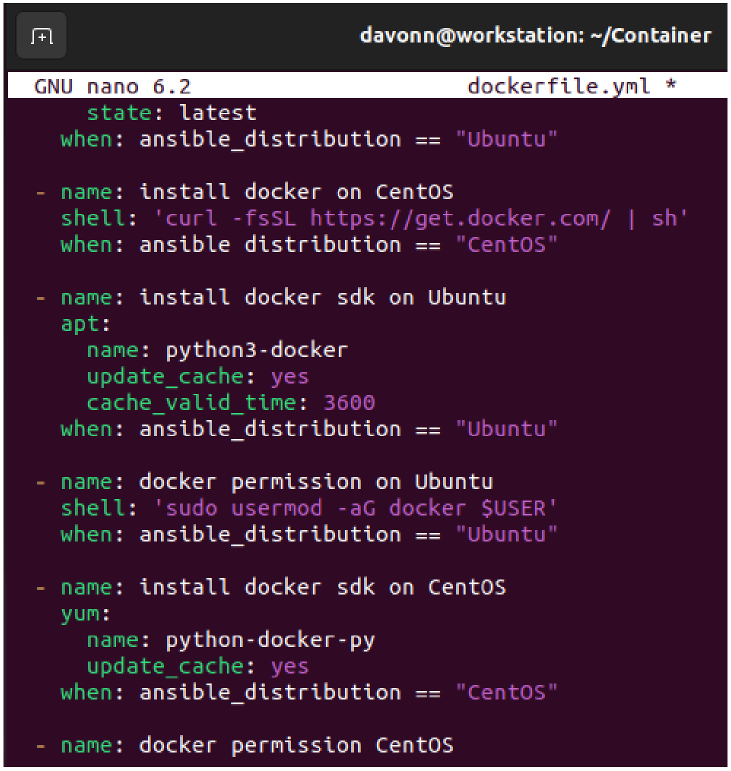


```
  GNU nano 6.2                     inventory
192.168.56.106
192.168.56.105
```

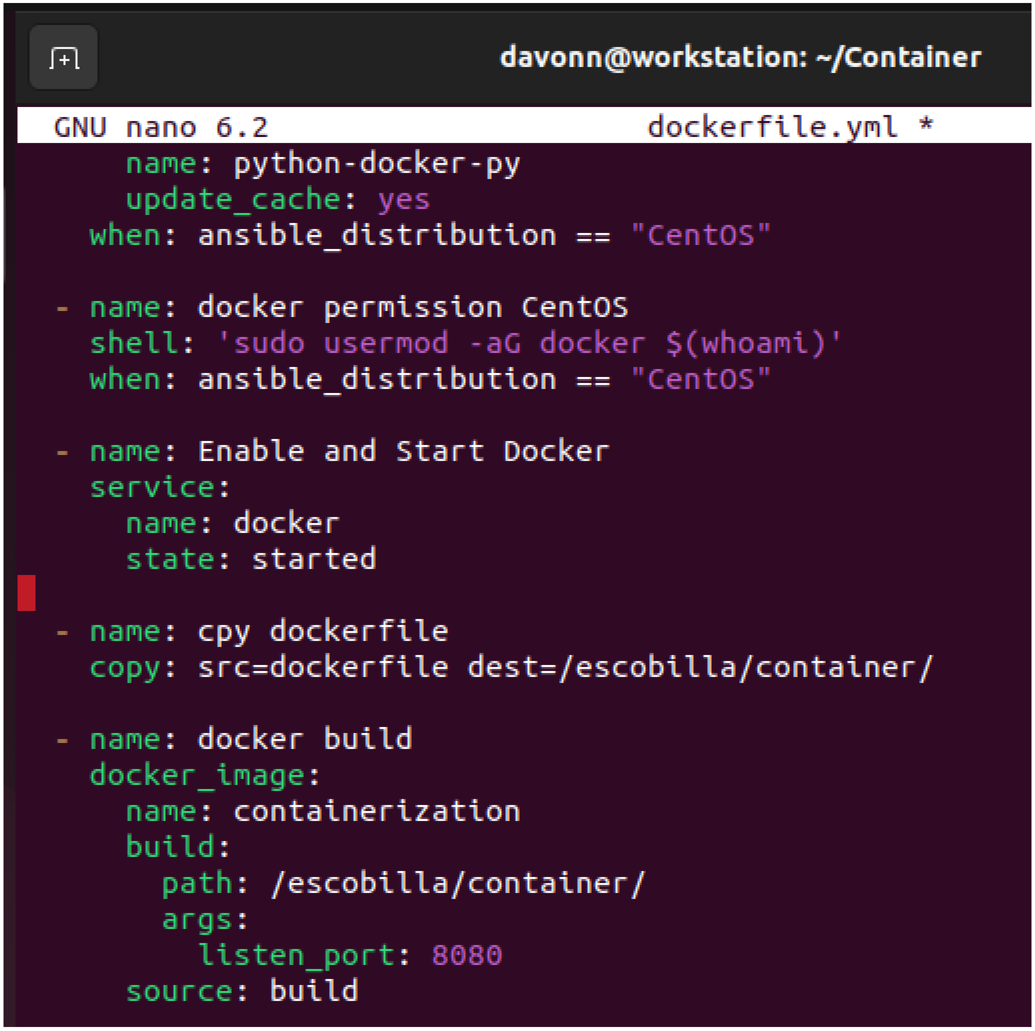
Then create playbook to install a web and dbserver on the control nodes.



```
  GNU nano 6.2                     dockerfile.yml *
---
- hosts: all
  become: true
  pre_tasks:

  - name: update repository index CentOS
    dnf:
      update_cache: yes
    changed_when: false
    when: ansible_distribution == "CentOS"

  - name: update repository index Ubuntu
    apt:
      upgrade: dist
      update_cache: yes
    changed_when: false
    when: ansible_distribution == "Ubuntu"

- hosts: all
  become: true
  tasks:

  - name: install docker on Ubuntu
    apt:
      name: docker.io
```

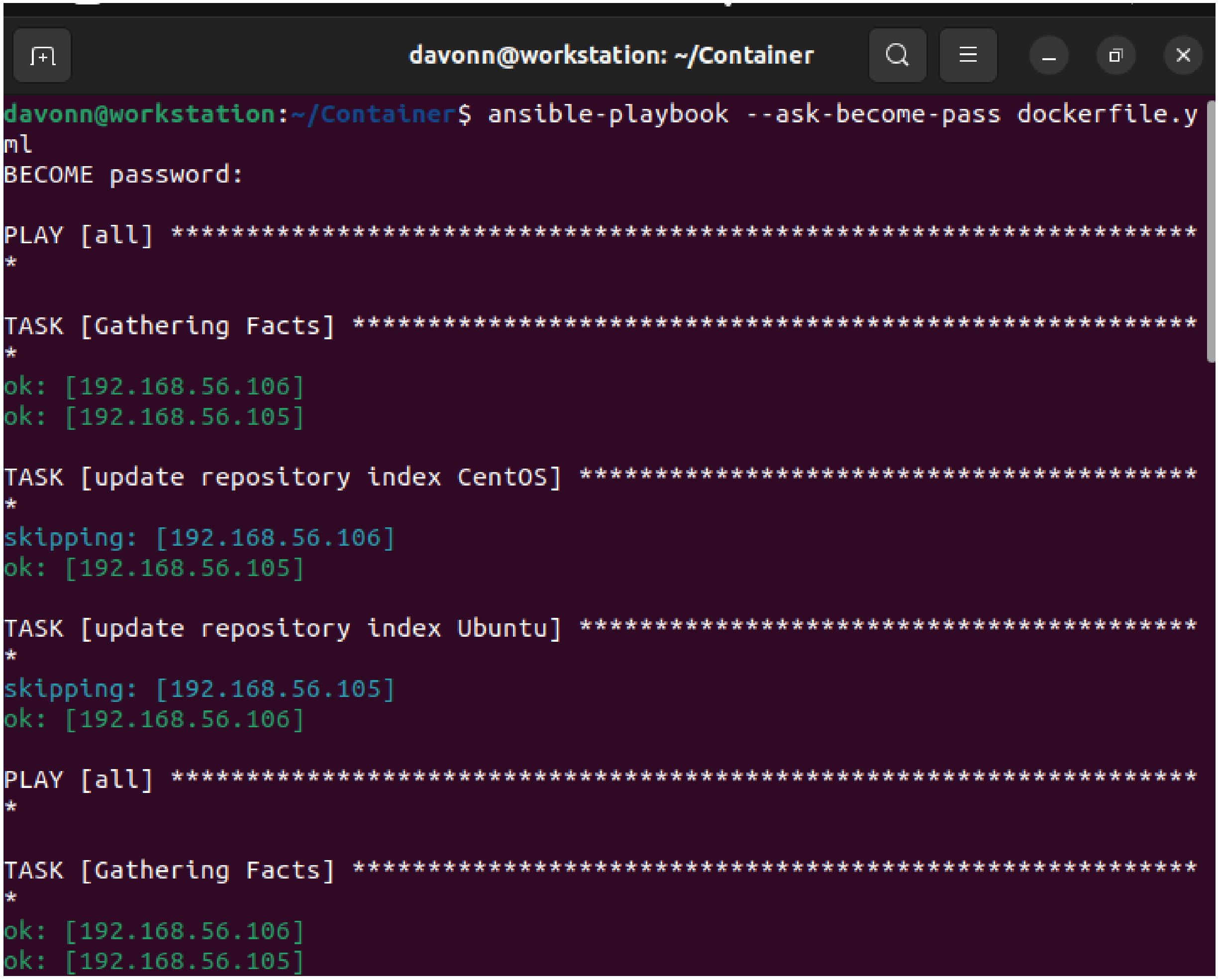


```
  GNU nano 6.2                        dockerfile.yml *
      state: latest
    when: ansible_distribution == "Ubuntu"

  - name: install docker on CentOS
    shell: 'curl -fsSL https://get.docker.com/ | sh'
    when: ansible distribution == "CentOS"

  - name: install docker sdk on Ubuntu
    apt:
      name: python3-docker
      update_cache: yes
      cache_valid_time: 3600
    when: ansible_distribution == "Ubuntu"

  - name: docker permission on Ubuntu
    shell: 'sudo usermod -aG docker $USER'
    when: ansible_distribution == "Ubuntu"

  - name: install docker sdk on CentOS
    yum:
      name: python-docker-py
      update_cache: yes
    when: ansible_distribution == "CentOS"

  - name: docker permission CentOS
```

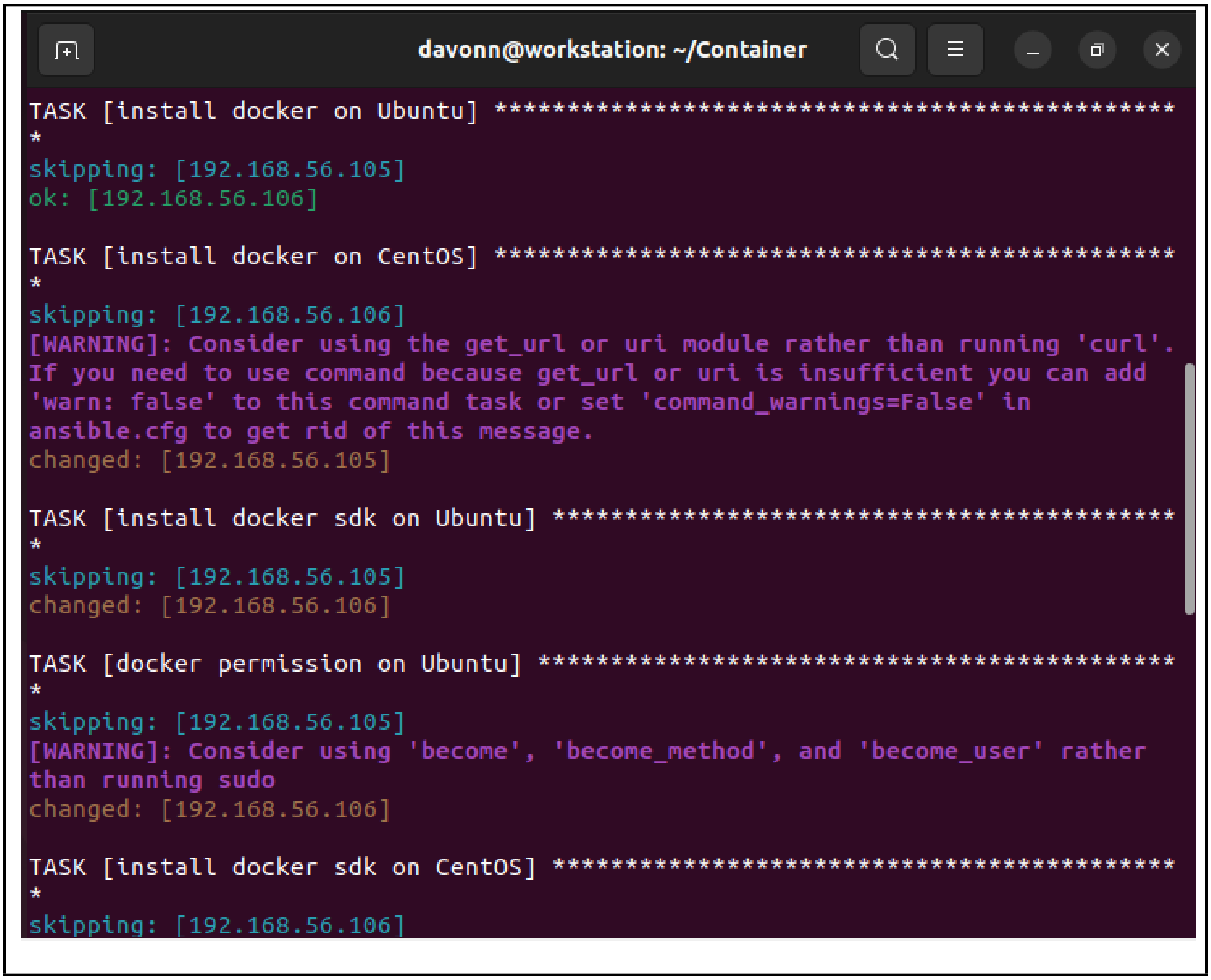


```
davonn@workstation: ~/Container

  GNU nano 6.2                   dockerfile.yml *
      name: python-docker-py
      update_cache: yes
    when: ansible_distribution == "CentOS"

  - name: docker permission CentOS
    shell: 'sudo usermod -aG docker $(whoami)'
    when: ansible_distribution == "CentOS"

  - name: Enable and Start Docker
    service:
      name: docker
      state: started

  - name: cpy dockerfile
    copy: src=dockerfile dest=/escobilla/container/

  - name: docker build
    docker_image:
      name: containerization
      build:
        path: /escobilla/container/
        args:
          listen_port: 8080
      source: build
```

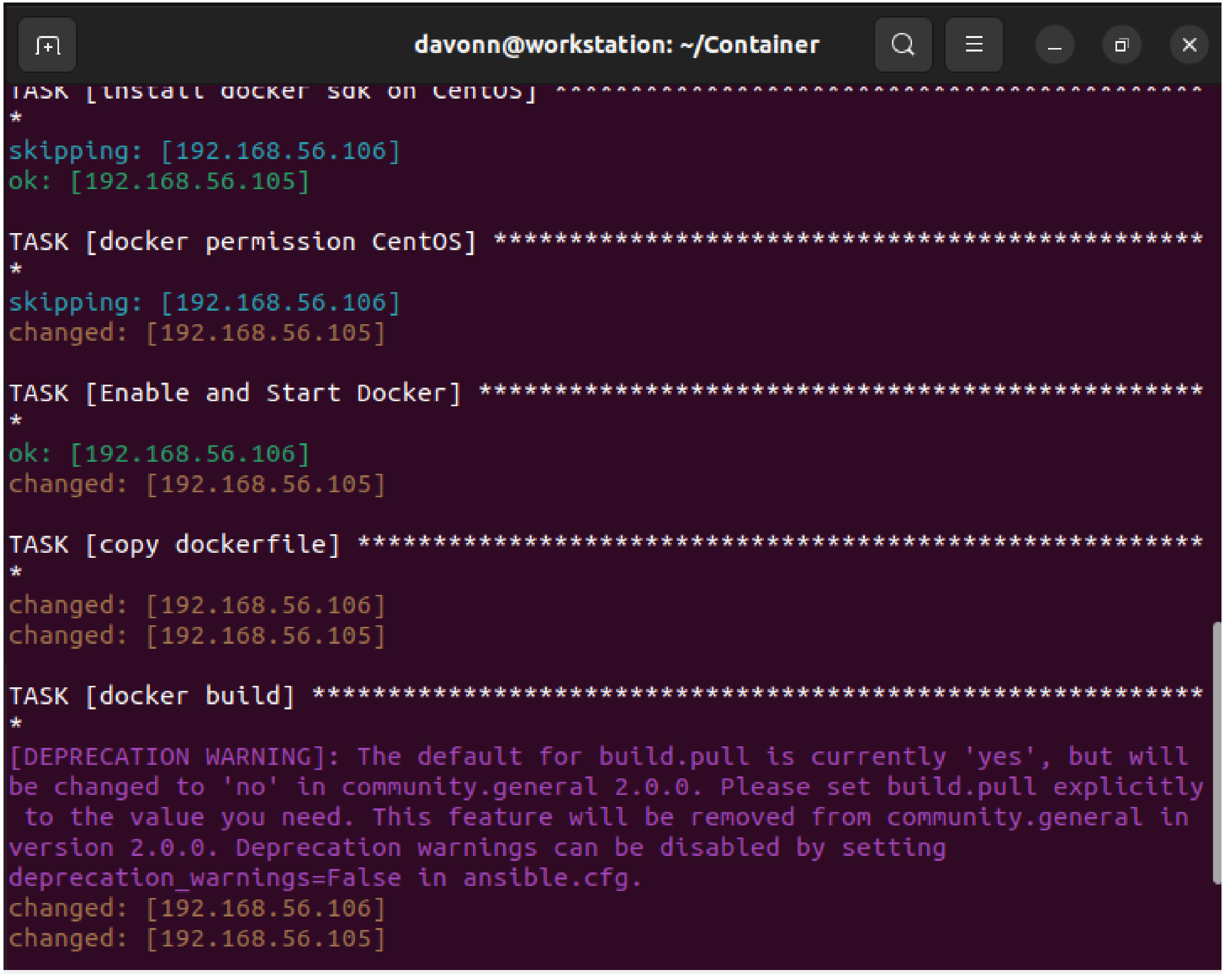
Now execute the playbook to install the tasks on control nodes.



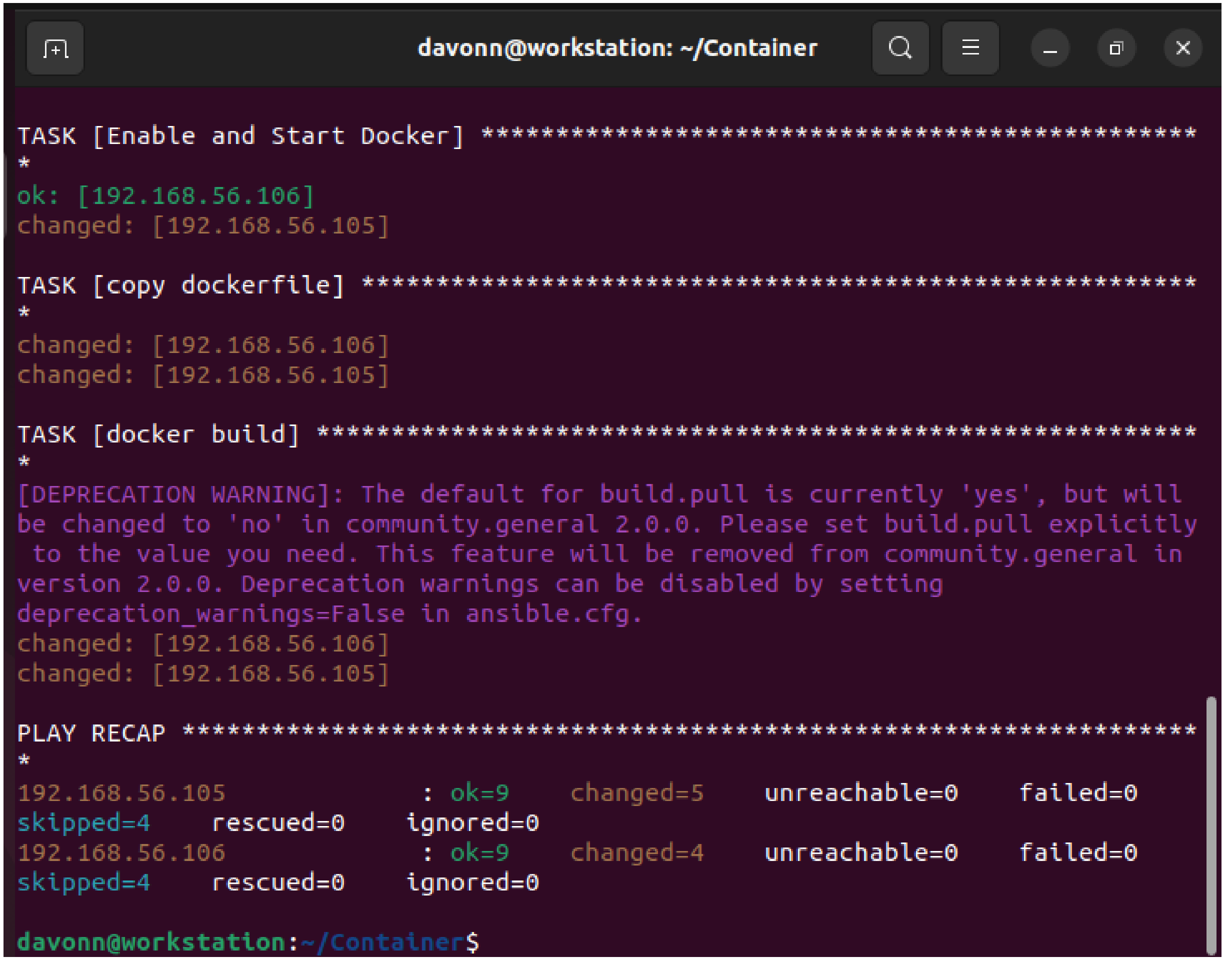
```
davonn@workstation:~/Container$ ansible-playbook --ask-become-pass dockerfile.y
ml
BECOME password:

PLAY [all] ********************************************************************
*

TASK [Gathering Facts] ********************************************************
*
ok: [192.168.56.106]
ok: [192.168.56.105]

TASK [update repository index CentOS] *****************************************
*
skipping: [192.168.56.106]
ok: [192.168.56.105]

TASK [update repository index Ubuntu] *****************************************
*
skipping: [192.168.56.105]
ok: [192.168.56.106]

PLAY [all] ********************************************************************
*

TASK [Gathering Facts] ********************************************************
*
ok: [192.168.56.106]
ok: [192.168.56.105]
```



```
TASK [install docker on Ubuntu] **********************************************
*
skipping: [192.168.56.105]
ok: [192.168.56.106]

TASK [install docker on CentOS] **********************************************
*
skipping: [192.168.56.106]
[WARNING]: Consider using the get_url or uri module rather than running 'curl'.
If you need to use command because get_url or uri is insufficient you can add
'warn: false' to this command task or set 'command_warnings=False' in
ansible.cfg to get rid of this message.
changed: [192.168.56.105]

TASK [install docker sdk on Ubuntu] ******************************************
*
skipping: [192.168.56.105]
changed: [192.168.56.106]

TASK [docker permission on Ubuntu] *******************************************
*
skipping: [192.168.56.105]
[WARNING]: Consider using 'become', 'become_method', and 'become_user' rather
than running sudo
changed: [192.168.56.106]

TASK [install docker sdk on CentOS] ******************************************
*
skipping: [192.168.56.106]
```



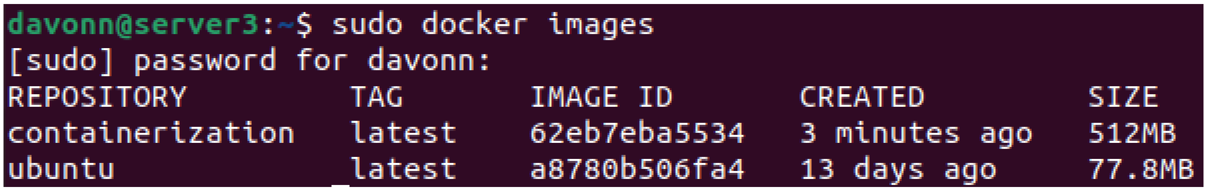
```
TASK [install docker sdk on CentOS] *******************************************
*
skipping: [192.168.56.106]
ok: [192.168.56.105]

TASK [docker permission CentOS] ***********************************************
*
skipping: [192.168.56.106]
changed: [192.168.56.105]

TASK [Enable and Start Docker] ************************************************
*
ok: [192.168.56.106]
changed: [192.168.56.105]

TASK [copy dockerfile] ********************************************************
*
changed: [192.168.56.106]
changed: [192.168.56.105]

TASK [docker build] ***********************************************************
*
[DEPRECATION WARNING]: The default for build.pull is currently 'yes', but will
be changed to 'no' in community.general 2.0.0. Please set build.pull explicitly
 to the value you need. This feature will be removed from community.general in
version 2.0.0. Deprecation warnings can be disabled by setting
deprecation_warnings=False in ansible.cfg.
changed: [192.168.56.106]
changed: [192.168.56.105]
```



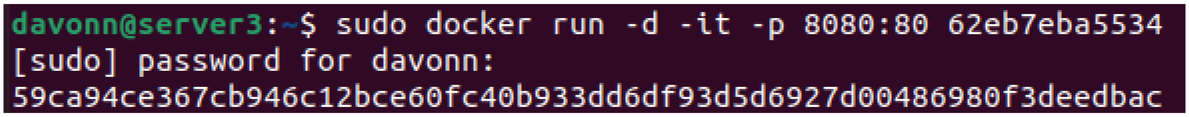
```
TASK [Enable and Start Docker] ************************************************
*
ok: [192.168.56.106]
changed: [192.168.56.105]

TASK [copy dockerfile] ********************************************************
*
changed: [192.168.56.106]
changed: [192.168.56.105]

TASK [docker build] ***********************************************************
*
[DEPRECATION WARNING]: The default for build.pull is currently 'yes', but will
be changed to 'no' in community.general 2.0.0. Please set build.pull explicitly
 to the value you need. This feature will be removed from community.general in
version 2.0.0. Deprecation warnings can be disabled by setting
deprecation_warnings=False in ansible.cfg.
changed: [192.168.56.106]
changed: [192.168.56.105]

PLAY RECAP ********************************************************************
*
192.168.56.105             : ok=9    changed=5    unreachable=0    failed=0
skipped=4    rescued=0    ignored=0
192.168.56.106             : ok=9    changed=4    unreachable=0    failed=0
skipped=4    rescued=0    ignored=0

davonn@workstation:~/Container$
```

After executing the playbook, we can now check the docker installation and verify it on the control nodes.

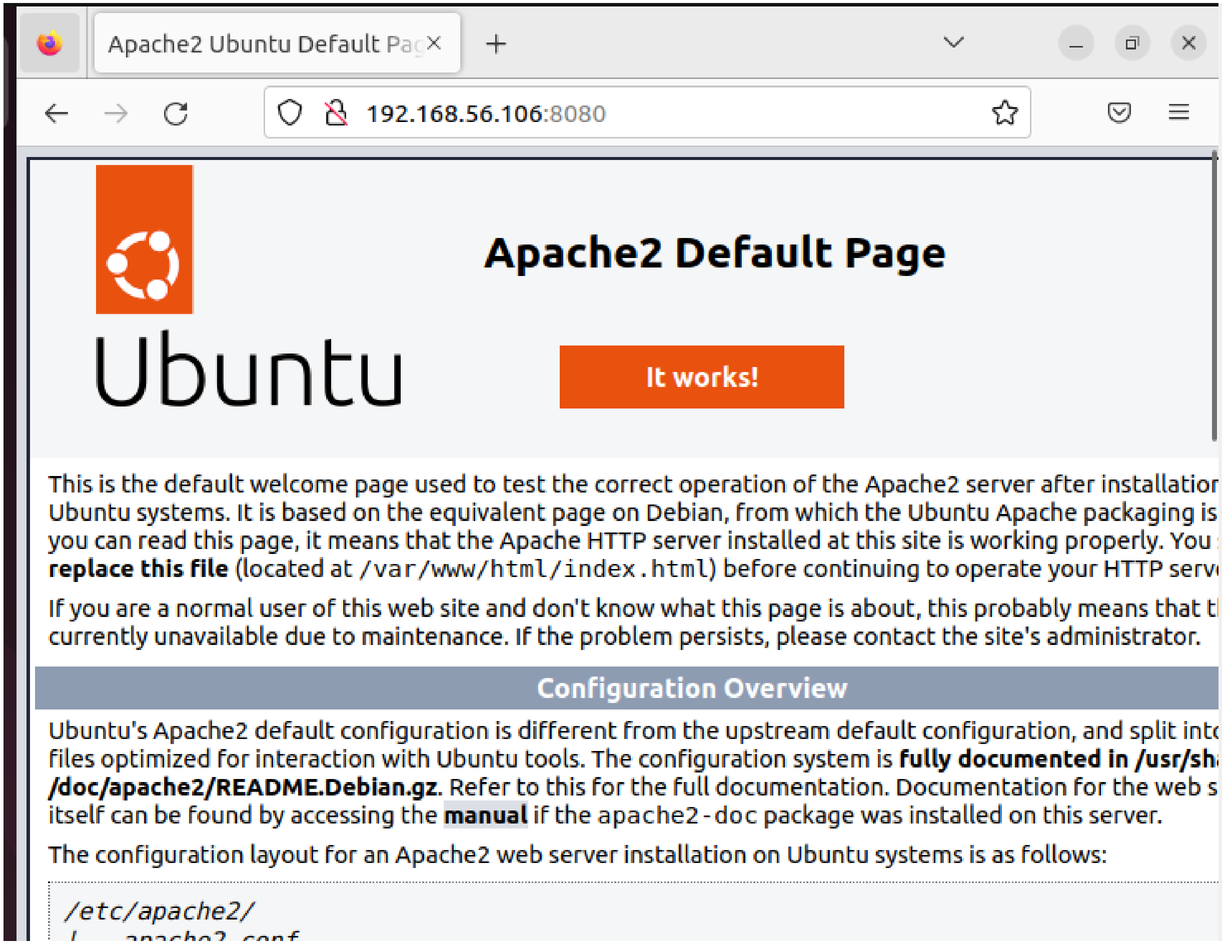
UBUNTU SERVER OUTPUT:



```
davonn@server3:~$ sudo docker images
[sudo] password for davonn:
REPOSITORY          TAG       IMAGE ID       CREATED          SIZE
containerization    latest    62eb7eba5534   3 minutes ago    512MB
ubuntu              latest    a8780b506fa4   13 days ago      77.8MB
```



```
davonn@server3:~$ sudo docker run -d -it -p 8080:80 62eb7eba5534
[sudo] password for davonn:
59ca94ce367cb946c12bce60fc40b933dd6df93d5d6927d00486980f3deedbac
```



Apache2 Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installatior Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is you can read this page, it means that the Apache HTTP server installed at this site is working properly. You **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP serv

If you are a normal user of this web site and don't know what this page is about, this probably means that t currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

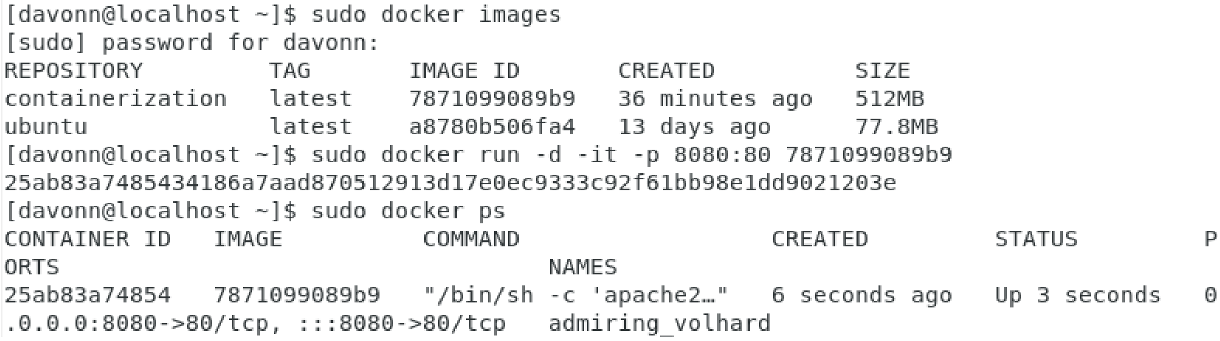
Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split intc files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/sh /doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web s itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

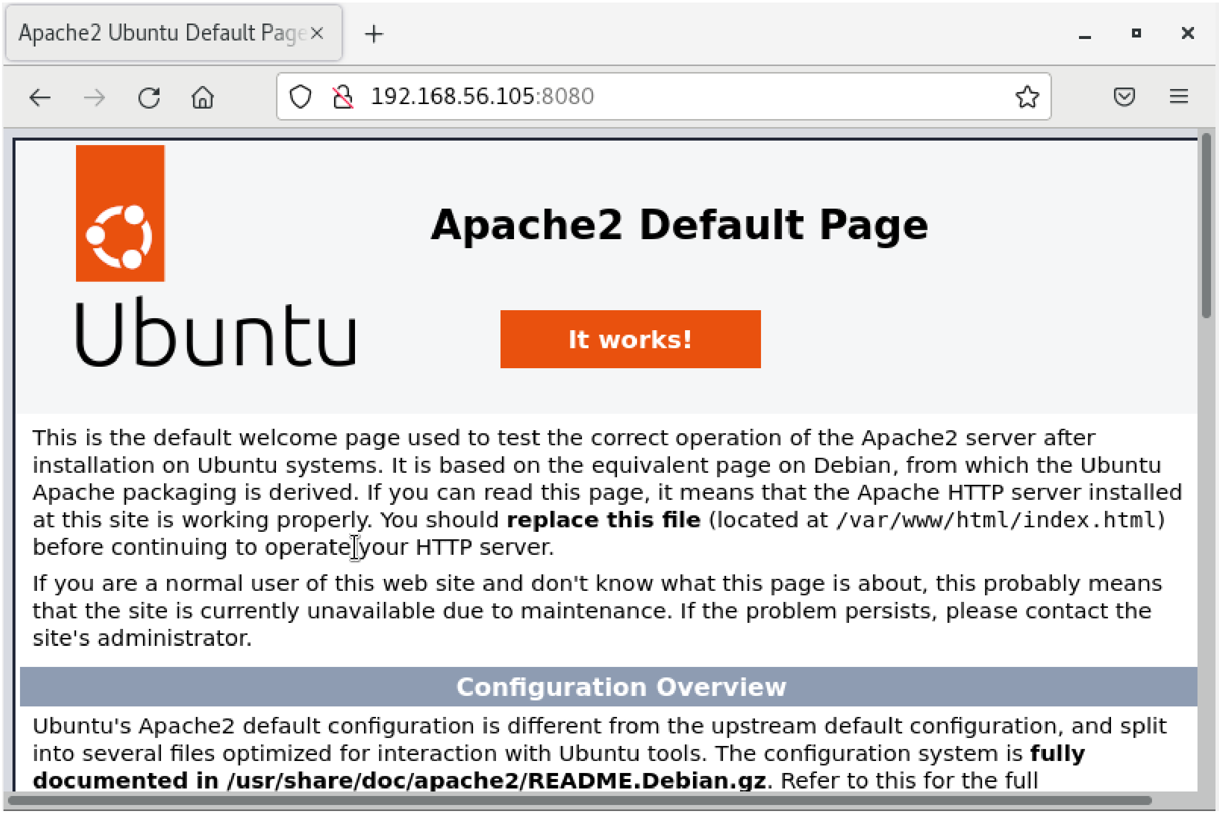
The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
```

CENTOS OUTPUT:



```
[davonn@localhost ~]$ sudo docker images
[sudo] password for davonn:
REPOSITORY         TAG       IMAGE ID       CREATED          SIZE
containerization   latest    7871099089b9   36 minutes ago   512MB
ubuntu             latest    a8780b506fa4   13 days ago      77.8MB
[davonn@localhost ~]$ sudo docker run -d -it -p 8080:80 7871099089b9
25ab83a7485434186a7aad870512913d17e0ec9333c92f61bb98e1dd9021203e
[davonn@localhost ~]$ sudo docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED         STATUS         P
ORTS                                NAMES
25ab83a74854   7871099089b9   "/bin/sh -c 'apache2…"   6 seconds ago   Up 3 seconds   0
.0.0.0:8080->80/tcp, :::8080->80/tcp   admiring_volhard
```



Apache2 Default Page

It works!

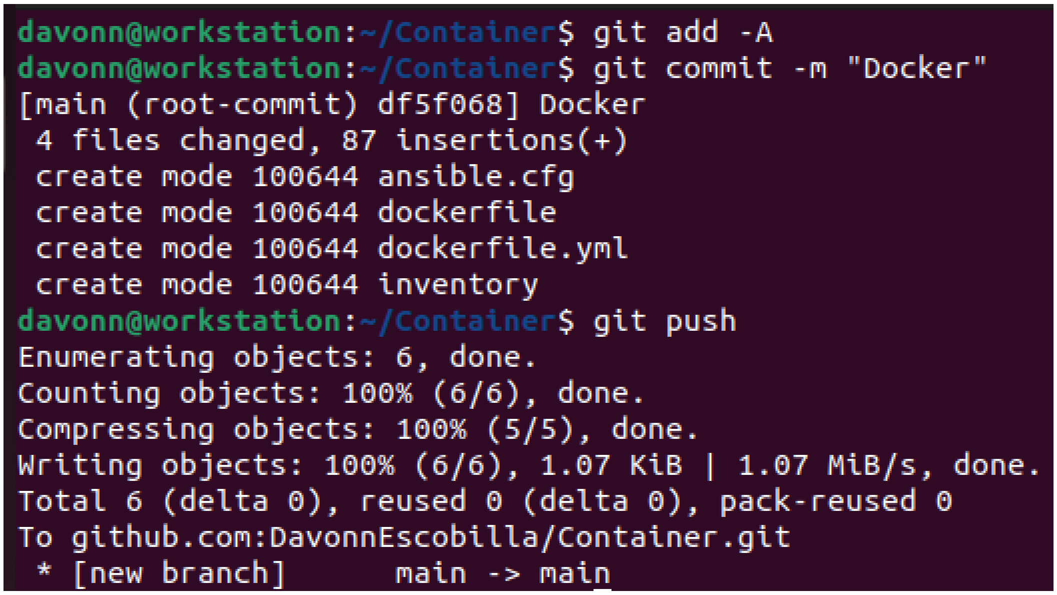
This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full

Lastly, commit the playbook to save the repository.



```
davonn@workstation:~/Container$ git add -A
davonn@workstation:~/Container$ git commit -m "Docker"
[main (root-commit) df5f068] Docker
 4 files changed, 87 insertions(+)
 create mode 100644 ansible.cfg
 create mode 100644 dockerfile
 create mode 100644 dockerfile.yml
 create mode 100644 inventory
davonn@workstation:~/Container$ git push
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 1.07 KiB | 1.07 MiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:DavonnEscobilla/Container.git
 * [new branch]      main -> main
```

Reflections:

Answer the following:

1. What are the benefits of implementing containerizations?

   Containers use less resources therefore it can be a good thing to efficiently save. It can also run almost anywhere since it can be easily copied and deployed. It can be also used as an alternative for virtual machines as it is not using much resources.

Conclusions:

Upon doing the activity, the simple task of implementing docker containerization is quite good to know since it can be an efficient tool to use if you are trying to save resources especially if you want to simulate or run multiple virtual machines. However, the downsides of using containers might be a problem in some cases so be better to know these things and take note of them. Containerization is a very good tool if used effectively.