Name:Hermitano, Johnny C.	**Date Performed: 11/23/2022**
Course/Section: CPE31S23	**Date Submitted: 11/23/2022**
Instructor: Engr. Jonathan Taylar	**Semester and SY: !st sem sy 2022**

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.

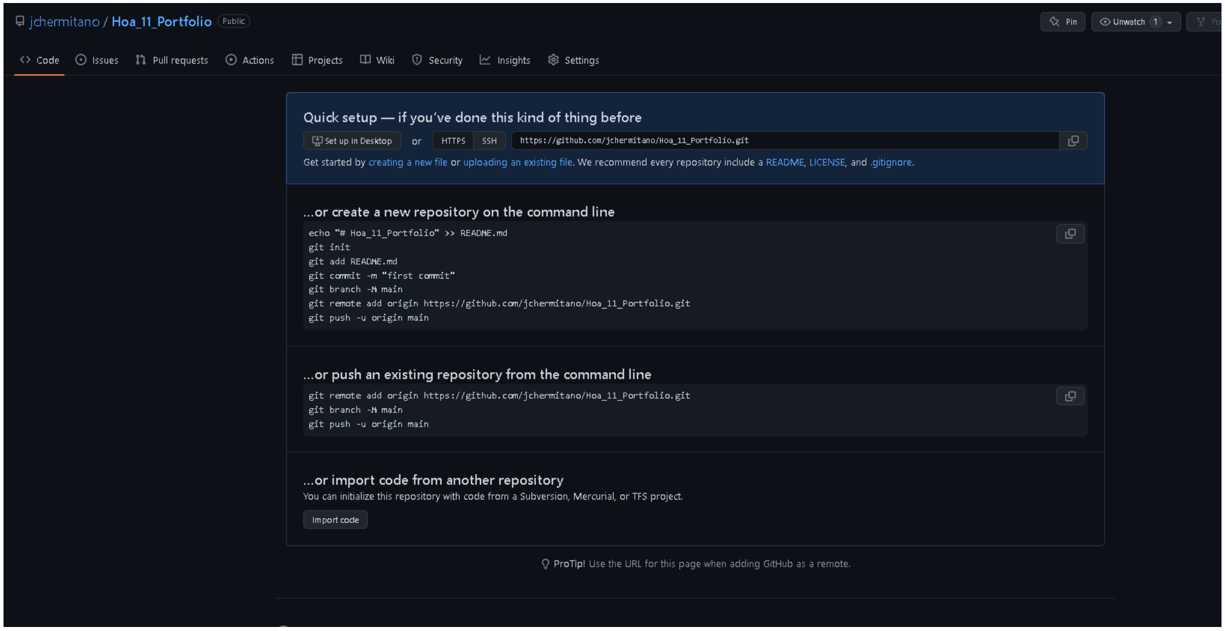
Source: https://docs.microsoft.com/en-us/virtualization/windowscontainers/about/containers-vs-vm

3. Tasks

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

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

Step 1: Create a new repository in your choice title for this activity.



Quick setup — if you've done this kind of thing before

```
https://github.com/jchermitano/Hoa_11_Portfolio.git
```

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

```
echo "# Hoa_11_Portfolio" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/jchermitano/Hoa_11_Portfolio.git
git push -u origin main
```

...or push an existing repository from the command line

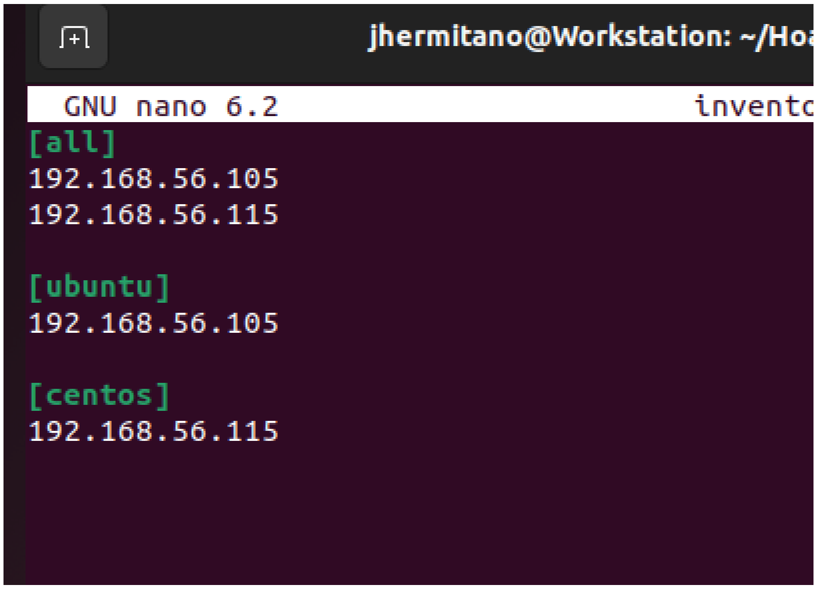
```
git remote add origin https://github.com/jchermitano/Hoa_11_Portfolio.git
git branch -M main
git push -u origin main
```

...or import code from another repository

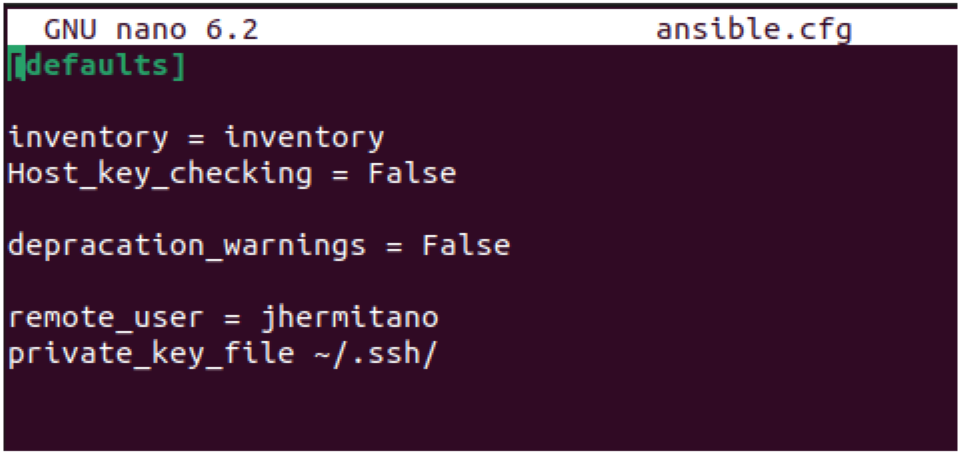
You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

ProTip! Use the URL for this page when adding GitHub as a remote.

Step 2: Inside your repository, create your nano inventory and ansible.cfg for your playbook.

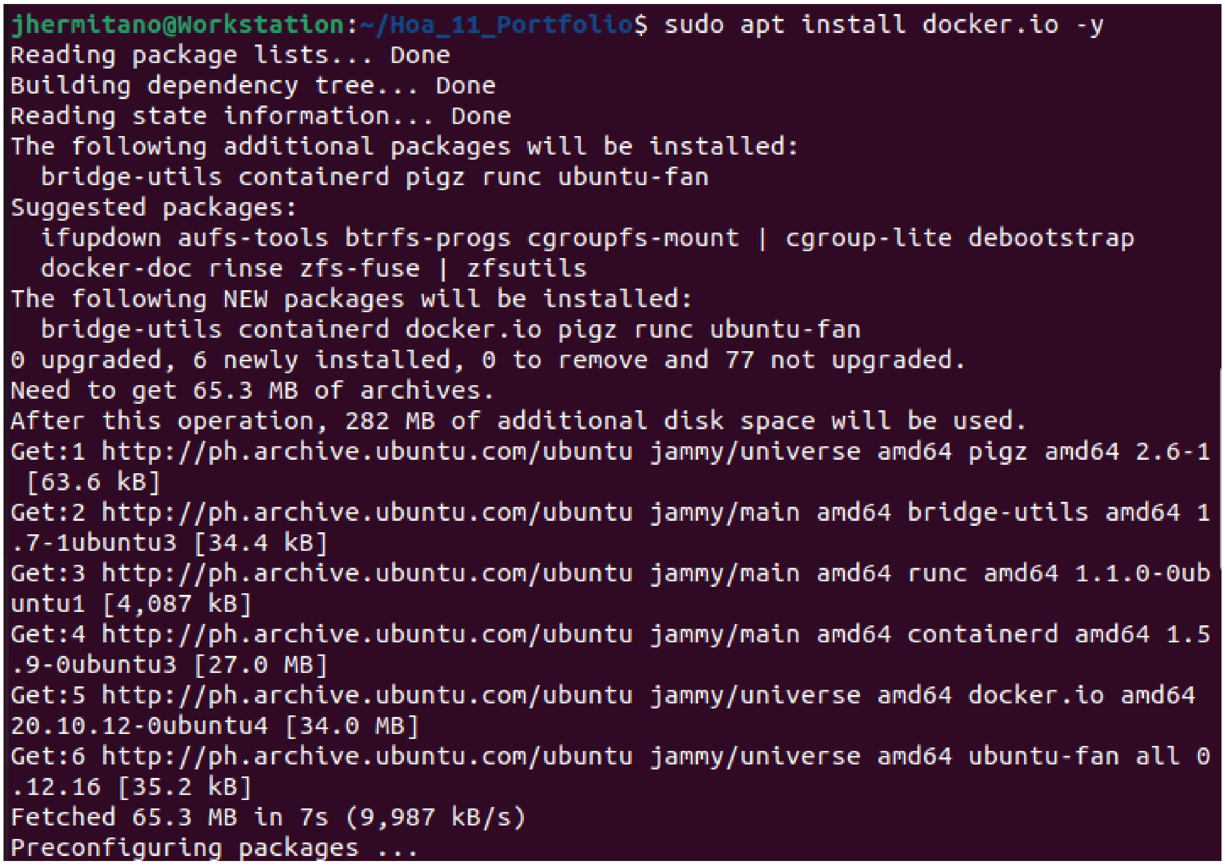


```
  GNU nano 6.2                          invento
[all]
192.168.56.105
192.168.56.115

[ubuntu]
192.168.56.105

[centos]
192.168.56.115
```



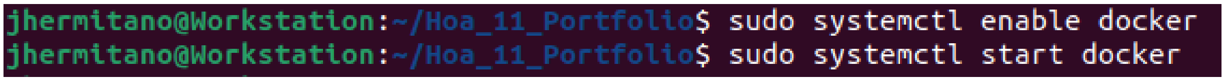
```
  GNU nano 6.2                          ansible.cfg
[defaults]

inventory = inventory
Host_key_checking = False

depracation_warnings = False

remote_user = jhermitano
private_key_file ~/.ssh/
```

Step 3: Install Docker and enable the docker socket.



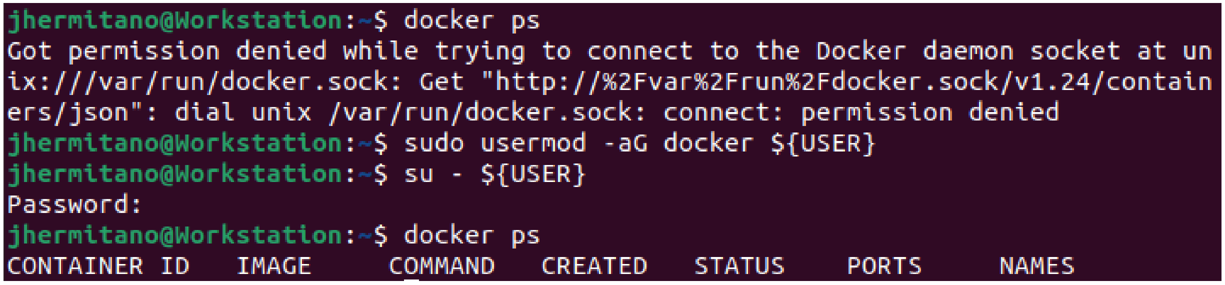
```
jhermitano@Workstation:~/Hoa_11_Portfolio$ sudo apt install docker.io -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd pigz runc ubuntu-fan
Suggested packages:
  ifupdown 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 77 not upgraded.
Need to get 65.3 MB of archives.
After this operation, 282 MB of additional disk space will be used.
Get:1 http://ph.archive.ubuntu.com/ubuntu jammy/universe amd64 pigz amd64 2.6-1
 [63.6 kB]
Get:2 http://ph.archive.ubuntu.com/ubuntu jammy/main amd64 bridge-utils amd64 1
.7-1ubuntu3 [34.4 kB]
Get:3 http://ph.archive.ubuntu.com/ubuntu jammy/main amd64 runc amd64 1.1.0-0ub
untu1 [4,087 kB]
Get:4 http://ph.archive.ubuntu.com/ubuntu jammy/main amd64 containerd amd64 1.5
.9-0ubuntu3 [27.0 MB]
Get:5 http://ph.archive.ubuntu.com/ubuntu jammy/universe amd64 docker.io amd64
20.10.12-0ubuntu4 [34.0 MB]
Get:6 http://ph.archive.ubuntu.com/ubuntu jammy/universe amd64 ubuntu-fan all 0
.12.16 [35.2 kB]
Fetched 65.3 MB in 7s (9,987 kB/s)
Preconfiguring packages ...
```

To install docker, enter the command: sudo apt install docker.io.



```
jhermitano@Workstation:~/Hoa_11_Portfolio$ sudo systemctl enable docker
jhermitano@Workstation:~/Hoa_11_Portfolio$ sudo systemctl start docker
```

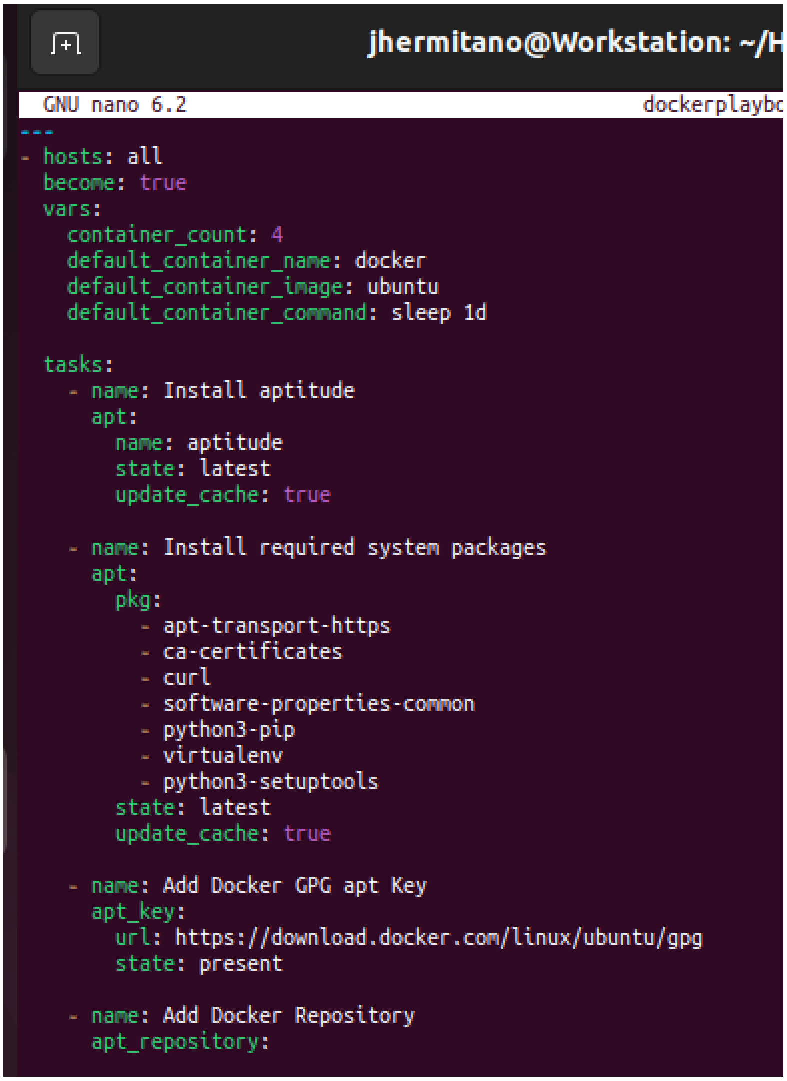
These commands is used to enable the docker to the user and start it.



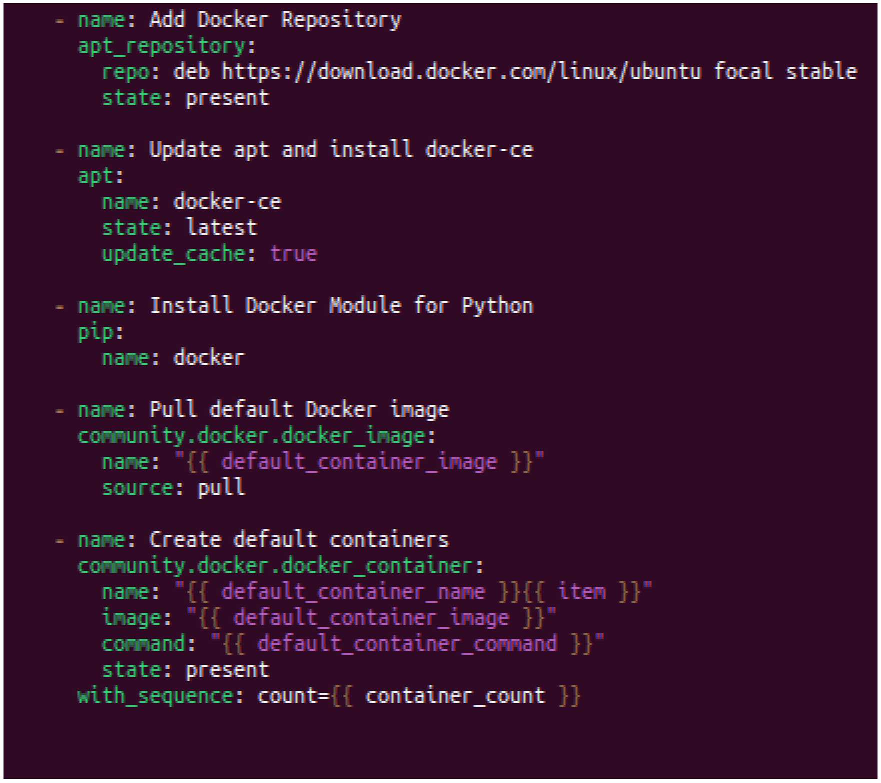
```
jhermitano@Workstation:~$ docker ps
Got permission denied while trying to connect to the Docker daemon socket at un
ix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/contain
ers/json": dial unix /var/run/docker.sock: connect: permission denied
jhermitano@Workstation:~$ sudo usermod -aG docker ${USER}
jhermitano@Workstation:~$ su - ${USER}
Password:
jhermitano@Workstation:~$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```

If you encountered this permission denied, follow the commands below it.

Step 4: Create a playbook that Install and build the Dockerfile to the server. Copy and follow the commands.

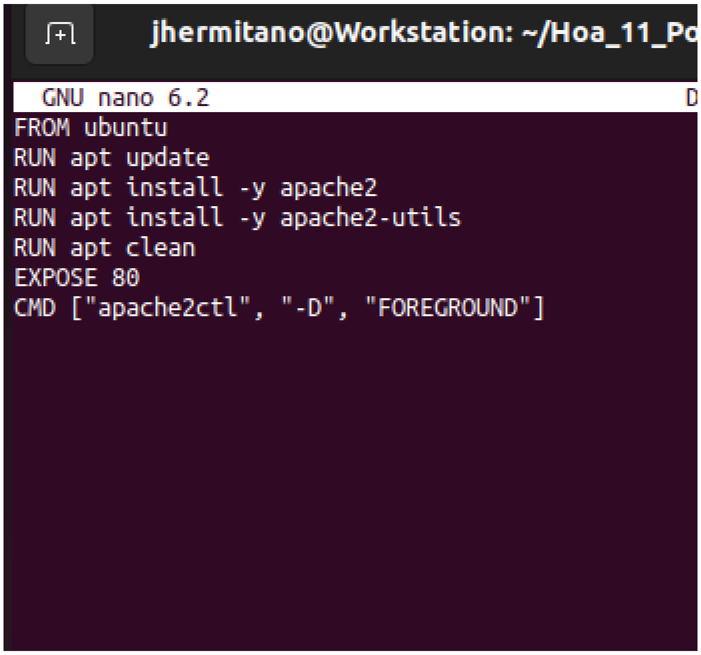


```
  GNU nano 6.2                                    dockerplayb
---
- hosts: all
  become: true
  vars:
    container_count: 4
    default_container_name: docker
    default_container_image: ubuntu
    default_container_command: sleep 1d

  tasks:
    - name: Install aptitude
      apt:
        name: aptitude
        state: latest
        update_cache: true

    - name: Install required system packages
      apt:
        pkg:
          - apt-transport-https
          - ca-certificates
          - curl
          - software-properties-common
          - python3-pip
          - virtualenv
          - python3-setuptools
        state: latest
        update_cache: true

    - name: Add Docker GPG apt Key
      apt_key:
        url: https://download.docker.com/linux/ubuntu/gpg
        state: present

    - name: Add Docker Repository
      apt_repository:
```



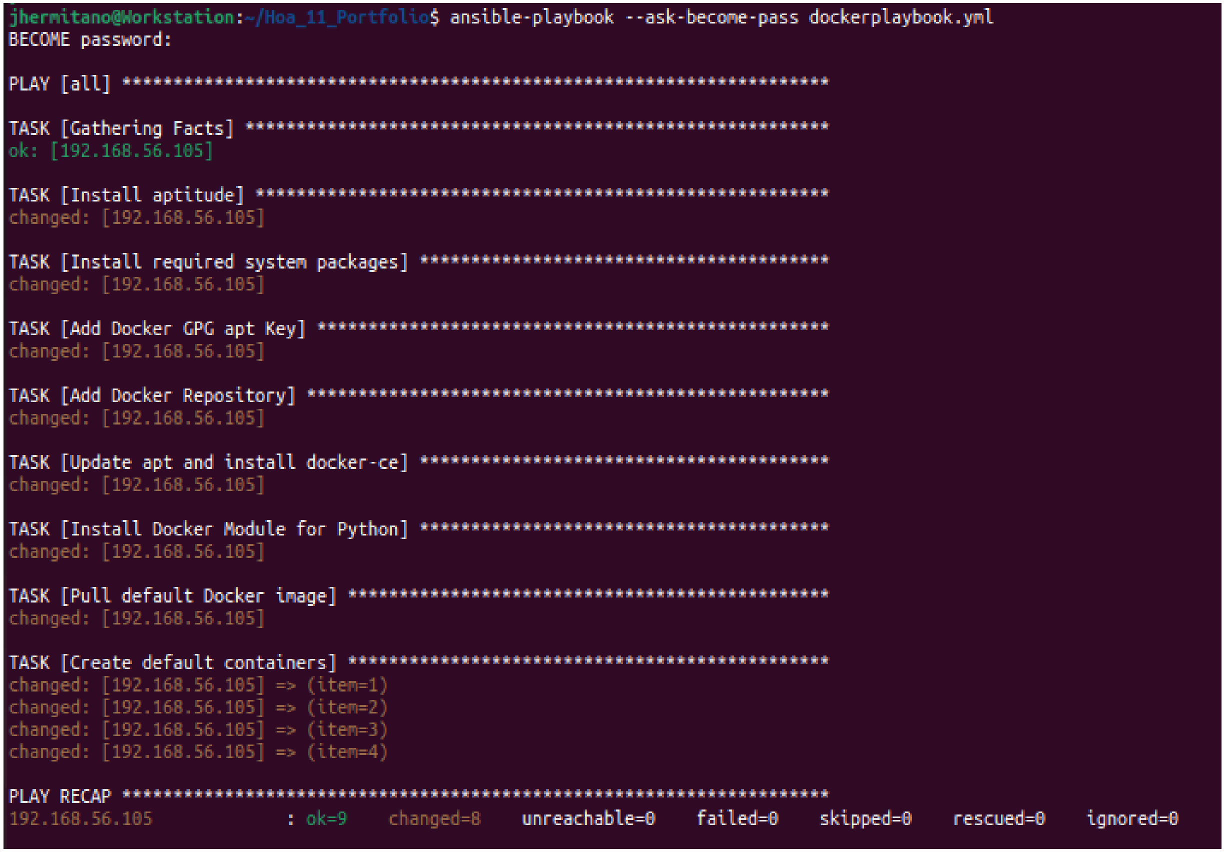
```
    - name: Add Docker Repository
      apt_repository:
        repo: deb https://download.docker.com/linux/ubuntu focal stable
        state: present

    - name: Update apt and install docker-ce
      apt:
        name: docker-ce
        state: latest
        update_cache: true

    - name: Install Docker Module for Python
      pip:
        name: docker

    - name: Pull default Docker image
      community.docker.docker_image:
        name: "{{ default_container_image }}"
        source: pull

    - name: Create default containers
      community.docker.docker_container:
        name: "{{ default_container_name }}{{ item }}"
        image: "{{ default_container_image }}"
        command: "{{ default_container_command }}"
        state: present
      with_sequence: count={{ container_count }}
```

Step 5: Create a Dockerfile to install web and DB servers.



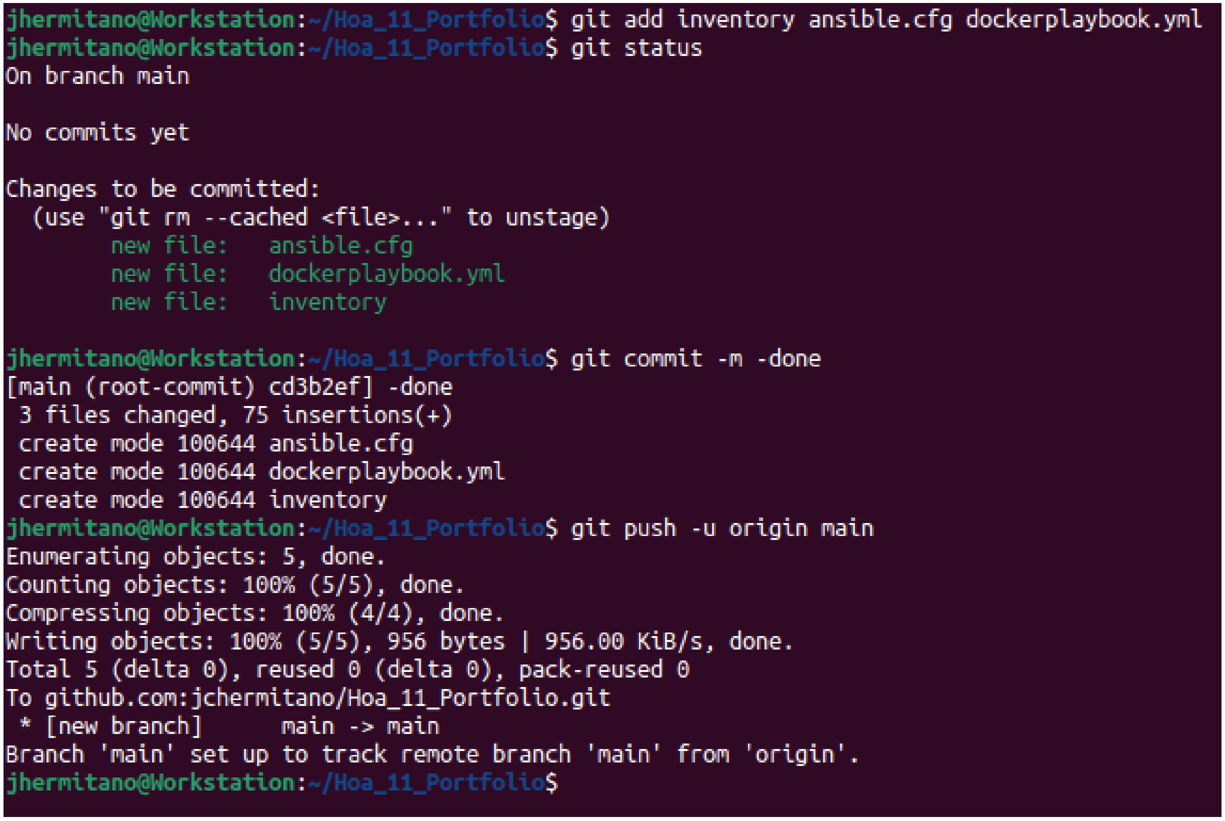
```
  GNU nano 6.2                                          D
FROM ubuntu
RUN apt update
RUN apt install -y apache2
RUN apt install -y apache2-utils
RUN apt clean
EXPOSE 80
CMD ["apache2ctl", "-D", "FOREGROUND"]
```

Step 6: Run the playbook to see if it is fully working and there are no errors in it.

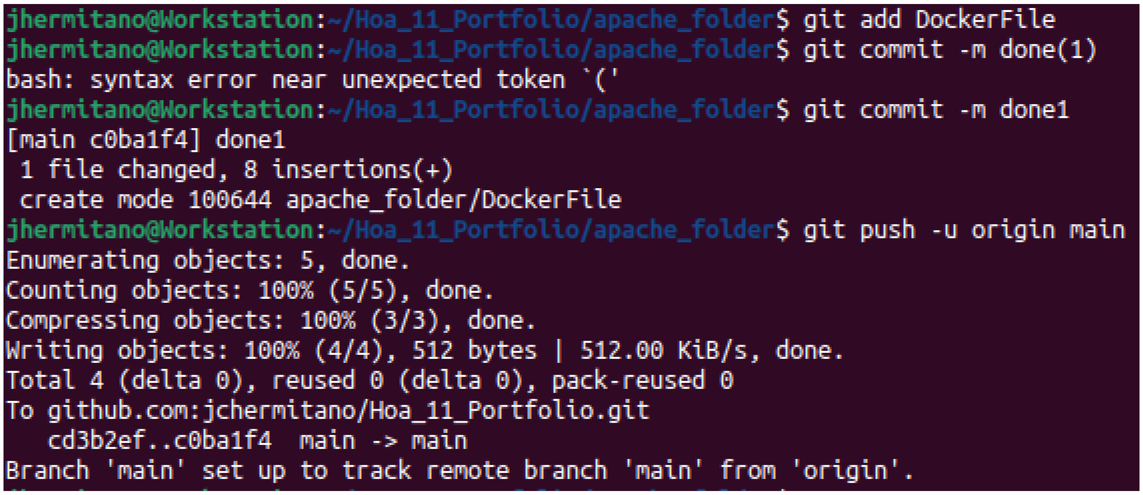


```
jhermitano@Workstation:~/Hoa_11_Portfolio$ ansible-playbook --ask-become-pass dockerplaybook.yml
BECOME password:

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

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

TASK [Install aptitude] ********************************************************
changed: [192.168.56.105]

TASK [Install required system packages] ****************************************
changed: [192.168.56.105]

TASK [Add Docker GPG apt Key] **************************************************
changed: [192.168.56.105]

TASK [Add Docker Repository] ***************************************************
changed: [192.168.56.105]

TASK [Update apt and install docker-ce] ****************************************
changed: [192.168.56.105]

TASK [Install Docker Module for Python] ****************************************
changed: [192.168.56.105]

TASK [Pull default Docker image] ***********************************************
changed: [192.168.56.105]

TASK [Create default containers] ***********************************************
changed: [192.168.56.105] => (item=1)
changed: [192.168.56.105] => (item=2)
changed: [192.168.56.105] => (item=3)
changed: [192.168.56.105] => (item=4)

PLAY RECAP *********************************************************************
192.168.56.105             : ok=9    changed=8    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Step 7: Add, commit and push it to your repository.

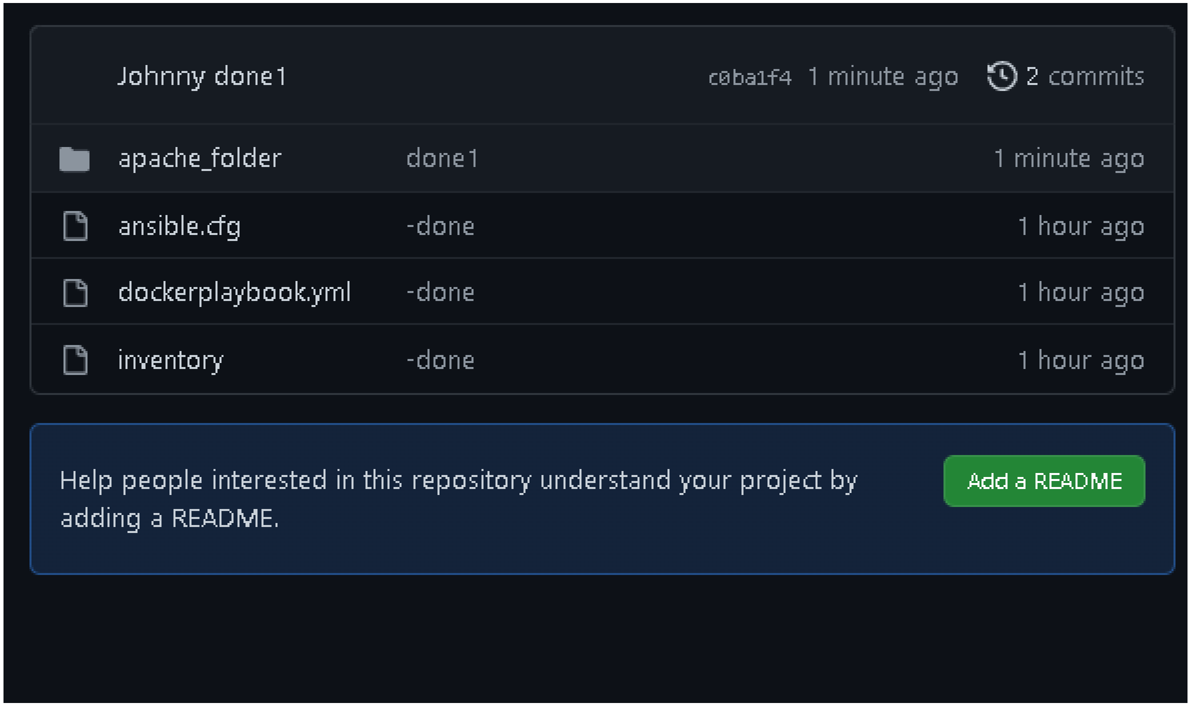


```
jhermitano@Workstation:~/Hoa_11_Portfolio$ git add inventory ansible.cfg dockerplaybook.yml
jhermitano@Workstation:~/Hoa_11_Portfolio$ git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   ansible.cfg
        new file:   dockerplaybook.yml
        new file:   inventory

jhermitano@Workstation:~/Hoa_11_Portfolio$ git commit -m -done
[main (root-commit) cd3b2ef] -done
 3 files changed, 75 insertions(+)
 create mode 100644 ansible.cfg
 create mode 100644 dockerplaybook.yml
 create mode 100644 inventory
jhermitano@Workstation:~/Hoa_11_Portfolio$ git push -u origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 956 bytes | 956.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:jchermitano/Hoa_11_Portfolio.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
jhermitano@Workstation:~/Hoa_11_Portfolio$
```



```
jhermitano@Workstation:~/Hoa_11_Portfolio/apache_folder$ git add DockerFile
jhermitano@Workstation:~/Hoa_11_Portfolio/apache_folder$ git commit -m done(1)
bash: syntax error near unexpected token `('
jhermitano@Workstation:~/Hoa_11_Portfolio/apache_folder$ git commit -m done1
[main c0ba1f4] done1
 1 file changed, 8 insertions(+)
 create mode 100644 apache_folder/DockerFile
jhermitano@Workstation:~/Hoa_11_Portfolio/apache_folder$ git push -u origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 512 bytes | 512.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:jchermitano/Hoa_11_Portfolio.git
   cd3b2ef..c0ba1f4  main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

Step 8: Check your repository through your github account.



Johnny done1 c0ba1f4 1 minute ago 2 commits

apache_folder	done1	1 minute ago
ansible.cfg	-done	1 hour ago
dockerplaybook.yml	-done	1 hour ago
inventory	-done	1 hour ago

Help people interested in this repository understand your project by adding a README.

Reflections:

Answer the following:

1. What are the benefits of implementing containerizations?

Containerization enables faster and more secure application development and deployment. Using conventional techniques, code is created in a particular computing environment, and when it is transferred to another location, it frequently has faults and errors.

Conclusions:

In this activity, I was able to accomplish the said tasks. But it wasn't that easy for me to achieve this completed activity. The most part that I literally put some time on is the part of making a playbook and correcting every error in it.

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