Name: Jefferson Langbid	**Date Performed:**
Course/Section: CPE 232 - CPE31S23	**Date Submitted:**
Instructor: Dr. Taylar	**Semester and SY:**

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.

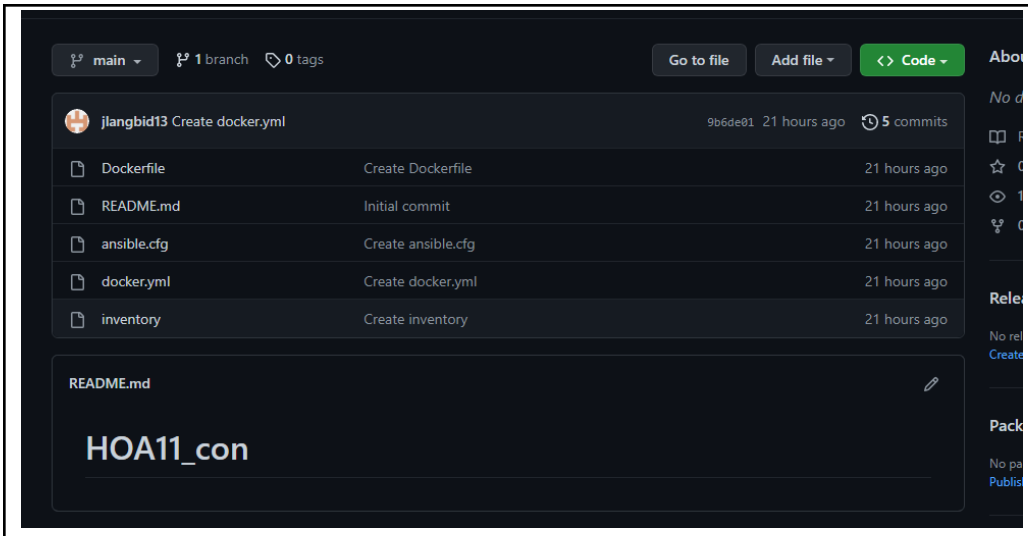
Source: https://docs.microsoft.com/en-us/virtualization/windowscontainers/about/containers-vs-vm

3. Tasks

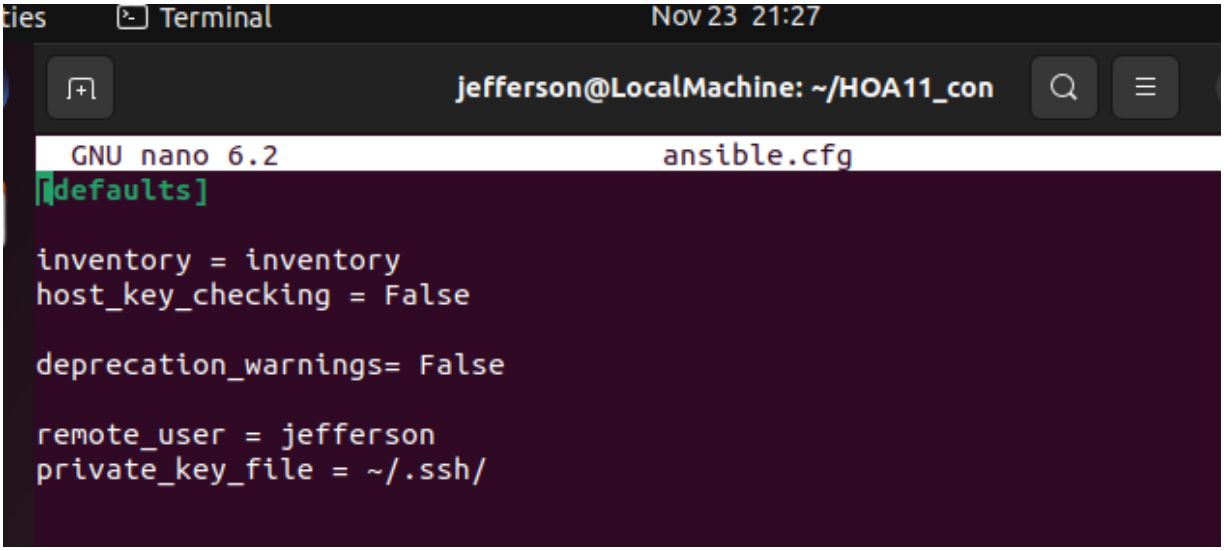
1. Create a new repository for this activity.
2. Install Docker and enable the docker socket.
3. Add to Docker group to your current user.
4. Create a Dockerfile to install web and DB server.
5. Install and build the Dockerfile using Ansible.
6. Add, commit and push it to your repository.

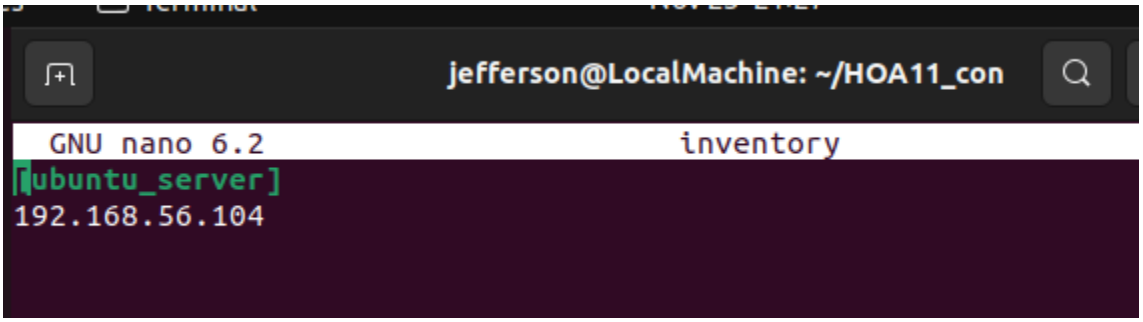
4. Output (screenshots and explanations)

- Create a new repository for this activity.



Creating ansible.cfg and inventory





- Install Docker and enable the docker socket.

```
jefferson@LocalMachine:~$ sudo apt install docker.io
[sudo] password for jefferson:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required
  libflashrom1 libftdi1-2 linux-headers-5.15.0-50
  linux-headers-5.15.0-50-generic linux-image-5.15.0-50-generic
  linux-modules-5.15.0-50-generic linux-modules-extra-5.15.0-50-generic
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 5 not upgraded.
```

```
Processing triggers for man-db (2.10.2-1) ...
jefferson@LocalMachine:~$ sudo systemctl start docker
jefferson@LocalMachine:~$ sudo systemctl enable docker
jefferson@LocalMachine:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor prese>
     Active: active (running) since Wed 2022-11-23 20:17:27 PST; 42s ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 36855 (dockerd)
      Tasks: 8
     Memory: 38.4M
        CPU: 399ms
     CGroup: /system.slice/docker.service
             └─36855 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/co>
```

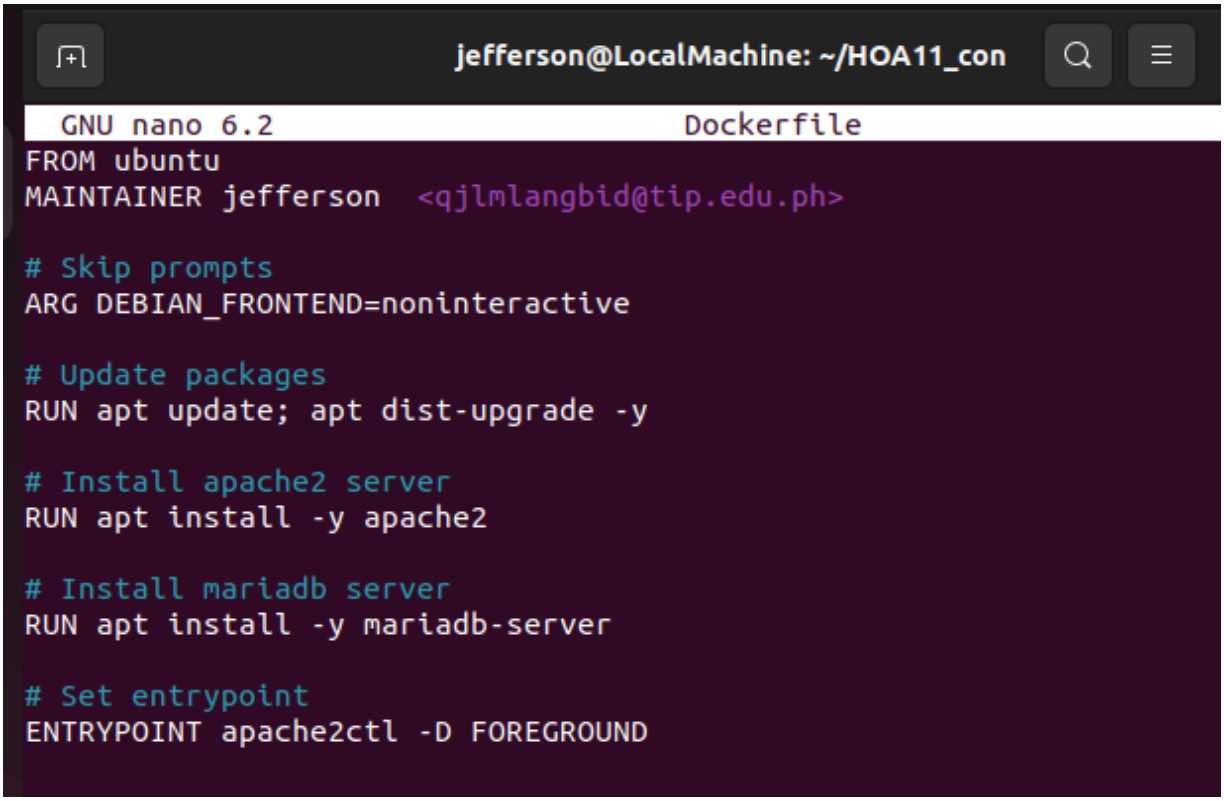
Docker is now installed

- Add to Docker group to your current user.

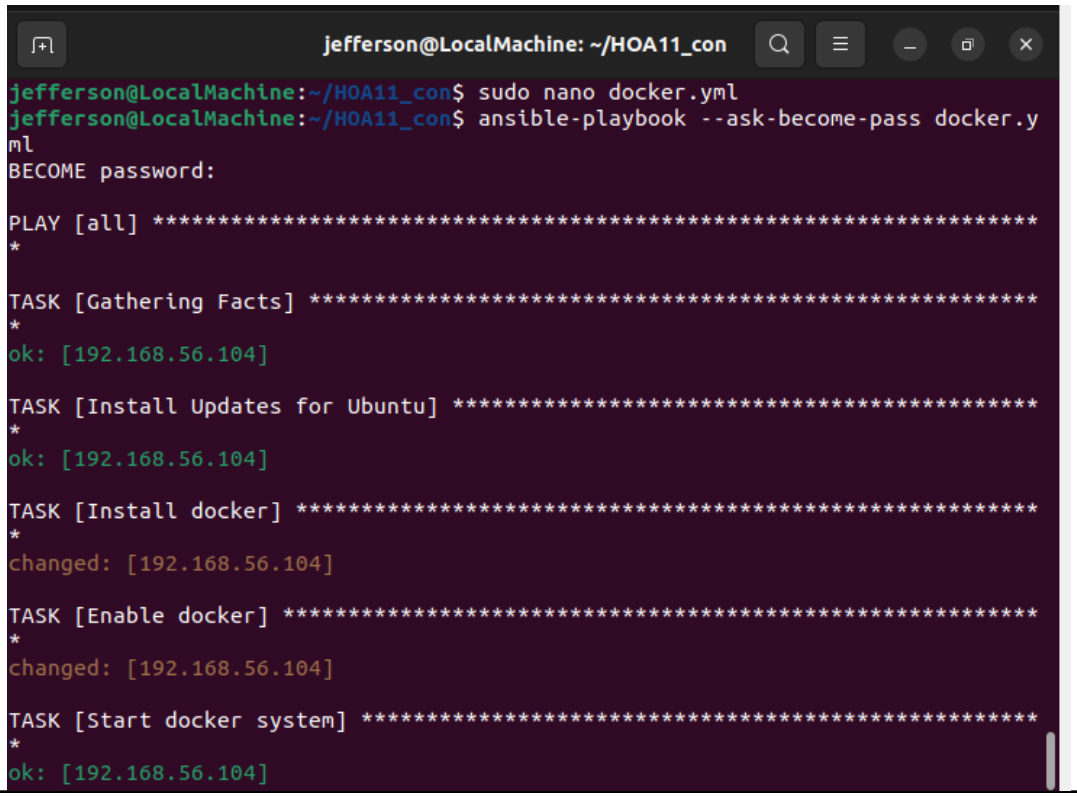
```
jefferson@LocalMachine:~/HOA11_con$ sudo usermod -aG docker jefferson
jefferson@LocalMachine:~/HOA11_con$ grep docker /etc/group
docker:x:137:jefferson
```

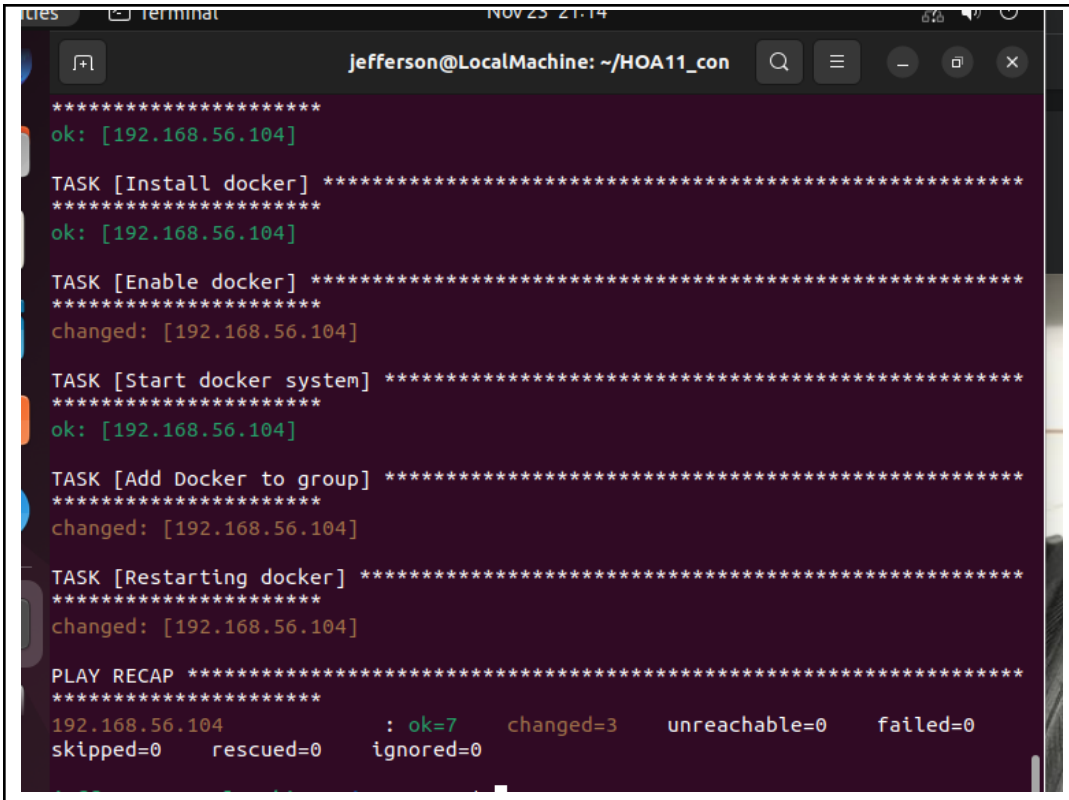
Adding group to docker

- Create a Dockerfile to install web and DB server.

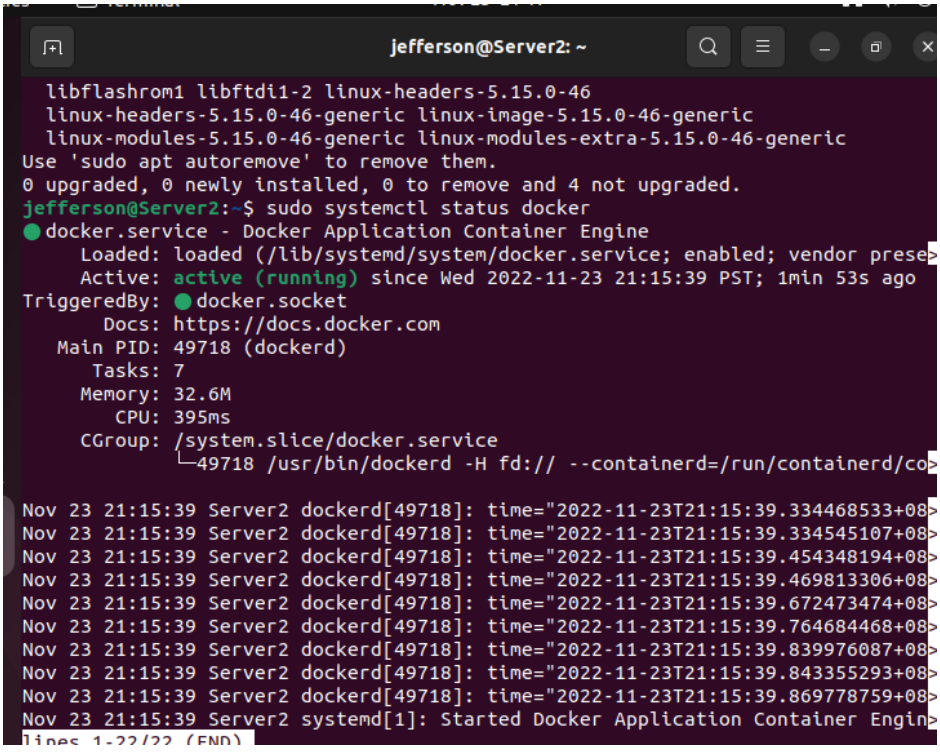


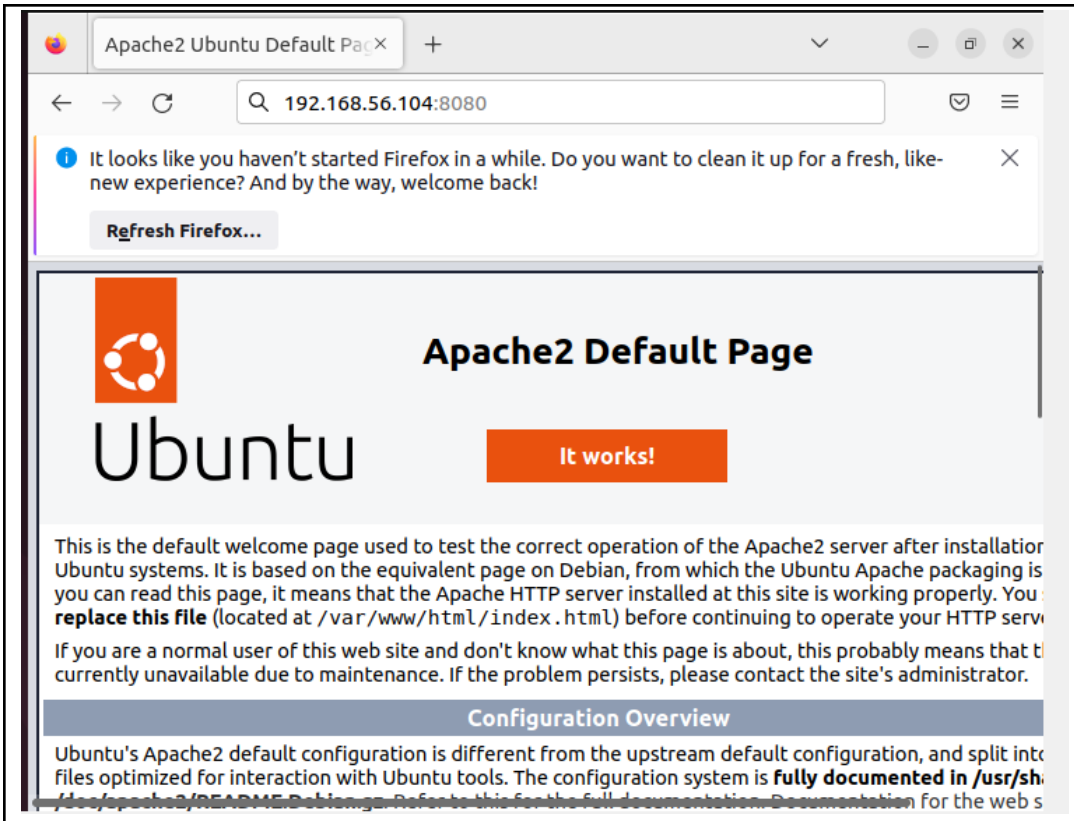
- Install and build the Dockerfile using Ansible.

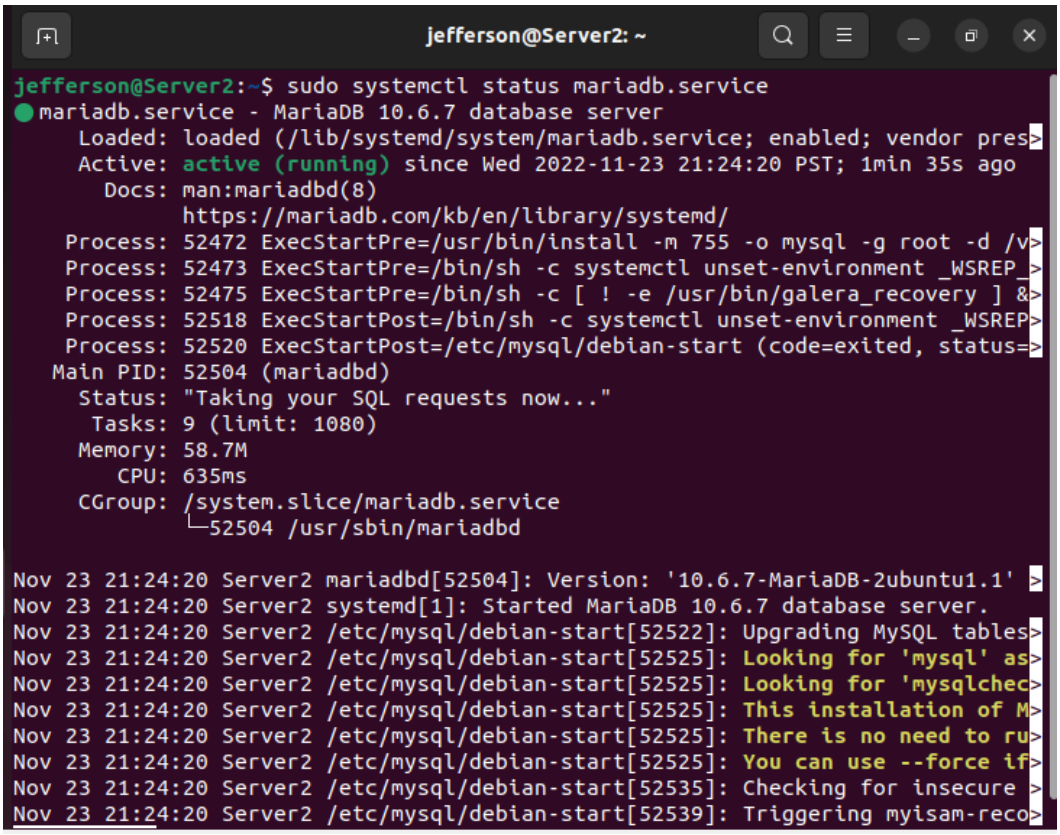


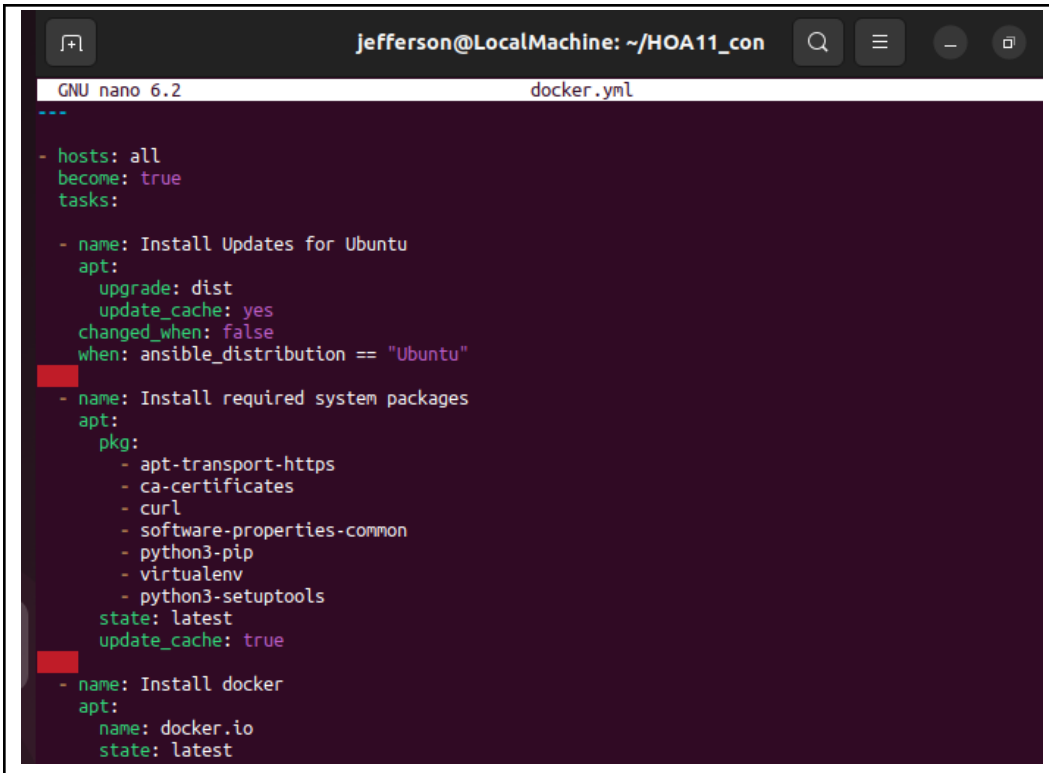


Installing docker and apache and mariadb using docker.

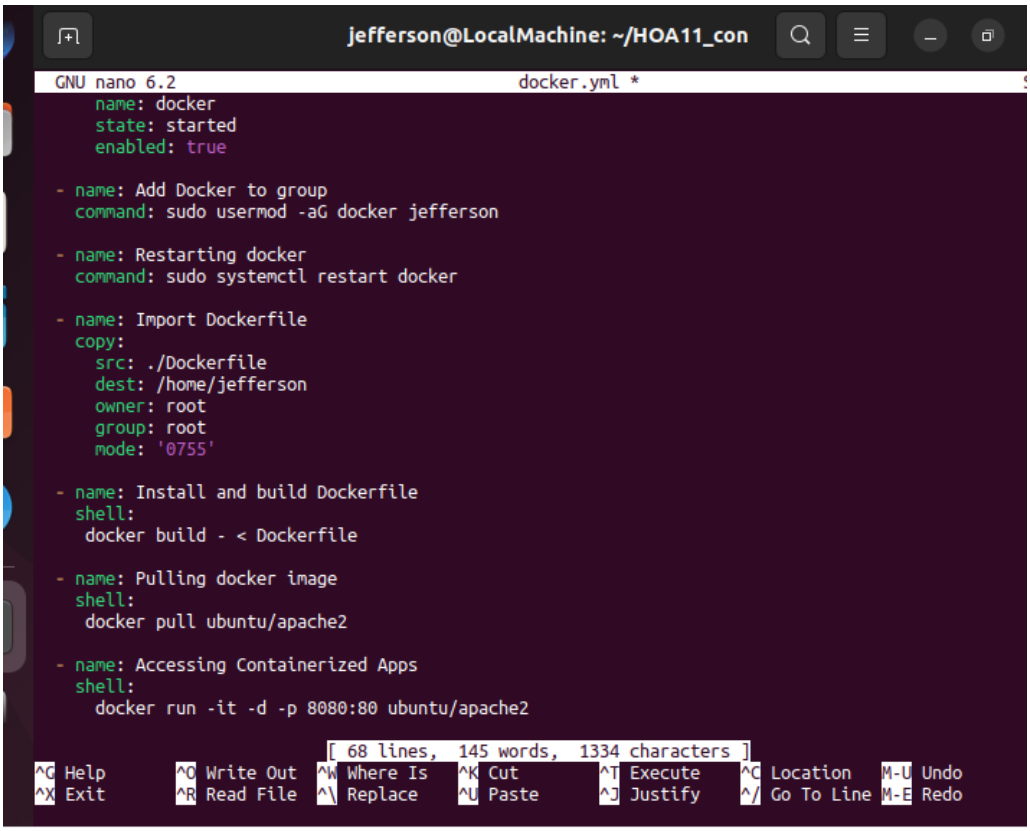



Apache2 Ubuntu Default Pa
192.168.56.104:8080
It looks like you haven't started Firefox in a while. Do you want to clean it up for a fresh, like-new experience? And by the way, welcome back!
Refresh Firefox...
Apache2 Default Page
Ubuntu
It works!
This is the default welcome page used to test the correct operation of the Apache2 server after installation
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 into
files optimized for interaction with Ubuntu tools. The configuration system is fully documented in /usr/sh
for the web s

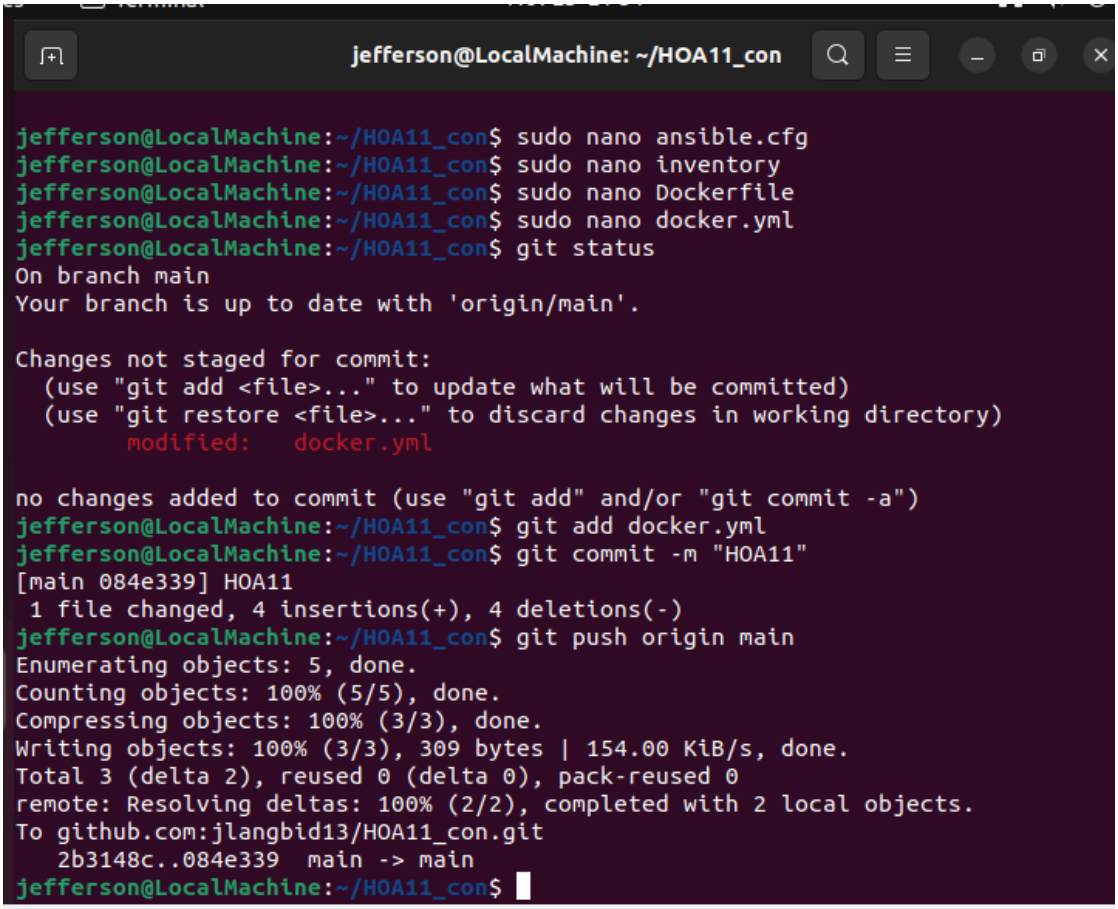

jefferson@Server2: ~
jefferson@Server2:~$ sudo systemctl status mariadb.service
mariadb.service - MariaDB 10.6.7 database server
Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor pres
Active: active (running) since Wed 2022-11-23 21:24:20 PST; 1min 35s ago
Docs: man:mariadbd(8)
https://mariadb.com/kb/en/library/systemd/
Process: 52472 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /v
Process: 52473 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_
Process: 52475 ExecStartPre=/bin/sh -c [! -e /usr/bin/galera_recovery] &
Process: 52518 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP
Process: 52520 ExecStartPost=/etc/mysql/debian-start (code=exited, status=
Main PID: 52504 (mariadbd)
Status: "Taking your SQL requests now..."
Tasks: 9 (limit: 1080)
Memory: 58.7M
CPU: 635ms
CGroup: /system.slice/mariadb.service
└─52504 /usr/sbin/mariadbd
Nov 23 21:24:20 Server2 mariadbd[52504]: Version: '10.6.7-MariaDB-2ubuntu1.1'
Nov 23 21:24:20 Server2 systemd[1]: Started MariaDB 10.6.7 database server.
Nov 23 21:24:20 Server2 /etc/mysql/debian-start[52522]: Upgrading MySQL tables
Nov 23 21:24:20 Server2 /etc/mysql/debian-start[52525]: Looking for 'mysql' as
Nov 23 21:24:20 Server2 /etc/mysql/debian-start[52525]: Looking for 'mysqlchec
Nov 23 21:24:20 Server2 /etc/mysql/debian-start[52525]: This installation of M
Nov 23 21:24:20 Server2 /etc/mysql/debian-start[52525]: There is no need to ru
Nov 23 21:24:20 Server2 /etc/mysql/debian-start[52525]: You can use --force if
Nov 23 21:24:20 Server2 /etc/mysql/debian-start[52535]: Checking for insecure
Nov 23 21:24:20 Server2 /etc/mysql/debian-start[52539]: Triggering myisam-reco

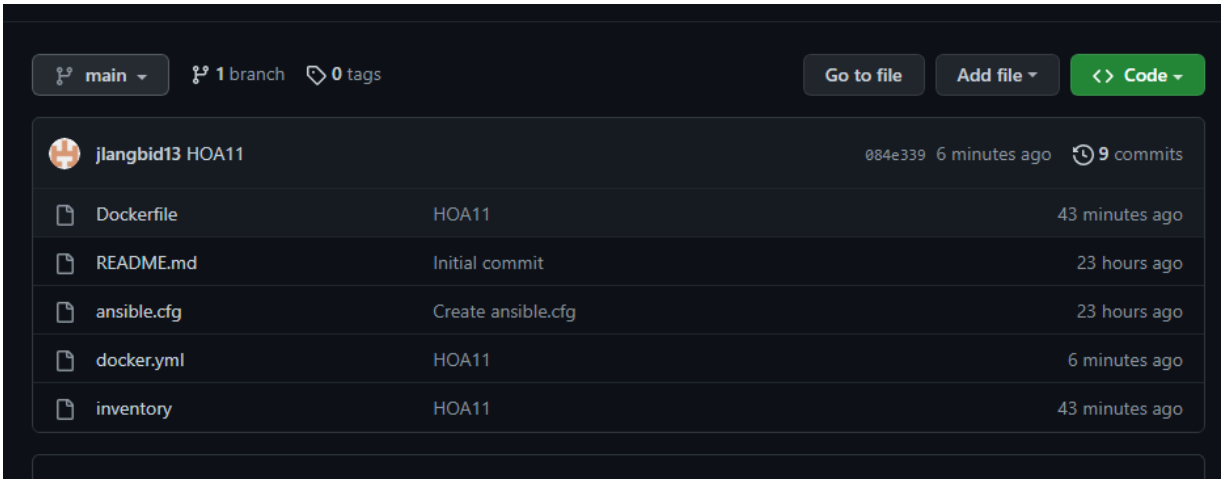

jefferson@LocalMachine: ~/HOA11_con
GNU nano 6.2 docker.yml

- hosts: all
become: true
tasks:
- name: Install Updates for Ubuntu
apt:
upgrade: dist
update_cache: yes
changed_when: false
when: ansible_distribution == "Ubuntu"
- 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: Install docker
apt:
name: docker.io
state: latest


jefferson@LocalMachine: ~/HOA11_con
GNU nano 6.2 docker.yml *
name: docker
state: started
enabled: true
- name: Add Docker to group
command: sudo usermod -aG docker jefferson
- name: Restarting docker
command: sudo systemctl restart docker
- name: Import Dockerfile
copy:
src: ./Dockerfile
dest: /home/jefferson
owner: root
group: root
mode: '0755'
- name: Install and build Dockerfile
shell:
docker build - < Dockerfile
- name: Pulling docker image
shell:
docker pull ubuntu/apache2
- name: Accessing Containerized Apps
shell:
docker run -it -d -p 8080:80 ubuntu/apache2
[68 lines, 145 words, 1334 characters]
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo

- Add, commit and push it to your repository.





Reflections:

Answer the following:

1. What are the benefits of implementing containerizations?
   The benefits of implementing containerization is to allow users to update and install applications everywhere by using containerization just like ansible but it is more efficient than ansible due to it can be used even without a server.

Conclusions:

All in all, I first created a repository in github and then cloned it to my local machine. I created the ansible and inventory file to the directory and then the yml to run the code in ansible. I also installed docker.io and activated the socket in the local machine and created a new group for the docker and new user. The next thing I did is to create a code for the ansible to install, enable the socket of docker and to build a docker on the other server and also to install the apache2 and mariadb using the dockerfile. Lastly I tried the apache and mariadb to know if it is already installed and then added the files into my github and committed it to my repository.