Name: Jaira Biane Maculada	**Date Performed:**11/16/23
Course/Section:CPE232/CPE31S6	**Date Submitted:**11/16/23
Instructor: Dr. Jonathan V. Taylar	**Semester and SY:** 1st Sem(2023-2024)

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

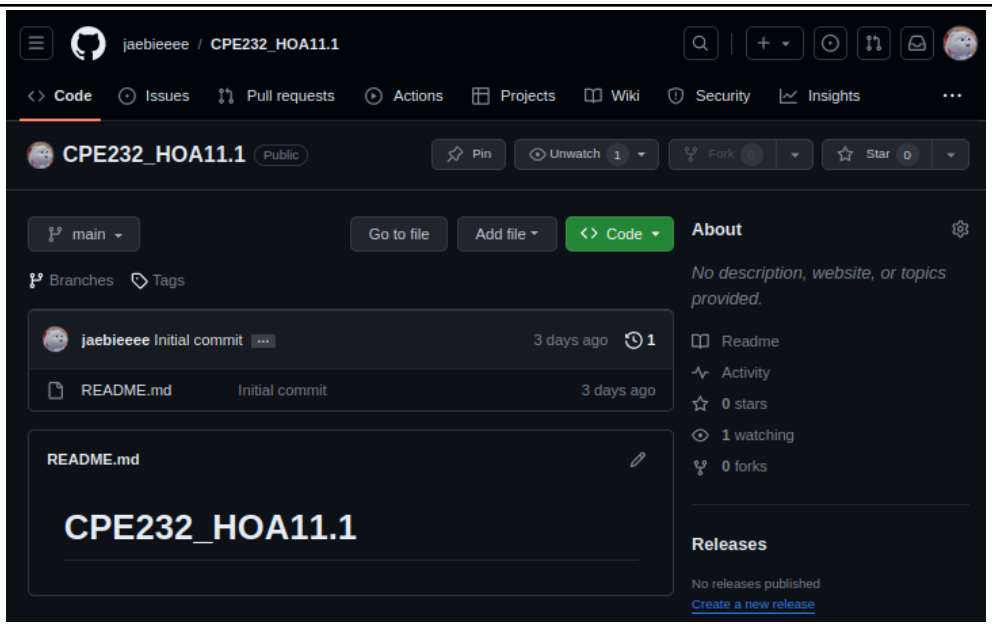
3. Tasks

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.

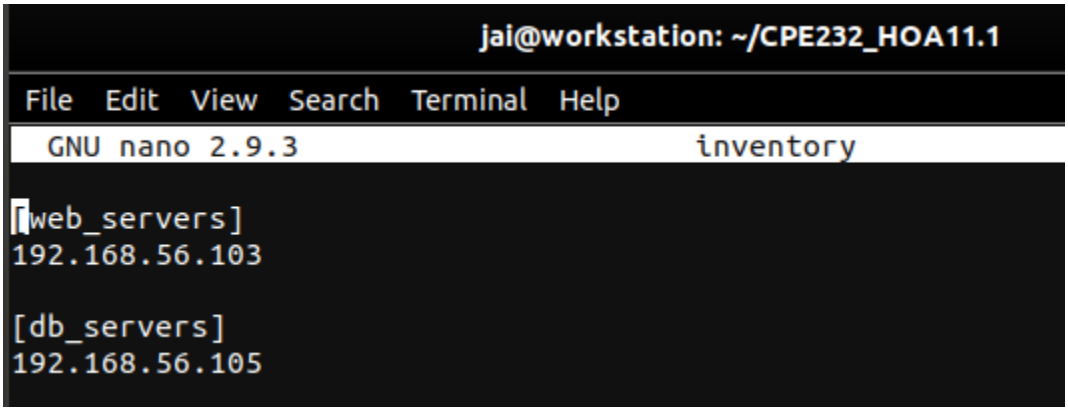
4. Output (screenshots and explanations)

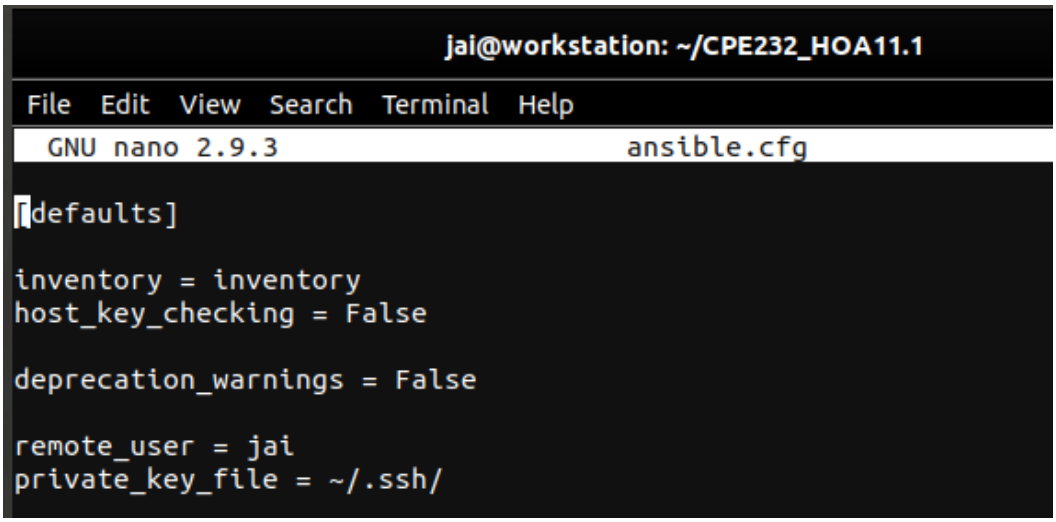
Task 1: Create a File

1. Create a new repository for this Hands-On Activity.



2. Create the ansible.cfg and inventory file (*must include one Ubuntu and CentOS*)





Task 2: Install and Create a dockerfile

1. Install the docker.io in the local machine.

```
jai@workstation:~/CPE232_HOA11.1$ sudo apt install docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
docker.io is already the newest version (20.10.21-0ubuntu1~18.04.3).
The following package was automatically installed and is no longer required:
  libllvm7
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

2. Show that the docker is now working in the local machine.

```
jai@workstation:~/CPE232_HOA11.1$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset:
   Active: active (running) since Thu 2023-11-16 16:10:28 PST; 25min ago
     Docs: https://docs.docker.com
 Main PID: 1047 (dockerd)
    Tasks: 15
   CGroup: /system.slice/docker.service
           └─1047 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/contai

Nov 16 16:10:27 workstation dockerd[1047]: time="2023-11-16T16:10:27.910452764+
Nov 16 16:10:27 workstation dockerd[1047]: time="2023-11-16T16:10:27.919761604+
Nov 16 16:10:28 workstation dockerd[1047]: time="2023-11-16T16:10:28.113687391+
Nov 16 16:10:28 workstation dockerd[1047]: time="2023-11-16T16:10:28.156654172+
Nov 16 16:10:28 workstation dockerd[1047]: time="2023-11-16T16:10:28.229878229+
Nov 16 16:10:28 workstation dockerd[1047]: time="2023-11-16T16:10:28.487337885+
Nov 16 16:10:28 workstation dockerd[1047]: time="2023-11-16T16:10:28.708275121+
Nov 16 16:10:28 workstation dockerd[1047]: time="2023-11-16T16:10:28.709976503+
Nov 16 16:10:28 workstation systemd[1]: Started Docker Application Container En
Nov 16 16:10:28 workstation dockerd[1047]: time="2023-11-16T16:10:28.785157532+

[1]+  Stopped                 sudo systemctl status docker
```

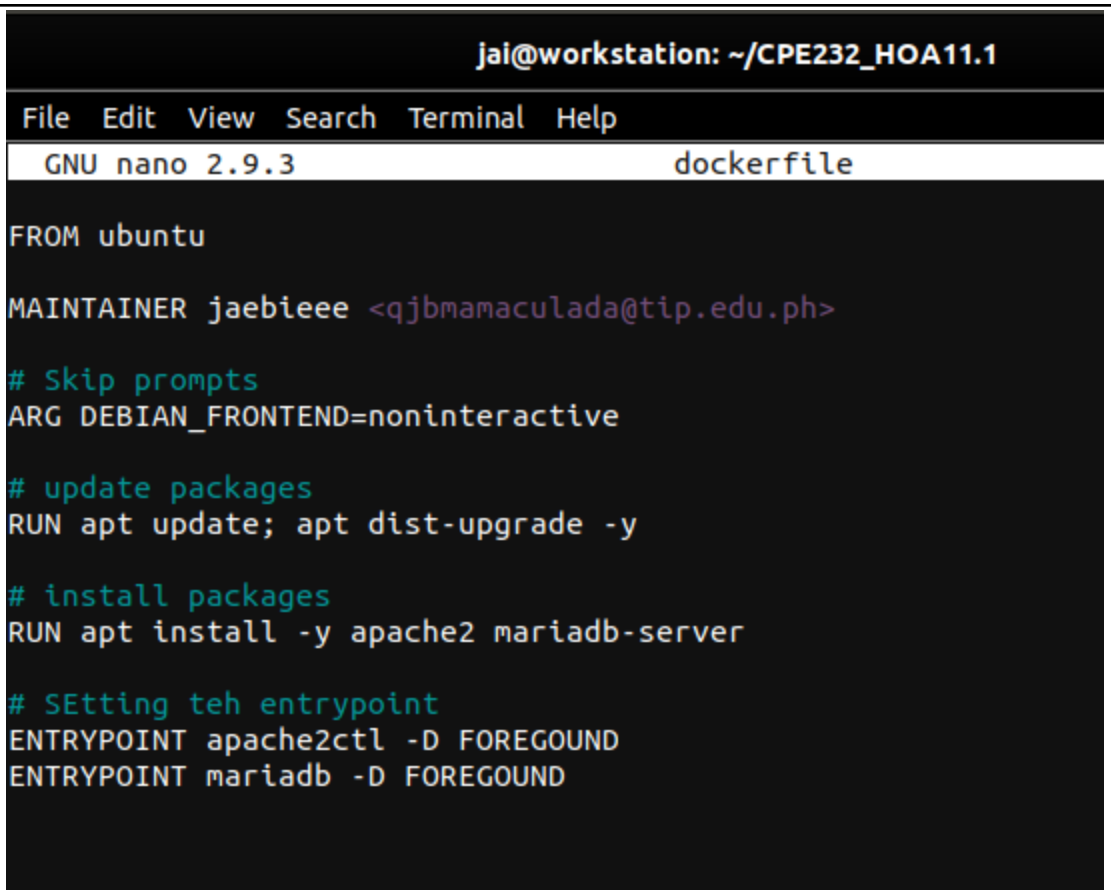
3. Enable the docker in the local machine.

```
jai@workstation:~/CPE232_HOA11.1$ sudo systemctl enable docker
```

4. Start the docker in the local machine.

```
jai@workstation:~/CPE232_HOA11.1$ sudo systemctl start docker
jai@workstation:~/CPE232_HOA11.1$
```

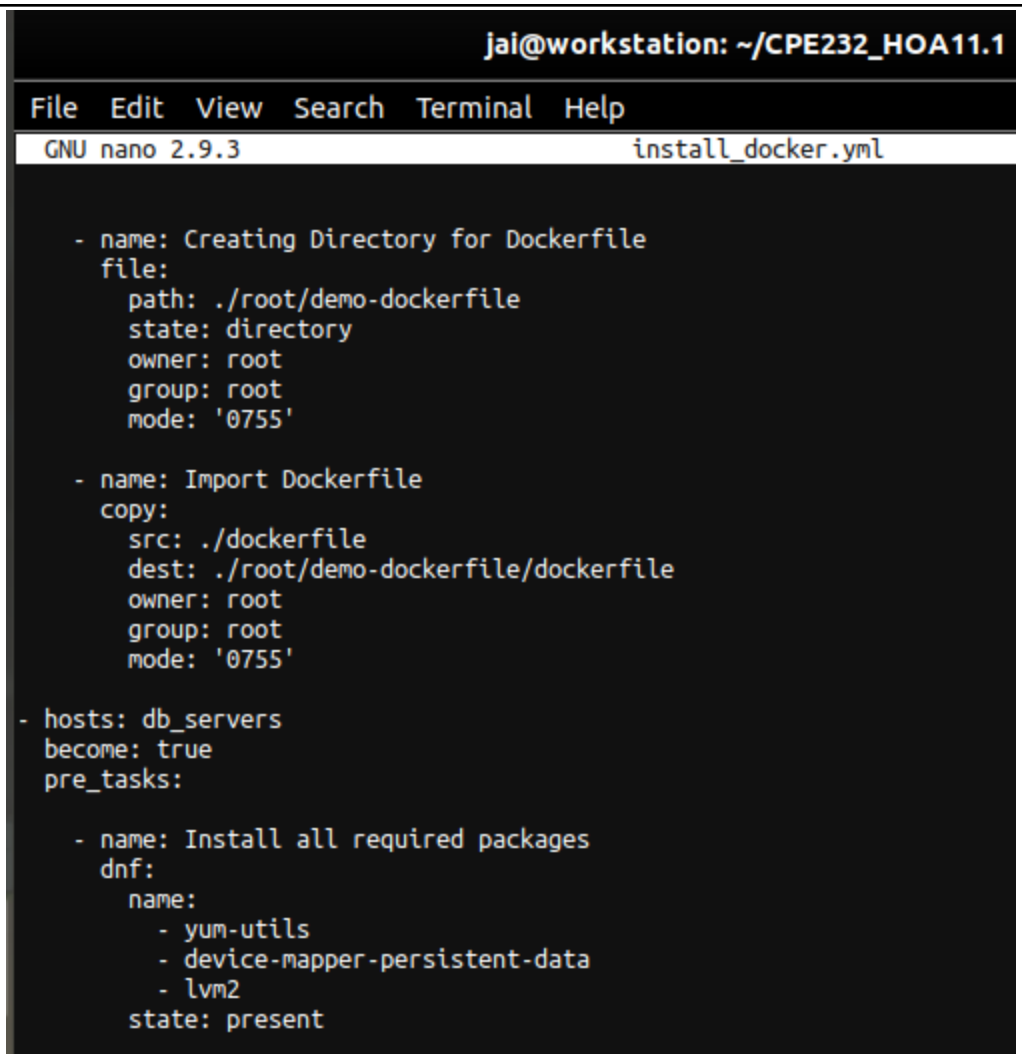
5. Create a docker for this activity.



Task 3: Create Playbook for Installing Docker in Ubuntu and CentOS

1. Create a playbook and name it install_docker.yml.


jai@workstation: ~/CPE232_HOA11.1
File Edit View Search Terminal Help
GNU nano 2.9.3 install_docker.yml
- hosts: web_servers
become: true
pre_tasks:
- name: dpkg for Ubuntu
shell:
dpkg --configure -a
when: ansible_distribution == "Ubuntu"
- name: Install Docker for Ubuntu
apt:
name: docker
state: latest
when: ansible_distribution == "Ubuntu"
- name: Install SDK for Ubuntu
shell:
pip3 install docker-py
- name: Add group to Docker
shell:
usermod -aG docker jai
- name: Enable and Restart Docker for Ubuntu
service:
name: docker
state: started
enabled: true


jai@workstation: ~/CPE232_HOA11.1
File Edit View Search Terminal Help
GNU nano 2.9.3 install_docker.yml
- name: Creating Directory for Dockerfile
file:
path: ./root/demo-dockerfile
state: directory
owner: root
group: root
mode: '0755'
- name: Import Dockerfile
copy:
src: ./dockerfile
dest: ./root/demo-dockerfile/dockerfile
owner: root
group: root
mode: '0755'
- hosts: db_servers
become: true
pre_tasks:
- name: Install all required packages
dnf:
name:
- yum-utils
- device-mapper-persistent-data
- lvm2
state: present

```
GNU nano 2.9.3                                  install_docker.yml

    state: present

- name: Add Docker repository
  yum_repository:
    name: docker-ce
    description: Docker CE Stable - $basearch
    baseurl: https://download.docker.com/linux/centos/7/$basearch/stable
    gpgkey: https://download.docker.com/linux/centos/gpg
    enabled: yes

- name: Install Docker for CentOS
  dnf:
    name: docker-ce
    state: present

- name: Start and enable Docker service for CentOS
  systemd:
    name: docker
    state: started
    enabled: yes
```

2. Save the file and exit.

Task 4: Run and Verify

1. Run the command ansible-playbook - - ask-become-pass install_docker.yml to completely install it in both Ubuntu server and CentOS.

ENTIRE ansible-playbook

```
jai@workstation:~/CPE232_HOA11.1$ ansible-playbook --ask-become-pass install_do
cker.yml
BECOME password:

PLAY [web_servers] ************************************************************
*

TASK [Gathering Facts] ********************************************************
*
ok: [192.168.56.103]

TASK [dpkg for Ubuntu] ********************************************************
*
changed: [192.168.56.103]

TASK [Install Docker for Ubuntu] **********************************************
*
ok: [192.168.56.103]

TASK [Install SDK for Ubuntu] *************************************************
*
changed: [192.168.56.103]

TASK [Add group to Docker] ****************************************************
*
changed: [192.168.56.103]

TASK [Enable and Restart Docker for Ubuntu] ***********************************
*
ok: [192.168.56.103]

TASK [Creating Directory for Dockerfile] **************************************
*
ok: [192.168.56.103]

TASK [Import Dockerfile] ******************************************************
*
ok: [192.168.56.103]

PLAY [db_servers] *************************************************************
*

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

TASK [Install all required packages] ******************************************
*
ok: [192.168.56.105]

TASK [Add Docker repository] **************************************************
*
ok: [192.168.56.105]
```

```
TASK [Install Docker for CentOS] ***********************************************
*
ok: [192.168.56.105]

TASK [Start and enable Docker service for CentOS] ******************************
*
ok: [192.168.56.105]

PLAY RECAP *********************************************************************
*
192.168.56.103             : ok=8    changed=3    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
192.168.56.105             : ok=5    changed=0    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

2. Show the screenshot of the systemctl status in both Server 2 and CentOS. The status should be active.

OUTPUT:

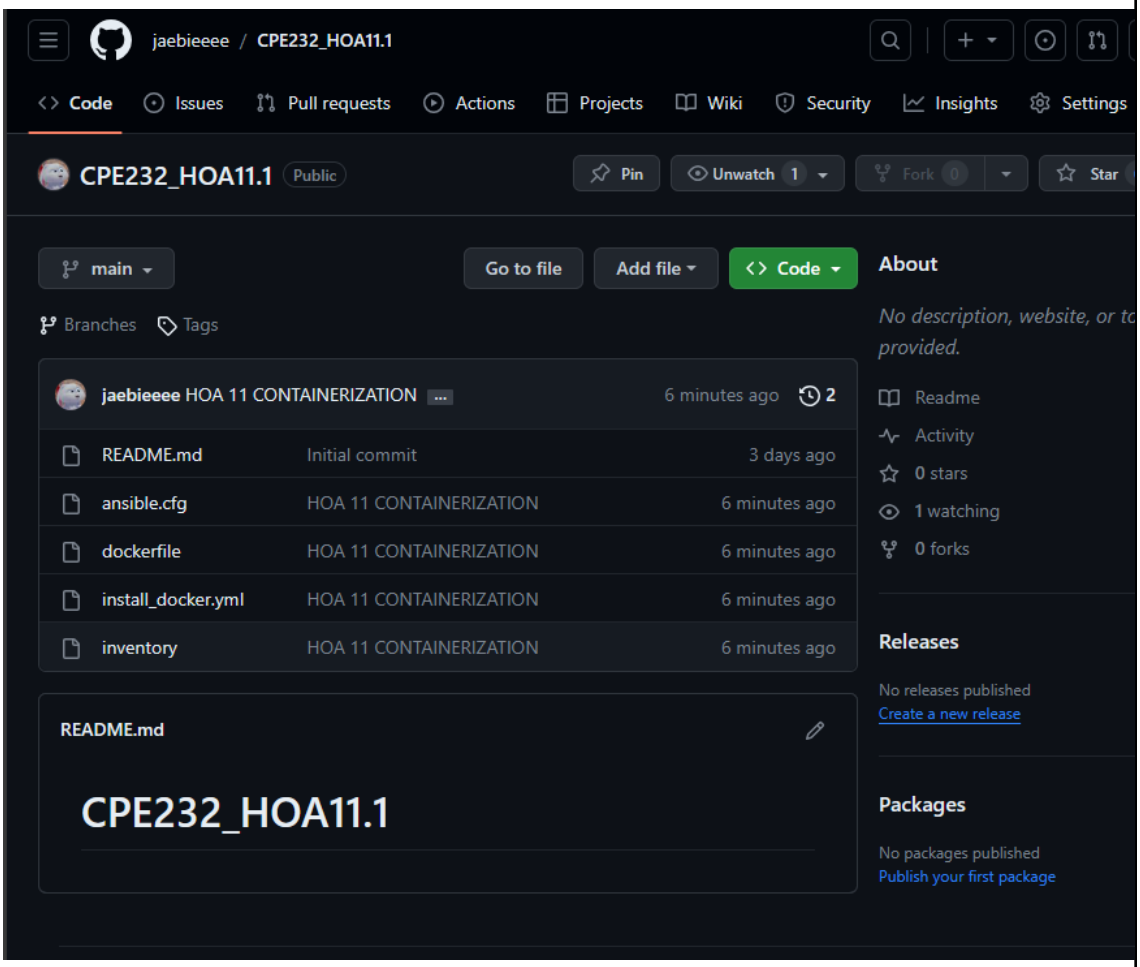
CENTOS

```
[jai@localhost ~]$ sudo systemctl status docker
[sudo] password for jai:
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disa
bled)
   Active: active (running) since Thu 2023-11-16 04:46:12 EST; 12min ago
     Docs: https://docs.docker.com
 Main PID: 10518 (dockerd)
    Tasks: 8
   Memory: 44.9M
   CGroup: /system.slice/docker.service
           └─10518 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd...

Nov 16 04:46:10 localhost.localdomain systemd[1]: Starting Docker Application Conta....
Nov 16 04:46:10 localhost.localdomain dockerd[10518]: time="2023-11-16T04:46:10.097..."
Nov 16 04:46:10 localhost.localdomain dockerd[10518]: time="2023-11-16T04:46:10.448..."
Nov 16 04:46:11 localhost.localdomain dockerd[10518]: time="2023-11-16T04:46:11.804..."
Nov 16 04:46:11 localhost.localdomain dockerd[10518]: time="2023-11-16T04:46:11.916..."
Nov 16 04:46:11 localhost.localdomain dockerd[10518]: time="2023-11-16T04:46:11.989...7
Nov 16 04:46:11 localhost.localdomain dockerd[10518]: time="2023-11-16T04:46:11.990..."
Nov 16 04:46:12 localhost.localdomain dockerd[10518]: time="2023-11-16T04:46:12.023..."
Nov 16 04:46:12 localhost.localdomain systemd[1]: Started Docker Application Contai....
Hint: Some lines were ellipsized, use -l to show in full.
[jai@localhost ~]$
```

UBUNTU

```
jai@server2:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset:
   Active: active (running) since Thu 2023-11-16 17:37:52 PST; 23min ago
     Docs: https://docs.docker.com
 Main PID: 17613 (dockerd)
    Tasks: 8
   CGroup: /system.slice/docker.service
           └─17613 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/conta

Nov 16 17:37:51 server2 dockerd[17613]: time="2023-11-16T17:37:51.762929586+08:
Nov 16 17:37:51 server2 dockerd[17613]: time="2023-11-16T17:37:51.762969670+08:
Nov 16 17:37:51 server2 dockerd[17613]: time="2023-11-16T17:37:51.763001149+08:
Nov 16 17:37:51 server2 dockerd[17613]: time="2023-11-16T17:37:51.763180554+08:
Nov 16 17:37:51 server2 dockerd[17613]: time="2023-11-16T17:37:51.912898246+08:
Nov 16 17:37:51 server2 dockerd[17613]: time="2023-11-16T17:37:51.961587739+08:
Nov 16 17:37:52 server2 dockerd[17613]: time="2023-11-16T17:37:52.021696433+08:
Nov 16 17:37:52 server2 dockerd[17613]: time="2023-11-16T17:37:52.021755730+08:
Nov 16 17:37:52 server2 systemd[1]: Started Docker Application Container Engine
Nov 16 17:37:52 server2 dockerd[17613]: time="2023-11-16T17:37:52.090507641+08:
 Terminal
[1]+  Stopped                 sudo systemctl status docker
jai@server2:~$
```

3. Upload it in the github.



GITHUB LINK:https://github.com/jaebieeee/CPE232_HOA11.1.git

Reflections:

Answer the following:

1. What are the benefits of implementing containerizations?
    - Containerization in Ubuntu and CentOS brings three key advantages. Firstly, it enhances scalability by allowing applications to run consistently across various environments. Secondly, it streamlines deployment, making it quicker and more reliable, thanks to isolated containers. Lastly, it improves resource efficiency, as containers share the host OS kernel, reducing overhead. Together, these benefits simplify development, enhance portability, and optimize resource utilization in a user-friendly manner for both Ubuntu and CentOS users.

Conclusions:

In this activity, I was able to encounter the docker as well as containerization. Wrapping up this activity was a real eye-opener into the magic of Dockerfile creation and playbook deployment on both Ubuntu and CentOS. Crafting Dockerfiles taught me the art of packaging applications, making deployment a breeze. Running playbooks added a layer of automation, saving time and effort. The dual OS experience broadened my adaptability. Witnessing the seamless orchestration of containers in action was nothing short of empowering. This hands-on venture not only polished my technical skills but also fueled my enthusiasm for streamlining future projects with the efficiency of Docker and playbooks.