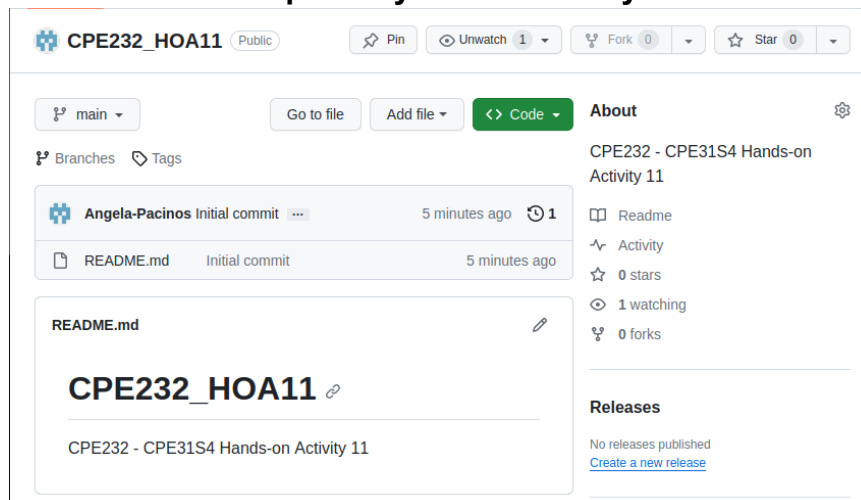


<b>Name:</b> Pacinos, Angela Monique A.	<b>Date Performed:</b> 11-13-23
<b>Course/Section:</b> CPE232 - CPE31S4	<b>Date Submitted:</b> 11-13-23
<b>Instructor:</b> Dr. Jonathan V. Taylor	<b>Semester and SY:</b> 1st Sem: '23 - '24
<b>Activity 11: Containerization</b>	
<b>1. Objectives</b>	
Create a Dockerfile and form a workflow using Ansible as Infrastructure as Code (IaC) to enable Continuous Delivery process	
<b>2. Discussion</b>	
<p>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.</p> <p>Source: <a href="https://docs.docker.com/get-started/overview/">https://docs.docker.com/get-started/overview/</a></p> <p>You may also check the difference between containers and virtual machines. Click the link given below.</p> <p>Source: <a href="https://docs.microsoft.com/en-us/virtualization/windowscontainers/about/containers-vs-vm">https://docs.microsoft.com/en-us/virtualization/windowscontainers/about/containers-vs-vm</a></p>	
<b>3. Tasks</b>	
<ol style="list-style-type: none"> <li>1. Create a new repository for this activity.</li> <li>2. Install Docker and enable the docker socket.</li> <li>3. Add to Docker group to your current user.</li> <li>4. Create a Dockerfile to install web and DB server.</li> <li>5. Install and build the Dockerfile using Ansible.</li> <li>6. Add, commit and push it to your repository.</li> </ol>	
<b>4. Output</b> (screenshots and explanations)	

## INPUT

### Create a new repository for the activity and clone it.



```
angela@workstation:~$ git clone https://github.com/Angela-Pacinos/CPE232_HOA11.git
Cloning into 'CPE232_HOA11'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
```

### Download and Enable Docker.

```
angela@workstation:~/CPE232_HOA11$ 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).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
angela@workstation:~/CPE232_HOA11$ systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2023-11-13 10:51:26 PST; 1h 19min ago
     Docs: https://docs.docker.com
   Main PID: 5831 (dockerd)
    Tasks: 28
   CGroup: /system.slice/docker.service
           └─ 5831 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/c
              19073 /usr/bin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-
              19081 /usr/bin/docker-proxy -proto tcp -host-ip :: -host-port

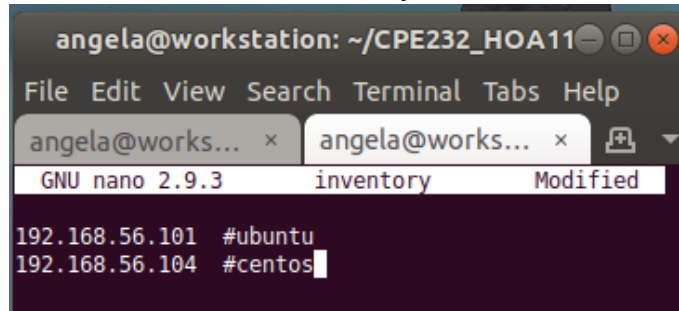
Nov 13 11:44:49 workstation dockerd[5831]: time="2023-11-13T11:44:49.095630
Nov 13 11:44:49 workstation dockerd[5831]: time="2023-11-13T11:44:49.240857
Nov 13 11:44:49 workstation dockerd[5831]: time="2023-11-13T11:44:49.240975
Nov 13 11:52:58 workstation dockerd[5831]: time="2023-11-13T11:52:58.259731
Nov 13 11:53:14 workstation dockerd[5831]: time="2023-11-13T11:53:14.398415
Nov 13 11:55:41 workstation dockerd[5831]: time="2023-11-13T11:55:41.787134
Nov 13 11:55:57 workstation dockerd[5831]: time="2023-11-13T11:55:57.256848
Nov 13 11:56:35 workstation dockerd[5831]: time="2023-11-13T11:56:35.535207
Nov 13 11:56:48 workstation dockerd[5831]: time="2023-11-13T11:56:48.558150
Nov 13 12:01:09 workstation dockerd[5831]: time="2023-11-13T12:01:09.576116
```

### Add Docker group to the current user.

```
angela@workstation:~/CPE232_H0A11$ sudo usermod -aG docker angela
angela@workstation:~/CPE232_H0A11$ sudo systemctl restart docker
```

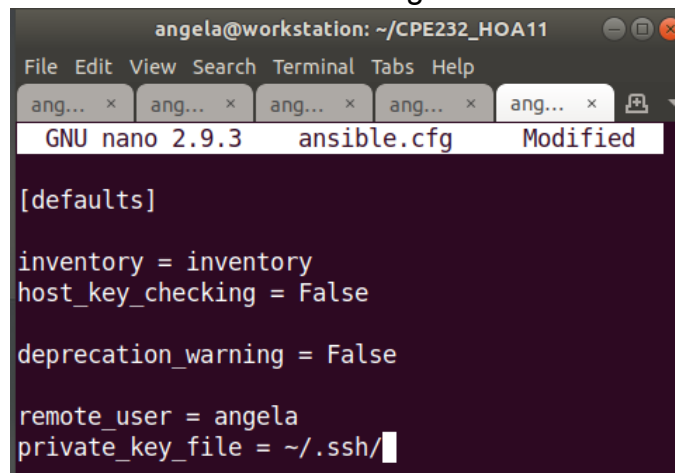
### Create a Dockerfile to install web and DB server.

inventory



```
angela@workstation: ~/CPE232_H0A11
File Edit View Search Terminal Tabs Help
angela@works... x angela@works... x
GNU nano 2.9.3 inventory Modified
192.168.56.101 #ubuntu
192.168.56.104 #centos
```

ansible.cfg



```
angela@workstation: ~/CPE232_H0A11
File Edit View Search Terminal Tabs Help
ang... x ang... x ang... x ang... x ang... x
GNU nano 2.9.3 ansible.cfg Modified

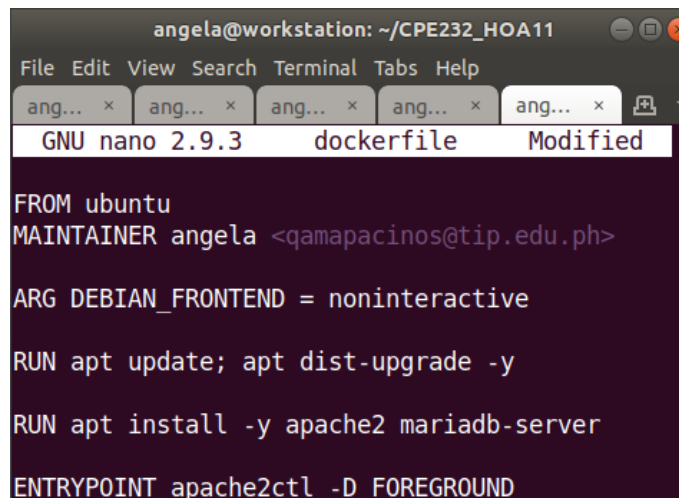
[defaults]

inventory = inventory
host_key_checking = False

deprecation_warning = False

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

dockerfile



```
angela@workstation: ~/CPE232_H0A11
File Edit View Search Terminal Tabs Help
ang... x ang... x ang... x ang... x ang... x
GNU nano 2.9.3 dockerfile Modified

FROM ubuntu
MAINTAINER angela <qamapacinos@tip.edu.ph>

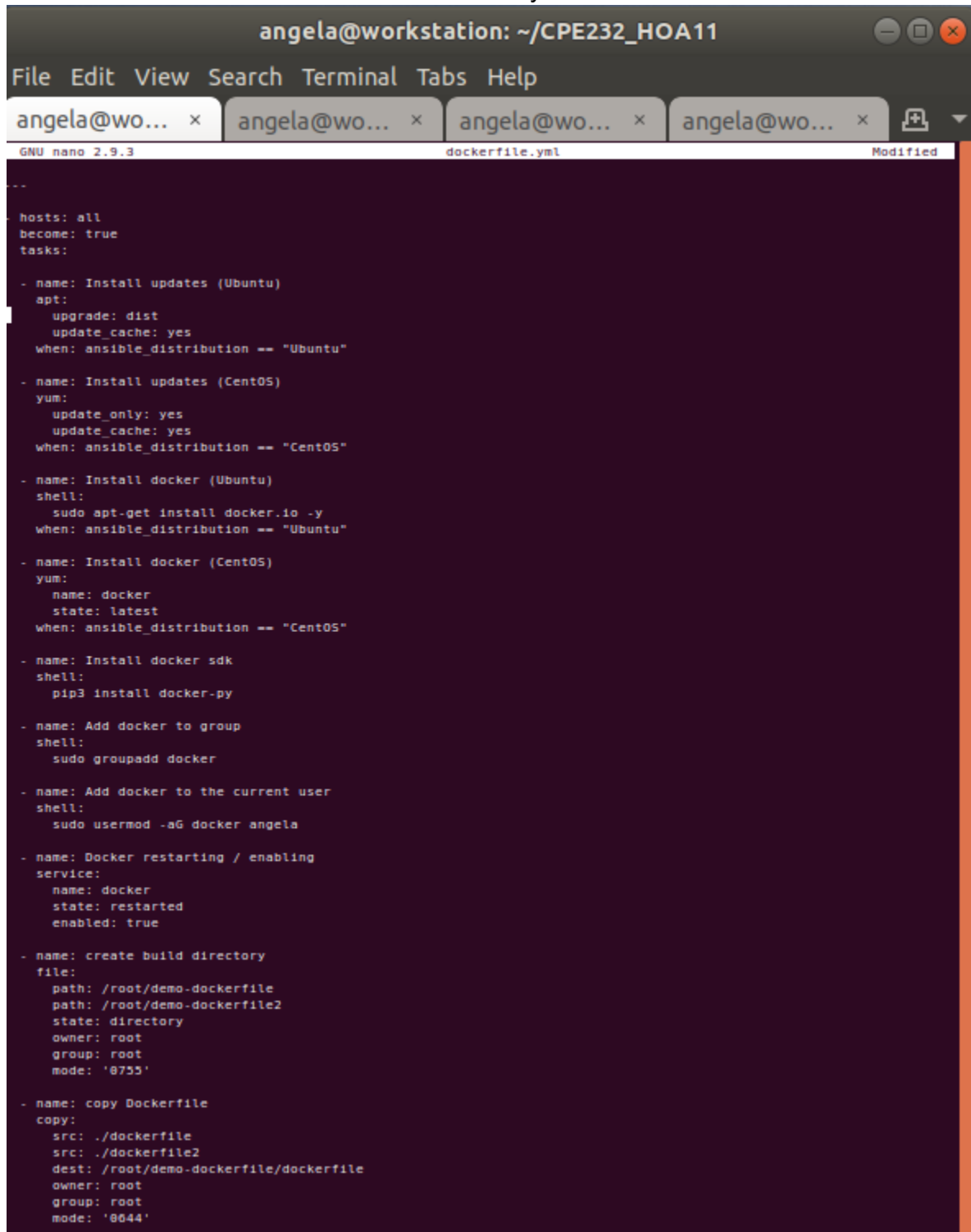
ARG DEBIAN_FRONTEND = noninteractive

RUN apt update; apt dist-upgrade -y

RUN apt install -y apache2 mariadb-server

ENTRYPOINT apache2ctl -D FOREGROUND
```

## Install and build the Dockerfile using Ansible. dockerfile.yml



```
angela@workstation: ~/CPE232_HOA11
File Edit View Search Terminal Tabs Help
angela@wo... x angela@wo... x angela@wo... x angela@wo... x
GNU nano 2.9.3 dockerfile.yml Modified
---
- hosts: all
  become: true
  tasks:
    - name: Install updates (Ubuntu)
      apt:
        upgrade: dist
        update_cache: yes
      when: ansible_distribution == "Ubuntu"
    - name: Install updates (CentOS)
      yum:
        update_only: yes
        update_cache: yes
      when: ansible_distribution == "CentOS"
    - name: Install docker (Ubuntu)
      shell:
        sudo apt-get install docker.io -y
      when: ansible_distribution == "Ubuntu"
    - name: Install docker (CentOS)
      yum:
        name: docker
        state: latest
      when: ansible_distribution == "CentOS"
    - name: Install docker sdk
      shell:
        pip3 install docker-py
    - name: Add docker to group
      shell:
        sudo groupadd docker
    - name: Add docker to the current user
      shell:
        sudo usermod -aG docker angela
    - name: Docker restarting / enabling
      service:
        name: docker
        state: restarted
        enabled: true
    - name: create build directory
      file:
        path: /root/demo-dockerfile
        path: /root/demo-dockerfile2
        state: directory
        owner: root
        group: root
        mode: '0755'
    - name: copy Dockerfile
      copy:
        src: ./dockerfile
        src: ./dockerfile2
        dest: /root/demo-dockerfile/dockerfile
        owner: root
        group: root
        mode: '0644'
```

## PROCESS

```
angela@workstation: ~/CPE232_HOA11
File Edit View Search Terminal Tabs Help
angela@workst... x angela@workst... x angela@workst... x angela@workst... x
angela@workstation:~/CPE232_HOA11$ sudo nano dockerrfile.yml
angela@workstation:~/CPE232_HOA11$ ansible-playbook --ask-become-pass dockerrfile.yml
[DEPRECATION WARNING]: Ansible will require Python 3.8 or newer on the controller starting with Ansible
2.12. Current version: 3.6.9 (default, Mar 10 2023, 16:46:00) [GCC 8.4.0]. This feature will be removed
from ansible-core in version 2.12. Deprecation warnings can be disabled by setting
deprecation_warnings=False in ansible.cfg.
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
[DEPRECATION WARNING]: Distribution Ubuntu 18.04 on host 192.168.56.101 should use /usr/bin/python3, but
is using /usr/bin/python for backward compatibility with prior Ansible releases. A future Ansible release
will default to using the discovered platform python for this host. See https://docs.ansible.com/ansible-
core/2.11/reference_appendices/interpreter_discovery.html for more information. This feature will be
removed in version 2.12. Deprecation warnings can be disabled by setting deprecation_warnings=False in
ansible.cfg.
ok: [192.168.56.101]
ok: [192.168.56.104]

TASK [Install updates (Ubuntu)] *****
skipping: [192.168.56.104]
ok: [192.168.56.101]

TASK [Install updates (CentOS)] *****
skipping: [192.168.56.101]
ok: [192.168.56.104]

TASK [Install docker (Ubuntu)] *****
skipping: [192.168.56.104]
changed: [192.168.56.101]

TASK [Install docker (CentOS)] *****
skipping: [192.168.56.101]
ok: [192.168.56.104]

TASK [Install docker sdk] *****
changed: [192.168.56.101]
changed: [192.168.56.104]

TASK [Add docker to group] *****
changed: [192.168.56.101]
changed: [192.168.56.104]

TASK [Add docker to the current user] *****
changed: [192.168.56.101]
changed: [192.168.56.104]

TASK [Docker restarting / enabling] *****
changed: [192.168.56.101]
changed: [192.168.56.104]

TASK [create build directory] *****
ok: [192.168.56.104]
ok: [192.168.56.101]

TASK [copy Dockerfile] *****
ok: [192.168.56.101]
ok: [192.168.56.104]

PLAY RECAP *****
192.168.56.101 : ok=9  changed=5  unreachable=0  failed=0  skipped=2  rescued=0  ignored=0
192.168.56.104 : ok=9  changed=4  unreachable=0  failed=0  skipped=2  rescued=0  ignored=0
```

## OUTPUT

**Reflections:**

Answer the following:

1. What are the benefits of implementing containerizations?

Docker containers are very helpful and can be used for easy sharing and deploying of images across different platforms. It also includes all the dependencies and libraries of the packages, avoiding conflicts when using the packages as all of the needed dependencies are already included. And from what we have implemented I saw how easy and quick it was when enabling resources as compared to the traditional virtual machine. And as I mentioned it can be deployed and shared to different platforms making it easy to change environments.

**Conclusions:**

For this module, we learned what a docker container is, its uses, and how we can use it. The given presentation about how we can install the docker and implement, and create images allows us to follow through this module. For this activity we have to install the docker in both our ubuntu and centos using playbook as well as create and install the apache2 and mariadb. Overall, it was okay at first but then so many errors kept appearing which made it difficult to debug and understand. I had a hard time where we have to build a docker image using playbook and it keeps having errors.