Name: Zamora, Angelo E.	**Date Performed:** 11 - 13 - 2024
Course/Section: CpE31S2	**Date Submitted:** 11 - 13 - 2024
Instructor: Engr. Robin Valenzuela	**Semester and SY:** 1st Sem 2024 - 2025

Activity 11: Containerization

- **Objectives**

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

- **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.

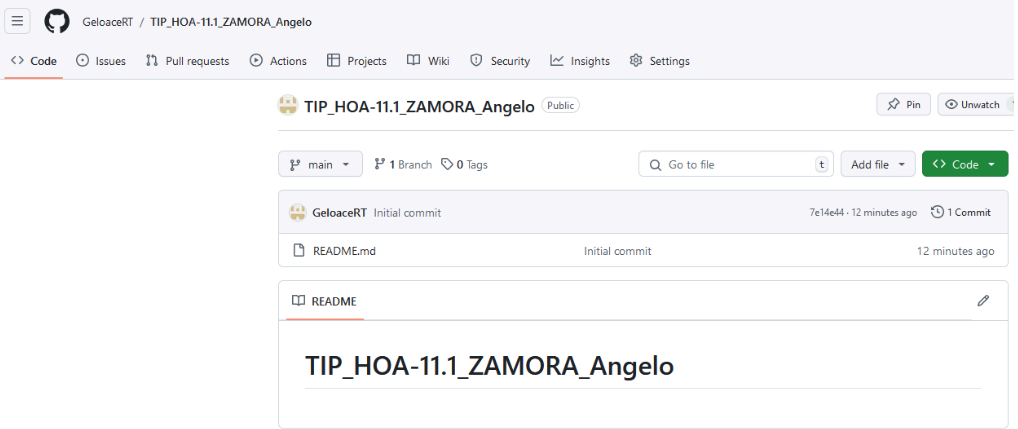
Source: https://docs.microsoft.com/en-us/virtualization/windowscontainers/about/containers-vs-vm

- **Tasks**

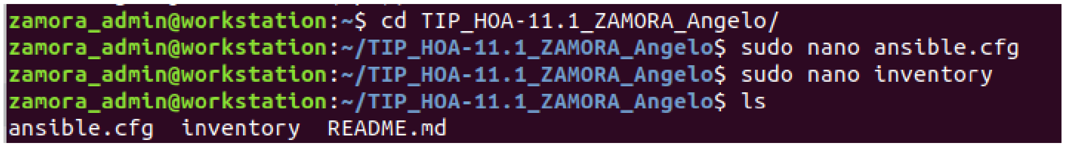
1. Create a new repository for this activity.
2. Install Docker and enable the docker socket.
3. Add to Docker group to your current user.
4. Create a Dockerfile to install web and DB server.
5. Install and build the Dockerfile using Ansible.
6. Add, commit and push it to your repository.

- **Output** (screenshots and explanations)

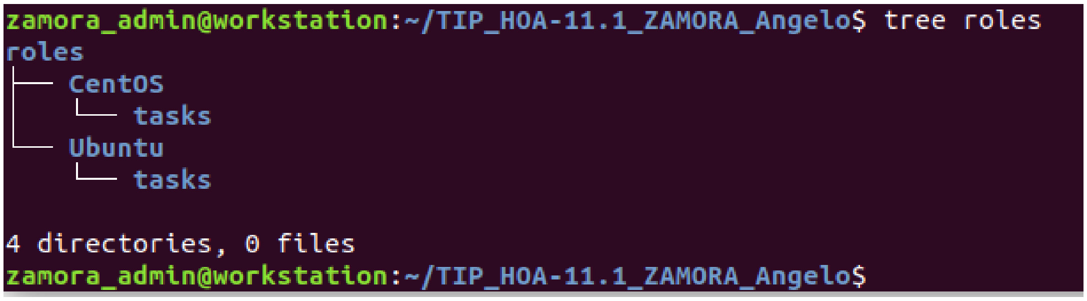
- Create a new repository for this activity.



TIP_HOA-11.1_ZAMORA_Angelo

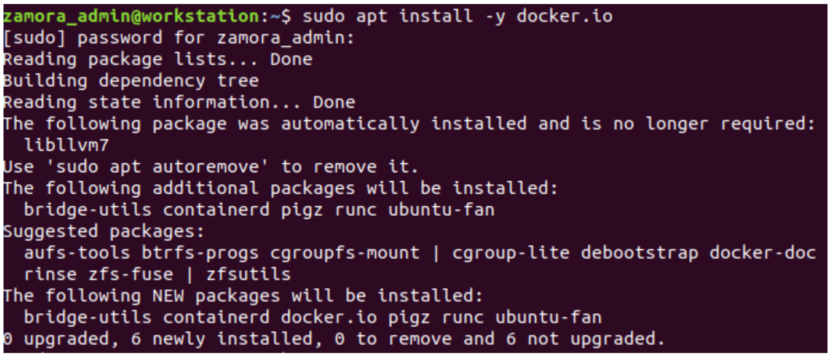


```
zamora_admin@workstation:~$ cd TIP_HOA-11.1_ZAMORA_Angelo/
zamora_admin@workstation:~/TIP_HOA-11.1_ZAMORA_Angelo$ sudo nano ansible.cfg
zamora_admin@workstation:~/TIP_HOA-11.1_ZAMORA_Angelo$ sudo nano inventory
zamora_admin@workstation:~/TIP_HOA-11.1_ZAMORA_Angelo$ ls
ansible.cfg  inventory  README.md
```

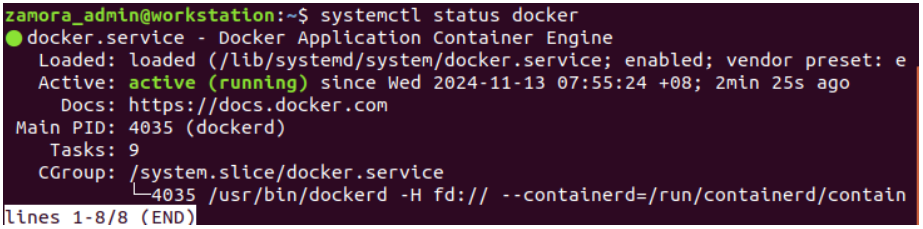


```
zamora_admin@workstation:~/TIP_HOA-11.1_ZAMORA_Angelo$ tree roles
roles
├── CentOS
│   └── tasks
└── Ubuntu
    └── tasks

4 directories, 0 files
zamora_admin@workstation:~/TIP_HOA-11.1_ZAMORA_Angelo$
```

- Install Docker and enable the docker socket.
  Control Node:

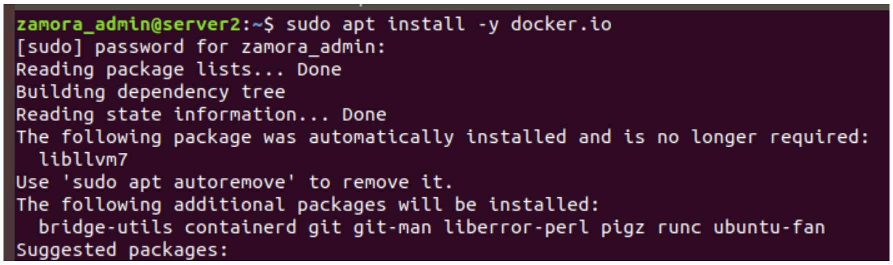


```
zamora_admin@workstation:~$ sudo apt install -y docker.io
[sudo] password for zamora_admin:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libllvm7
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  bridge-utils containerd pigz runc ubuntu-fan
Suggested packages:
  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 6 not upgraded.
```

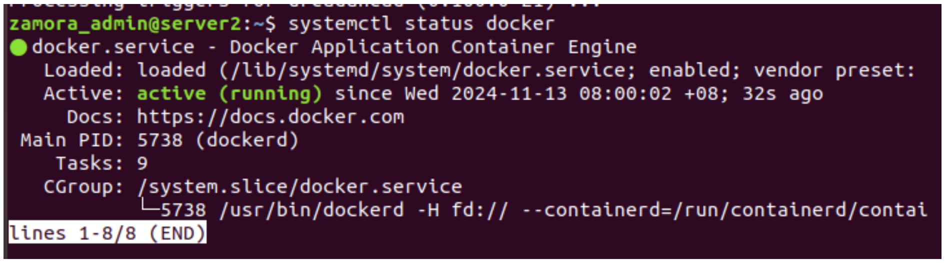


```
zamora_admin@workstation:~$ systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: e
   Active: active (running) since Wed 2024-11-13 07:55:24 +08; 2min 25s ago
     Docs: https://docs.docker.com
 Main PID: 4035 (dockerd)
    Tasks: 9
   CGroup: /system.slice/docker.service
           └─4035 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/contain
lines 1-8/8 (END)
```

Manage Node:

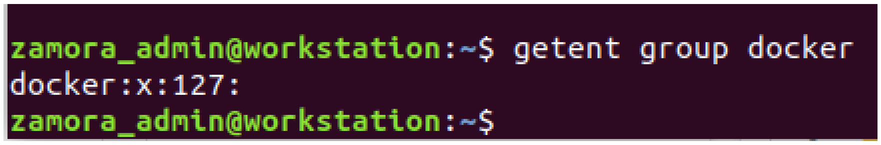


```
zamora_admin@server2:~$ sudo apt install -y docker.io
[sudo] password for zamora_admin:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libllvm7
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  bridge-utils containerd git git-man liberror-perl pigz runc ubuntu-fan
Suggested packages:
```

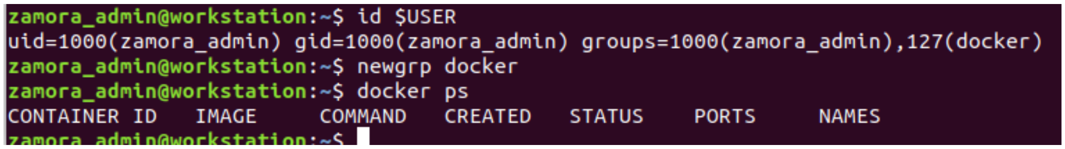


```
zamora_admin@server2:~$ systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset:
   Active: active (running) since Wed 2024-11-13 08:00:02 +08; 32s ago
     Docs: https://docs.docker.com
 Main PID: 5738 (dockerd)
    Tasks: 9
   CGroup: /system.slice/docker.service
           └─5738 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/contai
lines 1-8/8 (END)
```

- Add a Docker group to your current user.
  Control Node:

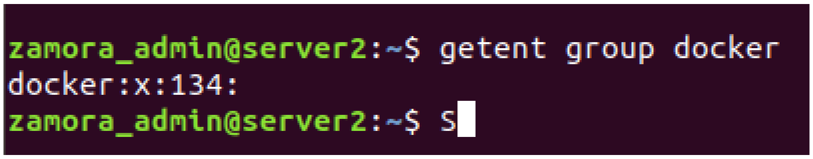


```
zamora_admin@workstation:~$ getent group docker
docker:x:127:
zamora_admin@workstation:~$
```

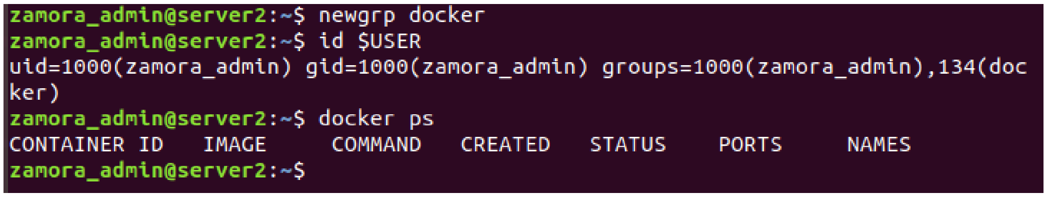


```
zamora_admin@workstation:~$ id $USER
uid=1000(zamora_admin) gid=1000(zamora_admin) groups=1000(zamora_admin),127(docker)
zamora_admin@workstation:~$ newgrp docker
zamora_admin@workstation:~$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
zamora_admin@workstation:~$
```

Manage Node:

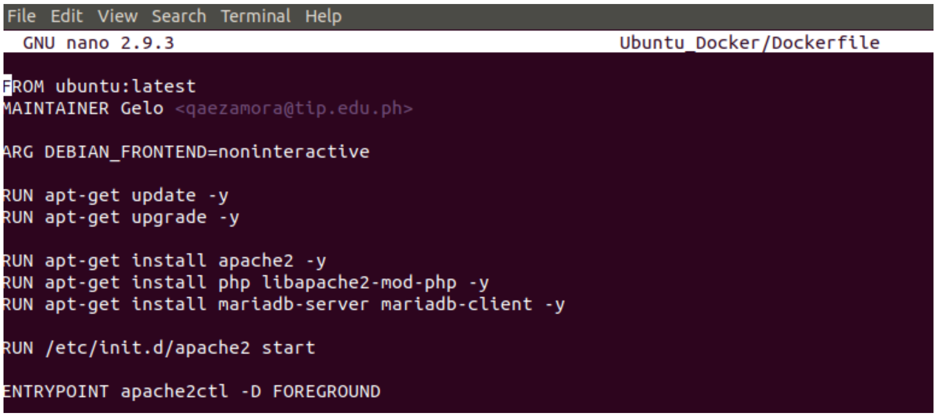


```
zamora_admin@server2:~$ getent group docker
docker:x:134:
zamora_admin@server2:~$ S
```



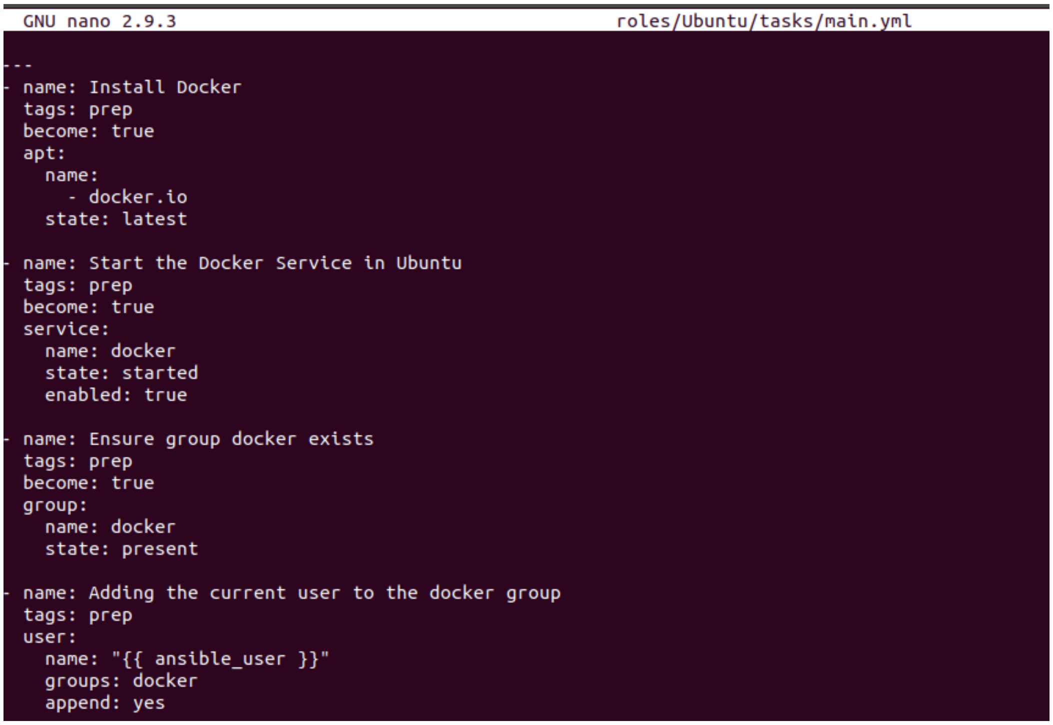
```
zamora_admin@server2:~$ newgrp docker
zamora_admin@server2:~$ id $USER
uid=1000(zamora_admin) gid=1000(zamora_admin) groups=1000(zamora_admin),134(doc
ker)
zamora_admin@server2:~$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
zamora_admin@server2:~$
```

- Create a Dockerfile to install web and DB server.

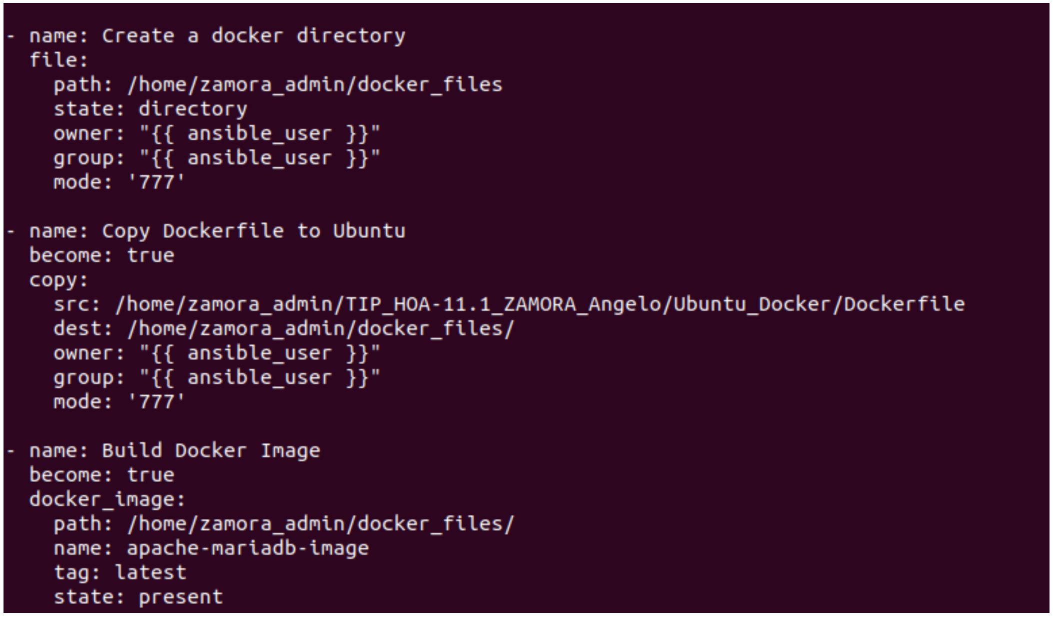


```
  GNU nano 2.9.3                                         Ubuntu_Docker/Dockerfile

FROM ubuntu:latest
MAINTAINER Gelo <qaezamora@tip.edu.ph>

ARG DEBIAN_FRONTEND=noninteractive

RUN apt-get update -y
RUN apt-get upgrade -y

RUN apt-get install apache2 -y
RUN apt-get install php libapache2-mod-php -y
RUN apt-get install mariadb-server mariadb-client -y

RUN /etc/init.d/apache2 start

ENTRYPOINT apache2ctl -D FOREGROUND
```

- Create a Docker Image using Ansible.
  Here is the main.yml under roles/Ubuntu/tasks



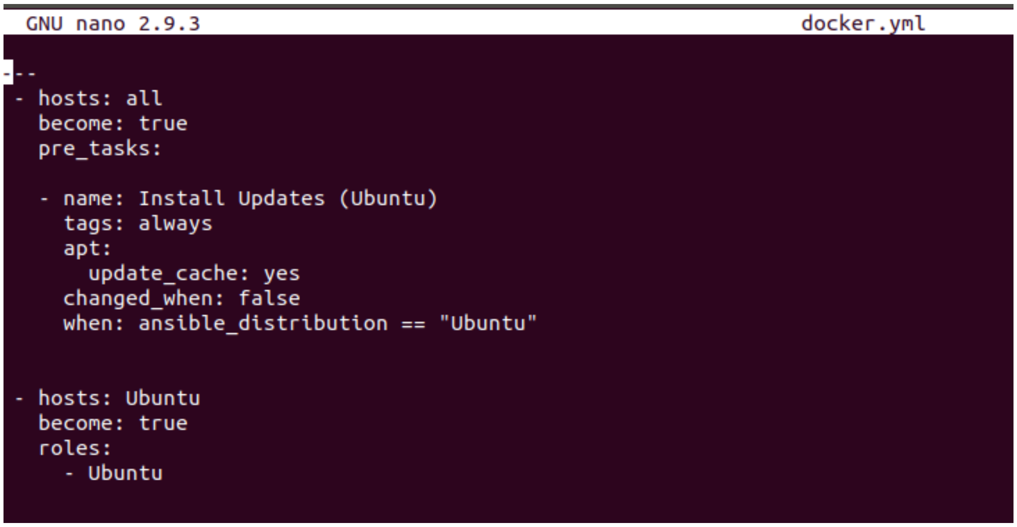
```
  GNU nano 2.9.3                                         roles/Ubuntu/tasks/main.yml

---
- name: Install Docker
  tags: prep
  become: true
  apt:
    name:
      - docker.io
    state: latest

- name: Start the Docker Service in Ubuntu
  tags: prep
  become: true
  service:
    name: docker
    state: started
    enabled: true

- name: Ensure group docker exists
  tags: prep
  become: true
  group:
    name: docker
    state: present

- name: Adding the current user to the docker group
  tags: prep
  user:
    name: "{{ ansible_user }}"
    groups: docker
    append: yes
```



```
- name: Create a docker directory
  file:
    path: /home/zamora_admin/docker_files
    state: directory
    owner: "{{ ansible_user }}"
    group: "{{ ansible_user }}"
    mode: '777'

- name: Copy Dockerfile to Ubuntu
  become: true
  copy:
    src: /home/zamora_admin/TIP_HOA-11.1_ZAMORA_Angelo/Ubuntu_Docker/Dockerfile
    dest: /home/zamora_admin/docker_files/
    owner: "{{ ansible_user }}"
    group: "{{ ansible_user }}"
    mode: '777'

- name: Build Docker Image
  become: true
  docker_image:
    path: /home/zamora_admin/docker_files/
    name: apache-mariadb-image
    tag: latest
    state: present
```

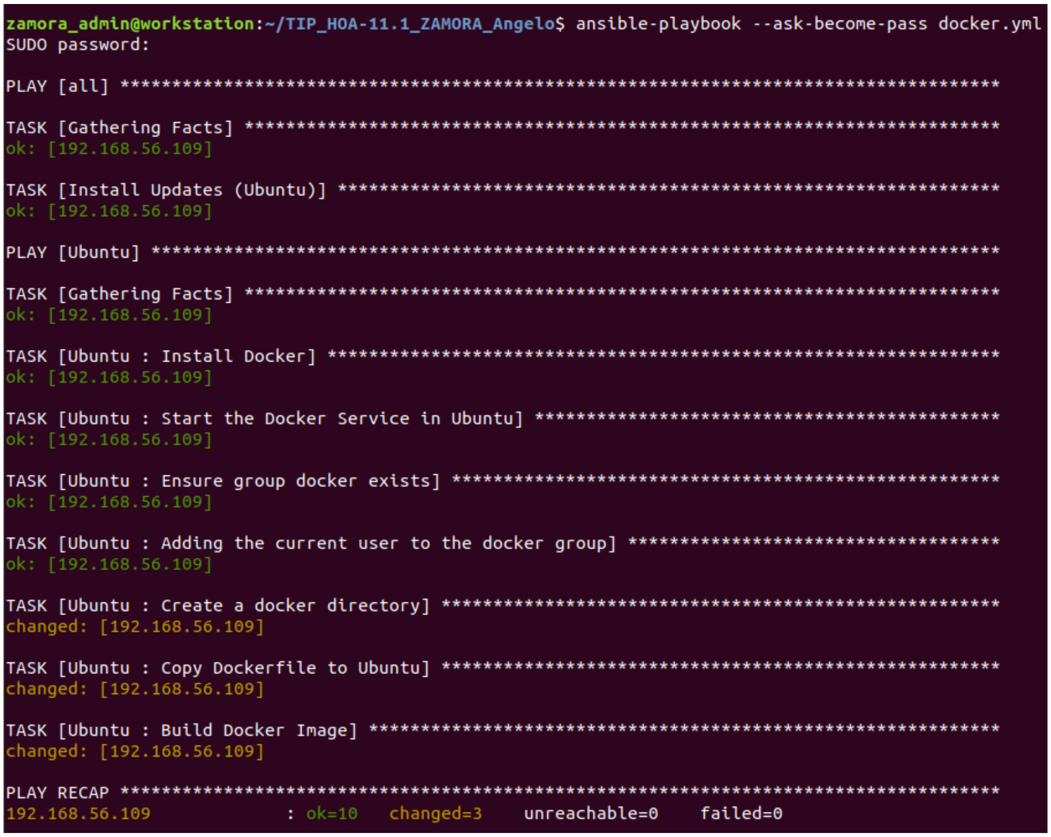
Explanation: First installs docker to the manage nodes, or it can serve as a checker in case there is no docker to the Servers. Then it will start the service Docker to the Servers. The creation of the docker group is created upon installing it, so I've just added the current user to the group so that it can access the docker commands and syntaxes. After that I've created the directory to where to put the Dockerfile itself. Setting the permissions right to ensure no errors on building the docker image. Also take note that in the building image part of the playbook it must be a directory on the path not set at the file itself.

Created the main playbook of repo docker.yml



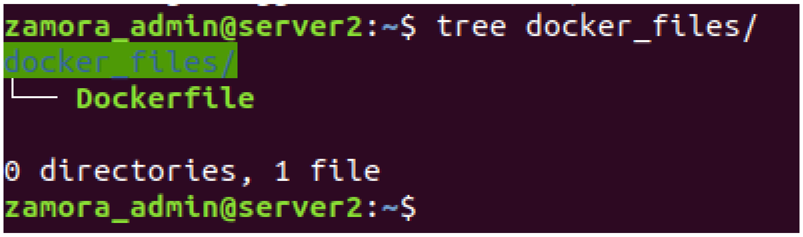
```
  GNU nano 2.9.3                                                docker.yml

---
 - hosts: all
   become: true
   pre_tasks:

   - name: Install Updates (Ubuntu)
     tags: always
     apt:
       update_cache: yes
     changed_when: false
     when: ansible_distribution == "Ubuntu"


 - hosts: Ubuntu
   become: true
   roles:
     - Ubuntu
```

Running the Playbook:

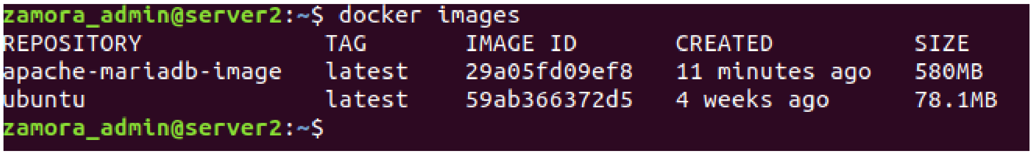


```
zamora_admin@workstation:~/TIP_HOA-11.1_ZAMORA_Angelo$ ansible-playbook --ask-become-pass docker.yml
SUDO password:

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

TASK [Gathering Facts] ***********************************************************************
ok: [192.168.56.109]

TASK [Install Updates (Ubuntu)] **************************************************************
ok: [192.168.56.109]

PLAY [Ubuntu] ********************************************************************************

TASK [Gathering Facts] ***********************************************************************
ok: [192.168.56.109]

TASK [Ubuntu : Install Docker] ***************************************************************
ok: [192.168.56.109]

TASK [Ubuntu : Start the Docker Service in Ubuntu] *******************************************
ok: [192.168.56.109]

TASK [Ubuntu : Ensure group docker exists] ***************************************************
ok: [192.168.56.109]

TASK [Ubuntu : Adding the current user to the docker group] **********************************
ok: [192.168.56.109]

TASK [Ubuntu : Create a docker directory] ****************************************************
changed: [192.168.56.109]

TASK [Ubuntu : Copy Dockerfile to Ubuntu] ****************************************************
changed: [192.168.56.109]

TASK [Ubuntu : Build Docker Image] ***********************************************************
changed: [192.168.56.109]

PLAY RECAP ***********************************************************************************
192.168.56.109             : ok=10   changed=3    unreachable=0    failed=0
```

Checking the docker image in the Manage Node

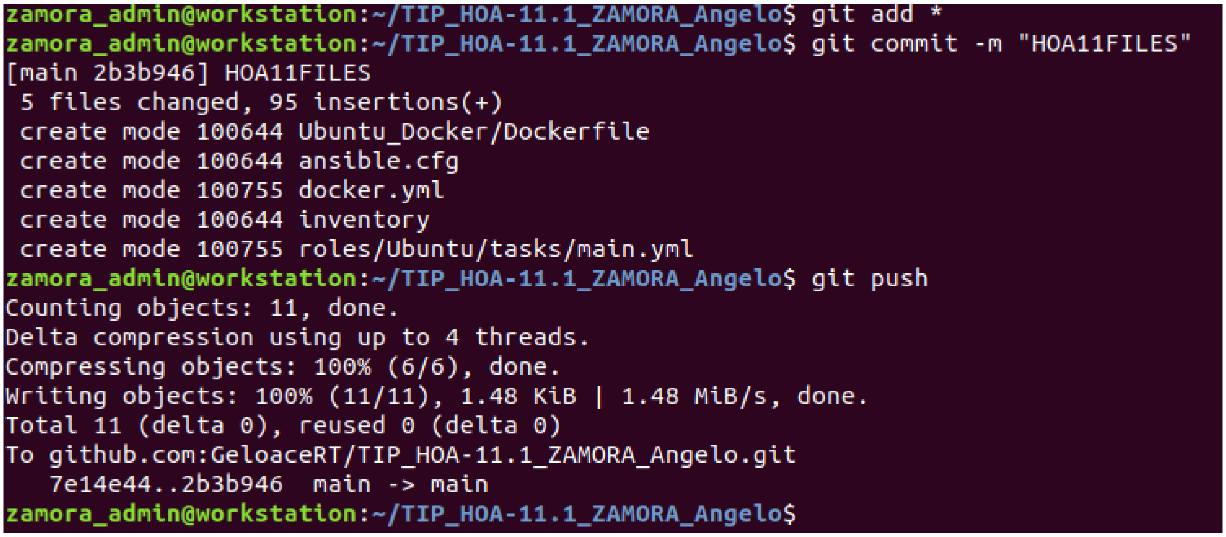


```
zamora_admin@server2:~$ tree docker_files/
docker_files/
└── Dockerfile

0 directories, 1 file
zamora_admin@server2:~$
```

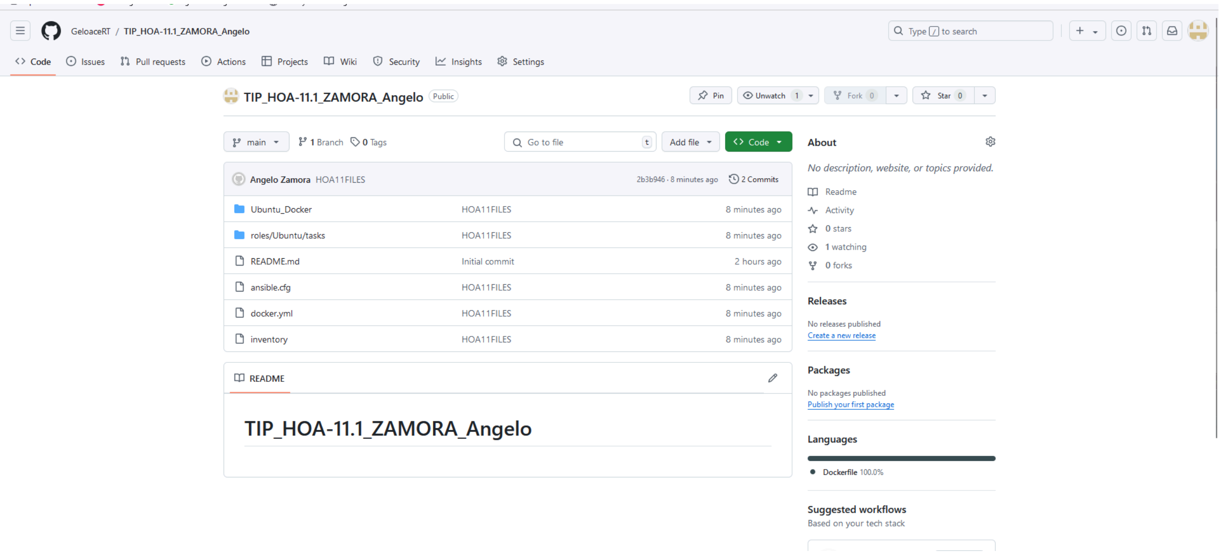


```
zamora_admin@server2:~$ docker images
REPOSITORY            TAG       IMAGE ID       CREATED          SIZE
apache-mariadb-image  latest    29a05fd09ef8   11 minutes ago   580MB
ubuntu                latest    59ab366372d5   4 weeks ago      78.1MB
zamora_admin@server2:~$
```

- Add, commit and push it to your repository.



```
zamora_admin@workstation:~/TIP_HOA-11.1_ZAMORA_Angelo$ git add *
zamora_admin@workstation:~/TIP_HOA-11.1_ZAMORA_Angelo$ git commit -m "HOA11FILES"
[main 2b3b946] HOA11FILES
 5 files changed, 95 insertions(+)
 create mode 100644 Ubuntu_Docker/Dockerfile
 create mode 100644 ansible.cfg
 create mode 100755 docker.yml
 create mode 100644 inventory
 create mode 100755 roles/Ubuntu/tasks/main.yml
zamora_admin@workstation:~/TIP_HOA-11.1_ZAMORA_Angelo$ git push
Counting objects: 11, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (11/11), 1.48 KiB | 1.48 MiB/s, done.
Total 11 (delta 0), reused 0 (delta 0)
To github.com:GeloaceRT/TIP_HOA-11.1_ZAMORA_Angelo.git
   7e14e44..2b3b946  main -> main
zamora_admin@workstation:~/TIP_HOA-11.1_ZAMORA_Angelo$
```



GeloaceRT / TIP_HOA-11.1_ZAMORA_Angelo

TIP_HOA-11.1_ZAMORA_Angelo Public

main · 1 Branch · 0 Tags

Angelo Zamora HOA11FILES · 2b3b946 · 8 minutes ago · 2 Commits

Ubuntu_Docker	HOA11FILES	8 minutes ago
roles/Ubuntu/tasks	HOA11FILES	8 minutes ago
README.md	Initial commit	2 hours ago
ansible.cfg	HOA11FILES	8 minutes ago
docker.yml	HOA11FILES	8 minutes ago
inventory	HOA11FILES	8 minutes ago

README

TIP_HOA-11.1_ZAMORA_Angelo

About

No description, website, or topics provided.

Readme
Activity
0 stars
1 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

Dockerfile 100.0%

Suggested workflows
Based on your tech stack

Github Link: https://github.com/GeloaceRT/TIP_HOA-11.1_ZAMORA_Angelo

Reflections:

Answer the following:

1. What are the benefits of implementing containerizations?

   A few of the main advantages of containerization are increased resource efficiency, scalability, and consistency. Developers may guarantee that applications perform consistently across many settings, from development to production, by encapsulating them and their dependencies into lightweight, portable containers. Containers provide simpler scaling, quicker deployment, and application isolation, which makes upgrades and testing easier. By enabling several containers to operate on a single host with no overhead, containerization also optimizes resource utilization and lowers infrastructure expenses. DevOps procedures and continuous integration/continuous delivery (CI/CD) pipelines are made easier by containerized systems' increased flexibility, resilience, and manageability.

Conclusions:

In conclusion, implementing Docker along with Ansible for Infrastructure as Code (IaC) in a Continuous Delivery (CD) workflow significantly enhances the development and deployment process. By using Docker to containerize applications and Ansible to automate the infrastructure setup and application deployment, I can ensure consistency, scalability, and speed in moving applications from development to production. This combination allows me to create repeatable, reliable processes for shipping software faster while reducing manual errors, improving collaboration, and maintaining a seamless, automated pipeline. By adopting this approach, I can focus more on development and innovation, knowing that deployment and infrastructure management are streamlined and automated. Although I’ve encountered errors pertaining to the permissions on copying the docker file and building the image, I was able to solve it as the building image process is good to go. I’ve learned how to use docker via ansible as it helps fulfill the objectives and expectations from this activity.