Name: Renier L. Lope	**Date Performed: 11/13/2024**
Course/Section: CPE212 - CPE31S2	**Date Submitted: 11/13/2024**
Instructor: Engr. Robin Valenzuela	**Semester and SY: 1st Sem(2024-2025)**

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)

@workstation

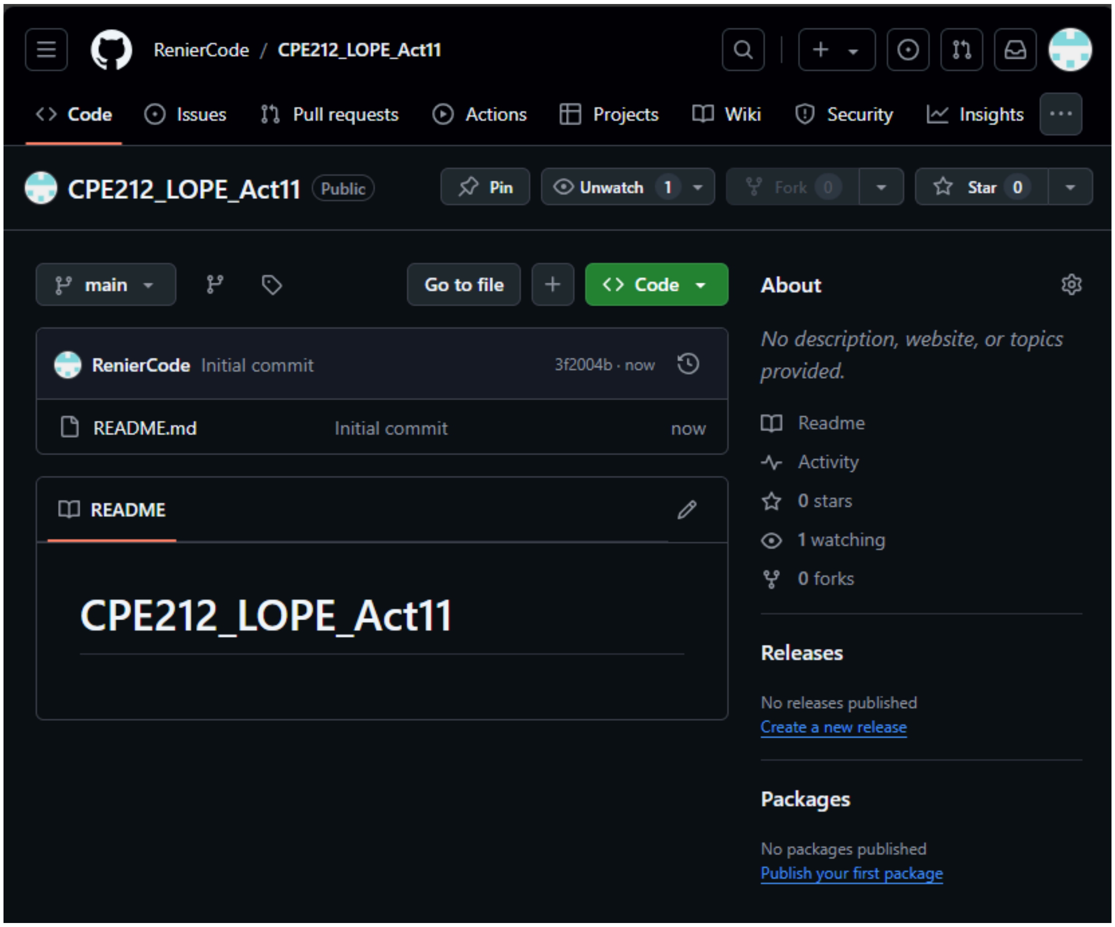
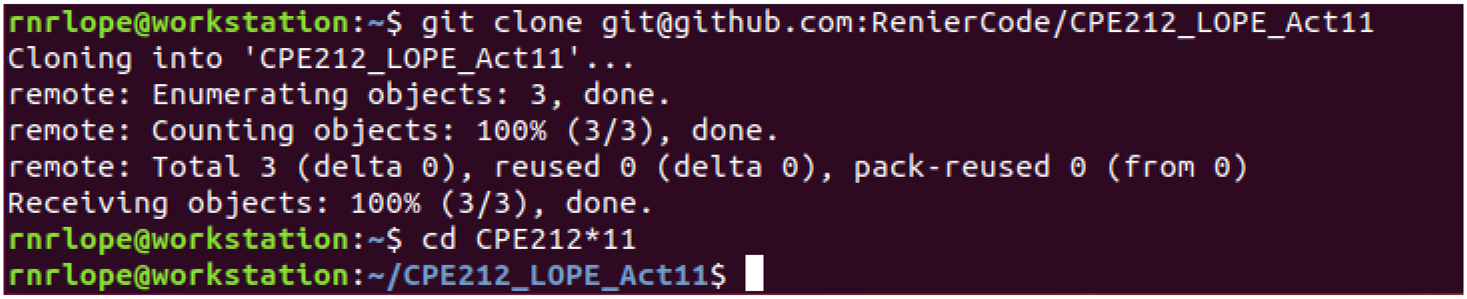
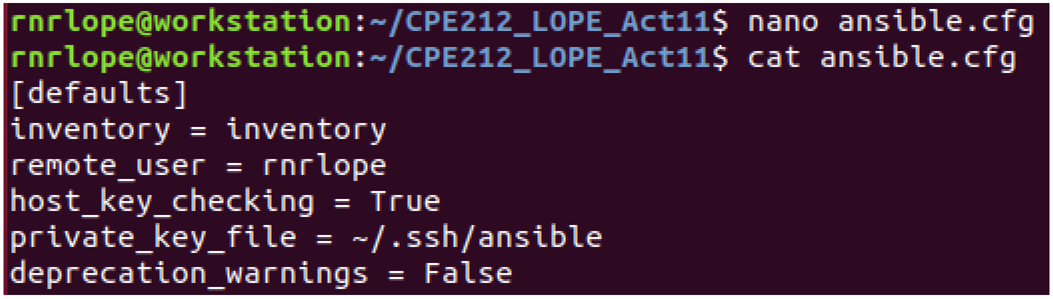


Figure 11.1: Create a new repository at github.



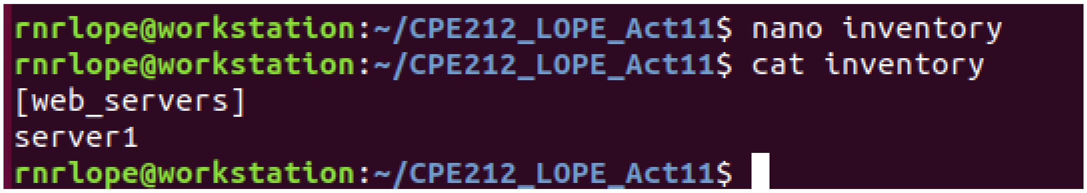
```
rnrlope@workstation:~$ git clone git@github.com:RenierCode/CPE212_LOPE_Act11
Cloning into 'CPE212_LOPE_Act11'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
rnrlope@workstation:~$ cd CPE212*11
rnrlope@workstation:~/CPE212_LOPE_Act11$
```

Figure 11.2: Use git clone to clone the repository in the local machine.



```
rnrlope@workstation:~/CPE212_LOPE_Act11$ nano ansible.cfg
rnrlope@workstation:~/CPE212_LOPE_Act11$ cat ansible.cfg
[defaults]
inventory = inventory
remote_user = rnrlope
host_key_checking = True
private_key_file = ~/.ssh/ansible
deprecation_warnings = False
```

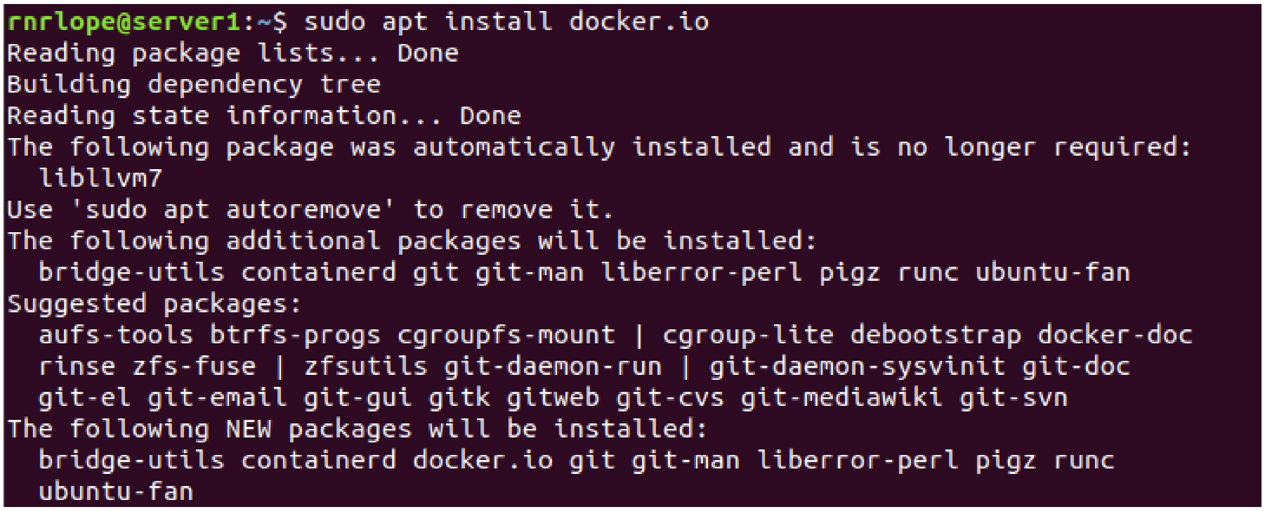
Figure 11.3: Create an ansible file.



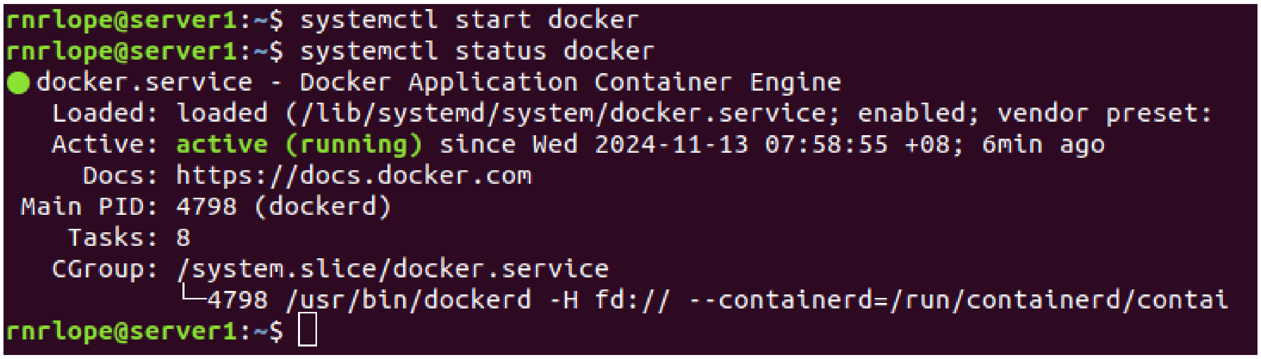
```
rnrlope@workstation:~/CPE212_LOPE_Act11$ nano inventory
rnrlope@workstation:~/CPE212_LOPE_Act11$ cat inventory
[web_servers]
server1
rnrlope@workstation:~/CPE212_LOPE_Act11$
```

Figure 11.4: Create an inventory file.

@server1



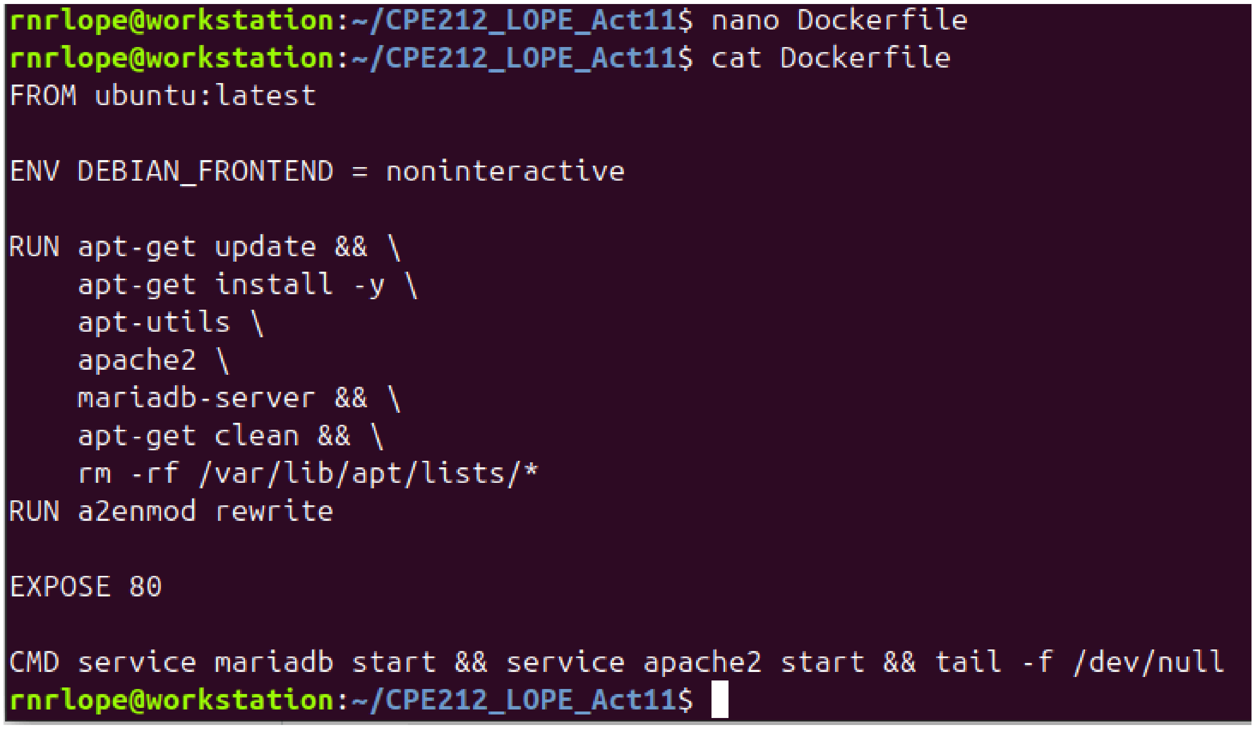
```
rnrlope@server1:~$ sudo apt install docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libllvm7
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  bridge-utils containerd git git-man liberror-perl pigz runc ubuntu-fan
Suggested packages:
  aufs-tools btrfs-progs cgroupfs-mount | cgroup-lite debootstrap docker-doc
  rinse zfs-fuse | zfsutils git-daemon-run | git-daemon-sysvinit git-doc
  git-el git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  bridge-utils containerd docker.io git git-man liberror-perl pigz runc
  ubuntu-fan
```



```
rnrlope@server1:~$ systemctl start docker
rnrlope@server1:~$ 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 07:58:55 +08; 6min ago
     Docs: https://docs.docker.com
 Main PID: 4798 (dockerd)
    Tasks: 8
   CGroup: /system.slice/docker.service
           └─4798 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/contai
rnrlope@server1:~$
```

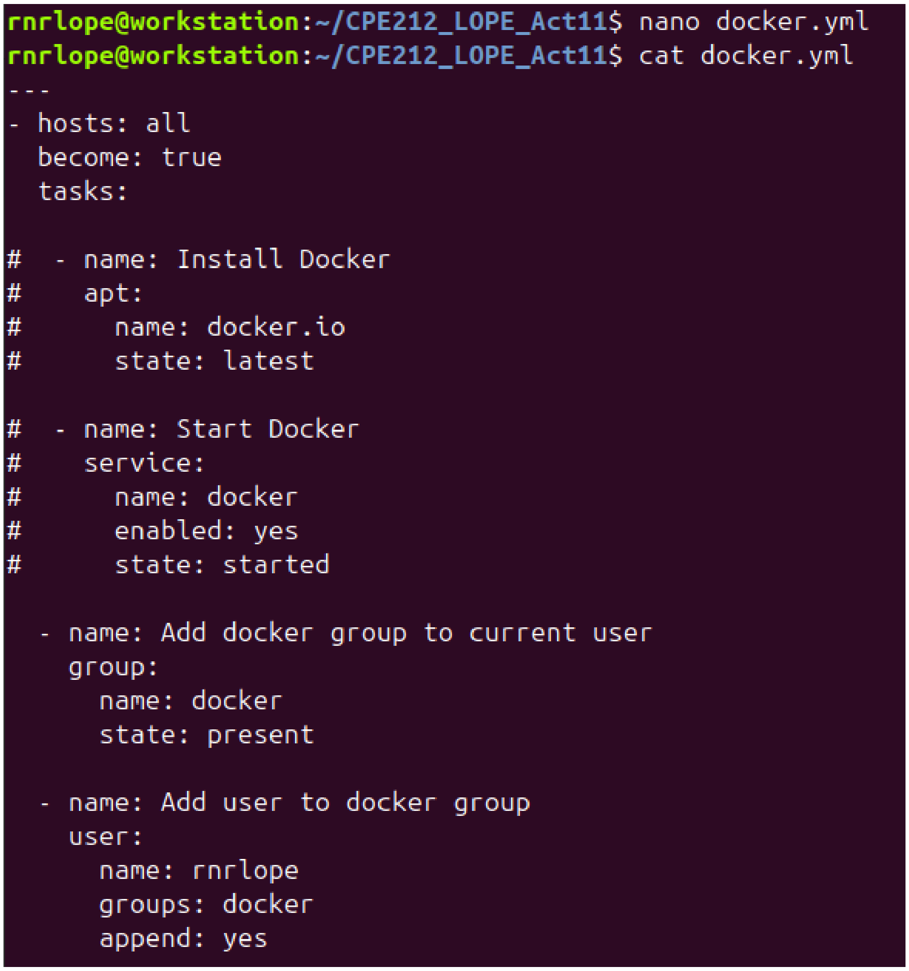
Figure 11.5 - 11.6: Install and Enable Docker at server1.

@workstation

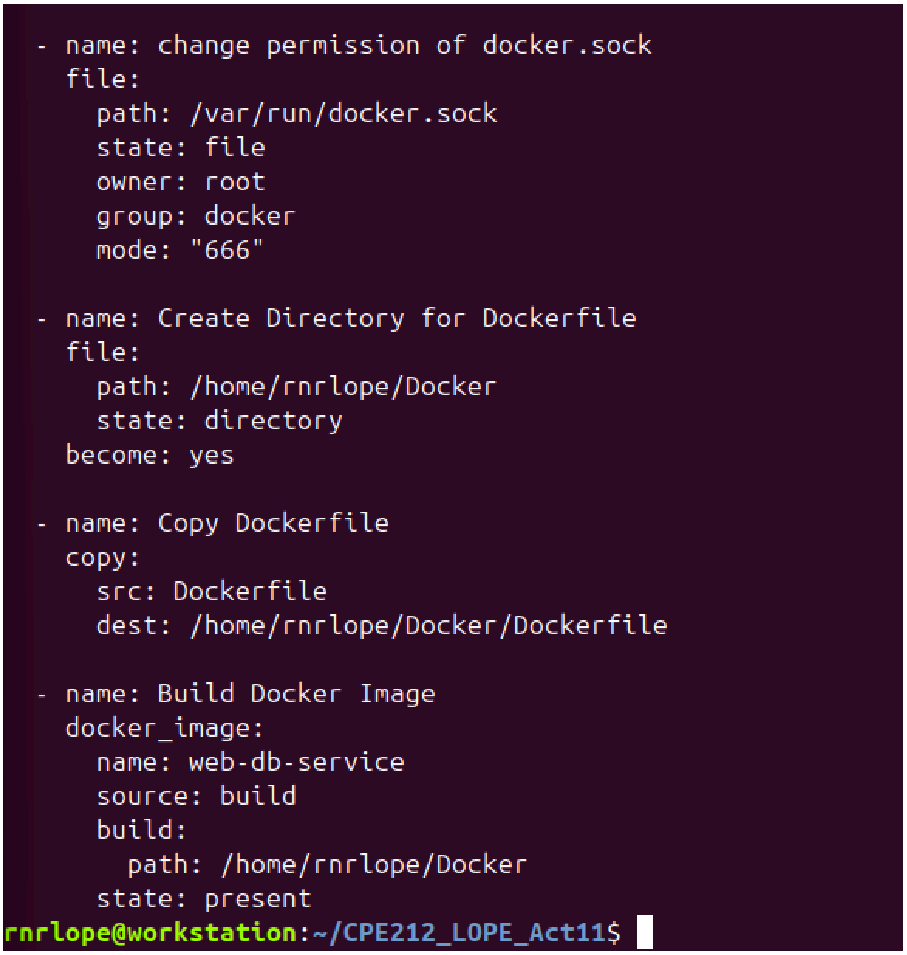


```
rnrlope@workstation:~/CPE212_LOPE_Act11$ nano Dockerfile
rnrlope@workstation:~/CPE212_LOPE_Act11$ cat Dockerfile
FROM ubuntu:latest

ENV DEBIAN_FRONTEND = noninteractive

RUN apt-get update && \
    apt-get install -y \
    apt-utils \
    apache2 \
    mariadb-server && \
    apt-get clean && \
    rm -rf /var/lib/apt/lists/*
RUN a2enmod rewrite

EXPOSE 80

CMD service mariadb start && service apache2 start && tail -f /dev/null
rnrlope@workstation:~/CPE212_LOPE_Act11$ 
```

Figure 11.7: Create a Dockerfile that will install and enable both apache2 and mariadb.

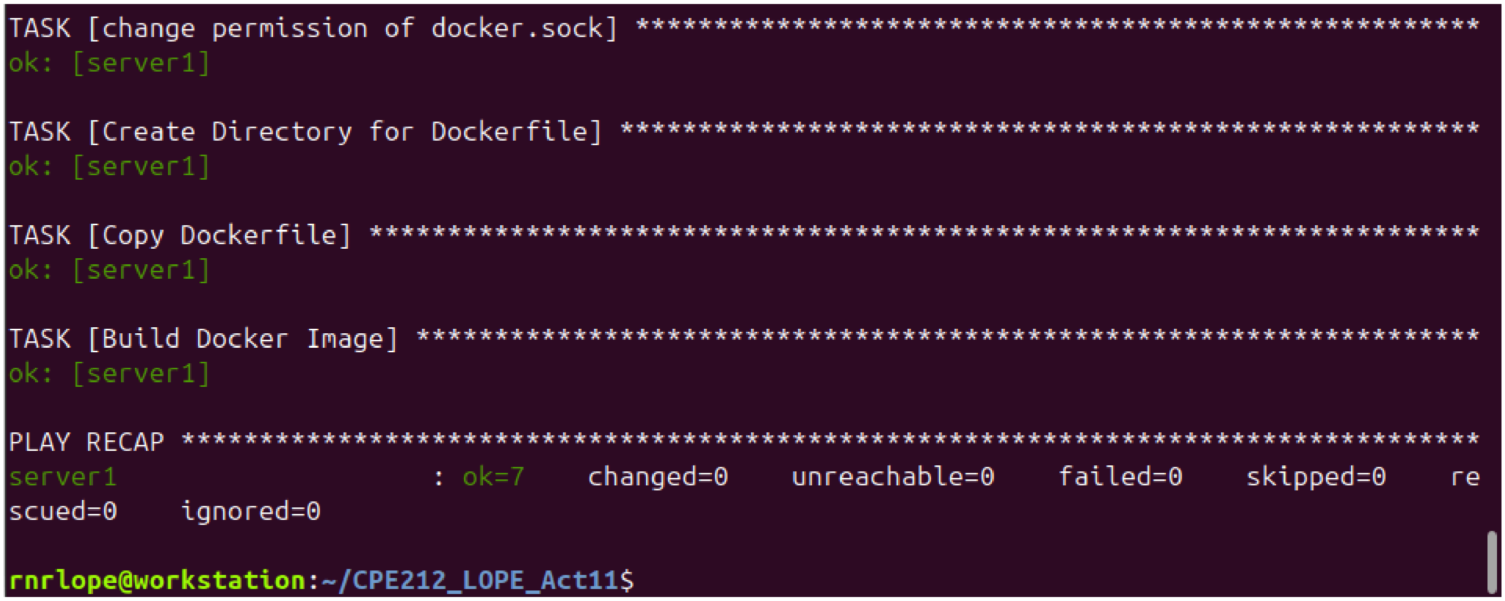


```
rnrlope@workstation:~/CPE212_LOPE_Act11$ nano docker.yml
rnrlope@workstation:~/CPE212_LOPE_Act11$ cat docker.yml
---
- hosts: all
  become: true
  tasks:

#   - name: Install Docker
#     apt:
#       name: docker.io
#       state: latest

#   - name: Start Docker
#     service:
#       name: docker
#       enabled: yes
#       state: started

    - name: Add docker group to current user
      group:
        name: docker
        state: present

    - name: Add user to docker group
      user:
        name: rnrlope
        groups: docker
        append: yes
```



```
- name: change permission of docker.sock
  file:
    path: /var/run/docker.sock
    state: file
    owner: root
    group: docker
    mode: "666"

- name: Create Directory for Dockerfile
  file:
    path: /home/rnrlope/Docker
    state: directory
  become: yes

- name: Copy Dockerfile
  copy:
    src: Dockerfile
    dest: /home/rnrlope/Docker/Dockerfile

- name: Build Docker Image
  docker_image:
    name: web-db-service
    source: build
    build:
      path: /home/rnrlope/Docker
    state: present
rnrlope@workstation:~/CPE212_LOPE_Act11$
```

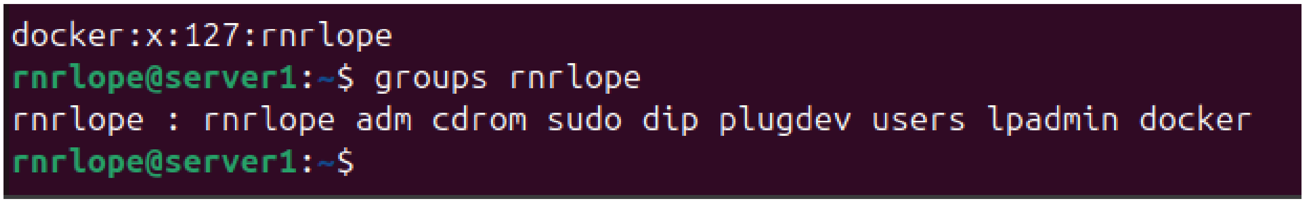
Figure 11.8 - 11.9: Create a playbook named “docker.yml” that will create a group named “docker” and add the user to the said group. This playbook will change the permission of “/var/run/docker.sock” so that we can use docker properly. This playbook is responsible for Copying the Dockerfile inside the Control node to the Manage Node, and also Building a Docker Image base on the copy of the Dockerfile inside the Managed Node.



```
TASK [change permission of docker.sock] ******************************************************
ok: [server1]

TASK [Create Directory for Dockerfile] *******************************************************
ok: [server1]

TASK [Copy Dockerfile] ***********************************************************************
ok: [server1]

TASK [Build Docker Image] ********************************************************************
ok: [server1]

PLAY RECAP ***********************************************************************************
server1                    : ok=7    changed=0    unreachable=0    failed=0    skipped=0    re
scued=0    ignored=0

rnrlope@workstation:~/CPE212_LOPE_Act11$
```

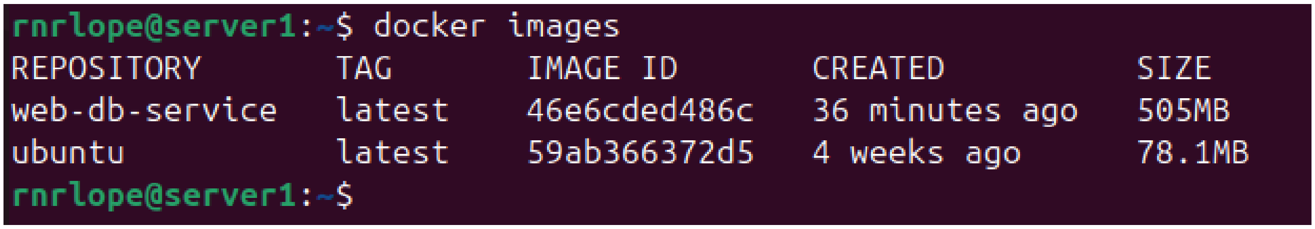
Figure 11.10: Play Recap of Executing the playbook “docker.yml”.

@server1



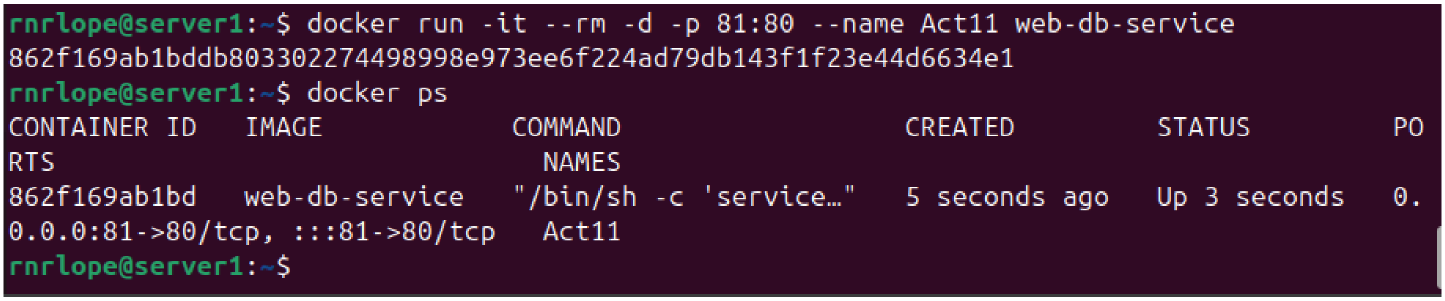
```
docker:x:127:rnrlope
rnrlope@server1:~$ groups rnrlope
rnrlope : rnrlope adm cdrom sudo dip plugdev users lpadmin docker
rnrlope@server1:~$
```

Figure 11.11: Verifying if docker is a group and if user is added to the docker group.



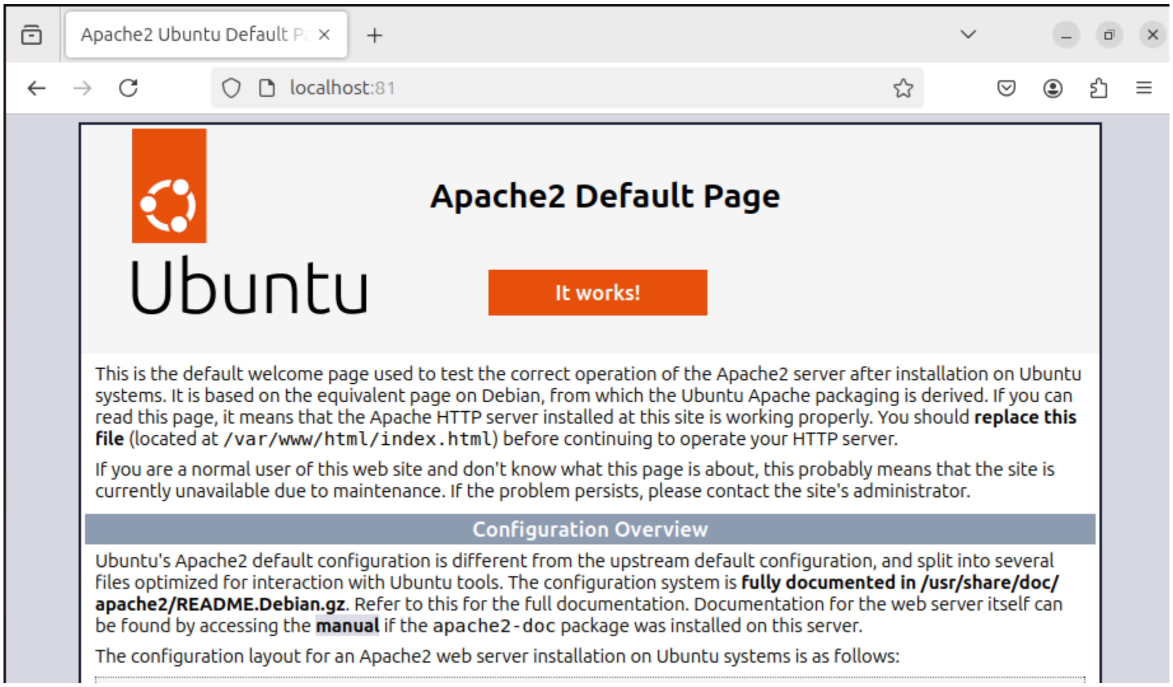
```
rnrlope@server1:~$ docker images
REPOSITORY        TAG       IMAGE ID       CREATED          SIZE
web-db-service    latest    46e6cded486c   36 minutes ago   505MB
ubuntu            latest    59ab366372d5   4 weeks ago      78.1MB
rnrlope@server1:~$
```

Figure 11.12: Verifying if the image is built.



```
rnrlope@server1:~$ docker run -it --rm -d -p 81:80 --name Act11 web-db-service
862f169ab1bddb803302274498998e973ee6f224ad79db143f1f23e44d6634e1
rnrlope@server1:~$ docker ps
CONTAINER ID   IMAGE            COMMAND                  CREATED         STATUS         PO
RTS                               NAMES
862f169ab1bd   web-db-service   "/bin/sh -c 'service…"   5 seconds ago   Up 3 seconds   0.
0.0.0:81->80/tcp, :::81->80/tcp   Act11
rnrlope@server1:~$
```

Figure 11.13: Creating a container based on the docker image. I used port 81 for the container.



Apache2 Default Page

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

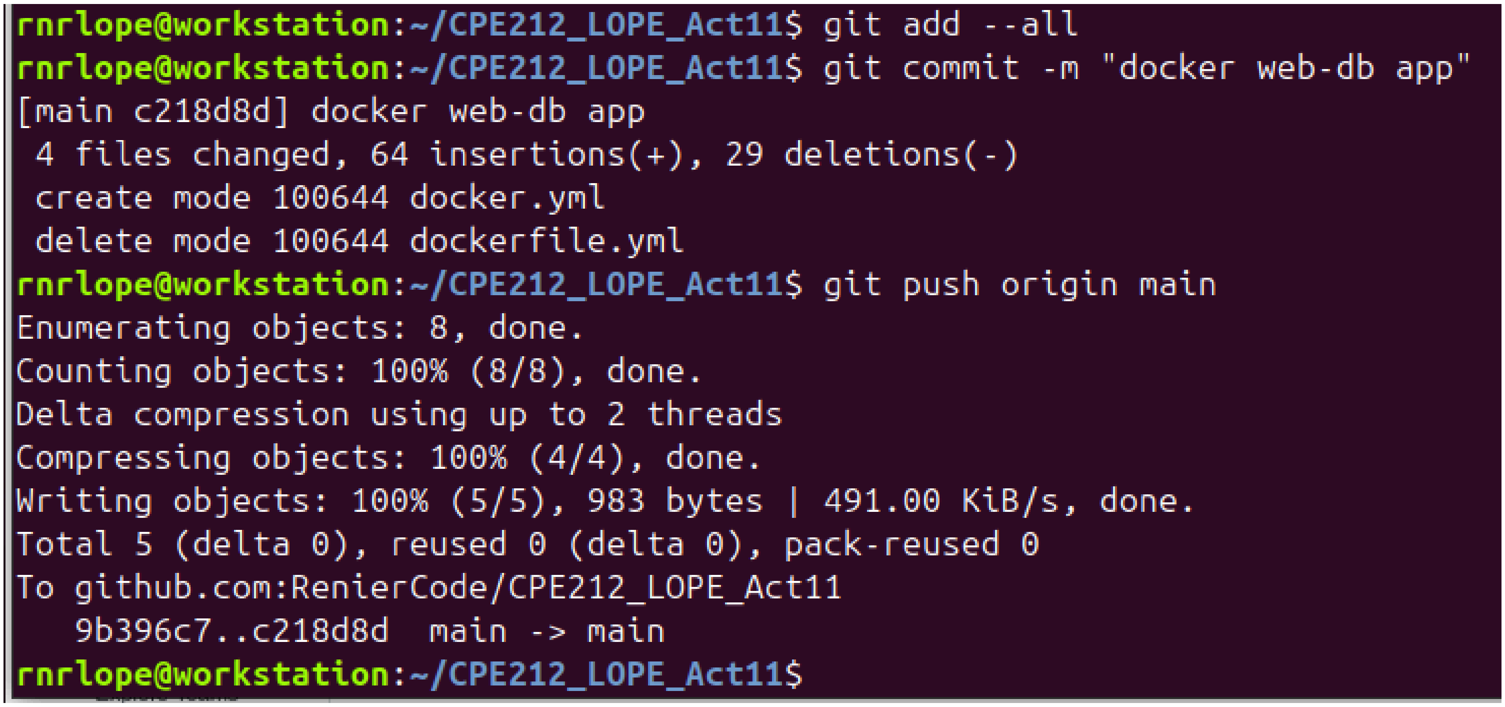
Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

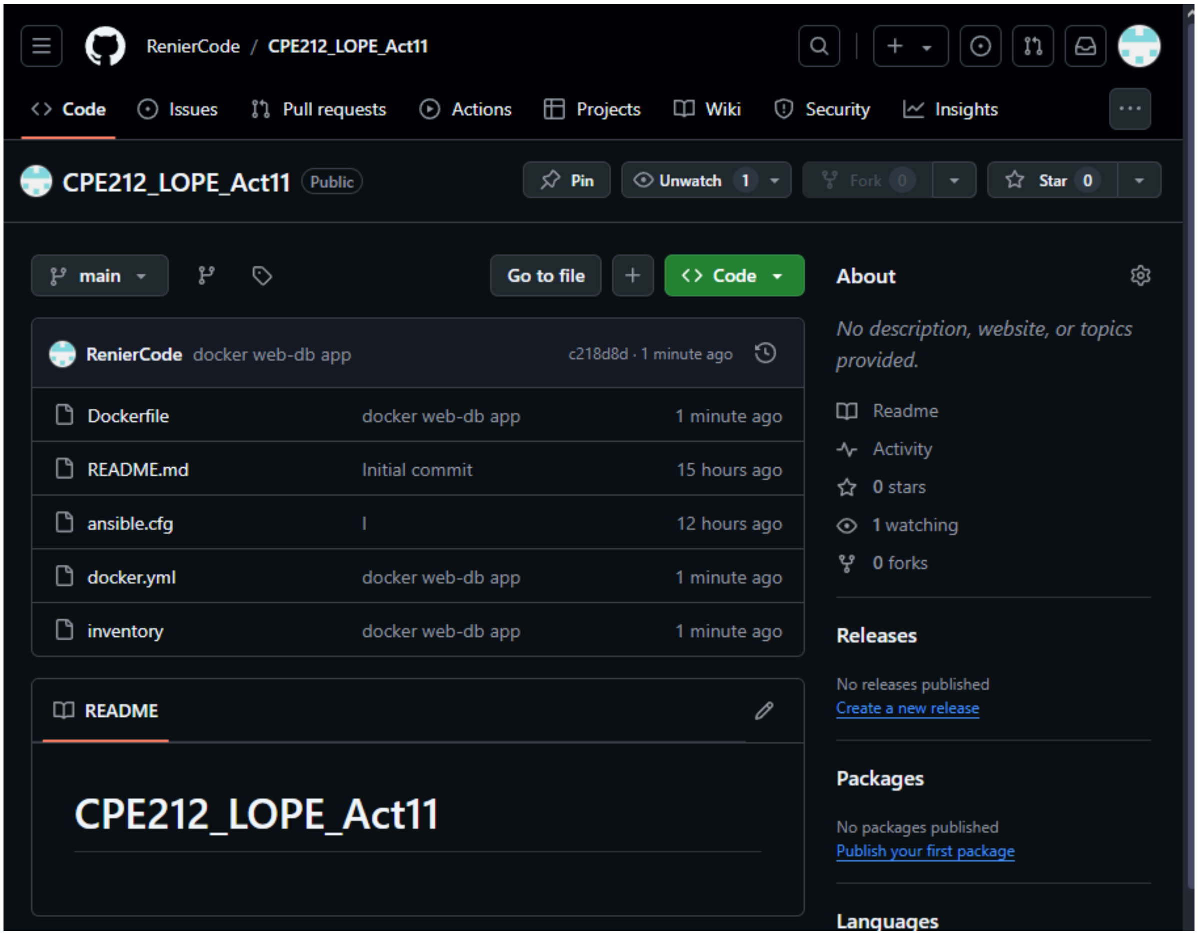
The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

Figure 11.14: Verifying if apache2 is running at port 81.

GIT PUSH:



```
rnrlope@workstation:~/CPE212_LOPE_Act11$ git add --all
rnrlope@workstation:~/CPE212_LOPE_Act11$ git commit -m "docker web-db app"
[main c218d8d] docker web-db app
 4 files changed, 64 insertions(+), 29 deletions(-)
 create mode 100644 docker.yml
 delete mode 100644 dockerfile.yml
rnrlope@workstation:~/CPE212_LOPE_Act11$ git push origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 2 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 983 bytes | 491.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:RenierCode/CPE212_LOPE_Act11
   9b396c7..c218d8d  main -> main
rnrlope@workstation:~/CPE212_LOPE_Act11$
```



RenierCode docker web-db app c218d8d · 1 minute ago

Dockerfile	docker web-db app	1 minute ago
README.md	Initial commit	15 hours ago
ansible.cfg	l	12 hours ago
docker.yml	docker web-db app	1 minute ago
inventory	docker web-db app	1 minute ago

README

CPE212_LOPE_Act11

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

GITHUB LINK:

https://github.com/RenierCode/CPE212_LOPE_Act11.git

Reflections:

Answer the following:

1. What are the benefits of implementing containerizations?
    - Containerization is a form of OS virtualization where you can run applications in contained user spaces called containers that use the same shared operating system making it portable. Software developers can create and deploy apps quickly and securely thanks to containerization. Developers refer to containers as “lightweight” because they share the host machine’s OS kernel and aren’t subjected to extra overhead. The scalability of application of container technology is high. By leveraging a service-oriented app design to reconfigure the current architecture to enable resources, an application container may manage growing workloads.

Conclusions:

- **In this activity, I managed to learn and demonstrate how to create a Docker Image based on a Dockerfile to the Manage Node which is the server1. I also managed to learn the basics of how to use Docker, I learned how to create a Dockerfile, Docker Image, and Docker Container. Lastly, I managed to use ansible as the tool to add a group and add the user to the said group and build a Docker Image to the Manage Node. Overall this activity helps me understand how Docker works and how to use it.**