

Name: Seruelas, Ronn Kristoper H.	Date Performed: 11-13-2023
Course/Section: CPE 232 - CPE31S4	Date Submitted: 11-14-2023
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	
<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: https://docs.docker.com/get-started/overview/</p> <p>You may also check the difference between containers and virtual machines. Click the link given below.</p> <p>Source: https://docs.microsoft.com/en-us/virtualization/windowscontainers/about/containers-vs-vm</p>	
3. Tasks	
<ol style="list-style-type: none"> 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)	
<ol style="list-style-type: none"> 1. Creation and preparation of Github Repository. 	

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner * TuRonnDraco / Repository name * CPE232_Act11_Seruelas
CPE232_Act11_Seruelas is available.

Great repository names are short and memorable. Need inspiration? How about [verbose-fiesta](#) ?

Description (optional)

☒ Public
Anyone on the internet can see this repository. You choose who can commit.

☐ Private
You choose who can see and commit to this repository.

Initialize this repository with:

☒ Add a README file
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore
.gitignore template: None
Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license
License: None
A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

This will set `main` as the default branch. Change the default name in your [settings](#).

ⓘ You are creating a public repository in your personal account.

[Create repository](#)

Figure 1.1 - Creation of Github Repository.

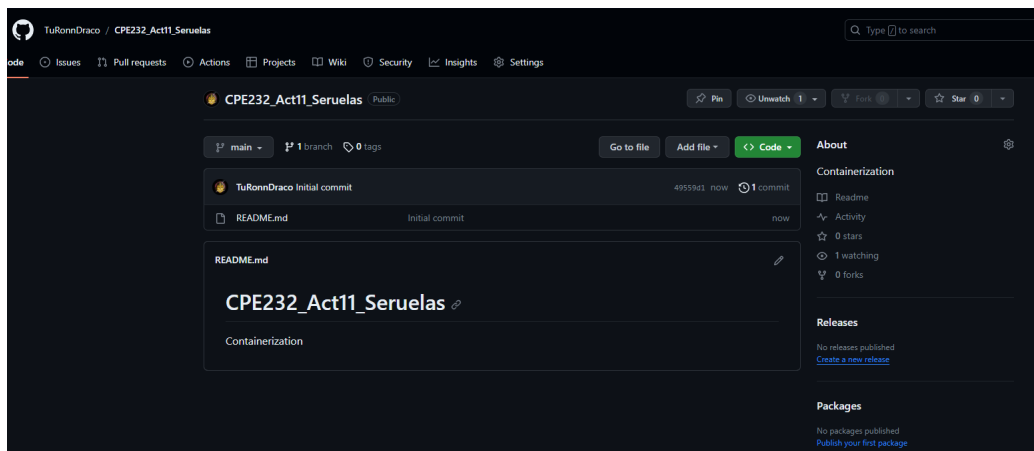
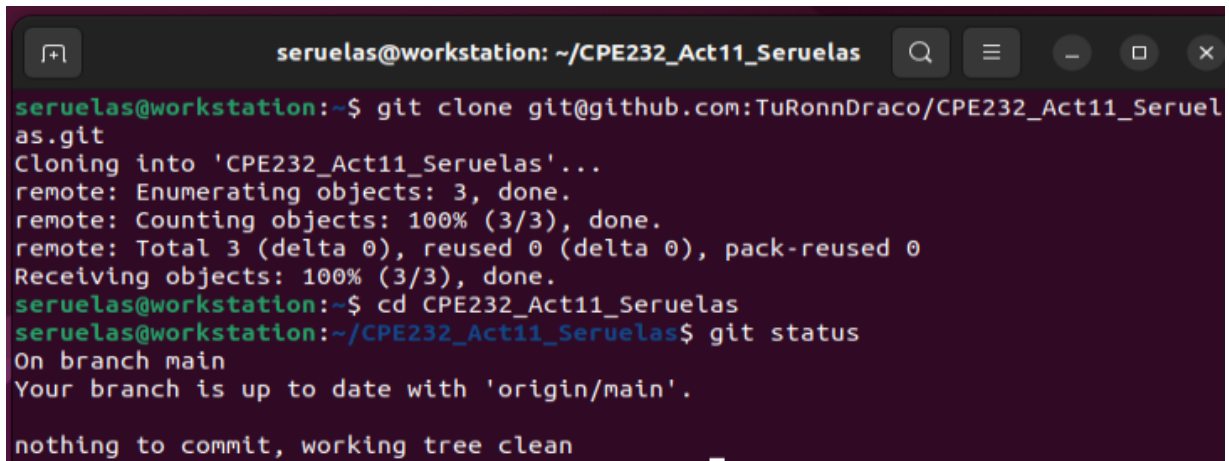


Figure 1.2 - Github Repository made.

2. Cloning of github repository to the local workstation.

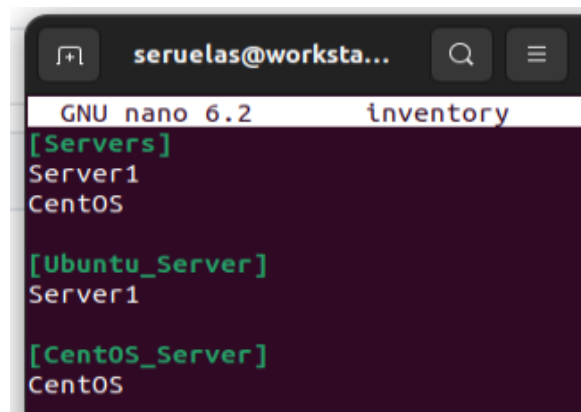


```
seruelas@workstation: ~/CPE232_Act11_Seruelas
seruelas@workstation:~$ git clone git@github.com:TuRonnDraco/CPE232_Act11_Seruel
as.git
Cloning into 'CPE232_Act11_Seruelas'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
seruelas@workstation:~$ cd CPE232_Act11_Seruelas
seruelas@workstation:~/CPE232_Act11_Seruelas$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

Figure 2.1 - Cloning of github repository to the local workstation or repository.

3. Configuration of the repository.

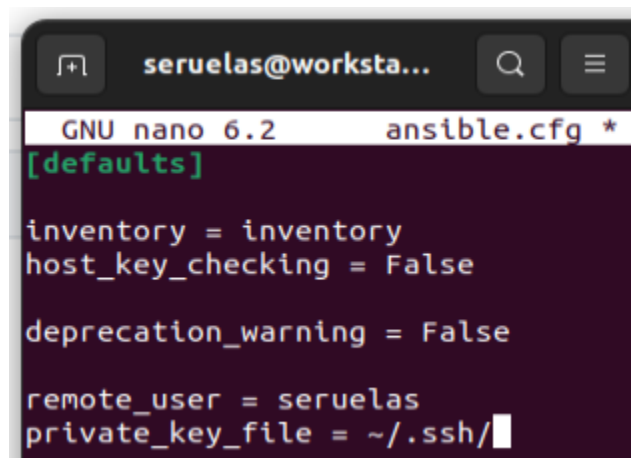


```
GNU nano 6.2 inventory
[Servers]
Server1
CentOS

[Ubuntu_Server]
Server1

[CentOS_Server]
CentOS
```

Figure 3.1 - Configuration of the inventory of the repository.



```
GNU nano 6.2 ansible.cfg *
[defaults]

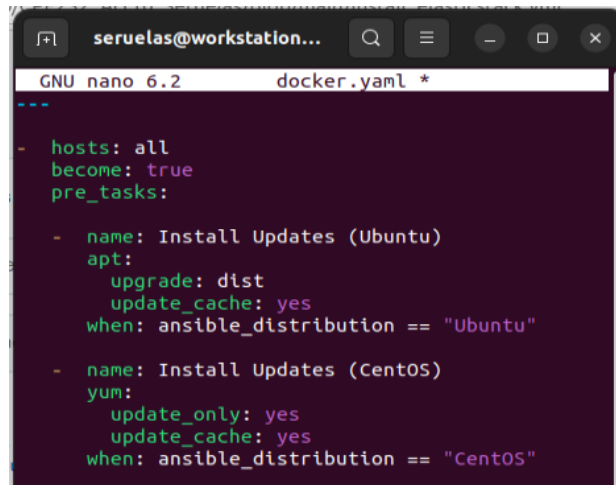
inventory = inventory
host_key_checking = False

deprecation_warning = False

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

Figure 3.2 - Configuration of the ansible.cfg of the repository.

4. Create the playbook that will hold all the installation and execution of docker.



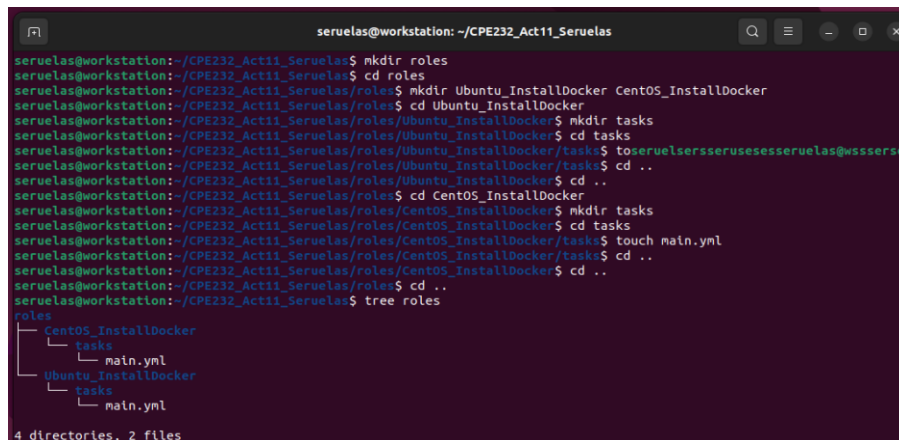
```
GNU nano 6.2 docker.yaml *
---
- hosts: all
  become: true
  pre_tasks:

    - 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"
```

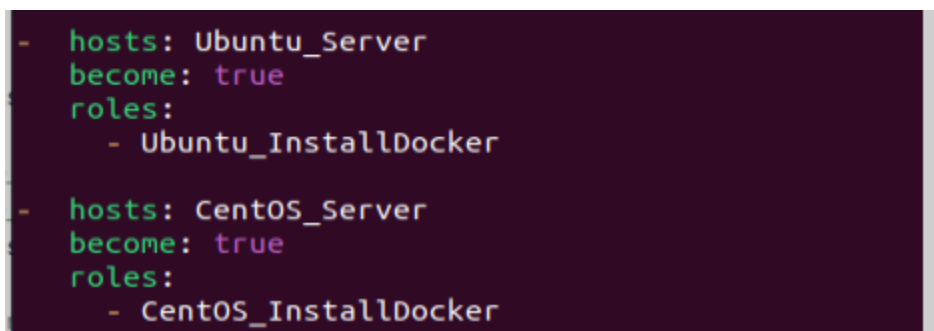
Figure 4.1 - Playbook created for the installation and execution for docker.

5. Create the roles needed for installation of docker and organization.



```
seruelas@workstation:~/CPE232_Act11_Seruelas$ mkdir roles
seruelas@workstation:~/CPE232_Act11_Seruelas$ cd roles
seruelas@workstation:~/CPE232_Act11_Seruelas/roles$ mkdir Ubuntu_InstallDocker CentOS_InstallDocker
seruelas@workstation:~/CPE232_Act11_Seruelas/roles$ cd Ubuntu_InstallDocker
seruelas@workstation:~/CPE232_Act11_Seruelas/roles/Ubuntu_InstallDocker$ mkdir tasks
seruelas@workstation:~/CPE232_Act11_Seruelas/roles/Ubuntu_InstallDocker$ cd tasks
seruelas@workstation:~/CPE232_Act11_Seruelas/roles/Ubuntu_InstallDocker/tasks$ touch main.yml
seruelas@workstation:~/CPE232_Act11_Seruelas/roles/Ubuntu_InstallDocker/tasks$ cd ..
seruelas@workstation:~/CPE232_Act11_Seruelas/roles$ cd CentOS_InstallDocker
seruelas@workstation:~/CPE232_Act11_Seruelas/roles/CentOS_InstallDocker$ mkdir tasks
seruelas@workstation:~/CPE232_Act11_Seruelas/roles/CentOS_InstallDocker$ cd tasks
seruelas@workstation:~/CPE232_Act11_Seruelas/roles/CentOS_InstallDocker/tasks$ touch main.yml
seruelas@workstation:~/CPE232_Act11_Seruelas/roles/CentOS_InstallDocker/tasks$ cd ..
seruelas@workstation:~/CPE232_Act11_Seruelas/roles$ cd ..
seruelas@workstation:~/CPE232_Act11_Seruelas$ tree roles
roles
├── CentOS_InstallDocker
│   └── tasks
│       └── main.yml
└── Ubuntu_InstallDocker
    └── tasks
        └── main.yml
4 directories, 2 files
```

Figure 5.1 - Creation of the necessary directories for each role.



```
- hosts: Ubuntu_Server
  become: true
  roles:
    - Ubuntu_InstallDocker

- hosts: CentOS_Server
  become: true
  roles:
    - CentOS_InstallDocker
```

Figure 5.2 - Inclusion of the roles in the playbook.

6. Create the main.yml for the Ubuntu role.

```
seruelas@workstation: ~/CPE232_Act11_Seruelas
GNU nano 6.2 roles/ubuntu_InstallDocker/tasks/main.yml

- name: Install Docker in Ubuntu
  apt:
    name: docker.io

- name: Start and Enable Docker in Ubuntu
  systemd:
    name: docker
    state: restarted
    enabled: true

- name: Add User to Docker Group
  user:
    name: "{{ ansible_user }}"
    groups: docker

- name: Create directory for Docker in Ubuntu
  file:
    path: ~/docker
    state: directory

- name: Create dockerfile in Docker Directory in Ubuntu
  file:
    path: ~/docker/dockerfile
    state: touch

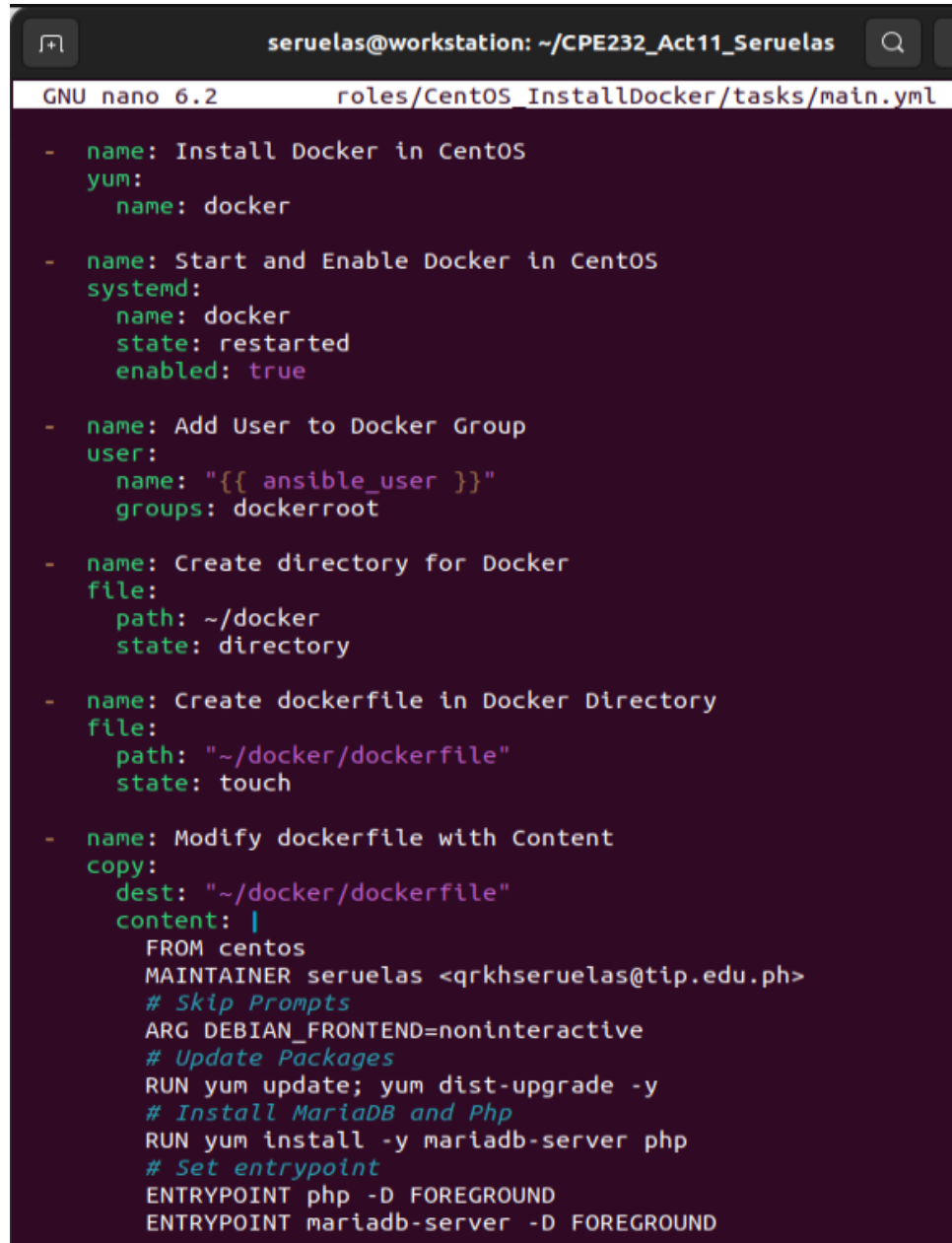
- name: Modify dockerfile with Content in Ubuntu
  copy:
    dest: ~/docker/dockerfile
    content: |
      FROM ubuntu
      MAINTAINER seruelas <qrkhseruelas@tip.edu.ph>
      # Skip Prompts
      ARG DEBIAN_FRONTEND=noninteractive
      # Update Packages
      RUN apt update; apt dist-upgrade -y
      # Install MariaDB and Apache2
      RUN apt install -y apache2
      RUN apt install -y mariadb-client-core-10.6
      # Set entrypoint
      ENTRYPOINT apache2ctl -D FOREGROUND

- name: Create container for apache2-mariadb in Ubuntu
  shell: |
    cd ~/docker
    docker build -t apache2-mariadb .

- name: Run container, apache2-mariadb in Ubuntu
  shell: |
    docker run -d -it -p 8080:80 apache2-mariadb
```

Figure 6.1 - Main.yml of the Ubuntu_InstallDocker role.

7. Create the main.yml for the CentOS role.

A screenshot of a terminal window with a dark background. The title bar shows the user 'seruelas@workstation' and the current directory '~/CPE232_Act11_Seruelas'. The terminal is running GNU nano 6.2, editing the file 'roles/CentOS_InstallDocker/tasks/main.yml'. The file contains a list of tasks for installing and configuring Docker on CentOS. The tasks include installing Docker via yum, starting and enabling the Docker service via systemd, adding the current user to the docker group, creating a directory for Docker, creating a Dockerfile, and copying specific content into the Dockerfile to set up a container environment with MariaDB and PHP.

```
seruelas@workstation: ~/CPE232_Act11_Seruelas
GNU nano 6.2 roles/CentOS_InstallDocker/tasks/main.yml

- name: Install Docker in CentOS
  yum:
    name: docker

- name: Start and Enable Docker in CentOS
  systemd:
    name: docker
    state: restarted
    enabled: true

- name: Add User to Docker Group
  user:
    name: "{{ ansible_user }}"
    groups: dockerroot

- name: Create directory for Docker
  file:
    path: ~/docker
    state: directory

- name: Create dockerfile in Docker Directory
  file:
    path: "~/docker/dockerfile"
    state: touch

- name: Modify dockerfile with Content
  copy:
    dest: "~/docker/dockerfile"
    content: |
      FROM centos
      MAINTAINER seruelas <qrkhseruelas@tip.edu.ph>
      # Skip Prompts
      ARG DEBIAN_FRONTEND=noninteractive
      # Update Packages
      RUN yum update; yum dist-upgrade -y
      # Install MariaDB and Php
      RUN yum install -y mariadb-server php
      # Set entrypoint
      ENTRYPOINT php -D FOREGROUND
      ENTRYPOINT mariadb-server -D FOREGROUND
```

Figure 8.1 - Main.yml for the CentOS_InstallDocker.

```
seruelas@workstation: ~/CPE232_Act11_Seruelas
seruelas@workstation:~/CPE232_Act11_Seruelas$ ansible-playbook --ask-become-pass docker.yaml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [Server1]
fatal: [CentOS]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host centos port 22: No route to host", "unreachable": true}

TASK [Install Updates (Ubuntu)] *****
ok: [Server1]

TASK [Install Updates (CentOS)] *****
skipping: [Server1]

PLAY [Ubuntu_Server] *****

TASK [Gathering Facts] *****
ok: [Server1]

TASK [Ubuntu_InstallDocker : Install Docker in Ubuntu] *****
ok: [Server1]

TASK [Ubuntu_InstallDocker : Start and Enable Docker in Ubuntu] *****
changed: [Server1]

TASK [Ubuntu_InstallDocker : Add User to Docker Group] *****
ok: [Server1]

TASK [Ubuntu_InstallDocker : Create directory for Docker in Ubuntu] *****
ok: [Server1]

TASK [Ubuntu_InstallDocker : Create dockerfile in Docker Directory in Ubuntu] ***
changed: [Server1]

TASK [Ubuntu_InstallDocker : Modify dockerfile with Content in Ubuntu] *****
ok: [Server1]

TASK [Ubuntu_InstallDocker : Create container for apache2-mariadb in Ubuntu] ***
changed: [Server1]

TASK [Ubuntu_InstallDocker : Run container, apache2-mariadb in Ubuntu] *****
changed: [Server1]

PLAY [CentOS_Server] *****

PLAY RECAP *****
CentOS      : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
Server1     : ok=11   changed=4    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

Figure 9.1 - Playbook recap for installation and execution of Docker (was unable to install to CentOS due to corruption of virtual machine)

```
Ubuntu Desktop S1 (Seruelas) [Running] - Oracle VM VirtualBox

Activities  Terminal  Nov 14 17:38

root@server1: ~

root@server1:~# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS
PORTS
e9790da5c845   apache2-mariadb  "/bin/sh -c 'apache2..."  9 seconds ago  Up 7 s
0.0.0.0:8080->80/tcp, :::8080->80/tcp  gallant_buck
root@server1:~#
```

Figure 10.1 - Successful execution of Docker in Ubuntu.

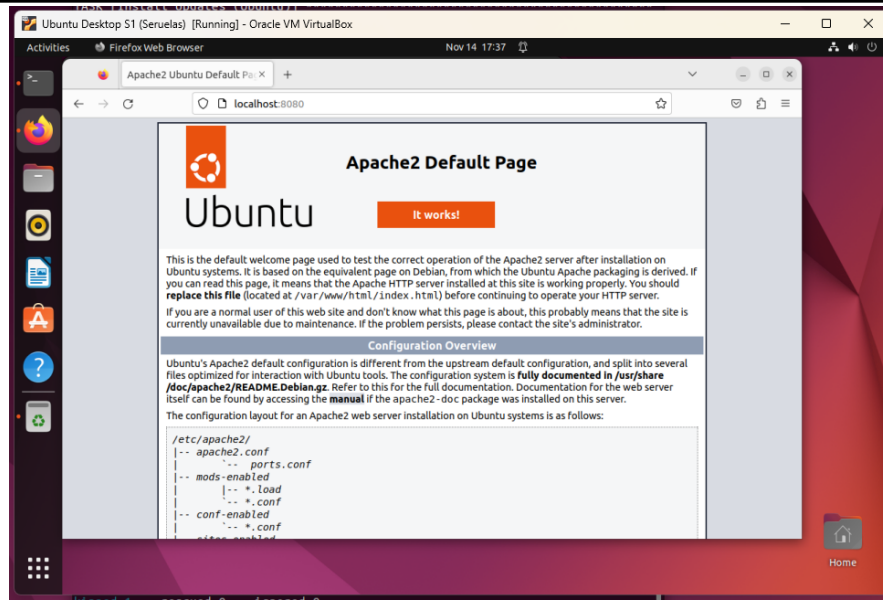


Figure 10.2 - Successful installation of Apache2 in Ubuntu.

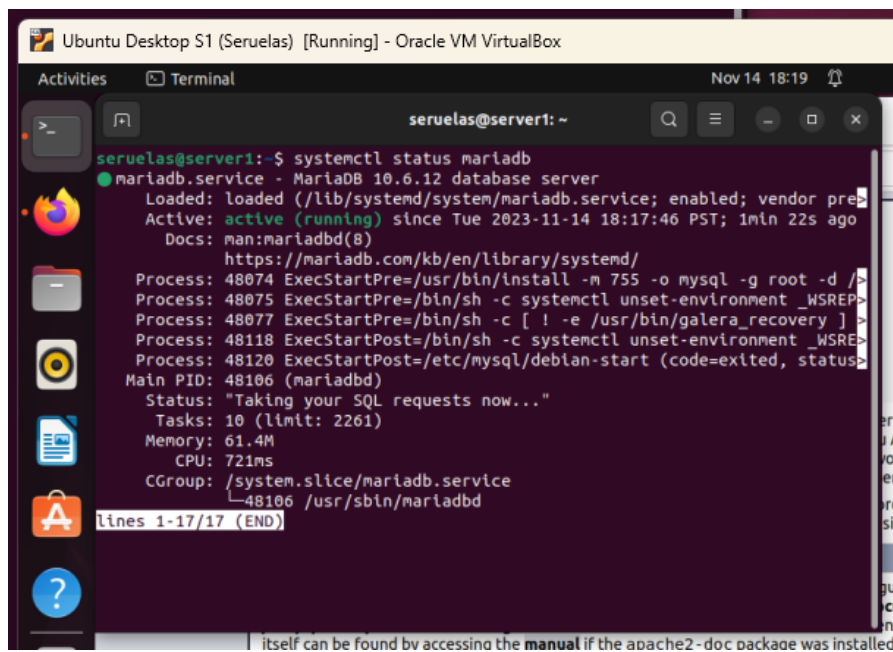


Figure 10.3 - Successful installation of Mariadb in Ubuntu.


```
seruelas@workstation: ~/CPE232_Act11_Seruelas
seruelas@workstation:~/CPE232_Act11_Seruelas$ git add *
seruelas@workstation:~/CPE232_Act11_Seruelas$ git commit -m "6:14pm at 11-14-2023"
[main 4cf0ca8] 6:14pm at 11-14-2023
 1 file changed, 12 insertions(+), 4 deletions(-)
seruelas@workstation:~/CPE232_Act11_Seruelas$ git push origin
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 2 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 620 bytes | 620.00 KiB/s, done.
Total 6 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To github.com:TuRonnDraco/CPE232_Act11_Seruelas.git
 a87116c..4cf0ca8  main -> main
```

Figure 11.1 - Saving all changes to the Github Repository

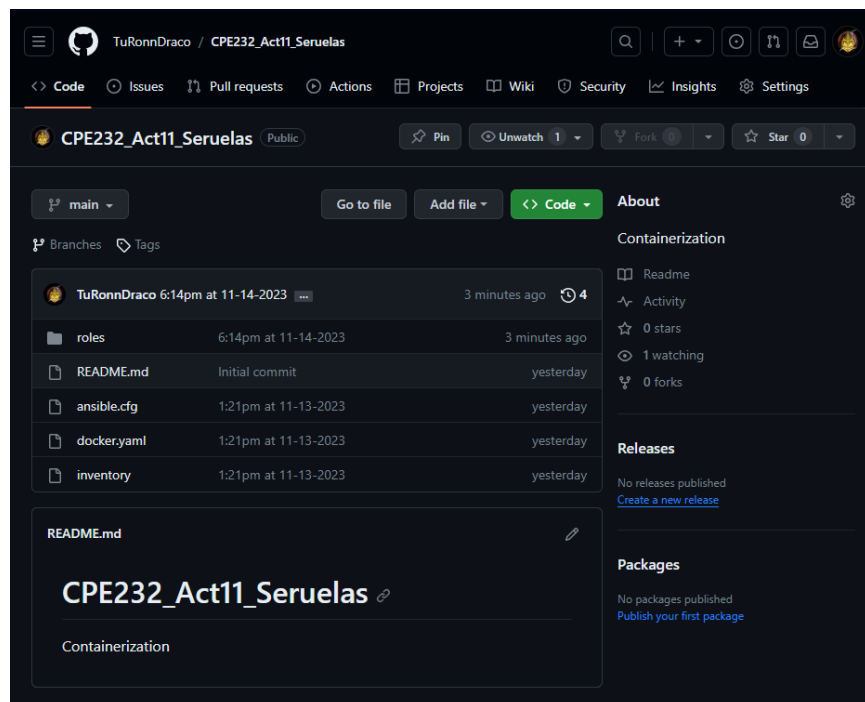


Figure 11.2 - Updated Github repository.

https://github.com/TuRonnDraco/CPE232_Act11_Seruelas

Reflections:

Answer the following:

1. What are the benefits of implementing containerizations?

- The benefits of implementing containerization is that it allows users or administrators to create a portable terminal or unit that allows the users to use images as products or lists of commands that contains each os. The implementation of containerization by administrators allows users to compile and to create a more compact application within their operating systems without changing too much in their workstation, allowing to create applications that may contain the useful products and services for them at needed times.

Conclusions:

In this activity, we were able to use and implement containerization in creation and installation of mariadb and apache2 in our Ubuntu and CentOS servers. We were able to learn on how much efficiency it gives the administrators to do their work and how compact and useful containerization in using such services and applications without modifying their workstations. To conclude this activity, we were able to implement containerization by creating a container in which it allows apache2 or php service to the servers and mariadb to the servers.