Name: Bernardo, Christian Emmanuel	Date Performed: 16/04/2024
Course/Section: CPE 232-CPE31S1	Date Submitted: 16/04/2024
Instructor: Dr. Jonathan Taylar	Semester and SY: 2nd, 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)

Create a new repository for this activity.

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 * Xerxes000 / **Repository name *** CPE232_Bernardo_ACT-11.1

CPE232_Bernardo_ACT-11.1 is available.

Great repository names are short and memorable. Need inspiration? How about upgraded-couscous ?

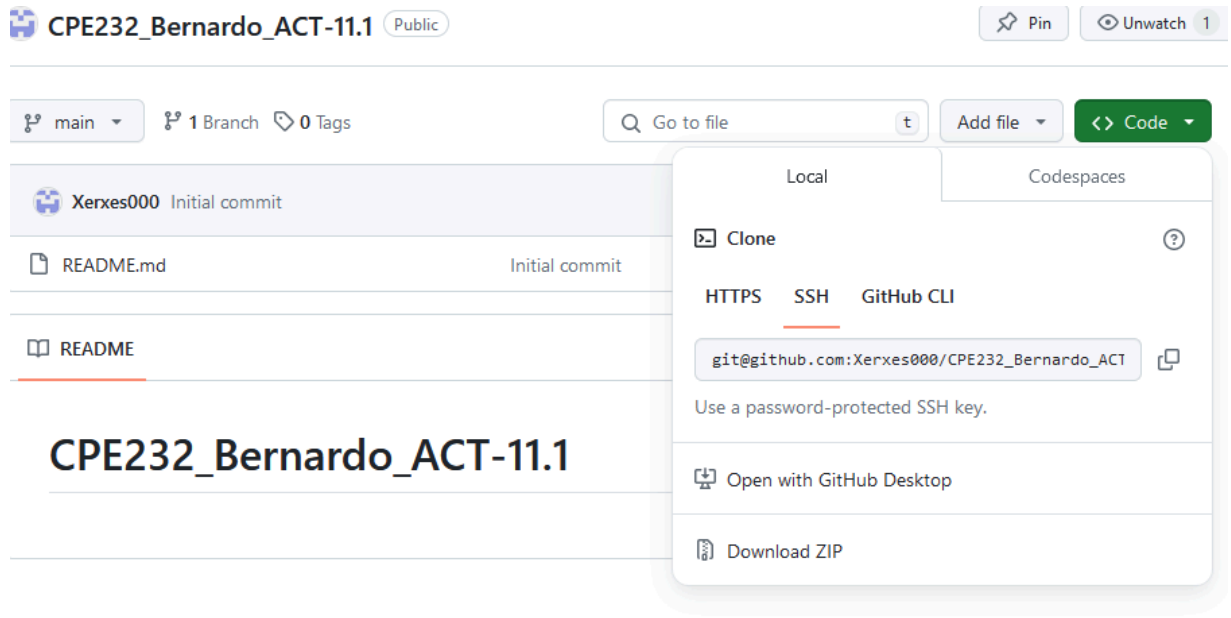
Description (optional)

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.



```
christian@workstation:~$ git clone git@github.com:Xerxes000/CPE232_Bernardo_ACT-
11.1.git
Cloning into 'CPE232_Bernardo_ACT-11.1'...
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.
christian@workstation:~$
```

```
christian@workstation:~/CPE232_Bernardo_ACT-11.1$ ls
ansible.cfg  dockerfile1  dockerfile2  dockerfile.yml  inventory  README.md
christian@workstation:~/CPE232_Bernardo_ACT-11.1$
```

```
christian@workstation:~/CPE232_Bernardo_ACT-11.1$ cat ansible.cfg
[defaults]

inventory = inventory
host_key_checking = false

deprecation_warnings = false

remote _user = christian
private_key_files = ~/.ssh/id_rsa


christian@workstation:~/CPE232_Bernardo_ACT-11.1$
```

```
christian@workstation:~/CPE232_Bernardo_ACT-11.1$ cat inventory
[UbuntuServer]
192.168.56.113

[CentOSServer]
192.168.56.108
christian@workstation:~/CPE232_Bernardo_ACT-11.1$
```

Install Docker and enable the docker socket.
Add to Docker group to your current user.
Create a Dockerfile to install web and DB server.

```
christian@workstation:~/CPE232_Bernardo_ACT-11.1$ cat dockerfile.yml
---
- hosts: all
  become: true
  pre_tasks:

  - name: install docker on Ubuntu
    shell:
      sudo apt-get install docker.io -y
    when: ansible_distribution == "Ubuntu"

  - name: install docker on CentOS
    yum:
      name: docker
      state: present
    when: ansible_distribution == "CentOS"

  - name: install docker sdk
    shell:
      pip3 install docker-py
```

```
  - name: start / enable docker service
    service:
      name: docker
      state: started
      enabled: true

  - name: add docker to user group
    shell:
      usermod -aG docker calderon

  - name: restart docker service
    service:
      name: docker
      state: restarted
      enabled: true

  - name: create dockerfile directory
    file:
      path: /root/demo-dockerfile
      state: directory
      owner: root
      group: root
      mode: '0755'
```

```
  - name: copy dockerfile for Ubuntu
    copy:
      src: dockerfile1
      dest: /root/demo-dockerfile/dockerfile
      owner: root
      group: root
      mode: '0755'
    when: ansible_distribution == "Ubuntu"

  - name: copy dockerfile for CentOS
    copy:
      src: dockerfile2
      dest: /root/demo-dockerfile/dockerfile
      owner: root
      group: root
      mode: '0755'
    when: ansible_distribution == "CentOS"

  - name: build docker image on Ubuntu
    shell:
      cmd: docker build -t docker_image /root/demo-dockerfile
    when: ansible_distribution == "Ubuntu"
```

```
  - name: build docker image on Ubuntu
    shell:
      cmd: docker build -t docker_image /root/demo-dockerfile
    when: ansible_distribution == "Ubuntu"

  - name: Remove existing container with conflicting name
    shell: docker rm -f docker_container5
    ignore_errors: yes  # Ignore errors if the container does not exist or
t running
    when: ansible_distribution == "Ubuntu"

  - name: build and run docker image on Ubuntu
    shell:
      cmd: docker run -d -p 8080 --name docker_container5 docker_image
    when: ansible_distribution == "Ubuntu"

christian@workstation:~/CPE232_Bernardo_ACT-11.1$
```

```
  - name: build docker image on Ubuntu
    shell:
      cmd: docker build -t docker_image /root/demo-dockerfile
    when: ansible_distribution == "Ubuntu"

  - name: Remove existing container with conflicting name
    shell: docker rm -f docker_container5
    ignore_errors: yes  # Ignore errors if the container does not exist or
t running
    when: ansible_distribution == "Ubuntu"

  - name: build and run docker image on Ubuntu
    shell:
      cmd: docker run -d -p 8080 --name docker_container5 docker_image
    when: ansible_distribution == "Ubuntu"

christian@workstation:~/CPE232_Bernardo_ACT-11.1$
```

```
christian@workstation:~/CPE232_Bernardo_ACT-11.1$ nano dockerfil
christian@workstation:~/CPE232_Bernardo_ACT-11.1$ cat dockerfile
FROM centos:latest
MAINTAINER christian  <christianbaenardo@gmail.com>

# skip prompts
ARG DEBIAN_FRONTEND=noninteractive

# update packages
RUN dnf -y install epel-release && dnf -y update

# install packages
RUN yum install -y httpd mariadb-server

# set entrypoint
ENTRYPOINT apache2ctl -D FOREGROUND
christian@workstation:~/CPE232_Bernardo_ACT-11.1$
```

Install and build the Dockerfile using Ansible.

```
christian@workstation:~/CPE232_Bernardo_ACT-11.1$ nano dockerfile.yml
christian@workstation:~/CPE232_Bernardo_ACT-11.1$ ansible-playbook --ask-become-pass dockerfile.yml
BECOME password:

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

TASK [Gathering Facts] ***********************************************************************************************
ok: [192.168.56.108]
ok: [192.168.56.109]

TASK [Install Python 3 and pip3] *************************************************************************************
ok: [192.168.56.108]
ok: [192.168.56.109]

TASK [install docker on Ubuntu] **************************************************************************************
[WARNING]: Consider using 'become', 'become_method', and 'become_user' rather than running sudo
changed: [192.168.56.109]
changed: [192.168.56.108]

TASK [install docker on CentOS] **************************************************************************************
skipping: [192.168.56.109]
skipping: [192.168.56.108]

TASK [install docker sdk] ********************************************************************************************
changed: [192.168.56.108]
changed: [192.168.56.109]

TASK [start / enable docker service] *********************************************************************************
ok: [192.168.56.108]
ok: [192.168.56.109]

TASK [add docker to user group] **************************************************************************************
changed: [192.168.56.109]
changed: [192.168.56.108]

TASK [restart docker service] ****************************************************************************************
changed: [192.168.56.108]
changed: [192.168.56.109]

TASK [create dockerfile directory] ***********************************************************************************
ok: [192.168.56.109]
ok: [192.168.56.108]

TASK [create dockerfile directory] ***********************************************************************************
ok: [192.168.56.109]
ok: [192.168.56.108]

TASK [copy dockerfile for Ubuntu] ************************************************************************************
ok: [192.168.56.109]
ok: [192.168.56.108]

TASK [copy dockerfile for CentOS] ************************************************************************************
skipping: [192.168.56.109]
skipping: [192.168.56.108]

TASK [Remove existing container with conflicting name] ***************************************************************
changed: [192.168.56.109]
changed: [192.168.56.108]

PLAY RECAP ***********************************************************************************************************
192.168.56.108             : ok=10   changed=5    unreachable=0    failed=0    skipped=2    rescued=0    ignore
192.168.56.109             : ok=10   changed=5    unreachable=0    failed=0    skipped=2    rescued=0    ignore

christian@workstation:~/CPE232_Bernardo_ACT-11.1$
```

Add, commit and push it to your repository.

```
christian@workstation:~/CPE232_Bernardo_ACT-11.1$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   ansible.cfg
        new file:   dockerfile.yml
        new file:   dockerfile1
        new file:   dockerfile2
        new file:   inventory

christian@workstation:~/CPE232_Bernardo_ACT-11.1$ git add *
christian@workstation:~/CPE232_Bernardo_ACT-11.1$ git commit -m 'Update'
[main b23928e] Update
 5 files changed, 120 insertions(+)
 create mode 100644 ansible.cfg
 create mode 100644 dockerfile.yml
 create mode 100644 dockerfile1
 create mode 100644 dockerfile2
 create mode 100644 inventory
```

```
christian@workstation:~/CPE232_Bernardo_ACT-11.1$ git push origin ma
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 1.35 KiB | 689.00 KiB/s, done.
Total 7 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To github.com:Xerxes000/CPE232_Bernardo_ACT-11.1.git
   9af3818..b23928e  main -> main
christian@workstation:~/CPE232_Bernardo_ACT-11.1$
```

Reflections:

Answer the following:

1. What are the benefits of implementing containerizations?
    - These are some of the benefits of containerization, firstly is its isolated state of in different applications that don't intrude in each other's settings. second it is very secure in protecting files in it but also in protecting the files outside its control because of its isolated state. Third, it is consistent in deploying applications because it packages its applications and dependencies together and its containers ensure that it consistently behaves across environments.

Conclusions:

- This module taught us about the definition, applications, and usage of a docker container. We can follow this module by using the provided presentation, which explains how to install Docker, implement it, and create images. We need to create and install Apache2 and MariaDB in addition to installing Docker in both Ubuntu and CentOS using Playbook for this task. After a while, everything was

fine overall, but there were a lot of errors that kept coming up and made it hard to debug and comprehend. I encountered a challenging scenario in which we needed to use Playbook to build a Docker image, but it kept crashing.