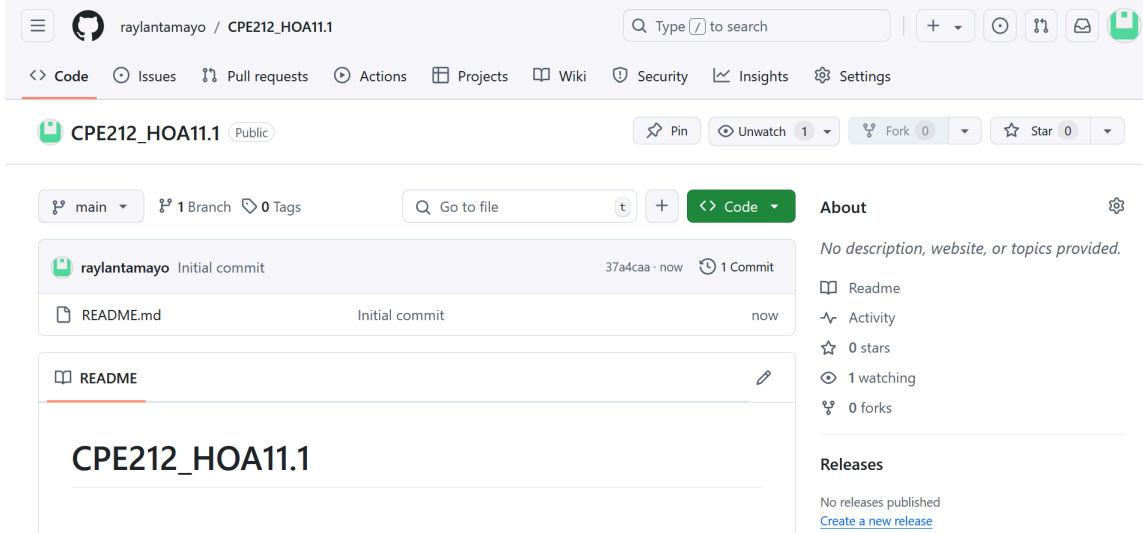


Name: Tamayo, Ray Lan A.	Date Performed: 11/15/2024
Course/Section: CPE 212-CPE31S21	Date Submitted: 11/15/2024
Instructor: Engr. Robin Valenzuela	Semester and SY: First 2024-2025
Activity 11: Containerization	
1. Objectives	
Create a Dockerfile and form a workflow using Ansible as Infrastructure as Code (IaC) to enable Continuous Delivery process	
2. Discussion	
<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)	
<p>Task 1: Create a File</p> <ol style="list-style-type: none"> 1. Create a new repository for this Hands-On Activity 	



2. Create the ansible.cfg and inventory file (must include one Ubuntu and CentOS)

```
tamayo@workstation: ~/CPE212_HOA11.1
File Edit View Search Terminal Help
GNU nano 2.9.3 inventory

[web_servers]
192.168.56.128

[db_servers]
192.168.56.131
```

```
tamayo@workstation: ~/CPE212_HOA11.1
File Edit View Search Terminal Help
GNU nano 2.9.3 ansible.cfg

[defaults]

inventory = inventory
host_key_checking = False

deprecation_warnings = False

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

Task 2: Install and Create a dockerfile

1. Install the docker.io in the local machine.

```
tamayo@workstation:~/CPE212_H0A11.1$ 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:
  liblvm2
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 0 not upgraded.
Need to get 65.7 MB of archives.
After this operation, 292 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ph.archive.ubuntu.com/ubuntu bionic/universe amd64 pigz amd64 2.4-1 [57.4 kB]
Get:2 http://ph.archive.ubuntu.com/ubuntu bionic/main amd64 bridge-utils amd64 1.5-15ubuntu1 [30.1 kB]
Get:3 http://ph.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 runc amd64 1.1.4-0ubuntu1~18.04.2 [3,822 kB]
Get:4 http://ph.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 containerd amd64 1.6.12-0ubuntu1~18.04.1 [31.5 MB]
Get:5 http://ph.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 docker.io amd64 20.10.21-0ubuntu1~18.04.3 [30.3 MB]
Get:6 http://ph.archive.ubuntu.com/ubuntu bionic/main amd64 ubuntu-fan all 0.12-0ubuntu1 [10.1 kB]
```

2. Show that the docker is now working in the local machine.

```
tamayo@workstation:~/CPE212_H0A11.1$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset:
   Active: active (running) since Fri 2024-11-15 08:28:41 +08; 2min 27s ago
     Docs: https://docs.docker.com
    Main PID: 5977 (dockerd)
      Tasks: 8
   CGroup: /system.slice/docker.service
           └─5977 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/contai

Nov 15 08:28:36 workstation dockerd[5977]: time="2024-11-15T08:28:36.820492606+
Nov 15 08:28:36 workstation dockerd[5977]: time="2024-11-15T08:28:36.820499613+
Nov 15 08:28:36 workstation dockerd[5977]: time="2024-11-15T08:28:36.820503135+
Nov 15 08:28:36 workstation dockerd[5977]: time="2024-11-15T08:28:36.820637781+
Nov 15 08:28:37 workstation dockerd[5977]: time="2024-11-15T08:28:37.526975681+
Nov 15 08:28:37 workstation dockerd[5977]: time="2024-11-15T08:28:37.995264101+
Nov 15 08:28:41 workstation dockerd[5977]: time="2024-11-15T08:28:41.036560411+
Nov 15 08:28:41 workstation dockerd[5977]: time="2024-11-15T08:28:41.036932103+
Nov 15 08:28:41 workstation systemd[1]: Started Docker Application Container En
Nov 15 08:28:41 workstation dockerd[5977]: time="2024-11-15T08:28:41.753093379+
lines 1-19/19 (END)
```

3. Enable the docker in the local machine.

```
tamayo@workstation:~/CPE212_H0A11.1$ sudo systemctl enable docker
```

4. Start the docker in the local machine.

```
tamayo@workstation:~/CPE212_H0A11.1$ sudo systemctl start docker
```

5. Create a docker for this activity.

```
tamayo@workstation: ~/CPE212_H0A11.1
File Edit View Search Terminal Help
GNU nano 2.9.3 dockerfile

FROM ubuntu

MAINTAINER raylantamayo <qrlatamayo@tip.edu.ph>

# Skip prompts
ARG DEBIAN_FRONTEND=noninteractive

# update packages
RUN apt update; apt dist-upgrade -y

# install packages
RUN apt install -y apache2 mariadb-server

# Setting the entrypoint
ENTRYPOINT apache2ctl -D FOREGROUND
ENTRYPOINT mariadb -D FOREGROUND
```

Task 3: Create Playbook for Installing Docker in Ubuntu and CentOS

1. Create a playbook and name it install_docker.yml.

tamayo@workstation: ~/CPE212_HOA11.1

File Edit View Search Terminal Help

GNU nano 2.9.3

install_docker.yml

```
- hosts: web_servers
  become: true
  pre_tasks:

    - name: dpkg for Ubuntu
      shell:
        dpkg --configure -a
      when: ansible_distribution == "Ubuntu"

    - name: Install Docker for Ubuntu
      apt:
        name: docker
        state: latest
      when: ansible_distribution == "Ubuntu"

    - name: Install SDK for Ubuntu
      shell:
        pip3 install docker-py

    - name: Add group to Docker
      shell:
        usermod -aG docker tamayo

- name: Enable and Restart Docker for Ubuntu
  service:
    name: docker
    state: started
    enabled: true

- name: Creating Directory for Dockerfile
  file:
    path: ./root/demo-dockerfile
    state: directory
    owner: root
    group: root

- name: Import Dockerfile
  copy:
    src: ./dockerfile
    dest: ./root/demo-dockerfile/dockerfile
    owner: root
    group: root
    mode: '0755'

- hosts: db_servers
  become: true
```

```
pre_tasks:

  - name: Install all required packages
    dnf:
      name:
        - yum-utils
        - device-mapper-persistent-data
        - lvm2
      state: present

  - name: Add Docker repository
    yum_repository:
```

```
  name: docker-ce
  description: Docker CE Stable - $basearch
  baseurl: https://download.docker.com/linux/centos/7/$basearch/stable
  gpgkey: https://download.docker.com/linux/centos/gpg
  enabled: yes

  - name: Install Docker for CentOS
    dnf:
      name: docker-ce
      state: present

  - name: Start and enable Docker service for CentOS
```

```
systemd:
  name: docker
  state: started
  enabled: yes
```

Task 4: Run and Verify

1. Run the command `ansible-playbook - - ask-become-pass install_docker.yml` to completely install it in both Ubuntu server and CentOS.

```

tamayo@workstation: ~/CPE212_HOA11.1
File Edit View Search Terminal Help
tamayo@workstation:~/CPE212_HOA11.1$ ansible-playbook --ask-become-pass install
_docker.yml
SUDO password:

PLAY [web_servers] *****
*

TASK [Gathering Facts] *****
*
ok: [192.168.56.128]

TASK [dpkg for Ubuntu] *****
*
changed: [192.168.56.128]

TASK [Install Docker for Ubuntu] *****
*
ok: [192.168.56.128]

TASK [Install SDK for Ubuntu] *****
*
changed: [192.168.56.128]

TASK [Add group to Docker] *****
*
changed: [192.168.56.128]

TASK [Enable and Restart Docker for Ubuntu] *****
*
ok: [192.168.56.128]

TASK [Creating Directory for Dockerfile] *****
*
ok: [192.168.56.128]

TASK [Import Dockerfile] *****
*
ok: [192.168.56.128]

```

2. Show the screenshot of the systemctl status in both Server 2 and CentOS. The status should be active.

CENTOS

```

● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)
   Active: active (running) since Thu 2023-11-16 04:46:12 EST; 12min ago
     Docs: https://docs.docker.com
    Main PID: 10518 (dockerd)
      Tasks: 8
     Memory: 44.9M
    CGroup: /system.slice/docker.service
            └─10518 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd...

Nov 16 04:46:10 localhost.localdomain systemd[1]: Starting Docker Application Conta...
Nov 16 04:46:10 localhost.localdomain dockerd[10518]: time="2023-11-16T04:46:10.097..."
Nov 16 04:46:10 localhost.localdomain dockerd[10518]: time="2023-11-16T04:46:10.448..."
Nov 16 04:46:11 localhost.localdomain dockerd[10518]: time="2023-11-16T04:46:11.804..."
Nov 16 04:46:11 localhost.localdomain dockerd[10518]: time="2023-11-16T04:46:11.916..."
Nov 16 04:46:11 localhost.localdomain dockerd[10518]: time="2023-11-16T04:46:11.989..."
Nov 16 04:46:11 localhost.localdomain dockerd[10518]: time="2023-11-16T04:46:11.990..."
Nov 16 04:46:12 localhost.localdomain dockerd[10518]: time="2023-11-16T04:46:12.023..."
Nov 16 04:46:12 localhost.localdomain systemd[1]: Started Docker Application Contai...
Hint: Some lines were ellipsized, use -l to show in full.

```

UBUNTU

```
tamayo@workstation:~/CPE212_HOA11.1$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset:
   Active: active (running) since Fri 2024-11-15 08:28:41 +08; 40min ago
     Docs: https://docs.docker.com
    Main PID: 5977 (dockerd)
      Tasks: 8
    CGroup: /system.slice/docker.service
            └─5977 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/contai

Nov 15 08:28:36 workstation dockerd[5977]: time="2024-11-15T08:28:36.820492606+
Nov 15 08:28:36 workstation dockerd[5977]: time="2024-11-15T08:28:36.820499613+
Nov 15 08:28:36 workstation dockerd[5977]: time="2024-11-15T08:28:36.820503135+
Nov 15 08:28:36 workstation dockerd[5977]: time="2024-11-15T08:28:36.820637781+
Nov 15 08:28:37 workstation dockerd[5977]: time="2024-11-15T08:28:37.526975681+
Nov 15 08:28:37 workstation dockerd[5977]: time="2024-11-15T08:28:37.995264101+
Nov 15 08:28:41 workstation dockerd[5977]: time="2024-11-15T08:28:41.036560411+
Nov 15 08:28:41 workstation dockerd[5977]: time="2024-11-15T08:28:41.036932103+
Nov 15 08:28:41 workstation systemd[1]: Started Docker Application Container En
Nov 15 08:28:41 workstation dockerd[5977]: time="2024-11-15T08:28:41.753093379+

[3]+  Stopped                  sudo systemctl status docker
```

DOCKER IMAGES

```
tamayo@workstation:~/CPE212_HOA11.1$ sudo docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
python              latest             c41ea8273365       3 weeks ago        1.02GB
mysql               latest            10db11fef9ce       4 weeks ago        602MB
ubuntu              latest            59ab366372d5       4 weeks ago        78.1MB
redis               latest            6c199afc1dae       5 weeks ago        117MB
nginx               latest            60c8a892f36f       6 weeks ago        192MB
alpine              latest            63b790fccc90       2 months ago       7.8MB
```

3. Upload it in the github.

```
tamayo@workstation:~/CPE212_HOA11.1$ git add .
tamayo@workstation:~/CPE212_HOA11.1$ git commit -m "HOA11.1"
[main ce90c02] HOA11.1
 5 files changed, 106 insertions(+)
 create mode 100644 ansible.cfg
 create mode 100644 dockerfile
 create mode 100644 install_docker.retry
 create mode 100644 install_docker.yml
 create mode 100644 inventory
tamayo@workstation:~/CPE212_HOA11.1$ git push origin main
Counting objects: 7, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (7/7), 1.35 KiB | 1.35 MiB/s, done.
Total 7 (delta 0), reused 0 (delta 0)
To github.com:raylantamayo/CPE212_HOA11.1.git
 37a4caa..ce90c02  main -> main
```


The screenshot shows a GitHub repository page for 'CPE212_HOA11.1' by user 'raylantamayo'. The repository is public and has 1 branch and 0 tags. The file list includes README.md, ansible.cfg, dockerfile, install_docker.retry, install_docker.yml, and inventory, all committed by raylantamayo at HOA11.1. The README file is selected, showing the title 'CPE212_HOA11.1'. The right sidebar contains sections for About, Releases, and Packages, all indicating no content is provided yet.

GITHUB LINK: https://github.com/raylantamayo/CPE212_HOA11.1.git

Reflections:

Answer the following:

1. What are the benefits of implementing containerizations?

At Ubuntu and CentOS containerization provides three major benefits. In the first place, it adds scale because applications can run in the same way in different environments. Second, it makes deployment more efficient by making it faster, as well as more reliable, thanks to the use of isolated containers. Finally, it increases resource efficiency because many containers operate in one host OS kernel, whereas virtual machines do not. Combined, these features are more straightforward to implement, easier to port and efficient in resource usage in a client side manner for Ubuntu and CentOS.

Conclusions:

In this activity I was able to meet docker as well as containerization. This activity ended as quite an eye-opener as much as Dockerfile creation and playbook deployment into both Ubuntu and CentOS. Creating Dockerfiles to package the apps was among the most valuable lessons that I acquired

because it made deployment easy. Running playbooks put an additional level of automation into motion, which was helpful in terms of time and energy. The weekly switch between the two operating systems increased my flexibility. Watching containers go through their operations was really encouraging especially given the perfect synchronization. This practical experience did not only refine my technical aptitude and actively ignite my passion regarding the optimization of possible following projects by means of Docker and playbooks.