Name: Daniela Marie D. Rabang	**Date Performed:** 11/13/2023
Course/Section: CPE232/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

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.

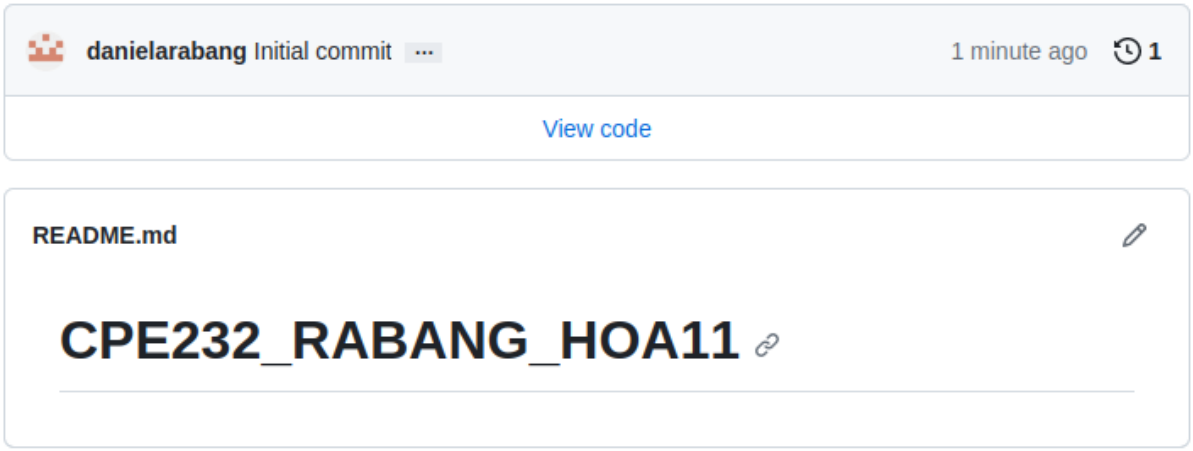
Source: https://docs.microsoft.com/en-us/virtualization/windowscontainers/about/containers-vs-vm

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 repository.



Clone the repository.

```
daniela@workstation:~$ git clone https://github.com/danielarabang/CPE232_RABANG_HOA11.git
Cloning into 'CPE232_RABANG_HOA11'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
```

Create an inventory file.

```
  GNU nano 2.9.3                                                inventory

[Ubuntu]
192.168.56.110  ansible_python_interpreter=/usr/bin/python3

[CentOS]
192.168.56.112  ansible_python_interpreter=/usr/bin/python
```

Create an ansible.cfg file.

```
  GNU nano 2.9.3                                                ansible.cfg

[defaults]

inventory = inventory
host_key checking = False

deprecation_warning = False

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

Install Docker.

```
daniela@workstation:~/CPE232_RABANG_HOA11$ sudo apt install docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
docker.io is already the newest version (20.10.21-0ubuntu1~18.04.3).
The following package was automatically installed and is no longer required:
  libllvm7
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

Enable the docker

```
daniela@workstation:~/CPE232_RABANG_HOA11$ sudo systemctl start docker
daniela@workstation:~/CPE232_RABANG_HOA11$ sudo systemctl enable docker
daniela@workstation:~/CPE232_RABANG_HOA11$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2023-11-13 12:08:30 PST; 12min ago
     Docs: https://docs.docker.com
 Main PID: 1073 (dockerd)
    Tasks: 16
   CGroup: /system.slice/docker.service
           └─1073 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Nov 13 12:08:28 workstation dockerd[1073]: time="2023-11-13T12:08:28.330184936+08:00" level=info msg="Removin
Nov 13 12:08:28 workstation dockerd[1073]: time="2023-11-13T12:08:28.345709670+08:00" level=warning msg="Erro
Nov 13 12:08:29 workstation dockerd[1073]: time="2023-11-13T12:08:29.385444907+08:00" level=info msg="Removin
Nov 13 12:08:29 workstation dockerd[1073]: time="2023-11-13T12:08:29.405078175+08:00" level=warning msg="Erro
Nov 13 12:08:29 workstation dockerd[1073]: time="2023-11-13T12:08:29.564257509+08:00" level=info msg="Default
Nov 13 12:08:29 workstation dockerd[1073]: time="2023-11-13T12:08:29.936887944+08:00" level=info msg="Loading
Nov 13 12:08:30 workstation dockerd[1073]: time="2023-11-13T12:08:30.276340787+08:00" level=info msg="Docker
Nov 13 12:08:30 workstation dockerd[1073]: time="2023-11-13T12:08:30.305477400+08:00" level=info msg="Daemon
Nov 13 12:08:30 workstation systemd[1]: Started Docker Application Container Engine.
Nov 13 12:08:30 workstation dockerd[1073]: time="2023-11-13T12:08:30.451911686+08:00" level=info msg="API lis
lines 1-19/19 (END)
```

Add a Docker group to your current user.

```
daniela@workstation:~/CPE232_RABANG_HOA11$ sudo usermod -aG docker daniela
daniela@workstation:~/CPE232_RABANG_HOA11$ sudo systemctl restart docker
```

Create a Dockerfile to install web and DB server.

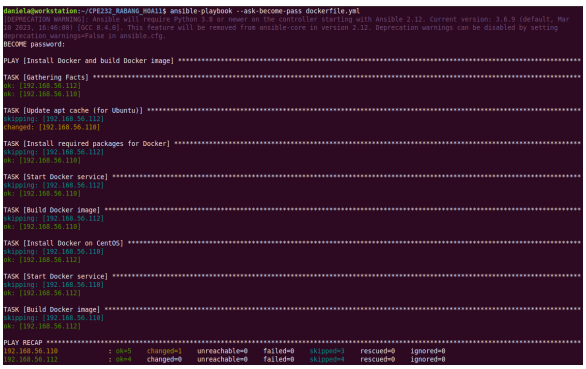
```
  GNU nano 2.9.3                                              dockerfile

FROM ubuntu
MAINTAINER daniela <qdmdrabang@tip.edu.ph>

#Skip prompts
ARG DEBIAN_FRONTEND=noninteractive

#Update packages
RUN apt-get update -y
RUN apt-get upgrade -y

#Install packages
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 packages
RUN /etc/init.d/apache2 start

#Set entrypoint
ENTRYPOINT apache2ctl -D FOREGROUND

FROM centos
# Install web server
RUN yum -y update && yum -y install httpd

# Install DB server

# Add relevant instructions for your database server

CMD ["/usr/sbin/httpd", "-D", "FOREGROUND"]
```

Install and build the Dockerfile using Ansible.

GNU nano 2.9.3 dockerfile.yml

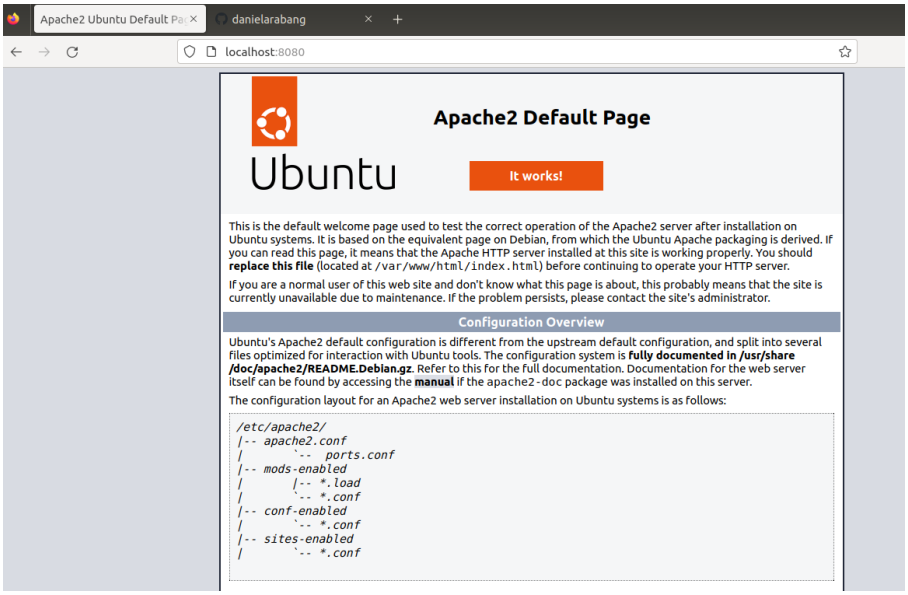
```
---
- hosts: all
  become: yes
  tasks:
    - name: Install Docker and enable the Docker socket
      apt:
        name: docker.io
        update_cache: yes
      when: ansible_distribution == "CentOS"

    - name: Add user to the Docker group
      user:
        name: "{{ daniela }}"
        groups: docker
      when: ansible_distribution == "CentOS"

    - name: Copy Dockerfile
      copy:
        src: dockerfile
        dest: /home/daniela/CPE232_RABANG_HOA11/dockerfile
      when: ansible_distribution == "CentOS"

    - name: Build Docker image
      command: docker build -t install1.0 /home/daniela/CPE232_RABANG_HOA11/file
      when: ansible_distribution == "CentOS"

    - name: Install Docker and enable the Docker socket
      apt:
        name: docker.io
        update_cache: yes
      when: ansible_distribution == "Ubuntu"

    - name: Add user to the Docker group
      user:
        name: "{{ daniela }}"
        groups: docker
      when: ansible_distribution == "Ubuntu"

    - name: Copy Dockerfile
      copy:
        src: dockerfile
        dest: /home/daniela/CPE232_RABANG_HOA11/dockerfile
      when: ansible_distribution == "Ubuntu"

    - name: Build Docker image
      command: docker build -t install1.0 /home/daniela/CPE232_RABANG_HOA11/file
      when: ansible_distribution == "Ubuntu"
```

IPO for Ubuntu:

Input	Process
GNU nano 2.9.3 dockerfile.yml --- - name: Install Docker and build Docker image hosts: all become: yes tasks: - name: Update apt cache (for Ubuntu) apt: update_cache: yes when: ansible_distribution == "Ubuntu" - name: Install required packages for Docker package: name: - docker - docker-compose when: ansible_distribution == "Ubuntu" - name: Start Docker service service: name: docker state: started when: ansible_distribution == "Ubuntu" - name: Build Docker image command: "docker build -t apache1.0 /home/daniela/CPE232_RABANG_HOA11/file" args: executable: /bin/bash when: ansible_distribution == "Ubuntu" - name: Install Docker on CentOS yum: name: docker state: present when: ansible_distribution == "CentOS" - name: Start Docker service service: name: docker state: started when: ansible_distribution == "CentOS" - name: Build Docker image command: "docker build -t apache1.0 /home/daniela/CPE232_RABANG_HOA11/file" when: ansible_distribution == "CentOS"	

Output

```
daniela@server1:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2023-11-14 16:49:51 PST; 37min ago
     Docs: https://docs.docker.com
 Main PID: 1050 (dockerd)
    Tasks: 8
   CGroup: /system.slice/docker.service
           └─1050 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Nov 14 16:49:48 server1 dockerd[1050]: time="2023-11-14T16:49:48.411517994+08:00" level=warning msg="Your kernel does not support C
Nov 14 16:49:48 server1 dockerd[1050]: time="2023-11-14T16:49:48.411521099+08:00" level=warning msg="Your kernel does not support c
Nov 14 16:49:48 server1 dockerd[1050]: time="2023-11-14T16:49:48.411523453+08:00" level=warning msg="Your kernel does not support c
Nov 14 16:49:48 server1 dockerd[1050]: time="2023-11-14T16:49:48.412440560+08:00" level=info msg="Loading containers: start."
Nov 14 16:49:50 server1 dockerd[1050]: time="2023-11-14T16:49:50.249593524+08:00" level=info msg="Default bridge (docker0) is assig
Nov 14 16:49:50 server1 dockerd[1050]: time="2023-11-14T16:49:50.702174266+08:00" level=info msg="Loading containers: done."
Nov 14 16:49:51 server1 dockerd[1050]: time="2023-11-14T16:49:51.050000972+08:00" level=info msg="Docker daemon" commit="20.10.21-0
Nov 14 16:49:51 server1 dockerd[1050]: time="2023-11-14T16:49:51.050874783+08:00" level=info msg="Daemon has completed initializati
Nov 14 16:49:51 server1 systemd[1]: Started Docker Application Container Engine.
Nov 14 16:49:51 server1 dockerd[1050]: time="2023-11-14T16:49:51.297834853+08:00" level=info msg="API listen on /var/run/docker.soc
lines 1-19/19 (END)
```

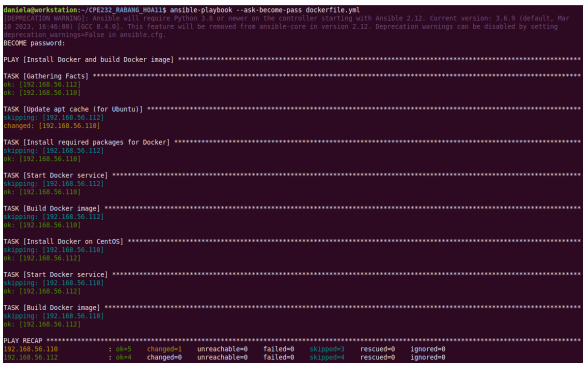


IPO for CentOS:

Input	Process

GNU nano 2.9.3 dockerfile.yml

```
---
- name: Install Docker and build Docker image
  hosts: all
  become: yes

  tasks:
    - name: Update apt cache (for Ubuntu)
      apt:
        update_cache: yes
      when: ansible_distribution == "Ubuntu"

    - name: Install required packages for Docker
      package:
        name:
          - docker
          - docker-compose
      when: ansible_distribution == "Ubuntu"

    - name: Start Docker service
      service:
        name: docker
        state: started
      when: ansible_distribution == "Ubuntu"

    - name: Build Docker image
      command: "docker build -t apache1.0 /home/daniela/CPE232_RABANG_HOA11/file"
      args:
        executable: /bin/bash
      when: ansible_distribution == "Ubuntu"

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

    - name: Start Docker service
      service:
        name: docker
        state: started
      when: ansible_distribution == "CentOS"

    - name: Build Docker image
      command: "docker build -t apache1.0 /home/daniela/CPE232_RABANG_HOA11/file"
      when: ansible_distribution == "CentOS"
```



Output

```
[daniela@localhost ~]$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; vendor preset: dis
abled)
   Active: active (running) since Tue 2023-11-14 04:49:43 EST; 6min ago
     Docs: http://docs.docker.com
 Main PID: 3677 (dockerd-current)
    Tasks: 17
   CGroup: /system.slice/docker.service
           ├─3677 /usr/bin/dockerd-current --add-runtime docker-runc=/usr/libexec/do...
           └─3682 /usr/bin/docker-containerd-current -l unix:///var/run/docker/libco...

Nov 14 04:49:41 localhost.localdomain dockerd-current[3677]: time="2023-11-14T04:49:...
Nov 14 04:49:42 localhost.localdomain dockerd-current[3677]: time="2023-11-14T04:49:...
Nov 14 04:49:42 localhost.localdomain dockerd-current[3677]: time="2023-11-14T04:49:...
Nov 14 04:49:42 localhost.localdomain dockerd-current[3677]: time="2023-11-14T04:49:...
Nov 14 04:49:43 localhost.localdomain dockerd-current[3677]: time="2023-11-14T04:49:...
Nov 14 04:49:43 localhost.localdomain dockerd-current[3677]: time="2023-11-14T04:49:...
Nov 14 04:49:43 localhost.localdomain dockerd-current[3677]: time="2023-11-14T04:49:...
Nov 14 04:49:43 localhost.localdomain dockerd-current[3677]: time="2023-11-14T04:49:...
Nov 14 04:49:43 localhost.localdomain systemd[1]: Started Docker Application Contai....
Nov 14 04:49:43 localhost.localdomain dockerd-current[3677]: time="2023-11-14T04:49:...
Hint: Some lines were ellipsized, use -l to show in full.
```

Add, commit and push it to your repository.

```
daniela@workstation:~/CPE232_RABANG_HOA11$ git add *
daniela@workstation:~/CPE232_RABANG_HOA11$ git commit -m "final"
[main 9c5a8fa] final
 6 files changed, 79 insertions(+)
 create mode 100644 ansible.cfg
 create mode 100644 dockerfile
 create mode 100644 dockerfile.retry
 create mode 100644 dockerfile.yml
 create mode 100644 file
 create mode 100644 inventory
daniela@workstation:~/CPE232_RABANG_HOA11$ git push origin main
Username for 'https://github.com': daniela
Password for 'https://daniela@github.com':
Counting objects: 8, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (8/8), 1.16 KiB | 1.16 MiB/s, done.
Total 8 (delta 0), reused 0 (delta 0)
To https://github.com/danielarabang/CPE232_RABANG_HOA11.git
   285fc82..9c5a8fa  main -> main
```

Reflections:

Answer the following:

1. **What are the benefits of implementing containerizations?**
   - The benefits of implementing containerizations are that it adds more in the compatibility and the development of the applications that are in that container. It also makes it more efficient for consistent operation.

Conclusions:

In this hands-on activity we are asked to do containerization to the use of the docker file. to install the docker into the two other machines where one ubuntu and one centos control node. This is using the ansible playbook.