

<b>Name: Frias, Abegail L.</b>	<b>Date Performed: Nov. 15, 2024</b>
<b>Course/Section: CPE212 - CPE31S21</b>	<b>Date Submitted: Nov. 15, 2024</b>
<b>Instructor: Engr. Robin Valenzuela</b>	<b>Semester and SY: 1st Sem/2024-2025</b>
<b>Activity 11: Containerization</b>	
<b>1. Objectives</b>	
Create a Dockerfile and form a workflow using Ansible as Infrastructure as Code (IaC) to enable Continuous Delivery process	
<b>2. Discussion</b>	
<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: <a href="https://docs.docker.com/get-started/overview/">https://docs.docker.com/get-started/overview/</a></p> <p>You may also check the difference between containers and virtual machines. Click the link given below.</p> <p>Source: <a href="https://docs.microsoft.com/en-us/virtualization/windowscontainers/about/containers-vs-vm">https://docs.microsoft.com/en-us/virtualization/windowscontainers/about/containers-vs-vm</a></p>	
<b>3. Tasks</b>	
<ol style="list-style-type: none"> <li>1. Create a new repository for this activity.</li> <li>2. Install Docker and enable the docker socket.</li> <li>3. Add to Docker group to your current user.</li> <li>4. Create a Dockerfile to install web and DB server.</li> <li>5. Install and build the Dockerfile using Ansible.</li> <li>6. Add, commit and push it to your repository.</li> </ol>	
<b>4. Output (screenshots and explanations)</b>	
<p><b>Cloning the repository to the server.</b></p> <pre>abegail@workstation:~\$ git clone git@github.com:wonbe/hoa_11.git Cloning into 'hoa_11'... warning: You appear to have cloned an empty repository.</pre>	

## Downloading docker.io to my workstation.

```
abegail@workstation:~$ sudo install -m 0755 -d /etc/apt/keyrings
abegail@workstation:~$ sudo curl -fsSL https://download.docker.com/linux/ubuntu
/gpg -o /etc/apt/keyrings/docker.asc
abegail@workstation:~$ sudo chmod a+r /etc/etc/apt/keyrings
chmod: cannot access '/etc/etc/apt/keyrings': No such file or directory
abegail@workstation:~$ sudo chmod a+r /etc/apt/keyrings
abegail@workstation:~$ echo \
> "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.
asc] https://download.docker.com/linux/ubuntu \
> $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
> sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
abegail@workstation:~$ sudo apt-get update
Hit:1 http://ph.archive.ubuntu.com/ubuntu bionic InRelease
Hit:2 http://ph.archive.ubuntu.com/ubuntu bionic-updates InRelease
Get:3 https://download.docker.com/linux/ubuntu bionic InRelease [64.4 kB]
Hit:4 http://ph.archive.ubuntu.com/ubuntu bionic-backports InRelease
Hit:5 http://security.ubuntu.com/ubuntu bionic-security InRelease
Get:6 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages [39
.0 kB]
Fetched 103 kB in 1s (107 kB/s)
Reading package lists... Done
abegail@workstation:~$ sudo systemctl enable docker.io
Failed to enable unit: Unit file docker.io.service does not exist.
abegail@workstation:~$ sudo apt-get install docker-ce docker-ce-cli containerd.
io docker-buildx-plugin docker-compose-plugin
Reading package lists... Done
```

```
abegail@workstation:~$ sudo apt show docker.io
Package: docker.io
Version: 20.10.21-0ubuntu1~18.04.3
Built-Using: glibc (= 2.27-3ubuntu1.5), golang-1.18 (= 1.18.1-1ubuntu1~18.04.4)
Priority: optional
Section: universe/admin
Origin: Ubuntu
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Original-Maintainer: Paul Tagliamonte <paultag@debian.org>
Bugs: https://bugs.launchpad.net/ubuntu/+filebug
Installed-Size: 140 MB
Depends: adduser, containerd (>= 1.2.6-0ubuntu1~), iptables, debconf (>= 0.5) |
debconf-2.0, libc6 (>= 2.8), libdevmapper1.02.1 (>= 2:1.02.97), libsystemd0 (>
= 209~)
Recommends: ca-certificates, git, pigz, ubuntu-fan, xz-utils, apparmor
Suggests: aufs-tools, btrfs-progs, cgroupfs-mount | cgroup-lite, debootstrap, d
ocker-doc, rinse, zfs-fuse | zfsutils
Breaks: docker (<< 1.5~)
Replaces: docker (<< 1.5~)
Homepage: https://www.docker.com/community-edition
Download-Size: 30.3 MB
APT-Sources: http://ph.archive.ubuntu.com/ubuntu bionic-updates/universe amd64
```

## Starting, enabling and checking the status of docker.

```
abegail@workstation:~$ sudo systemctl start docker
abegail@workstation:~$ sudo systemctl enable docker
Synchronizing state of docker.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable docker
abegail@workstation:~$ 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 08:05:08 +08; 12min ago
     Docs: https://docs.docker.com
   Main PID: 8646 (dockerd)
    Tasks: 9
   CGroup: /system.slice/docker.service
           └─8646 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/contai

Nov 15 08:05:05 workstation dockerd[8646]: time="2024-11-15T08:05:05.199015291+
Nov 15 08:05:06 workstation dockerd[8646]: time="2024-11-15T08:05:06.594750369+
Nov 15 08:05:06 workstation dockerd[8646]: time="2024-11-15T08:05:06.717711605+
Nov 15 08:05:07 workstation dockerd[8646]: time="2024-11-15T08:05:07.474089350+
Nov 15 08:05:07 workstation dockerd[8646]: time="2024-11-15T08:05:07.515600786+
Nov 15 08:05:07 workstation dockerd[8646]: time="2024-11-15T08:05:07.891504652+
Nov 15 08:05:07 workstation dockerd[8646]: time="2024-11-15T08:05:07.891801745+
Nov 15 08:05:07 workstation dockerd[8646]: time="2024-11-15T08:05:07.895014306+
Nov 15 08:05:08 workstation dockerd[8646]: time="2024-11-15T08:05:08.146752095+
Nov 15 08:05:08 workstation systemd[1]: Started Docker Application Container En
```

## Creating the docker group and adding my user to the docker group.

```
abegail@workstation:~$ sudo groupadd docker
[sudo] password for abegail:
groupadd: group 'docker' already exists
abegail@workstation:~$ sudo usermod -aG docker abegail
```

To verify that I can run the docker without sudo.

```
Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

abegail@workstation:~$
```

now create a docker file.

```
abegail@workstation:~/hoa_11$ ls
ansible.cfg  docker.yml  inventory  roles
Dockerfile   html        README.md  Ubuntu_Docker
abegail@workstation:~/hoa_11$
```

touch docker.yml

```
abegail@workstation:~/hoa_11$ cat docker.yml
---
- hosts: all
  become: true
  pre_tasks:

    - name: Install Updates (Ubuntu)
      tags: always
      apt:
        update_cache: yes
        changed_when: false
        when: ansible_distribution == "Ubuntu"

- hosts: Ubuntu
  become: true
  roles:
    - Ubuntu
abegail@workstation:~/hoa_11$
```

## Make another directory and create Dockerfile

```
abegail@workstation:~/hoa_11/Ubuntu_Docker$ cat Dockerfile
FROM ubuntu:latest
MAINTAINER abby <qafrias01@tip.edu.ph>

ARG DEBIAN_FRONTEND=noninteractive

RUN apt-get update -y
RUN apt-get upgrade -y

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 /etc/init.d/apache2 start

ENTRYPOINT apache2ctl -D FOREGROUND
abegail@workstation:~/hoa_11/Ubuntu_Docker$
```

make another directory named it roles inside make another and named it Ubuntu last is tasks and inside that tasks touch the main.yml

```
abegail@workstation:~/hoa_11/roles/Ubuntu/tasks$ cat main.yml
---
- name: Start the Docker Service
  tags: prep
  become: true
  service:
    name: docker
    state: started
    enabled: true

- name: Ensure group docker exists
  tags: prep
  become: true
  group:
    name: docker
    state: present

- name: Adding the current user to the docker group
  tags: prep
  user:
    name: "{{ ansible_user }}"
    groups: docker
    append: yes

- name: Create a docker directory
  file:
    path: /home/abegail/dockerfiles
    state: directory
    owner: "{{ ansible_user }}"
```

```
- name: Create a docker directory
  file:
    path: /home/abegail/dockerfiles
    state: directory
    owner: "{{ ansible_user }}"
    group: "{{ ansible_user }}"
    mode: '777'

- name: Copy Dockerfile to Ubuntu
  become: true
  copy:
    src: /home/abegail/hoa_11/Ubuntu_Docker/Dockerfile
    dest: /home/abegail/dockerfiles/
    owner: "{{ ansible_user }}"
    group: "{{ ansible_user }}"
    mode: '777'

- name: Build Docker Image
  become: true
  docker_image:
    path: /home/abegail/dockerfiles/
    name: apache-mariadb-image
    tag: latest
    state: present
```

here is the result:

```
abegail@workstation:~/hoa_11$ ansible-playbook -i inventory docker.yml --ask-be
come-pass
SUDO password:

PLAY [all] *****
*

TASK [Gathering Facts] *****
*
ok: [192.168.56.131]

TASK [Install Updates (Ubuntu)] *****
*
ok: [192.168.56.131]

PLAY [Ubuntu] *****
*

TASK [Gathering Facts] *****
*
ok: [192.168.56.131]

TASK [Ubuntu : Start the Docker Service] *****
*
ok: [192.168.56.131]

TASK [Ubuntu : Ensure group docker exists] *****
```

```

TASK [Gathering Facts] *****
*
ok: [192.168.56.131]

TASK [Ubuntu : Start the Docker Service] *****
*
ok: [192.168.56.131]

TASK [Ubuntu : Ensure group docker exists] *****
*
ok: [192.168.56.131]

TASK [Ubuntu : Adding the current user to the docker group] *****
*
ok: [192.168.56.131]

TASK [Ubuntu : Create a docker directory] *****
*
ok: [192.168.56.131]

TASK [Ubuntu : Copy Dockerfile to Ubuntu] *****
*
ok: [192.168.56.131]
```



```
abegail@workstation:~/hoa_11$ sudo docker images
[sudo] password for abegail:
REPOSITORY      TAG         IMAGE ID      CREATED        SIZE
mywebserver      latest      358b07d50876  20 minutes ago 227MB
hello-world      latest      d2c94e258dcb  18 months ago 13.3kB
abegail@workstation:~/hoa_11$
```

#### Github Link:

[https://github.com/wonbe/hoa\\_11](https://github.com/wonbe/hoa_11)

#### Reflections:

Answer the following:

1. What are the benefits of implementing containerizations?
  - By packaging apps and their dependencies to run reliably across environments and with fewer resources than virtual machines, containerization provides portability, efficiency, and scalability. It facilitates deployment with quicker starting times, offers security isolation, and works well with CI/CD pipelines. Containers are perfect for microservices, which allow for the autonomous management and scaling of application components, and they provide version control for simple rollbacks. Containerization is a potent strategy for contemporary software development and deployment because of these advantages.

#### Conclusions:

- In conclusion, software development and deployment are much improved by containerization, which is made possible by tools like Docker and comprehensive customizations via Dockerfiles. Docker improves efficiency and scalability by ensuring consistent performance across several environments by packaging apps with all of their dependencies. Developers may ensure reproducible and version-controlled environments by automating the setup and installation process with Dockerfiles. A key component of contemporary, effective, and secure software practices, this combination results in quicker, more dependable application deployment, improved resource management, and easier updates and rollbacks.