

| | |
|--|-----------------------------------|
| Name: Perez, Dean Lenard D | Date Performed: 15/11/24 |
| Course/Section: CPE 212-CPE31S21 | Date Submitted: 15/11/242 |
| Instructor: Engr. Robin Valenzuela | Semester and SY: 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 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) | |

For ansible connection

inventory file

```
dldperez@control:~/HOA_11.1$ cat inventory
[server1]
192.168.56.135
[control]
192.168.56.134
```

ansible.cfg file

```
dldperez@control:~/HOA_11.1$ cat ansible.cfg
[defaults]
inventory = inventory
remote_user = dldperez
Host_key_checking = False
retry_files_enabled = False
```

For installing docker

sudo apt install docker.io

```
dldperez@control:~/HOA_11.1$ sudo apt install docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
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 74 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 [15.4 kB]
```

```
Get:6 http://ph.archive.ubuntu.com/ubuntu bionic/main amd64 ubuntu-fan all 0.12
.10 [34.7 kB]
Fetched 65.7 MB in 29s (2,287 kB/s)
Preconfiguring packages ...
Selecting previously unselected package pigz.
(Reading database ... 182530 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.4-1_amd64.deb ...
Unpacking pigz (2.4-1) ...
Selecting previously unselected package bridge-utils.
Preparing to unpack .../1-bridge-utils_1.5-15ubuntu1_amd64.deb ...
Unpacking bridge-utils (1.5-15ubuntu1) ...
Selecting previously unselected package runc.
Preparing to unpack .../2-runc_1.1.4-0ubuntu1~18.04.2_amd64.deb ...
Unpacking runc (1.1.4-0ubuntu1~18.04.2) ...
Selecting previously unselected package containerd.
Preparing to unpack .../3-containerd_1.6.12-0ubuntu1~18.04.1_amd64.deb ...
Unpacking containerd (1.6.12-0ubuntu1~18.04.1) ...
Selecting previously unselected package docker.io.
Preparing to unpack .../4-docker.io_20.10.21-0ubuntu1~18.04.3_amd64.deb ...
Unpacking docker.io (20.10.21-0ubuntu1~18.04.3) ...
Selecting previously unselected package ubuntu-fan.
Preparing to unpack .../5-ubuntu-fan_0.12.10_all.deb ...
Unpacking ubuntu-fan (0.12.10) ...
Setting up runc (1.1.4-0ubuntu1~18.04.2) ...
Setting up containerd (1.6.12-0ubuntu1~18.04.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service
→ /lib/systemd/system/containerd.service.
Setting up bridge-utils (1.5-15ubuntu1) ...
Setting up ubuntu-fan (0.12.10) ...
```

```
Setting up bridge-utils (1.5-15ubuntu1) ...
Setting up ubuntu-fan (0.12.10) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ubuntu-fan.service
→ /lib/systemd/system/ubuntu-fan.service.
Setting up pigz (2.4-1) ...
Setting up docker.io (20.10.21-0ubuntu1~18.04.3) ...
Adding group `docker' (GID 130) ...
Done.
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /l
ib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /lib/s
ystemd/system/docker.socket.
Processing triggers for systemd (237-3ubuntu10.57) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for ureadahead (0.100.0-21) ...
dldperez@control:~/HOA_11.1$
```

To enable,start and see the status of docker

```
sudo systemctl start docker
sudo systemctl enable docker
sudo systemctl status docker
```

```

dldperez@control:~/HOA_11.1$ 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 10:13:20 +08; 3min 12s ago
     Docs: https://docs.docker.com
   Main PID: 7495 (dockerd)
      Tasks: 8
     CGroup: /system.slice/docker.service
             └─7495 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/contai

Nov 15 10:13:15 control dockerd[7495]: time="2024-11-15T10:13:15.351985850+08:0
Nov 15 10:13:15 control dockerd[7495]: time="2024-11-15T10:13:15.351995392+08:0
Nov 15 10:13:15 control dockerd[7495]: time="2024-11-15T10:13:15.351999084+08:0
Nov 15 10:13:15 control dockerd[7495]: time="2024-11-15T10:13:15.352188340+08:0
Nov 15 10:13:16 control dockerd[7495]: time="2024-11-15T10:13:16.285481311+08:0
Nov 15 10:13:16 control dockerd[7495]: time="2024-11-15T10:13:16.923556569+08:0
Nov 15 10:13:19 control dockerd[7495]: time="2024-11-15T10:13:19.965528267+08:0
Nov 15 10:13:19 control dockerd[7495]: time="2024-11-15T10:13:19.965753056+08:0
Nov 15 10:13:20 control dockerd[7495]: time="2024-11-15T10:13:20.435653866+08:0
Nov 15 10:13:20 control systemd[1]: Started Docker Application Container Engine
lines 1-19/19 (END)

```

It shows that docker is active and running.

Testing to know if sample image hello-world is running

```

dldperez@control:~/HOA_11.1$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
c1ec31eb5944: Pull complete
Digest: sha256:305243c734571da2d100c8c8b3c3167a098cab6049c9a5b066b6021a60fcb966
Status: Downloaded newer image for hello-world:latest

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/

```

Adding Docker group to your current user.

```
dldperez@control:~/HOA_11.1$ sudo usermod -aG docker dldperez
dldperez@control:~/HOA_11.1$ sudo systemctl restart docker
```

Create a Dockerfile to install web and DB server.

```
dldperez@control: ~/HOA_11.1
File Edit View Search Terminal Help
GNU nano 2.9.3 Dockerfile

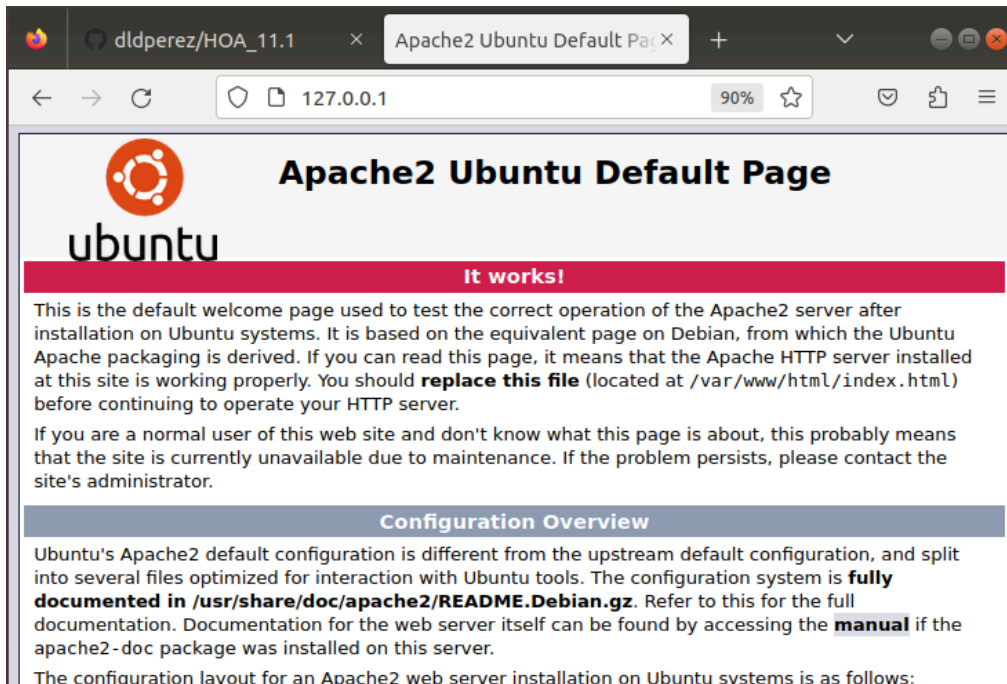
FROM ubuntu

ENV DEBIAN_FRONTEND=noninteractive
RUN apt-get update
RUN apt-get install apache2 -y
RUN apt-get install apache2-utils -y
RUN apt-get clean

EXPOSE 80

CMD ["apache2ctl", "-D", "FOREGROUND"]
```

```
dldperez@control:~/HOA_11.1$ sudo docker build -t mywebserver
Sending build context to Docker daemon 56.83kB
Step 1/8 : FROM ubuntu
---> 59ab366372d5
Step 2/8 : ENV DEBIAN_FRONTEND=noninteractive
---> Using cache
---> 6e5f94483352
Step 3/8 : RUN apt-get update
---> Using cache
---> 710d4dfac13f
Step 4/8 : RUN apt-get install apache2 -y
---> Using cache
---> 09f67b204a3d
Step 5/8 : RUN apt-get install apache2-utils -y
---> Using cache
---> be63e2b892f3
Step 6/8 : RUN apt-get clean
---> Running in 4119afcde605
Removing intermediate container 4119afcde605
---> adaac37a4428
Step 7/8 : EXPOSE 80
---> Running in ab128e98329b
Removing intermediate container ab128e98329b
---> f4f75f9a8c9b
Step 8/8 : CMD ["apache2ctl", "-D", "FOREGROUND"]
---> Running in d9054bf20ab2
Removing intermediate container d9054bf20ab2
---> a7a779319964
Successfully built a7a779319964
Successfully built a7a779319964
Successfully tagged mywebserver:latest
dldperez@control:~/HOA_11.1$
```



It was successful.

dockerfile.yml

```
File Edit View Search Terminal Help
GNU nano 2.9.3 dockerfile.yml

---
- hosts: server1
  become: true
  pre_tasks:

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

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

    - name: install docker (Ubuntu)
      apt:
        name: docker
        state: latest
      when: ansible_distribution == "Ubuntu"

    - name: install SDK (Ubuntu)
      shell:
```

[Read 61 lines]


```
dldperez@control: ~/HOA_11.1
File Edit View Search Terminal Help
GNU nano 2.9.3 dockerfile.yml

- name: adding group to Docker
  shell:
    usermod -aG docker dldperez

- name: Enable/Restart Docker (Ubuntu)
  service:
    name: docker
    state: started
    enabled: true

- name: Create Directory for Dockerfile
  file:
    path: ./root/demo-dockerfile
    state: directory
    owner: root
    group: root
    mode: '0755'

- name: Importing of Dockerfile
  copy:
    src: ./dockerfile
    dest: ./root/demo-dockerfile/dockerfile
```

sudo docker images

```
dldperez@control:~/HOA_11.1$ sudo docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
mywebserver          latest             a7a779319964       36 minutes ago     227MB
python               latest             c41ea8273365       3 weeks ago        1.02GB
ubuntu               latest             59ab366372d5       5 weeks ago        78.1MB
hello-world          latest             d2c94e258dcb       18 months ago      13.3kB
dldperez@control:~/HOA_11.1$
```

the image mywebserver was successfully created


```

dldperez@control:~/H0A_11.1$ ansible-playbook dockerfile.yml --ask-become-pass
SUDO password:

PLAY [server1] *****
*

TASK [Gathering Facts] *****
*
ok: [192.168.56.135]

TASK [install docker] *****
*
[WARNING]: Consider using 'become', 'become_method', and 'become_user' rather
than running sudo
changed: [192.168.56.135]

TASK [dpkg for Ubuntu] *****
*
changed: [192.168.56.135]

TASK [install docker (Ubuntu)] *****
*
changed: [192.168.56.135]

TASK [install SDK (Ubuntu)] *****
changed: [192.168.56.135]

TASK [dpkg for Ubuntu] *****
*
changed: [192.168.56.135]

TASK [install docker (Ubuntu)] *****
*
changed: [192.168.56.135]

TASK [install SDK (Ubuntu)] *****
*
changed: [192.168.56.135]

TASK [adding group to Docker] *****
*
changed: [192.168.56.135]

TASK [Enable/Restart Docker (Ubuntu)] *****
*
ok: [192.168.56.135]

TASK [Create Directory for Dockerfile] *****
*
changed: [192.168.56.135]

```

git add,commit and push

```
dldperez@control:~/HOA_11.1$ git add .
dldperez@control:~/HOA_11.1$ git commit "done"
error: pathspec 'done' did not match any file(s) known to git.
dldperez@control:~/HOA_11.1$ git commit -m "done"
[main 2e3b35c] done
 1 file changed, 5 insertions(+), 5 deletions(-)
dldperez@control:~/HOA_11.1$ git push
Counting objects: 3, done.
Delta compression using up to 3 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 303 bytes | 303.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:dldperez/HOA_11.1.git
   d36c01a..2e3b35c  main -> main
dldperez@control:~/HOA_11.1$
```

Reflections:

Answer the following:

1. What are the benefits of implementing containerizations?

-Benefits include executable software applications that are isolated in containers so that we can easily find errors or failures and it will not affect your entire system. Being isolated also increases its security. It is also efficient because it loads quickly and speed is essential.

Conclusions:

Containerization is a big help to software development because of the benefits mentioned above. The applications that are being created can be easily configured, tested and released using containerization. Using ansible to automate the process is a big help especially when releasing to a lot of users or remote hosts.