Name: Maxine Audrey D. Pulao	Date Performed: November 17, 2022
Course/Section: CPE31S2	Date Submitted: November 19, 2022
Instructor: Dr. Jonathan Taylar	Semester and SY: 2022-2023

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)

Install Docker and enable the docker socket.

```
aud@rey:~/ansible/CPE232-Activity_11$ sudo apt install docker.io
[sudo] password for aud:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libflashrom1 libftdi1-2 linux-headers-5.15.0-25
  linux-headers-5.15.0-25-generic linux-image-5.15.0-25-generic
  linux-modules-5.15.0-25-generic linux-modules-extra-5.15.0-25-generic
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  bridge-utils containerd git git-man liberror-perl pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools btrfs-progs cgroupfs-mount | cgroup-lite debootstrap
  docker-doc rinse zfs-fuse | zfsutils git-daemon-run | git-daemon-sysvinit
  git-doc git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  bridge-utils containerd docker.io git git-man liberror-perl pigz runc
  ubuntu-fan
0 upgraded, 9 newly installed, 0 to remove and 4 not upgraded.
Need to get 69.6 MB of archives.
After this operation, 303 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ph.archive.ubuntu.com/ubuntu jammy/universe amd64 pigz amd64 2.6-1
 [63.6 kB]
Get:2 http://ph.archive.ubuntu.com/ubuntu jammy/main amd64 bridge-utils amd64 1
.7-1ubuntu3 [34.4 kB]
Get:3 http://ph.archive.ubuntu.com/ubuntu jammy-updates/main amd64 runc amd64 1
.1.0-0ubuntu1.1 [4,242 kB]
```

- Installing the docker application

```
aud@rey:~/ansible/CPE232-Activity_11$ sudo systemctl enable docker
aud@rey:~/ansible/CPE232-Activity_11$ sudo systemctl start docker
aud@rey:~/ansible/CPE232-Activity_11$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor prese>
     Active: active (running) since Fri 2022-11-18 18:43:00 PST; 2min 0s ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 4724 (dockerd)
      Tasks: 7
     Memory: 37.2M
        CPU: 204ms
     CGroup: /system.slice/docker.service
             └─4724 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/con>

Nov 18 18:42:57 rey dockerd[4724]: time="2022-11-18T18:42:57.973364213+08:00" >
Nov 18 18:42:57 rey dockerd[4724]: time="2022-11-18T18:42:57.973597823+08:00" >
Nov 18 18:42:58 rey dockerd[4724]: time="2022-11-18T18:42:58.080308287+08:00" >
Nov 18 18:42:59 rey dockerd[4724]: time="2022-11-18T18:42:59.734899284+08:00" >
Nov 18 18:42:59 rey dockerd[4724]: time="2022-11-18T18:42:59.919773858+08:00" >
Nov 18 18:43:00 rey dockerd[4724]: time="2022-11-18T18:43:00.119375164+08:00" >
Nov 18 18:43:00 rey dockerd[4724]: time="2022-11-18T18:43:00.253255582+08:00" >
Nov 18 18:43:00 rey dockerd[4724]: time="2022-11-18T18:43:00.253341503+08:00" >
Nov 18 18:43:00 rey systemd[1]: Started Docker Application Container Engine.
Nov 18 18:43:00 rey dockerd[4724]: time="2022-11-18T18:43:00.344775617+08:00" >
lines 1-22/22 (END)
```

-the following commands show the that the docker is properly installed and it is working.

```
aud@rey:~/ansible/CPE232-Activity_11$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:faa03e786c97f07ef34423fccceeec2398ec8a5759259f94d99078f264e9d7af
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/
```

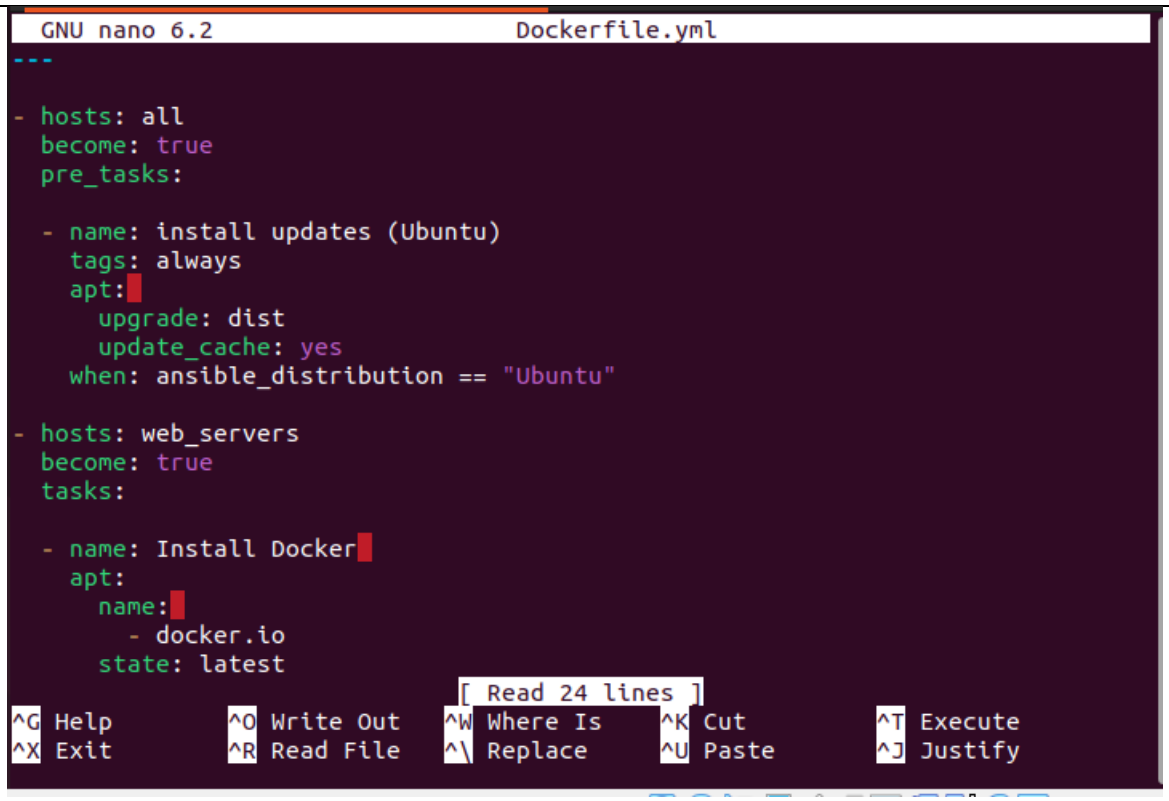
- This command proves that the docker is working.

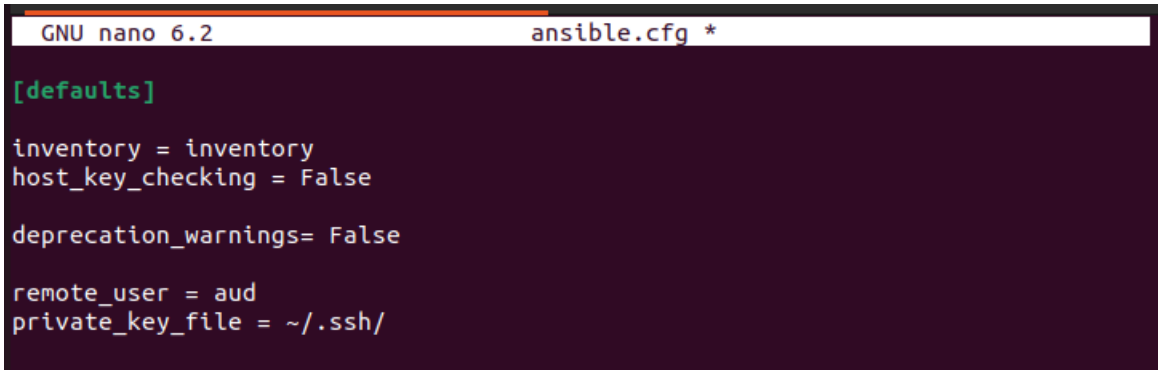
Add to Docker group to your current user.

```
root@rey:~# grep docker /etc/group
docker:x:137:aud
root@rey:~# usermod -aG docker aud
root@rey:~# id aud
uid=1000(aud) gid=1000(aud) groups=1000(aud),4(adm),24(cdrom),27(sudo),30(dip),
46(plugdev),122(lpadmin),134(lxd),135(sambashare),137(docker)
root@rey:~#
```

- these commands adds a user to the docker application

Create a Dockerfile to install web and DB server.

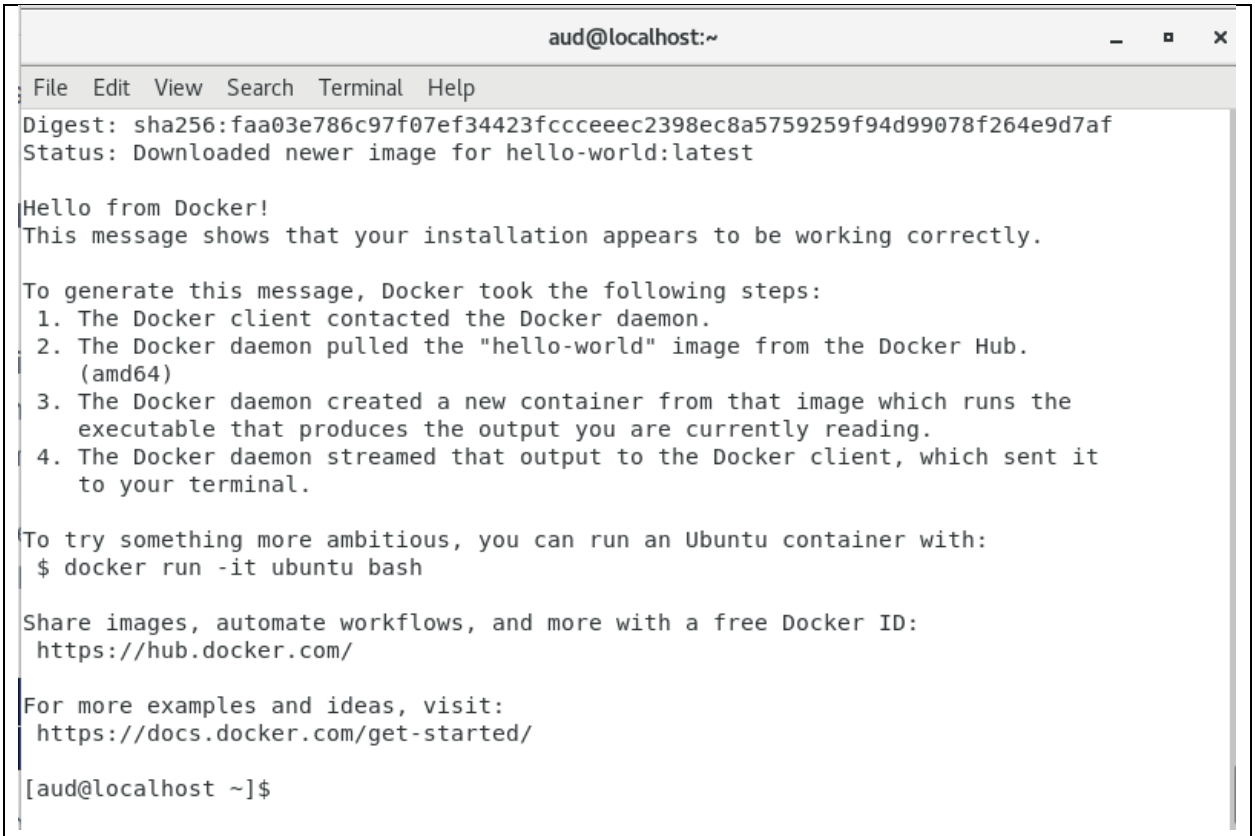




Install and build the Dockerfile using Ansible.

```
aud@rey:~/ansible/CPE232-Activity_11$ ansible-playbook --ask-become-pass Docker
file.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
*
ok: [10.0.2.15]
ok: [192.168.56.120]

TASK [install updates (CentOS)] ************************************************
*
skipping: [192.168.56.120]
skipping: [10.0.2.15]

TASK [install updates (Ubuntu)] ************************************************
*
ok: [10.0.2.15]

PLAY [web_servers] *************************************************************
*
[WARNING]: Could not match supplied host pattern, ignoring: db_servers

PLAY [db_servers] **************************************************************
*
skipping: no hosts matched

PLAY RECAP *********************************************************************
*
10.0.2.15                  : ok=2    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
```

- Successfully installing the docker application



- The docker application successfully installed in the db server.

Reflections:

Answer the following:

1. What are the benefits of implementing containerizations?
   Implementing containerizations has the advantage of often using fewer resources than virtual machines. Nearly anywhere can run Docker containers, and they are simple to copy and deploy. Additionally, it may be run on platforms like AWS, Linode, Digital Ocean, Google Cloud, and others and is frequently less expensive to maintain than virtual machines.

Conclusions:

In conclusion, by introducing a new platform called Dockerfile, I was able to accomplish the goals of this practical project. Regarding the technique, I looked for information on how to install Dockerfile on CentOS (for database servers) and Ubuntu (for web servers) systems. I turned its installation instructions into an Ansible playbook that can install and run services on both Ubuntu and CentOS systems once I figured out how to install it on both servers. Its installation process was similar to how I had done it before as I installed Docker and

launched the service. I gained a general understanding of Docker, a new platform for creating, distributing, and operating applications.