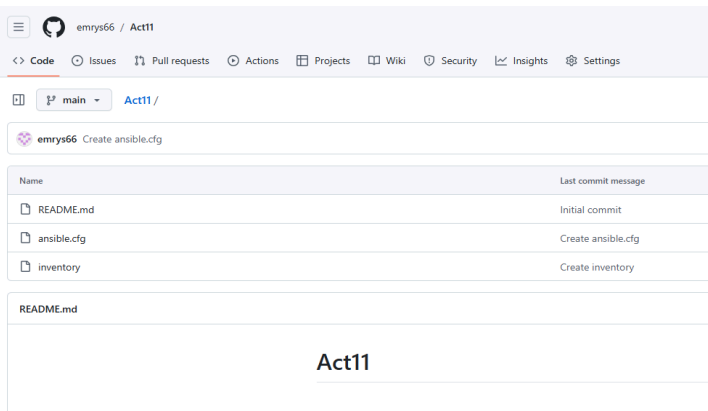


Name: Escosia, Jerico James	Date Performed: 11/15/24
Course/Section: CPE212- CPE31S21	Date Submitted: 11/15/24
Instructor: Engr. Robin Valenzuela	Semester and SY: 1st Sem
Activity 11: Containerization	
1. Objectives	
Create a Dockerfile and form a workflow using Ansible as Infrastructure as Code (IaC) 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 servers. 5. Install and build the Dockerfile using Ansible. 6. Add, commit and push it to your repository. 	

4. Output (screenshots and explanations)

1.



2.

```
workstation@workstation:~$ sudo apt install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools btrfs-progs cgroupfs-mount | cgroup-lite debootstrap
  docker-buildx docker-compose-v2 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 28 not upgraded.
Need to get 76.4 MB of archives.
After this operation, 288 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ph.archive.ubuntu.com/ubuntu noble/universe amd64 pigz amd64 2.8-1 [65.6 kB]
Get:2 http://ph.archive.ubuntu.com/ubuntu noble/main amd64 bridge-utils amd64 1.7.1-1ubuntu2 [33.9 kB]
Get:3 http://ph.archive.ubuntu.com/ubuntu noble-updates/main amd64 runc amd64 1.1.12-0ubuntu3.1 [8,599 kB]
Get:4 http://ph.archive.ubuntu.com/ubuntu noble-updates/main amd64 containerd amd64 1.7.12-0ubuntu4.1 [38.6 MB]
Get:5 http://ph.archive.ubuntu.com/ubuntu noble-updates/universe amd64 docker.io amd64 24.0.7-0ubuntu4.1 [29.1 MB]
Get:6 http://ph.archive.ubuntu.com/ubuntu noble/universe amd64 ubuntu-fan all 0.12.16 [35.2 kB]
Fetched 76.4 MB in 35s (2,196 kB/s)
Preconfiguring packages ...
Selecting previously unselected package pigz.
(Reading database ... 213213 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.8-1_amd64.deb ...
Unpacking pigz (2.8-1) ...
Selecting previously unselected package bridge-utils.
```

```
workstation@workstation:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
   Active: active (running) since Fri 2024-11-15 09:06:17 PST; 57s ago
     TriggeredBy: ● docker.socket
   Docs: https://docs.docker.com
  Main PID: 5274 (dockerd)
    Tasks: 10
   Memory: 25.1M (peak: 25.8M)
      CPU: 496ms
   CGroup: /system.slice/docker.service
           └─5274 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
```

3.

```
workstation@workstation: ~  
workstation@workstation:~$ sudo usermod -aG docker $USER  
workstation@workstation:~$ newgrp docker  
workstation@workstation:~$ docker run hello-world  
Unable to find image 'hello-world:latest' locally  
latest: Pulling from library/hello-world  
c1ec31eb5944: Pull complete  
Digest: sha256:305243c734571da2d100c8c8b3c3167a098cab6049c9a5b066b6021a60fcb9  
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/
```

4.

```
GNU nano 7.2 dockerfile1 *  
FROM ubuntu:latest  
MAINTAINER workstation <qjjiescosia@tip.edu.ph>  
  
# skip prompts  
ARG DEBIAN_FRONTEND=noninteractive  
  
# update packages  
RUN apt update  
RUN apt upgrade -y  
  
# install packages  
RUN apt-get install -y apache2 mariadb-server  
  
# set entrypoint  
ENTRYPOINT apache2ctl -D FOREGROUND
```

5.

```
workstation@workstation:~/Act11$ ansible-playbook --ask-become-pass docker.yml
BECOME password:

PLAY [localhost] *****

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

TASK [Install required packages for Debian] *****
ok: [localhost]

TASK [Install required packages for RedHat] *****
skipping: [localhost]

TASK [Install Docker from default repositories (Debian)] *****
ok: [localhost]

TASK [Install Docker from default repositories (RedHat)] *****
skipping: [localhost]

TASK [Start and enable Docker] *****
ok: [localhost]

TASK [Install Python pip for Debian] *****
ok: [localhost]
```

```
workstation@workstation:~/Act11$ sudo docker build -t mywebserver .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon 73.22kB
Step 1/5 : FROM nginx:alpine
alpine: Pulling from library/nginx
da9db072f522: Pull complete
795a89dcc4f0: Pull complete
9ad567d3b8a2: Pull complete
85e8836903ab: Pull complete
276e8818f0df: Pull complete
20212ccf8a7a: Pull complete
3917e8bc7be0: Pull complete
bcf46b1b1b14: Pull complete
Digest: sha256:74175cf34632e88c6cfe206897cbfe2d2fecf9bf033c40e7f9775a3689e8adc7
Status: Downloaded newer image for nginx:alpine
--> a5967740120f
Step 2/5 : RUN apk add --no-cache mysql mysql-client
--> Running in bc14048fba8a
fetch https://dl-cdn.alpinelinux.org/alpine/v3.20/main/x86_64/APKINDEX.tar.gz
fetch https://dl-cdn.alpinelinux.org/alpine/v3.20/community/x86_64/APKINDEX.tar.gz
(1/12) Installing mariadb-common (10.11.10-r0)
(2/12) Installing libaio (0.3.113-r2)
(3/12) Installing libgcc (13.2.1_git20240309-r0)
(4/12) Installing skalibs (2.14.1.1-r0)
(5/12) Installing utmps-lib (0.1.2.2-r1)
(6/12) Installing linux-pam (1.6.0-r0)
(7/12) Installing libstdc++ (13.2.1_git20240309-r0)
(8/12) Installing mariadb (10.11.10-r0)
Executing mariadb-10.11.10-r0.pre-install
```

```
workstation@workstation:~/Act11$ sudo docker images -a
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
<none>	<none>	008c6082bfcd	About a minute ago	306MB
nginx	alpine	a5967740120f	6 weeks ago	52.5MB
hello-world	latest	d2c94e258dcb	18 months ago	13.3kB

```
workstation@workstation:~/Act11$ git push origin main
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 3 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (9/9), 1.42 KiB | 1.42 MiB/s, done.
Total 9 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:emrys66/Act11.git
   f77a9a6..d9a070d  main -> main
```

```
workstation@server1:~$ docker --version
Docker version 20.10.21, build 20.10.21-0ubuntu1~18.04.3
```

6.

```
workstation@server1:~$ sudo systemctl status docker
[sudo] password for workstation:
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset:
   Active: active (running) since Fri 2024-11-15 09:41:18 +08; 2h 53min ago
     Docs: https://docs.docker.com
    Main PID: 6568 (dockerd)
      Tasks: 10
     CGroup: /system.slice/docker.service
             └─6568 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/contai

Nov 15 09:41:17 server1 dockerd[6568]: time="2024-11-15T09:41:17.652280383+08:0
Nov 15 09:41:17 server1 dockerd[6568]: time="2024-11-15T09:41:17.652287091+08:0
Nov 15 09:41:17 server1 dockerd[6568]: time="2024-11-15T09:41:17.652290453+08:0
Nov 15 09:41:17 server1 dockerd[6568]: time="2024-11-15T09:41:17.652467180+08:0
Nov 15 09:41:17 server1 dockerd[6568]: time="2024-11-15T09:41:17.954995707+08:0
Nov 15 09:41:18 server1 dockerd[6568]: time="2024-11-15T09:41:18.062035115+08:0
Nov 15 09:41:18 server1 dockerd[6568]: time="2024-11-15T09:41:18.271081272+08:0
Nov 15 09:41:18 server1 dockerd[6568]: time="2024-11-15T09:41:18.271302652+08:0
Nov 15 09:41:18 server1 systemd[1]: Started Docker Application Container Engine
Nov 15 09:41:18 server1 dockerd[6568]: time="2024-11-15T09:41:18.327128502+08:0
lines 1-19/19 (END)
```

Reflections:

Answer the following:

1. What are the benefits of implementing containerizations?

Containerization provides key benefits, like portability because the application runs the same way across the different environments, and scalability because the replication and managing of services is very easy. It also offers isolation to the application, which improves security and resource utilization, and simplifies the development and testing by establishing consistent environments that reduce problems at the time of deployment. In a nutshell, containerization simplifies the management and deployment of applications and, as such, has become the preferred choice for modern software development.

https://github.com/emrys66/CPE_MIDEXAM_ESCOSIA.git

Conclusions:

In this activity we created a Docker file to create a containerized environment and then ran an Ansible playbook from the control node over to the managed node. This step automated the installation of a web and database server on the managed node, to verify if the docker was installed in the manage node we can type in the manage node `docker --version`, if there is an output that means the docker was successfully installed using ansible.