

|  |   |
|--|---|
| <b>Name:</b> Onod, Earel John E.   | <b>Date Performed:</b> 10-24-2025         |
| <b>Course/Section:</b> CPE212 - CPE31S4  | <b>Date Submitted:</b> 10-24-2025         |
| <b>Instructor:</b> Sir Robin Valenzuela  | <b>Semester and SY:</b> 1st Sem   25 - 26 |
| <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</b> (screenshots and explanations)  |   |
| 1.   |   |

Screenshot of a web browser showing the GitHub repository page for `EarelJohnOnod/docker-ansible-setup`. The repository is public and contains a Dockerfile and an Ansible setup. The commit history shows three commits: Dockerfile, README.md, and build1.yml.

```
try@controller:~$ git clone https://github.com/EarelJohnOnod/docker-ansible-setup.git
Cloning into 'docker-ansible-setup'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
try@controller:~$ cd docker-ansible-setup
try@controller:~/docker-ansible-setup$ sudo apt update
```

2.

```
try@controller:~/docker-ansible-setup$ sudo systemctl enable docker
try@controller:~/docker-ansible-setup$ sudo systemctl start docker
try@controller:~/docker-ansible-setup$ 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 2025-10-24 14:27:19 PST; 21s ago
     TriggeredBy: ● docker.socket
    Docs: https://docs.docker.com
   Main PID: 31704 (dockerd)
      Tasks: 10
     Memory: 35.8M (peak: 35.9M)
        CPU: 1.130s
    CGroup: /system.slice/docker.service
            └─31704 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
```

3.

```
try@controller:~/docker-ansible-setup$ sudo usermod -aG docker try
try@controller:~/docker-ansible-setup$ newgrp docker
```

4.

```
try@controller:~/docker-ansible-setup$ nano Dockerfile
```

```
try@controller: ~/docker-ansible-setup
GNU nano 7.2 Dockerfile
FROM ubuntu:latest

ENV DEBIAN_FRONTEND=noninteractive

RUN apt update && apt install -y apache2 mysql-server && apt clean

EXPOSE 80

CMD service mysql start && apachectl -D FOREGROUND
```

5.

```
try@controller:~/docker-ansible-setup$ nano build1.yml
```

```
try@controller: ~/docker-ansible-setup
GNU nano 7.2 build1.yml
---
- name: build run Docker container
  hosts: localhost
  become: yes
  tasks:

    - name: build Docker image
      community.docker.docker_image:
        name: web_db_server
        source: build
        build:
          path: .

    - name: run Docker container
      community.docker.docker_container:
        name: web_db_container
        image: nginx:latest
        state: started
        ports:
          - "8080:80"
```

```
try@controller:~/docker-ansible-setup$ ansible-playbook build1.yml
[WARNING]: No inventory was parsed, only implicit localhost is available
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match 'all'

PLAY [build run Docker container] *****

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

TASK [build Docker image] *****
ok: [localhost]

TASK [run Docker container] *****
ok: [localhost]

PLAY RECAP *****
localhost : ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

*only implemented to localhost due to recent SSH problems with two of the nodes*

6.

```
try@controller:~/docker-ansible-setup$ git add .
try@controller:~/docker-ansible-setup$ git commit -m "Dockerfile and ansible setup"
[main 64f310c] Dockerfile and ansible setup
3 files changed, 54 insertions(+)
create mode 100644 Dockerfile
create mode 100644 build.yml
create mode 100644 build1.yml
try@controller:~/docker-ansible-setup$ git branch -M main
try@controller:~/docker-ansible-setup$ git push origin main
```

```
remote: Permission to EarelJohnOnod/docker-ansible-setup.git denied to EarelJohnOnod.
fatal: unable to access 'https://github.com/EarelJohnOnod/docker-ansible-setup.git/': The requested URL returned error:
403
```

*due to permission access error, I've manually committed the files done in this activity*

### **Github Link:**

<https://github.com/EarelJohnOnod/docker-ansible-setup.git>

### **Reflections:**

Answer the following:

1. What are the benefits of implementing containerizations?
  - It makes the distribution of applications easier.
  - It also makes the update of these distributions easier.
  - Applications work the same way as they do even on different PCs.
  - Due its isolation, one container if crashed or compromised won't affect other containers.

### **Conclusions:**

By combining both Dockerbuild and Ansible, you can easily distribute applications and tasks at the same time on all or selected PCs. With this implementation, it is possible to make an automated system that will just require some few human inputs and a bit of maintenance every now and then.