

Name: Erebete, Jan Kenneth F.	Date Performed: 10/24/25
Course/Section: CPE32S4	Date Submitted: 10/24/25
Instructor: Engr. Robin Valenzuela	Semester and SY: 2025 - 2026

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)

1. In this part i just create a repository inside the CPE232_Erebete directory and just create Docker as a folder inside of it

```
erebete@Workstation:~/CPE232_Erebete/Docker$ ls
deploy_nginx.yml  Docker  docker.yml  index.html  install_docker.yaml
```

2. In this part i already install the docker using the code that given during the online class and just enable it to be activated and used the systemctl status docker to check if the docker was installed already on my ubuntu

```
erebete@Workstation:~/CPE232_Erebete$ systemctl status docker
● docker.service - Docker Application Container Engine
  Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: en>
  Active: active (running) since Fri 2025-10-24 05:58:42 UTC; 14min ago
TriggeredBy: ● docker.socket
  Docs: https://docs.docker.com
 Main PID: 1324 (dockerd)
    Tasks: 19
   Memory: 105.1M (peak: 106.2M)
      CPU: 1.442s
     CGroup: /system.slice/docker.service
             └─1324 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/cont>
               ├─1746 /usr/bin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-por>

Warning: some journal files were not opened due to insufficient permissions.
lines 1-14/14 (END)
```

3. In this part i just create a group for the docker user account the necessary permissions to interact with the Docker daemon without needing to prefix every command with sudo. It achieves this by adding your user to the special docker Linux group, simplifying command-line usage.

```
erebete@Workstation:~/CPE232_Erebete/Docker$ sudo usermod -aG docker $USER
erebete@Workstation:~/CPE232_Erebete/Docker$ newgrp docker
erebete@Workstation:~/CPE232_Erebete/Docker$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
17eec7bbc9d7: Pull complete

Digest: sha256:6dc565aa630927052111f823c303948cf83670a3903ffa3849f1488ab517f891
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:
```

4. Here I just created inside the docker directory a Dockerfile is created as a script of instructions to automatically build a new custom container image. This file specifies a base operating system and lists the commands to install both a web server and a database.

```
GNU nano 7.2                               Dockerfile *
FROM ubuntu:22.04

# Set environment variables for non-interactive install
ENV DEBIAN_FRONTEND=noninteractive

# Install Web Server (Apache) and DB Client (MariaDB)
RUN apt update && \
    apt install -y apache2 mariadb-client procps vim && \
    apt clean && \
    rm -rf /var/lib/apt/lists/*

# Copy a simple index.html file (create this file too!)
COPY index.html /var/www/html/

# Expose the standard HTTP port
EXPOSE 80

# Configure the entrypoint to keep the web server running
CMD ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]
```

```
GNU nano 7.2                               index.html
<!DOCTYPE html>
<html>
<head>
<title>Docker-Ansible Build Successful</title>
</head>
<body>
    <h1>Container Build Success!</h1>
    <p>Apache Web Server and MariaDB Client are installed in this container.</p>
</body>
</html>
```

5. In here I just created a playbook build_docker.yml and executed the Docker build process that processes across one or more target machines. It uses a playbook with the docker_image module to read the Dockerfile and create the final, runnable container image.

```
erebete@Workstation:~/CPE232_Erebete/Docker$ ansible-playbook build_docker.yml -K
BECOME password:
[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 and Manage Docker Image] ****
TASK [Ensure Docker image is built] ****
ok: [localhost]

TASK [Ensure existing container is stopped] ****
ok: [localhost]

TASK [Ensure existing container is removed] ****
changed: [localhost]
```

```
GNU nano 7.2                               build_docker.yml *
-- 
name: Build and Manage Docker Image
hosts: localhost
connection: local
gather_facts: false
tasks:
  - name: Ensure Docker image is built
    community.docker.docker_image:
      build:
        path: .
        dockerfile: Dockerfile
        name: web-db-server-image
        tag: latest
        state: present
        source: build

  - name: Ensure existing container is stopped
    community.docker.docker_container:
      name: web-db-server-container
      state: stopped
```

← → ⌂ ⚡ http://localhost:8080

Success! Nginx is serving content via Docker.

This page was deployed using Ansible and Docker.

6. In this part after all the configuration and yml files i did on the repository i just used the git status to check all the files i created and used git add to add all files i created next is to commit and lastly was to push on my CPE232_Erebete

```
erebete@Workstation:~/CPE232_Erebete$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:  Docker/Docker
    new file:  Docker/Dockerfile
    new file:  Docker/build_docker.yml
    renamed:  deploy_nginx.yml -> Docker/deploy_nginx.yml
    renamed:  docker.yml -> Docker/docker.yml
    new file:  Docker/index.html
    renamed:  install_docker.yaml -> Docker/install_docker.yaml
    deleted:  Dockerfile
    new file:  files/default_site.html
    deleted:  index.html
    new file:  yer.yml
```

```
erebete@Workstation:~/CPE232_Erebete$ git add Docker/Docker Docker/Dockerfile Docker/build_docker.yml Docker/deploy_nginx.yml Docker/docker.yml Docker/index.html Docker/install_docker.yaml files/default_site.html
erebete@Workstation:~/CPE232_Erebete$ git commit -m "Hoa11.1"
[main 806e004] Hoa11.1
11 files changed, 130 insertions(+), 29 deletions(-)
create mode 160000 Docker/Docker
create mode 100644 Docker/Dockerfile
create mode 100644 Docker/build_docker.yml
rename deploy_nginx.yml => Docker/deploy_nginx.yml (69%)
rename docker.yml => Docker/docker.yml (100%)
create mode 100644 Docker/index.html
rename install_docker.yaml => Docker/install_docker.yaml (100%)
delete mode 100644 Dockerfile
create mode 100644 files/default_site.html
delete mode 100644 index.html
create mode 100644 yer.yml
```

```
erebete@Workstation:~/CPE232_Erebete$ git push origin main
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 5 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (10/10), 2.73 KiB | 932.00 KiB/s, done.
Total 10 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:qjkferebete/CPE232_Erebete.git
  cf179f5..806e004  main -> main
```

main		
CPE232_Erebete / Docker /		
qjkferebete Hoa11.1		806e004 · 2 minutes ago
Name	Last commit message	Last commit date
..		
Docker	Hoa11.1	2 minutes ago
Dockerfile	Hoa11.1	2 minutes ago
build_docker.yml	Hoa11.1	2 minutes ago
deploy_nginx.yml	Hoa11.1	2 minutes ago
docker.yml	Hoa11.1	2 minutes ago
index.html	Hoa11.1	2 minutes ago
install_docker.yaml	Hoa11.1	2 minutes ago

Reflections:

Answer the following:

1. What are the benefits of implementing containerizations?

- The benefits of implementing containerization was it is easy to configure and do and also containerization lets apps run smoothly anywhere by packaging everything they need into one portable and it is lightweight and efficient, saving resources and cutting costs.

Conclusions:

My conclusion to this was I learned the use of docker and how docker works on ubuntu and also made me learn on how to configure files on docker and also the running on ansible makes it a lot easier to run on using docker.