Name: Potestades, North Nygel G.	**Date Performed:** 10/24/25
Course/Section: CPE31S2	**Date Submitted:** 10/24/25
Instructor: Engr. Robin Valenzuela	**Semester and SY:** 1st Semester 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)

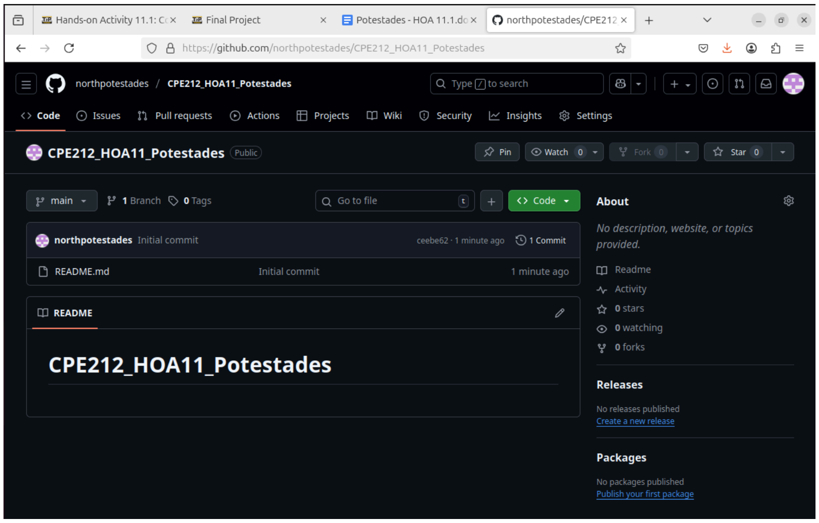
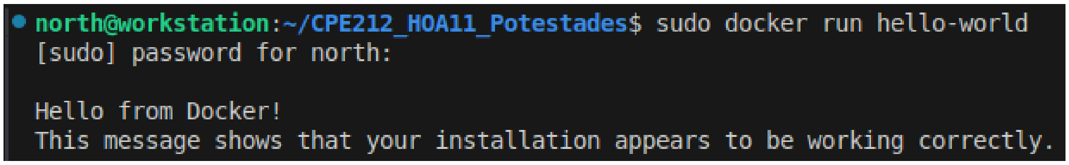


Figure 4.1. New Repository

Here we can see the new repository which I created just for this activity.

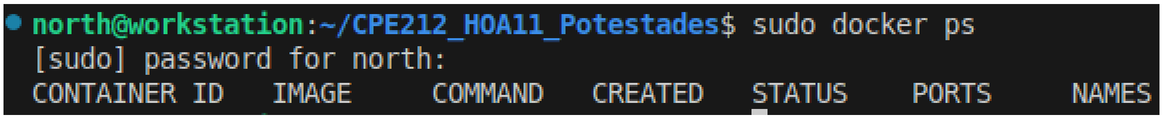


```
north@workstation:~/CPE212_HOA11_Potestades$ sudo docker run hello-world
[sudo] password for north:

Hello from Docker!
This message shows that your installation appears to be working correctly.
```

Figure 4.2. Successful Docker Installation

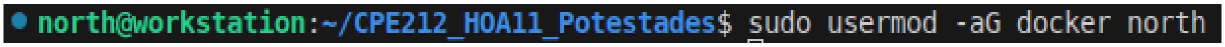
Here we can see that Docker is successfully installed on the control node by running the example container called hello-world.



```
north@workstation:~/CPE212_HOA11_Potestades$ sudo docker ps
[sudo] password for north:
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```

Figure 4.3. Docker Socket Enabled

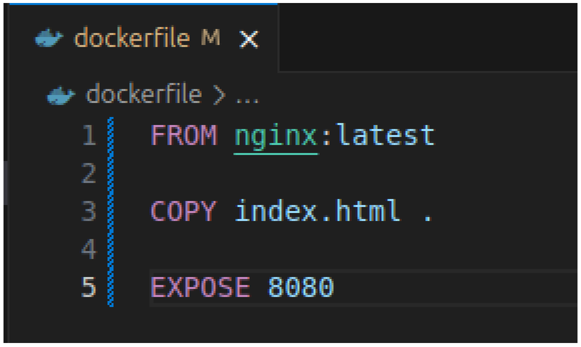
Here we can see that Docker's socket is successfully enabled on the control node using sudo docker ps.



```
north@workstation:~/CPE212_HOA11_Potestades$ sudo usermod -aG docker north
```

Figure 4.4. User Added to Docker Group

Here we can see that my main user, north, is now added to the Docker group using usermod.



```
FROM nginx:latest

COPY index.html .

EXPOSE 8080
```

Figure 4.5. Created Dockerfile

Here we can see that I have created a Dockerfile which will install web and DB servers for both Ubuntu and CentOS hosts.

N/A

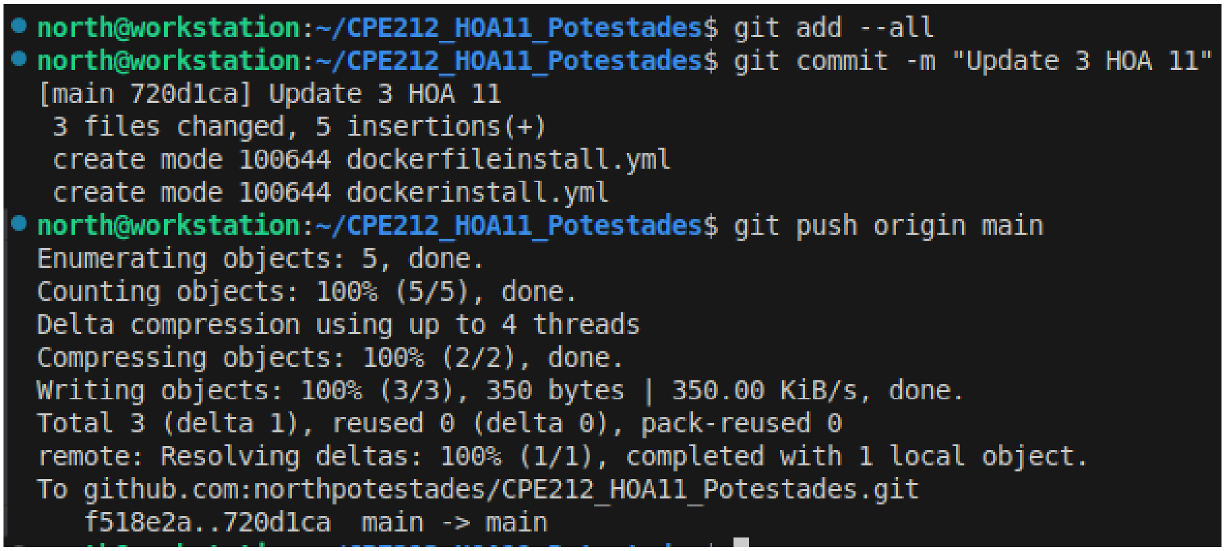
Figure 4.6. Ansible Results

Here we can see the results of the Ansible playbook which installed and built the dockerfile on the remote hosts.

N/A

Figure 4.7. Running Docker Container on Hosts

Here we can see the Docker containers running on the remote hosts.



```
● north@workstation:~/CPE212_HOA11_Potestades$ git add --all
● north@workstation:~/CPE212_HOA11_Potestades$ git commit -m "Update 3 HOA 11"
  [main 720d1ca] Update 3 HOA 11
   3 files changed, 5 insertions(+)
   create mode 100644 dockerfileinstall.yml
   create mode 100644 dockerinstall.yml
● north@workstation:~/CPE212_HOA11_Potestades$ git push origin main
  Enumerating objects: 5, done.
  Counting objects: 100% (5/5), done.
  Delta compression using up to 4 threads
  Compressing objects: 100% (2/2), done.
  Writing objects: 100% (3/3), 350 bytes | 350.00 KiB/s, done.
  Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
  remote: Resolving deltas: 100% (1/1), completed with 1 local object.
  To github.com:northpotestades/CPE212_HOA11_Potestades.git
     f518e2a..720d1ca  main -> main
```

Figure 4.8. Final Commit to Repository

Here we can see the final commit to the repository created for this activity.

Reflections:

Answer the following:

1. What are the benefits of implementing containerizations?

The benefits of implementing containerization include consistency, being sure that your applications will run smoothly once the container is deployed, increased speed in deploying applications to different systems, and great scalability, as containers are typically lightweight, being able to be scaled up or down depending on the requirements.

Conclusions:

In conclusion, this hands-on activity focused on the utilization of Docker and Ansible to deploy containers onto remote hosts. This was quite the challenge, as it intertwines two different technologies to perform one task, which requires patience as errors can occur. Overall, this activity was a good experience to ensure that we can create the Dockerfile required for our final project, and using Ansible to apply these onto remote systems.