Name: Shaniah Rose Hope M. Sumaoang	Date Performed: November 21, 2023
Course/Section: CPE 232-CPE31S5	Date Submitted: December 7, 2023
Instructor: Engr. Roman Richard	Semester and SY: 1st Sem SY 23-24

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.

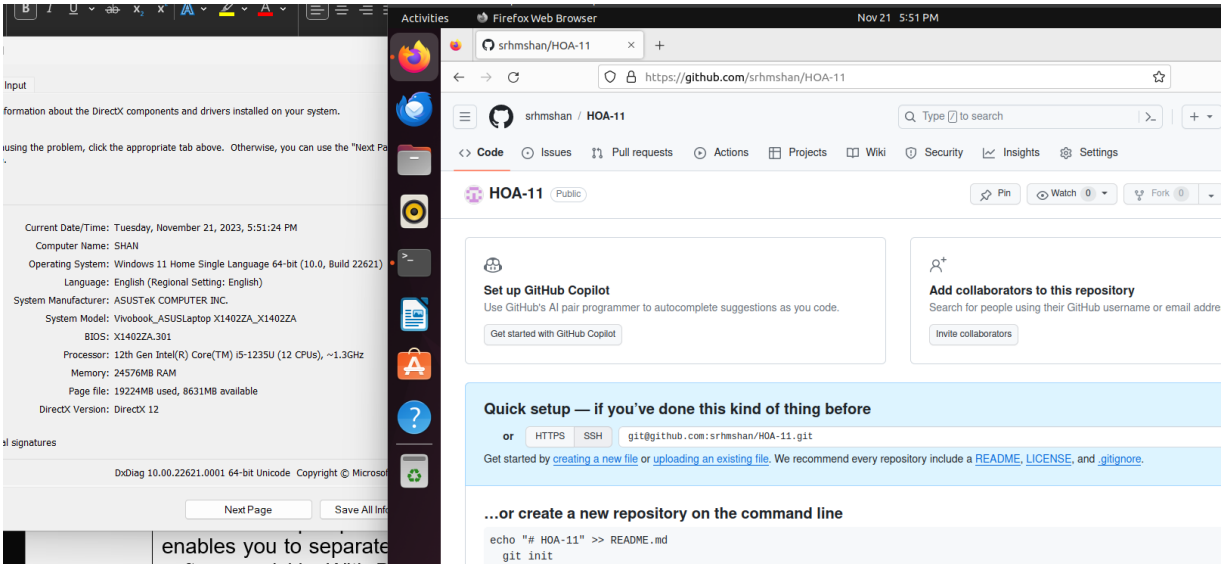
Source: https://docs.microsoft.com/en-us/virtualization/windowscontainers/about/containers-vs-vm

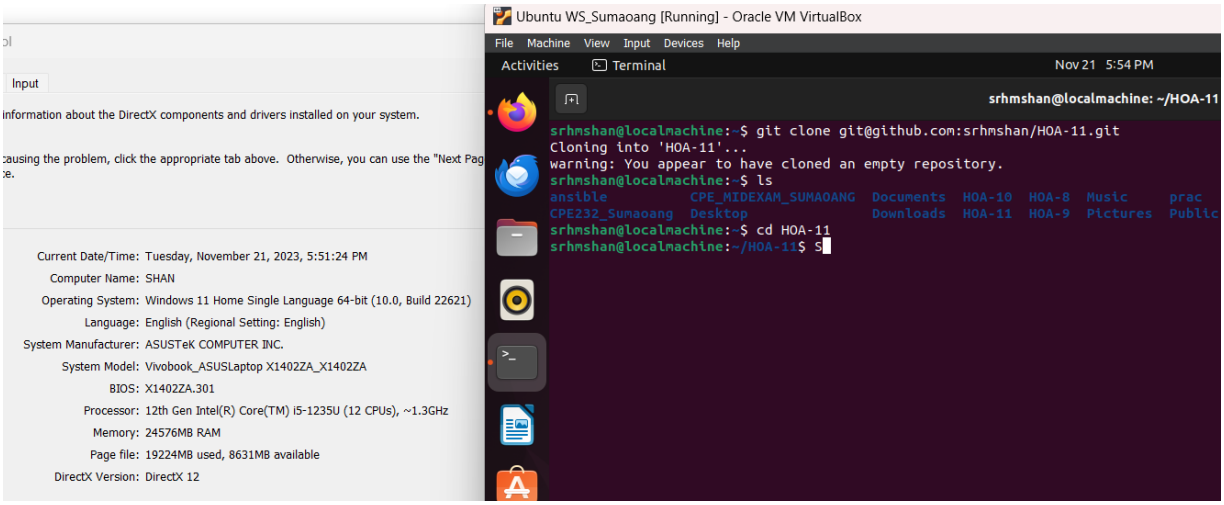
3. Tasks

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)

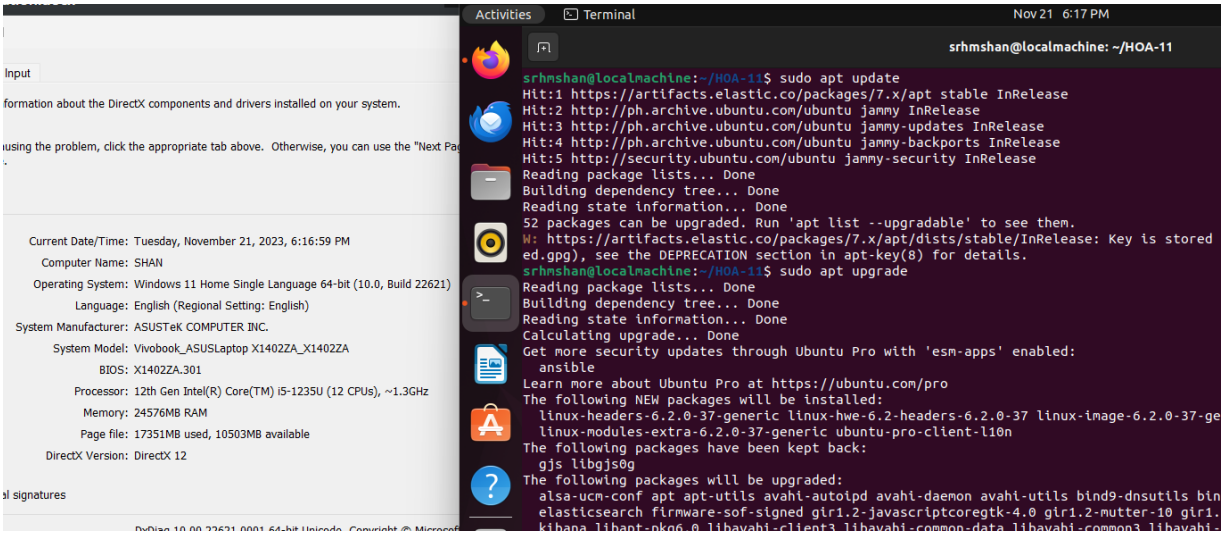
1. Create a new repository for this activity.



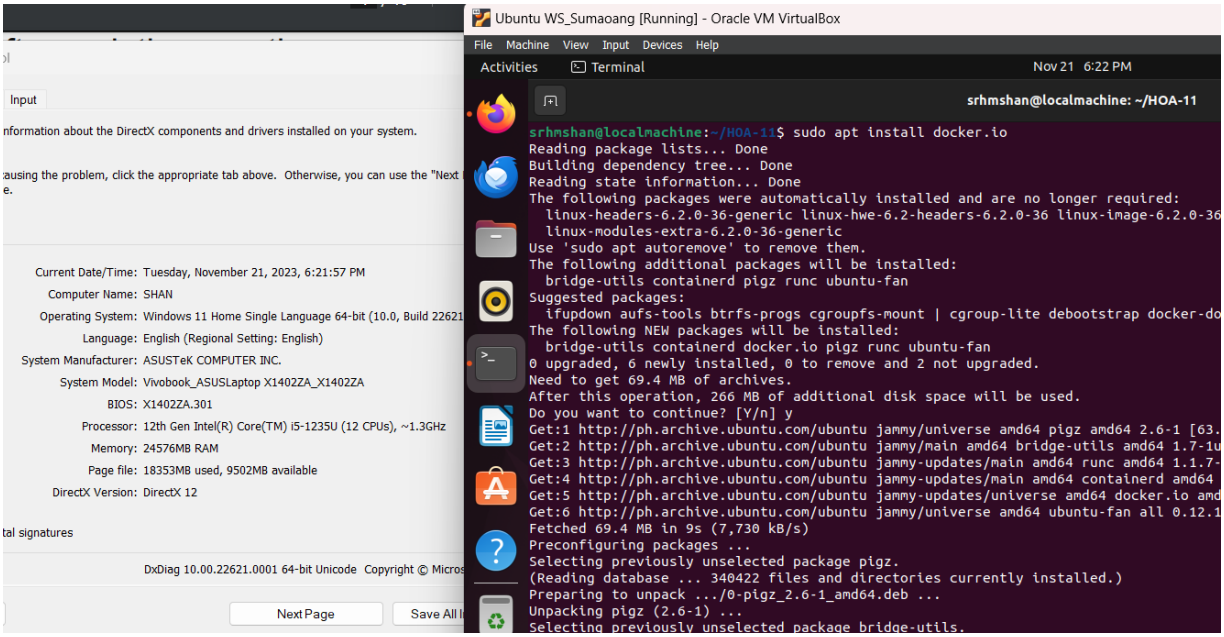


Using **git clone**, I copied and pasted the repository's link from GitHub to my local machine. I then changed my directory to HOA-11.

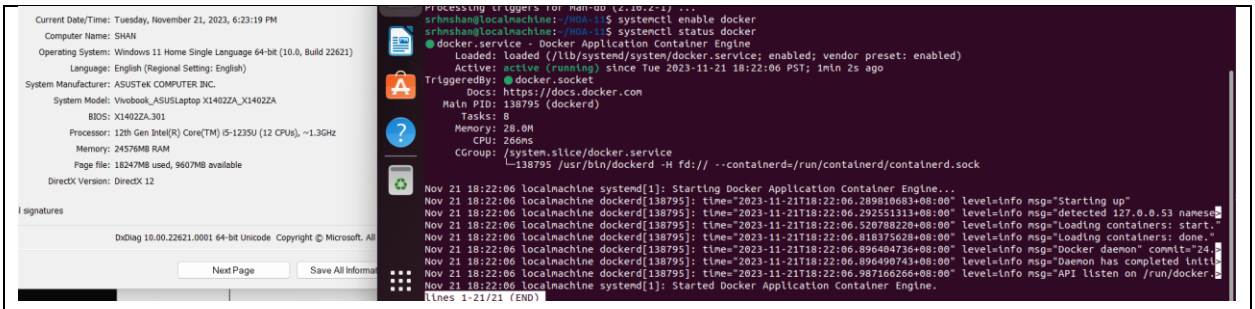
2. Install Docker and enable the docker socket.

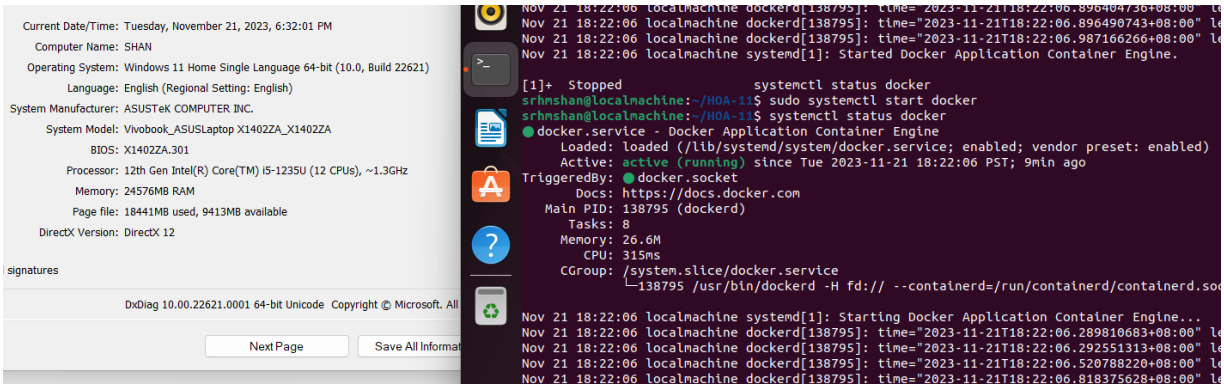


Before you start installing Docker, run the following commands: **sudo apt update** and **sudo apt upgrade**. This ensures that your system is fully updated.



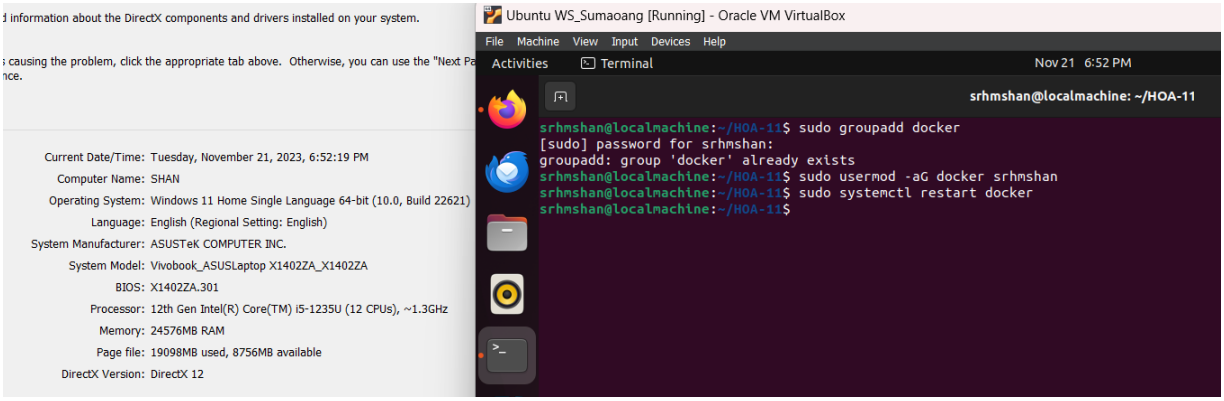
After updating, execute the command **sudo apt install docker.io** to install Docker on your local machine.





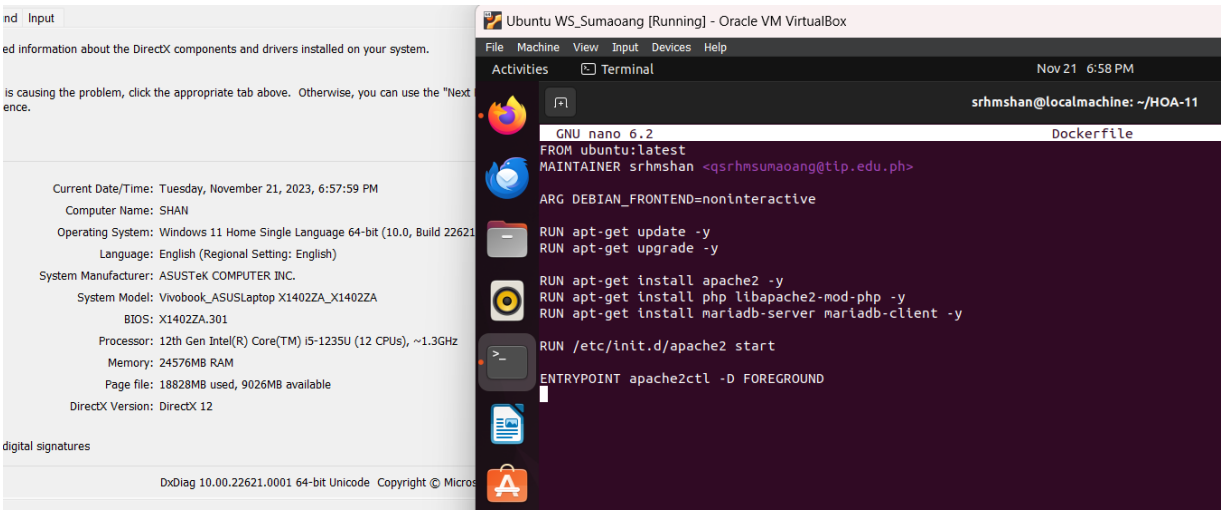
Activate Docker by using the command **systemctl enable docker**. To monitor the process status, utilize the command **systemctl status docker**.

3. Add to Docker group to your current user.



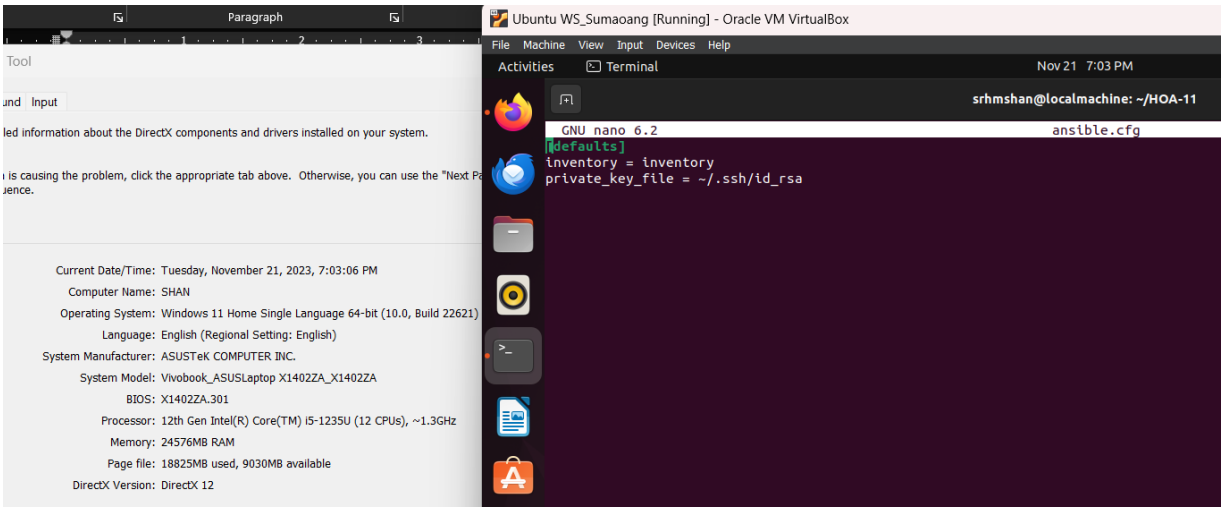
Use the command **sudo groupadd docker** to add a Docker group to you current user.

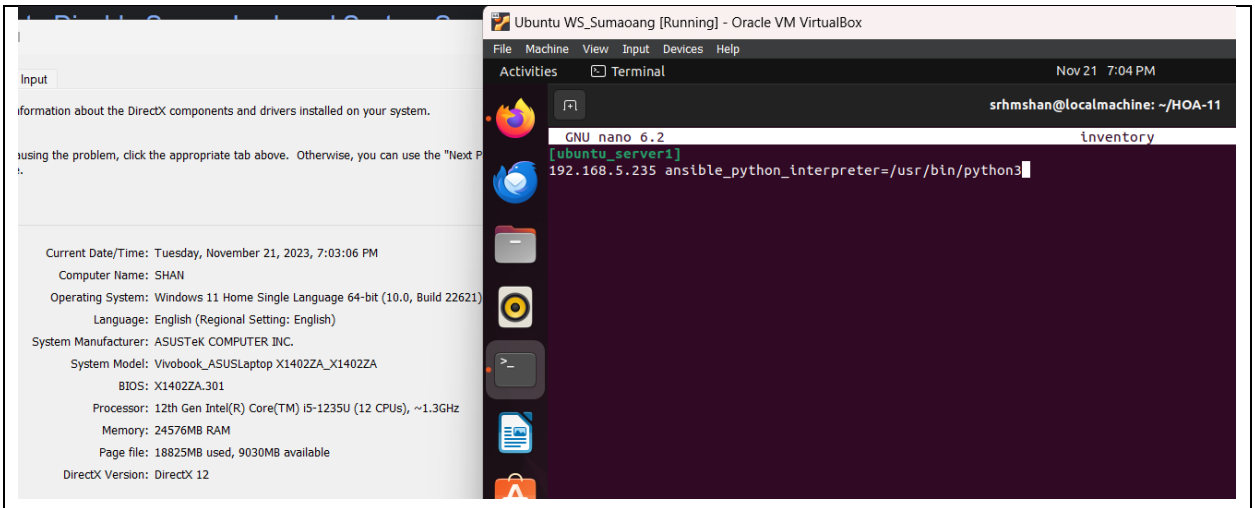
4. Create a Dockerfile to install web and DB server.



Use the command **sudo usermod -aG docker <USER>** to include the current user in the group. Once the user is added, restart the Docker process with the command **sudo systemctl restart docker**.

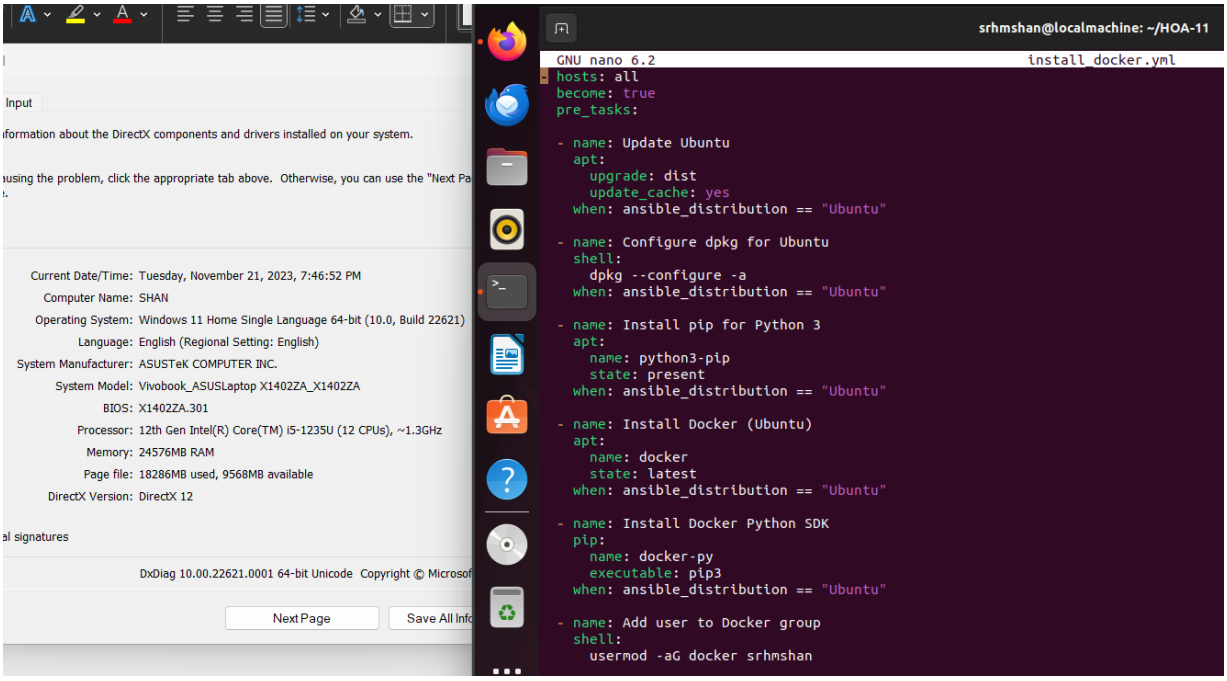
5. Install and build the Dockerfile using Ansible.

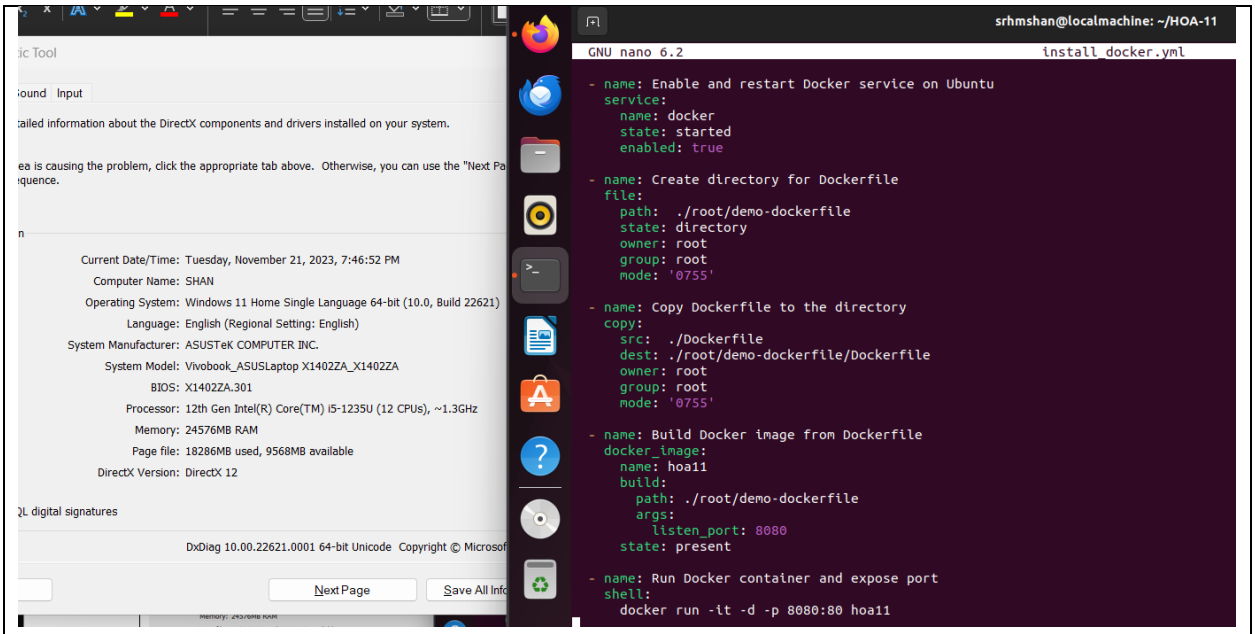




Here, I created my ansible.cfg and inventory files.

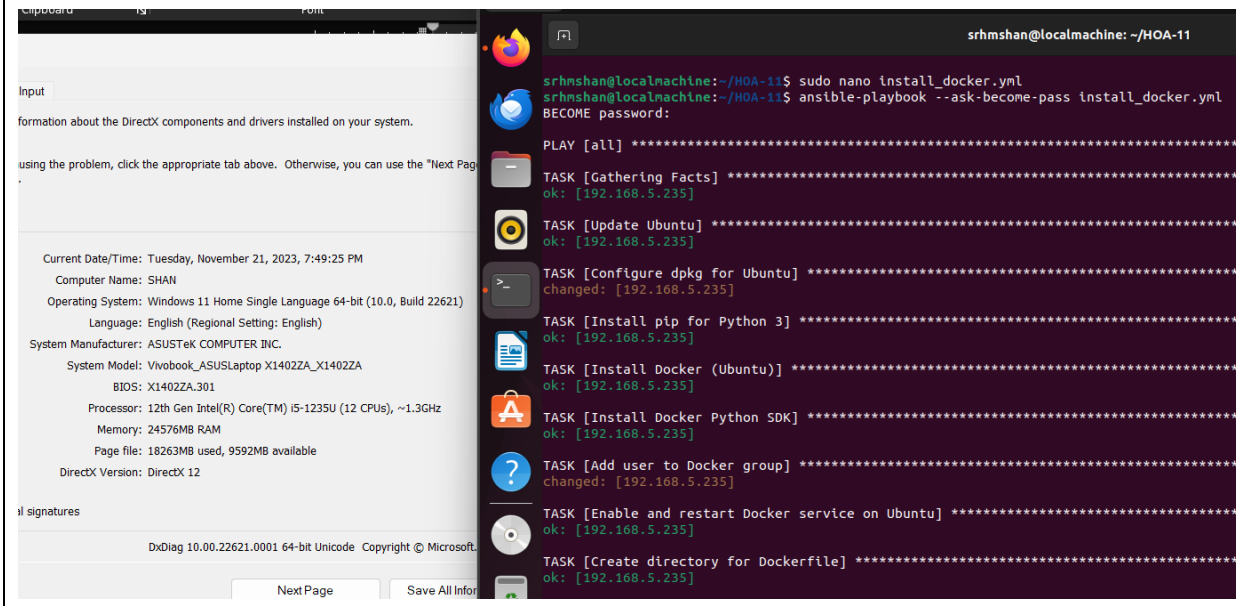
Create “install_docker.yml” playbook.



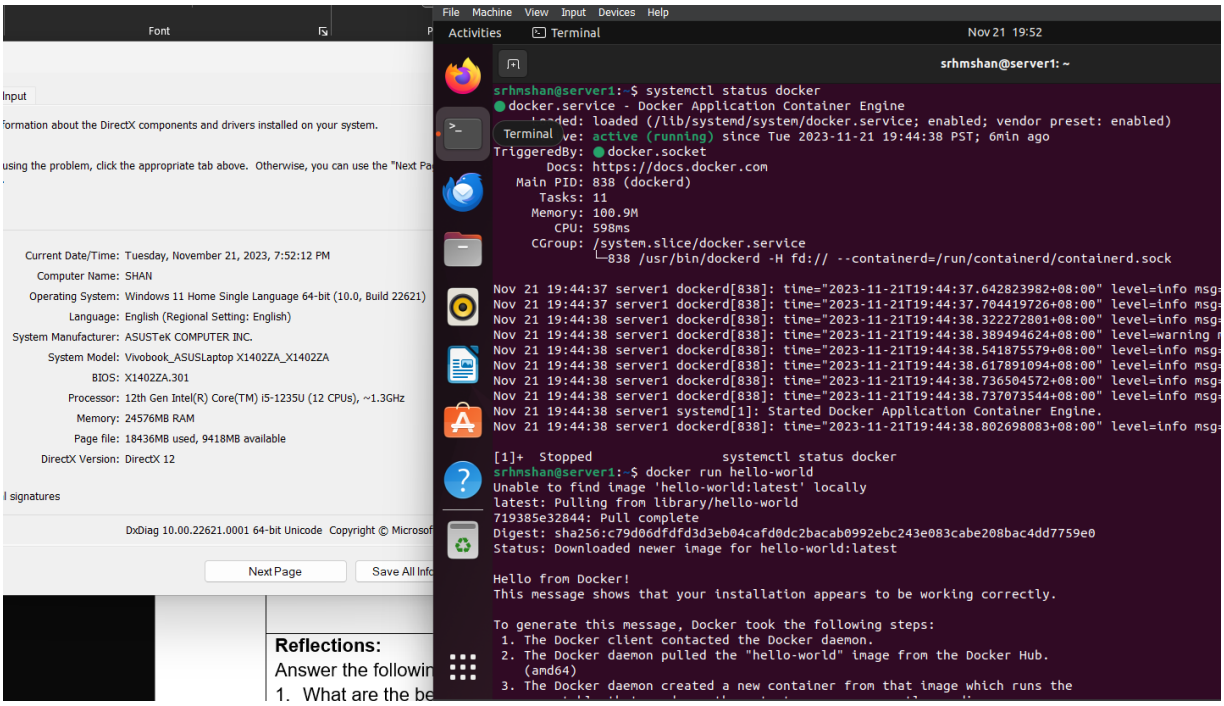


Use the **sudo nano <file name>** to create a playbook **install_docker.yml**.

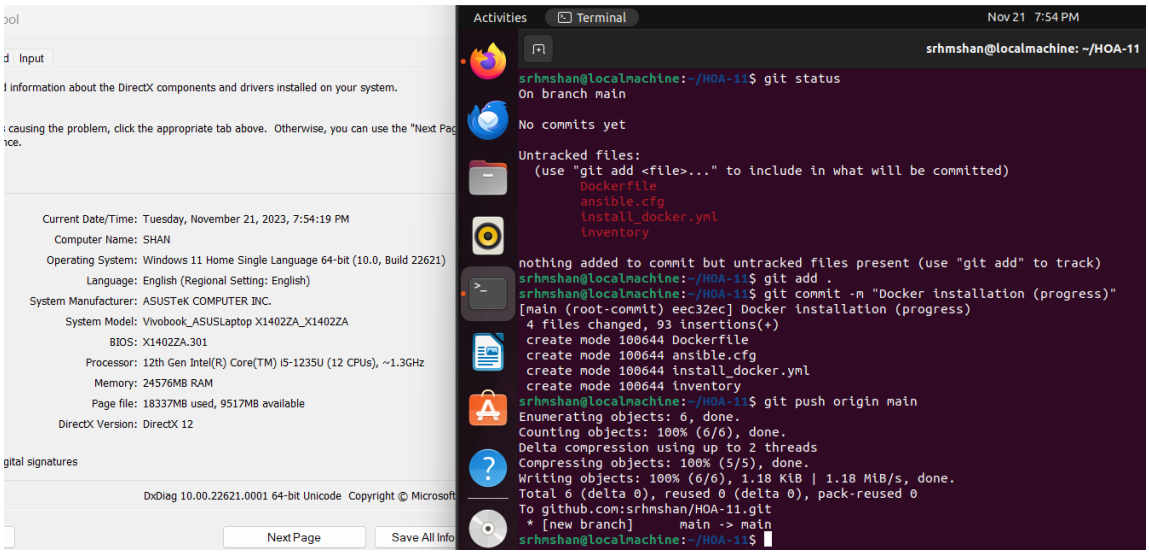
Run “install_docker.yml”

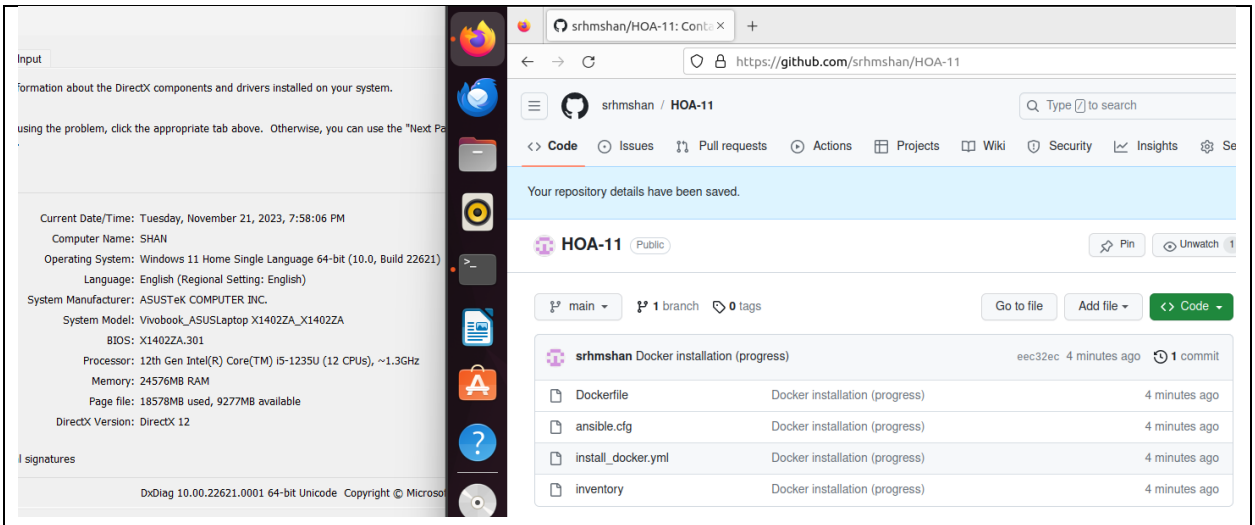


Verify successful installation:



6. Add, commit and push it to your repository.





https://github.com/srhmshan/HOA-11

Reflections:

Answer the following:

1. What are the benefits of implementing containerizations?

Using Docker to implement containers or containerization can enhance the portability of processes or entire software systems resulting in efficiency. Moreover, containerization also enhances security and management by ensuring that files or images are stateless meaning that data is not stored.

Conclusions:

In this activity, I was able to create a Dockerfile and establish a workflow using Ansible as Infrastructure, as Code (IaC). This allowed for a Continuous Delivery process for Server1. The main challenge arose when I had to incorporate a software layer while using Ansible.

Working through the process of creating a Dockerfile and seamlessly integrating it into the framework posed some difficulties. However, by troubleshooting and resolving errors, I successfully managed the synergy between Ansible and Docker.

Despite the challenges of this activity, I was able to complete the integration within the given timeframe. This experience greatly improved my proficiency in handling these tasks. I was also able to provide insights, into implementing Containerization.