Name: Rustom C. Cariño	Date Performed: 11/21/2023
Course/Section: CPE31S5	Date Submitted: 11/23/2023
Instructor: Engr. Roman Richard	Semester and SY: 1st semester 2023-20234

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.

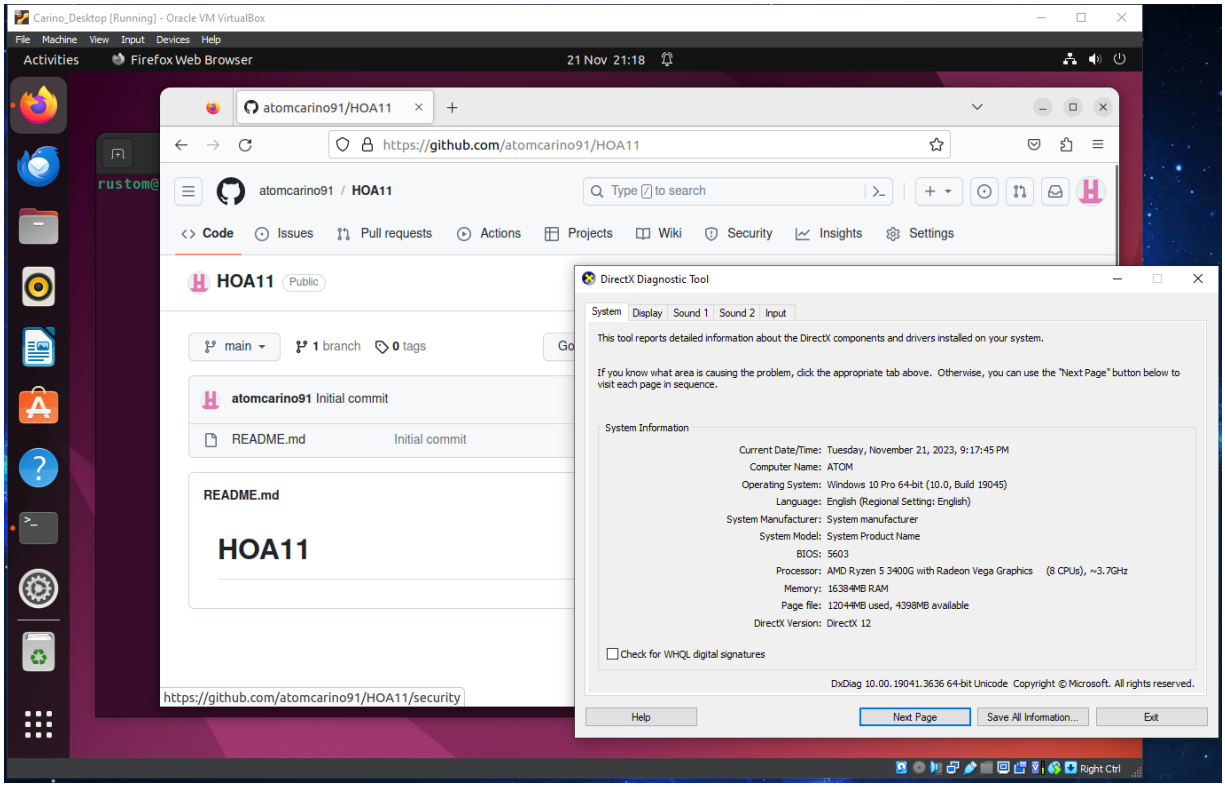
Source: https://docs.microsoft.com/en-us/virtualization/windowscontainers/about/containers-vs-vm

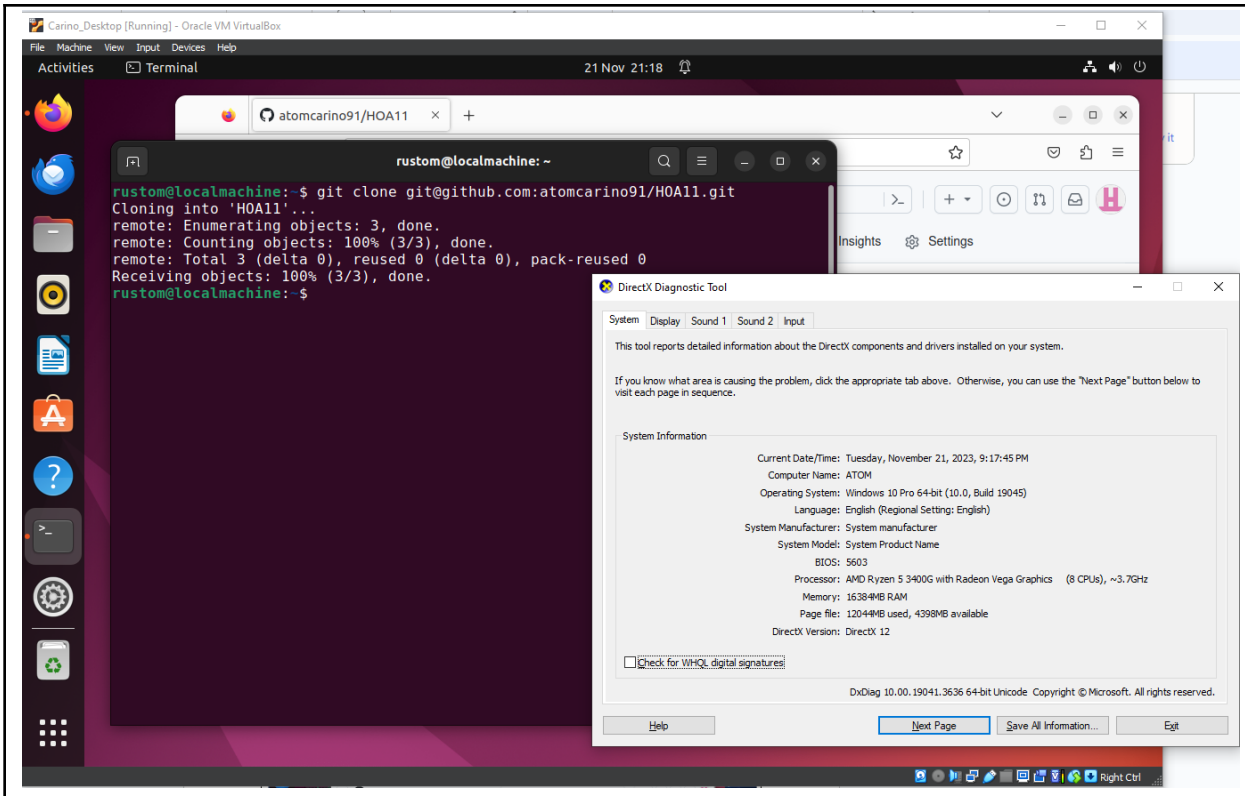
3. Tasks

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.

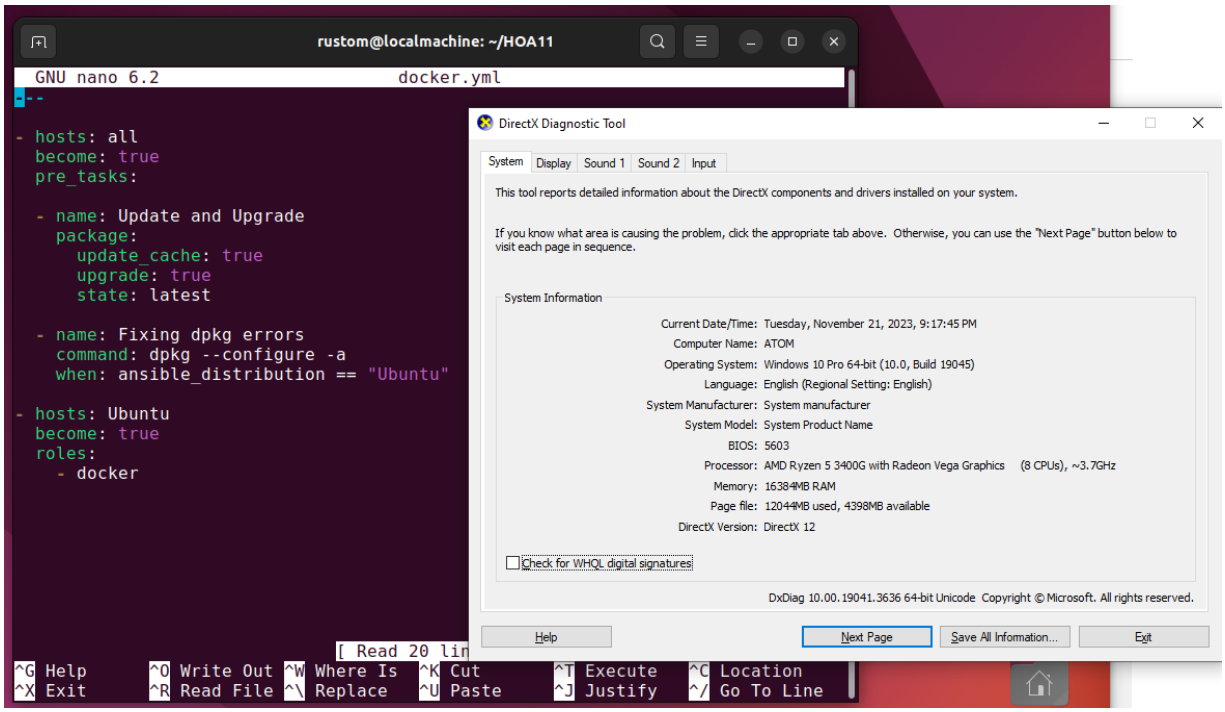
4. Output (screenshots and explanations)

1. Creating New Repository

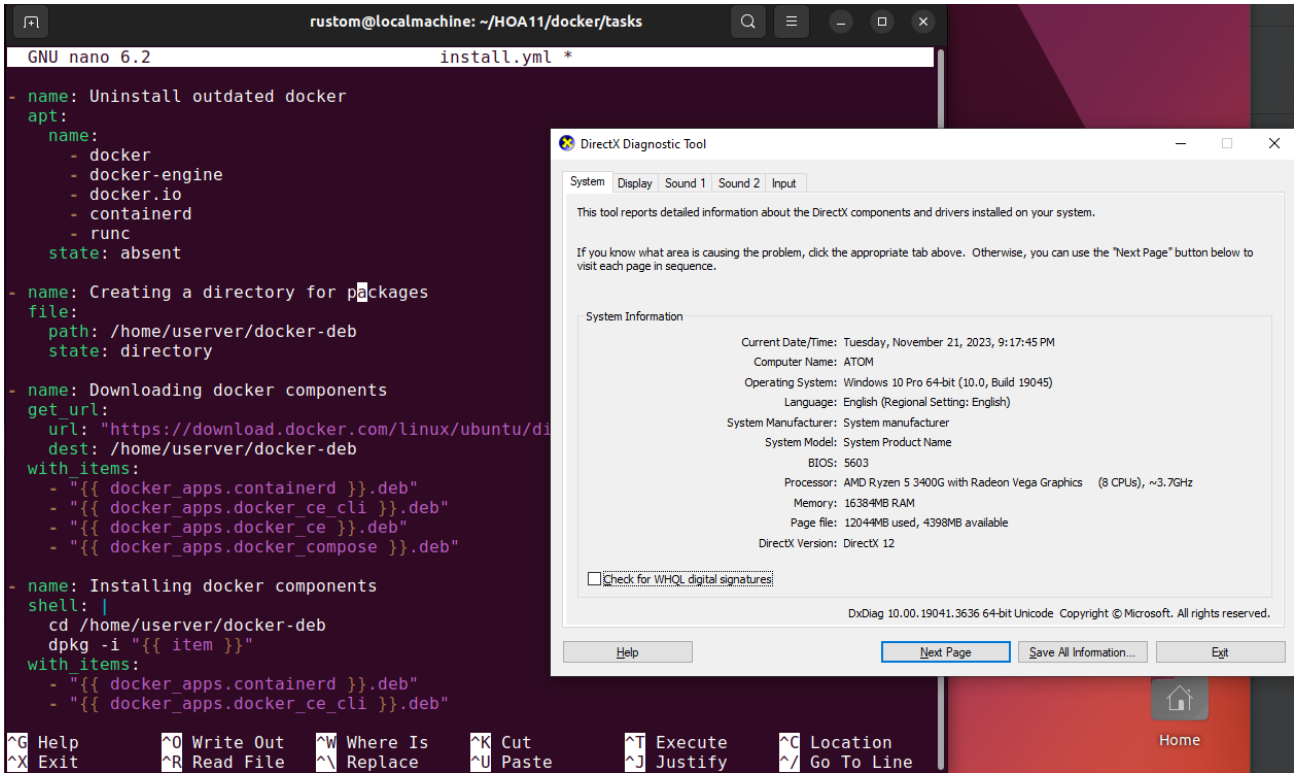




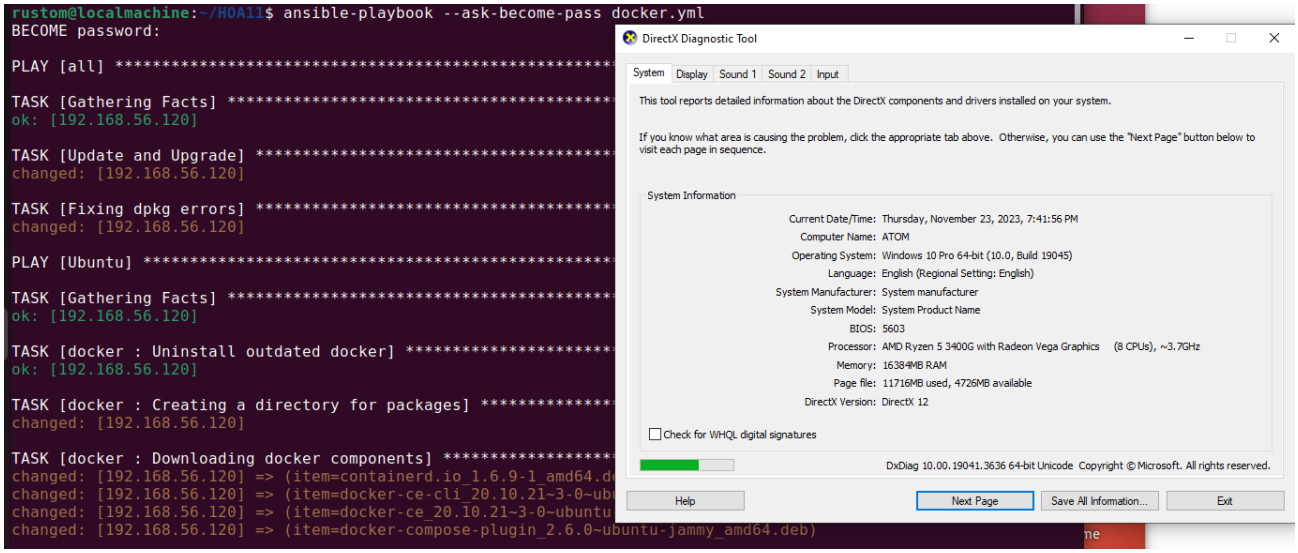
2. Creating playbook

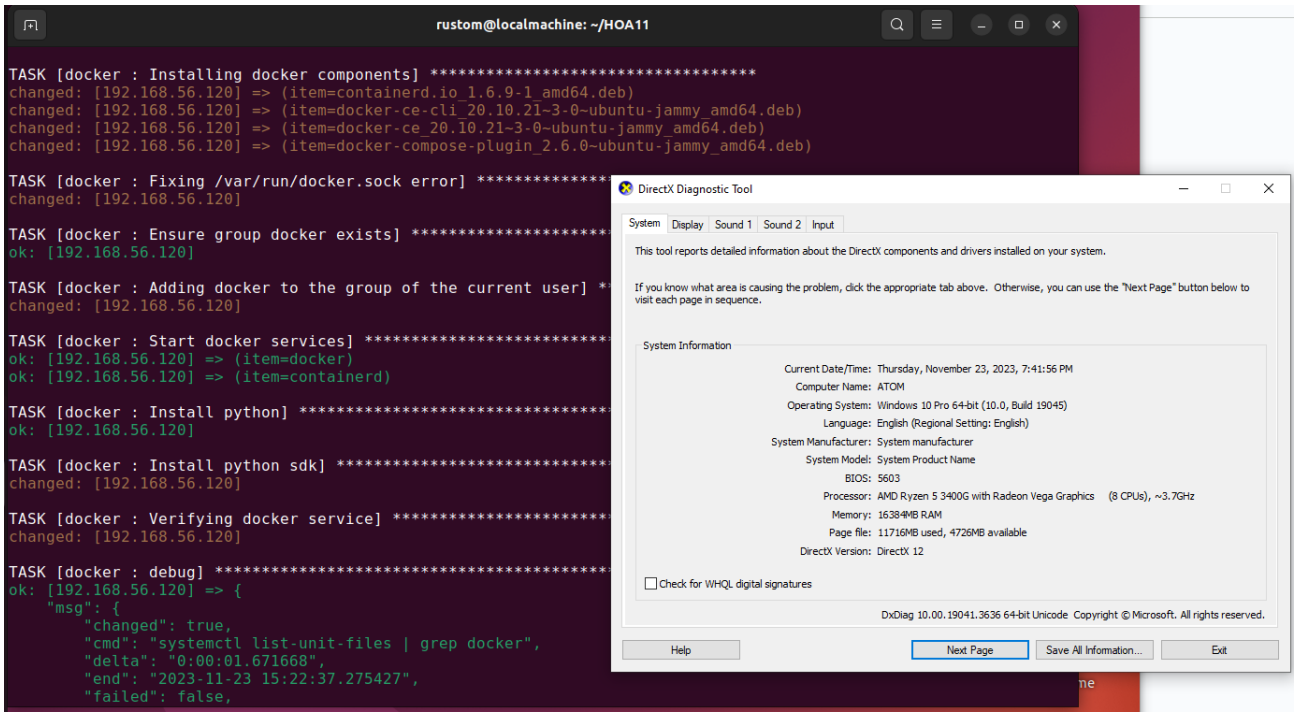


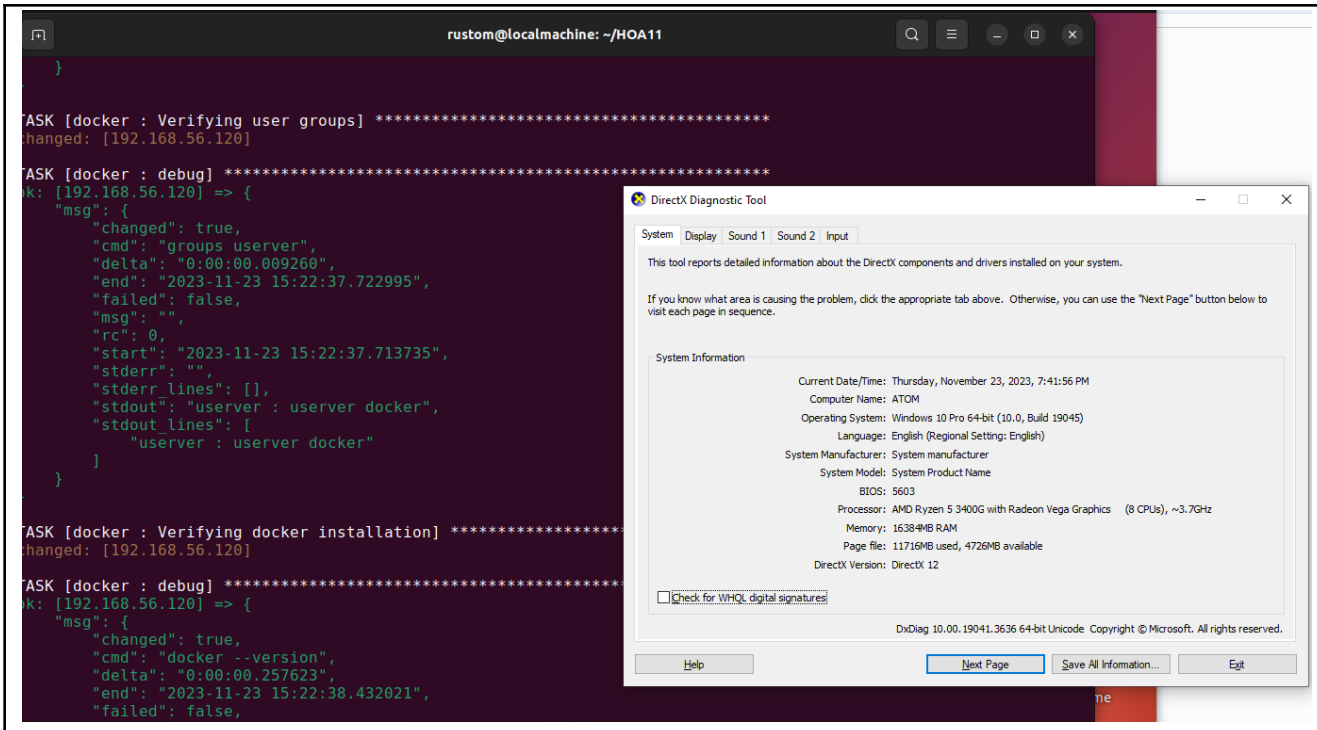
3. Install docker

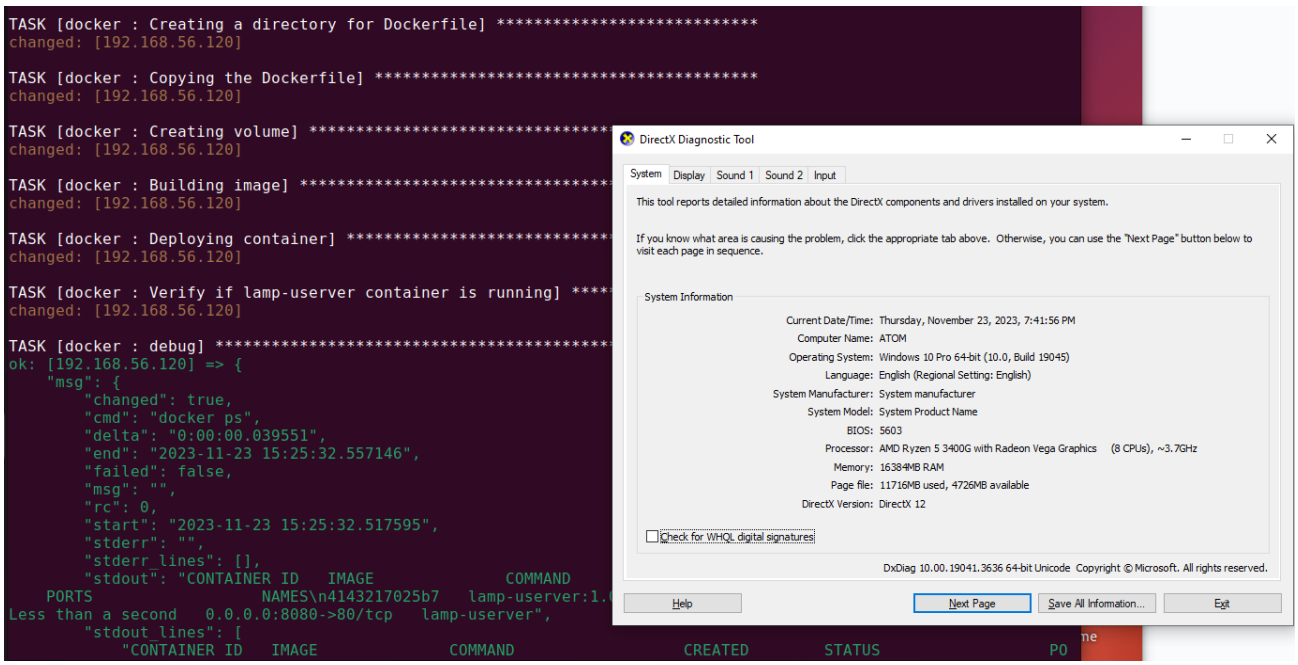


4. Running playbook

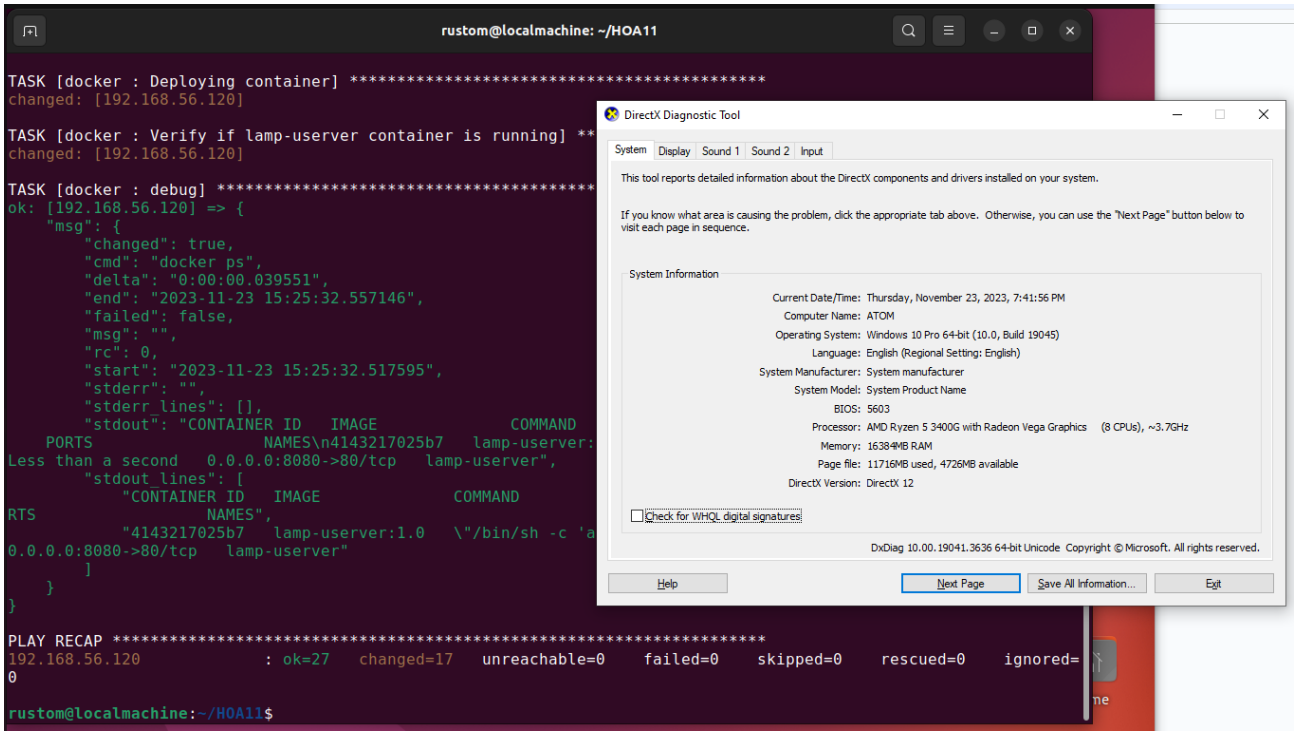




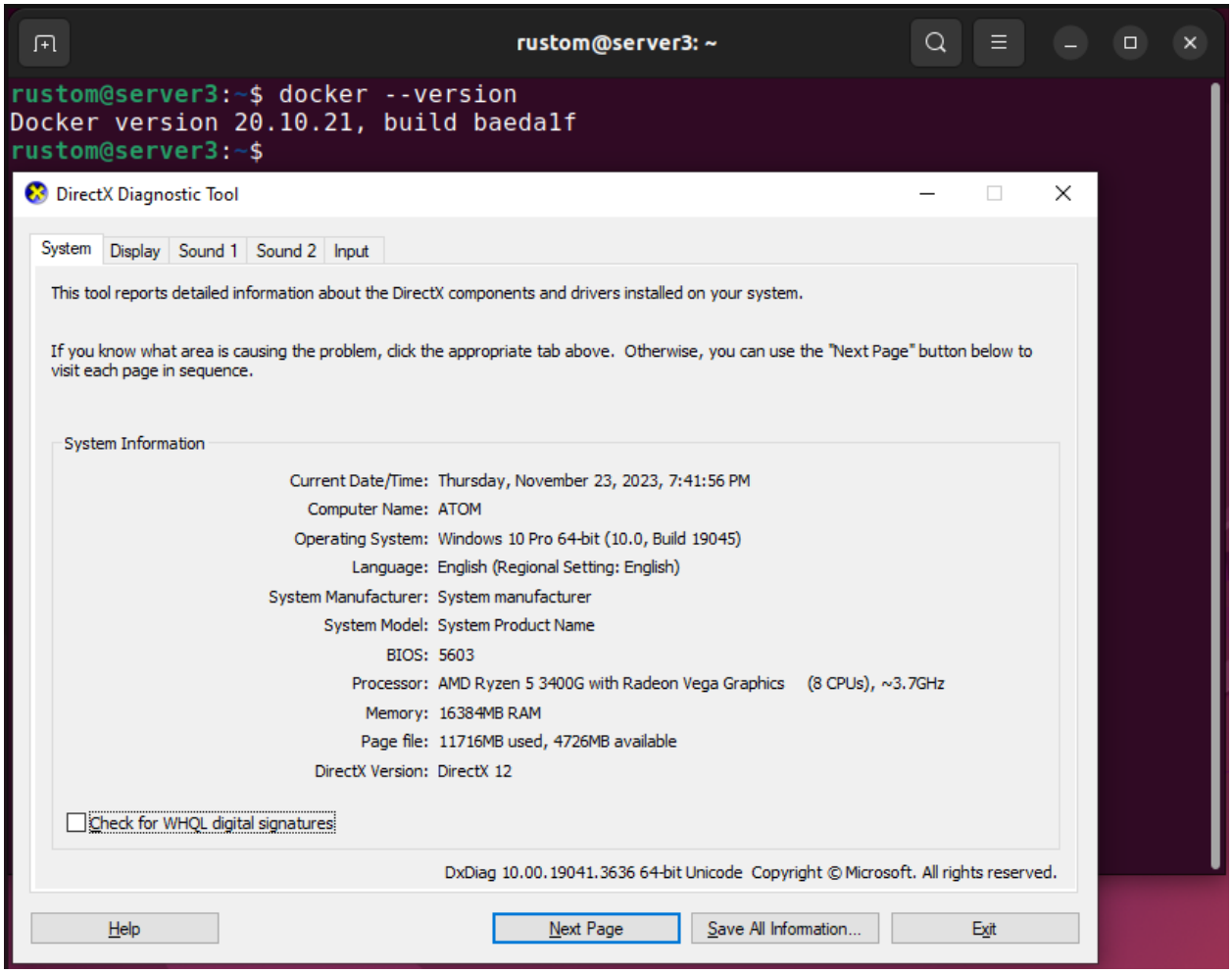

rustom@localmachine: ~/HOA11
}
TASK [docker : Verifying user groups] **
changed: [192.168.56.120]
TASK [docker : debug] **
ok: [192.168.56.120] => {
"msg": {
"changed": true,
"cmd": "groups userver",
"delta": "0:00:00.009260",
"end": "2023-11-23 15:22:37.722995",
"failed": false,
"msg": "",
"rc": 0,
"start": "2023-11-23 15:22:37.713735",
"stderr": "",
"stderr_lines": [],
"stdout": "userver : userver docker",
"stdout_lines": [
"userver : userver docker"
]
}
TASK [docker : Verifying docker installation] ********************
changed: [192.168.56.120]
TASK [docker : debug] **
ok: [192.168.56.120] => {
"msg": {
"changed": true,
"cmd": "docker --version",
"delta": "0:00:00.257623",
"end": "2023-11-23 15:22:38.432021",
"failed": false,
DirectX Diagnostic Tool
System
Display
Sound 1
Sound 2
Input
This tool reports detailed information about the DirectX components and drivers installed on your system.
If you know what area is causing the problem, click the appropriate tab above. Otherwise, you can use the "Next Page" button below to visit each page in sequence.
System Information
Current Date/Time: Thursday, November 23, 2023, 7:41:56 PM
Computer Name: ATOM
Operating System: Windows 10 Pro 64-bit (10.0, Build 19045)
Language: English (Regional Setting: English)
System Manufacturer: System manufacturer
System Model: System Product Name
BIOS: 5603
Processor: AMD Ryzen 5 3400G with Radeon Vega Graphics (8 CPUs), ~3.7GHz
Memory: 16384MB RAM
Page file: 11716MB used, 4726MB available
DirectX Version: DirectX 12
Check for WHQL digital signatures
DxDiag 10.00.19041.3636 64-bit Unicode Copyright © Microsoft. All rights reserved.
Help
Next Page
Save All Information...
Exit

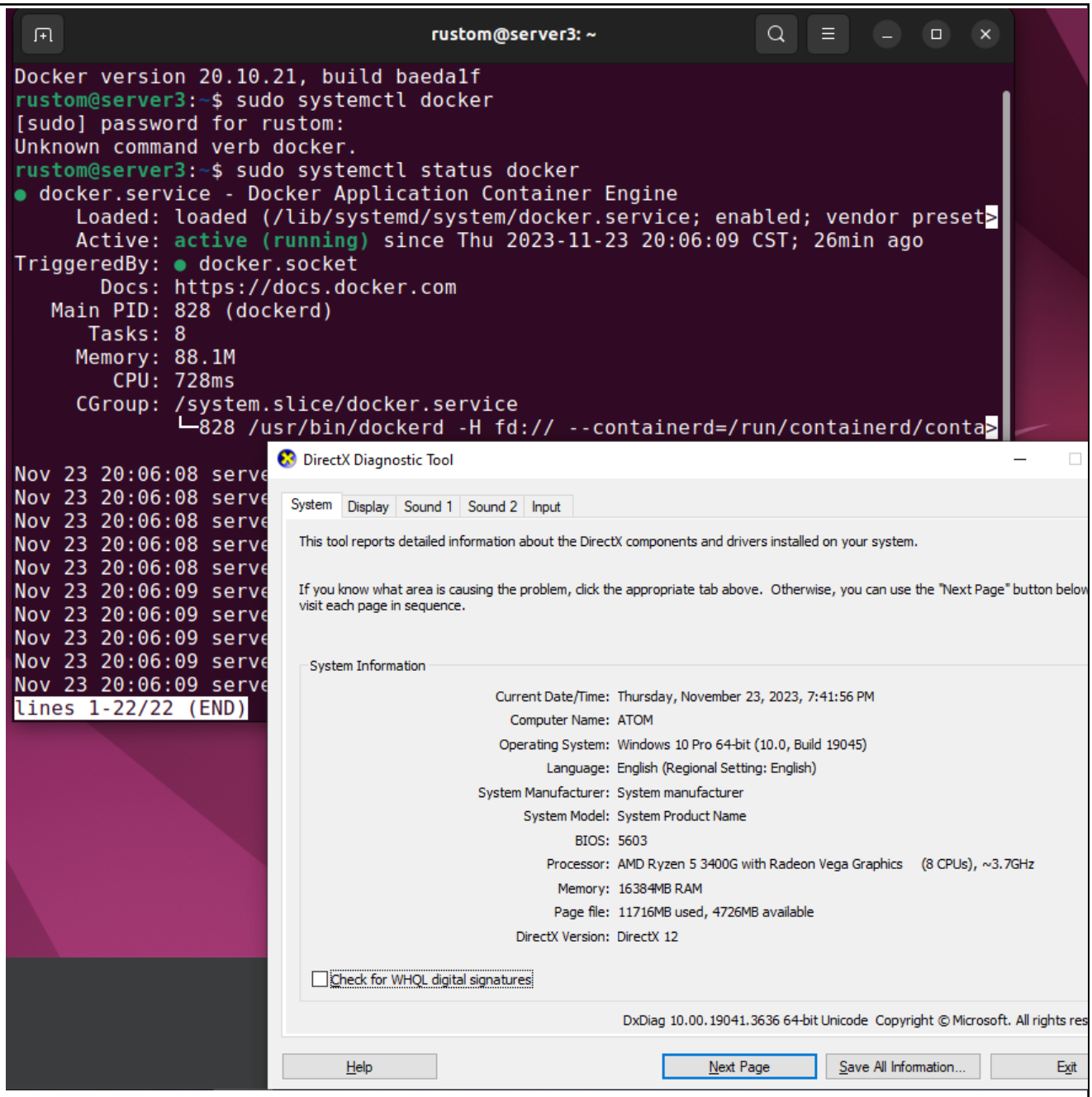

TASK [docker : Creating a directory for Dockerfile] ***************************
changed: [192.168.56.120]
TASK [docker : Copying the Dockerfile] ***************************************
changed: [192.168.56.120]
TASK [docker : Creating volume] **********************************
changed: [192.168.56.120]
TASK [docker : Building image] ***********************************
changed: [192.168.56.120]
TASK [docker : Deploying container] ******************************
changed: [192.168.56.120]
TASK [docker : Verify if lamp-userver container is running] ******
changed: [192.168.56.120]
TASK [docker : debug] **
ok: [192.168.56.120] => {
"msg": {
"changed": true,
"cmd": "docker ps",
"delta": "0:00:00.039551",
"end": "2023-11-23 15:25:32.557146",
"failed": false,
"msg": "",
"rc": 0,
"start": "2023-11-23 15:25:32.517595",
"stderr": "",
"stderr_lines": [],
"stdout": "CONTAINER ID IMAGE COMMAND
PORTS NAMES\n4143217025b7 lamp-userver:1.
Less than a second 0.0.0.0:8080->80/tcp lamp-userver",
"stdout_lines": [
"CONTAINER ID IMAGE COMMAND CREATED STATUS PO
DirectX Diagnostic Tool
System
Display
Sound 1
Sound 2
Input
This tool reports detailed information about the DirectX components and drivers installed on your system.
If you know what area is causing the problem, click the appropriate tab above. Otherwise, you can use the "Next Page" button below to visit each page in sequence.
System Information
Current Date/Time: Thursday, November 23, 2023, 7:41:56 PM
Computer Name: ATOM
Operating System: Windows 10 Pro 64-bit (10.0, Build 19045)
Language: English (Regional Setting: English)
System Manufacturer: System manufacturer
System Model: System Product Name
BIOS: 5603
Processor: AMD Ryzen 5 3400G with Radeon Vega Graphics (8 CPUs), ~3.7GHz
Memory: 16384MB RAM
Page file: 11716MB used, 4726MB available
DirectX Version: DirectX 12
Check for WHQL digital signatures
DxDiag 10.00.19041.3636 64-bit Unicode Copyright © Microsoft. All rights reserved.
Help
Next Page
Save All Information...
Exit

5. Playbook result

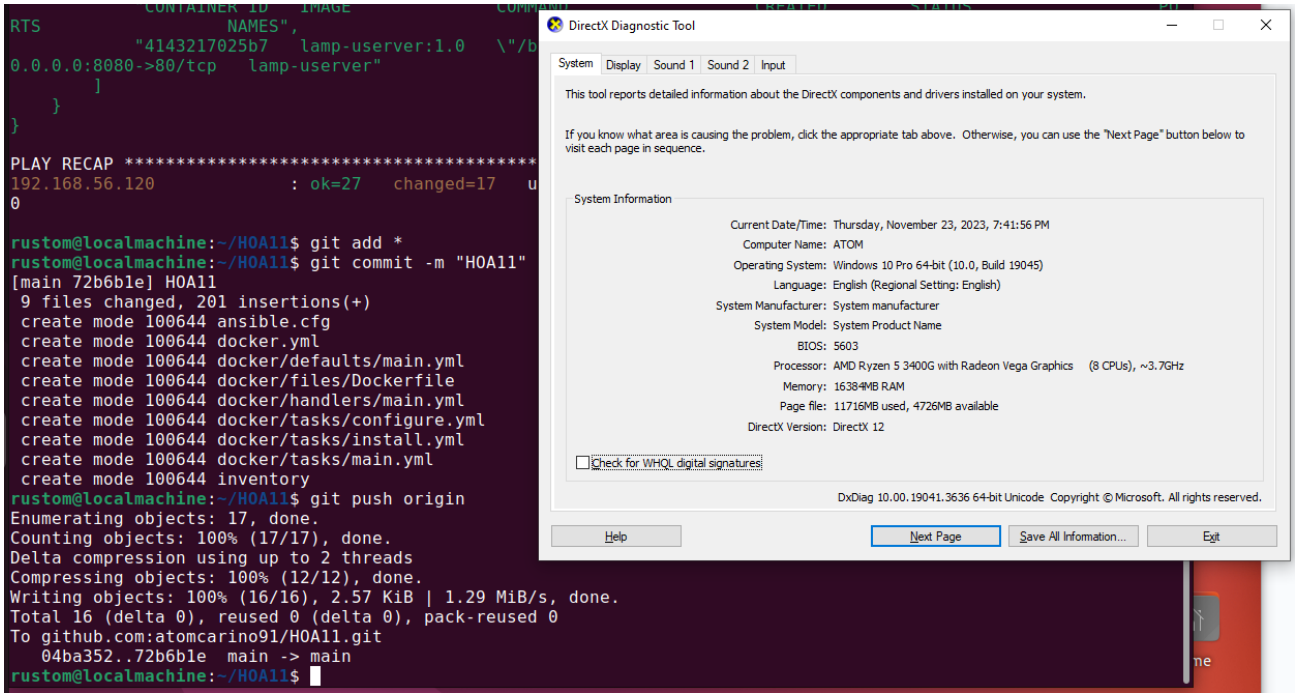


6. Proof that docker is installed in server.




rustom@server3: ~
Docker version 20.10.21, build baeda1f
rustom@server3:~$ sudo systemctl docker
[sudo] password for rustom:
Unknown command verb docker.
rustom@server3:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset
Active: active (running) since Thu 2023-11-23 20:06:09 CST; 26min ago
TriggeredBy: ● docker.socket
Docs: https://docs.docker.com
Main PID: 828 (dockerd)
Tasks: 8
Memory: 88.1M
CPU: 728ms
CGroup: /system.slice/docker.service
└828 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/conta
Nov 23 20:06:08 serve
Nov 23 20:06:08 serve
Nov 23 20:06:08 serve
Nov 23 20:06:08 serve
Nov 23 20:06:08 serve
Nov 23 20:06:09 serve
Nov 23 20:06:09 serve
Nov 23 20:06:09 serve
Nov 23 20:06:09 serve
Nov 23 20:06:09 serve
lines 1-22/22 (END)
DirectX Diagnostic Tool
System
Display
Sound 1
Sound 2
Input
This tool reports detailed information about the DirectX components and drivers installed on your system.
If you know what area is causing the problem, click the appropriate tab above. Otherwise, you can use the "Next Page" button below
visit each page in sequence.
System Information
Current Date/Time: Thursday, November 23, 2023, 7:41:56 PM
Computer Name: ATOM
Operating System: Windows 10 Pro 64-bit (10.0, Build 19045)
Language: English (Regional Setting: English)
System Manufacturer: System manufacturer
System Model: System Product Name
BIOS: 5603
Processor: AMD Ryzen 5 3400G with Radeon Vega Graphics (8 CPUs), ~3.7GHz
Memory: 16384MB RAM
Page file: 11716MB used, 4726MB available
DirectX Version: DirectX 12
Check for WHQL digital signatures
DxDiag 10.00.19041.3636 64-bit Unicode Copyright © Microsoft. All rights res
Help
Next Page
Save All Information...
Exit

7. Add, commit and push to your repository.



Reflections:

Answer the following:

1. **What are the benefits of implementing containerizations?**

   Containerization in Ubuntu offers numerous benefits when implemented, including increased resource efficiency, improved application portability, enhanced scalability, faster development and deployment, simplified application management, improved isolation and security, support for microservices architecture, reduced vendor lock-in, enhanced collaboration and productivity, and future-proofing due to its strong community and industry support. These benefits make containers an ideal choice for businesses.

Conclusions:

In this activity I will be able to create a dockerfile and form a workflow using ansible as infrastructure as code to enable continuous delivery process.This activity involves installing Docker using an Ansible Playbook. It involves installing Docker, enabling it, building a Docker file, and deploying the container. The task is slightly easy, except for debugging and creating a new server due to a corrupted first server. The activity is knowledgeable in Docker installation, image building, and container deployment.