Name: Juanson, Aliya Dane P.	Date Performed: October 16, 2024
Course/Section: CPE31S2	Date Submitted: October 28,2024
Instructor: Sir Robin Valenzuela	Semester and SY: 2024-2025

Activity 9: Install, Configure, and Manage Performance Monitoring tools

1. Objectives

Create and design a workflow that installs, configure and manage enterprise performance tools using Ansible as an Infrastructure as Code (IaC) tool.

2. Discussion

Performance monitoring is a type of monitoring tool that identifies current resource consumption of the workload, in this page we will discuss multiple performance monitoring tool.

Prometheus

Prometheus fundamentally stores all data as timeseries: streams of timestamped values belonging to the same metric and the same set of labeled dimensions. Besides stored time series, Prometheus may generate temporary derived time series as the result of queries. Source: Prometheus - Monitoring system & time series database

Cacti

Cacti is a complete network graphing solution designed to harness the power of RRDTool's data storage and graphing functionality. Cacti provides a fast poller, advanced graph templating, multiple data acquisition methods, and user management features out of the box. All of this is wrapped in an intuitive, easy to use interface that makes sense for LAN-sized installations up to complex networks with thousands of devices. Source: Cacti® - The Complete RRDTool-based Graphing Solution

3. Tasks

1. Create a playbook that installs Prometheus in both Ubuntu and CentOS. Apply the concept of creating roles.
2. Describe how you did step 1. (Provide screenshots and explanations in your report. Make your report detailed such that it will look like a manual.)
3. Show an output of the installed Prometheus for both Ubuntu and CentOS.
4. Make sure to create a new repository in GitHub for this activity.

4. Output (screenshots and explanations)

Create a repository containing ansibe.cfg, inventory and roles.

```
qadjuanson@workstation:~/HOA9$ ls
ansible.cfg  install.yml  inventory  roles
```

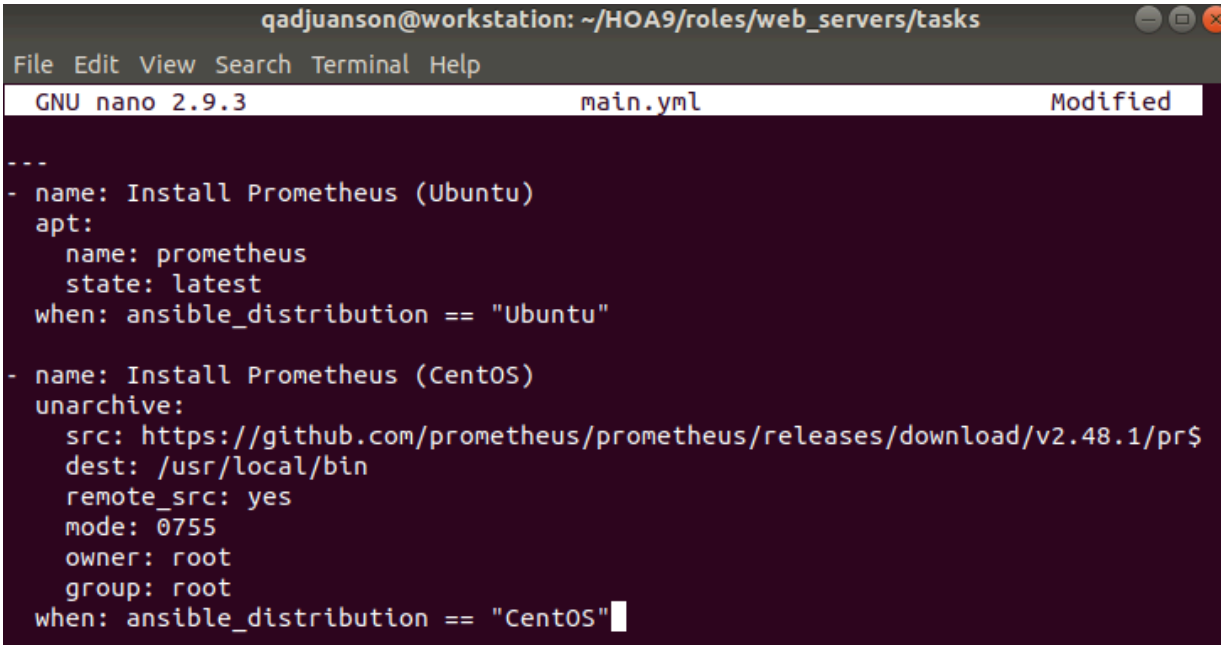
```
  GNU nano 2.9.3                          ansible.cfg

[defaults]
inventory = ~/HOA9/inventory
remote_user = qadjuanson
host_key_checking = True
```

```
  GNU nano 2.9.3                          inventory

[workstations]
192.168.56.116

[web_servers]
192.168.56.116
192.168.56.115 ansible_user=qadjuanson

[db_servers]
192.168.56.117
192.168.56.115 ansible_user=qadjuanson

[file_servers]
192.168.56.118
```

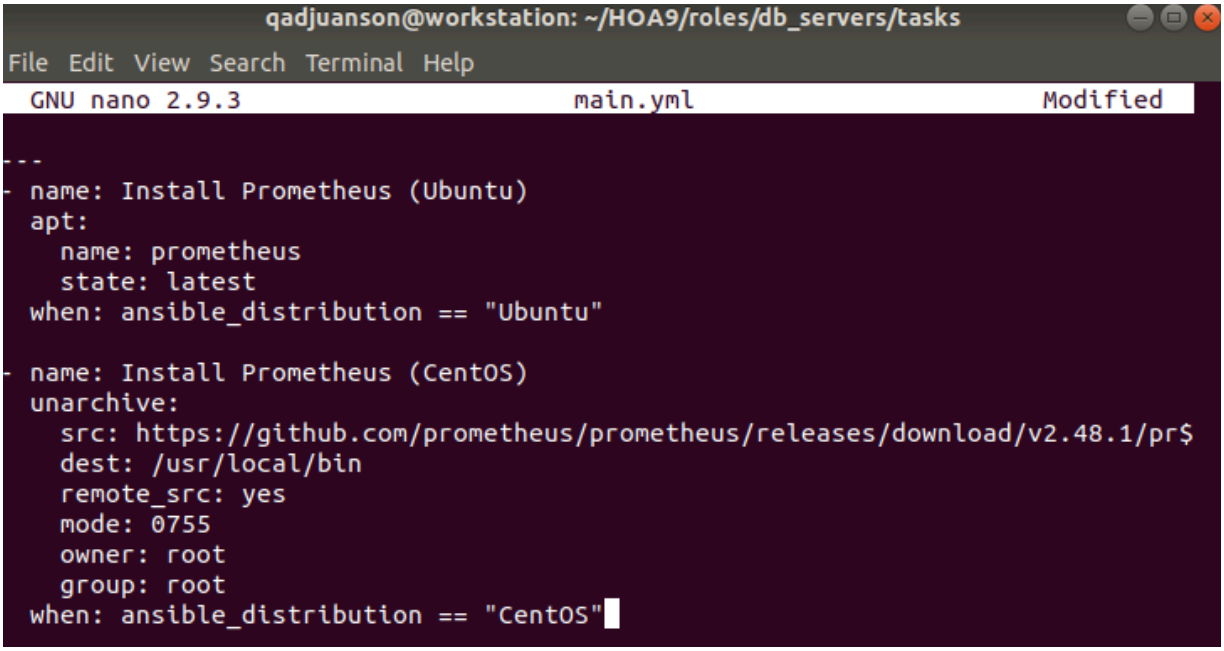
Inside the directory roles, create a directory named web_servers, db_servers, file_servers and workstations. Inside of each directory create a directory named tasks.

```
qadjuanson@workstation:~/HOA9/roles$ mkdir web_servers
qadjuanson@workstation:~/HOA9/roles$ cd web_servers
qadjuanson@workstation:~/HOA9/roles/web_servers$ mkdir tasks
qadjuanson@workstation:~/HOA9/roles/web_servers$ ls
tasks
qadjuanson@workstation:~/HOA9/roles/web_servers$ cd ..
qadjuanson@workstation:~/HOA9/roles$ mkdir db_servers
qadjuanson@workstation:~/HOA9/roles$ cd db_servers
qadjuanson@workstation:~/HOA9/roles/db_servers$ mkdir tasks
qadjuanson@workstation:~/HOA9/roles/db_servers$ ls
tasks
qadjuanson@workstation:~/HOA9/roles/db_servers$ cd ..
qadjuanson@workstation:~/HOA9/roles$ mkdir file_servers
qadjuanson@workstation:~/HOA9/roles$ cd file_servers
qadjuanson@workstation:~/HOA9/roles/file_servers$ mkdir tasks
qadjuanson@workstation:~/HOA9/roles/file_servers$ ls
tasks
qadjuanson@workstation:~/HOA9/roles/file_servers$ cd ..
qadjuanson@workstation:~/HOA9/roles$ mkdir workstations
qadjuanson@workstation:~/HOA9/roles$ cd workstations
qadjuanson@workstation:~/HOA9/roles/workstations$ mkdir tasks
qadjuanson@workstation:~/HOA9/roles/workstations$ ls
tasks
```

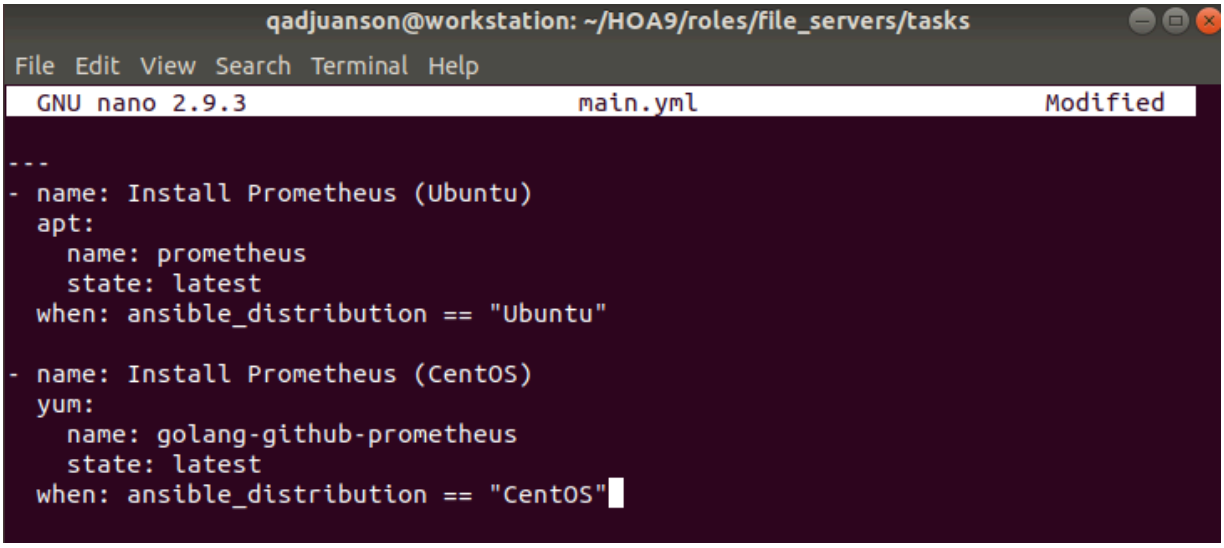
Inside the ~/HOA9/roles/web_servers/tasks create a main.yml.



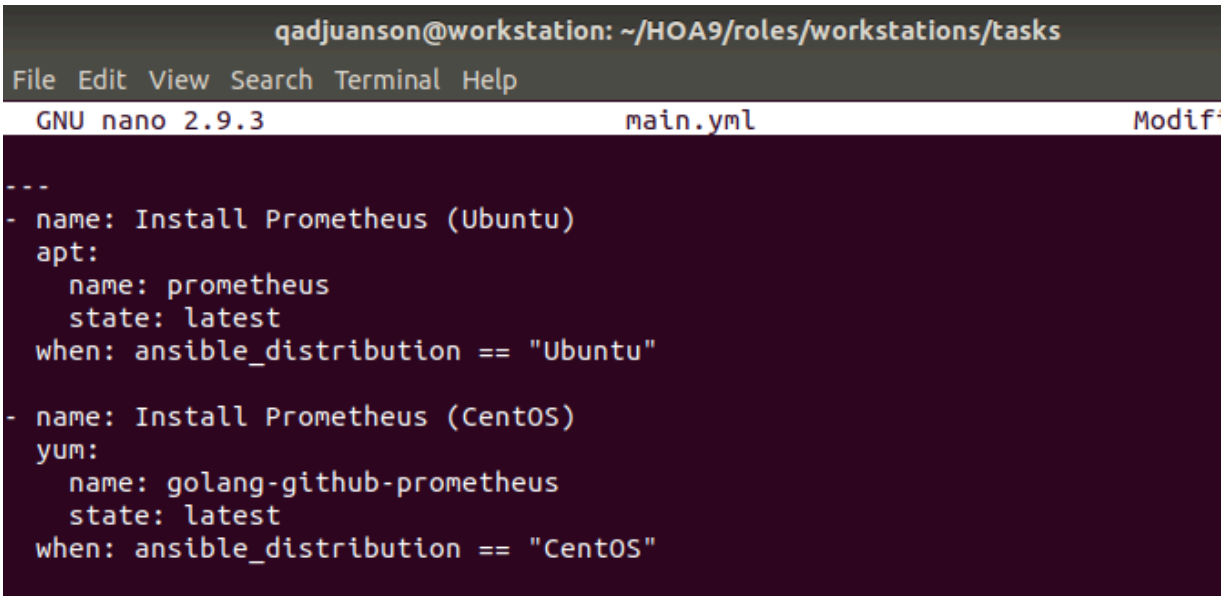
Inside the ~/HOA9/roles/db_servers/tasks create a main.yml.



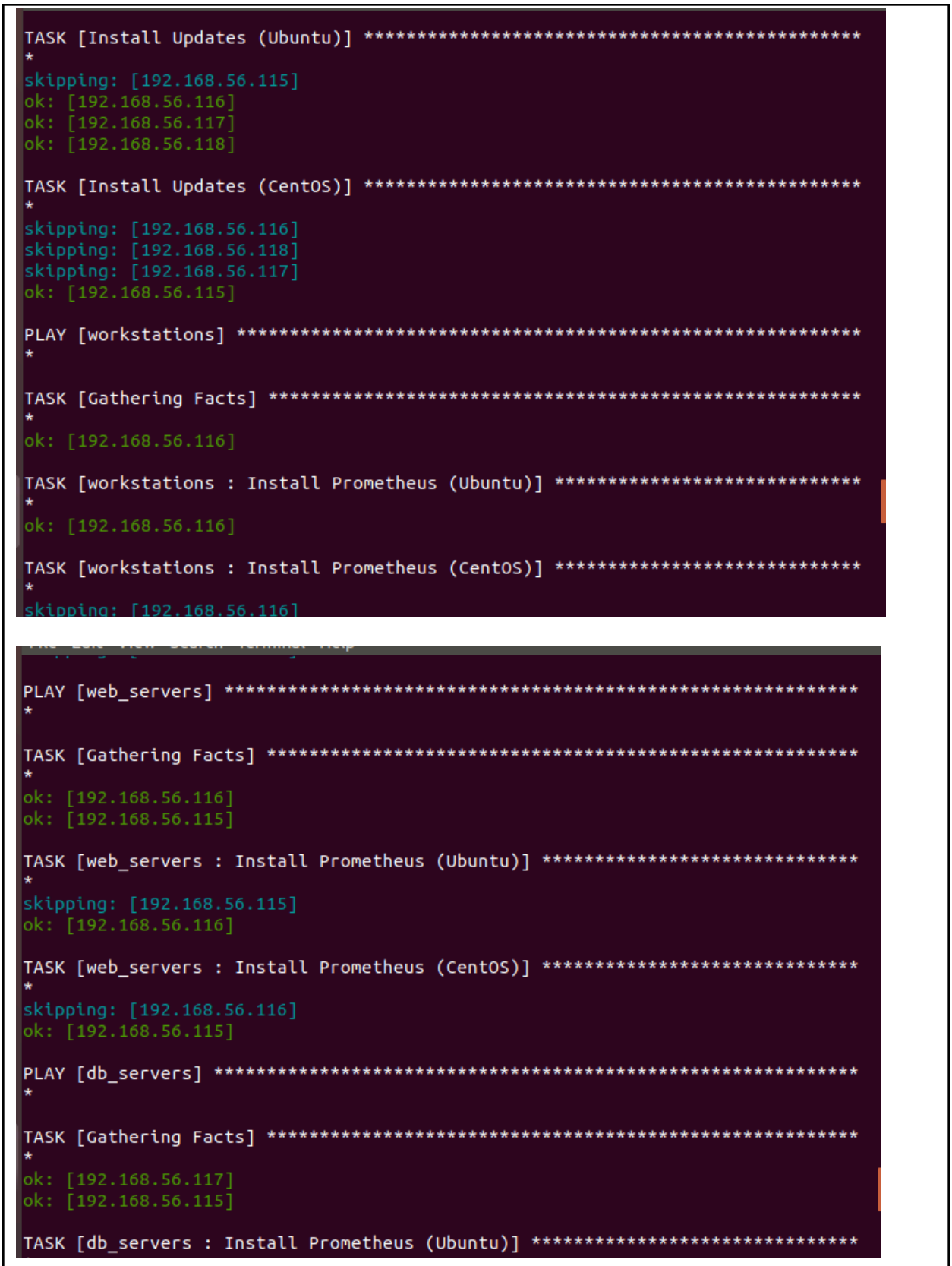
Inside the ~/HOA9/roles/file_servers/tasks create a main.yml.



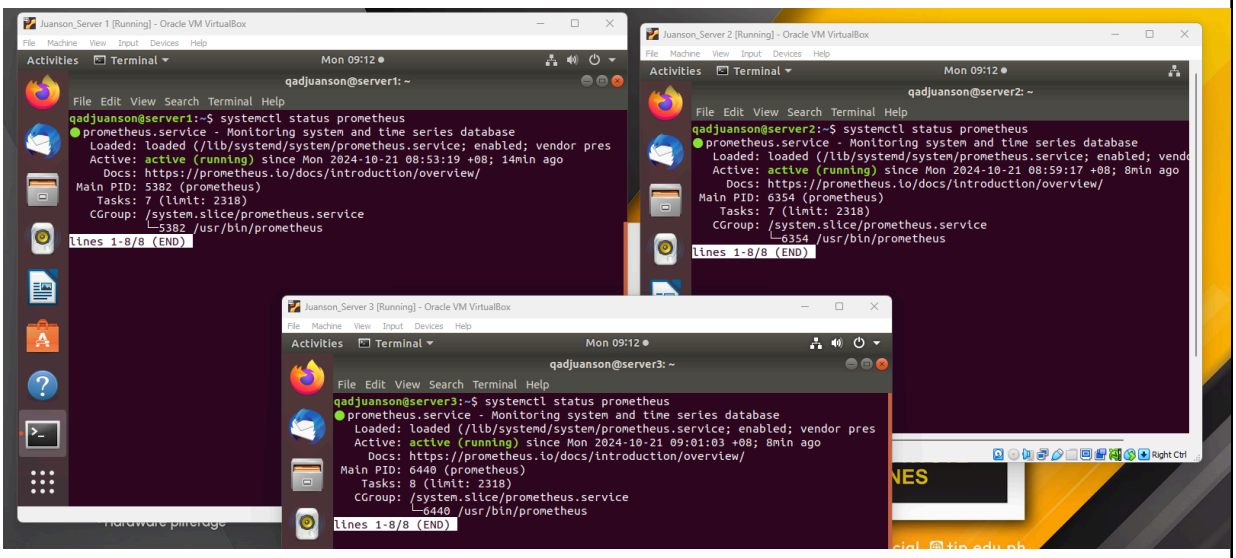
Inside the ~/HOA9/roles/workstations/tasks create a main.yml.

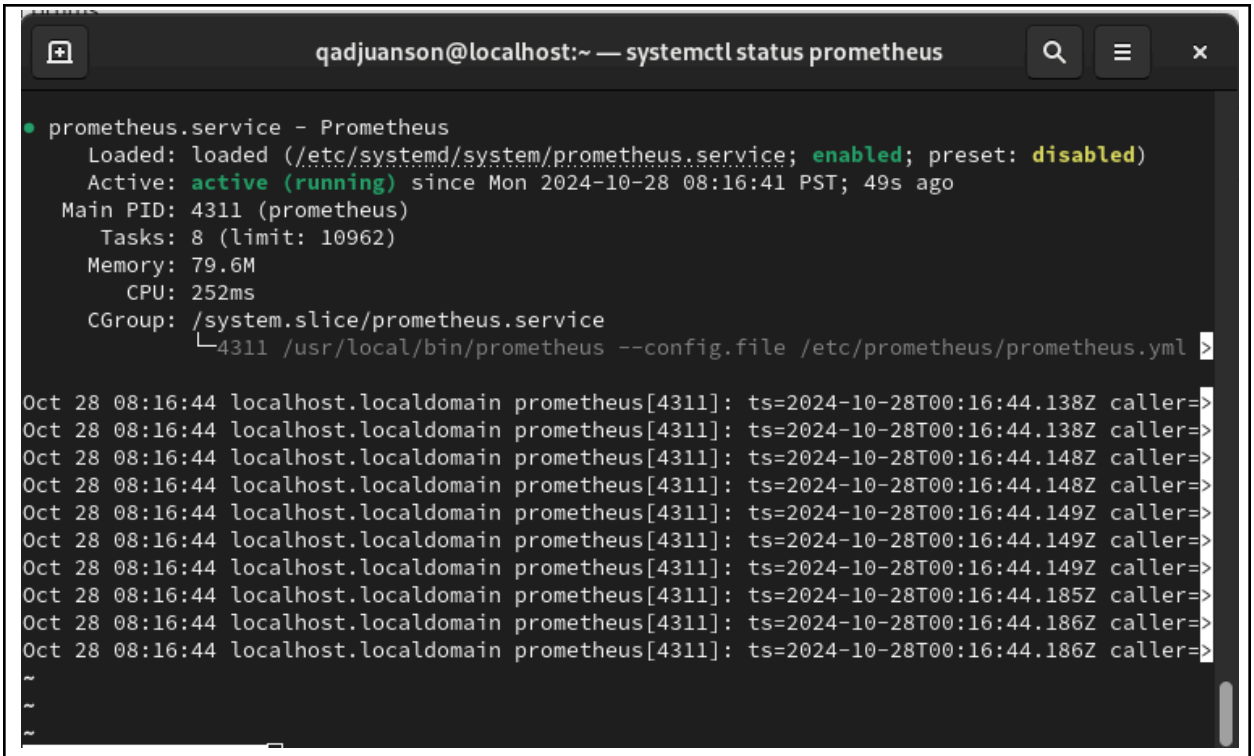


Run the playbook. *ansible-playbook install.yml –ask-become-pass*


TASK [Install Updates (Ubuntu)] ***********
*
skipping: [192.168.56.115]
ok: [192.168.56.116]
ok: [192.168.56.117]
ok: [192.168.56.118]
TASK [Install Updates (CentOS)] ************
*
skipping: [192.168.56.116]
skipping: [192.168.56.118]
skipping: [192.168.56.117]
ok: [192.168.56.115]
PLAY [workstations] ************************
*
TASK [Gathering Facts] *********************
*
ok: [192.168.56.116]
TASK [workstations : Install Prometheus (Ubuntu)] ***********
*
ok: [192.168.56.116]
TASK [workstations : Install Prometheus (CentOS)] ***********
*
skipping: [192.168.56.116]
PLAY [web_servers] *************************
*
TASK [Gathering Facts] *********************
*
ok: [192.168.56.116]
ok: [192.168.56.115]
TASK [web_servers : Install Prometheus (Ubuntu)] ************
*
skipping: [192.168.56.115]
ok: [192.168.56.116]
TASK [web_servers : Install Prometheus (CentOS)] ************
*
skipping: [192.168.56.116]
ok: [192.168.56.115]
PLAY [db_servers] **************************
*
TASK [Gathering Facts] *********************
*
ok: [192.168.56.117]
ok: [192.168.56.115]
TASK [db_servers : Install Prometheus (Ubuntu)] *************

```
skipping: [192.168.56.117]
ok: [192.168.56.115]

PLAY [file_servers] ************************************************************
*

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.118]

TASK [file_servers : Install Prometheus (Ubuntu)] ******************************
*
ok: [192.168.56.118]

TASK [file_servers : Install Prometheus (CentOS)] ******************************
*
skipping: [192.168.56.118]

PLAY RECAP *********************************************************************
*
192.168.56.115             : ok=6    changed=0    unreachable=0    failed=0
skipped=3    rescued=0    ignored=0
192.168.56.116             : ok=6    changed=0    unreachable=0    failed=0
skipped=3    rescued=0    ignored=0
192.168.56.117             : ok=4    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.118             : ok=4    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
```





Reflections:

Answer the following:

1. What are the benefits of having a performance monitoring tool?
   - A performance monitoring tool helps keep systems running smoothly by constantly checking for issues and spotting potential problems early. By catching problems quickly, these tools help reduce downtime, making systems more reliable and ensuring users have a good experience.

Conclusions:

- In this activity, we are tasked to install a performance monitoring tool (prometheus) in our Ubuntu and CentOs. While doing this activity, I've encountered issues in my playbook and by enabling the prometheus in my CentOS. But despite everything I am able to install prometheus in my Ubuntu and CentOS.