Name: Pacinos, Angela Monique A.	**Date Performed:** 10-23-23
Course/Section: CPE232 - CPE31S4	**Date Submitted:** 10-23-23
Instructor: Dr. Jonathan V. Taylar	**Semester and SY:** 1st Sem '23 - '24

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

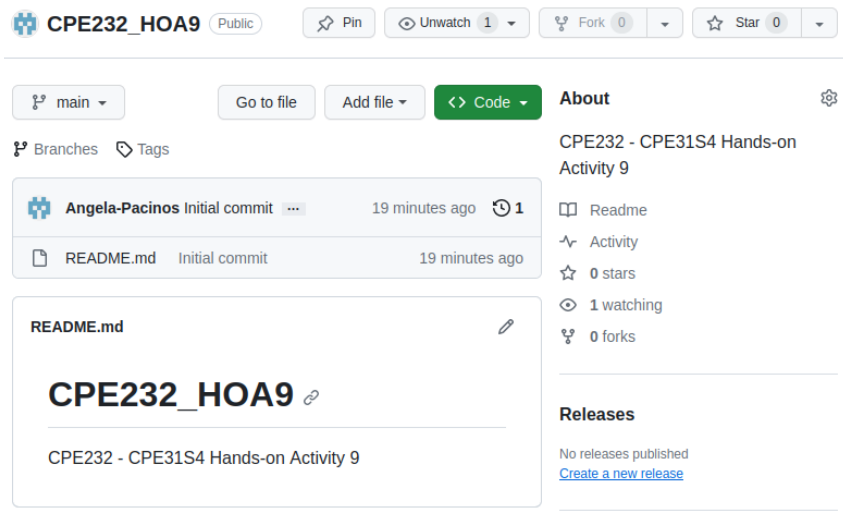
3. Tasks

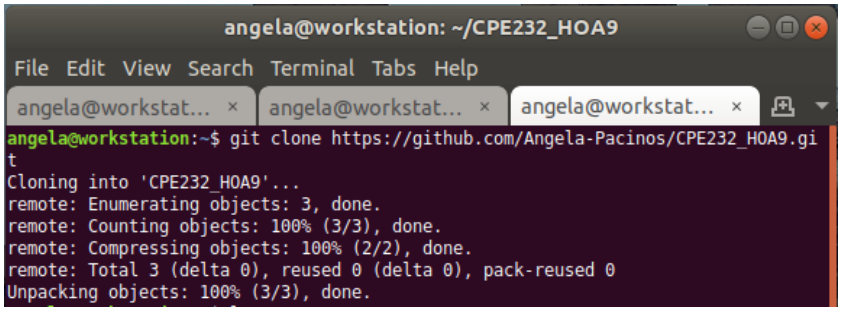
1. Create a playbook that installs Prometheus in both Ubuntu and CentOS. Apply the concept of creating roles.
2. Describe how you did step 1. (Provide screenshots and explanations in your report. Make your report detailed such that it will look like a manual.)
3. Show an output of the installed Prometheus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

4. Output

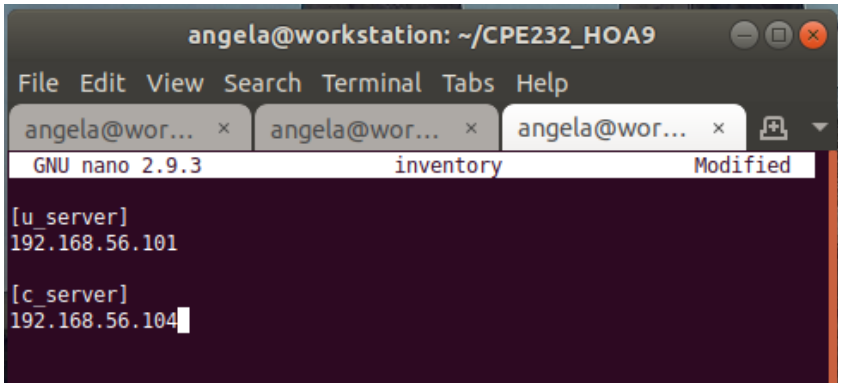
INPUT

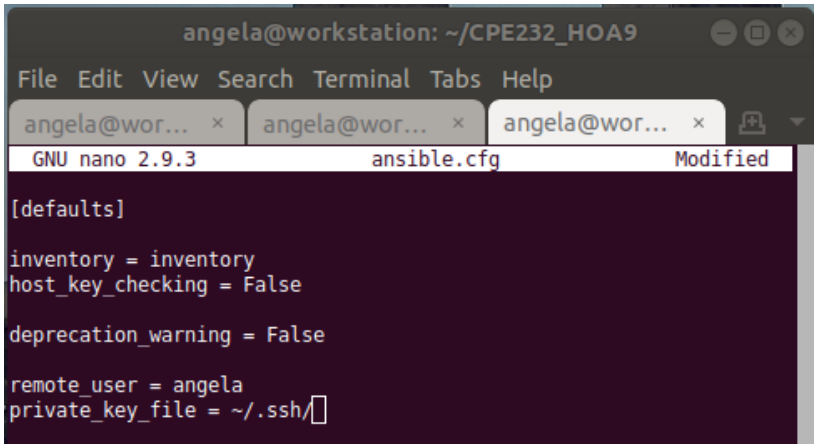
- **Create a new repository and configure it with the needed files.**

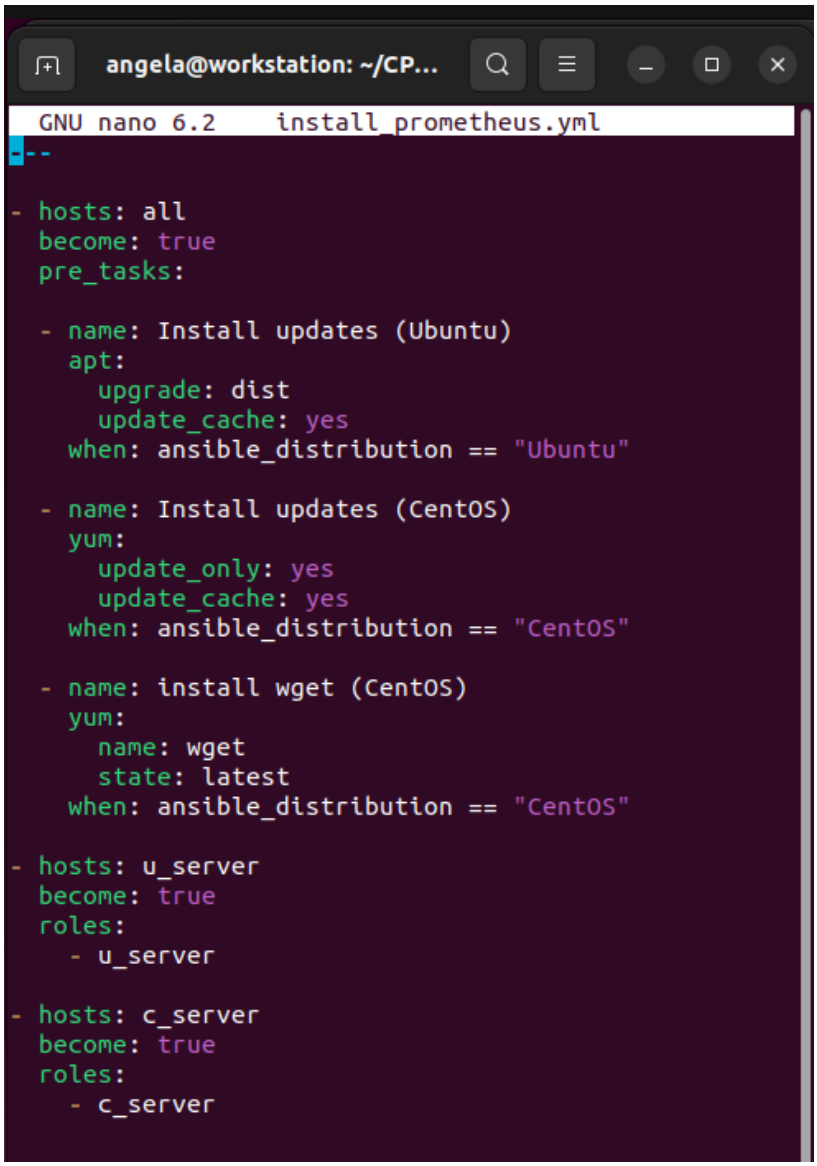




Create the inventory, ansible.cfg, and your yml playbook then configure as follows.

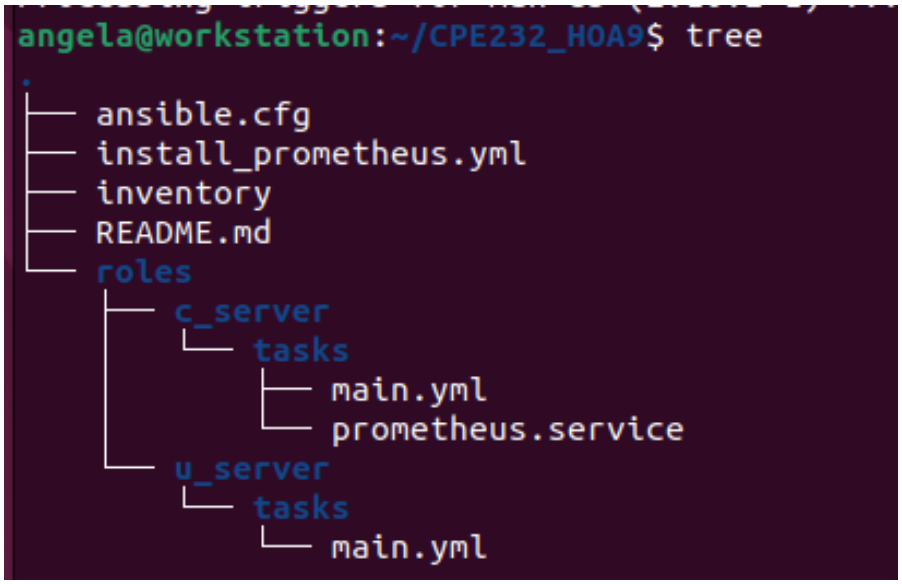



angela@workstation: ~/CPE232_HOA9
File Edit View Search Terminal Tabs Help
angela@wor...
angela@wor...
angela@wor...
GNU nano 2.9.3 ansible.cfg Modified
[defaults]
inventory = inventory
host_key_checking = False
deprecation_warning = False
remote_user = angela
private_key_file = ~/.ssh/

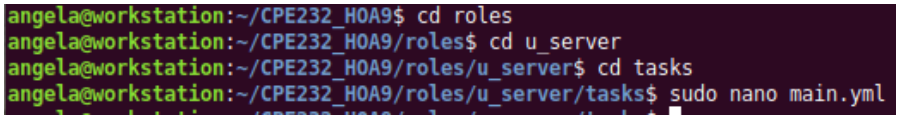

angela@workstation: ~/CP...
GNU nano 6.2 install_prometheus.yml

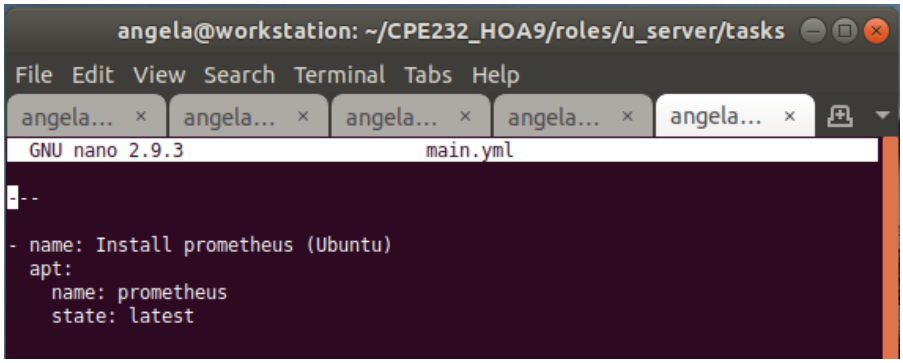
- hosts: all
become: true
pre_tasks:
- name: Install updates (Ubuntu)
apt:
upgrade: dist
update_cache: yes
when: ansible_distribution == "Ubuntu"
- name: Install updates (CentOS)
yum:
update_only: yes
update_cache: yes
when: ansible_distribution == "CentOS"
- name: install wget (CentOS)
yum:
name: wget
state: latest
when: ansible_distribution == "CentOS"
- hosts: u_server
become: true
roles:
- u_server
- hosts: c_server
become: true
roles:
- c_server

Under the same directory, create a new directory and name it roles. Enter the roles directory and create new directories: Ubuntu and CentOS. For each directory, create a directory and name it tasks.



- **Install Premotheus package**
  Edit the main.yml for both Ubuntu and CentOS directory as follows. Save and exit





```
angela@workstation:~/CPE232_HOA9$ cd roles
angela@workstation:~/CPE232_HOA9/roles$ cd c_server
angela@workstation:~/CPE232_HOA9/roles/c_server$ cd tasks
angela@workstation:~/CPE232_HOA9/roles/c_server/tasks$ sudo nano main.yml
[sudo] password for angela:
```

angela@workstation: ~/CPE232_HOA9/roles/c_server/tasks

GNU nano 6.2 main.yml

```
---

- name: Creating a directory (where the downloaded files will be stored)
  tags: directory
  file:
    path: ~/prometheus
    state: directory

- name: Downloading and extracting Prometheus
  tags: source
  unarchive:
    src: https://github.com/prometheus/prometheus/releases/download/v2.8.1/prometheus-2.8.1.linux-amd64.tar.gz
    dest: ~/prometheus
    remote_src: yes
    mode: 0777
    owner: root
    group: root

- name: Adding the Prometheus executables to a PATH
  tags: executables
  shell: |
    cd ~/prometheus/prometheus*
    cp -r . /usr/local/bin/prometheus

- name: Copying the Prometheus service file
  tags: servicefile
  copy:
    src: prometheus.service
    dest: /etc/systemd/system/
    owner: root
    group: root
    mode: 777

- name: Making sure that Prometheus is started and enabled
  tags: serviceon
  service:
    name: prometheus
    state: restarted
    enabled: true
```

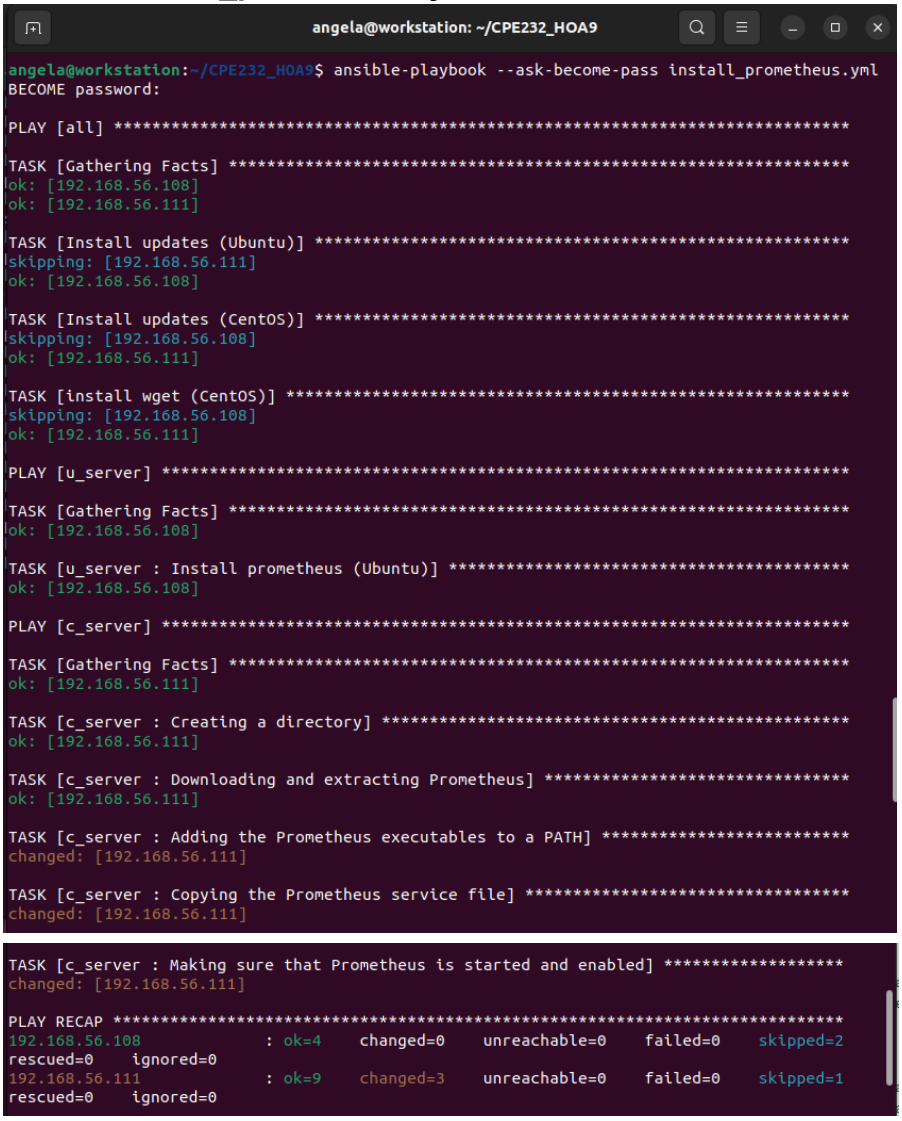
- **Make sure that the repository is sync in the Github**

angela@workstation: ~/CPE232_HOA9

```
angela@workstation:~/CPE232_HOA9$ git add *
angela@workstation:~/CPE232_HOA9$ git commit -m "HOA9"
[main 7a5a9f6] HOA9
 5 files changed, 69 insertions(+), 72 deletions(-)
 delete mode 100644 roles/c_server/prometheus.service,j2
 rewrite roles/c_server/tasks/main.yml (89%)
 create mode 100644 roles/c_server/tasks/prometheus.service
angela@workstation:~/CPE232_HOA9$ git push origin main
Username for 'https://github.com': Angela-Pacinos
Password for 'https://Angela-Pacinos@github.com':
Enumerating objects: 16, done.
Counting objects: 100% (16/16), done.
Delta compression using up to 2 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (9/9), 1.35 KiB | 231.00 KiB/s, done.
Total 9 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Angela-Pacinos/CPE232_HOA9.git
   1bd9400..7a5a9f6  main -> main
```

PROCESS

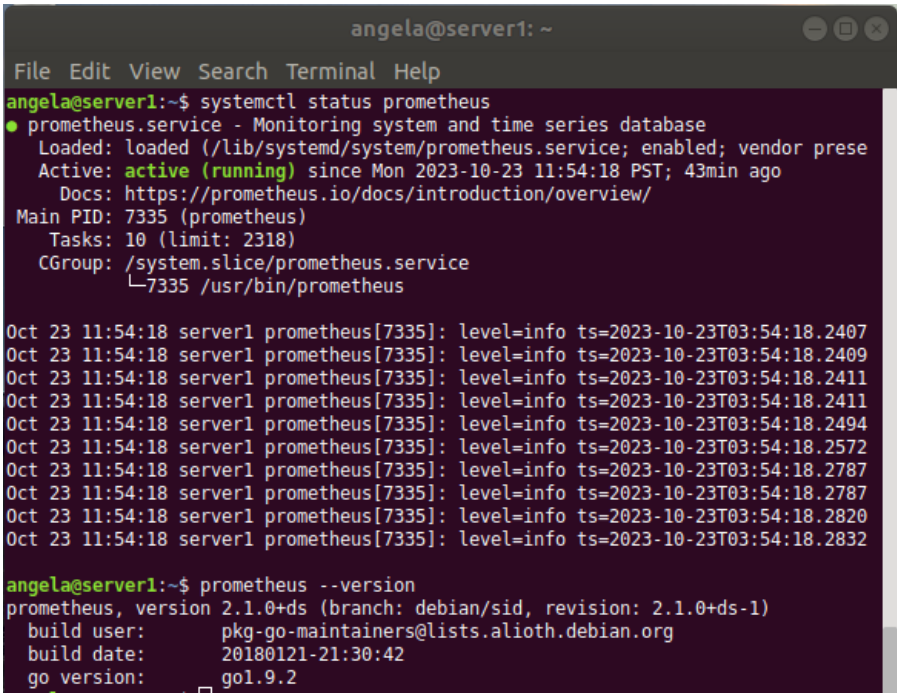
- **Run the install_prometheus.yml**

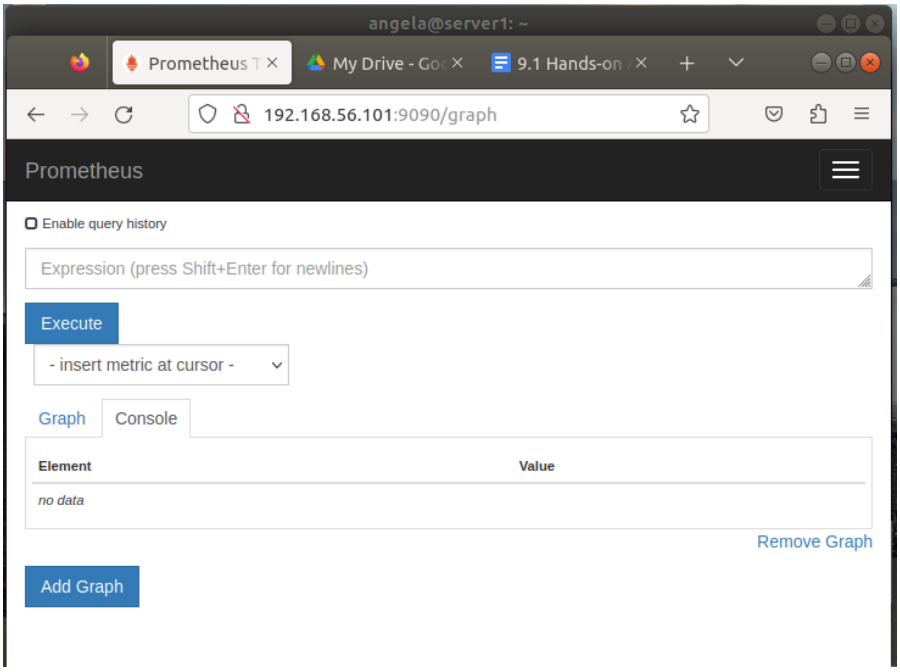


OUTPUT

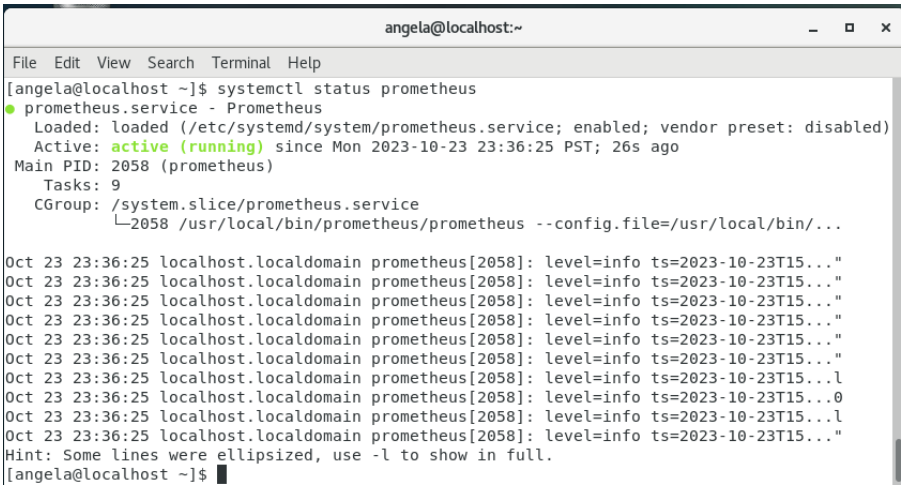
- **Check to see if the prometheus was successfully installed.**

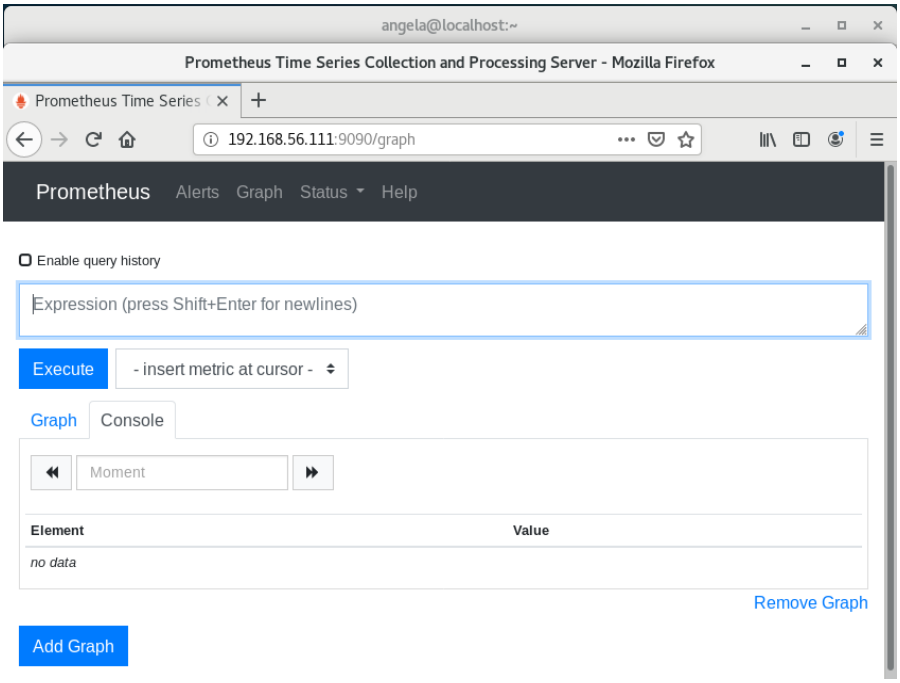
  Ubuntu





CentOS


angela@localhost:~
File Edit View Search Terminal Help
[angela@localhost ~]$ systemctl status prometheus
● prometheus.service - Prometheus
Loaded: loaded (/etc/systemd/system/prometheus.service; enabled; vendor preset: disabled)
Active: active (running) since Mon 2023-10-23 23:36:25 PST; 26s ago
Main PID: 2058 (prometheus)
Tasks: 9
CGroup: /system.slice/prometheus.service
└─2058 /usr/local/bin/prometheus/prometheus --config.file=/usr/local/bin/...
Oct 23 23:36:25 localhost.localdomain prometheus[2058]: level=info ts=2023-10-23T15..."
Oct 23 23:36:25 localhost.localdomain prometheus[2058]: level=info ts=2023-10-23T15..."
Oct 23 23:36:25 localhost.localdomain prometheus[2058]: level=info ts=2023-10-23T15..."
Oct 23 23:36:25 localhost.localdomain prometheus[2058]: level=info ts=2023-10-23T15..."
Oct 23 23:36:25 localhost.localdomain prometheus[2058]: level=info ts=2023-10-23T15..."
Oct 23 23:36:25 localhost.localdomain prometheus[2058]: level=info ts=2023-10-23T15..."
Oct 23 23:36:25 localhost.localdomain prometheus[2058]: level=info ts=2023-10-23T15...l
Oct 23 23:36:25 localhost.localdomain prometheus[2058]: level=info ts=2023-10-23T15...0
Oct 23 23:36:25 localhost.localdomain prometheus[2058]: level=info ts=2023-10-23T15...l
Oct 23 23:36:25 localhost.localdomain prometheus[2058]: level=info ts=2023-10-23T15..."
Hint: Some lines were ellipsized, use -l to show in full.
[angela@localhost ~]$


angela@localhost:~
Prometheus Time Series Collection and Processing Server - Mozilla Firefox
Prometheus Time Series
192.168.56.111:9090/graph
Prometheus Alerts Graph Status Help
Enable query history
Expression (press Shift+Enter for newlines)
Execute
- insert metric at cursor -
Graph Console
Moment
Element Value
no data
Remove Graph
Add Graph

Reflections:

1. **What are the benefits of having a performance monitoring tool?**
   Performance monitoring tools have many benefits especially in various industries. These powerful tools help to monitor and optimize the performance of a system. They can see the real-time happenings in the systems and if ever they detect issues, they can easily fix and troubleshoot it. Minimizing the downtime of the system which allows for more efficiency. They can also strengthen the security of the system as they can also detect malicious logins or unauthorized users that are trying to access sensitive data.

Conclusions:

This activity introduces us with a Performance monitoring tool which is a power tool that is used in many industries. We were tasked to install the Prometheus which is a kind of performance monitoring tool. We have to implement the roles and install it in both Ubuntu and CentOS. Creating the playbook for its installation in Ubuntu was pretty easy but with installing it on CentOS, I did lots of modifying the playbook since there are errors. I tried installing it manually in CentOS but there is no package for that, which means that the source must be from somewhere. I search for sources where I can install and find the package for the Prometheus and from there I modified the playbook so it will run successfully. Overall, this activity challenges me again to create and install a package from scratch and with the help of researching I was able to install Prometheus in both Ubuntu and CentOS.