Name: Jefferson Langbid	**Date Performed:**
Course/Section: CPE232-CPE31S23	**Date Submitted:**
Instructor: Dr. Taylar	**Semester and SY:**

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

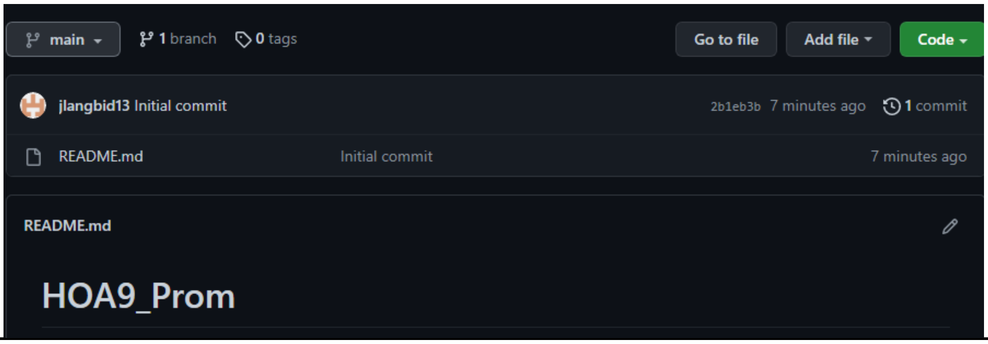
Cacti

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)



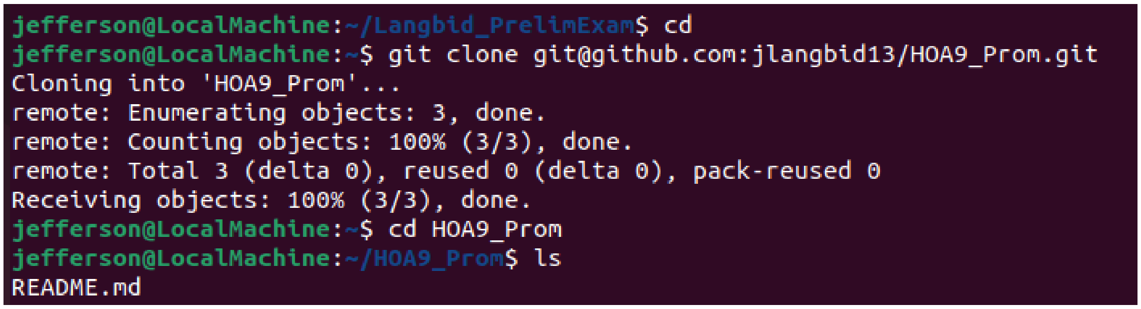
jlangbid13 Initial commit 2b1eb3b 7 minutes ago 1 commit

README.md Initial commit 7 minutes ago

README.md

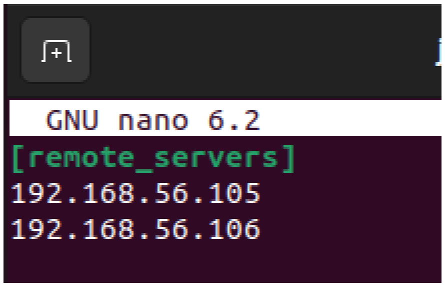
HOA9_Prom

Create new repository in the github



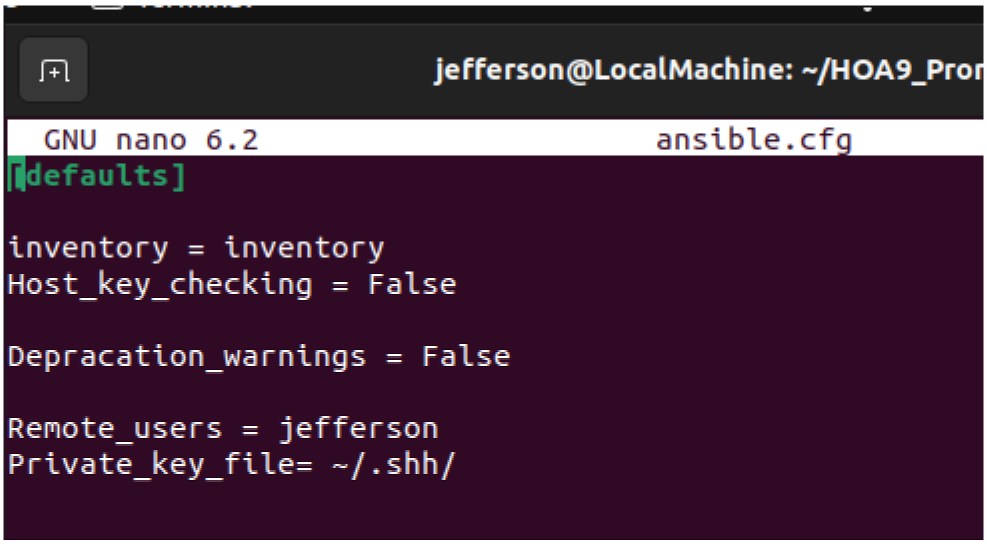
```
jefferson@LocalMachine:~/Langbid_PrelimExam$ cd
jefferson@LocalMachine:~$ git clone git@github.com:jlangbid13/HOA9_Prom.git
Cloning into 'HOA9_Prom'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
jefferson@LocalMachine:~$ cd HOA9_Prom
jefferson@LocalMachine:~/HOA9_Prom$ ls
README.md
```

Clone the github repository in the local machine.



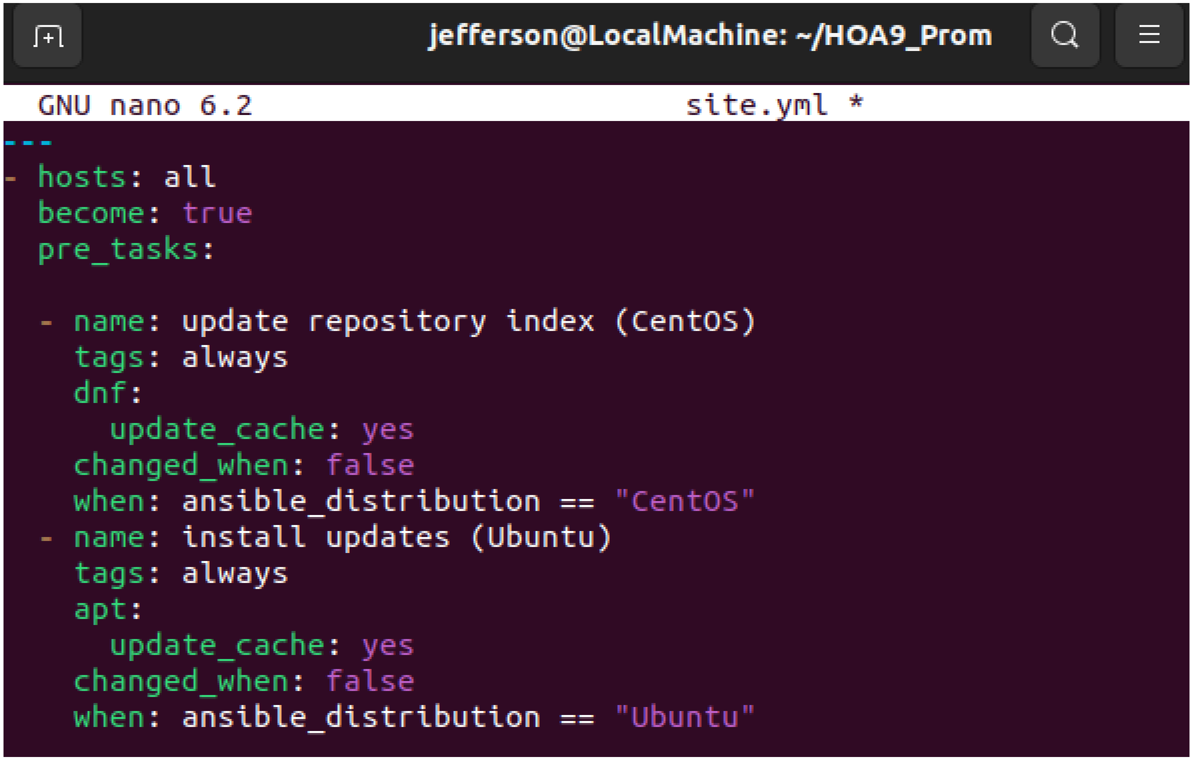
```
  GNU nano 6.2
[remote_servers]
192.168.56.105
192.168.56.106
```

Create the inventory file



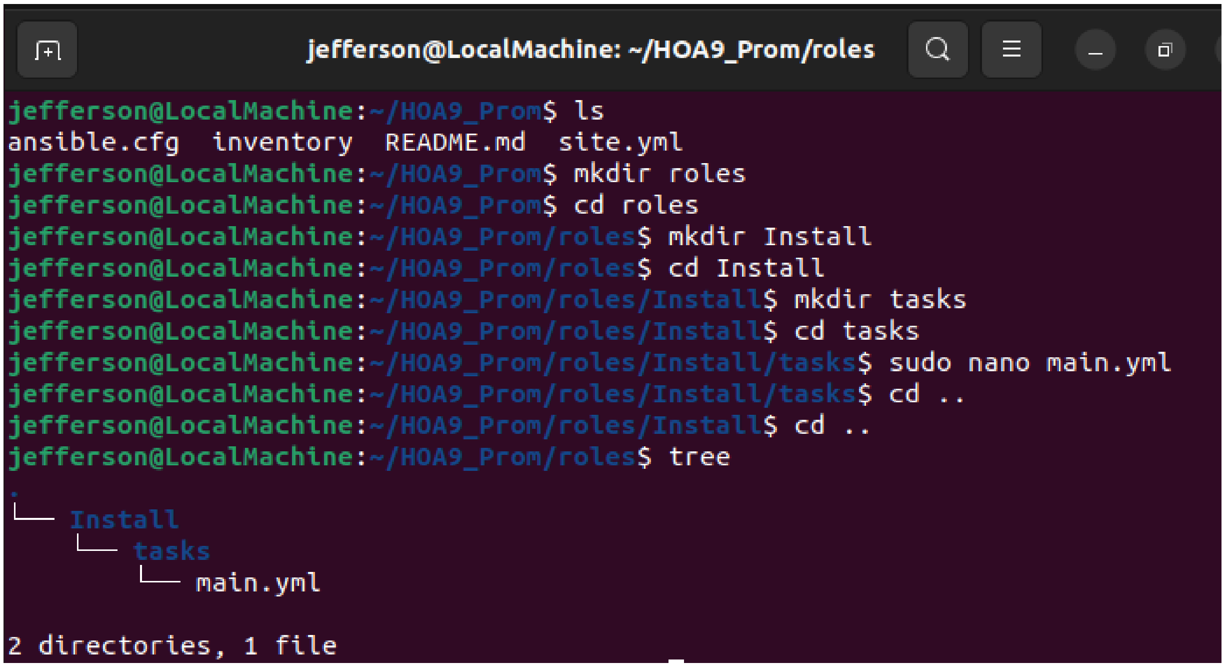
```
  GNU nano 6.2                          ansible.cfg
[defaults]

inventory = inventory
Host_key_checking = False

Depracation_warnings = False

Remote_users = jefferson
Private_key_file= ~/.shh/
```

Create the ansible.cfg file to configure the remote users.



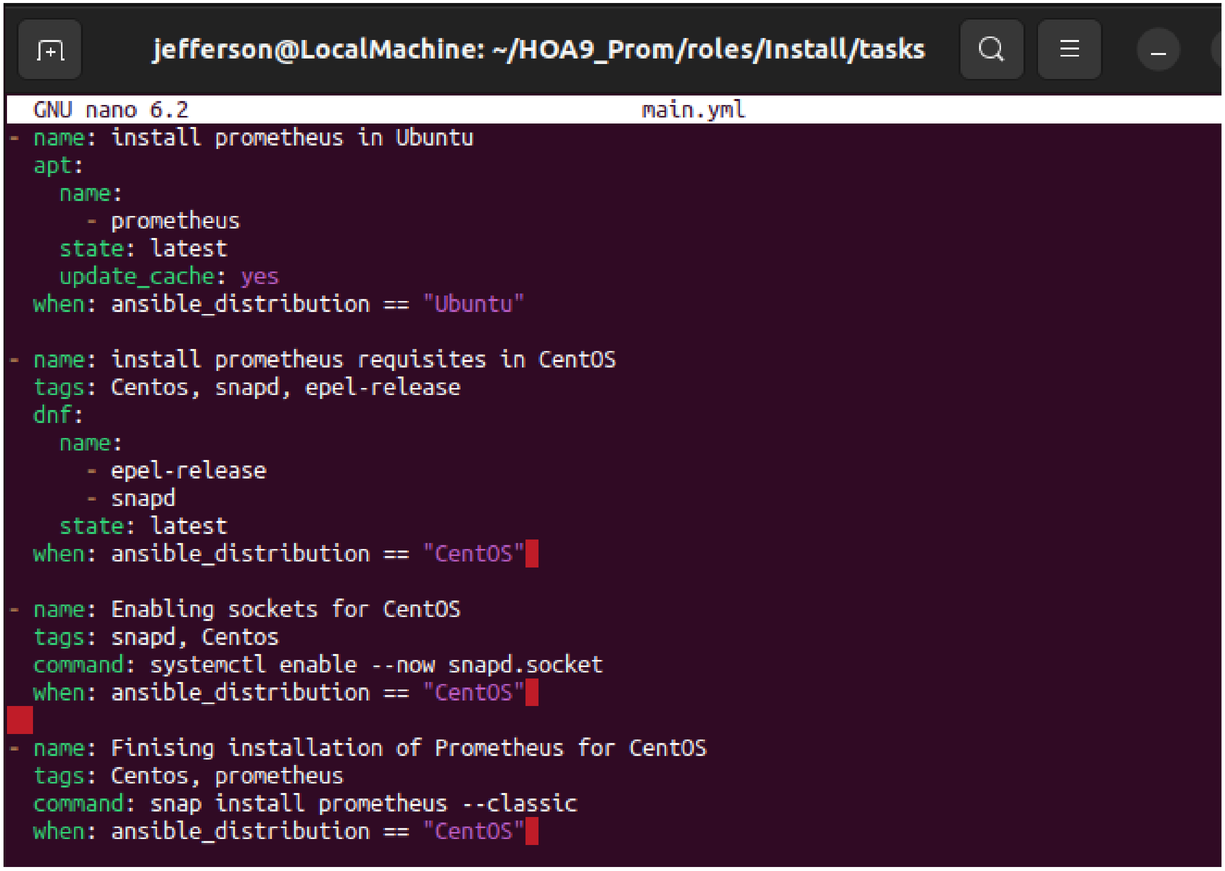
```
  GNU nano 6.2                              site.yml *
---
- hosts: all
  become: true
  pre_tasks:

  - name: update repository index (CentOS)
    tags: always
    dnf:
      update_cache: yes
    changed_when: false
    when: ansible_distribution == "CentOS"
  - name: install updates (Ubuntu)
    tags: always
    apt:
      update_cache: yes
    changed_when: false
    when: ansible_distribution == "Ubuntu"
```

Create the site.yml for the pretask and later on the code to run the roles.



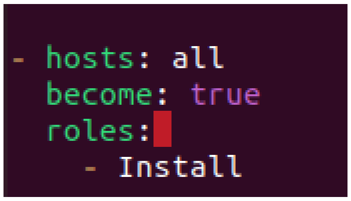
```
jefferson@LocalMachine:~/HOA9_Prom$ ls
ansible.cfg  inventory  README.md  site.yml
jefferson@LocalMachine:~/HOA9_Prom$ mkdir roles
jefferson@LocalMachine:~/HOA9_Prom$ cd roles
jefferson@LocalMachine:~/HOA9_Prom/roles$ mkdir Install
jefferson@LocalMachine:~/HOA9_Prom/roles$ cd Install
jefferson@LocalMachine:~/HOA9_Prom/roles/Install$ mkdir tasks
jefferson@LocalMachine:~/HOA9_Prom/roles/Install$ cd tasks
jefferson@LocalMachine:~/HOA9_Prom/roles/Install/tasks$ sudo nano main.yml
jefferson@LocalMachine:~/HOA9_Prom/roles/Install/tasks$ cd ..
jefferson@LocalMachine:~/HOA9_Prom/roles/Install$ cd ..
jefferson@LocalMachine:~/HOA9_Prom/roles$ tree
.
└── Install
    └── tasks
        └── main.yml

2 directories, 1 file
```

Create a new directory for the roles and a new directory which is Install and inside it will be the tasks where I will put the main.yml file.



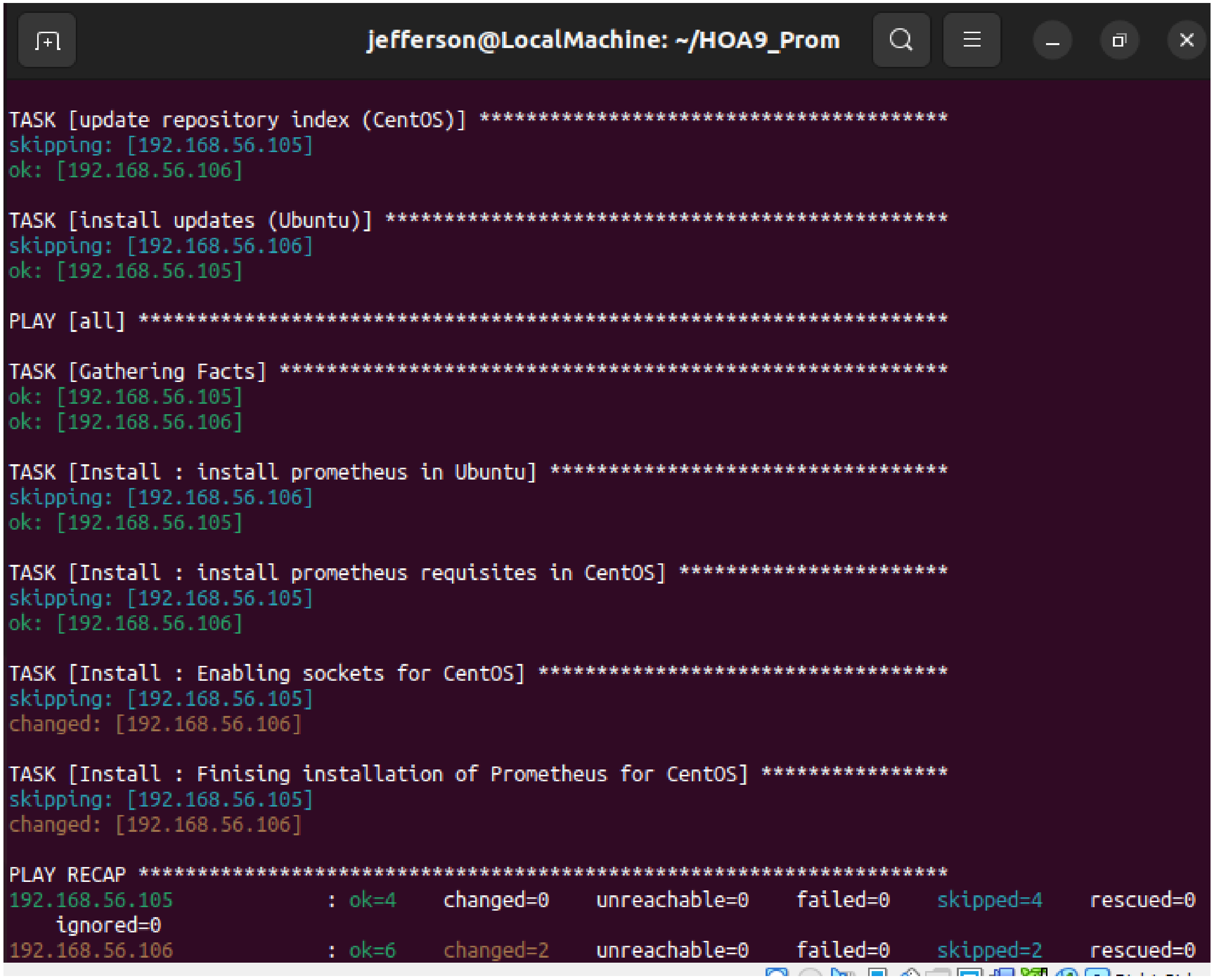
```
  GNU nano 6.2                                      main.yml
- name: install prometheus in Ubuntu
  apt:
    name:
      - prometheus
    state: latest
    update_cache: yes
  when: ansible_distribution == "Ubuntu"

- name: install prometheus requisites in CentOS
  tags: Centos, snapd, epel-release
  dnf:
    name:
      - epel-release
      - snapd
    state: latest
  when: ansible_distribution == "CentOS"

- name: Enabling sockets for CentOS
  tags: snapd, Centos
  command: systemctl enable --now snapd.socket
  when: ansible_distribution == "CentOS"

- name: Finising installation of Prometheus for CentOS
  tags: Centos, prometheus
  command: snap install prometheus --classic
  when: ansible_distribution == "CentOS"
```

Input in the main.yml the command that will install Prometheus in both Ubuntu and CentOS.



```
- hosts: all
  become: true
  roles:
    - Install
```

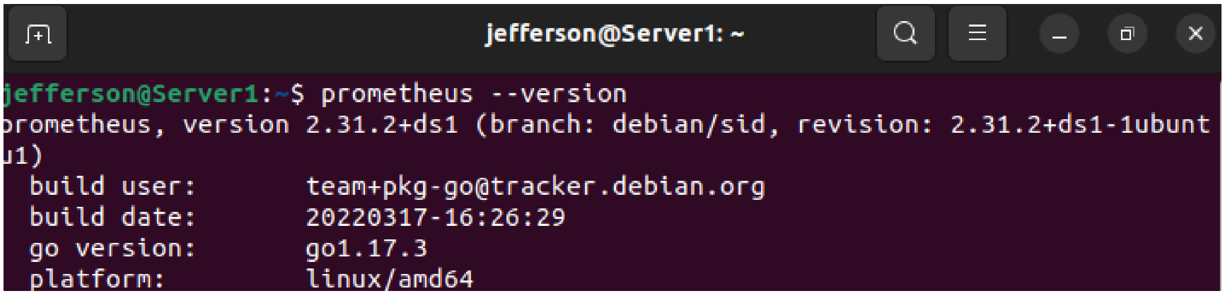
The command inside the site.yml to run the roles.



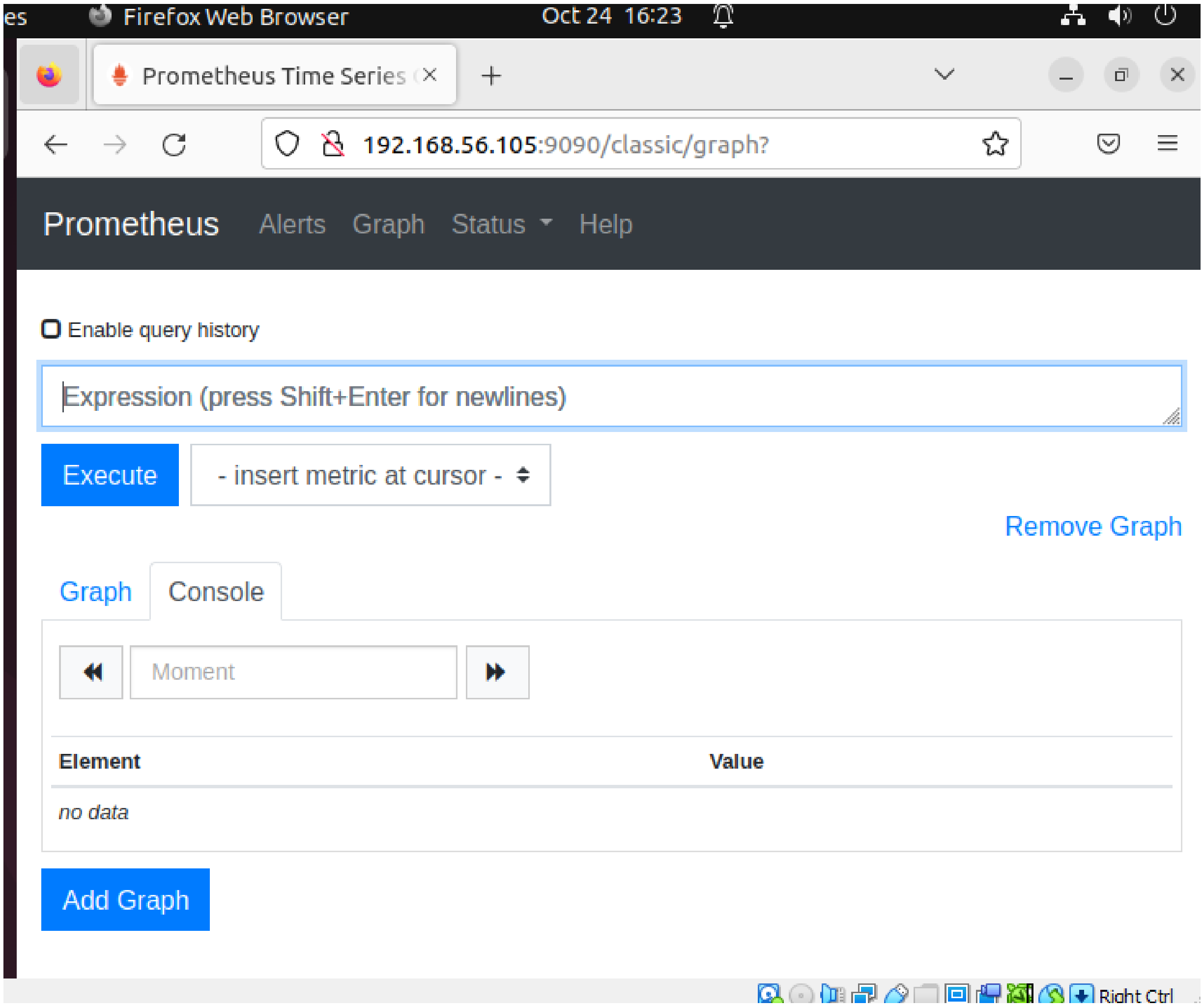
```
TASK [update repository index (CentOS)] ****************************************
skipping: [192.168.56.105]
ok: [192.168.56.106]

TASK [install updates (Ubuntu)] ************************************************
skipping: [192.168.56.106]
ok: [192.168.56.105]

PLAY [all] *********************************************************************

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.105]
ok: [192.168.56.106]

TASK [Install : install prometheus in Ubuntu] **********************************
skipping: [192.168.56.106]
ok: [192.168.56.105]

TASK [Install : install prometheus requisites in CentOS] ***********************
skipping: [192.168.56.105]
ok: [192.168.56.106]

TASK [Install : Enabling sockets for CentOS] ***********************************
skipping: [192.168.56.105]
changed: [192.168.56.106]

TASK [Install : Finising installation of Prometheus for CentOS] ****************
skipping: [192.168.56.105]
changed: [192.168.56.106]

PLAY RECAP *********************************************************************
192.168.56.105             : ok=4    changed=0    unreachable=0    failed=0    skipped=4    rescued=0
    ignored=0
192.168.56.106             : ok=6    changed=2    unreachable=0    failed=0    skipped=2    rescued=0
```

The code ran and it successfully downloaded in both Ubuntu and CentOS.

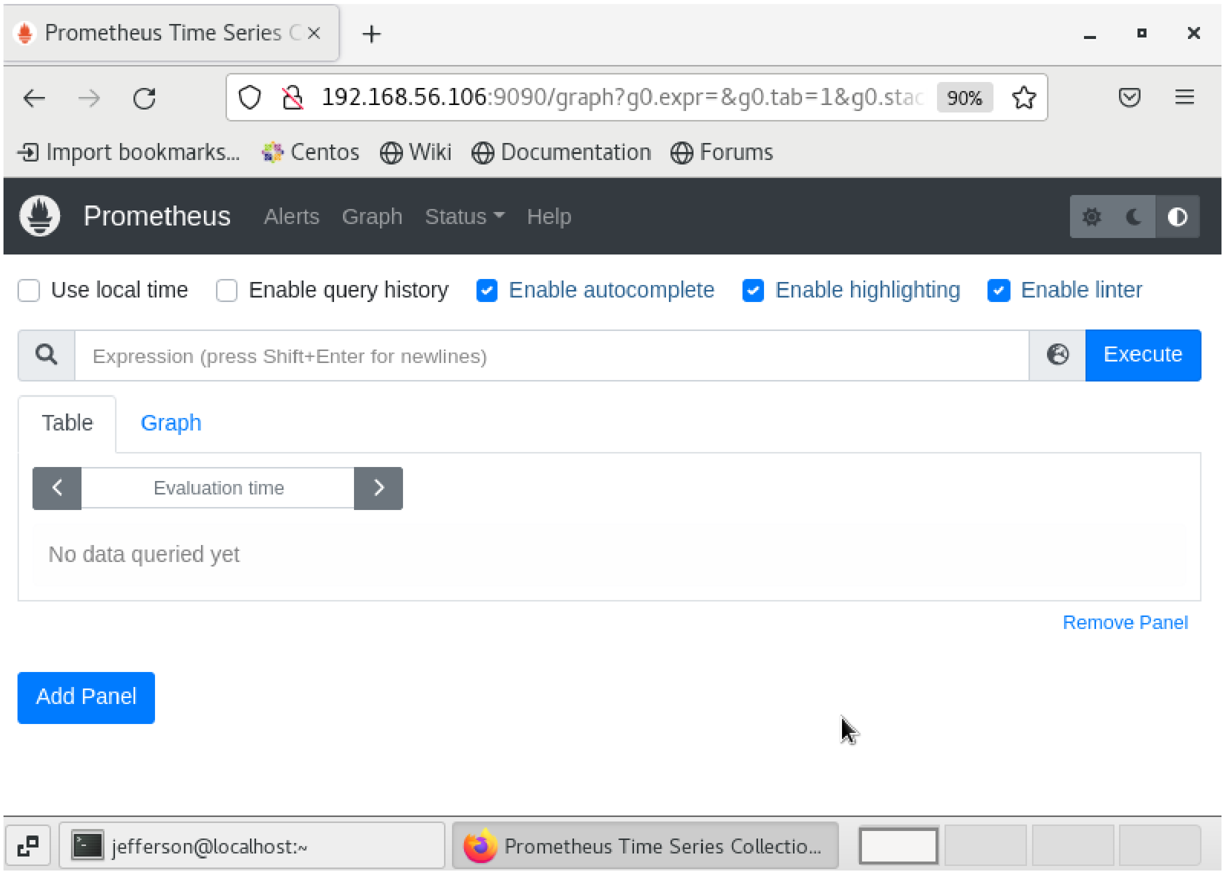
Successfully installed in Ubuntu



```
jefferson@Server1:~$ prometheus --version
prometheus, version 2.31.2+ds1 (branch: debian/sid, revision: 2.31.2+ds1-1ubunt
u1)
  build user:       team+pkg-go@tracker.debian.org
  build date:       20220317-16:26:29
  go version:       go1.17.3
  platform:         linux/amd64
```



Successfully installed in CentOS



192.168.56.106:9090/graph?g0.expr=&g0.tab=1&g0.stac

☐ Use local time ☐ Enable query history ☑ Enable autocomplete ☑ Enable highlighting ☑ Enable linter

Expression (press Shift+Enter for newlines)

No data queried yet

Reflections:

Answer the following:

1. What are the benefits of having a performance monitoring tool?
   The benefit of an uptime monitoring tool is that you can monitor what everyone is doing and see if it's good or right, resulting in better work results. Monitoring staff can guide people to improve performance and achieve better work results.

Conclusions:

In conclusion , I created a new directory for the activity and cloned it in my local machine. I created the file which is the ansible.cfg and inventory to configure the remote user to connect with the ansible. After creating the file I created a new file for the ansible playbook which is the site.yml and input the commands. After creating the file I created the roles directory and tasks to input the main.yml file for the command that will install the prometheus and its prerequisites in both Ubuntu and CentOs. After that I ran the site.yml playbook to run and it successfully ran and installed the prometheus in both Ubuntu and CentOS and able to open the prometheus site