Name: Renier L. Lope	Date Performed: 10/16/2024
Course/Section: CPE212 - CPE31S2	Date Submitted: 10/21/2024
Instructor: Engr. Robin Valenzuela	Semester and SY: 1st Sem (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)

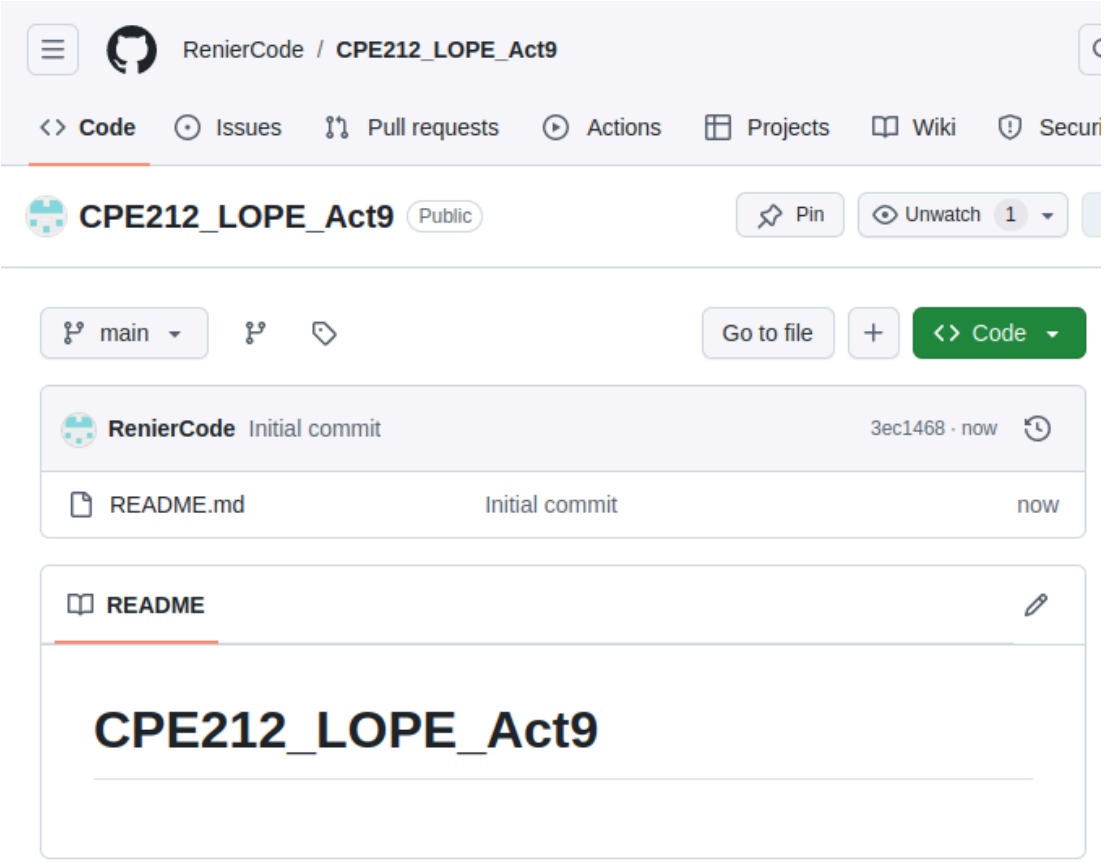


Figure 9.1: Create a new repository in github for this activity.

```
rnrlope@workstation:~$ git clone git@github.com:RenierCode/CPE212_LOPE_Act9
Cloning into 'CPE212_LOPE_Act9'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
rnrlope@workstation:~$ cd CPE*9
rnrlope@workstation:~/CPE212_LOPE_Act9$
```

Figure 9.2: Clone the new repository to the local machine.

```
rnrlope@workstation:~$ cp CPE*8/ansible.cfg CPE*9/
rnrlope@workstation:~$ cp CPE*8/inventory CPE*9/
rnrlope@workstation:~$ cd CPE*9
rnrlope@workstation:~/CPE212_LOPE_Act9$ ls
ansible.cfg  inventory  README.md
rnrlope@workstation:~/CPE212_LOPE_Act9$
```

Figure 9.3: Copy the ansible.cfg and inventory of the previous activity to the new repository.

```
rnrlope@workstation:~/CPE212_LOPE_Act9$ cat ansible.cfg
[defaults]
inventory = inventory
remote_user = rnrlope
host_key_checking = True
deprecation_warnings = False
rnrlope@workstation:~/CPE212_LOPE_Act9$ cat inventory
[web_servers]
server1

[db_servers]
centOS
```

Figure 9.4: Contents of the ansible.cfg and inventory files.

```
rnrlope@workstation:~/CPE212_LOPE_Act9$ nano prometheus.yml
rnrlope@workstation:~/CPE212_LOPE_Act9$ cat prometheus.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"

- hosts: all
  become: true
  roles:
    -  base
- hosts: db_servers:web_servers
  become: true
  roles:
    - install_prometheus

- hosts: db_servers:web_servers
  become: true
  roles:
    - config_prometheus

- hosts: db_servers:web_servers
  become: true
  roles:
    - start_prometheus
rnrlope@workstation:~/CPE212_LOPE_Act9$
```

Figure 9.5: Create a playbook named “prometheus.yml”. This playbook will play the task inside the desired roles.

```
rnrlope@workstation:~/CPE212_LOPE_Act9$ mkdir roles
rnrlope@workstation:~/CPE212_LOPE_Act9$ cd roles
rnrlope@workstation:~/CPE212_LOPE_Act9/roles$ mkdir base install_prometheus
config_prometheus start_prometheus
rnrlope@workstation:~/CPE212_LOPE_Act9/roles$ ls
base  config_prometheus  install_prometheus  start_prometheus
rnrlope@workstation:~/CPE212_LOPE_Act9/roles$
```

Figure 9.6: Create a new directory named "roles", then inside "roles" create new directories named "base", "install_prometheus", "config_prometheus", "start_prometheus" .

This roles will contain tasks according to its assigned name.

```
rnrlope@workstation:~/CPE212_LOPE_Act9/roles$ cd base
rnrlope@workstation:~/CPE212_LOPE_Act9/roles/base$ mkdir tasks
rnrlope@workstation:~/CPE212_LOPE_Act9/roles/base$ cd ..
rnrlope@workstation:~/CPE212_LOPE_Act9/roles$ cd install_prometheus
rnrlope@workstation:~/CPE212_LOPE_Act9/roles/install_prometheus$ mkdir tasks
rnrlope@workstation:~/CPE212_LOPE_Act9/roles/install_prometheus$ cd ..
rnrlope@workstation:~/CPE212_LOPE_Act9/roles$ cd config_prometheus
rnrlope@workstation:~/CPE212_LOPE_Act9/roles/config_prometheus$ mkdir tasks
rnrlope@workstation:~/CPE212_LOPE_Act9/roles/config_prometheus$ cd ..
rnrlope@workstation:~/CPE212_LOPE_Act9/roles$ cd start_prometheus
rnrlope@workstation:~/CPE212_LOPE_Act9/roles/start_prometheus$ mkdir tasks
```

Figure 9.7: Create a directory named "tasks" inside the directories under roles. This directory "tasks" will contain the playbooks assigned for each roles.

```
rnrlope@workstation:~/CPE212_LOPE_Act9/roles$ cd base/tasks
rnrlope@workstation:~/CPE212_LOPE_Act9/roles/base/tasks$ nano main.yml
rnrlope@workstation:~/CPE212_LOPE_Act9/roles/base/tasks$ cat main.yml
---
- name: install updates (CentOS)
  tags: always
  dnf:
    update_only: yes
    update_cache: yes
  when: ansible_distribution == "CentOS"

- name: install updates (Ubuntu)
  tags: always
  apt:
    upgrade: dist
    update_cache: yes
  when: ansible_distribution == "Ubuntu"
rnrlope@workstation:~/CPE212_LOPE_Act9/roles/base/tasks$
```

Figure 9.8: Create a playbook file named "main.yml" inside "roles/base/tasks". This playbook will update both CentOS and Ubuntu.

```
rnrlope@workstation:~/CPE212_LOPE_Act9/roles/install_prometheus/tasks$ nano main.yml
rnrlope@workstation:~/CPE212_LOPE_Act9/roles/install_prometheus/tasks$ cat main.yml
---
- name: Open port 9090 (Ubuntu)
  ufw:
    rule: allow
    port: 9090
    proto: tcp
    state: enabled
  when: ansible_distribution == "Ubuntu"

- name: Allow Prometheus for Firewall (CentOS)
  firewalld:
    port: 9090/tcp
    permanent: yes
    state: enabled
  when: ansible_distribution == "CentOS"

- name: Install Prometheus (Ubuntu)
  apt:
    name: prometheus
    state: latest
  when: ansible_distribution == "Ubuntu"

- name: Install Prometheus (CentOS)
  unarchive:
    src: https://github.com/prometheus/prometheus/releases/download/v2.30.0/promethe
us-2.30.0.linux-amd64.tar.gz
    dest: /usr/local/bin
    remote_src: yes
    mode: 0755
    owner: root
    group: root
  when: ansible_distribution == "CentOS"

- name: install apache2 (Ubuntu)
  apt:
    name: apache2
    state: latest
  when: ansible_distribution == "Ubuntu"

- name: install apache (CentOS)
  dnf:
    name: httpd
    state: latest
  when: ansible_distribution == "CentOS"
rnrlope@workstation:~/CPE212_LOPE_Act9/roles/install_prometheus/tasks$
```

Figure 9.9: Create a playbook file named “main.yml” inside “roles/install_prometheus/tasks”. This playbook will allow prometheus for firewall and install it on both CentOS and Ubuntu and also install apache on both CentOS and Ubuntu.

```
rnrlope@workstation:~/CPE212_LOPE_Act9/roles$ cd config_prometheus/tasks
rnrlope@workstation:~/CPE212_LOPE_Act9/roles/config_prometheus/tasks$ nano main.yml
rnrlope@workstation:~/CPE212_LOPE_Act9/roles/config_prometheus/tasks$ cat main.yml
---
- name: Copy Prometheus binaries
  copy:
    src: /usr/local/bin/prometheus-2.30.0.linux-amd64/prometheus
    dest: /usr/local/bin/prometheus
    mode: 0755
    remote_src: yes
  when: ansible_distribution == "CentOS"

- name: Copy Promtool binaries
  copy:
    src: /usr/local/bin/prometheus-2.30.0.linux-amd64/prometheus
    dest: /usr/local/bin/promtool
    mode: 0755
    remote_src: yes
  when: ansible_distribution == "CentOS"

- name: Create Prometheus directories
  file:
    path: "{{ item }}"
    state: directory
  loop:
    - /etc/prometheus
    - /var/lib/prometheus
  when: ansible_distribution == "CentOS"

- name: Copy prometheus.yml to /etc/prometheus
  command: cp /usr/local/bin/prometheus-2.30.0.linux-amd64/prometheus.yml /etc/prome
theus
  when: ansible_distribution == "CentOS"

- name: Copy consoles directory to /etc/prometheus
  command: cp -r /usr/local/bin/prometheus-2.30.0.linux-amd64/consoles /etc/promethe
us
  when: ansible_distribution == "CentOS"

- name: Copy console_libraries directory to /etc/prometheus
  command: cp -r /usr/local/bin/prometheus-2.30.0.linux-amd64/console_libraries /etc
/prometheus
  when: ansible_distribution == "CentOS"

- name: Create prometheus.service file
  copy:
    dest: /etc/systemd/system/prometheus.service
    content: |
      [Unit]
      Description=Prometheus
      Wants=network-online.target
      After=network-online.target

      [Service]
      User=root
      Group=root
      Type=simple
      ExecStart=/usr/local/bin/prometheus \
              --config.file /etc/prometheus/prometheus.yml \
              --storage.tsdb.path /var/lib/prometheus \
              --web.console.templates=/etc/prometheus/consoles \
              --web.console.libraries=/etc/prometheus/console_libraries \

      [Install]
      WantedBy=multi-user.target

- name: Reload systemd
  command: systemctl daemon-reload
  when: ansible_distribution == "CentOS"
rnrlope@workstation:~/CPE212_LOPE_Act9/roles/config_prometheus/tasks$
```

Figure 9.10: Create a playbook file named “main.yml” inside “roles/config_prometheus/tasks”. This playbook will configure prometheus on both CentOS and Ubuntu.

```
rnrlope@workstation:~/CPE212_LOPE_Act9/roles/start_prometheus/tasks$ nano main.yml
rnrlope@workstation:~/CPE212_LOPE_Act9/roles/start_prometheus/tasks$ cat main.yml
---
- name: Start apache2 (Ubuntu)
  systemd:
     name: apache2
     enabled: yes
     state: started
  when: ansible_distribution == "Ubuntu"

- name: Start apache2 (CentOS)
  systemd:
     name: httpd
     enabled: yes
     state: started
  when: ansible_distribution == "CentOS"

- name: Start Prometheus Service
  systemd:
     name: prometheus
     enabled: yes
     state: started

rnrlope@workstation:~/CPE212_LOPE_Act9/roles/start_prometheus/tasks$
```

Figure 9.11: Create a playbook file named “main.yml” inside “roles/start_prometheus/tasks”. This playbook will start apache and prometheus on both CentOS and Ubuntu.

```
TASK [config_prometheus : Copy console_libraries directory to /etc/prometheus] *****
skipping: [server1]
changed: [centOS]

TASK [config_prometheus : Create prometheus.service file] **************************
changed: [server1]
changed: [centOS]

TASK [config_prometheus : Reload systemd] ******************************************
skipping: [server1]
changed: [centOS]

PLAY [db_servers:web_servers] ******************************************************

TASK [Gathering Facts] *************************************************************
ok: [server1]
ok: [centOS]

TASK [start_prometheus : Start Prometheus Service] *********************************
ok: [server1]
changed: [centOS]

PLAY RECAP *************************************************************************
centOS                     : ok=18   changed=11   unreachable=0    failed=0    skipp
ed=4    rescued=0    ignored=0
server1                    : ok=11   changed=3    unreachable=0    failed=0    skipp
ed=11   rescued=0    ignored=0

rnrlope@workstation:~/CPE212_LOPE_Act9$
```

Figure 9.12: Play Recap of executing the playbook “promentheus.yml”.

```
rnrlope@server1:~$ which prometheus
/usr/bin/prometheus
rnrlope@server1:~$ prometheus --version
prometheus, version 2.45.3+ds (branch: debian/sid, revision: 2.45.3+ds-2ubuntu0.1)
  build user:       team+pkg-go@tracker.debian.org
  build date:       20240716-06:12:19
  go version:       go1.22.2
  platform:         linux/amd64
  tags:             unknown
rnrlope@server1:~$
```

```
rnrlope@server1:~$ systemctl status prometheus
● prometheus.service - Monitoring system and time series database
     Loaded: loaded (/lib/systemd/system/prometheus.service; enabled; vendor pres
     Active: active (running) since Mon 2024-10-21 07:44:47 +08; 27min ago
       Docs: https://prometheus.io/docs/introduction/overview/
   Main PID: 4758 (prometheus)
      Tasks: 10 (limit: 4541)
     CGroup: /system.slice/prometheus.service
             └─4758 /usr/bin/prometheus
lines 1-8/8 (END)
```

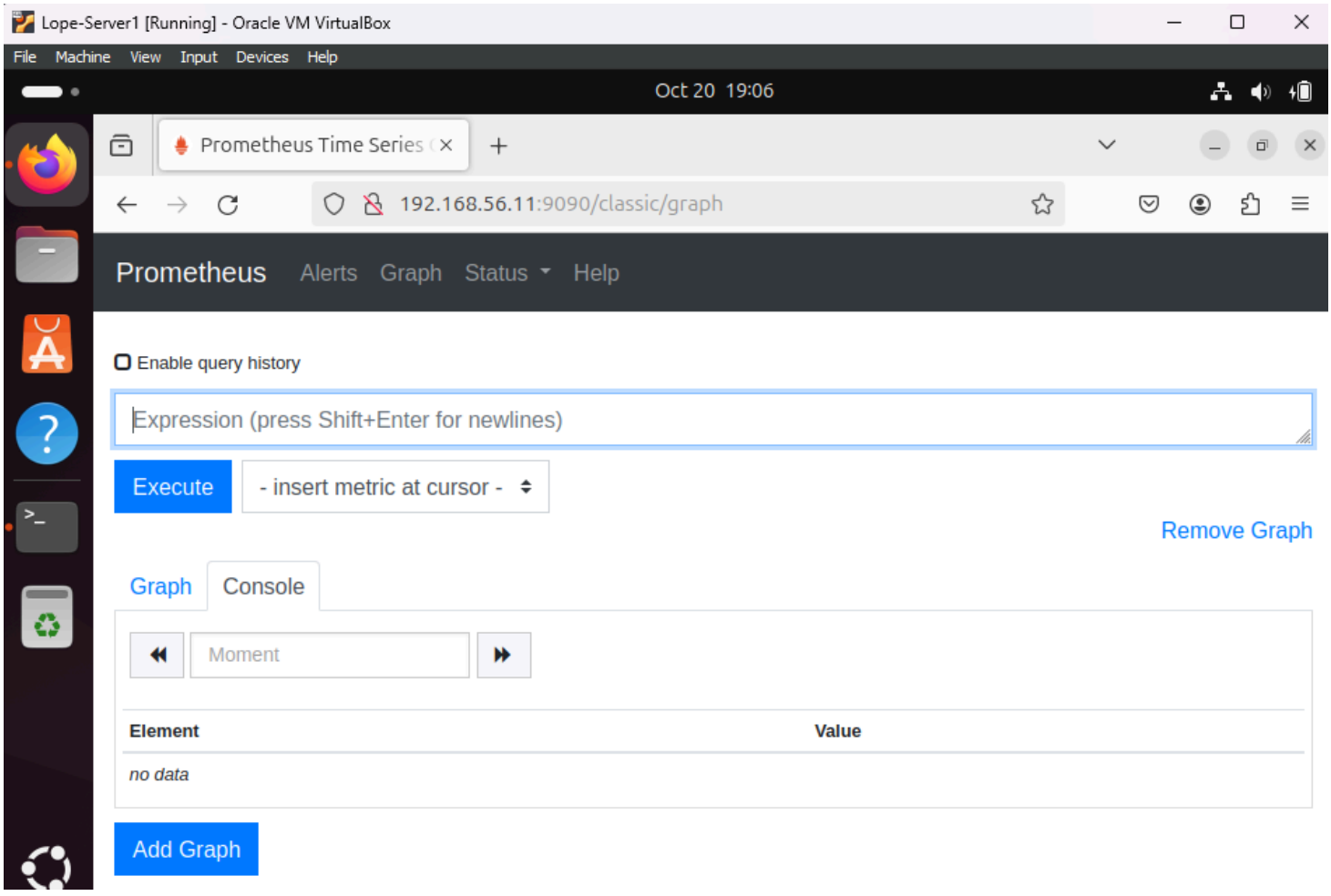


Figure 9.13 - 9.15: Verifying if prometheus is installed and running on server1.

```
[rnrlope@localhost ~]$ which prometheus
/usr/local/bin/prometheus
[rnrlope@localhost ~]$ prometheus --version
prometheus, version 2.30.0 (branch: HEAD, revision: 37468d88dce85ac507f3fb7864c7d1c0
78e3e27d)
  build user:       root@e913c65aa170
  build date:       20210914-09:49:24
  go version:       go1.17.1
  platform:         linux/amd64
[rnrlope@localhost ~]$
```

```
[rnrlope@localhost ~]$ systemctl status prometheus
● prometheus.service - Prometheus
   Loaded: loaded (/etc/systemd/system/prometheus.service; enabled; vendor preset:
 disabled)
   Active: active (running) since Sun 2024-10-20 19:45:36 EDT; 25min ago
 Main PID: 5639 (prometheus)
    Tasks: 7
   CGroup: /system.slice/prometheus.service
           └─5639 /usr/local/bin/prometheus --config.file /etc/prometheus/pro...
```

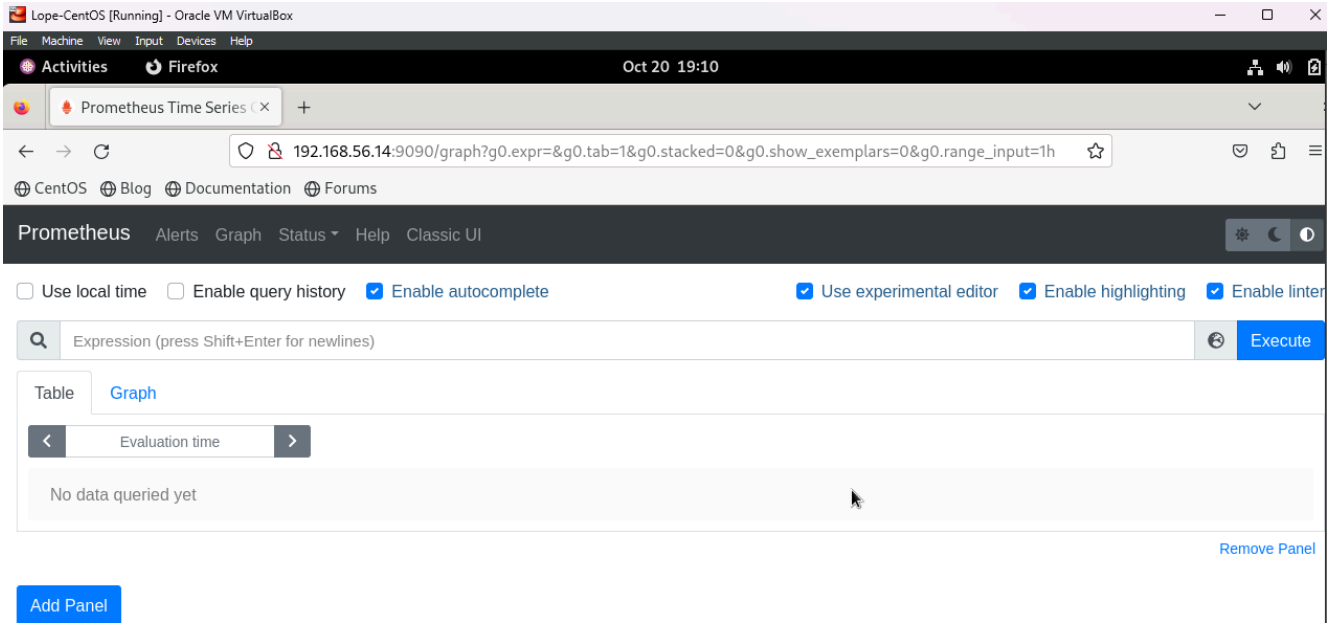


Figure 9.16 - 9.18: Verifying if prometheus is installed and running on centOS.

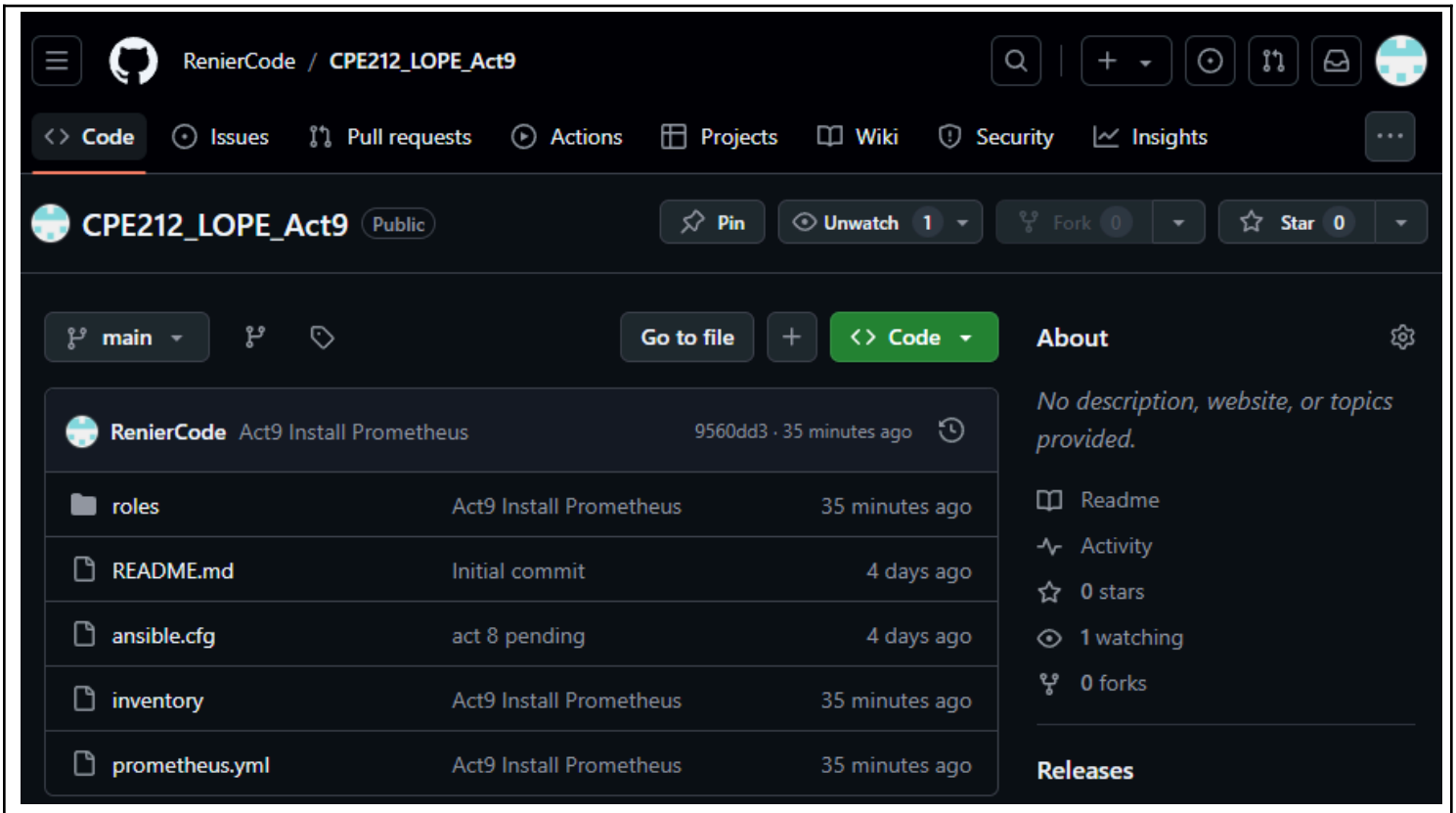
```
rnrlope@workstation:~/CPE212_LOPE_Act9$ tree
.
├── ansible.cfg
├── inventory
├── prometheus.yml
├── README.md
└── roles
    ├── base
    │   └── tasks
    │       └── main.yml
    ├── config_prometheus
    │   └── tasks
    │       └── main.yml
    ├── install_prometheus
    │   └── tasks
    │       └── main.yml
    └── start_prometheus
        └── tasks
            └── main.yml

10 directories, 8 files
rnrlope@workstation:~/CPE212_LOPE_Act9$
```

Figure 9.19: Contents of the Repository.

GIT PUSH:

```
rnrlope@workstation:~/CPE212_LOPE_Act9$ git add ansible.cfg
rnrlope@workstation:~/CPE212_LOPE_Act9$ git add inventory
rnrlope@workstation:~/CPE212_LOPE_Act9$ git add prometheus.yml
rnrlope@workstation:~/CPE212_LOPE_Act9$ git add roles
rnrlope@workstation:~/CPE212_LOPE_Act9$ git commit -m "Act9 Install Prometheus"
[main 9560dd3] Act9 Install Prometheus
 6 files changed, 124 insertions(+), 3 deletions(-)
 create mode 100644 roles/base/tasks/main.yml
 create mode 100644 roles/config_prometheus/tasks/main.yml
 create mode 100644 roles/install_prometheus/tasks/main.yml
 create mode 100644 roles/start_prometheus/tasks/main.yml
rnrlope@workstation:~/CPE212_LOPE_Act9$ git push origin main
Enumerating objects: 20, done.
Counting objects: 100% (20/20), done.
Delta compression using up to 2 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (17/17), 2.06 KiB | 264.00 KiB/s, done.
Total 17 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:RenierCode/CPE212_LOPE_Act9
   98e92e7..9560dd3  main -> main
```



GITHUB LINK:

https://github.com/RenierCode/CPE212_LOPE_Act9.git

Reflections:

Answer the following:

1. What are the benefits of having a performance monitoring tool?
   - Performance monitoring tool comes with several benefits, such as reducing outages, improving system reliability by detecting issues early, and offering a faster response time thus enhancing user experience. A performance monitoring tool also aids in troubleshooting by providing detailed data and root cause analysis, resulting in quicker resolutions. It also supports better optimization of resource usage and security by detecting anomalies.

Conclusions:

- **In this activity I manage to create and demonstrate a workflow that will install a Prometheus in both Ubuntu and CentOS server, while utilizing ansible-playbook and applying the concept of creating roles. Firstly, I created a main playbook named "prometheus.yml" inside the repository "CPE212_LOPE_Act9" that will run all the tasks inside the desired roles. Secondly, I created a new directory named "roles", then inside created new directories named "base", "install_prometheus", "config_prometheus", and "start_prometheus". Thirdly, inside of the new directories I created a new directory named "tasks". Lastly, I created playbooks inside of "tasks", separating the plays based on the names of various roles. Overall, I manage to design and create a workflow that installs a Prometheus in both Ubuntu and CentOS server, while utilizing ansible-playbook and applying the concept of creating roles, thereby increasing my knowledge about playbooks.**