Name: Clarence B. Chavez	**Date Performed:** October 21, 2024
Course/Section: CPE31S2	**Date Submitted:** October 23, 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)

Task No. 2

```
qcchavez@workstation:~/CPE212_Chavez_HOA9.1$ tree
.
├── ansible.cfg
├── install_prometheus.yml
├── inventory
├── README.md
└── roles
    ├── CentOS
    │   └── tasks
    │       └── main.yml
    └── Ubuntu
        └── tasks
            └── main.yml

6 directories, 6 files
qcchavez@workstation:~/CPE212_Chavez_HOA9.1$
```

1. This is the file content for my repository in this Hands-on Activity 9.1

```
qcchavez@workstation:~/CPE212_Chavez_HOA9.1$ cat ansible.cfg
[defaults]
inventory = inventory
remote_user = qcchavez
host_key_checking = True
```

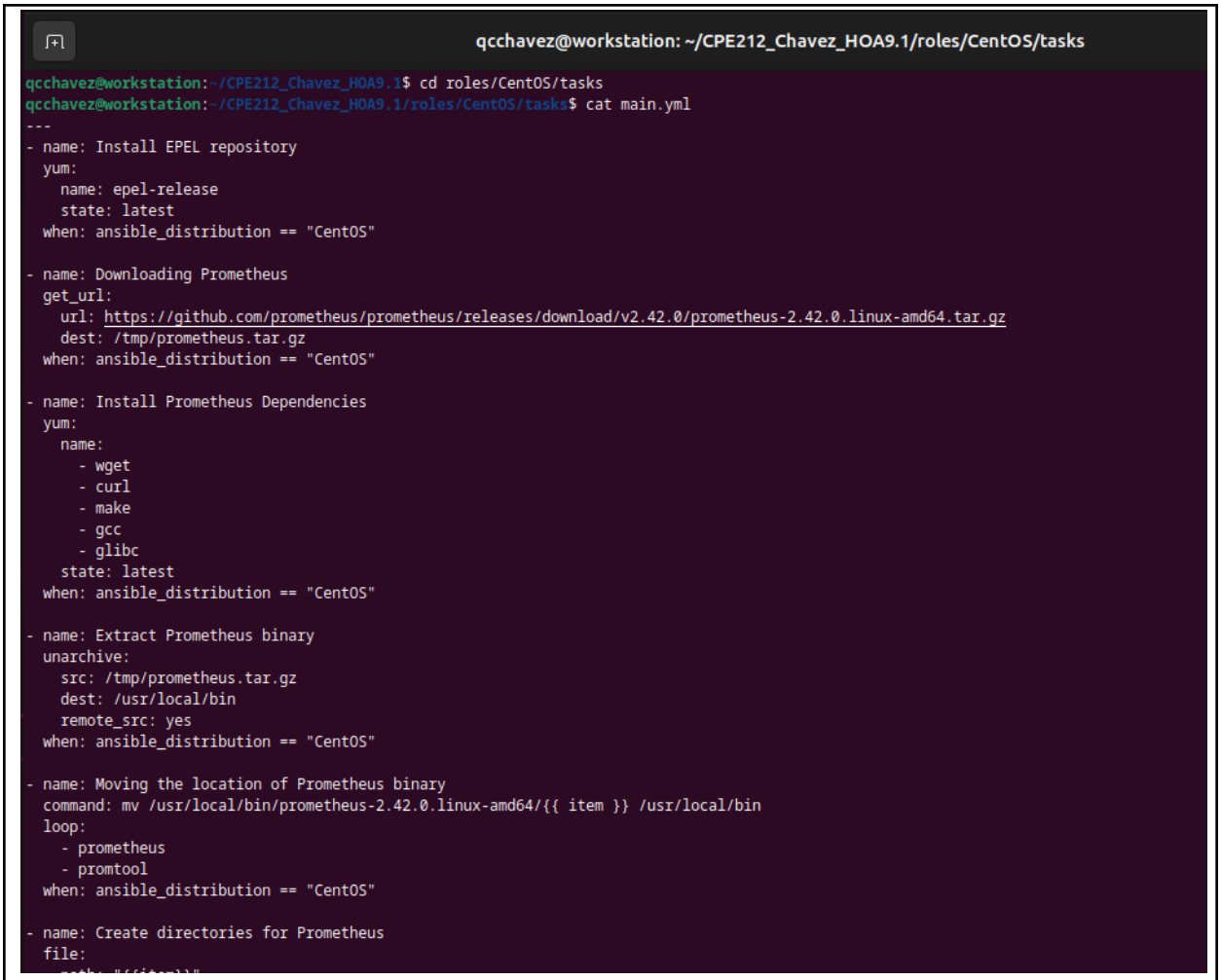
2. This is the content of the **ansible.cfg**, which means that this is the configuration file for the ansible.

```
qcchavez@workstation:~/CPE212_Chavez_HOA9.1$ cat install_prometheus.yml
---
- hosts: all
  become: true
  pre_tasks:

  - name: update repository index / install Updates (CentOS)
    tags: always
    dnf:
      update_cache: yes
    changed_when: false
    when: ansible_distribution == "CentOS"

  - name: update repository index / install Updates (Ubuntu)
    tags: always
    apt:
      update_cache: yes
    changed_when: false
    when: ansible_distribution == "Ubuntu"

  roles:
    - Ubuntu
    - CentOS
```

3. This is the file content for the **install_prometheus.yml,** this is basically the main ansible playbook file to run in order to do the tasks for both Ubuntu and CentOS remote servers.

```
qcchavez@workstation:~/CPE212_Chavez_HOA9.1$ cat inventory
[Ubuntu]
#Server 1
192.168.56.116

[CentOS]
#CentOS 7 with GUI
192.168.56.119 ansible_user=cchavez
```

4. This is the file content for **inventory,** it shows that the **Server 1** is my remote server for **Ubuntu**, and then my **CentOS 7 with GUI** is my **CentOS** remote server.

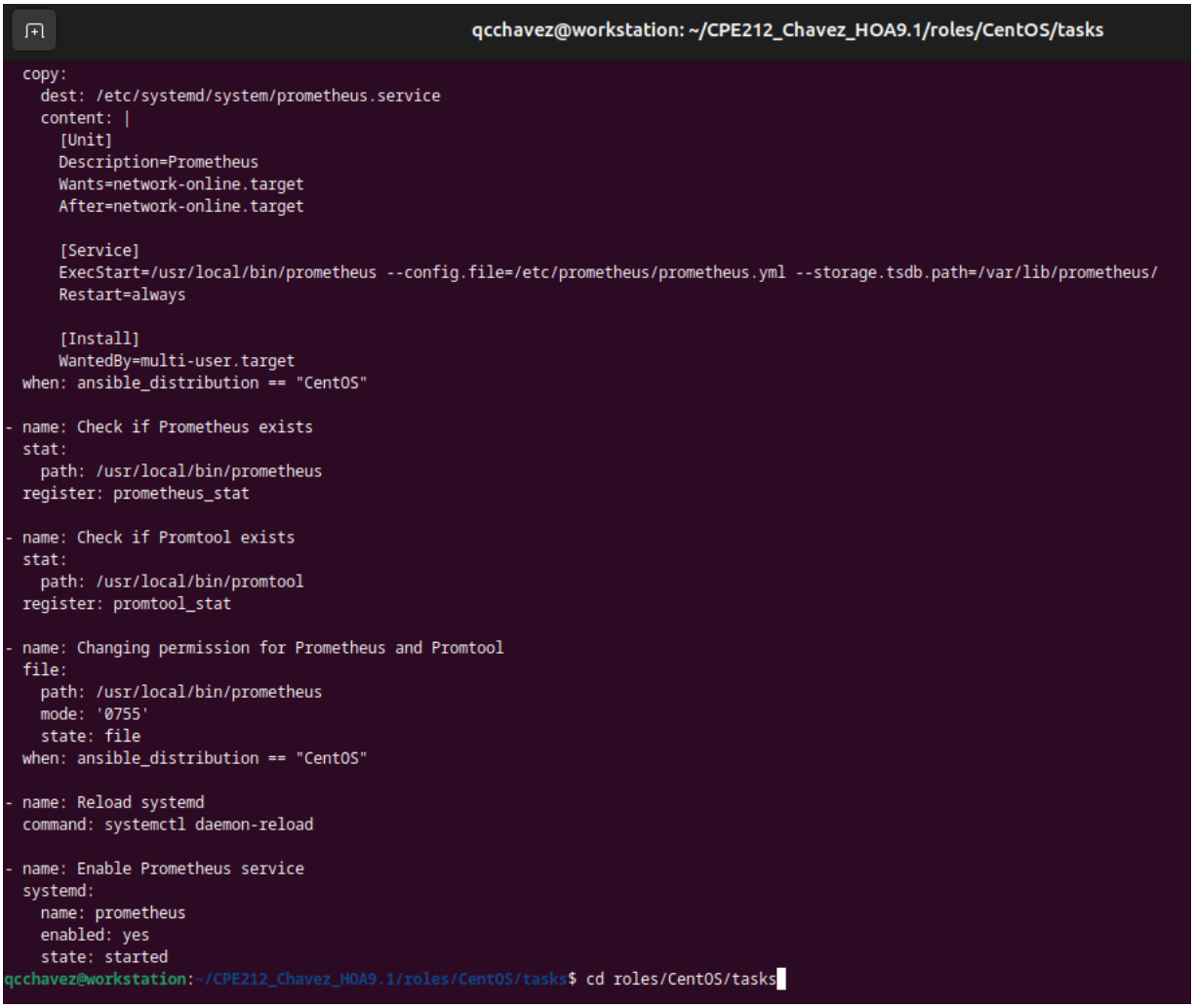


5. This is the **main.yml** for the **CentOS**, this is where the tasks of **CentOS** remote server comes from. In this file, first we needed to install the **EPEL repository** which is required in order to install other packages, which are reliant on the **EPEL repository**. After that, we can download the **prometheus file**, extract it, and install its **dependencies**. After the downloading and installing the required **dependencies** and the **prometheus** itself, we will move the binary of **Prometheus** to the required directory.

qcchavez@workstation: ~/CPE212_Chavez_HOA9.1/roles/CentOS/tasks

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

- name: Create Prometheus configuration file
  copy:
    dest: /etc/prometheus/prometheus.yml
    content: |
      global:
        scrape_interval: 15s

      scrape_configs:
        - job_name: 'prometheus'
          static_configs:
          - targets: ['localhost:9090']
  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]
      ExecStart=/usr/local/bin/prometheus --config.file=/etc/prometheus/prometheus.yml --storage.tsdb.path=/var/lib/prometheus/
      Restart=always

      [Install]
      WantedBy=multi-user.target
  when: ansible_distribution == "CentOS"

- name: Check if Prometheus exists
```

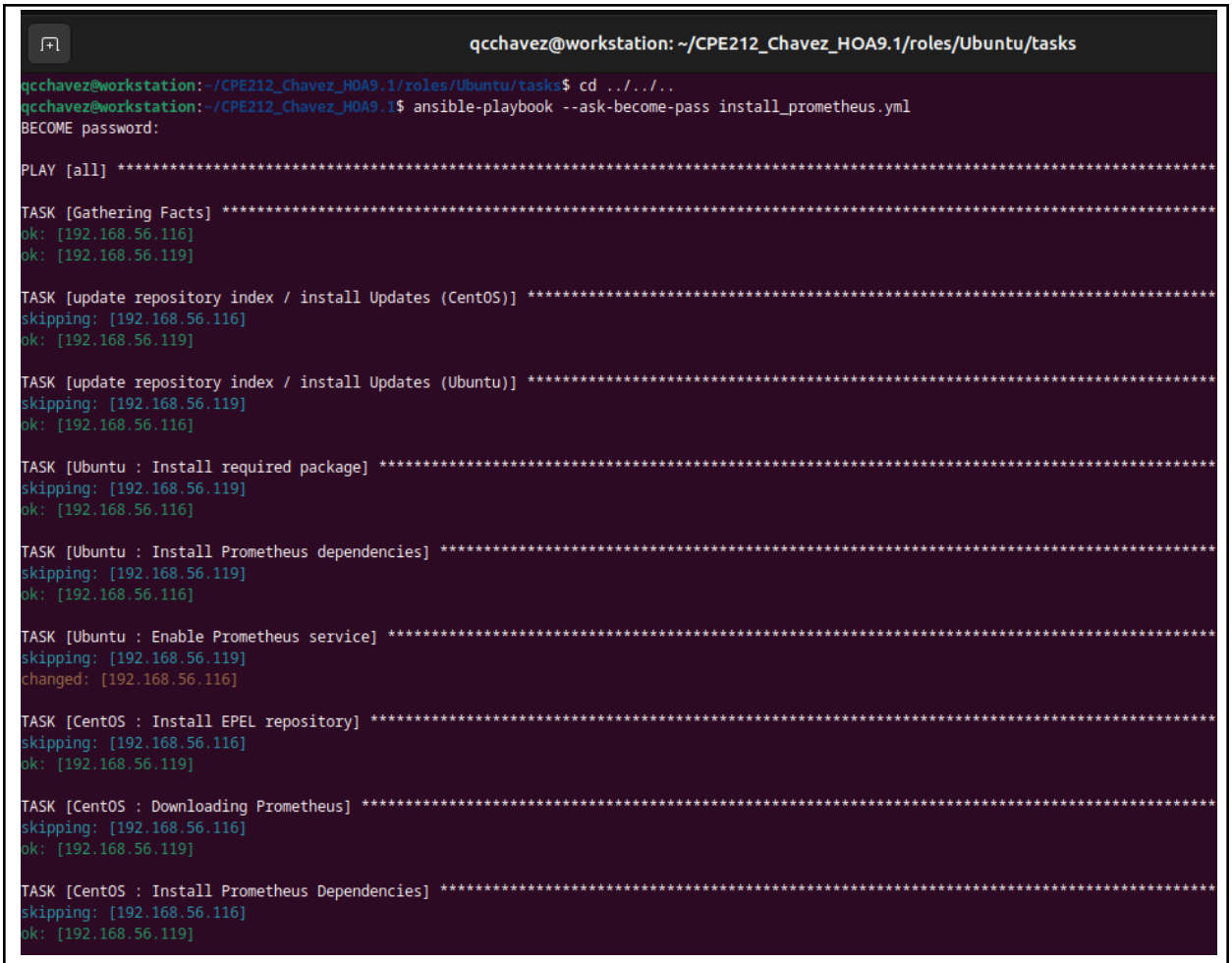
6. We also need to create a directory for the **Prometheus** itself so it can be located where it is needed to be. And then, we will create the **configuration and service file** which are also required in order to make it work

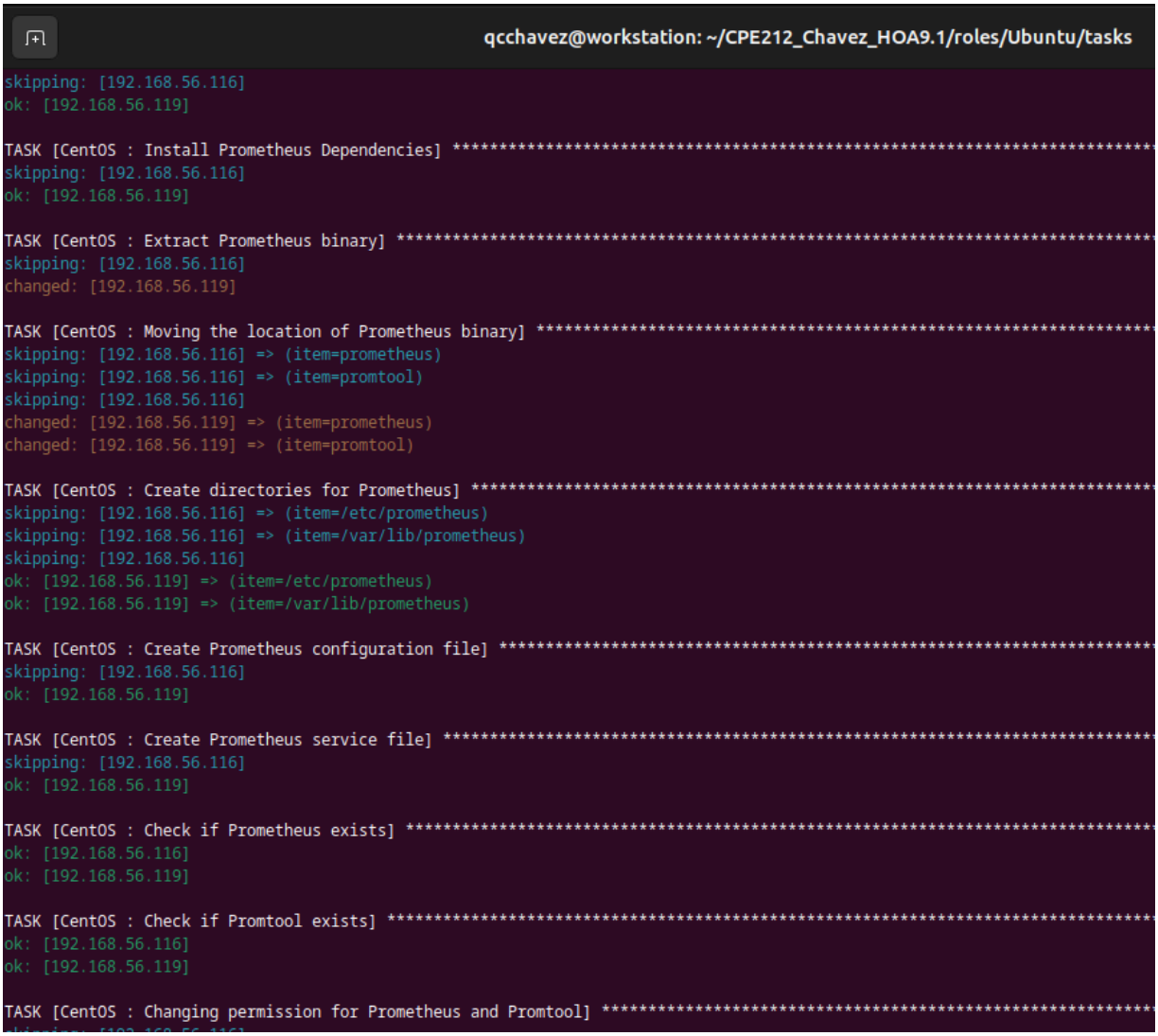


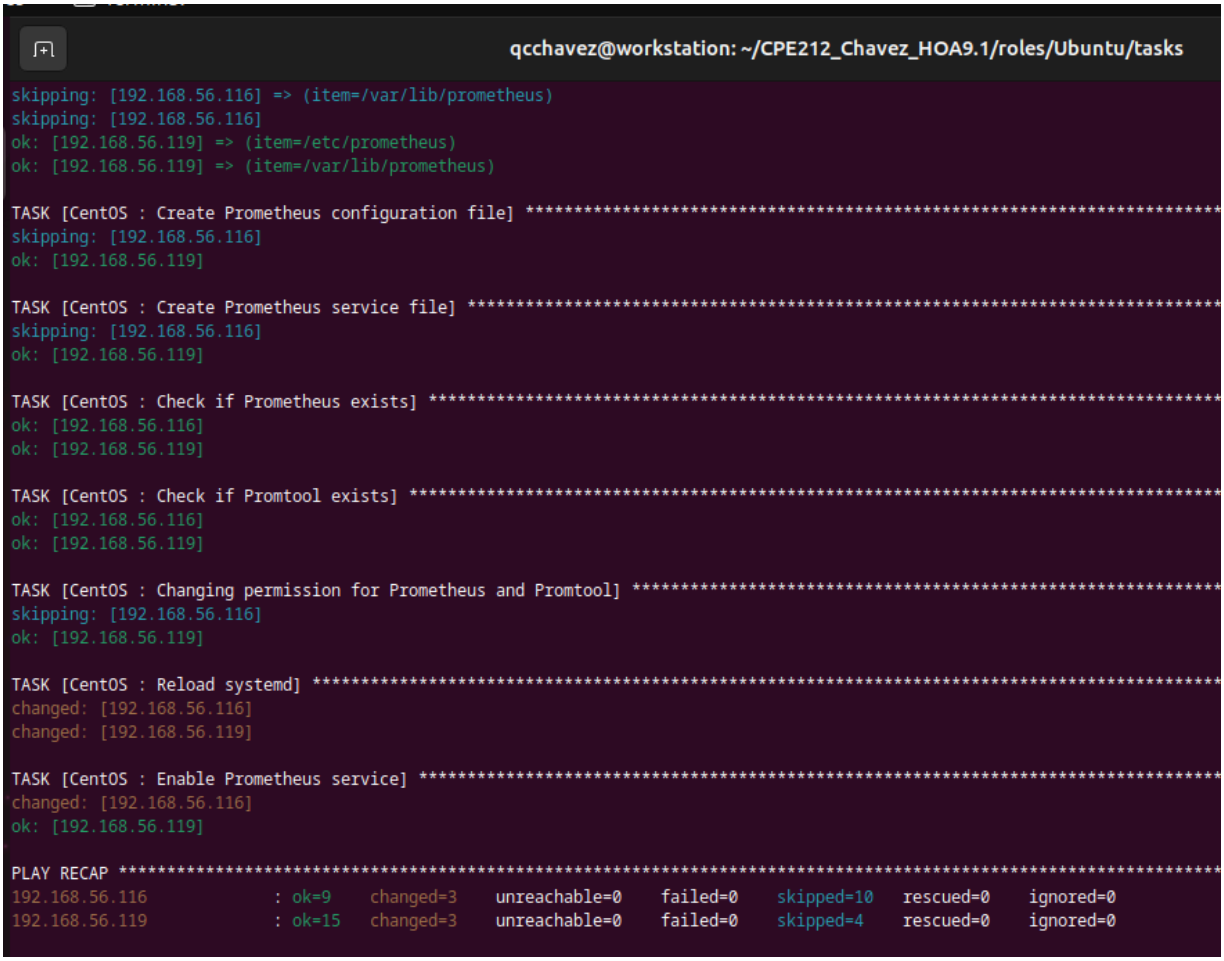
7. Before we will change permissions to **promtool and prometheus .exe files**, we need to double check if they exist on our target directory, which is good based on the screenshots. After that, we will reload the **systemd,** and then run the **prometheus monitoring tool.**

```
qcchavez@workstation:~/CPE212_Chavez_HOA9.1/roles/CentOS/tasks$ cd ../../..
qcchavez@workstation:~/CPE212_Chavez_HOA9.1$ cd roles/Ubuntu/tasks
qcchavez@workstation:~/CPE212_Chavez_HOA9.1/roles/Ubuntu/tasks$ cat main.yml
---
- name: Install required package
  apt:
    name: prometheus
    state: latest
  when: ansible_distribution == "Ubuntu"

- name: Install Prometheus dependencies
  apt:
    name:
    - gcc
    - make
    - wget
    - libgd-dev
    state: latest
  when: ansible_distribution == "Ubuntu"

- name: Enable Prometheus service
  service:
    name: prometheus
    state: restarted
    enabled: true
  when: ansible_distribution == "Ubuntu"
qcchavez@workstation:~/CPE212_Chavez_HOA9.1/roles/Ubuntu/tasks$
```

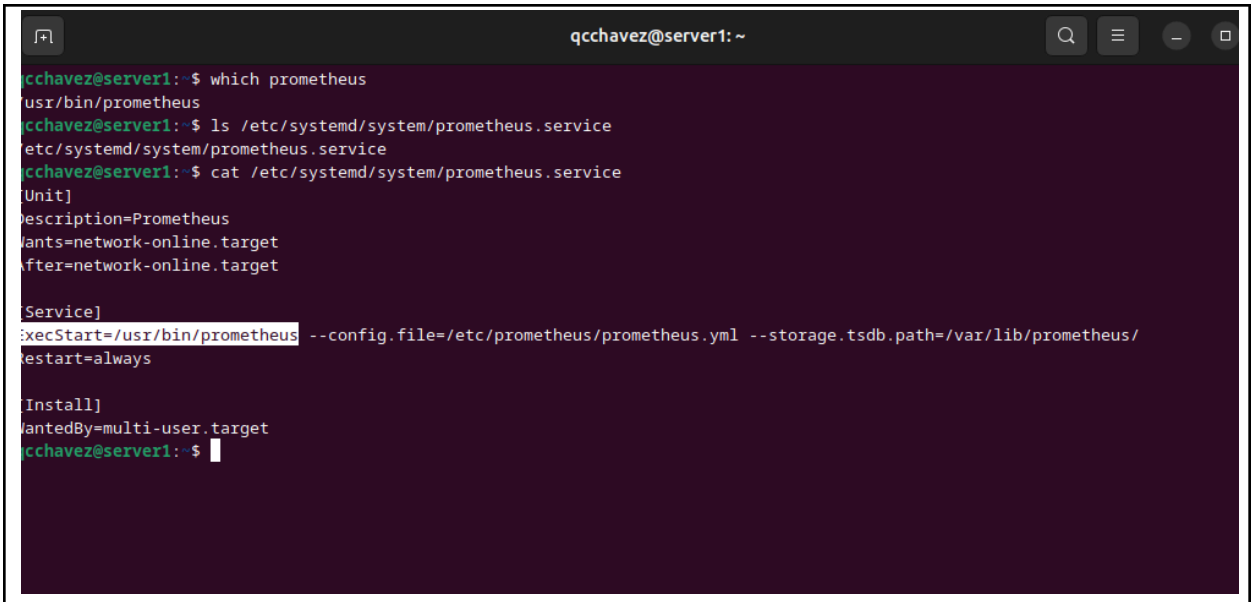
8. This is the file content for **main.yml** of the Ubuntu, first, it needs to install the required package which is the **prometheus** package itself, installing the **prometheus dependencies,** and **enabling the package**.


qcchavez@workstation: ~/CPE212_Chavez_HOA9.1/roles/Ubuntu/tasks
qcchavez@workstation:~/CPE212_Chavez_HOA9.1/roles/Ubuntu/tasks$ cd ../../..
qcchavez@workstation:~/CPE212_Chavez_HOA9.1$ ansible-playbook --ask-become-pass install_prometheus.yml
BECOME password:
PLAY [all] ****
TASK [Gathering Facts] ****
ok: [192.168.56.116]
ok: [192.168.56.119]
TASK [update repository index / install Updates (CentOS)] ****
skipping: [192.168.56.116]
ok: [192.168.56.119]
TASK [update repository index / install Updates (Ubuntu)] ****
skipping: [192.168.56.119]
ok: [192.168.56.116]
TASK [Ubuntu : Install required package] ****
skipping: [192.168.56.119]
ok: [192.168.56.116]
TASK [Ubuntu : Install Prometheus dependencies] ****
skipping: [192.168.56.119]
ok: [192.168.56.116]
TASK [Ubuntu : Enable Prometheus service] ****
skipping: [192.168.56.119]
changed: [192.168.56.116]
TASK [CentOS : Install EPEL repository] ****
skipping: [192.168.56.116]
ok: [192.168.56.119]
TASK [CentOS : Downloading Prometheus] ****
skipping: [192.168.56.116]
ok: [192.168.56.119]
TASK [CentOS : Install Prometheus Dependencies] ****
skipping: [192.168.56.116]
ok: [192.168.56.119]


qcchavez@workstation: ~/CPE212_Chavez_HOA9.1/roles/Ubuntu/tasks
skipping: [192.168.56.116]
ok: [192.168.56.119]
TASK [CentOS : Install Prometheus Dependencies] ********
skipping: [192.168.56.116]
ok: [192.168.56.119]
TASK [CentOS : Extract Prometheus binary] ********
skipping: [192.168.56.116]
changed: [192.168.56.119]
TASK [CentOS : Moving the location of Prometheus binary] ********
skipping: [192.168.56.116] => (item=prometheus)
skipping: [192.168.56.116] => (item=promtool)
skipping: [192.168.56.116]
changed: [192.168.56.119] => (item=prometheus)
changed: [192.168.56.119] => (item=promtool)
TASK [CentOS : Create directories for Prometheus] ********
skipping: [192.168.56.116] => (item=/etc/prometheus)
skipping: [192.168.56.116] => (item=/var/lib/prometheus)
skipping: [192.168.56.116]
ok: [192.168.56.119] => (item=/etc/prometheus)
ok: [192.168.56.119] => (item=/var/lib/prometheus)
TASK [CentOS : Create Prometheus configuration file] ********
skipping: [192.168.56.116]
ok: [192.168.56.119]
TASK [CentOS : Create Prometheus service file] ********
skipping: [192.168.56.116]
ok: [192.168.56.119]
TASK [CentOS : Check if Prometheus exists] ********
ok: [192.168.56.116]
ok: [192.168.56.119]
TASK [CentOS : Check if Promtool exists] ********
ok: [192.168.56.116]
ok: [192.168.56.119]
TASK [CentOS : Changing permission for Prometheus and Promtool] ********



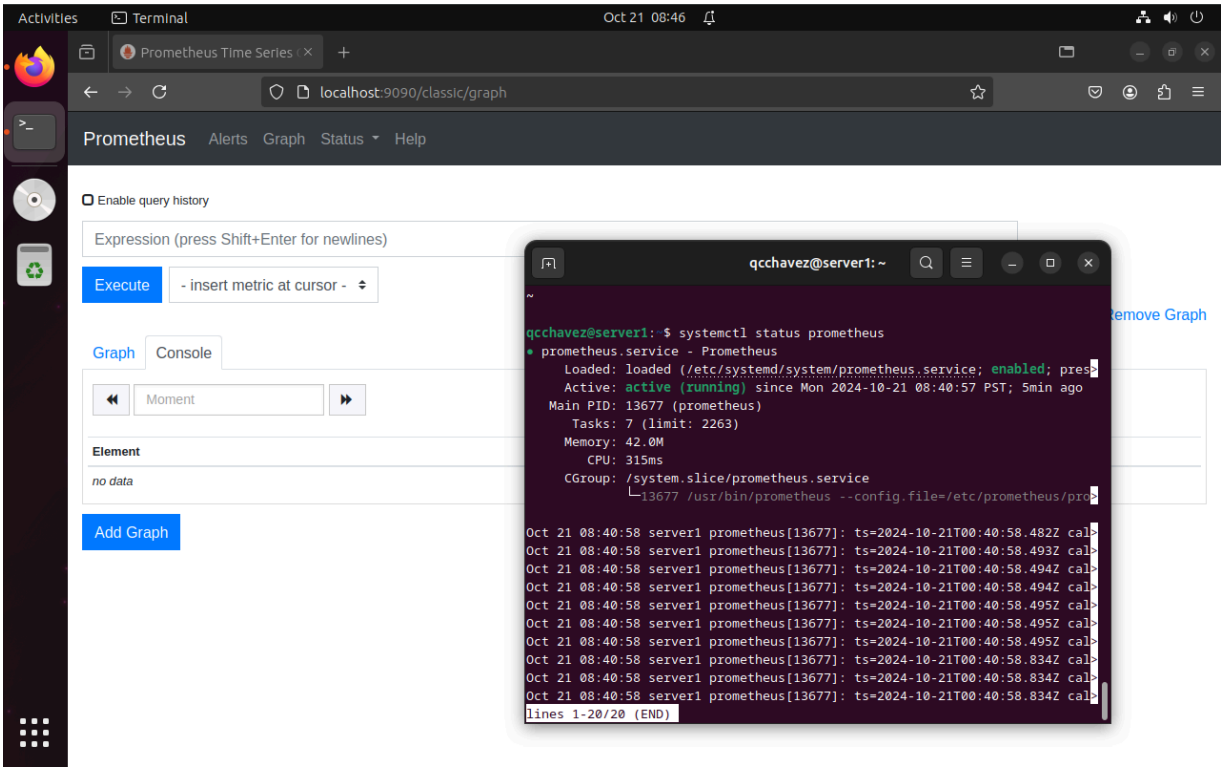
9.1 This is the output of running the **install_prometheus.yml**, the main ansible-playbook yml file for this activity, it shows that all of the tasks worked properly.



9.2 Incase that the **remote servers** does not start properly or it says **“failed”** when prompting the command **“systemctl status prometheus”**, make sure to double check where the prometheus package was installed/extracted by entering the syntax **which prometheus**, and if it shows the correct directory or where the **prometheus package** is located, go to the **service file** for the **prometheus** which is the **prometheus.service**. Make sure that the directory in the **execstart** is the same as the directory when you prompted **which prometheus**, and after that, re-run the main ansible playbook yml file.

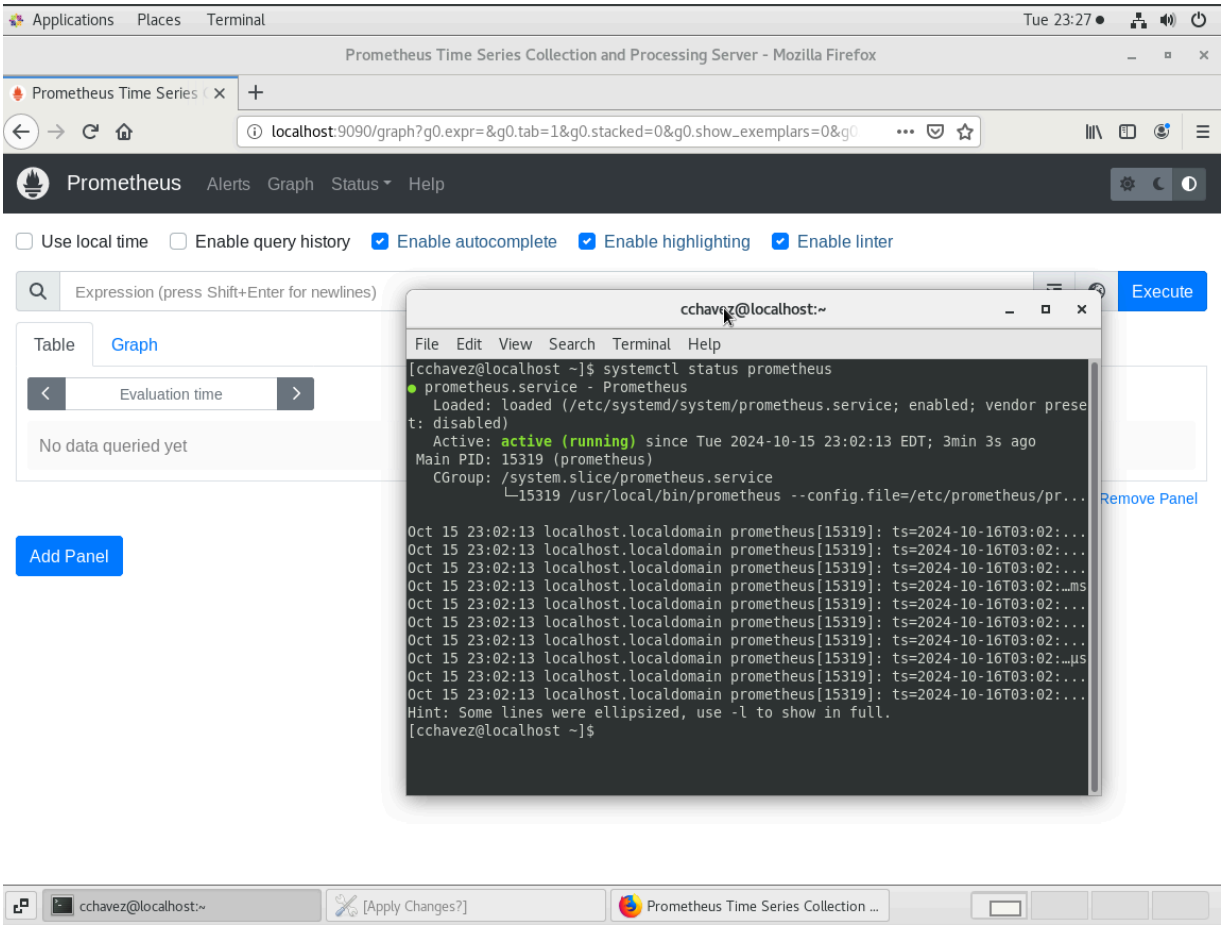
Task No. 3

Ubuntu Desktop



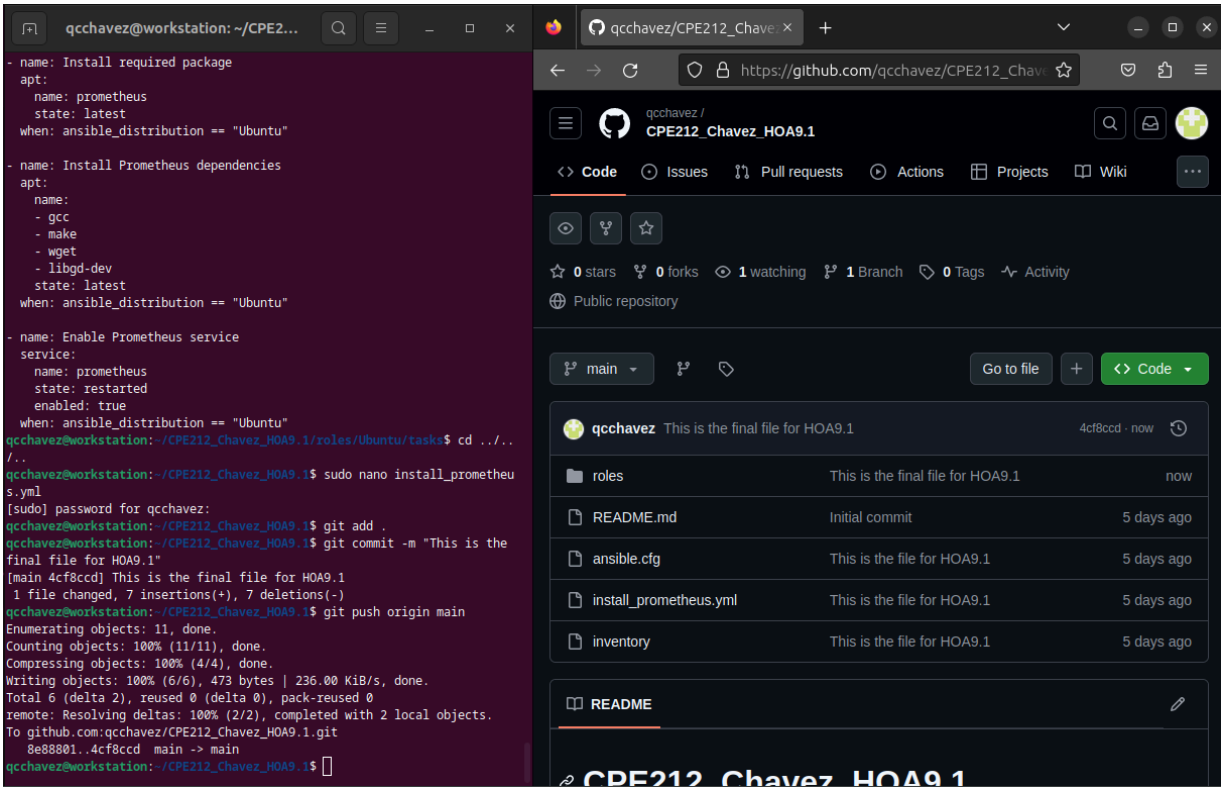
- In order to show that **prometheus** is working properly, I opened the remote server for the Ubuntu which is my **Server1,** and prompted the command **systemctl status prometheus** to check if prometheus is **active**, prompting **localhost:9090** on the web browser will also determine if prometheus is working properly.

CentOS 7



- In order to show that **prometheus** is working properly, I opened the remote server for the CentOS**,** and prompted the command **systemctl status prometheus** to check if prometheus is **active**, prompting **localhost:9090** on the web browser will also determine if prometheus is working properly.

Task No. 4



10. This is where I committed my **repository** for this activity to **GitHub**.

Reflections:

Answer the following:

1. What are the benefits of having a performance monitoring tool?
   - The benefits of having a performance monitoring tool is that it provides a real-time tracking of system performance of the remote servers. It also optimizes resources, detects issues as soon as possible, and enhances the security of its remote servers.

Conclusions:

- In this activity, I have learned that the performance monitoring tool is really important when managing remote servers. It is also important to double check the syntaxes in order to prevent excessive errors when running the playbooks. Make sure that the packages are running in the remote servers by prompting **systemctl status (name of the package).**