| Name: Gabiano, Chris Leonard A. | Date Performed: 10/23/23 |
|---|---|
| Course/Section: CPE32S6 | Date Submitted: 10/23/23 |
| Instructor: Dr. Jonathan Taylar | Semester and SY: 2023 - 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 (screenshots and explanations)

**PART 1: creating repository**

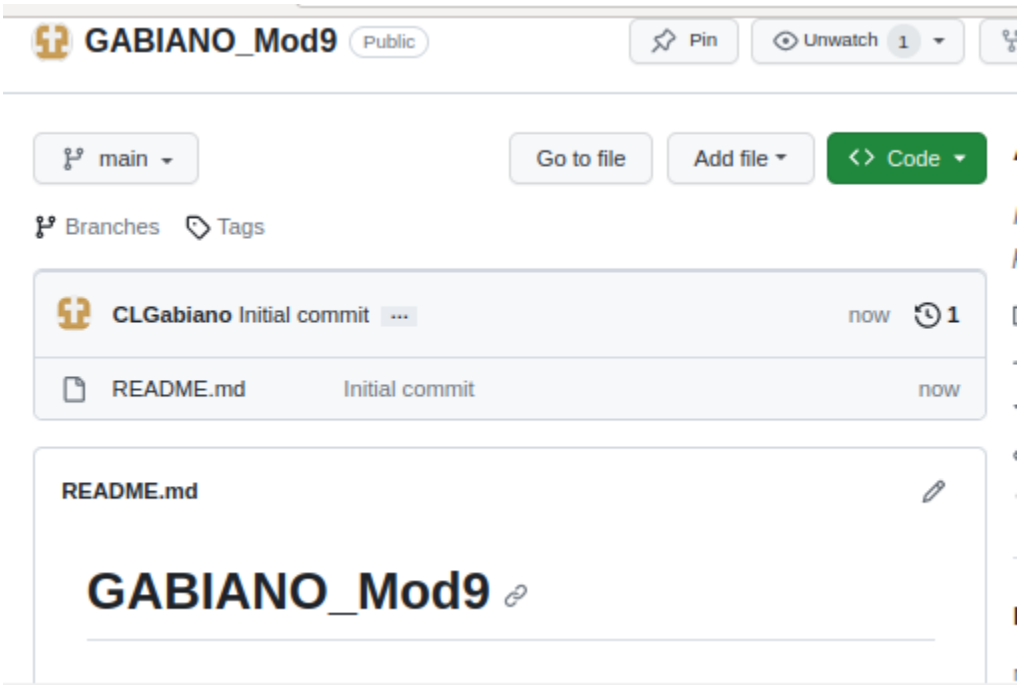



fig 1: create **activity 9** repository.

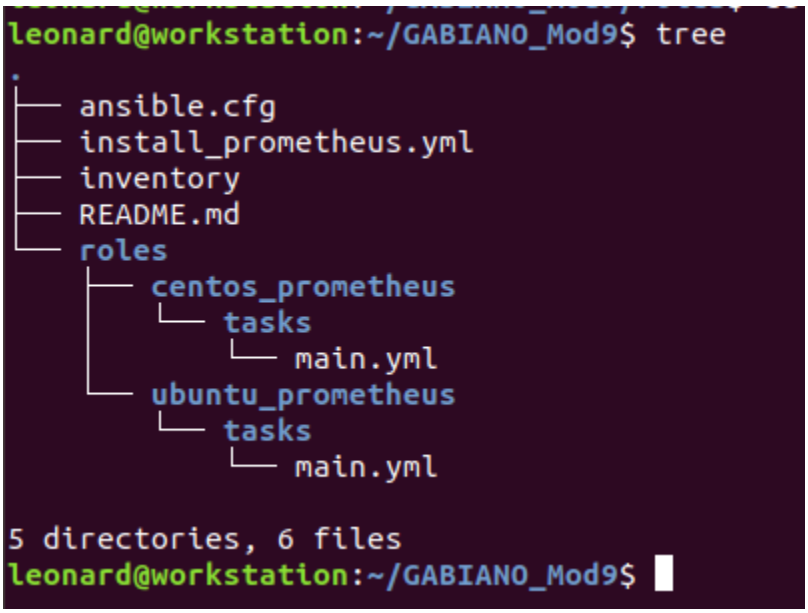



fig 2: files ansible.cfg, inventory created among directories.

## PART 2: creating files for playbooks

```
  GNU nano 2.9.3                                    install_prometheus.yml

---

- hosts: all
  become: true
  pre_tasks:

  - name: install updates (CentOS)
    dnf:
      update_only: yes
      update_cache: yes
    when: ansible_distribution == "CentOS"

  - name: install wget (CentOS)
    dnf:
      name: wget
      state: latest
    when: ansible_distribution == "CentOS"

  - name: install updates (Ubuntu)
    apt:
      upgrade: dist
      update_cache: yes
    when: ansible_distribution == "Ubuntu"

- hosts: ubuntu_prometheus
  become: true
  roles:
    - ubuntu_prometheus

- hosts: centos_prometheus
  become: true
  roles:
    - centos_prometheus
```

fig 3: playbook “install_prometheus.yml

```
GNU nano 2.9.3                                      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
```

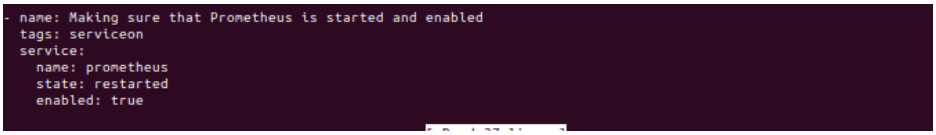



fig 4: main.yml of centos_prometheus

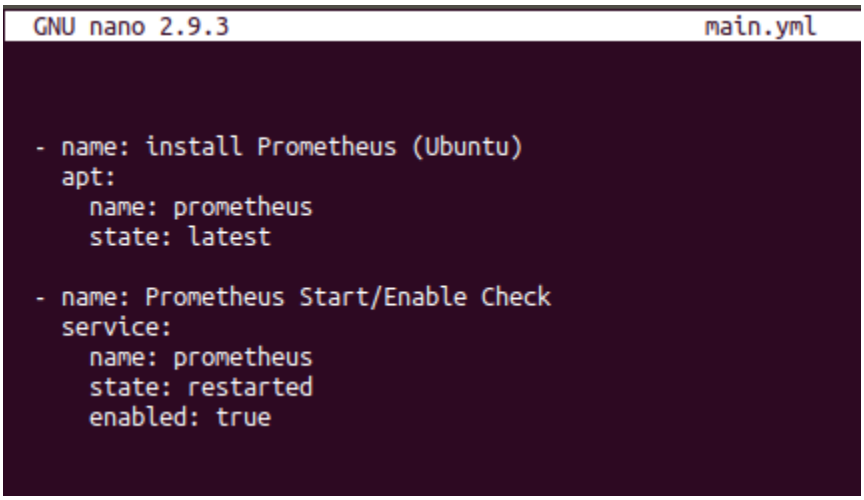
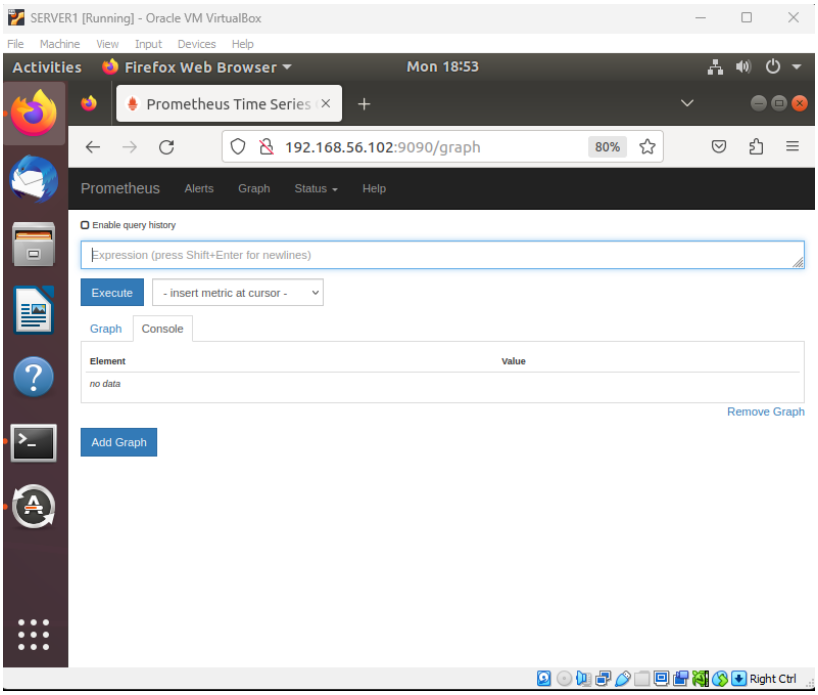
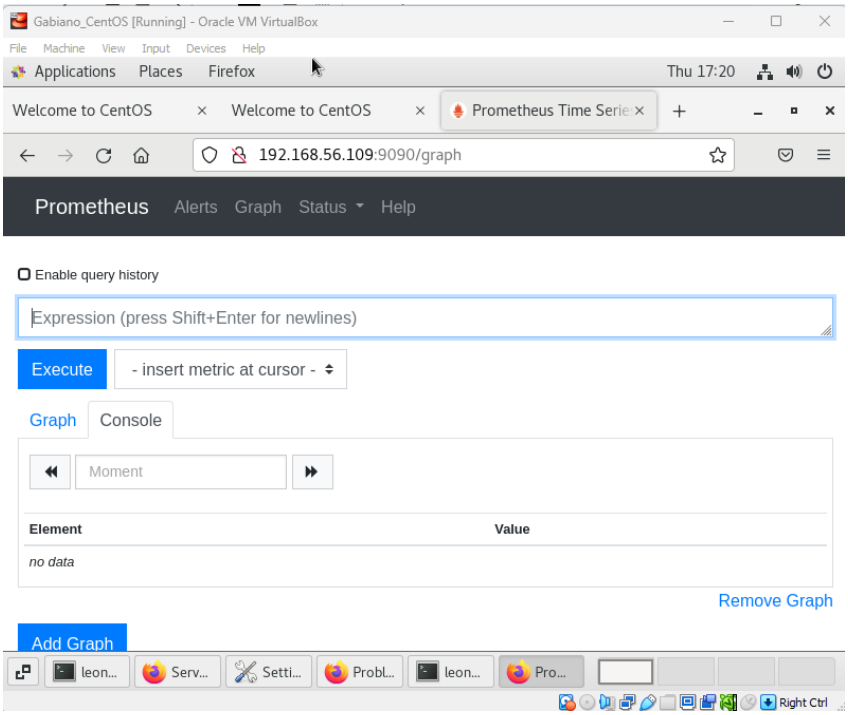



fig 5: main.yml of ubuntu_prometheus

## PART 3: Installation Verification




**UBUNTU**




**CentOS**

https://github.com/CLGabiano/GABIANO_Mod9.git

**Reflections:**

Answer the following:

1. What are the benefits of having a performance monitoring tool?

   A performance monitoring tool provides real-time insights into system health, enabling proactive issue detection and resolution, thus minimizing downtime and improving overall system reliability. Additionally, it helps optimize resource utilization, leading to cost savings and enhanced user experience.

**Conclusions:**

In conclusion, we designed an Ansible workflow to install, configure, and manage performance monitoring tools, specifically Prometheus, for both Ubuntu and CentOS. We organized the playbook using roles for modularity and ease of maintenance. By following the step-by-step process, we successfully installed Prometheus on both Ubuntu and CentOS, ensuring effective performance monitoring.