

<b>Name: Ilagan, Carlo Hideki D.</b>	<b>Date Performed: 10/16/2024</b>
<b>Course/Section: CPE31S2</b>	<b>Date Submitted: 10/28/2024</b>
<b>Instructor: Engr. Robin Valenzuela</b>	<b>Semester and SY: 1st Sem/2024-2025</b>
<b>Activity 9: Install, Configure, and Manage Performance Monitoring tools</b>	
<b>1. Objectives</b>	
Create and design a workflow that installs, configure and manage enterprise performance tools using Ansible as an Infrastructure as Code (IaC) tool.	
<b>2. Discussion</b>	
<p>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.</p> <p><b>Prometheus</b></p> <p>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: <a href="#">Prometheus - Monitoring system &amp; time series database</a></p> <p><b>Cacti</b></p> <p>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: <a href="#">Cacti® - The Complete RRDTool-based Graphing Solution</a></p>	
<b>3. Tasks</b>	
<ol style="list-style-type: none"> <li>1. Create a playbook that installs Prometheus in both Ubuntu and CentOS. Apply the concept of creating roles.</li> <li>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.)</li> <li>3. Show an output of the installed Prometheus for both Ubuntu and CentOS.</li> <li>4. Make sure to create a new repository in GitHub for this activity.</li> </ol>	
<b>4. Output (screenshots and explanations)</b>	
<b>main.yml file Playbook:</b>	

```
hideki@workstation: ~/HOA9.1-chilagan/roles/workstations/tasks
GNU nano 7.2 main.yml
---
- name: Add GPG key for the Prometheus repository (Ubuntu)
  apt_key:
    url: https://packages.grafana.com/gpg.key
    state: present
  when: ansible_distribution == "Ubuntu"

- name: Add Prometheus APT repository (Ubuntu)
  apt_repository:
    repo: deb https://packages.grafana.com/oss/deb stable main
    state: present
    filename: grafana
  when: ansible_distribution == "Ubuntu"

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

- name: Enable and start Prometheus service (Ubuntu)
  service:
    name: prometheus
    enabled: yes
    state: started
  when: ansible_distribution == "Ubuntu"

- name: Open port 9090 for Prometheus (Ubuntu)
  ufw:
    rule: allow
    port: 9090
    proto: tcp
    state: enabled
  when: ansible_distribution == "Ubuntu"

- name: Download Prometheus(CentOS)
```

```
hideki@workstation: ~/HOA9.1-chilagan/roles/workstations/tasks
GNU nano 7.2 main.yml
- name: Download Prometheus(CentOS)
  get_url:
    url: "https://github.com/prometheus/prometheus/releases/download/v3.0.0-beta.1/p
    dest: /tmp/prometheus.tar.gz
  when: ansible_distribution == "CentOS"

- name: Extract Prometheus (CentOS)
  ansible.builtin.shell: tar -zxvf /tmp/prometheus.tar.gz -C /tmp/
  when: ansible_distribution == "CentOS"

- name: Create Prometheus user (CentOS)
  ansible.builtin.user:
    name: prometheus
    state: present
  when: ansible_distribution == "CentOS"

- name: Create Prometheus directory (CentOS)
  ansible.builtin.file:
    path: /opt/prometheus
    state: directory
  when: ansible_distribution == "CentOS"

- name: Set ownership and permissions for Prometheus (CentOS)
  ansible.builtin.file:
    path: /opt/prometheus
    owner: prometheus
    group: prometheus
    mode: '0755'
  when: ansible_distribution == "CentOS"

- name: Create Prometheus service file (CentOS)
  ansible.builtin.template:
    src: prometheus.service.j2
    dest: /etc/systemd/system/prometheus.service
  when: ansible_distribution == "CentOS"
```

```
- name: Start Prometheus service (CentOS)
  ansible.builtin.service:
    name: prometheus
    enabled: yes
    state: started
  when: ansible_distribution == "CentOS"

- name: Open Firewall Port for Prometheus (CentOS)
  ansible.builtin.shell: firewall-cmd --add-port=9090/tcp --permanent
  when: ansible_distribution == "CentOS"

- name: Ensure Firewall Rule Reloaded (CentOS)
  ansible.builtin.shell: firewall-cmd --reload
  when: ansible_distribution == "CentOS"

- name: Enable Prometheus on system boot (CentOS)
  ansible.builtin.service:
    name: prometheus
    enabled: yes
  when: ansible_distribution == "CentOS"
```

Provided screenshots above were the commands or queries that were used in order to install, extract, and obtain prometheus in both Ubuntu and CentOS. Note that this is only the queries for workstations, there are still yml files that are needed to run these commands.

**prometheus.service.j2 file:**

```
GNU nano 7.2 prometheus.service.j2
[Unit]
Description=Prometheus Server
Wants=network-online.target
After=network-online.target

[Service]
User=prometheus
Group=prometheus
Type=simple
ExecStart=/opt/prometheus/prometheus-2.30.0.linux-amd64/prometheus --config.file=/opt/>
Restart=always

[Install]
WantedBy=multi-user.target
```

This part of the activity is a configuration that will allow Prometheus to start at boot and restart if it stops, running as a background service that will run in both Ubuntu and Centos since there are queries to copy the set of commands found in the main.yml.

**prometheus.yml file:**

```
hideki@workstation:~/HOA9.1-chillagan$ cat prometheus.yml
```

```
---
- hosts: all
  become: true
  pre_tasks:

    - name: update repository index (CentOS)
      tags: always
      yum:
        update_cache: yes
        use_backend: dnf4
      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: workstations
  become: true
  roles:
    - workstations
```

In this part of the activity, this is now the executable file for the commands created above, first the `pre_tasks` to update both Ubuntu and Centos will run. Next, the `main.yml` file will run through the `-hosts: workstations` since this is where the `yml` file is found.

## Ansible playbook run:

```
TASK [workstations : Start Prometheus service (CentOS)] *****
*****
changed: [192.168.56.105]

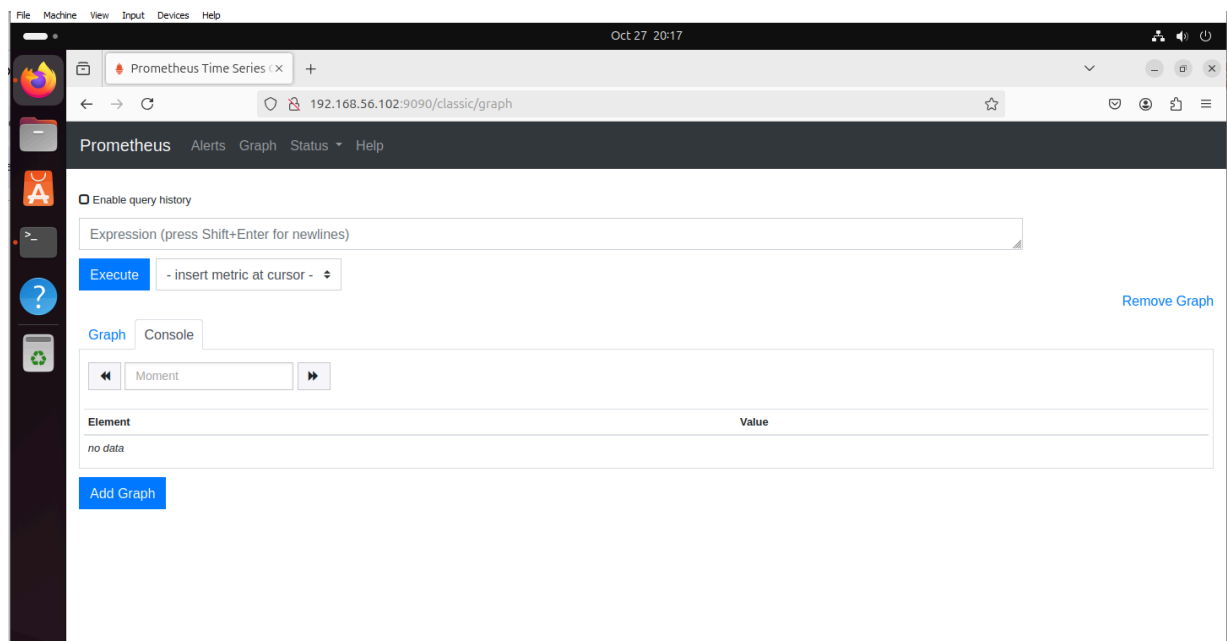
TASK [workstations : Open Firewall Port for Prometheus (CentOS)] *****
*****
changed: [192.168.56.105]

TASK [workstations : Ensure Firewall Rule Reloaded (CentOS)] *****
*****
changed: [192.168.56.105]

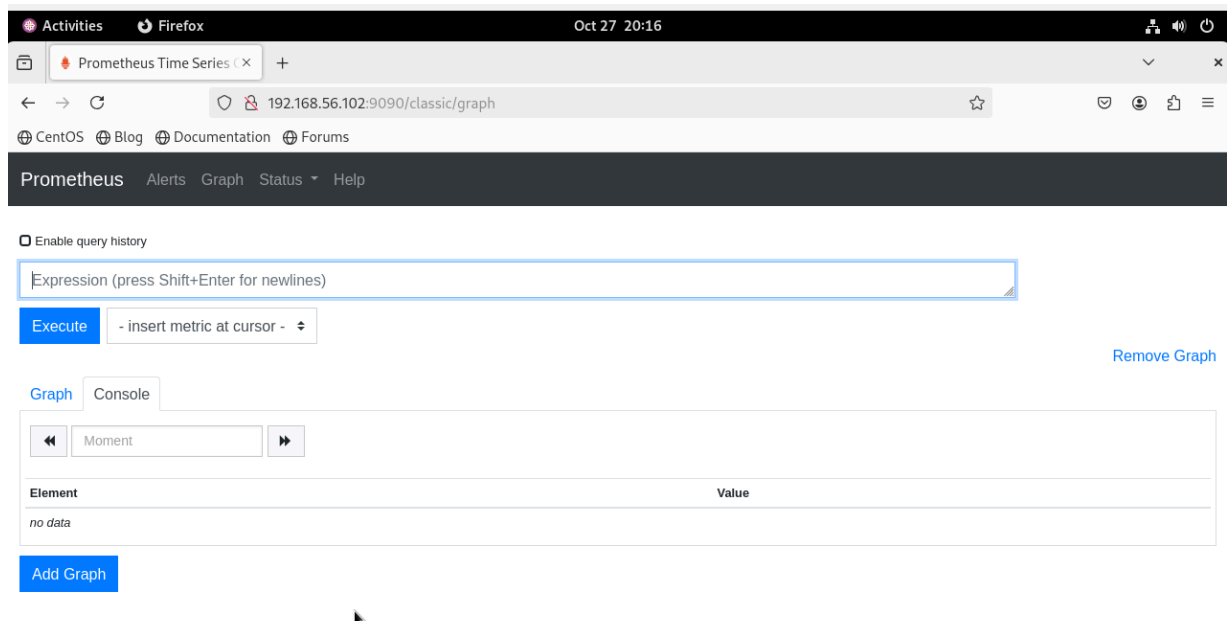
TASK [workstations : Enable Prometheus on system boot (CentOS)] *****
*****
ok: [192.168.56.105]

PLAY RECAP *****
*****
192.168.56.102      : ok=9    changed=2    unreachable=0    failed=0    skipped=
2    rescued=0    ignored=0
192.168.56.105    : ok=16   changed=5    unreachable=0    failed=0    skipped=
7    rescued=0    ignored=0
```

## Ubuntu Prometheus on browser:



## CentOS Prometheus on browser:



By typing the IP together with the created firewall port, the prometheus will run on both servers.

**HOA9.1-chilagan all files:**

```
hideki@workstation:~/HOA9.1-chilagan$ tree
.
├── ansible.cfg
├── inventory
├── prometheus.service.j2
├── prometheus.yml
├── README.md
├── roles
│   ├── base
│   │   └── tasks
│   │       └── main.yml
│   └── workstations
│       └── tasks
│           └── main.yml
```



**Reflections:**

Answer the following:

1. What are the benefits of having a performance monitoring tool?
  - The main obvious benefit of having a performance monitoring tool such as prometheus is to detect issues that will or can occur as early as possible. Monitoring tools can alert an administrator or even the user of the workstation the issues before it gets worse, this will allow them to address the said problem instantly. Another benefit is an enhanced security for workstations. Irregularities in performance metrics can indicate potential security incidents, such as DDoS attacks or unauthorized access. With this type of application or software, we can prevent these attacks and provide a safer environment for users.

**Conclusions:**

To conclude this activity, the use of the Ansible playbook to install Prometheus on both Ubuntu and CentOS simplifies the setup and ensures consistency of different workstations. While automation handles these configurations, administrators can spend less time on manual setup and more on fine-tuning performance and responding to critical alerts. This approach not only boosts reliability and consistency but also makes managing monitoring solutions more scalable and easier to maintain. Performing this activity provided me knowledge about the importance of monitoring tools. Although we already installed another tool from the previous activity, it is good to know that there are also other applications or software that are available and can be installed in linux operating systems and redhat operating systems.