

Name: Zamora, Angelo E.	Date Performed: 10 - 28 - 2024
Course/Section: BSCPE - CpE31S2	Date Submitted: 10 - 28 - 2024
Instructor: Engr. Robin Valenzuela	Semester and SY: 1st Semester 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	
<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>Prometheus</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: Prometheus - Monitoring system & time series database</p> <p>Cacti</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: Cacti® - The Complete RRDTool-based Graphing Solution</p>	
3. Tasks	
<ol style="list-style-type: none"> 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)	

Step 1: Create a Repository

```
zamora_admin@workstation:~/TIP_HOA-9.1_ZAMORA_Angelo$ ls
ansible.cfg  inventory  prometheus.yml  README.md  roles
zamora_admin@workstation:~/TIP_HOA-9.1_ZAMORA_Angelo$
```

Create a repo for the HOA 9.1 and we will build the needed files for the task.

Here are the contents of the foundational files for the repo the inventory and the ansible.cfg to setup the repo (based on your own manage Nodes setup):

ansible.cfg

```
File Edit View Search Terminal Help
GNU nano 2.9.3 ansible.cfg

[defaults]
inventory = /home/zamora_admin/TIP_HOA-9.1_ZAMORA_Angelo/inventory
remote_user = zamora_admin
host_key_checking = True
```

inventory

```
File Edit View Search Terminal Help
GNU nano 2.9.3

[Ubuntu]
192.168.56.108
192.168.56.109
192.168.56.112

[CentOS]
192.168.56.110 ansible_user=azamora_admin
192.168.56.111 ansible_user=azamora_admin
```

192.168.56.108 - Server 1

192.168.56.109 - Server 2

192.168.56.112 - Server 3

192.168.56.110 - CentOS Node 1

192.168.56.111 - CentOS Node 2

Step 2: Create the directory “roles”

Create the directory roles and create two directories inside the “roles” directory. The name for the two is base and workstations as these will set up the roles approach in ansible playbook.

```
zamora_admin@workstation:~/TIP_H0A-9.1_ZAMORA_Angelo$ tree roles
roles
├── base
│   └── tasks
│       └── main.yml
└── workstations
    └── tasks
        └── main.yml

4 directories, 2 files
zamora_admin@workstation:~/TIP_H0A-9.1_ZAMORA_Angelo$
```

After creating the base and workstations directory, create a directory name tasks in each of the two so that we will create the main.yml playbook for each role.

Step 3: Code the main playbook:

Create a file main.yml and this is the content per role

base:

```
File Edit View Search Terminal Help
GNU nano 2.9.3
--
- name: install updates (CentOS)
  tags: always
  dnf:
    name: "*"
    state: latest
  when: ansible_distribution == "CentOS"

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

Here, the purpose of these playbook serves as a routine task for each Manage node to update to its latest everytime we ran the Key Playbook

workstations:

Unset

```
- 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/prometheus-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: 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/prometheus
  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/prometheus
  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
  when: ansible_distribution == "CentOS"

- name: Reload systemd
  command: systemctl daemon-reload
  when: ansible_distribution == "CentOS"

- name: Start Prometheus Service
  systemd:
    name: prometheus
    enabled: yes
    state: started
  when: ansible_distribution == "CentOS"

```

```
- name: Start Prometheus Service (Ubuntu)
  systemd:
    name: prometheus
    enabled: yes
    state: started
  when: ansible_distribution == "Ubuntu"
```

Explanation:

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

```
- name: Start Prometheus Service (Ubuntu)
  systemd:
    name: prometheus
    enabled: yes
    state: started
  when: ansible_distribution == "Ubuntu"
```

For Ubuntu, it needs two protocols: Install the prometheus and run the service itself. This is needed so that when we run the Manage Nodes everything is setup and ready to go for the Prometheus Tool.

```
- name: Install Prometheus (CentOS)
  unarchive:
    src: https://github.com/prometheus/prometheus/releases/download/v2.30.0/prometheus-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: 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/prometheus
  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/prometheus
  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
  when: ansible_distribution == "CentOS"

```

For the CentOS, it requires a lot of setup to make sure the tool is up and running. Since the package of Prometheus does not exist in the package repo of CentOS. We source the package of the tool itself from github.

```
- name: Reload systemd
  command: systemctl daemon-reload
  when: ansible_distribution == "CentOS"

- name: Start Prometheus Service
  systemd:
    name: prometheus
    enabled: yes
    state: started
  when: ansible_distribution == "CentOS"
```

After we install into the machines, we transfer the files and create a directory for the binaries of the tool itself. After doing so, this will create a file, a prometheus.service file, its task is to run the service of the Prometheus in the CentOS. Then we let the playbook start the service for automation purposes.

After that create prometheus.yml in the main repo and this is the content:

```
GNU nano 2.9.3 prometheus.yml
--
- hosts: all
  become: true
  roles:
    - base

- hosts: Ubuntu, CentOS
  become: true
  roles:
    - workstations
```

This will activate the roles folder and run the main.yml files for each roles indicated here.

Run the prometheus.yml playbook to install the prometheus for Ubuntu and CentOS.


```
zamora_admin@workstation:~/TIP_HOA-9.1_ZAMORA_Angelo$ ansible-playbook --ask-become-pass prometheus.yml
SUDO password:
```

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

```
TASK [Gathering Facts] *****
```

```
ok: [192.168.56.109]
ok: [192.168.56.108]
ok: [192.168.56.112]
ok: [192.168.56.110]
ok: [192.168.56.111]
```

```
TASK [base : install updates (CentOS)] *****
```

```
skipping: [192.168.56.108]
skipping: [192.168.56.109]
skipping: [192.168.56.112]
ok: [192.168.56.110]
ok: [192.168.56.111]
```

```
TASK [base : install updates (Ubuntu)] *****
```

```
skipping: [192.168.56.110]
skipping: [192.168.56.111]
ok: [192.168.56.108]
ok: [192.168.56.112]
ok: [192.168.56.109]
```

```
PLAY [Ubuntu, CentOS] *****
```

```
TASK [Gathering Facts] *****
```

```
ok: [192.168.56.108]
ok: [192.168.56.109]
ok: [192.168.56.110]
ok: [192.168.56.111]
ok: [192.168.56.112]
```

```
TASK [workstations : Install Prometheus (Ubuntu)] *****
```

```
skipping: [192.168.56.110]
skipping: [192.168.56.111]
ok: [192.168.56.108]
ok: [192.168.56.109]
ok: [192.168.56.112]
```

```
TASK [workstations : Install Prometheus (CentOS)] *****
```

```
skipping: [192.168.56.108]
skipping: [192.168.56.109]
skipping: [192.168.56.112]
changed: [192.168.56.111]
changed: [192.168.56.110]
```

```
TASK [workstations : Copy Prometheus binaries] *****
```

```
skipping: [192.168.56.108]
skipping: [192.168.56.109]
skipping: [192.168.56.112]
changed: [192.168.56.110]
changed: [192.168.56.111]
```

```
TASK [workstations : Copy Promtool binaries] *****
```

```
skipping: [192.168.56.108]
skipping: [192.168.56.109]
skipping: [192.168.56.112]
changed: [192.168.56.110]
changed: [192.168.56.111]
```

```
TASK [workstations : Create Prometheus directories] *****
skipping: [192.168.56.108] => (item=/etc/prometheus)
skipping: [192.168.56.108] => (item=/var/lib/prometheus)
skipping: [192.168.56.109] => (item=/etc/prometheus)
skipping: [192.168.56.109] => (item=/var/lib/prometheus)
skipping: [192.168.56.112] => (item=/etc/prometheus)
skipping: [192.168.56.112] => (item=/var/lib/prometheus)
changed: [192.168.56.110] => (item=/etc/prometheus)
changed: [192.168.56.111] => (item=/etc/prometheus)
changed: [192.168.56.111] => (item=/var/lib/prometheus)
changed: [192.168.56.110] => (item=/var/lib/prometheus)
```

```
TASK [workstations : Copy prometheus.yml to /etc/prometheus] *****
skipping: [192.168.56.108]
skipping: [192.168.56.109]
skipping: [192.168.56.112]
changed: [192.168.56.111]
changed: [192.168.56.110]
```

```
TASK [workstations : Copy consoles directory to /etc/prometheus] *****
skipping: [192.168.56.108]
skipping: [192.168.56.109]
skipping: [192.168.56.112]
changed: [192.168.56.110]
changed: [192.168.56.111]
```

```
TASK [workstations : Copy console_libraries directory to /etc/prometheus] *****
skipping: [192.168.56.108]
skipping: [192.168.56.109]
skipping: [192.168.56.112]
changed: [192.168.56.110]
changed: [192.168.56.111]
```

```
TASK [workstations : Create prometheus.service file] *****
skipping: [192.168.56.108]
skipping: [192.168.56.109]
skipping: [192.168.56.112]
changed: [192.168.56.110]
changed: [192.168.56.111]
```

```
TASK [workstations : Reload systemd] *****
skipping: [192.168.56.108]
skipping: [192.168.56.109]
skipping: [192.168.56.112]
changed: [192.168.56.111]
changed: [192.168.56.110]
```

```
TASK [workstations : Start Prometheus Service] *****
skipping: [192.168.56.108]
skipping: [192.168.56.109]
skipping: [192.168.56.112]
changed: [192.168.56.111]
changed: [192.168.56.110]
```

```
TASK [workstations : Start Prometheus Service (Ubuntu)] *****
skipping: [192.168.56.110]
skipping: [192.168.56.111]
ok: [192.168.56.108]
ok: [192.168.56.109]
ok: [192.168.56.112]
```

```
PLAY RECAP *****
192.168.56.108      : ok=5    changed=0    unreachable=0    failed=0
192.168.56.109      : ok=5    changed=0    unreachable=0    failed=0
192.168.56.110      : ok=13   changed=10   unreachable=0    failed=0
192.168.56.111      : ok=13   changed=10   unreachable=0    failed=0
192.168.56.112      : ok=5    changed=0    unreachable=0    failed=0
```

Step 3: Check the Nodes if Prometheus is installed

To check if it's installed we will prompt ***systemctl status prometheus*** in Ubuntu and CentOS to check if the service is active.

Ubuntu Nodes:

Server 1:

```
zamora_admin@server1:~$ systemctl status prometheus
● prometheus.service - Monitoring system and time series database
   Loaded: loaded (/lib/systemd/system/prometheus.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2024-10-28 07:46:05 +08; 1h 7min ago
     Docs: https://prometheus.io/docs/introduction/overview/
    Main PID: 829 (prometheus)
      Tasks: 13 (limit: 4656)
    CGroup: /system.slice/prometheus.service
            └─829 /usr/bin/prometheus

Warning: Journal has been rotated since unit was started. Log output is incomplete or oldest than 10 minutes.
lines 1-10/10 (END)
```

Server 2:

```
zamora_admin@server2:~$ systemctl status prometheus
● prometheus.service - Monitoring system and time series database
   Loaded: loaded (/lib/systemd/system/prometheus.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2024-10-28 07:45:56 +08; 1h 8min ago
     Docs: https://prometheus.io/docs/introduction/overview/
    Main PID: 845 (prometheus)
      Tasks: 13 (limit: 4656)
    CGroup: /system.slice/prometheus.service
            └─845 /usr/bin/prometheus

Warning: Journal has been rotated since unit was started. Log output is incomplete or oldest than 10 minutes.
lines 1-10/10 (END)
```

Server 3:

```
zamora_admin@server3:~$ systemctl status prometheus
● prometheus.service - Monitoring system and time series database
   Loaded: loaded (/lib/systemd/system/prometheus.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2024-10-28 07:47:50 +08; 1h 7min ago
     Docs: https://prometheus.io/docs/introduction/overview/
    Main PID: 839 (prometheus)
      Tasks: 12 (limit: 4656)
    CGroup: /system.slice/prometheus.service
            └─839 /usr/bin/prometheus

Warning: Journal has been rotated since unit was started. Log output is incomplete or oldest than 10 minutes.
lines 1-10/10 (END)
```

CentOS Nodes:

Node 1:

```
[azamora_admin@CentOS ~]$ systemctl status prometheus
● prometheus.service - Prometheus
   Loaded: loaded (/etc/systemd/system/prometheus.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2024-10-28 08:04:04 PST; 52min ago
   Main PID: 5579 (prometheus)
     Tasks: 6
    CGroup: /system.slice/prometheus.service
            └─5579 /usr/local/bin/prometheus --config.file /etc/prometheus/prometheus...

Oct 28 08:04:04 CentOS prometheus[5579]: level=info ts=2024-10-28T00:04:04.218Z ca...y"
Oct 28 08:04:04 CentOS prometheus[5579]: level=info ts=2024-10-28T00:04:04.218Z ca...42µs
Oct 28 08:04:04 CentOS prometheus[5579]: level=info ts=2024-10-28T00:04:04.218Z ca...e"
Oct 28 08:04:04 CentOS prometheus[5579]: level=info ts=2024-10-28T00:04:04.218Z ca...=0
Oct 28 08:04:04 CentOS prometheus[5579]: level=info ts=2024-10-28T00:04:04.218Z ca...68µs
Oct 28 08:04:04 CentOS prometheus[5579]: level=info ts=2024-10-28T00:04:04.219Z ca...IC
Oct 28 08:04:04 CentOS prometheus[5579]: level=info ts=2024-10-28T00:04:04.219Z ca...d"
Oct 28 08:04:04 CentOS prometheus[5579]: level=info ts=2024-10-28T00:04:04.219Z ca...ml
Oct 28 08:04:04 CentOS prometheus[5579]: level=info ts=2024-10-28T00:04:04.219Z call...µs
Oct 28 08:04:04 CentOS prometheus[5579]: level=info ts=2024-10-28T00:04:04.219Z ca..."
Hint: Some lines were ellipsized, use -l to show in full.
[azamora_admin@CentOS ~]$
```

Node 2:

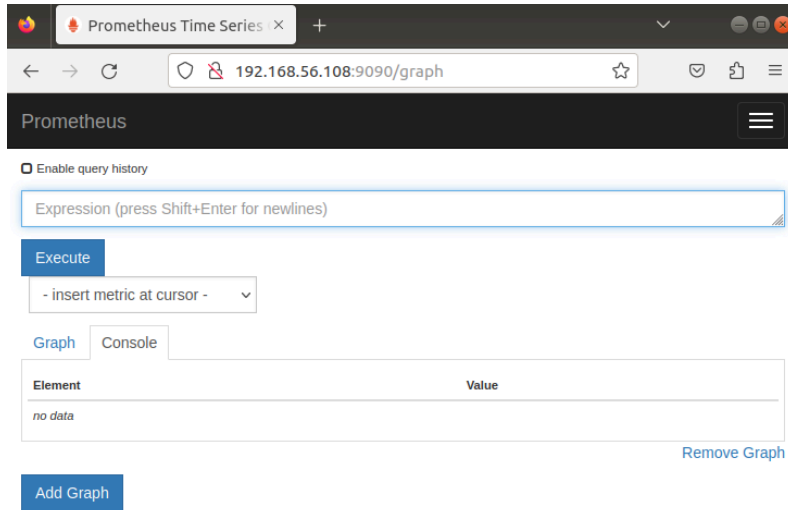
```
[azamora_admin@CentOS2 ~]$ sudo systemctl status prometheus
[sudo] password for azamora_admin:
● prometheus.service - Prometheus
   Loaded: loaded (/etc/systemd/system/prometheus.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2024-10-28 08:03:56 PST; 4min 9s ago
   Main PID: 4344 (prometheus)
     Tasks: 6
    CGroup: /system.slice/prometheus.service
            └─4344 /usr/local/bin/prometheus --config.file /etc/prometheus/prometheus...

Oct 28 08:03:56 CentOS2 prometheus[4344]: level=info ts=2024-10-28T00:03:56.862Z c...y"
Oct 28 08:03:56 CentOS2 prometheus[4344]: level=info ts=2024-10-28T00:03:56.862Z c...78µs
Oct 28 08:03:56 CentOS2 prometheus[4344]: level=info ts=2024-10-28T00:03:56.862Z c...e"
Oct 28 08:03:56 CentOS2 prometheus[4344]: level=info ts=2024-10-28T00:03:56.862Z c...=0
Oct 28 08:03:56 CentOS2 prometheus[4344]: level=info ts=2024-10-28T00:03:56.862Z c...78µs
Oct 28 08:03:56 CentOS2 prometheus[4344]: level=info ts=2024-10-28T00:03:56.862Z c...IC
Oct 28 08:03:56 CentOS2 prometheus[4344]: level=info ts=2024-10-28T00:03:56.862Z c...d"
Oct 28 08:03:56 CentOS2 prometheus[4344]: level=info ts=2024-10-28T00:03:56.862Z c...ml
Oct 28 08:03:56 CentOS2 prometheus[4344]: level=info ts=2024-10-28T00:03:56.863Z cal...µs
Oct 28 08:03:56 CentOS2 prometheus[4344]: level=info ts=2024-10-28T00:03:56.863Z c..."
Hint: Some lines were ellipsized, use -l to show in full.
[azamora_admin@CentOS2 ~]$
```

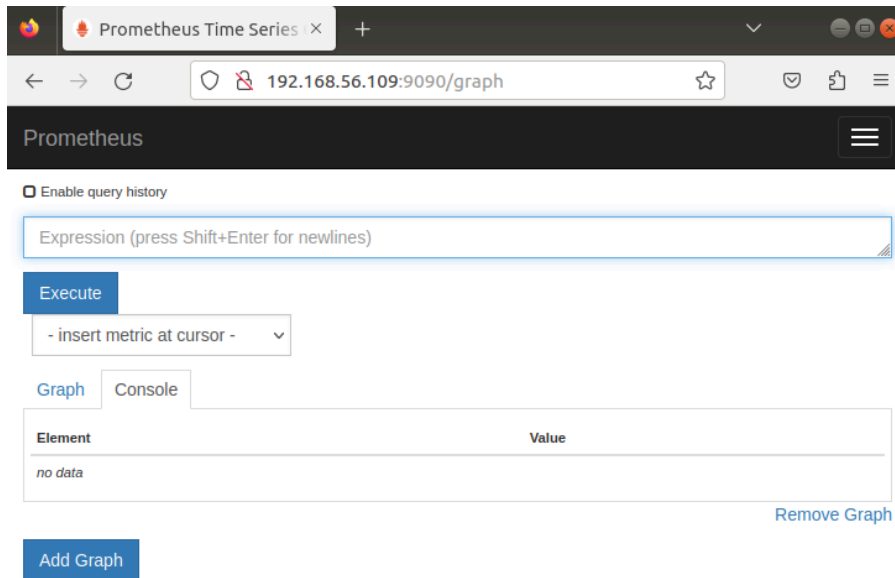
- Another way to check is by accessing the tool via browser, all you need to do is type ***"ip address":9090*** into your browser.

Ubuntu Nodes:

Server 1:



Server 2:



Server 3:

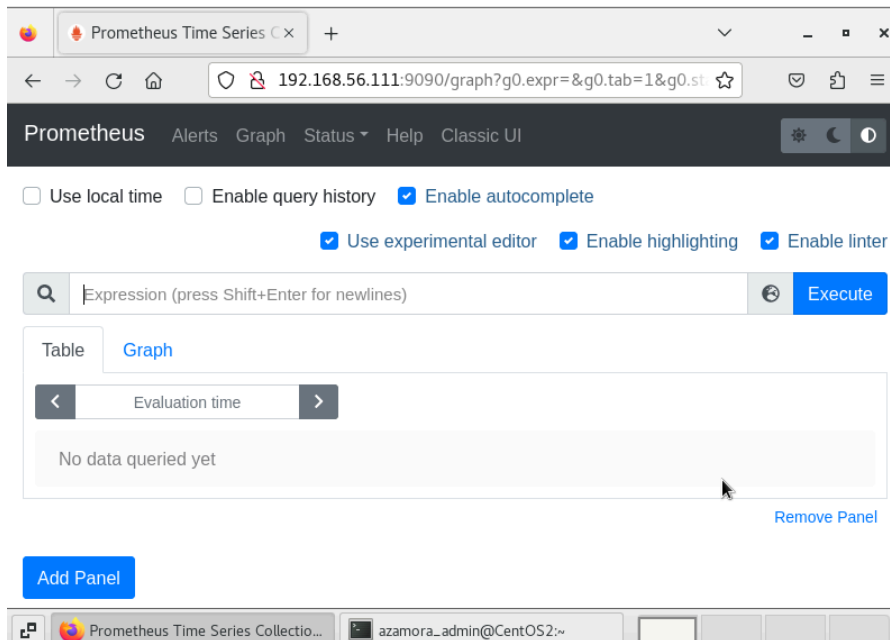
The screenshot shows the Prometheus Time Series web interface in a browser. The address bar displays the URL `192.168.56.112:9090/graph`. The interface includes a search bar with the placeholder text "Expression (press Shift+Enter for newlines)", an "Execute" button, and a dropdown menu currently showing "- insert metric at cursor -". Below the search bar are two tabs: "Graph" (selected) and "Console". The "Graph" tab displays a table with two columns, "Element" and "Value", and a single row containing the text "no data". To the right of the table is a "Remove Graph" link. At the bottom left of the interface is an "Add Graph" button.

CentOS Nodes

Node 1:

The screenshot shows the Prometheus Time Series web interface on Node 1. The address bar displays the URL `192.168.56.110:9090/graph?g0.expr=&g0.tab=1&g0.st`. The interface features a top navigation bar with links for "Prometheus", "Alerts", "Graph", "Status", "Help", and "Classic UI". Below the navigation bar are several configuration checkboxes: "Use local time" (unchecked), "Enable query history" (unchecked), "Enable autocomplete" (checked), "Use experimental editor" (checked), "Enable highlighting" (checked), and "Enable linter" (checked). The search bar contains the placeholder text "Expression (press Shift+Enter for newlines)" and an "Execute" button. Below the search bar are two tabs: "Table" and "Graph" (selected). The "Graph" tab displays a "No data queried yet" message. To the right of the graph area is a "Remove Panel" link. At the bottom left is an "Add Panel" button. The bottom of the screenshot shows a taskbar with several open applications, including a terminal window showing the user `azamora_admin@C...`.

Node 2:



- After installing make sure to git push your local repo to your github to save the files.
- Github Link:
https://github.com/GeloaceRT/TIP_HOA-9.1_ZAMORA_Angelo

Reflections:

Answer the following:

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

Numerous advantages come with using an availability monitoring solution for Ubuntu server administration, such as preemptive issue detection, real-time performance data, and timely notifications that reduce downtime. These technologies enhance capacity planning and resource usage by studying historical data, which also improves user experience. They offer a comprehensive solution for preserving server performance and dependability. They also facilitate compliance reporting and frequently connect easily with other administration tools.

Conclusions:

To summarize, I've discovered that performance monitoring is critical for analyzing resource use, and Prometheus stands out as an excellent solution because of its time series data storage and querying features. Prometheus allows me to collect and analyze timestamped statistics that provide useful insights into the metrics I'm tracking. The ability to generate derived time series from queries enables more in-depth examination of workload performance, which is critical for system efficiency.

To simplify the installation process, I use Ansible Playbook and roles, which make it easy to configure Prometheus in diverse contexts. This technique not only improves my ability to track performance indicators, but it also streamlines management and configuration. Finally, combining Prometheus with Ansible has enhanced my workflow, resulting in more reliable systems and optimal resource usage. With this I conclude that this activity helps me to understand the installation of monitoring tools via source URL and extraction as well checking the service status via local browser and system status. This fulfills the ILO's and the objective for this activity.