Name: Baltazar, Paul Eimar R,	**Date Performed:** Oct 21, 2024
Course/Section: CPE31S2	**Date Submitted:** Oct 30, 2024
Instructor: Engr. Robin Valenzuela	**Semester and SY:**

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

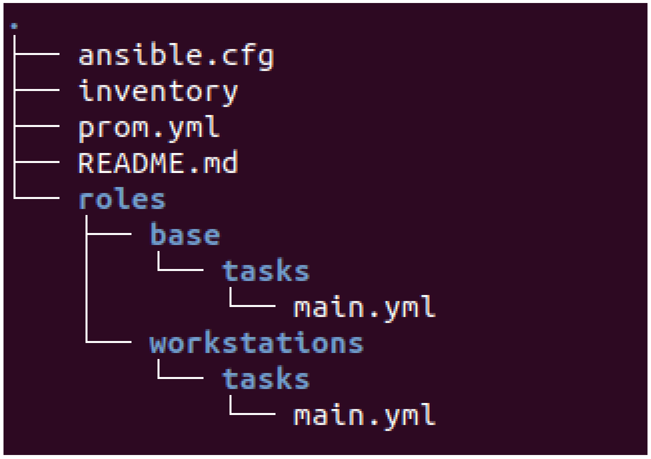
Cacti

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.

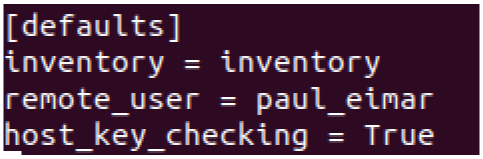
4. Output (screenshots and explanations)



```
.
├── ansible.cfg
├── inventory
├── prom.yml
├── README.md
└── roles
    ├── base
    │   └── tasks
    │       └── main.yml
    └── workstations
        └── tasks
            └── main.yml
```

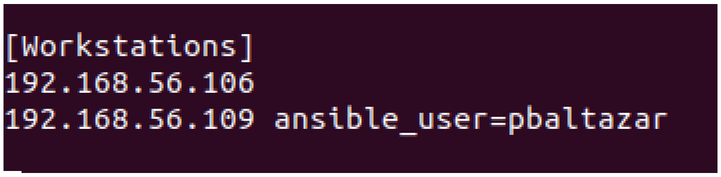
Create a directory which contains the following: ansible.cfg file, inventory file, yml file, and roles folder.

The roles folder should contain a base task, and also a task for the installation process.



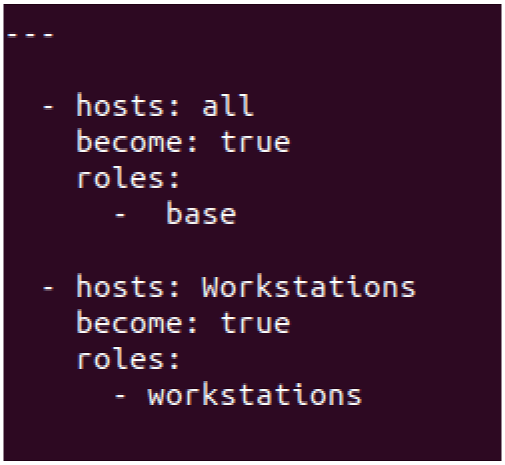
```
[defaults]
inventory = inventory
remote_user = paul_eimar
host_key_checking = True
```

The ansible.cfg should contain these lines. Locate the path of your inventory file and place it in the inventory section. In this activity, I have my inventory file in the same directory as the ansible.cfg file.



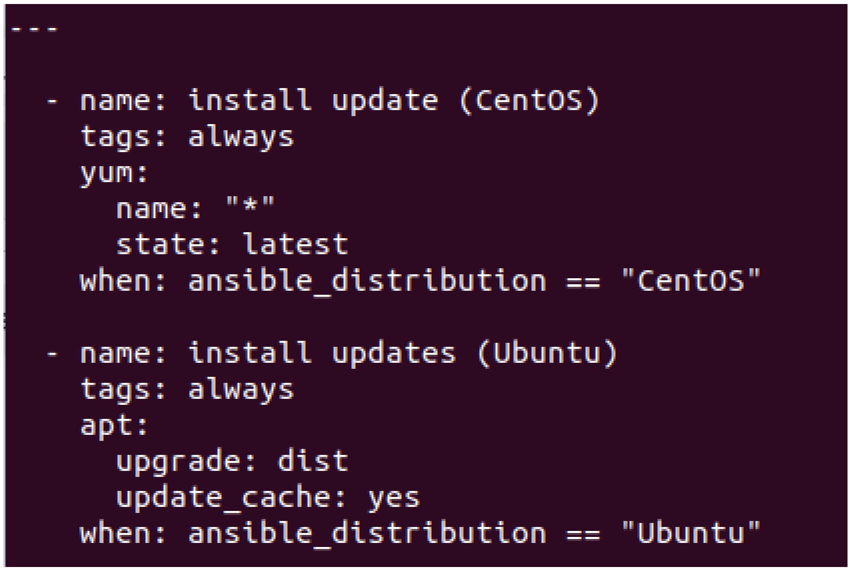
```
[Workstations]
192.168.56.106
192.168.56.109 ansible_user=pbaltazar
```

This is the content of the inventory file. It must point to your remote servers. The *ansible_user=username* is added for CentOS servers.



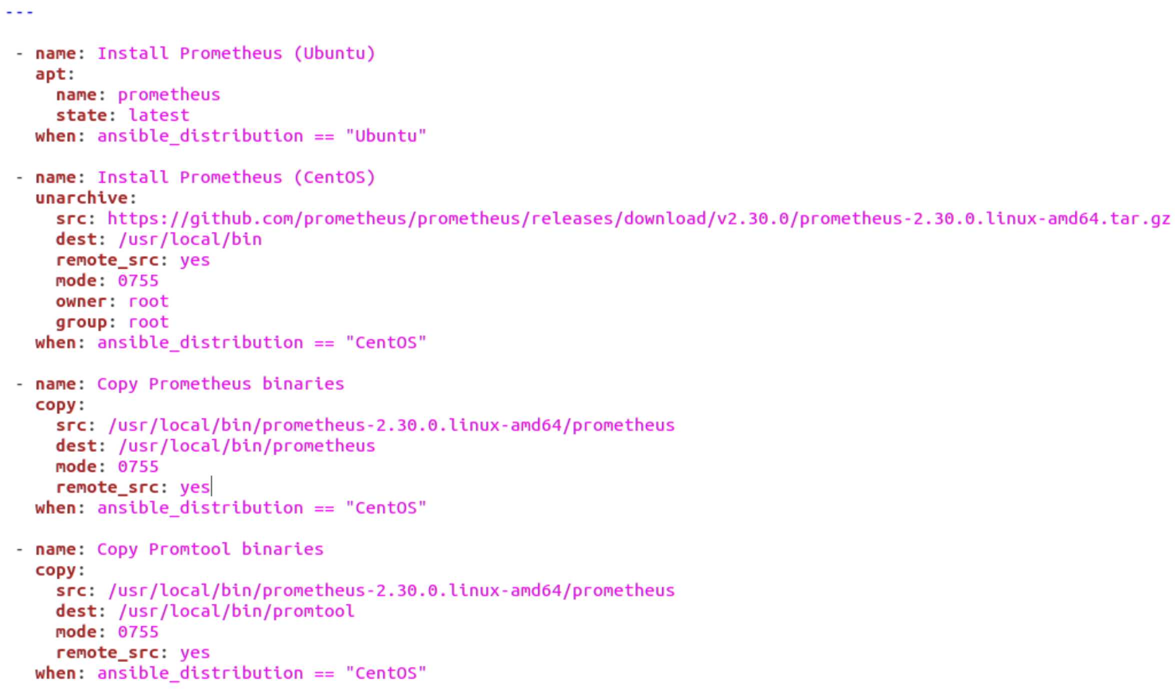
```
---

  - hosts: all
    become: true
    roles:
      -  base

  - hosts: Workstations
    become: true
    roles:
      - workstations
```

This is the content of the main playbook, in this case, the prom.yml. This yml's content specifies which yml files in the roles folder are to be executed.

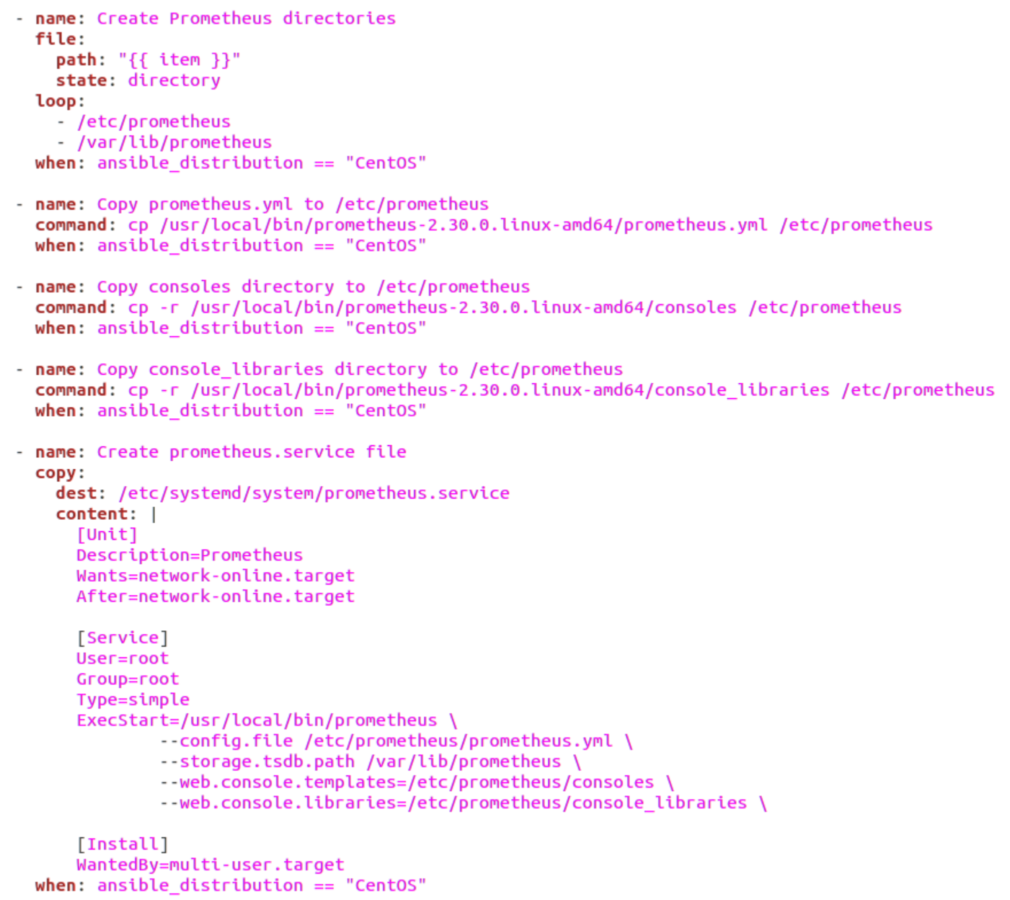


```
---

  - name: install update (CentOS)
    tags: always
    yum:
      name: "*"
      state: latest
    when: ansible_distribution == "CentOS"

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

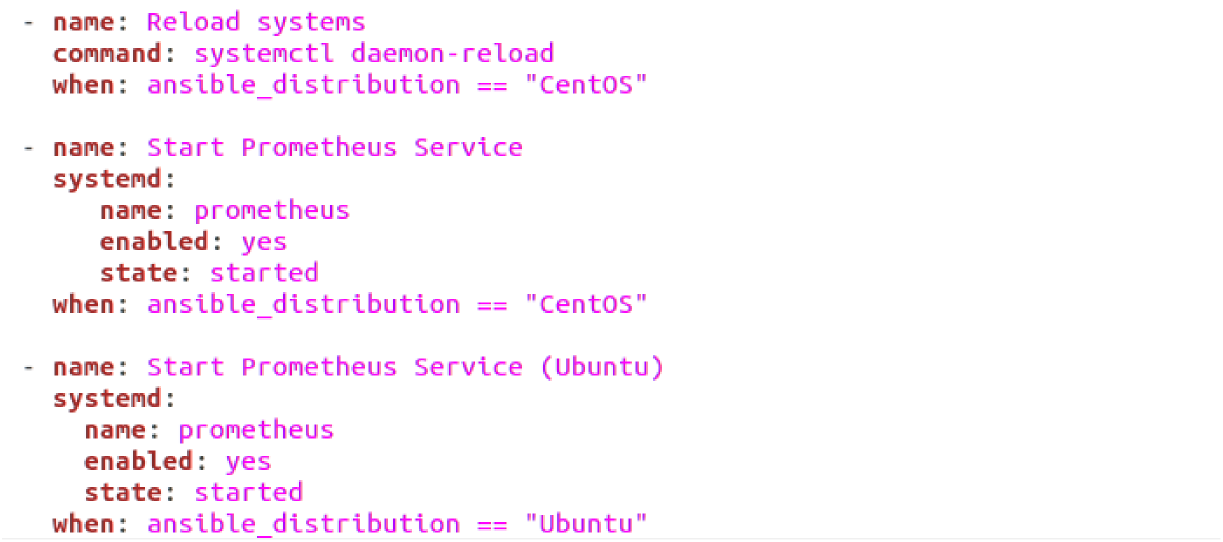
This is the main.yml in the roles/base/tasks folder. In this activity, the base role's task is only to update both remote servers.



```
---

 - 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"
```



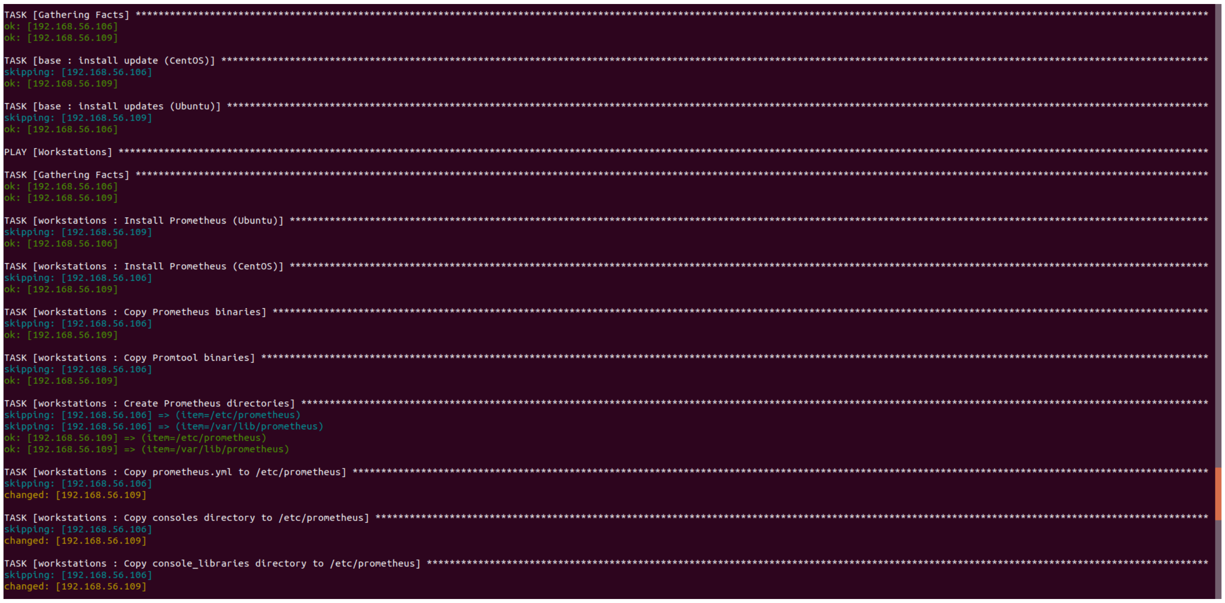
```
- name: Create Prometheus directories
  file:
    path: "{{ item }}"
    state: directory
  loop:
    - /etc/prometheus
    - /var/lib/prometheus
  when: ansible_distribution == "CentOS"

- 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"
```



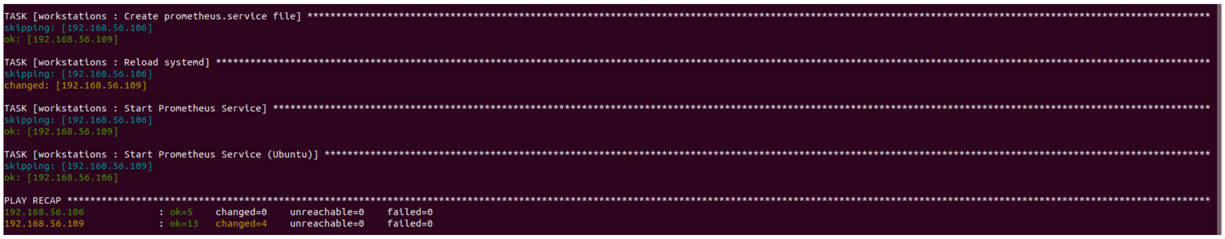
```
- name: Reload systems
  command: systemctl daemon-reload
  when: ansible_distribution == "CentOS"

- 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"
```

This is the main.yml file under roles/workstations/task directory. This is the commands to be executed in order to install and run prometheus on both Ubuntu and CentOS servers. This yml file also includes the required steps to setup prometheus.

 In order to execute the playbook. Type the command *-ansible-playbook --ask-become-pass prom.yml. The terminal should look like this:*

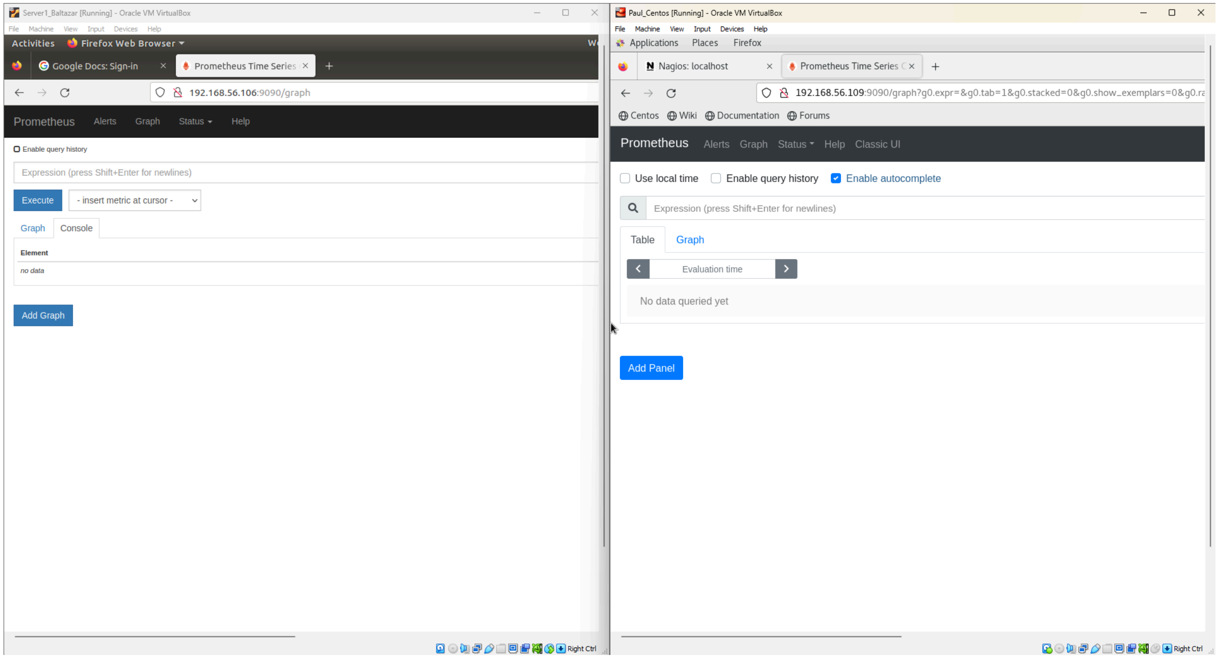


```
TASK [Gathering Facts] ************************************************************************
ok: [192.168.56.106]
ok: [192.168.56.109]

TASK [base : install update (CentOS)] *********************************************************
skipping: [192.168.56.106]
ok: [192.168.56.109]

TASK [base : install updates (Ubuntu)] ********************************************************
skipping: [192.168.56.109]
ok: [192.168.56.106]

PLAY [Workstations] ***************************************************************************

TASK [Gathering Facts] ************************************************************************
ok: [192.168.56.106]
ok: [192.168.56.109]

TASK [workstations : Install Prometheus (Ubuntu)] *********************************************
skipping: [192.168.56.109]
ok: [192.168.56.106]

TASK [workstations : Install Prometheus (CentOS)] *********************************************
skipping: [192.168.56.106]
ok: [192.168.56.109]

TASK [workstations : Copy Prometheus binaries] ************************************************
skipping: [192.168.56.106]
ok: [192.168.56.109]

TASK [workstations : Copy Promtool binaries] **************************************************
skipping: [192.168.56.106]
ok: [192.168.56.109]

TASK [workstations : Create Prometheus directories] *******************************************
skipping: [192.168.56.106] => (item=/etc/prometheus)
skipping: [192.168.56.106] => (item=/var/lib/prometheus)
ok: [192.168.56.109] => (item=/etc/prometheus)
ok: [192.168.56.109] => (item=/var/lib/prometheus)

TASK [workstations : Copy prometheus.yml to /etc/prometheus] **********************************
skipping: [192.168.56.106]
changed: [192.168.56.109]

TASK [workstations : Copy consoles directory to /etc/prometheus] ******************************
skipping: [192.168.56.106]
changed: [192.168.56.109]

TASK [workstations : Copy console_libraries directory to /etc/prometheus] *********************
skipping: [192.168.56.106]
changed: [192.168.56.109]
```



```
TASK [workstations : Create prometheus.service file] ******************************************
skipping: [192.168.56.106]
ok: [192.168.56.109]

TASK [workstations : Reload systemd] **********************************************************
skipping: [192.168.56.106]
changed: [192.168.56.109]

TASK [workstations : Start Prometheus Service] ************************************************
skipping: [192.168.56.106]
ok: [192.168.56.109]

TASK [workstations : Start Prometheus Service (Ubuntu)] ***************************************
skipping: [192.168.56.109]
ok: [192.168.56.106]

PLAY RECAP ************************************************************************************
192.168.56.106             : ok=5    changed=0    unreachable=0    failed=0
192.168.56.109             : ok=13   changed=4    unreachable=0    failed=0
```

After running the playbook. You can confirm that the installation is successful by opening up a web browser and going to [IP ADDRESS]:9090. In this case, its 192.168.56.106:9090 and 192.168.56.109:9090.


Server1_Baltazar [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Firefox Web Browser
Google Docs: Sign-in
Prometheus Time Series
192.168.56.106:9090/graph
Prometheus Alerts Graph Status Help
Enable query history
Expression (press Shift+Enter for newlines)
Execute
- insert metric at cursor -
Graph Console
Element
no data
Add Graph
Right Ctrl
Paul_Centos [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Firefox
Nagios: localhost
Prometheus Time Series
192.168.56.109:9090/graph?g0.expr=&g0.tab=1&g0.stacked=0&g0.show_exemplars=0&g0.ra
Centos Wiki Documentation Forums
Prometheus Alerts Graph Status Help Classic UI
Use local time
Enable query history
Enable autocomplete
Expression (press Shift+Enter for newlines)
Table Graph
Evaluation time
No data queried yet
Add Panel
Right Ctrl

Reflections:

Answer the following:

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

– Performance monitoring tools like Prometheus help track and analyze the health and efficiency of systems and applications. Performance monitoring tools provide real-time insights into resource usage, response times, and error rates. This can helpp identify bottlenecks, detect issues before they get worse, and optimize performance. Monitoring tools also support setting up alerts so that teams can respond quickly to incidents, which can reduce downtime and improve reliability.

Conclusions:

In this activity, I have learned to create an ansible workflow to install a performance monitoring tool to a remote server.