Name: Recto, Jon Jeous J**.**	**Date Performed:** **Mar 25, 2024**
Course/Section: CPE31S1	**Date Submitted:** **Mar 26, 2024**
Instructor: Dr. Taylar	**Semester and SY:** 2nd Sem 23-24

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

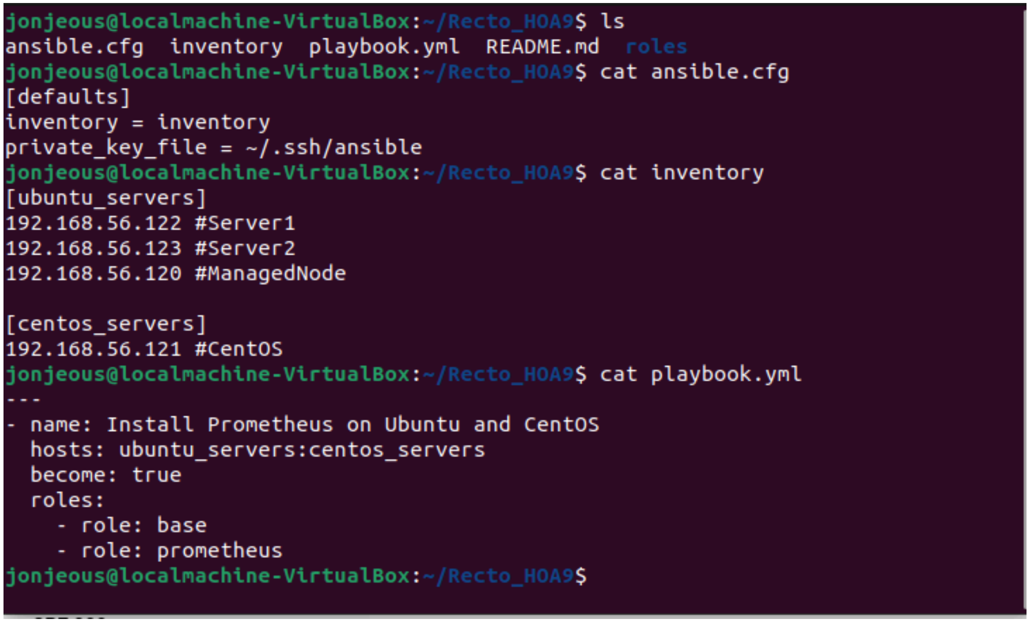
Cacti is a complete network graphing solution designed to harness the power of RRDTool's data storage and graphing functionality. Cacti provides a fast poller, advanced graph templating, multiple data acquisition methods, and user management features out of the box. All of this is wrapped in an intuitive, 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.

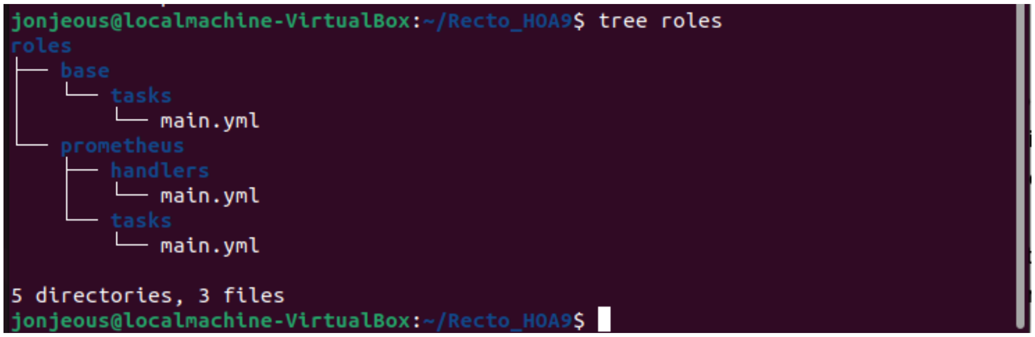
4. Output (screenshots and explanations)

a. Add the necessary files and its contents



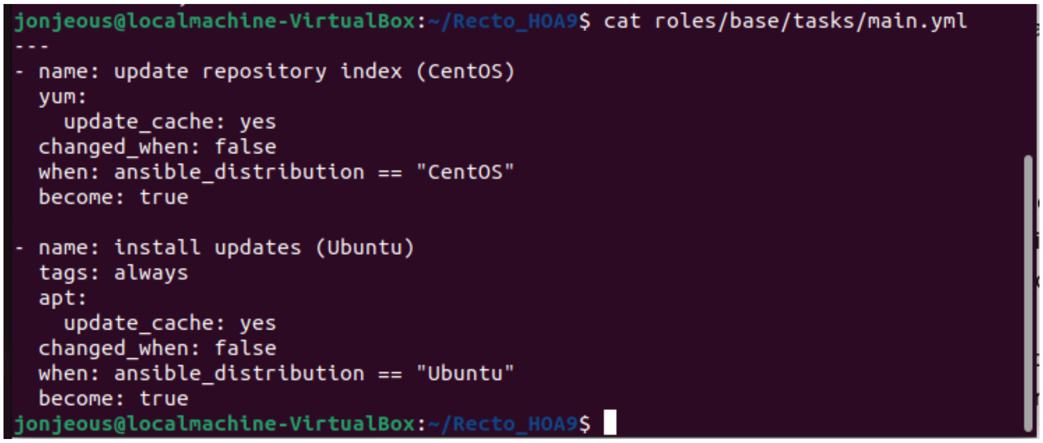
```
jonjeous@localmachine-VirtualBox:~/Recto_HOA9$ ls
ansible.cfg  inventory  playbook.yml  README.md  roles
jonjeous@localmachine-VirtualBox:~/Recto_HOA9$ cat ansible.cfg
[defaults]
inventory = inventory
private_key_file = ~/.ssh/ansible
jonjeous@localmachine-VirtualBox:~/Recto_HOA9$ cat inventory
[ubuntu_servers]
192.168.56.122 #Server1
192.168.56.123 #Server2
192.168.56.120 #ManagedNode

[centos_servers]
192.168.56.121 #CentOS
jonjeous@localmachine-VirtualBox:~/Recto_HOA9$ cat playbook.yml
---
- name: Install Prometheus on Ubuntu and CentOS
  hosts: ubuntu_servers:centos_servers
  become: true
  roles:
    - role: base
    - role: prometheus
jonjeous@localmachine-VirtualBox:~/Recto_HOA9$
```

b. Roles

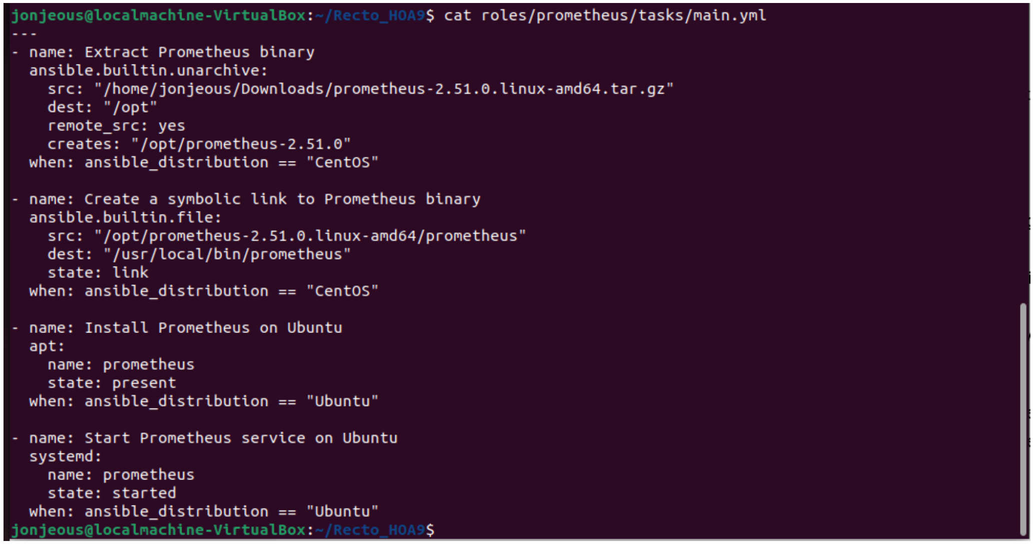


```
jonjeous@localmachine-VirtualBox:~/Recto_HOA9$ tree roles
roles
├── base
│   └── tasks
│       └── main.yml
└── prometheus
    ├── handlers
    │   └── main.yml
    └── tasks
        └── main.yml

5 directories, 3 files
jonjeous@localmachine-VirtualBox:~/Recto_HOA9$ 
```

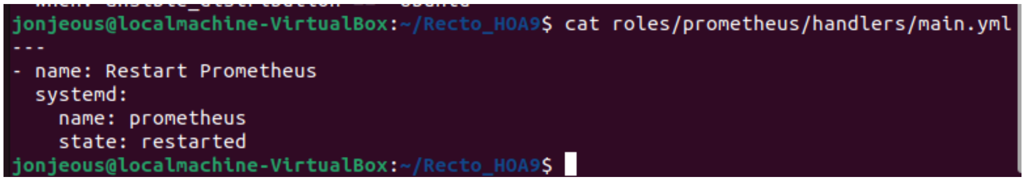
c. Tasks for each role.



```
jonjeous@localmachine-VirtualBox:~/Recto_HOA9$ cat roles/base/tasks/main.yml
---
- name: update repository index (CentOS)
  yum:
    update_cache: yes
  changed_when: false
  when: ansible_distribution == "CentOS"
  become: true

- name: install updates (Ubuntu)
  tags: always
  apt:
    update_cache: yes
  changed_when: false
  when: ansible_distribution == "Ubuntu"
  become: true
jonjeous@localmachine-VirtualBox:~/Recto_HOA9$ 
```

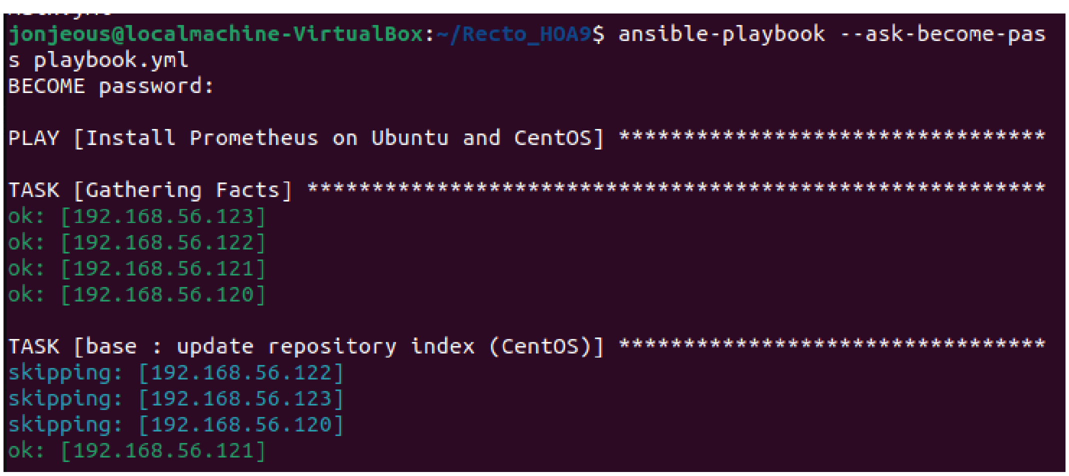


```
jonjeous@localmachine-VirtualBox:~/Recto_HOA9$ cat roles/prometheus/tasks/main.yml
---
- name: Extract Prometheus binary
  ansible.builtin.unarchive:
    src: "/home/jonjeous/Downloads/prometheus-2.51.0.linux-amd64.tar.gz"
    dest: "/opt"
    remote_src: yes
    creates: "/opt/prometheus-2.51.0"
  when: ansible_distribution == "CentOS"

- name: Create a symbolic link to Prometheus binary
  ansible.builtin.file:
    src: "/opt/prometheus-2.51.0.linux-amd64/prometheus"
    dest: "/usr/local/bin/prometheus"
    state: link
  when: ansible_distribution == "CentOS"

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

- name: Start Prometheus service on Ubuntu
  systemd:
    name: prometheus
    state: started
  when: ansible_distribution == "Ubuntu"
jonjeous@localmachine-VirtualBox:~/Recto_HOA9$
```

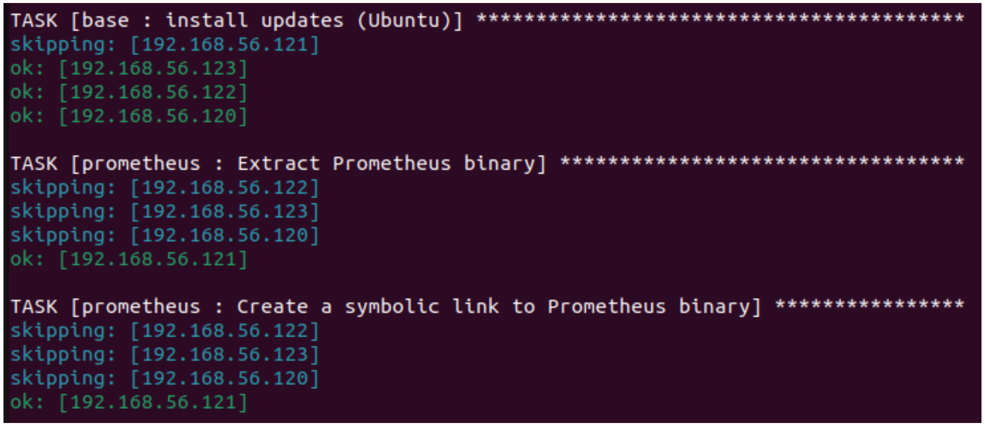


```
jonjeous@localmachine-VirtualBox:~/Recto_HOA9$ cat roles/prometheus/handlers/main.yml
---
- name: Restart Prometheus
  systemd:
    name: prometheus
    state: restarted
jonjeous@localmachine-VirtualBox:~/Recto_HOA9$
```

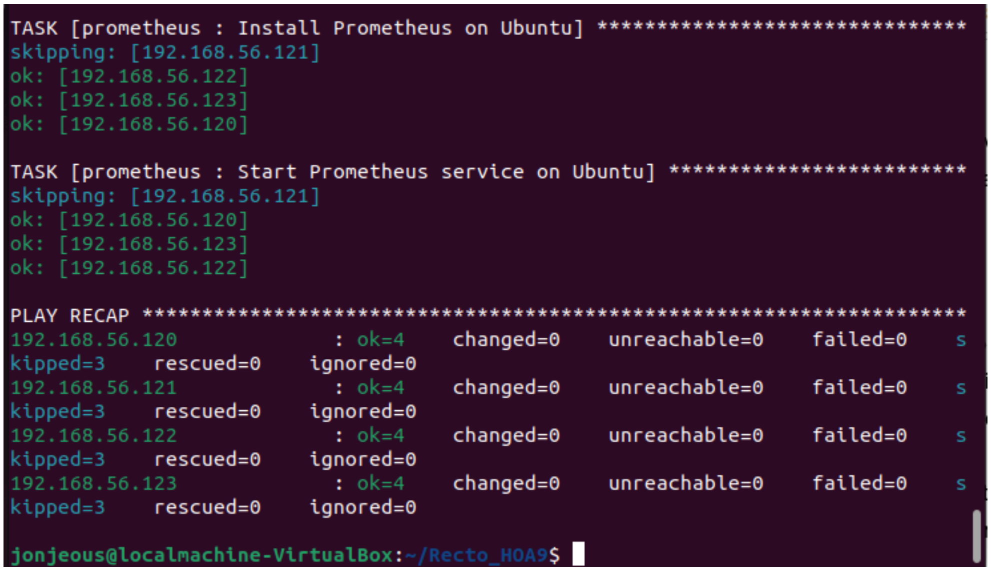
d. Run the playbook.



```
jonjeous@localmachine-VirtualBox:~/Recto_HOA9$ ansible-playbook --ask-become-pas
s playbook.yml
BECOME password:

PLAY [Install Prometheus on Ubuntu and CentOS] *********************************

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.123]
ok: [192.168.56.122]
ok: [192.168.56.121]
ok: [192.168.56.120]

TASK [base : update repository index (CentOS)] *********************************
skipping: [192.168.56.122]
skipping: [192.168.56.123]
skipping: [192.168.56.120]
ok: [192.168.56.121]
```

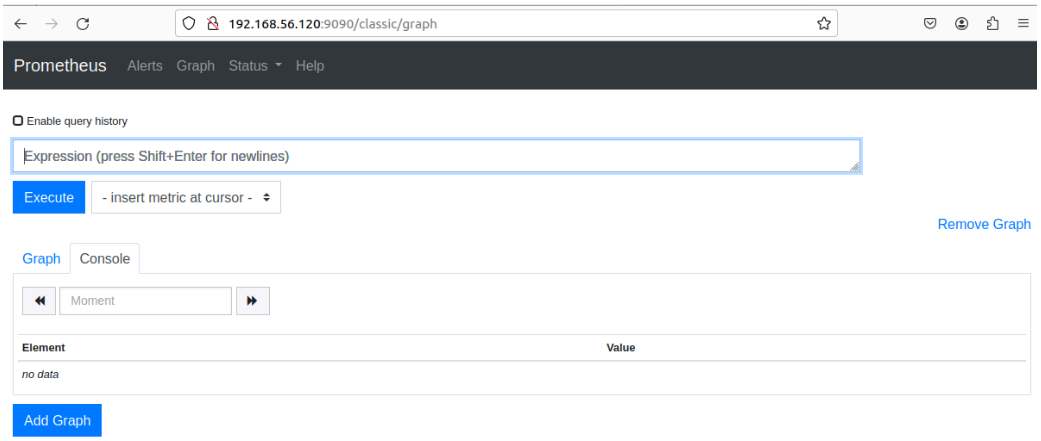


```
TASK [base : install updates (Ubuntu)] *****************************************
skipping: [192.168.56.121]
ok: [192.168.56.123]
ok: [192.168.56.122]
ok: [192.168.56.120]

TASK [prometheus : Extract Prometheus binary] **********************************
skipping: [192.168.56.122]
skipping: [192.168.56.123]
skipping: [192.168.56.120]
ok: [192.168.56.121]

TASK [prometheus : Create a symbolic link to Prometheus binary] ****************
skipping: [192.168.56.122]
skipping: [192.168.56.123]
skipping: [192.168.56.120]
ok: [192.168.56.121]
```



```
TASK [prometheus : Install Prometheus on Ubuntu] *******************************
skipping: [192.168.56.121]
ok: [192.168.56.122]
ok: [192.168.56.123]
ok: [192.168.56.120]

TASK [prometheus : Start Prometheus service on Ubuntu] *************************
skipping: [192.168.56.121]
ok: [192.168.56.120]
ok: [192.168.56.123]
ok: [192.168.56.122]

PLAY RECAP *********************************************************************
192.168.56.120             : ok=4    changed=0    unreachable=0    failed=0    s
kipped=3    rescued=0    ignored=0
192.168.56.121             : ok=4    changed=0    unreachable=0    failed=0    s
kipped=3    rescued=0    ignored=0
192.168.56.122             : ok=4    changed=0    unreachable=0    failed=0    s
kipped=3    rescued=0    ignored=0
192.168.56.123             : ok=4    changed=0    unreachable=0    failed=0    s
kipped=3    rescued=0    ignored=0

jonjeous@localmachine-VirtualBox:~/Recto_HOA9$
```

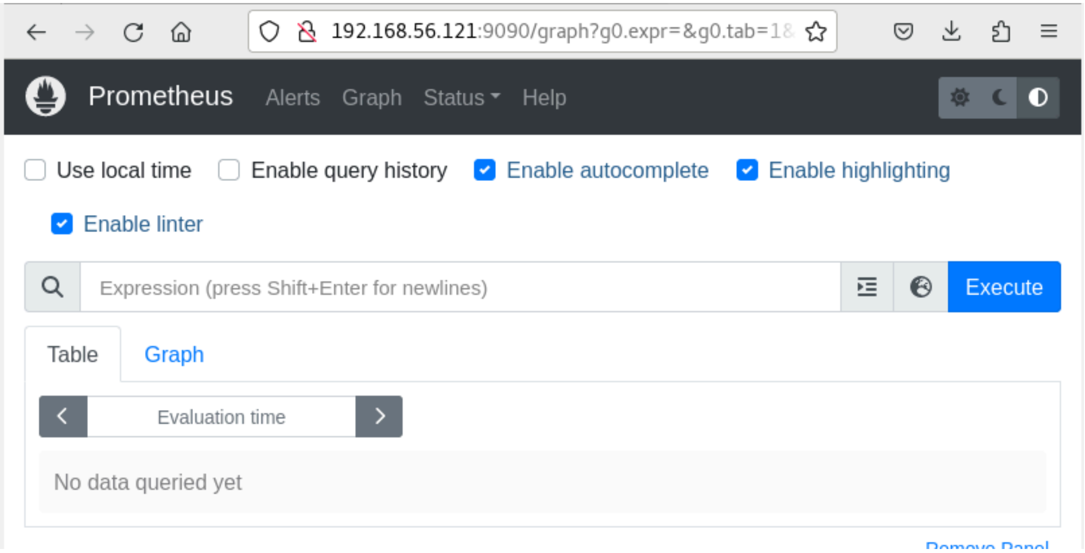
e. Check if it is installed properly



192.168.56.120:9090/classic/graph

Element	Value
no data	

UBUNTU



192.168.56.121:9090/graph?g0.expr=&g0.tab=1&
Prometheus
Alerts
Graph
Status
Help

☐ Use local time ☐ Enable query history ☑ Enable autocomplete ☑ Enable highlighting

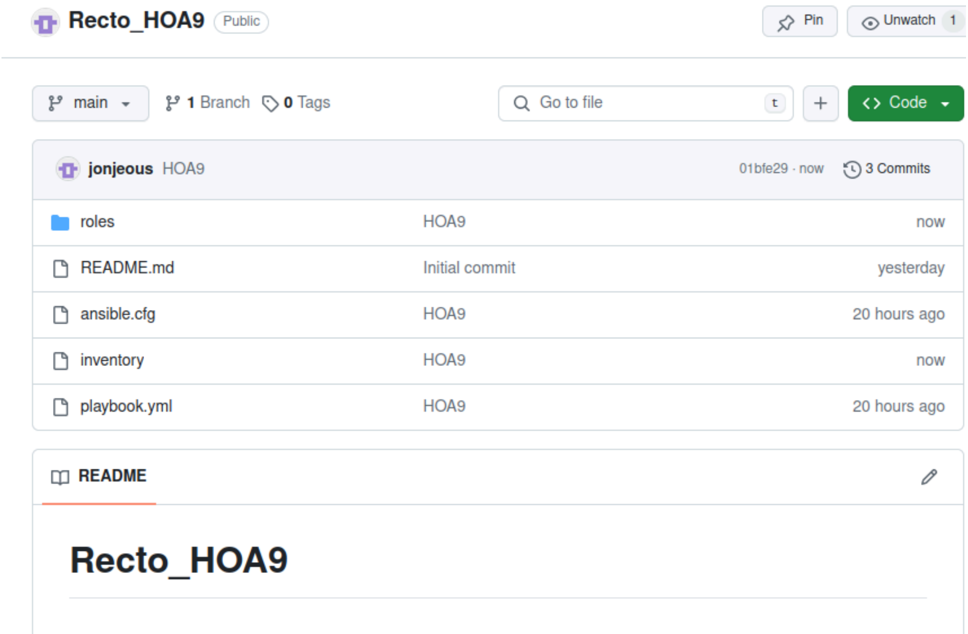
☑ Enable linter

Expression (press Shift+Enter for newlines)
Execute
Table
Graph
Evaluation time

No data queried yet

CENTOS

f. Commit changes to github



jonjeous HOA9 01bfe29 · now 3 Commits

roles	HOA9	now
README.md	Initial commit	yesterday
ansible.cfg	HOA9	20 hours ago
inventory	HOA9	now
playbook.yml	HOA9	20 hours ago

README

Recto_HOA9

https://github.com/jonjeous/Recto_HOA9.git

Reflections:

Answer the following:

1. What are the benefits of having a performance monitoring tool?
    - Performance monitoring tools provide valuable insights into the health and efficiency of computer systems and applications. They help identify issues early, optimize resource usage, and plan for future needs. These tools also assist in troubleshooting problems, ensuring systems meet performance standards, and enhancing security. Ultimately, they contribute to better user experiences and overall business success.

Conclusions:

In conclusion, when it comes to setting up and managing enterprise performance monitoring tools, using Ansible as an Infrastructure as Code (IaC) tool simplifies the process. Among the various options, I find Prometheus particularly notable. It stores data as time series, which means it organizes information by time and labels. With Ansible's automation features, I can easily install, configure, and manage Prometheus, along with other performance monitoring tools. This allows me to keep track of resource usage and ensure our systems run smoothly.