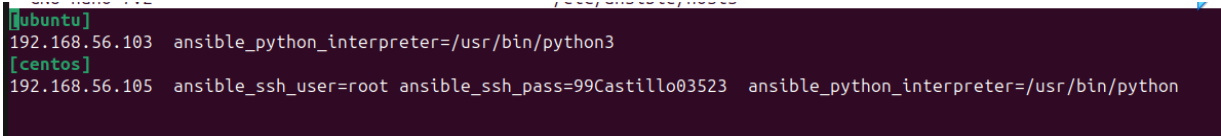| Name: Castillo, Joshua L.	Date Performed: 3/30/2024
Course/Section: CPE31S1	Date Submitted: 3/30/2024
Instructor: Dr. Jonathan Tylar	Semester and SY:

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)

inventory:


```
[ubuntu]
192.168.56.103  ansible_python_interpreter=/usr/bin/python3
[centos]
192.168.56.105  ansible_ssh_user=root ansible_ssh_pass=99Castillo03523  ansible_python_interpreter=/usr/bin/python
```

ansible.cfg:

```
[defaults]
inventory = /etc/ansible/hosts
host_key_checking = False

decreation_warnings = False


remote_user = joshua
private_ket_file = ~/.ssh/
```

i created directories tree consisting of this:

```
joshua@ManagedNode:~/HOA9$ cd roles
joshua@ManagedNode:~/HOA9/roles$ tree
.
├── centos
│   └── tasks
│       └── main.yml
└── ubuntu
    └── tasks
        └── main.yml

5 directories, 2 files
joshua@ManagedNode:~/HOA9/roles$
```

after creating the sub directories shown above, I created commands for main.yml for each role.

install_prometheus.yml

```yaml
---

- hosts: all
  become: true
  pre_tasks:

  - name: DPKG in Ubuntu Server
    shell: |
      dpkg --configure -a
    when: ansible_distribution == "Ubuntu"

  - name: Update in Ubuntu Server
    apt:
      update_cache: yes
      upgrade: yes
    when: ansible_distribution == "Ubuntu"

  - name: Install epel-release and dnf
    dnf:
      name:
        - epel-release
        - dnf
    when: ansible_distribution == "CentOS"

  - name: Update CentOS Server
    dnf:
      update_cache: yes
      name: "*"
      state: latest
    when: ansible_distribution == "CentOS"

- hosts: ubuntu
  become: true
  roles:
    - ubuntu

- hosts: centos
  become: true
  roles:
    - centos
```

roles/ubuntu/tasks/main.yml:

```yaml
- name: Create directory
  tags: dir
  file:
    path: ~/prometheus
    state: directory

- name: Download Prometheus
  unarchive:
    src: https://github.com/prometheus/prometheus/releases/download/v2.39.1/prometheus-2.39.1.linux-amd64.tar.gz
    dest: ~/prometheus
    remote_src: yes
    mode: 0777
    owner: root
    group: root

- name: Add executables of Prometheus in a path
  tags: exe
  shell: |
    cd ~/prometheus/prometheus*
    sudo rsync -av . /usr/local/bin/prometheus/

- name: Copy Prometheus file
  tags: file
  copy:
    src: prometheus.service
    dest: /etc/systemd/system/
    owner: root
    group: root
    mode: 777

- name: Start HTTPD
  tags: httpd
  service:
    name: prometheus
    state: restarted
    enabled: true
```

roles/centos/tasks/main.yml:

```yaml
- name: Create directory
  tags: dir
  file:
    path: ~/prometheus
    state: directory

- name: Download Prometheus
  unarchive:
    src: https://github.com/prometheus/prometheus/releases/download/v2.39.1/prometheus-2.39.1.linux-amd64.tar.gz
    dest: ~/prometheus
    remote_src: yes
    mode: 0777
    owner: root
    group: root

- name: Add executables of Prometheus in a path
  tags: exe
  shell: |
    cd ~/prometheus*
    sudo rsync -av . /usr/local/bin/prometheus/

- name: Copy Prometheus file
  tags: file
  copy:
    src: prometheus.service
    dest: /etc/systemd/system/
    owner: root
    group: root
    mode: 777

- name: Start HTTPD
  tags: httpd
  service:
    name: prometheus
    state: restarted
    enabled: true
```

Output:

```
joshua@ManagedNode:~/HOA9$ ansible-playbook --ask-become-pass install_prometheus.yml
BECOME password:

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

TASK [Gathering Facts] **************************************************************************
ok: [192.168.56.103]
ok: [192.168.56.105]

TASK [DPKG in Ubuntu Server] ********************************************************************
skipping: [192.168.56.105]
changed: [192.168.56.103]

TASK [Update in Ubuntu Server] ******************************************************************
skipping: [192.168.56.105]
ok: [192.168.56.103]

TASK [Install epel-release and dnf] ***********************************************************
skipping: [192.168.56.103]
ok: [192.168.56.105]

TASK [Update CentOS Server] *********************************************************************
skipping: [192.168.56.103]
ok: [192.168.56.105]

PLAY [ubuntu] ***********************************************************************************

TASK [Gathering Facts] **************************************************************************
ok: [192.168.56.103]

TASK [ubuntu : Create directory] **************************************************************
ok: [192.168.56.103]

TASK [ubuntu : Download Prometheus] ***********************************************************
```

```
TASK [ubuntu : Add executables of Prometheus in a path] ***************************************
changed: [192.168.56.103]

TASK [ubuntu : Copy Prometheus file] **********************************************************
ok: [192.168.56.103]

TASK [ubuntu : Start HTTPD] *******************************************************************
changed: [192.168.56.103]

PLAY [centos] **********************************************************************************

TASK [Gathering Facts] *************************************************************************
ok: [192.168.56.105]

TASK [centos : Create directory] *************************************************************
ok: [192.168.56.105]

TASK [centos : Download Prometheus] **********************************************************
ok: [192.168.56.105]

TASK [centos : Add executables of Prometheus in a path] **************************************
changed: [192.168.56.105]

TASK [centos : Copy Prometheus file] *********************************************************
ok: [192.168.56.105]

TASK [centos : Start HTTPD] ******************************************************************
changed: [192.168.56.105]

PLAY RECAP ***********************************************************************************
192.168.56.103             : ok=9    changed=3    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.105             : ok=9    changed=2    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0

joshua@ManagedNode:~/HOA9$ nano /etc/ansible/hosts
```

Explanation:

Explanation: The provided Ansible playbook executed with the following results:

● Two hosts were involved in the playbook execution, with IP addresses 192.168.56.103 (Ubuntu) and 192.168.56.105 (CentOS).

● The "Gathering Facts" task collected information about both hosts, which completed successfully.

● The summary at the end of the playbook execution shows that for the host 192.168.56.103, seven tasks were successful (ok), two tasks resulted in changes (changed), and there were no failures, unreachable hosts, or skipped tasks.

● For the host 192.168.56.105, seven task was successful (ok), no tasks resulted in changes (changed=0), two tasks were skipped, and there were no failures, unreachable hosts, or other issues.

Verify prometheus configuration:

192.168.56.103:9090
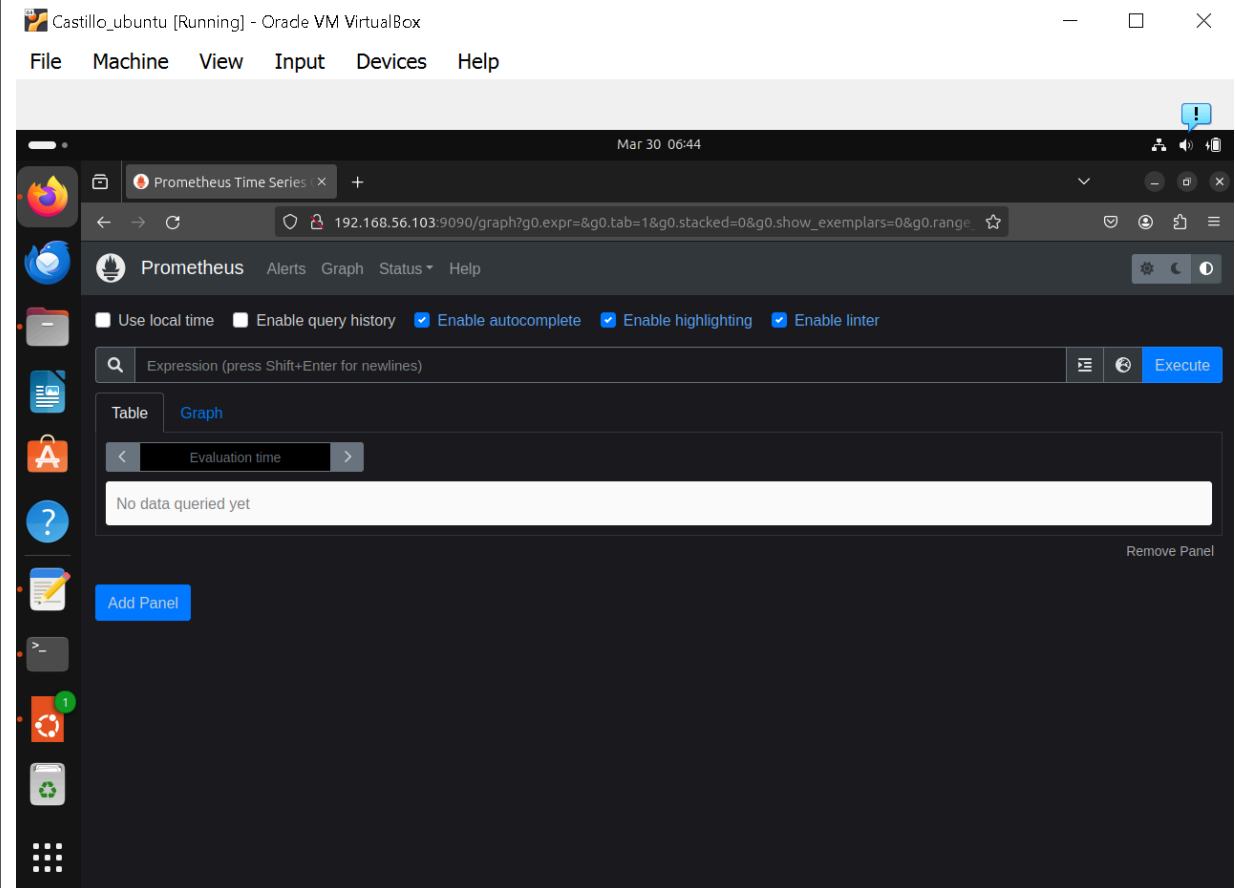
192.168.56.105:9090
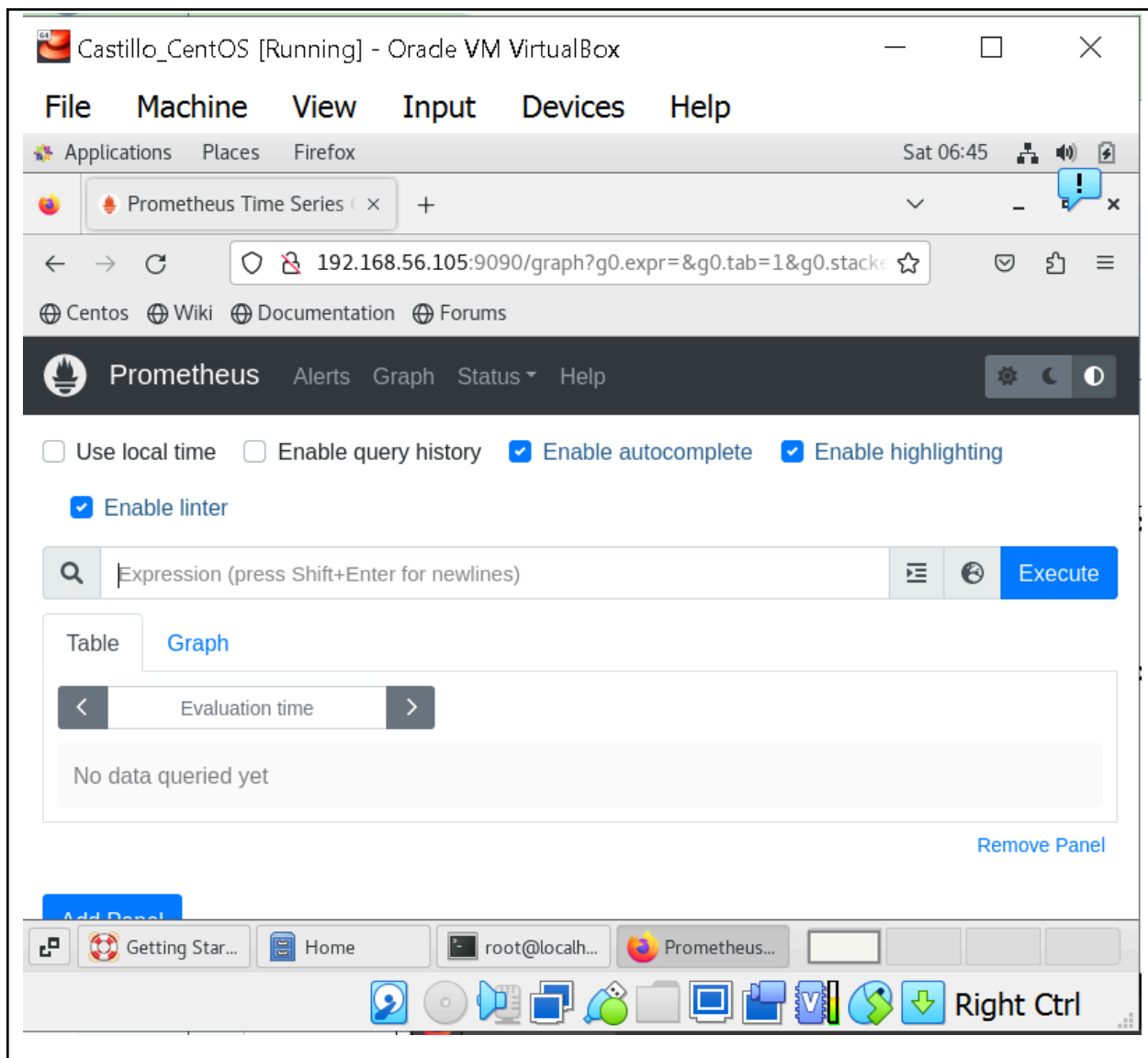
add the configurations to github repository:

```
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   files/prometheus.service
        new file:   install_prometheus.yml
        new file:   roles/centos/tasks/main.yml
        new file:   roles/ubuntu/tasks/main.yml

joshua@ManagedNode:~/HOA9$ git push
error: src refspec refs/heads/main does not match any
error: failed to push some refs to 'github.com:qjlcastillo/HOA9.git'
joshua@ManagedNode:~/HOA9$ git commit -m "HOA9"
[main (root-commit) 814b3fe] HOA9
 4 files changed, 123 insertions(+)
 create mode 100644 files/prometheus.service
 create mode 100644 install_prometheus.yml
 create mode 100644 roles/centos/tasks/main.yml
 create mode 100644 roles/ubuntu/tasks/main.yml
joshua@ManagedNode:~/HOA9$ git push
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 2 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (12/12), 1.36 KiB | 1.36 MiB/s, done.
Total 12 (delta 1), reused 0 (delta 0), pack-reused 0
```

<> **Code**   ⊙ Issues   ⣿ Pull requests

⊡ **Files**

ᛘ main ▾   +   🔍

🔍 Go to file   t

∨ 📁 files
  📄 prometheus.service
∨ 📁 roles
  ∨ 📁 centos/tasks
    📄 main.yml
  ∨ 📁 ubuntu/tasks
    📄 main.yml
  📄 install_prometheus.yml
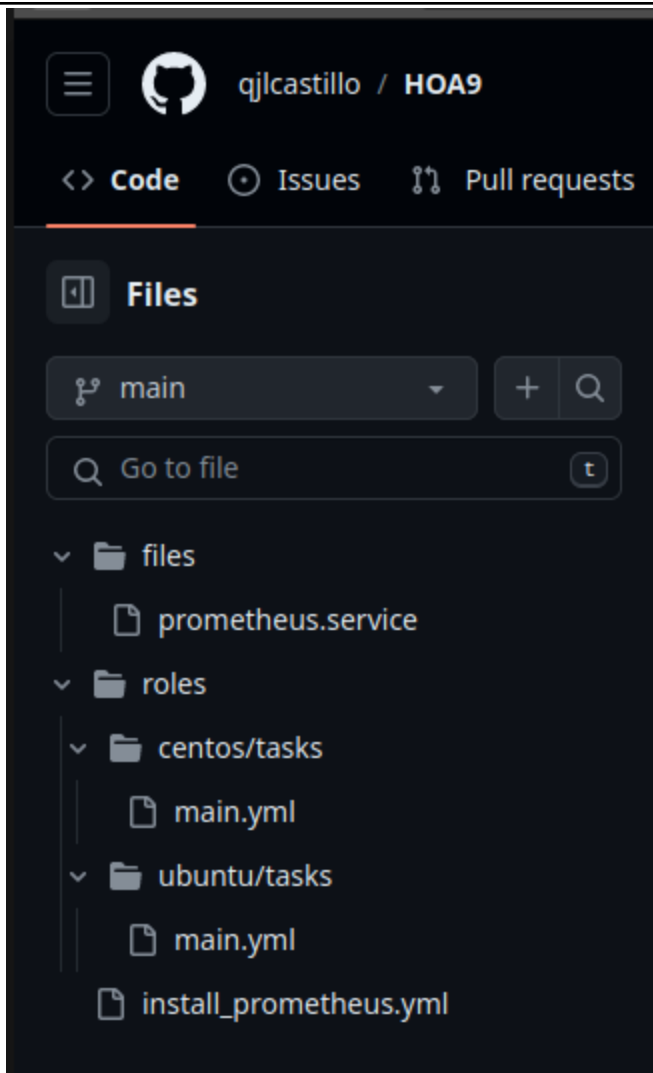
Reflections:

Answer the following:

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

   Having an availability monitoring tool like prometheus provides several significant benefits for organizations. It enables proactive monitoring, allowing businesses to identify and address issues before they impact operations or users. By continuously checking the availability and performance of systems, services, and network devices, prometheus helps ensure high uptime and reliability.

Conclusions:

In this Activity , we explored the process of installing, configuring, and managing availability monitoring tools. Specifically, we focused on prometheus as an example of such a tool.

By effectively installing, configuring, and managing availability monitoring tools like prometheus, organizations can enhance system reliability, reduce downtime, improve

incident response, and ensure optimal performance. These tools contribute to better customer satisfaction, productivity, and overall business success.