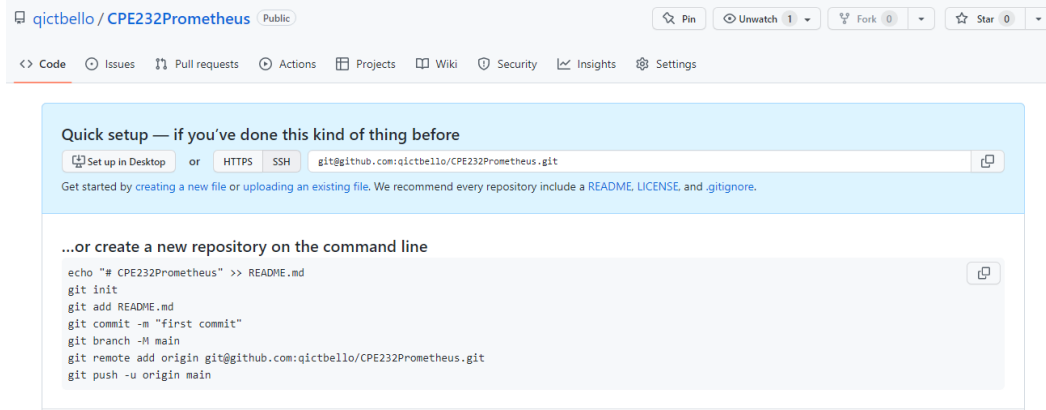


Name: Ian Carlo T. Bello	Date Performed: October 27, 2022
Course/Section: CPE31S24	Date Submitted: October 27, 2022
Instructor: Dr. Jonathan V. Taylar	Semester and SY: 1 st sem – 3 rd year
Activity 9: Install, Configure, and Manage Performance Monitoring tools	
1. Objectives	
Create and design a workflow 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)	
<p>First we need to create a new GitHub repository for this activity we will name it CPE232Prometheus.</p> <div data-bbox="722 1749 889 1814" data-label="Image"> </div>	



After creating we need to git clone it in our workstation or control node and work inside this repository

```
ubuntuhost@workstation:~$ git clone git@github.com:qictbello/CPE232Prometheus.git
Cloning into 'CPE232Prometheus'...
warning: You appear to have cloned an empty repository.
ubuntuhost@workstation:~$ cd CPE232Prometheus/
ubuntuhost@workstation:~/CPE232Prometheus$
```

In order to create a playbook we need both ansible config and inventory that will contain the ip of the servers. Now we will create it

```
ubuntuhost@workstation:~/CPE232Prometheus$ nano inventory
ubuntuhost@workstation:~/CPE232Prometheus$ nano ansible.cfg
ubuntuhost@workstation:~/CPE232Prometheus$
```

```
GNU nano 6.2 inventory *
[servers]
server1
servercent
```

```
GNU nano 6.2 ansible.cfg *
[defaults]
inventory = inventory
private_key_file = ~/.ssh/ansible
```

Now we created both inventory and our config files we need to create roles. In this scenario we need to install Prometheus in both servers so we will just create a role for Prometheus install.

```
ubuntuhost@workstation:~/CPE232Prometheus$ mkdir roles
ubuntuhost@workstation:~/CPE232Prometheus$ cd roles/
ubuntuhost@workstation:~/CPE232Prometheus/roles$ mkdir prometheus
ubuntuhost@workstation:~/CPE232Prometheus/roles$ cd prometheus/
ubuntuhost@workstation:~/CPE232Prometheus/roles/prometheus$ mkdir tasks
ubuntuhost@workstation:~/CPE232Prometheus/roles/prometheus$ cd tasks
ubuntuhost@workstation:~/CPE232Prometheus/roles/prometheus/tasks$
```

Then we will create a command playbook yml that will run for Prometheus to install in both servers.

```
GNU nano 6.2 main.yml *
---
- name: Installation Prometheus Ubuntu
  tags: ubuntu, prometheus
  apt:
    name: prometheus
    state: latest
  when: ansible_distribution == "Ubuntu"

- name: Pre-req install CentOS
  tags: centos, snapd
  yum:
    name:
      - snapd
    state: latest
  when: ansible_distribution == "CentOS"

- name: Enabling snapd CentOS
  tags: snapd, centos
  command: systemctl enable --now snapd.socket
  when: ansible_distribution == "CentOS"

- name: Installation Prometheus CentOS
  tags: prometheus, centos
  command: snap install prometheus --classic
  when: ansible_distribution == "CentOS"
```

After creating our main.yml we will go back to the directory and create our playbook that we will run.

```
GNU nano 6.2 prometheus.yml
- hosts: all
  become: true
  pre_tasks:

    - name: update repository index CentOS
      tags: always
      dnf:
        update_cache: yes
        changed_when: false
      when: ansible_distribution == "CentOS"

    - name: update repository index Ubuntu
      tags: always
      apt:
        update_cache: yes
        changed_when: false
      when: ansible_distribution == "Ubuntu"

    - name: start httpd for Prometheus CentOS
      tags: apache, centos, httpd, prometheus
      service:
        name: httpd
        state: started
      when: ansible_distribution == "CentOS"

- hosts: all
  become: true
  roles:
    - prometheus
```

After creating the playbook we can now run it and install Prometheus on both servers using ansible playbook.

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

TASK [Gathering Facts] *****
ok: [server1]
ok: [servercent]

TASK [prometheus : Installation Prometheus Ubuntu] *****
skipping: [servercent]
changed: [server1]

TASK [prometheus : Pre-req install CentOS] *****
skipping: [server1]
changed: [servercent]

TASK [prometheus : Enabling snapd CentOS] *****
skipping: [server1]
changed: [servercent]

TASK [prometheus : Installation Prometheus CentOS] *****
skipping: [server1]
fatal: [servercent]: FAILED! => [{"changed": true, "cmd": ["snap", "install", "prometheus", "--classic"], "delta": "0:00:01.535972", "end": "2022-10-27 14:00:29.636440", "msg": "non-zero return code", "rc": 10, "start": "2022-10-27 14:00:28.100468", "stderr": "error: too early for operation, device not yet seeded or device model not acknowledged", "stderr_lines": ["error: too early for operation, device not yet seeded or device model not acknowledged"], "stdout": "", "stdout_lines": []}]

PLAY RECAP *****
server1      : ok=4  changed=1  unreachable=0  failed=0  skipped=5  rescued=0  ignored=0
servercent   : ok=6  changed=2  unreachable=0  failed=1  skipped=2  rescued=0  ignored=0
```

We will get this error first because snapd wasn't started/install because it was a pre req for Prometheus to run. We just need to rerun the ansible playbook and we will install it successfully.

```
ubuntuhost@workstation:~/CPE232Prometheus$ ansible-playbook --ask-become-pass prometheus.yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [server1]
ok: [servercent]

TASK [update repository index CentOS] *****
skipping: [server1]
ok: [servercent]

TASK [update repository index Ubuntu] *****
skipping: [servercent]
ok: [server1]

TASK [start httpd for Prometheus CentOS] *****
skipping: [server1]
ok: [servercent]

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [server1]
ok: [servercent]

TASK [prometheus : Installation Prometheus Ubuntu] *****
skipping: [servercent]
ok: [server1]

TASK [prometheus : Pre-req install CentOS] *****
skipping: [server1]
```

```
TASK [prometheus : Pre-req install CentOS] *****
skipping: [server1]
ok: [servercent]

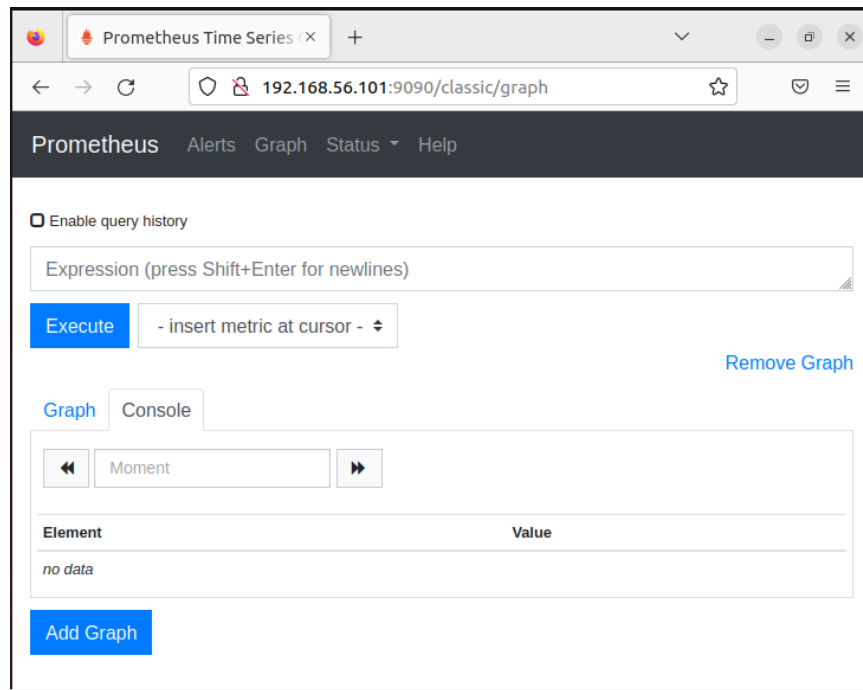
TASK [prometheus : Enabling snapd CentOS] *****
skipping: [server1]
changed: [servercent]

TASK [prometheus : Installation Prometheus CentOS] *****
skipping: [server1]
changed: [servercent]

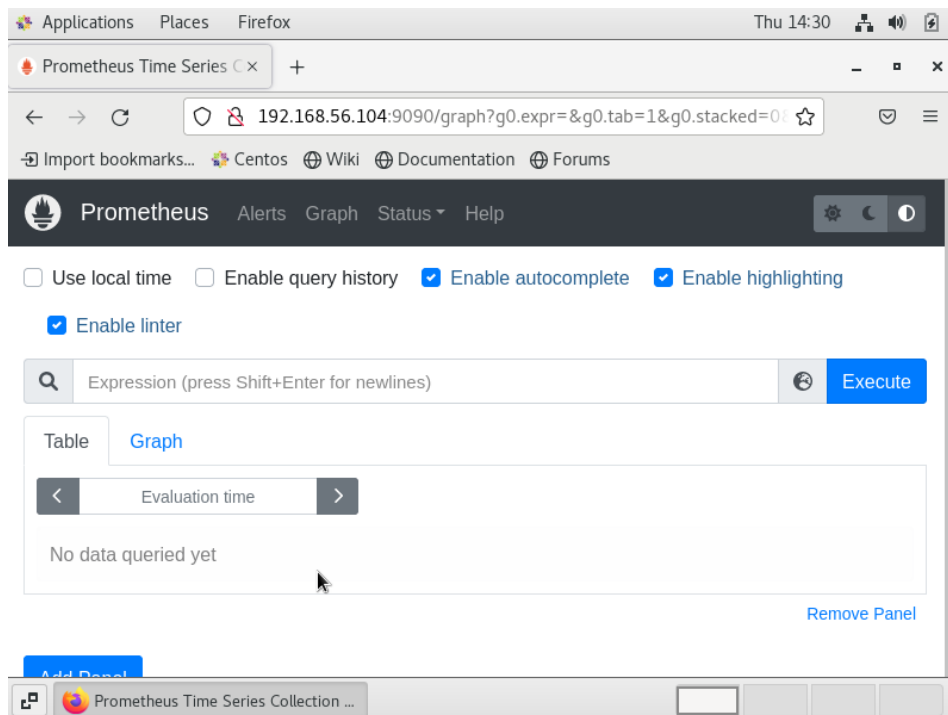
PLAY RECAP *****
server1      : ok=4  changed=0  unreachable=0  failed=0  skipped=5  rescued=0  ignored=0
servercent   : ok=7  changed=2  unreachable=0  failed=0  skipped=2  rescued=0  ignored=0
```

After installing Prometheus on both servers we can check if they do work and have an output.

UBUNTU



CENTOS



After we're done we need to git add, commit and push the repository.

```
ubuntuhost@workstation:~/CPE232Prometheus$ git add -A
ubuntuhost@workstation:~/CPE232Prometheus$ git commit -m "Prometheus"
[main (root-commit) 41ed37c] Prometheus
 4 files changed, 61 insertions(+)
 create mode 100644 ansible.cfg
 create mode 100644 inventory
 create mode 100644 prometheus.yml
 create mode 100644 roles/prometheus/tasks/main.yml
ubuntuhost@workstation:~/CPE232Prometheus$ git push
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (9/9), 1022 bytes | 1022.00 KiB/s, done.
Total 9 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:qictbello/CPE232Prometheus.git
 * [new branch]      main -> main
ubuntuhost@workstation:~/CPE232Prometheus$
```

Reflections:

Answer the following:

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

The benefit of having a performance monitoring tool is knowing how resources and consumption are doing in a log or graph format. In this activity, we installed Prometheus, which has a timestamped log that saves the monitoring system of servers.

Conclusions:

Finally, we learned how to use roles and ansible playbooks to install Prometheus on both Centos and Ubuntu servers. We encountered an error that is a prerequisite for Prometheus in Centos. We installed a tool that monitors how resources are consumed in workload saving in a timestamped and graphed interface using Prometheus. This activity is similar to activity 9, which is installing Nagios. We can use these commands and implementation in other installations and scenarios.