Name: Jose Mari T. Dela Peña	**Date Performed:** 10/21/2024
Course/Section: CPE31S2	**Date Submitted:** 10/28/2024
Instructor: Engr. Robin Valenzuela	**Semester and SY:** 1st Sem 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

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

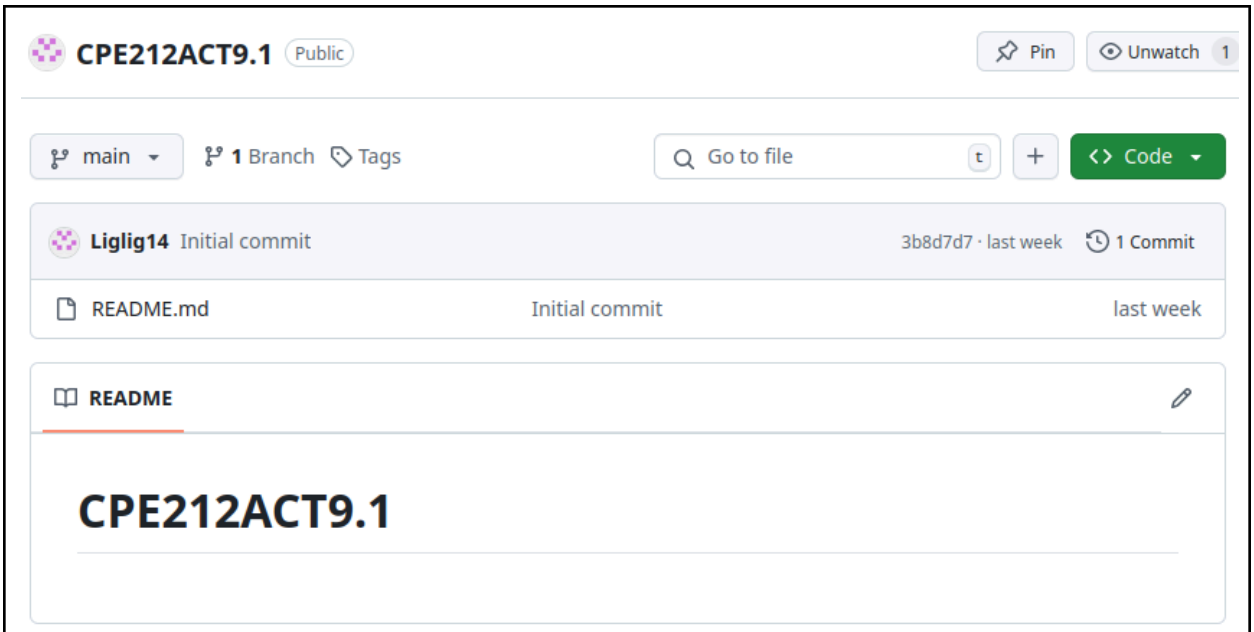
3. Tasks

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)

Step 1: Create and clone a new repository for this activity.

```
jose@workstation:~/CPE212ACT9.1$
```



Step 2: Set up ansible.cfg, inventory, roles, and install.yml

```
jose@workstation:~/CPE212ACT9.1$ ls
ansible.cfg  install.yml  inventory  README.md  roles
```

Step 3: Configure your inventory to group Ubuntu servers from CentOS servers

```
[UbuntuServers]
192.168.56.100
192.168.56.101
192.168.56.102
192.168.56.104
[CentOSServers]
192.168.56.103 ansible_user=Jose
```

Step 4: Configure your install.yml to have it run the roles

```yaml
---
- hosts: all
  become: true
  pre_tasks:

  - name: update repository index (CentOS)
    tags: always
    yum:
      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"

- hosts: UbuntuServers
  become: true
  roles:
    - UbuntuTasks

- hosts: CentOSServers
  become: true
  roles:
    - CentOSTasks
```

Step 5: Configure the main.yml in your roles to install, extract, run Prometheus

```
PLAY [UbuntuServers] ***********************************************************

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.100]
ok: [192.168.56.102]
ok: [192.168.56.101]
ok: [192.168.56.104]

TASK [UbuntuTasks : Install required packages] *********************************
ok: [192.168.56.100]
ok: [192.168.56.104]
ok: [192.168.56.101]
ok: [192.168.56.102]

TASK [UbuntuTasks : Create Prometheus directory] *******************************
ok: [192.168.56.100]
ok: [192.168.56.101]
ok: [192.168.56.104]
ok: [192.168.56.102]

TASK [UbuntuTasks : Download Prometheus] ***************************************
ok: [192.168.56.100]
ok: [192.168.56.102]
ok: [192.168.56.101]
ok: [192.168.56.104]

TASK [UbuntuTasks : Extract Prometheus] ****************************************
changed: [192.168.56.100]
changed: [192.168.56.104]
changed: [192.168.56.101]
changed: [192.168.56.102]

TASK [UbuntuTasks : Move Prometheus binary to /usr/local/bin] ******************
changed: [192.168.56.101]
changed: [192.168.56.102]
changed: [192.168.56.100]
changed: [192.168.56.104]

TASK [UbuntuTasks : Move Prometheus configuration file to /etc/prometheus] *****
ok: [192.168.56.102]
ok: [192.168.56.104]
ok: [192.168.56.100]
ok: [192.168.56.101]

TASK [UbuntuTasks : Create Prometheus service file] ****************************
ok: [192.168.56.100]
ok: [192.168.56.102]
ok: [192.168.56.104]
ok: [192.168.56.101]

TASK [UbuntuTasks : Start and enable Prometheus] *******************************
ok: [192.168.56.100]
ok: [192.168.56.104]
ok: [192.168.56.102]
ok: [192.168.56.101]
```

```
PLAY [CentOSServers] ***********************************************************

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

TASK [CentOSTasks : Install required packages] *********************************
ok: [192.168.56.103]

TASK [CentOSTasks : Create Prometheus directory] *******************************
ok: [192.168.56.103]

TASK [CentOSTasks : Download Prometheus] ***************************************
ok: [192.168.56.103]

TASK [CentOSTasks : Extract Prometheus] ****************************************
changed: [192.168.56.103]

TASK [CentOSTasks : Move Prometheus binary to /usr/local/bin] ******************
changed: [192.168.56.103]

TASK [CentOSTasks : Move Prometheus configuration file to /etc/prometheus] *****
ok: [192.168.56.103]

TASK [CentOSTasks : Create Prometheus service file] ****************************
changed: [192.168.56.103]

TASK [CentOSTasks : Start and enable Prometheus] *******************************
changed: [192.168.56.103]
```

Step 6: Verify that you have successfully have Prometheus on all servers

Ubuntu

```
jose@workstation:~/CPE212ACT9.1$ which prometheus
/usr/local/bin/prometheus
jose@workstation:~/CPE212ACT9.1$ prometheus --version
prometheus, version 2.54.1 (branch: HEAD, revision: e6cfa720fbe6280153fab13090a483dbd
40bece3)
  build user:       root@812ffd741951
  build date:       20240827-10:56:41
  go version:       go1.22.6
  platform:         linux/amd64
  tags:             netgo,builtinassets,stringlabels
```

CentOS

```
[Jose@localhost ~]$ which prometheus
/usr/local/bin/prometheus
[Jose@localhost ~]$ prometheus --version
prometheus, version 2.54.1 (branch: HEAD, revision: e6cfa720fbe6280153fab13090a483dbd40bece3)
  build user:       root@812ffd741951
  build date:       20240827-10:56:41
  go version:       go1.22.6
  platform:         linux/amd64
  tags:             netgo,builtinassets,stringlabels
```

Step 7: Push all the files to your Github repository

```
jose@workstation:~/CPE212ACT9.1$ git add ansible.cfg
jose@workstation:~/CPE212ACT9.1$ git add install.yml
jose@workstation:~/CPE212ACT9.1$ git add inventory
jose@workstation:~/CPE212ACT9.1$ git add roles
jose@workstation:~/CPE212ACT9.1$ git commit -m "10/28/2024"
[main 2226c00] 10/28/2024
 7 files changed, 170 insertions(+)
 create mode 100644 ansible.cfg
 create mode 100644 install.yml
 create mode 100644 inventory
 create mode 100644 roles/CentOSTasks/tasks/main.yml
 create mode 100644 roles/CentOSTasks/templates/prometheus.yml.j2
 create mode 100644 roles/UbuntuTasks/tasks/main.yml
 create mode 100644 roles/UbuntuTasks/templates/prometheus.yml.j2
jose@workstation:~/CPE212ACT9.1$ git push origin main
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 2 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (14/14), 1.76 KiB | 1.76 MiB/s, done.
Total 14 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To github.com:Liglig14/CPE212ACT9.1.git
   3b8d7d7..2226c00  main -> main
```

Reflections:

Answer the following:

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

- Prometheus is for real-time monitoring of Ubuntu and CentOS servers by watching the performance of the system from beginning to recent times. Also, it incorporates automated service discovery and alerting via Alertmanager, executing the act of sending relevant notifications for issue keeping in line with time. As an open-source

solution, it is cost-effective and backed by a strong community which helps organizations base their decisions on historical data.
Conclusions: - We have met the objectives of activity through installing this through playbook enhances our capabilities to automate server management.

Github Repository: https://github.com/Liglig14/CPE212ACT9.1