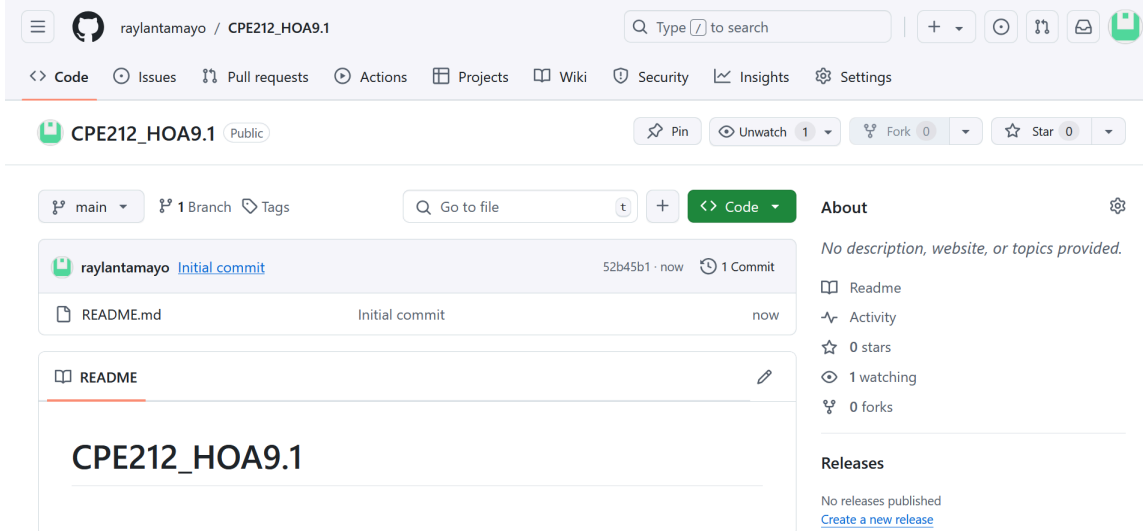


Name: Tamayo, Ray Lan A.	Date Performed: 10/18/2024
Course/Section: CPE212-CPE31S21	Date Submitted: 10/18/2024
Instructor: Engr. Robin Valenzuela	Semester and SY: First 2024-2025
Activity 9: Install, Configure, and Manage Performance Monitoring tools	
1. Objectives	
Create and design a workflow that installs, configure and manage enterprise performance tools using Ansible as an Infrastructure as Code (IaC) tool.	
2. Discussion	
<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	
<p>Task 1: Create a File</p> <ol style="list-style-type: none"> 1. Create a new repository for this activity. 	



2. Clone the repository to the local machine.

```
tamayo@workstation: ~/CPE212_HOA9.1
File Edit View Search Terminal Help
tamayo@workstation:~$ git clone git@github.com:raylantamayo/CPE212_HOA9.1.git
Cloning into 'CPE212_HOA9.1'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
tamayo@workstation:~$ ls
CPE212_HOA6.1 Desktop HOA7 site.retry Videos
CPE212_HOA8.1 Documents Music sysad2
CPE212_HOA9.1 Downloads Pictures Tamayo_PrelimExam
CPE232_Tamayo examples.desktop Public Templates
tamayo@workstation:~$ cd CPE212_HOA9.1
tamayo@workstation:~/CPE212_HOA9.1$
```

3. Create the ansible.cfg and inventory file (must include one Ubuntu and CentOS)

```
tamayo@workstation: ~/CPE212_HOA9.1
File Edit View Search Terminal Help
GNU nano 2.9.3 inventory

[ubuntu_prometheus]
192.168.56.128

[centos_prometheus]
192.168.56.130
```

```
tamayo@workstation: ~/CPE212_HOA9.1
File Edit View Search Terminal Help
GNU nano 2.9.3 ansible.cfg

[defaults]

inventory = inventory
host_key_checking = False

deprecation_warnings = False

remote_user = tamayo
private_key_file = ~/.ssh/
```

Task 2: Create Playbook for Installing Prometheus in Ubuntu and CentOS

1. Create a playbook and name it install_prometheus.yml.

```
tamayo@workstation: ~/CPE212_HOA9.1
File Edit View Search Terminal Help
GNU nano 2.9.3 install_prometheus.yml

---
- hosts: all
  become: true
  pre_tasks:

    - name: install updates (CentOS)
      dnf:
        update_only: yes
        update_cache: yes
        when: ansible_distribution == "CentOS"

    - name: install wget (CentOS)
      dnf:
        name: wget
        state: latest
        when: ansible_distribution == "CentOS"

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

```

- hosts: ubuntu_prometheus
  become: true
  roles:
    - ubuntu_prometheus

- hosts: centos_prometheus
  become: true
  roles:
    - centos_prometheus

```

Code Explanation

This code snippet is an Ansible task that updates only the packages on a CentOS system. It uses the dnf package manager to refresh the package cache and apply any available updates, but only if the operating system is identified as CentOS. Essentially, it's a way to ensure that a CentOS machine is kept up-to-date without installing new packages.

```

- name: install updates (CentOS)
  dnf:
    update_only: yes
    update_cache: yes
  when: ansible_distribution == "CentOS"

```

This code installs the latest version of the wget tool on a CentOS system. It uses the dnf package manager and only runs if the system is identified as CentOS. Essentially, it ensures that wget is up to date on that specific operating system.

```

- name: install wget (CentOS)
  dnf:
    name: wget
    state: latest
  when: ansible_distribution == "CentOS"

```

This code installs updates on an Ubuntu system. It uses the apt package manager to upgrade all packages to their latest versions and refreshes the package cache. It only runs if the system is recognized as Ubuntu, ensuring the system is kept up to date.

```

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

```

This code defines two groups of hosts: one for Ubuntu systems and one for CentOS systems. It uses the `become: true` option to grant administrative privileges for executing tasks, and it applies specific roles (`ubuntu_prometheus` and `centos_prometheus`) to each group, allowing for tailored configurations or installations on both types of systems.

```
- hosts: ubuntu_prometheus
  become: true
  roles:
    - ubuntu_prometheus

- hosts: centos_prometheus
  become: true
  roles:
    - centos_prometheus
```

Task 3: Create Roles

1. Create a new directory and name its roles. Enter the roles directory and create new directories: `centos_prometheus` and `ubuntu_prometheus`. For each directory, create a directory and name it `tasks`.

```
tamayo@workstation:~/CPE212_H0A9.1$ mkdir roles
tamayo@workstation:~/CPE212_H0A9.1$ cd roles
tamayo@workstation:~/CPE212_H0A9.1/roles$ mkdir ubuntu_prometheus
tamayo@workstation:~/CPE212_H0A9.1/roles$ cd ubuntu_prometheus
tamayo@workstation:~/CPE212_H0A9.1/roles/ubuntu_prometheus$ mkdir tasks
tamayo@workstation:~/CPE212_H0A9.1/roles$ cd ..
tamayo@workstation:~/CPE212_H0A9.1/roles$ mkdir centos_prometheus
tamayo@workstation:~/CPE212_H0A9.1/roles$ cd centos_prometheus
tamayo@workstation:~/CPE212_H0A9.1/roles/centos_prometheus$ mkdir tasks
tamayo@workstation:~/CPE212_H0A9.1/roles/centos_prometheus$ cd ..
tamayo@workstation:~/CPE212_H0A9.1/roles$
```

```
tamayo@workstation:~/CPE212_H0A9.1$ tree
.
├── ansible.cfg
├── install_prometheus.yml
├── inventory
├── README.md
└── roles
    ├── centos_prometheus
    │   └── tasks
    └── ubuntu_prometheus
        └── tasks

5 directories, 4 files
```

2. In each of the tasks for the two directory (`centos_prometheus` and `ubuntu_prometheus`), create another file and name it `main.yml`

```
tamayo@workstation:~/CPE212_H0A9.1/roles$ cd ubuntu_prometheus
tamayo@workstation:~/CPE212_H0A9.1/roles/ubuntu_prometheus$ cd tasks
tamayo@workstation:~/CPE212_H0A9.1/roles/ubuntu_prometheus/tasks$ touch main.yml
```

```
tamayo@workstation:~/CPE212_H0A9.1/roles$ cd centos_prometheus
tamayo@workstation:~/CPE212_H0A9.1/roles/centos_prometheus$ cd tasks
tamayo@workstation:~/CPE212_H0A9.1/roles/centos_prometheus/tasks$ touch main.yml
```

```
tamayo@workstation:~/CPE212_H0A9.1/roles$ tree
.
├── centos_prometheus
│   └── tasks
│       └── main.yml
└── ubuntu_prometheus
    └── tasks
        └── main.yml

4 directories, 2 files
```

3. Copy the code to the main.yml of the Ubuntu subdirectory.

```
tamayo@workstation: ~/CPE212_H0A9.1/roles/ubuntu_prometheus/tasks
File Edit View Search Terminal Help
GNU nano 2.9.3 main.yml Mi
- name: install Prometheus (Ubuntu)
  apt:
    name: prometheus
    state: latest

- name: Prometheus Start/Enable Check
  service:
    name: prometheus
    state: restarted
    enabled: true

- name: Apache Start/Enable Check
  service:
    name: prometheus
    state: restarted
    enabled: true
```

4. Copy the code to the main.yml of the CentOS subdirectory.

```
tamayo@workstation: ~/CPE212_HOA9.1/roles/centos_prometheus/tasks
File Edit View Search Terminal Help
GNU nano 2.9.3 main.yml Modified

- name: Creating a directory (where the downloaded files will be stored)
  tags: directory
  file:
    path: ~/prometheus
    state: directory

- name: Downloading and extracting Prometheus
  tags: source
  unarchive:
    src: https://github.com/prometheus/prometheus/releases/download/v2.8.1/prometheus-2.8.1-linux-amd64.tar.gz
    dest: ~/prometheus
    remote_src: yes
    mode: 0777
    owner: root
    group: root

- name: Stopping the service
  service:
    name: prometheus
    state: stopped

- name: Adding the Prometheus executables to a PATH
  tags: executables
```

```
shell: |
  cd ~/prometheus/prometheus*
  cp -r . /usr/local/bin/prometheus

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

```
- name: Making sure that Prometheus is started and enabled
  tags: serviceon
  service:
    name: prometheus
    state: restarted
    enabled: true
```

Task 4: Run and Verify

1. Run the command `ansible-playbook - - ask-become-pass install_prometheus.yml` to completely install Nagios in both Ubuntu server and CentOS.

```

tamayo@workstation: ~/CPE212_HOA9.1
File Edit View Search Terminal Help
tamayo@workstation:~/CPE212_HOA9.1$ ansible-playbook --ask-become-pass install_
prometheus.yml
SUDO password:

PLAY [all] *****
*

TASK [Gathering Facts] *****
*
ok: [192.168.56.130]
ok: [192.168.56.128]

TASK [install updates (CentOS)] *****
*
skipping: [192.168.56.130]
skipping: [192.168.56.128]

TASK [install wget (CentOS)] *****
*
skipping: [192.168.56.130]
skipping: [192.168.56.128]

TASK [install updates (Ubuntu)] *****
*
ok: [192.168.56.130]
ok: [192.168.56.128]

PLAY [ubuntu_prometheus] *****
*

TASK [Gathering Facts] *****
*
ok: [192.168.56.128]

TASK [ubuntu_prometheus : install Prometheus (Ubuntu)] *****
*
changed: [192.168.56.128]

TASK [ubuntu_prometheus : Prometheus Start/Enable Check] *****
*
changed: [192.168.56.128]

TASK [ubuntu_prometheus : Apache Start/Enable Check] *****
*
changed: [192.168.56.128]

PLAY [centos_prometheus] *****
*

TASK [Gathering Facts] *****
*
ok: [192.168.56.130]

TASK [centos_prometheus : Creating a directory (where the downloaded files will
be stored)] ***
changed: [192.168.56.130]

```

2. Show the screenshot of the Nagios in both Server and CentOS, by simply typing its ip address in the web browser and :9090.

UBUNTU

Prometheus Time Series

192.168.56.128:9090/graph

Prometheus

☐ Enable query history

Expression (press Shift+Enter for newlines)

Execute

- insert metric at cursor -

Graph Console

Element	Value
no data	

Remove Graph

Add Graph

CENTOS

Prometheus Time Series

192.168.56.131:9090/graph

Prometheus

☐ Enable query history

Expression (press Shift+Enter for newlines)

Execute

- insert metric at cursor -

Graph Console

Element	Value
no data	

Remove Graph

Add Graph

3. Upload it in the github.

```
tamayo@workstation:~/CPE212_HOA9.1$ git add .
tamayo@workstation:~/CPE212_HOA9.1$ git commit -m "HOA 9.1"
[main 501beb1] HOA 9.1
 7 files changed, 114 insertions(+)
 create mode 100644 ansible.cfg
 create mode 100644 "d\0330Masd\0330M\0330M\0330M\0330M"
 create mode 100644 install_prometheus.retry
 create mode 100644 install_prometheus.yml
 create mode 100644 inventory
 create mode 100644 roles/centos_prometheus/tasks/main.yml
 create mode 100644 roles/ubuntu_prometheus/tasks/main.yml
tamayo@workstation:~/CPE212_HOA9.1$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
tamayo@workstation:~/CPE212_HOA9.1$ git push origin
Counting objects: 14, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (9/9), done.
Writing objects: 100% (14/14), 1.95 KiB | 1.95 MiB/s, done.
Total 14 (delta 0), reused 0 (delta 0)
To github.com:raylantanamayo/CPE212_HOA9.1.git
 52b45b1..501beb1  main -> main
```

GITHUB LINK: https://github.com/raylantanamayo/CPE212_HOA9.1.git

Reflections:

Answer the following:

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

Having a performance monitoring tool through Ansible offers several benefits. First, it automates the setup and management of monitoring systems, saving time and reducing human error. Second, it provides real-time insights into how your systems are performing, allowing you to quickly spot and fix issues before they escalate. Additionally, it helps ensure that your applications are running smoothly, which leads to a better experience for users. Overall, it makes it easier to keep everything running efficiently and reliably.

Conclusions:

In this activity, I learned how to use Ansible to install and manage performance monitoring tools like Prometheus on both Ubuntu and CentOS systems. By creating roles, I streamlined the installation process, making it easier to apply

the same setup across different environments. I discovered the importance of performance monitoring in keeping systems running smoothly and being able to quickly identify issues. This hands-on experience also emphasized the benefits of using Infrastructure as Code, as it automates tasks and reduces the chances of errors. Overall, I gained practical skills in automation and a better understanding of how to monitor system performance effectively.