Name: Abigail Laxamana	Date Performed: October 19, 2023
Course/Section: CPE 232 - CPE31S6	Date Submitted: October 19, 2023
Instructor: Dr. Jonathan Taylar	Semester and SY: 1st sem, 2023 - 2024

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

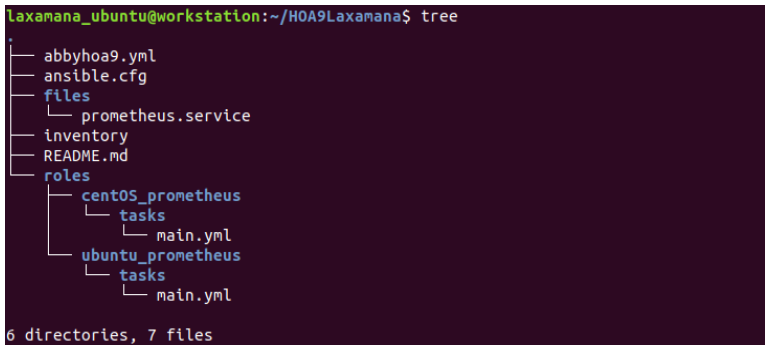
1. Create a playbook that installs Prometheus in both Ubuntu and CentOS. Apply the concept of creating roles.
2. Describe how you did step 1. (Provide screenshots and explanations in your report. Make your report detailed such that it will look like a manual.)

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)

In this activity, you'll create a tree that would look like the image below. You will implement creating roles again and putting tasks inside it that contains a file. First, you need to clone the newly created repository from your github to your remote server first using the **git clone** command.

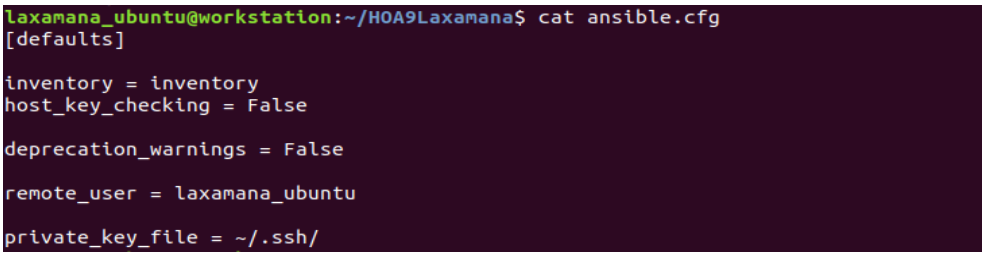
Tree



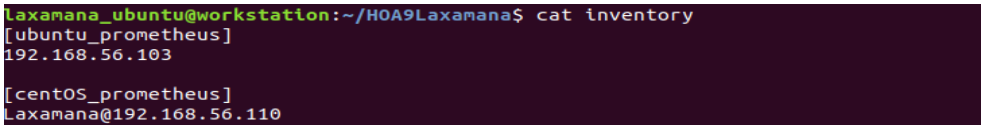
After cloning, you will have to create files named **inventory** and **ansible.cfg**.

ansible.cfg

*The content of **ansible.cfg** file should look like this.*



*The **inventory** file should contain the ip addresses of the server you'd want to manipulate and metagroups must also be stated. It should look like this.*



To create the *tree*, you have to make a directory named **roles**. Inside the roles directory, create another directories, one for centos, and one for ubuntu server, called **centOS_prometheus** and **ubuntu_prometheus** (suggested but not required names). Then, create another directory called **tasks** inside of each. The **tasks** directory will contain the playbooks called **main.yml**.

The content of the abbyhoa9.yml or the main ansible playbook inside the directory ***~/HOA9Laxamana/abbyhoa9.yml*** *should look like this.*

```
laxamana_ubuntu@workstation:~/HOA9Laxamana$ cat abbyhoa9.yml
---

- hosts: all
  become: true
  pre_tasks:

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

    - name: install wget (CentOS)
      package:
        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
```

The content of the main.yml file for centOS in the directory ***~/HOA9Laxamana/roles/centOS_prometheus/tasks/main.yml*** *should look like this.*

```
laxamana_ubuntu@workstation:~/HOA9Laxamana/roles/centOS_prometheus/tasks$ cat m
ain.yml
- 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/prom
etheus-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
laxamana_ubuntu@workstation:~/HOA9Laxamana/roles/centOS_prometheus/tasks$
```

The content of the main.yml file for the ubuntu server in the directory ***~/HOA9Laxamana/roles/ubuntu_prometheus/tasks/main.yml*** *should look like this.*

```
laxamana_ubuntu@workstation:~/HOA9Laxamana/roles/ubuntu_prometheus/tasks$ cat m
ain.yml
  - 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
```

Inside the repository, you will create another directory ~/**HOA9Laxamana**/**files** that contains a **prometheus.service** file that includes the information about prometheus management. It is where the binaries or the processes will be found.

The ***prometheus.service*** *file should look like this.*

```
laxamana_ubuntu@workstation:~/HOA9Laxamana/files$ cat prometheus.service
[Unit]
Description=Prometheus
After=network.target

[Service]
Type=simple
ExecStart=/usr/local/bin/prometheus/prometheus --config.file=/usr/local/bin/pro
metheus/prometheus.yml

[Install]
WantedBy=multi-user.target
```

verification

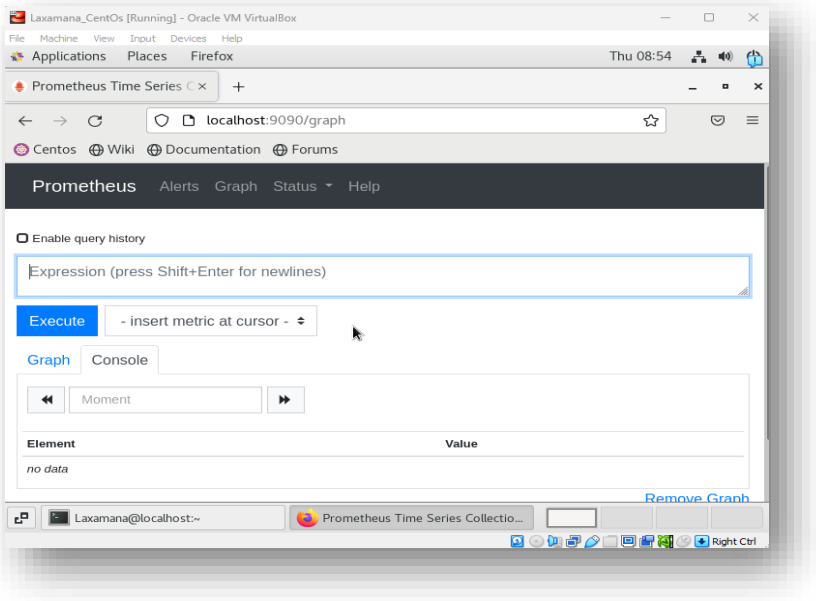
server 1

```
laxamana_ubuntu@server1:~$ systemctl status prometheus
● prometheus.service - Monitoring system and time series database
   Loaded: loaded (/lib/systemd/system/prometheus.service; enabled; vendor pres
   Active: active (running) since Thu 2023-10-19 16:45:43 PST; 27min ago
     Docs: https://prometheus.io/docs/introduction/overview/
 Main PID: 29556 (prometheus)
    Tasks: 9 (limit: 4656)
   CGroup: /system.slice/prometheus.service
           └─29556 /usr/bin/prometheus

Oct 19 16:45:43 server1 prometheus[29556]: level=info ts=2023-10-19T08:45:43.12
Oct 19 16:45:43 server1 prometheus[29556]: level=info ts=2023-10-19T08:45:43.12
Oct 19 16:45:43 server1 prometheus[29556]: level=info ts=2023-10-19T08:45:43.12
Oct 19 16:45:43 server1 prometheus[29556]: level=info ts=2023-10-19T08:45:43.12
Oct 19 16:45:43 server1 prometheus[29556]: level=info ts=2023-10-19T08:45:43.13
Oct 19 16:45:43 server1 prometheus[29556]: level=info ts=2023-10-19T08:45:43.13
Oct 19 16:45:43 server1 prometheus[29556]: level=info ts=2023-10-19T08:45:43.16
Oct 19 16:45:43 server1 prometheus[29556]: level=info ts=2023-10-19T08:45:43.16
Oct 19 16:45:43 server1 prometheus[29556]: level=info ts=2023-10-19T08:45:43.16
Oct 19 16:45:43 server1 prometheus[29556]: level=info ts=2023-10-19T08:45:43.16
lines 1-19/19 (END)
```

```
laxamana_ubuntu@server1:~$ prometheus --version
prometheus, version 2.1.0+ds (branch: debian/sid, revision: 2.1.0+ds-1)
  build user:       pkg-go-maintainers@lists.alioth.debian.org
  build date:       20180121-21:30:42
  go version:       go1.9.2
```

centOS



```
[Laxamana@localhost ~]$ systemctl status prometheus
● prometheus.service - Prometheus
   Loaded: loaded (/etc/systemd/system/prometheus.service; enabled; vendor preset: disa
bled)
   Active: active (running) since Thu 2023-10-19 08:52:14 EDT; 39s ago
 Main PID: 18785 (prometheus)
   CGroup: /system.slice/prometheus.service
           └─18785 /usr/local/bin/prometheus/prometheus --config.file=/usr/local/bin...

Oct 19 08:52:14 localhost.localdomain prometheus[18785]: level=info ts=2023-10-19T12...
Oct 19 08:52:14 localhost.localdomain prometheus[18785]: level=info ts=2023-10-19T12...
Oct 19 08:52:14 localhost.localdomain prometheus[18785]: level=info ts=2023-10-19T12...
Oct 19 08:52:14 localhost.localdomain prometheus[18785]: level=info ts=2023-10-19T12...
Oct 19 08:52:14 localhost.localdomain prometheus[18785]: level=info ts=2023-10-19T12...
Oct 19 08:52:14 localhost.localdomain prometheus[18785]: level=info ts=2023-10-19T12...
Oct 19 08:52:14 localhost.localdomain prometheus[18785]: level=info ts=2023-10-19T12...
Oct 19 08:52:14 localhost.localdomain prometheus[18785]: level=info ts=2023-10-19T12...
Oct 19 08:52:14 localhost.localdomain prometheus[18785]: level=info ts=2023-10-19T12...
Oct 19 08:52:14 localhost.localdomain prometheus[18785]: level=info ts=2023-10-19T12...
Hint: Some lines were ellipsized, use -l to show in full.
```

playbook process

```
laxamana_ubuntu@workstation:~/HOA9Laxamana$ ansible-playbook --ask-become-pass
abbyhoa9.yml
BECOME password:

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

TASK [Gathering Facts] ********************************************************
*
ok: [192.168.56.103]
ok: [Laxamana@192.168.56.110]

TASK [install updates (CentOS)] ***********************************************
*
skipping: [192.168.56.103]
ok: [Laxamana@192.168.56.110]

TASK [install wget (CentOS)] **************************************************
*
skipping: [192.168.56.103]
ok: [Laxamana@192.168.56.110]

TASK [install updates (Ubuntu)] ***********************************************
*
skipping: [Laxamana@192.168.56.110]
ok: [192.168.56.103]
```

```
PLAY [ubuntu_prometheus] ***************************************************
*

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

TASK [ubuntu_prometheus : install Prometheus (Ubuntu)] *********************
*
ok: [192.168.56.103]

TASK [ubuntu_prometheus : Prometheus Start/Enable Check] *******************
*
changed: [192.168.56.103]

TASK [ubuntu_prometheus : Apache Start/Enable Check] ************************
*
changed: [192.168.56.103]

PLAY [centOS_prometheus] ***************************************************
*

TASK [Gathering Facts] *****************************************************
*
ok: [Laxamana@192.168.56.110]
```

```
TASK [centOS_prometheus : Creating a directory (where the downloaded files will
 be stored)] ***
ok: [Laxamana@192.168.56.110]

TASK [centOS_prometheus : Downloading and extracting Prometheus] ***********
*
ok: [Laxamana@192.168.56.110]

TASK [centOS_prometheus : Stopping the service] *****************************
*
changed: [Laxamana@192.168.56.110]

TASK [centOS_prometheus : Adding the Prometheus executables to a PATH] ********
*
changed: [Laxamana@192.168.56.110]

TASK [centOS_prometheus : Copying the Prometheus service file] ***************
*
ok: [Laxamana@192.168.56.110]

TASK [centOS_prometheus : Making sure that Prometheus is started and enabled] *
**
changed: [Laxamana@192.168.56.110]
```

```
PLAY RECAP *********************************************************************
*
192.168.56.103             : ok=6    changed=2    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
Laxamana@192.168.56.110    : ok=10   changed=3    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
```

git commit

```
laxamana_ubuntu@workstation:~/HOA9Laxamana$ git add .
laxamana_ubuntu@workstation:~/HOA9Laxamana$ git commit -m "FINAL HOA 9 DONE"
[main f963354] FINAL HOA 9 DONE
 5 files changed, 77 insertions(+), 84 deletions(-)
 rewrite abbyhoa9.yml (66%)
 create mode 100644 files/.product.service.swp
 create mode 100644 files/.prometheus.servic.swp
 rewrite roles/centOS_prometheus/tasks/main.yml (99%)
laxamana_ubuntu@workstation:~/HOA9Laxamana$ git push origin
Counting objects: 11, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (9/9), done.
Writing objects: 100% (11/11), 1.42 KiB | 1.42 MiB/s, done.
Total 11 (delta 3), reused 0 (delta 0)
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.
To github.com:Abigaiiiil/HOA9Laxamana.git
   69c001d..f963354  main -> main
```

Reflections:

Answer the following:

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

The capacity to swiftly recognize and fix performance issues is a major advantage. IT teams may identify abnormalities and bottlenecks with the use of these technologies, which give real-time or historical data on the performance and health of systems. Early detection facilitates quick troubleshooting and reduces downtime, both of which are essential for preserving business continuity. The ability to maximize resource consumption is another benefit. Utilizing performance monitoring tools, businesses may examine how much CPU, memory, and bandwidth are being used and decide how to scale back or allocate resources accordingly. This helps manage infrastructure expenses effectively and enhances system performance.

Conclusions:

In conclusion, using an Ansible playbook to install Prometheus offers a quick and organized way to manage monitoring infrastructure in CentOS and Ubuntu systems. The playbook design, roles, and inventory management of Ansible make it possible to automate and adapt the installation process to a variety of server setups. By utilizing roles designed for various operating systems, you can quickly extend and maintain the playbook as your infrastructure changes. By doing this, you can streamline the deployment procedure and guarantee that Prometheus and other monitoring tools are installed consistently and securely throughout your environment. With this automation in place, you can efficiently monitor your systems, gather crucial performance data, and address problems, eventually resulting in a more reliable and stable IT infrastructure in your company. By working together, Ansible and Prometheus give administrators the resources they need to monitor system health, make wise choices, and give users and customers better service.