Name: Hermitano, Johnny	Date Performed: 10/26/2022
Course/Section: CPE31S23	Date Submitted: 10/26/2022
Instructor: Engr. Jonathan Taylar	Semester and SY: 1st sem sy 2022-2023

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. Enter the command **ssh-keygen** to generate an rsa key.

```
jhermitano@Workstation:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/jhermitano/.ssh/id_rsa):
/home/jhermitano/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/jhermitano/.ssh/id_rsa
Your public key has been saved in /home/jhermitano/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:YRwMmMM8DwRolNqmDuHPOVg2qnJxN6SrAeZNw36qHKw jhermitano@Workstation
The key's randomart image is:
+---[RSA 3072]----+
|.oo=.o.o.        |
|.o  O  ...       |
|o.   =  +        |
|o +   o. .       |
|o= + o  S        |
|*o=++ o          |
|o+O=o+ .         |
|++o*+            |
|E+oo.            |
+----[SHA256]-----+
```

Step 2. Connect your control node to your manage node through ssh by entering the code **ssh-copy-id server@ip address** for Ubuntu and **ssh-copy-id -i ~/.ssh/id_rsa server@ip address** for CentOS.

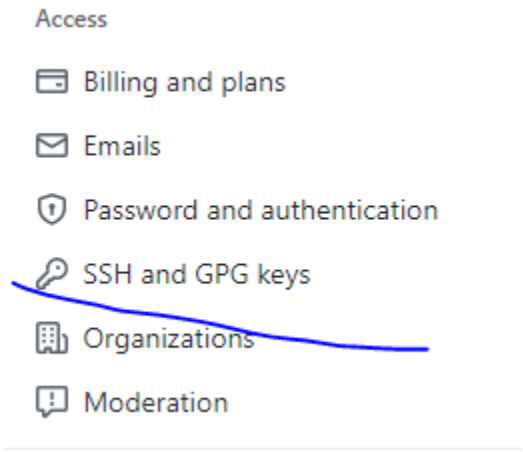
Ubuntu

```
jhermitano@Workstation:~/Hoa_8.1_Portfolio$ ssh-copy-id 192.168.56.105
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
 out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are promp
ted now it is to install the new keys
jhermitano@192.168.56.105's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh '192.168.56.105'"
and check to make sure that only the key(s) you wanted were added.
```

CentOS

```
jhermitano@Workstation:~/Hoa_8.1_Portfolio$ ssh-copy-id -i ~/.ssh/id_rsa jhermi
tano@192.168.56.115
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/jhermitano
/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
 out any that are already installed

/usr/bin/ssh-copy-id: WARNING: All keys were skipped because they already exist
 on the remote system.
                (if you think this is a mistake, you may want to use -f option)
```

Step 3. Connect your control node to your github account by adding your ssh key onto your github. To connect go to settings and click the **SSH and GPG keys.**



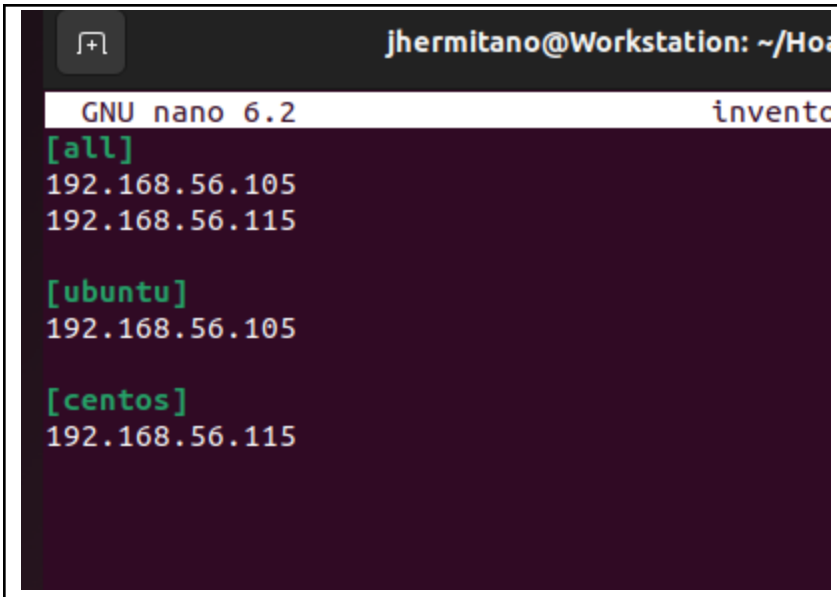
Just click the **New SSH keys** and then you may copy paste your rsa key from your control node. To see the rsa keys, enter **cat id_rsa.pub** and you will see your rsa keys.

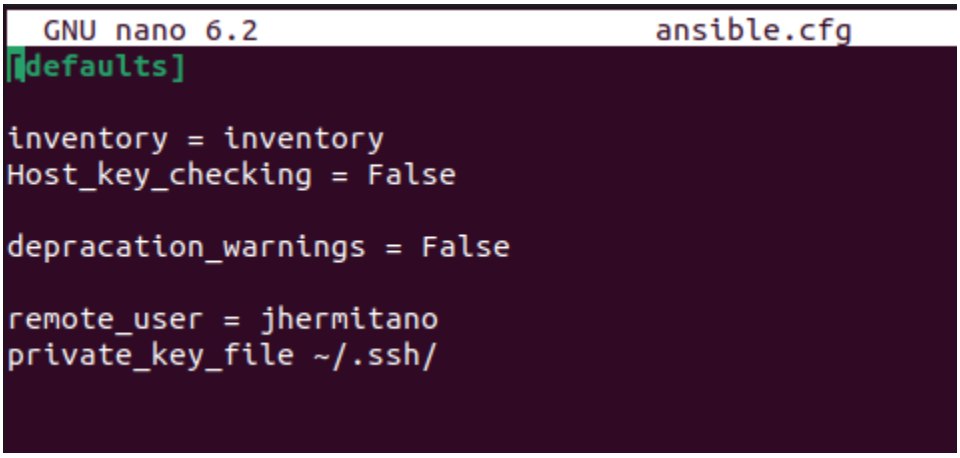
```
jhermitano@Workstation:~/.ssh$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCSZcs0YK385QI9dLl66GRWac+J1/Uc3D2tt0c4gH9
11mCna6vcqmLKfUSvyZKkEI0KI8rYvzoZSG6cvsSt9cQ3YK8F9PTUid1o/EenT7YvU7ehA+1R8e5aSb
bWu2zgvqaHIkvnqVoxHruehArk5ddMY4WF9toAHytZOANNOCVUuI33u8omdCFRFvHnxiq+UjccZKeGA
gbQ8GvW+5zC53pv9UDKOF/KUEHVX1wvt8nQRP1LmWmBvodnITs+EL7iwmtvKLD1KRdKtTYo+FfVrEjE
s5a8GEMaqeFPl6kMcAbwUdoliebIw+ZLBMRkmEwpywioG4kc74RT7Jh2NB3psWHF9fkJyItWPN4ZaZL
09n6wf5BT3cBzPI/fUgSGfhNc06le24I2cXaB3EFIu7pbiuTolDtfeEbov3m2FO3GFkaItutTpbyvgH
htQOGQReYG45obdBx46wREQVS+npF4kirHZYtkKystZf+afdTX8LcMBKOhTJmfMvJYulSY/6CQ7N0=
jhermitano@Workstation
```

Step 4. Clone your github repository by entering **git clone “your github repository link”**.

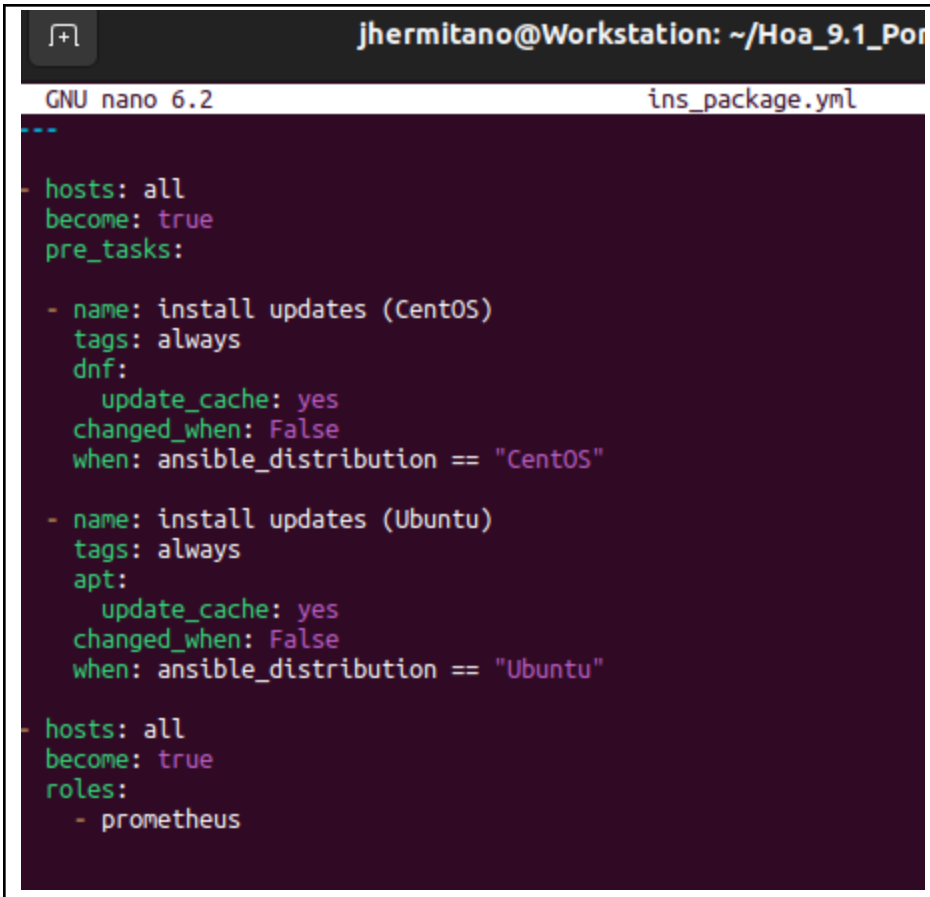
```
jhermitano@Workstation:~$ git clone git@github.com:jchermitano/Hoa_9.1_Portfoli
o.git
Cloning into 'Hoa_9.1_Portfolio'...
warning: You appear to have cloned an empty repository.
jhermitano@Workstation:~$ ls
ansible.cfg       Documents          Hoa_9.1_Portfolio    snap
ansible_hoa5-1    Downloads          Music                Templates
CPE232_Hermitano  get-pip.py         Pictures             Videos
CPE232_Johnny     hermitano          Public
Desktop           Hoa_8.1_Portfolio  SecondSemRepository
```

Step 5. Inside your repository, create your nano inventory and ansible.cfg for your playbook.





Step 6. Create your playbook.



Step 7. Create a new directory named roles and inside it is another directory of your choice to put your main.yml. Then create your main.yml playbook.

Step 8. After setting up your playbook, run it now with “ansible-playbook –ask-become-pass ‘playbook’” command.

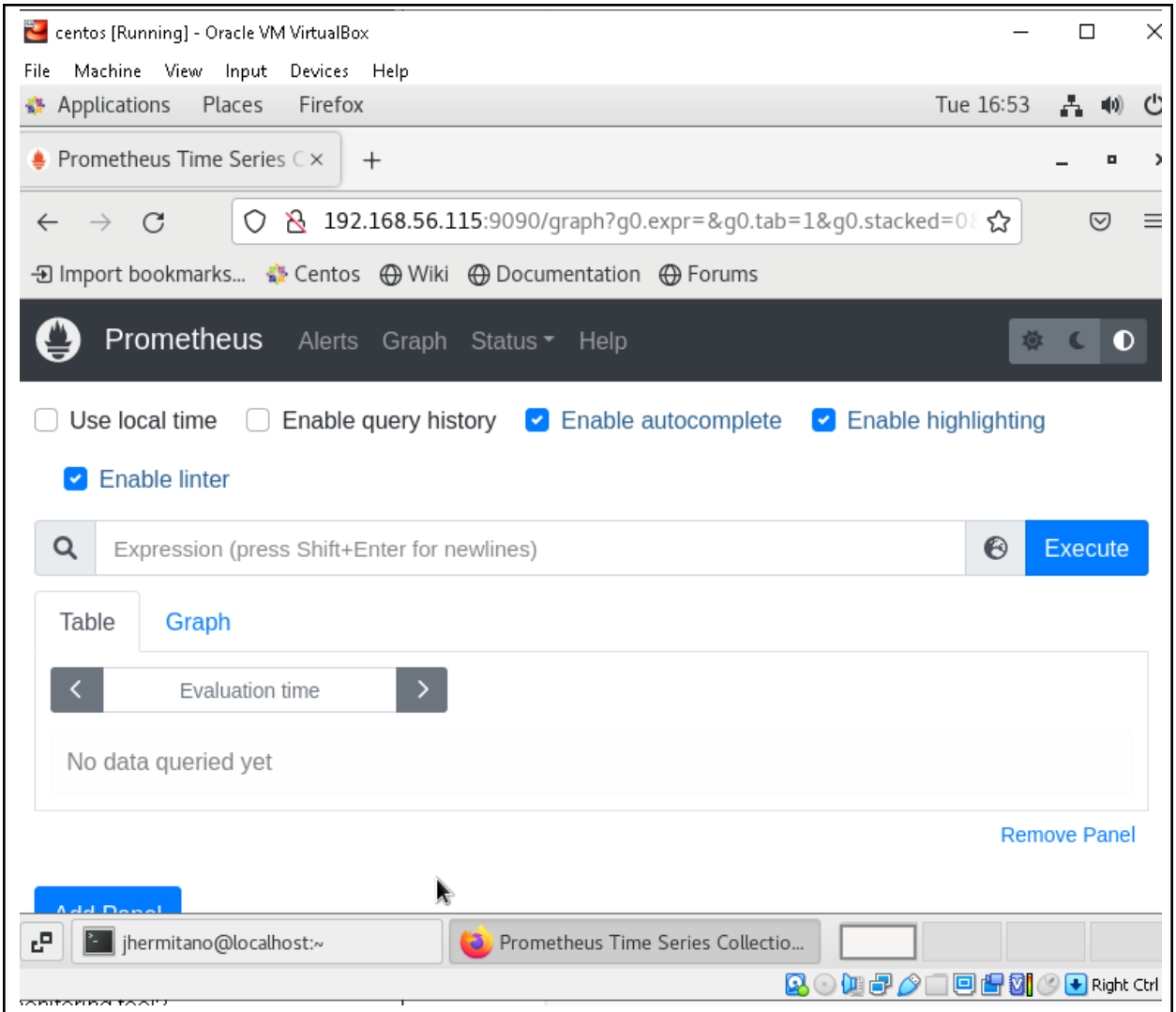
```
jhermitano@Workstation:~/Hoa_9.1_Portfolio$ ansible-playbook --ask-become-pass ins_package.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************************
ok: [192.168.56.105]
ok: [192.168.56.115]

TASK [install updates (CentOS)] ************************************************************
skipping: [192.168.56.105]
ok: [192.168.56.115]

TASK [install updates (Ubuntu)] ************************************************************
skipping: [192.168.56.115]
ok: [192.168.56.105]

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

TASK [Gathering Facts] *********************************************************************
ok: [192.168.56.105]
ok: [192.168.56.115]

TASK [prometheus : Prometheus installation (Ubuntu)] ***************************************
skipping: [192.168.56.115]
ok: [192.168.56.105]

TASK [prometheus : Pre-requisite of installation for CentOS] *******************************
skipping: [192.168.56.105]
ok: [192.168.56.115]

TASK [prometheus : enabling snapd sockets for CentOS] **************************************
skipping: [192.168.56.105]
changed: [192.168.56.115]

TASK [prometheus : Prometheus installation (CentOS)] ***************************************
skipping: [192.168.56.105]
changed: [192.168.56.115]

PLAY RECAP *********************************************************************************
192.168.56.105             : ok=4    changed=0    unreachable=0    failed=0    skipped=4    rescued=0    ignored=0
192.168.56.115             : ok=6    changed=2    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0

jhermitano@Workstation:~/Hoa_9.1_Portfolio$
```

Step 9. Check your control nodes if it is successfully installed on the prometheus.

Ubuntu

```
jhermitano@Server1:~$ prometheus --version
prometheus, version 2.31.2+ds1 (branch: debian/sid, revision: 2.31.2+ds1-1ubunt
u1)
  build user:       team+pkg-go@tracker.debian.org
  build date:       20220317-16:26:29
  go version:       go1.17.3
  platform:         linux/amd64
jhermitano@Server1:~$
```

CentOS



Reflections:

Answer the following:

1. What are the benefits of having a performance monitoring tool?
   Monitoring systems should be able to help with capacity management by looking at both network and device consumption in addition to providing information on trends. This is essential to the operation because it can identify specific sites where network bandwidth is being restricted.

Conclusions:

In this project, I was able to create and refine a playbook that employs ansible to install monitoring tools in CentOS and Ubuntu. Additionally, I now have a better understanding of the roles and the Ansible playbook.

Github link: https://github.com/jchermitano/Hoa_9.1_Portfolio.git