



Name: Andreu John L. Salvador	Date Performed: 24/10/2023
Course/Section: CPE31S5	Date Submitted: 25/10/2023
Instructor: Engr. Roman Richard	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	
<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>Part 1: Making the repository</p> <p>Before anything else we need to make a specific or exclusive directory for our installation and in order to do that we will make a new repository in our github and copy this to our workstation to generate our current working directory where the playbook will be created.</p>	


← → ↻ <https://github.com/new> ☆

☰  New repository 🔍 Type  to search >_ + ⌵ ⌚ ⚙

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)


Required fields are marked with an asterisk (*).


Owner *  qajlsalvador / Repository name *

✔ Prometheus is available.

Great repository names are short and memorable. Need inspiration? How about [automatic-succotash](#) ?

Description (optional)

☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

☒ **Add a README file**
This is where you can write a long description for your project. [Learn more about READMEs.](#)

☐ **Add .gitignore**

```
salvador@Workstation:~$ git clone git@github.com:qajlsalvador/Prometheus.git
Cloning into 'Prometheus'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
salvador@Workstation:~$ ls
ansible Desktop Downloads id_rsa.pub Nagios Pictures Public snap Videos
CPE232_AndreuSalvador Documents id_rsa Music nagios-ansible Prometheus Salvador_PrelimExam Templates
salvador@Workstation:~$ cd Prometheus
salvador@Workstation:~/Prometheus$
```

Part 2: Installation of Prometheus in Ubuntu server

1. Create our inventory where we will put the ip address of our live servers and group them in their respective roles.

```

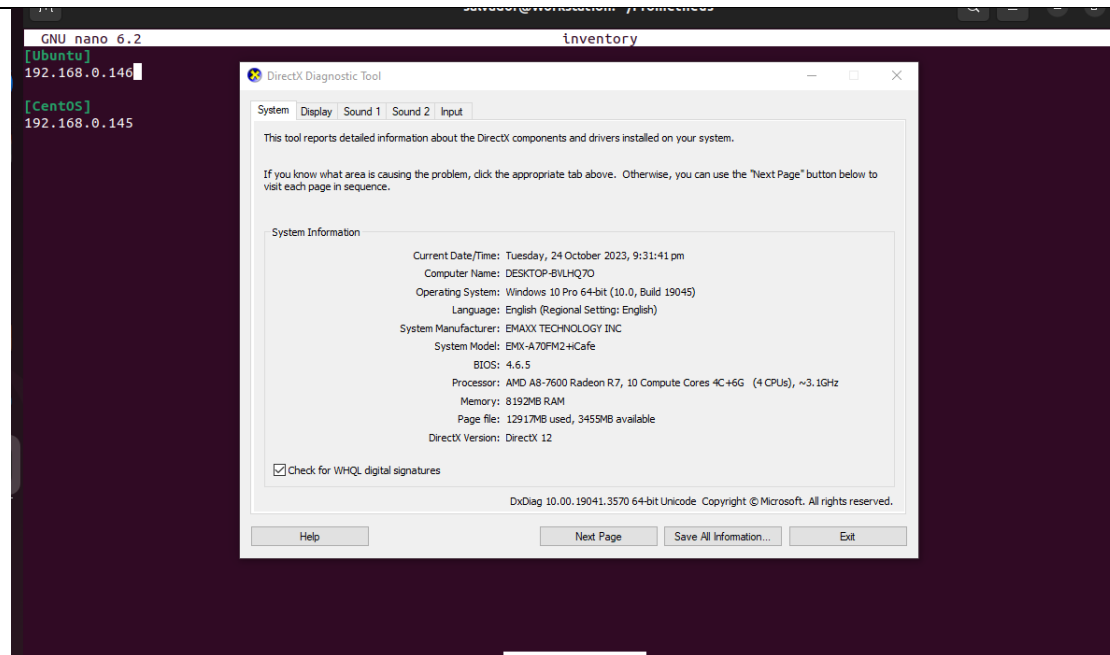
Your branch is up to date with 'origin/master'.
nothing to commit, working tree clean
salvador@Workstation:~/Prometheus$ git commit -m 'Initial commit'
[salvador@Workstation:~/Prometheus$ git push
salvador@Workstation:~/Prometheus$
ansible
CPE232_AndreuSalvador
salvador@Workstation:~/Prometheus$ git clone git@github.com:qajlsalvador/Prometheus.git
Cloning into 'Prometheus'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
salvador@Workstation:~/Prometheus$ ls
ansible Desktop Downloads id_rsa.pub Nagios Pictures Public snap Videos
CPE232_AndreuSalvador Documents id_rsa Music nagios-ansible Prometheus Salvador_PrelimExam Templates
salvador@Workstation:~/Prometheus$ cd Prometheus
salvador@Workstation:~/Prometheus$ sudo nano inventory
[sudo] password for salvador:

```

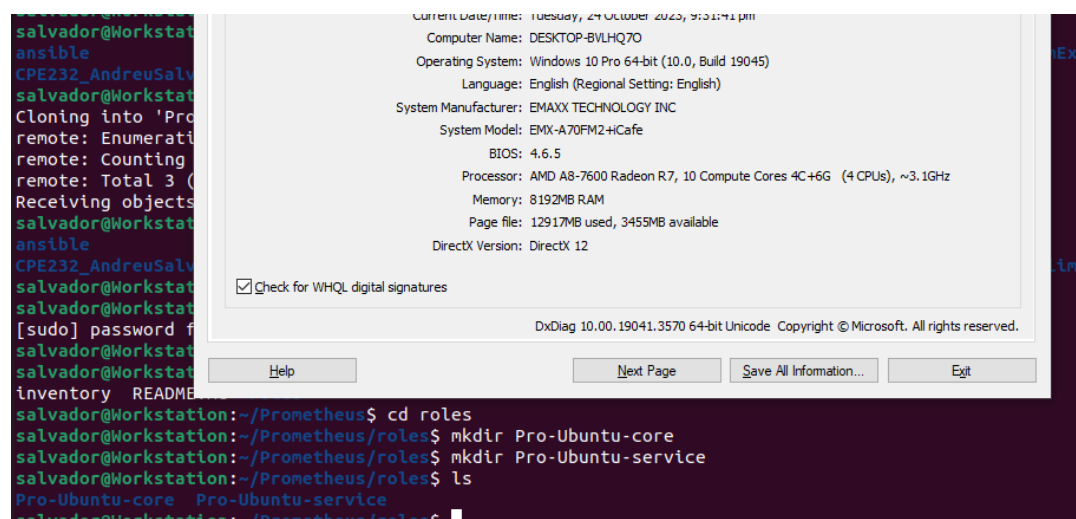
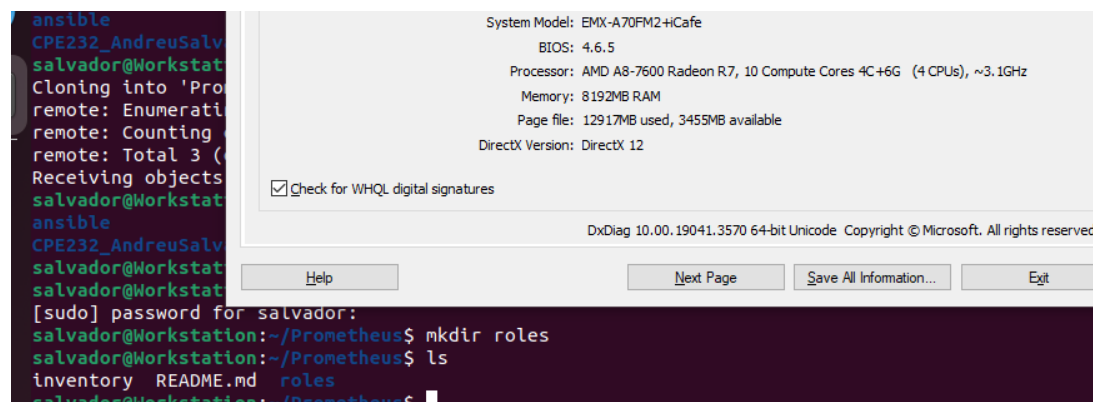
Operating System: Windows 10 Pro 64-bit (10.0, Build 19045)
 Language: English (Regional Setting: English)
 System Manufacturer: EMAXX TECHNOLOGY INC
 System Model: EMX-A70FM2-HiCafe
 BIOS: 4.6.5
 Processor: AMD A8-7600 Radeon R7, 10 Compute Cores 4C+6G (4 CPUs), ~3.1GHz
 Memory: 8192MB RAM
 Page file: 12917MB used, 3455MB available
 DirectX Version: DirectX 12

☒ Check for WHQL digital signatures

DxDiag 10.00.19041.3570 64-bit Unicode Copyright © Microsoft. All rights reserved.

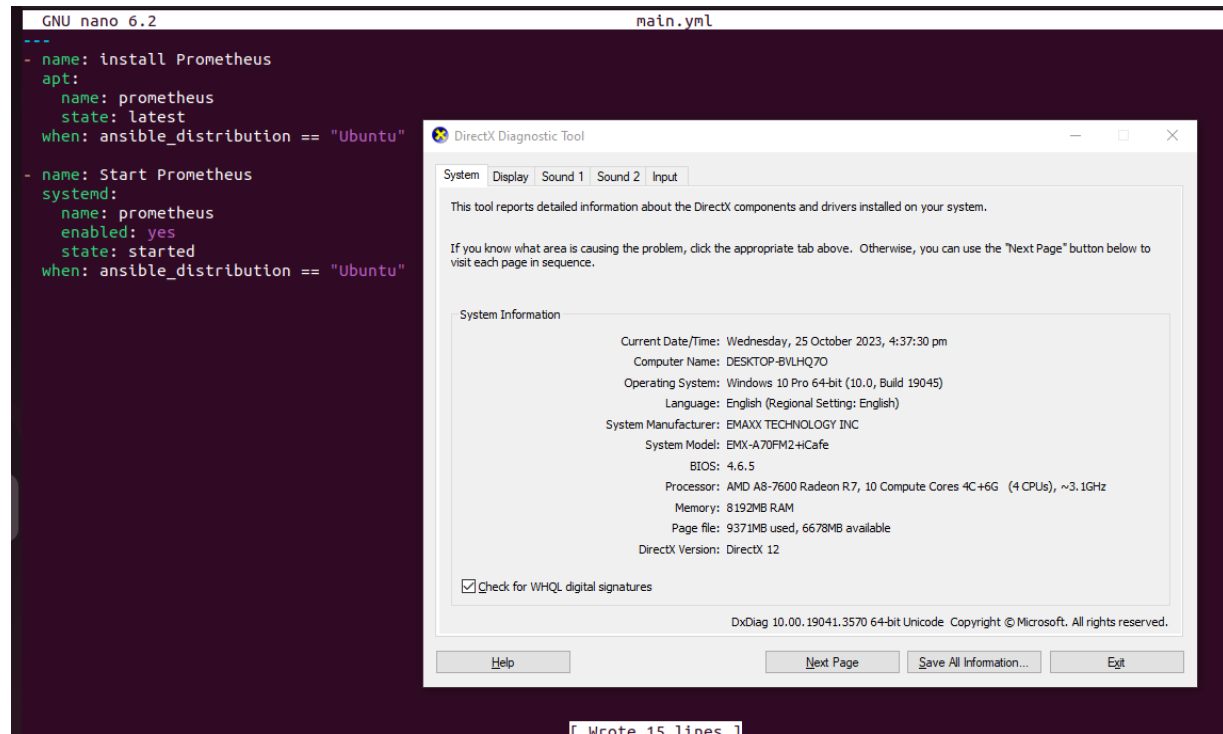


2. create roles directory and within the roles directory create the necessary roles for our Ubuntu server.

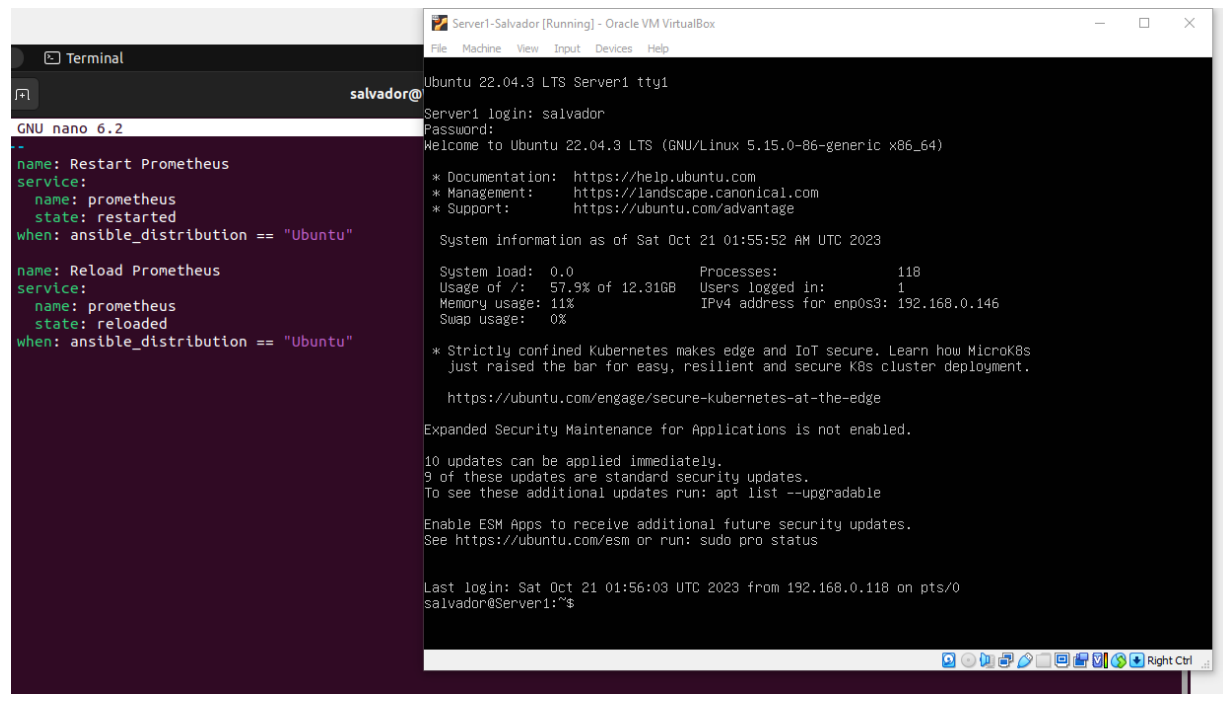


3. inside the roles, create a directory named tasks and inside the directory tasks create the .yml file as main.yml, lastly put the necessary tasks inside the .yml file.

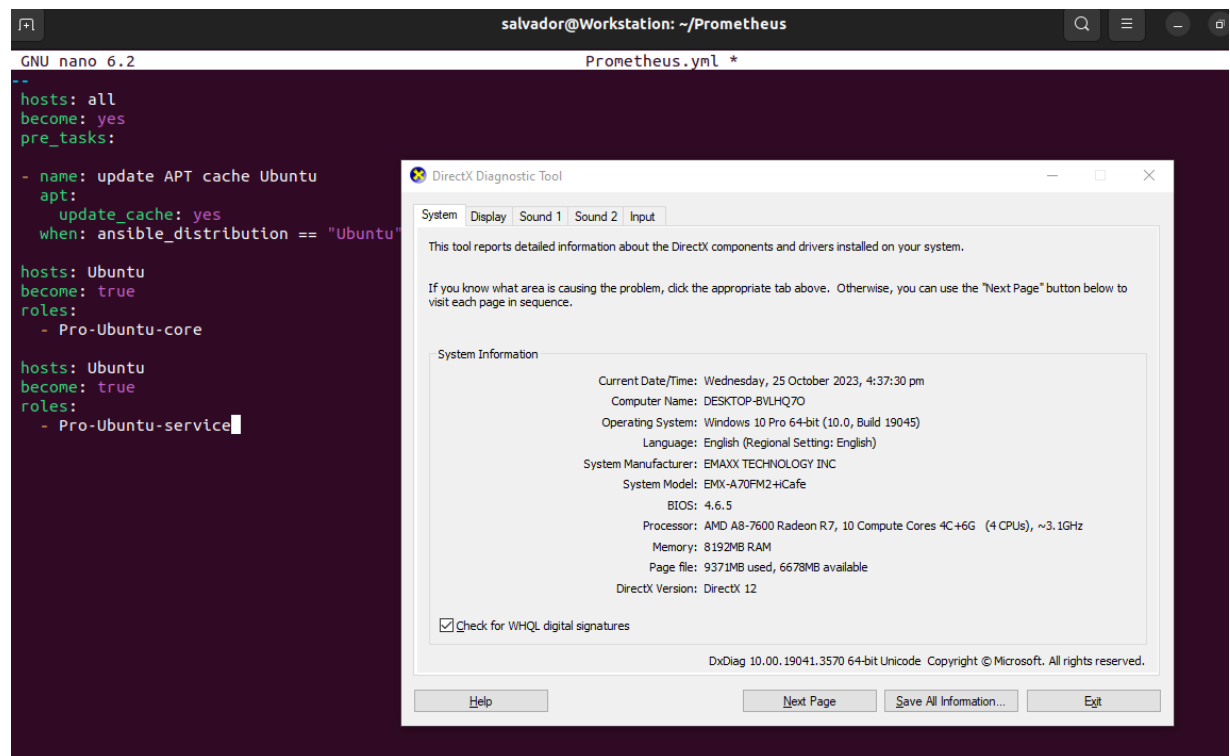
Pro-Ubuntu-core



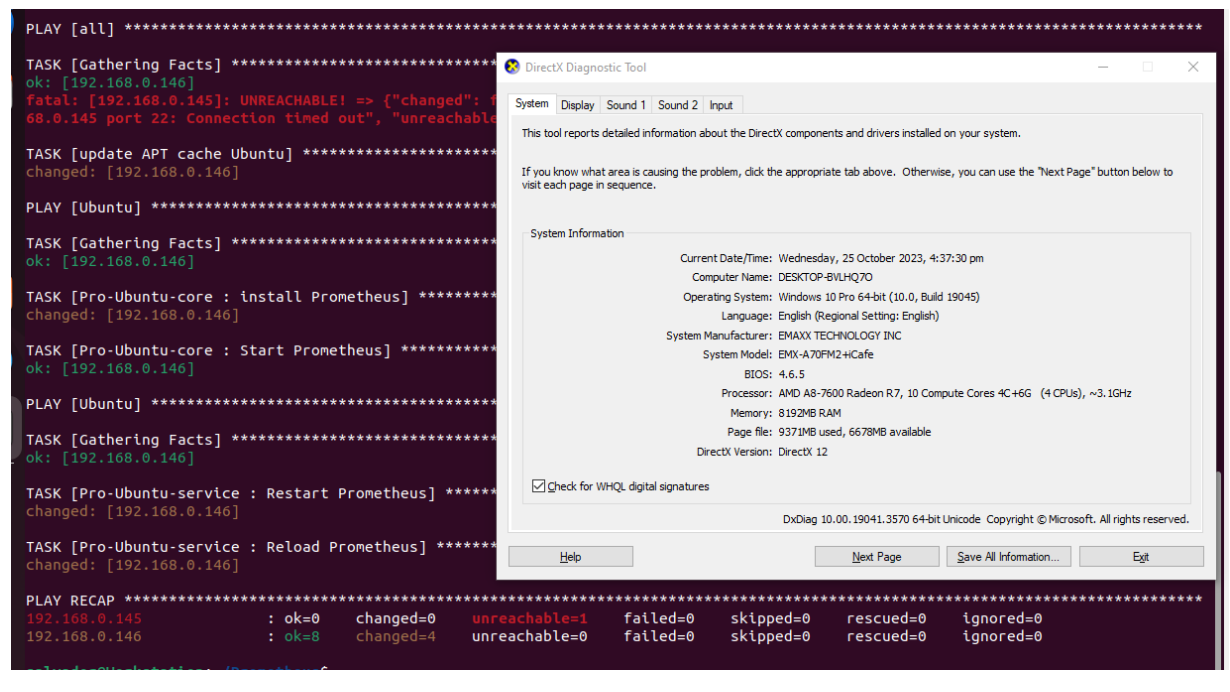
Pro-Ubuntu-service

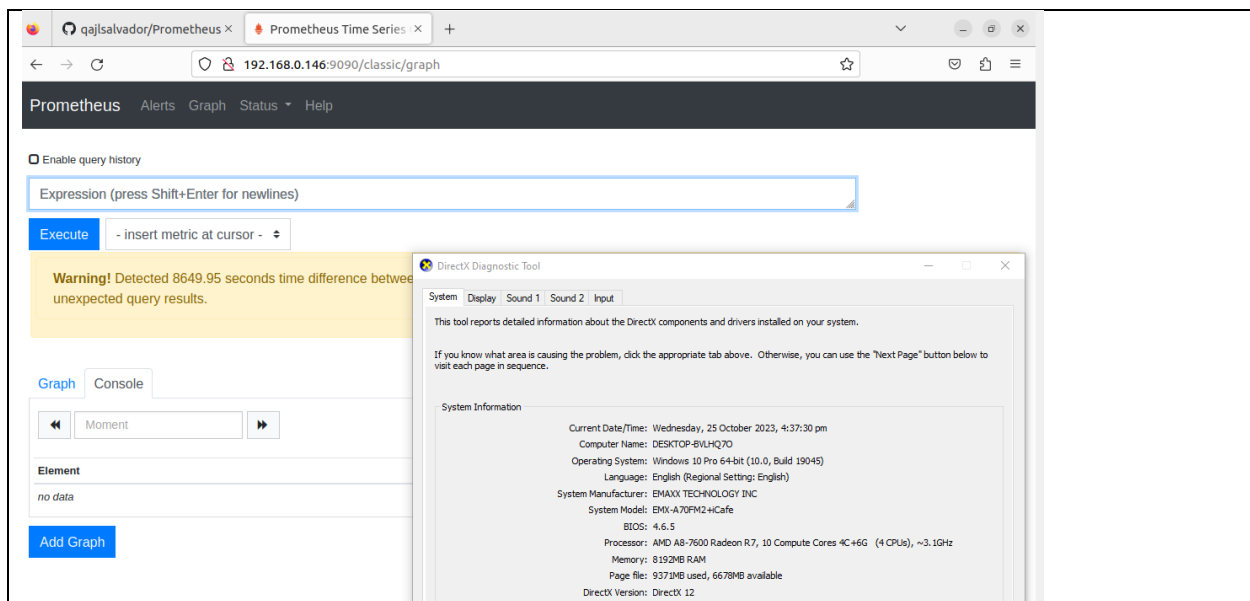


4. Create the main yml file to play the tasks.



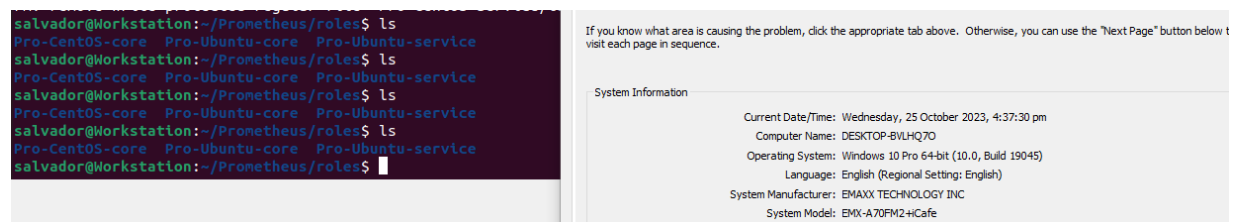
5. Run the playbook





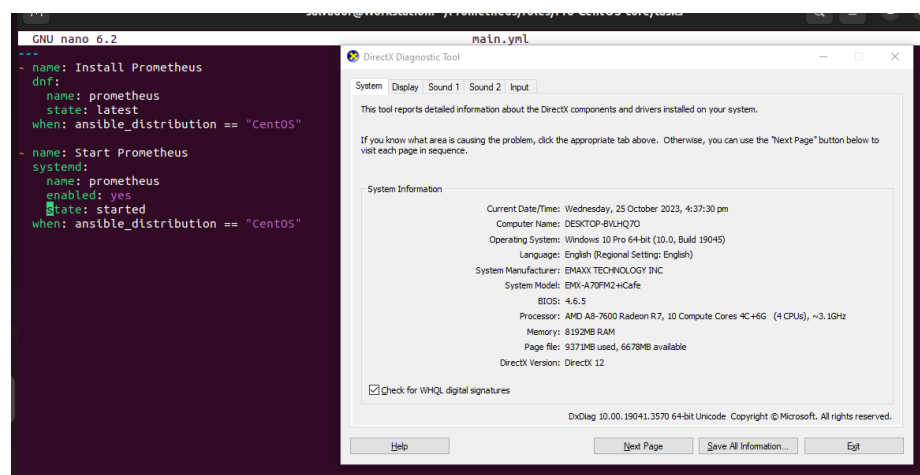
Part 3: Installation of Prometheus in CentOS server

1. Create the necessary roles in installing the Prometheus on CentOS

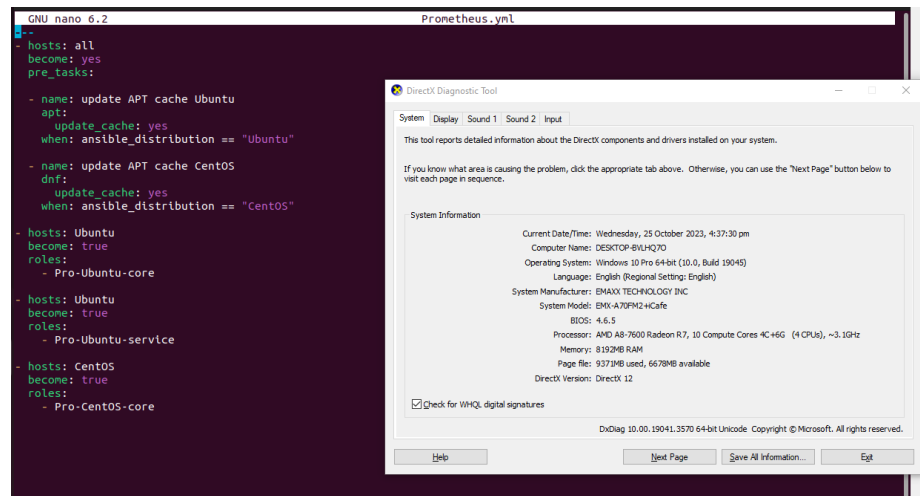


2. Just like on the part 2 inside the roles, create a directory named tasks and inside the directory tasks create the .yml file as main.yml, lastly put the necessary tasks inside the .yml file.

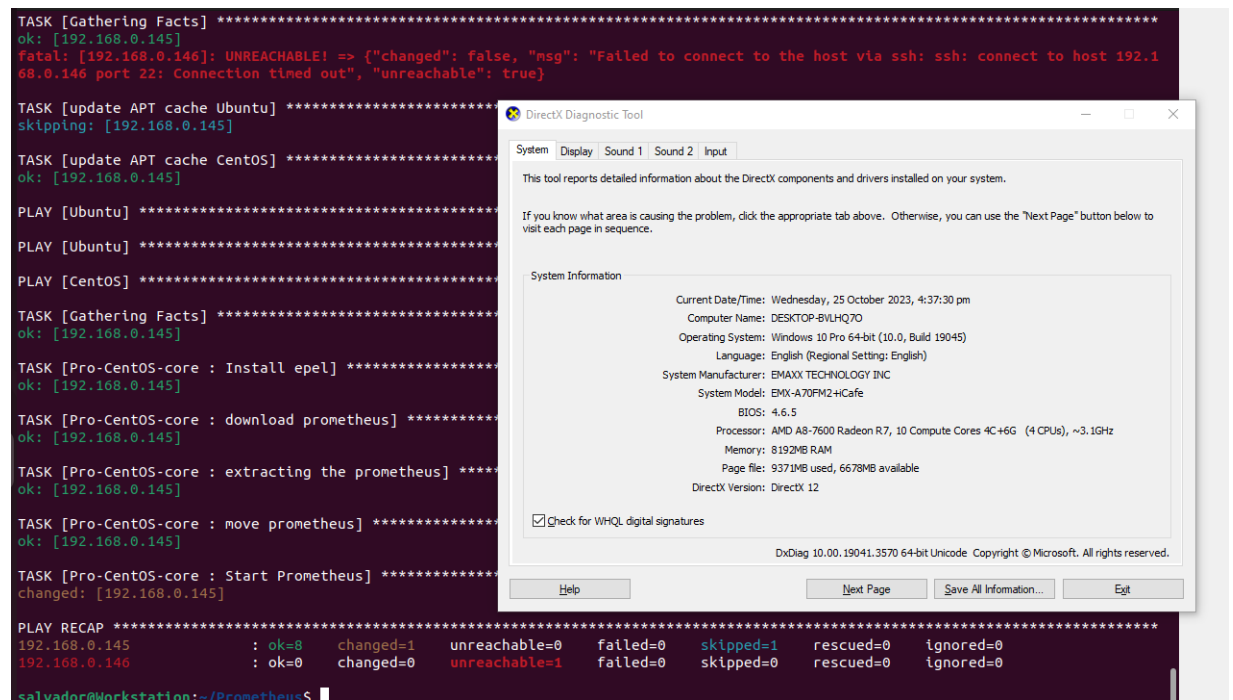
Pro-CentOS-core



3. Add the roles inside the main Prometheus .yml file in order to play it in the playbook.



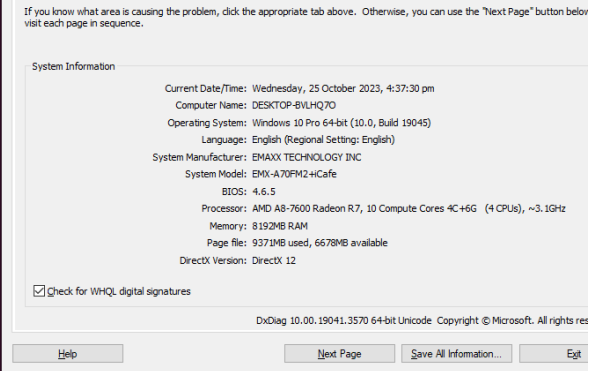
4. Run the Playbook



Part 4: adding everything to git repository

```
salvador@Workstation:~$ cd Prometheus
salvador@Workstation:~/Prometheus$ git add *
salvador@Workstation:~/Prometheus$ git -m commit "New "
unknown option: -m
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
      [--exec-path<=path>] [--html-path] [--man-path] [--info-path]
      [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
      [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
      [--super-prefix=<path>] [--config-env=<name>=<envvar>]
      <command> [<args>]
salvador@Workstation:~/Prometheus$ git add *
salvador@Workstation:~/Prometheus$ git commit -m "NEW entry"
[main 937019b] NEW entry
 5 files changed, 90 insertions(+)
 create mode 100644 Prometheus.yml
 create mode 100644 inventory
 create mode 100644 roles/Pro-CentOS-core/tasks/main.yml
 create mode 100644 roles/Pro-Ubuntu-core/tasks/main.yml
 create mode 100644 roles/Pro-Ubuntu-service/tasks/main.yml
salvador@Workstation:~/Prometheus$ git push origin main
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Compressing objects: 100% (7/7), done.
Writing objects: 100% (14/14), 1.51 KiB | 775.00 KiB/s, done.
Total 14 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:qajlsalvador/Prometheus.git
   0fcidd5..937019b  main -> main
salvador@Workstation:~/Prometheus$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
salvador@Workstation:~/Prometheus$
```



Reflections:

Answer the following:

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

It's a great thing to have in your arsenal as to monitor the performance of an application or something related. With this you are able to configure and monitor the errors and other functionality of that certain application / program. It's significance to your work is something that you could count on.

Conclusions:

In This activity we were tasked to implement the function of the roles inside a playbook in order to install a Performance Monitoring tool which is the Prometheus tool. In such manner that we can lessen the workload of the codes and tasks that will be implemented while doing the activity, roles are an important function to do so. With the implementation of the said function, I was able to manipulate certain moves like where to extract an archive that have been downloaded. This premise is what I did in order to install the monitoring app to my remote server which is the CentOS live server. Upon creating the playbook, I have faced a lot of difficulties especially in the part of installing the core of that Monitoring tool. In the end I was able to find a way and finished the activity. To Summarize all, this demonstrates the Installation of the Performance Monitoring Tool which is called Prometheus, by writing tasks inside the playbook I was able to manage the configuration of the said package.