

Name: Sonny Jay B. Bencito	Date Performed: 10/24/2022
Course/Section: CPE31S24	Date Submitted: 10/24/2022
Instructor: Dr. Jonathan Taylar	Semester and SY: 1st sem 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	
<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)	

Create a new repository

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

Owner * Bencitoo / Repository name * Bencito_Act9Prom ✓

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

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:
Skip this step if you're importing an existing repository.

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

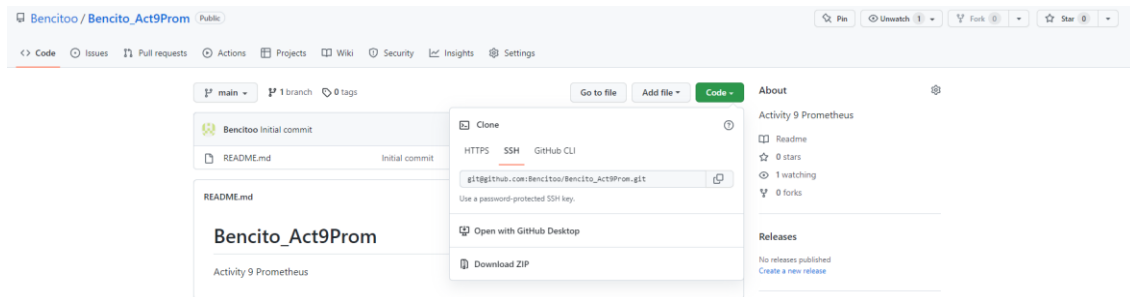
Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

This will set [main](#) as the default branch. Change the default name in your [settings](#).

ⓘ You are creating a public repository in your personal account.

[Create repository](#)



I create a repository on my github name Bencito Act9Prom

```
bencito@workstation: ~/Bencito_Act9Prom
bencito@workstation:~$ git clone git@github.com:Bencitoo/Bencito_Act9Prom.git
Cloning into 'Bencito_Act9Prom'...
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.
bencito@workstation:~$ cd Bencito_Act9Prom.git
bash: cd: Bencito_Act9Prom.git: No such file or directory
bencito@workstation:~$ cd Bencito_Act9Prom
bencito@workstation:~/Bencito_Act9Prom$
```

When it's done I git clone it to my manage node.

```
bencito@workstation:~/Bencito_Act9Prom$ nano inventory
bencito@workstation:~/Bencito_Act9Prom$ nano ansible.cfg
bencito@workstation:~/Bencito_Act9Prom$
```

```
GNU nano 6.2 inventory
[remote_servers]
192.168.56.101
192.168.56.105
```

```
bencito@workstation: ~/Bencito_Act9Prom
GNU nano 6.2 ansible.cfg
[defaults]

inventory = inventory
Host_key_checking = False

deprecation_warnings = False

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

```
bencito@workstation:~/Bencito_Act9Prom$ ansible -m ping all
192.168.56.105 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
192.168.56.101 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
bencito@workstation:~/Bencito_Act9Prom$
```

After when I git clone it. I create an inventory that inside of it was my two control nodes IP and the ansible.cfg that control it. After that I ping it and it was successful.

```
bencito@workstation: ~/Bencito_Act9Prom
GNU nano 6.2 prom.yml
---
- hosts: all
  become: true
  pre_tasks:

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

    - name: start httpd for Prometheus (CentOS)
      tags: apache, centos, httpd, Prometheus
      service:
        name: httpd
        state: started

- hosts: all
  become: true
  roles:
    - Prometheus
```

After that I create a prom.yml that will use a head playbook to install the Prometheus and the to update those repositories.

```
bencito@workstation:~/Bencito_Act9Prom/roles$ mkdir Prometheus
bencito@workstation:~/Bencito_Act9Prom/roles$ cd Prometheus
bencito@workstation:~/Bencito_Act9Prom/roles/Prometheus$ mkdir tasks
bencito@workstation:~/Bencito_Act9Prom/roles/Prometheus$ cd tasks
bencito@workstation:~/Bencito_Act9Prom/roles/Prometheus/tasks$ nano main.yml
```

After creating the main playbook. I create a directory name Prometheus to add there the task that install the Prometheus.

```
bencito@workstation: ~/Bencito_Act9Prom/roles/Promethe...
GNU nano 6.2 main.yml *
- name: Installation Prometheus (Ubuntu)
  tags: ubuntu, prometheus
  apt:
    name: prometheus
    state: latest
    when: ansible_distribution == "Ubuntu"

- name: Pre-req installation for (CentOS)
  tags: centos, snapd
  dnf:
    name:
      - snapd
    state: latest
    when: ansible_distribution == "CentOS"

- name: Enabling snapd for (CentOS)
  tags: snapd, centos
  command: systemctl enable --now snapd.socket
  when: ansible_distribution == "CentOS"

- name: Installation of Prometheus (CentOS)
  tags: prometheus, centos
  command: snap install prometheus --classic
  when: ansible_distribution == "CentOS"

^G Help      ^O Write Out  ^W Where Is   ^K Cut
^X Exit      ^R Read File  ^\ Replace    ^U Paste
```

As you can see here. I create a playbook name main.yml inside of it was the code that will install the Prometheus. I use the Snapd for the CentOS. It is like a playstore that have all the apps on linux. Because When I'm searching There is no direct Prometheus on CentOS.

```
bencito@workstation:~/Bencito_Act9Prom$ tree
.
├── ansible.cfg
├── inventory
├── prom.yml
├── README.md
├── roles
│   └── Prometheus
│       └── tasks
│           └── main.yml
└── 3 directories, 5 files
bencito@workstation:~/Bencito_Act9Prom$
```

After creating all of it. I use the command tree to show all the created directory if its correctly.

```
bencito@workstation:~/Bencito_Act9Prom$ ansible-playbook --ask-become-pass prom
.yml
BECOME password:
```

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

```
TASK [Gathering Facts] *****
*
ok: [192.168.56.101]
ok: [192.168.56.105]
```

```
TASK [update repository index (CentOS)] *****
*
skipping: [192.168.56.101]
ok: [192.168.56.105]
```

```
TASK [update repository index (Ubuntu)] *****
*
skipping: [192.168.56.105]
ok: [192.168.56.101]
```

```
TASK [start httpd for Prometheus (CentOS)] *****
*
skipping: [192.168.56.101]
changed: [192.168.56.105]
```

```
ok: [192.168.56.105]
ok: [192.168.56.101]

TASK [Prometheus : Installation Prometheus (Ubuntu)] *****
*
skipping: [192.168.56.105]
changed: [192.168.56.101]

TASK [Prometheus : Pre-req installation for (CentOS)] *****
*
skipping: [192.168.56.101]
changed: [192.168.56.105]

TASK [Prometheus : Enabling snapd for (CentOS)] *****
*
skipping: [192.168.56.101]
changed: [192.168.56.105]

TASK [Prometheus : Installation of Prometheus (CentOS)] *****
*
skipping: [192.168.56.101]
fatal: [192.168.56.105]: FAILED! => {"changed": true, "cmd": ["snap", "install", "prometheus", "--classic"], "delta": "0:00:00.834461", "end": "2022-10-24 20:46:21.649851", "msg": "non-zero return code", "rc": 10, "start": "2022-10-24 20:46:20.815390", "stderr": "error: too early for operation, device not yet seeded or device model not acknowledged", "stderr_lines": ["error: too early for operation, device not yet seeded or device model not acknowledged"], "stdout": "", "stdout_lines": []}
```

```
PLAY RECAP *****
*
192.168.56.101      : ok=4    changed=1    unreachable=0    failed=0
skipped=5    rescued=0    ignored=0
192.168.56.105      : ok=6    changed=3    unreachable=0    failed=1
skipped=2    rescued=0    ignored=0
```

As you can see here. I run the main playbook by using the ansible-playbook --ask-become-pass (name of the main playbook). When I run it on my first try there is an error on installing the Prometheus. Because the snapd and Prometheus did not see each other because they are same installing it. The snapd was not running because it was installing it and you need to re-run again the playbook. To read the new install snapd

```
bencito@workstation:~/Bencito_Act9Prom$ ansible-playbook --ask-become-pass prom
.yml
BECOME password:

PLAY [all] *****
*

TASK [Gathering Facts] *****
*
ok: [192.168.56.105]
ok: [192.168.56.101]

TASK [update repository index (CentOS)] *****
*
skipping: [192.168.56.101]
ok: [192.168.56.105]

TASK [update repository index (Ubuntu)] *****
*
skipping: [192.168.56.105]
ok: [192.168.56.101]

TASK [start httpd for Prometheus (CentOS)] *****
*
skipping: [192.168.56.101]
ok: [192.168.56.105]
```

```
TASK [Prometheus : Installation Prometheus (Ubuntu)] *****
*
skipping: [192.168.56.105]
ok: [192.168.56.101]

TASK [Prometheus : Pre-req installation for (CentOS)] *****
*
skipping: [192.168.56.101]
ok: [192.168.56.105]

TASK [Prometheus : Enabling snapd for (CentOS)] *****
*
skipping: [192.168.56.101]
changed: [192.168.56.105]

TASK [Prometheus : Installation of Prometheus (CentOS)] *****
*
skipping: [192.168.56.101]
changed: [192.168.56.105]

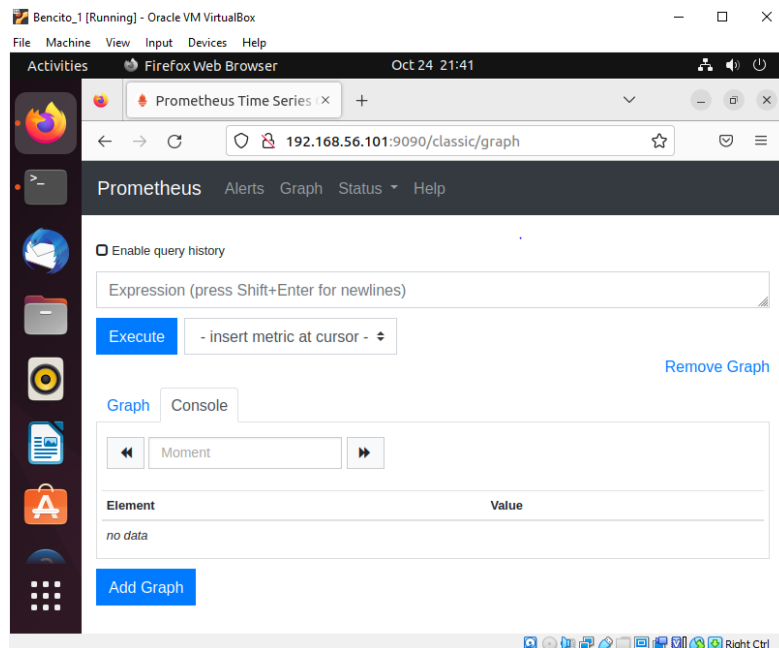
PLAY RECAP *****
*
192.168.56.101      : ok=4    changed=0    unreachable=0    failed=0
skipped=5    rescued=0    ignored=0
192.168.56.105      : ok=7    changed=2    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0

bencito@workstation:~/Bencito_Act9Prom$
```

I re-run again and it was successfully installed because. The snapd was read on my CentOS.

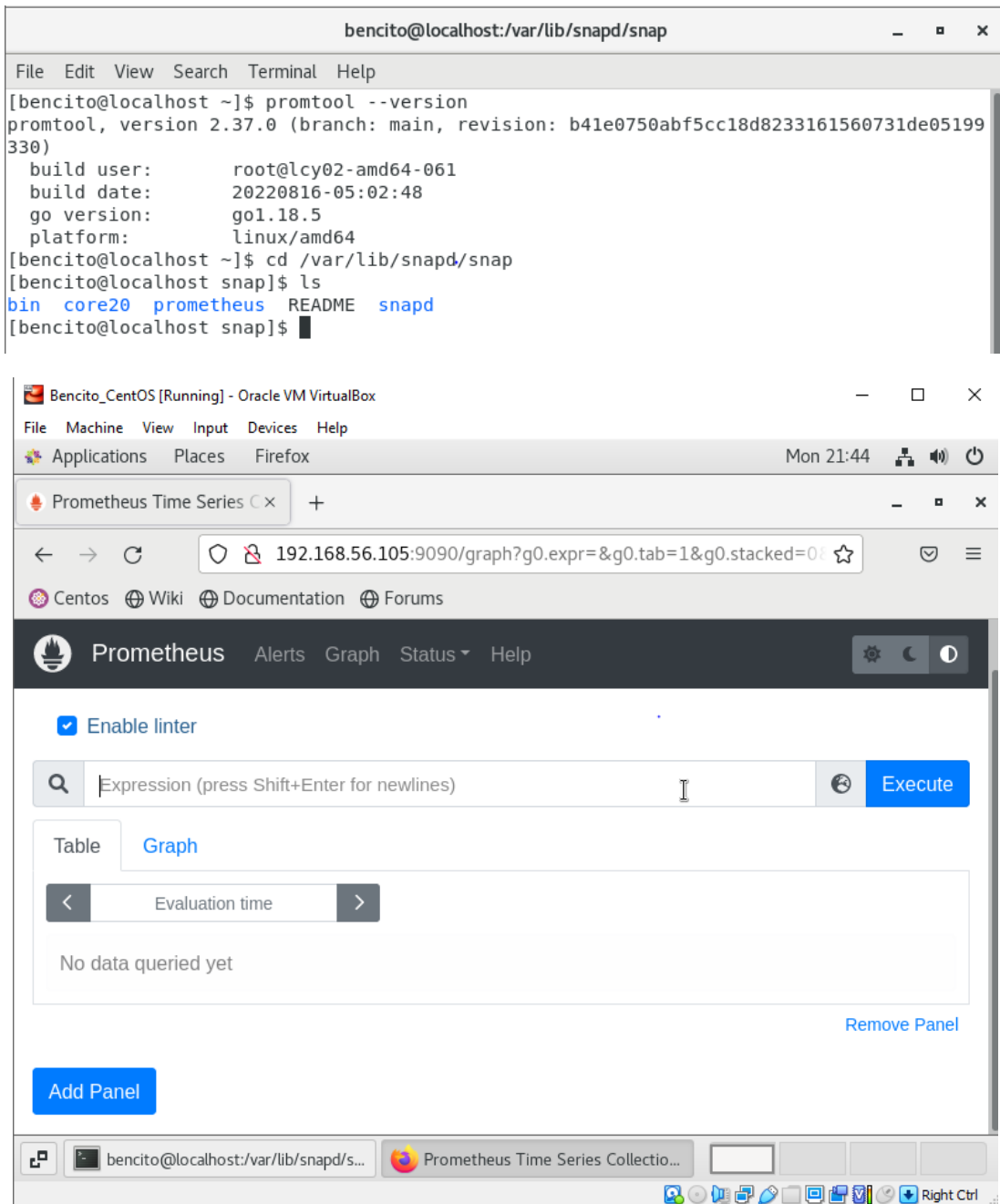
Output Ubuntu

```
bencito@Server1: ~  
bencito@Server1:~$ prometheus --version  
prometheus, version 2.31.2+ds1 (branch: debian/sid, revision: 2.31.2+ds1-1ubuntu1)  
  build user:      team+pkg-go@tracker.debian.org  
  build date:      20220317-16:26:29  
  go version:      go1.17.3  
  platform:        linux/amd64  
bencito@Server1:~$ promtool --version  
promtool, version 2.31.2+ds1 (branch: debian/sid, revision: 2.31.2+ds1-1ubuntu1)  
  build user:      team+pkg-go@tracker.debian.org  
  build date:      20220317-16:26:29  
  go version:      go1.17.3  
  platform:        linux/amd64  
bencito@Server1:~$
```



As you can see. It was successfully installed on my ubuntu.

Output CentOS



The image shows two screenshots. The top screenshot is a terminal window titled 'bencito@localhost:/var/lib/snapd/snap'. It shows the command 'promtool --version' being executed, which returns 'promtool, version 2.37.0 (branch: main, revision: b41e0750abf5cc18d8233161560731de05199330)'. Below this, the build details are shown: 'build user: root@lcy02-amd64-061', 'build date: 20220816-05:02:48', 'go version: go1.18.5', and 'platform: linux/amd64'. Then, the user navigates to '/var/lib/snapd/snap' and runs 'ls', which lists 'bin', 'core20', 'prometheus', 'README', and 'snapd'. The bottom screenshot is a web browser window titled 'Bencito_CentOS [Running] - Oracle VM VirtualBox'. It shows the Prometheus Time Series interface at '192.168.56.105:9090/graph?g0.expr=&g0.tab=1&g0.stacked=0'. The interface includes a search bar, a 'Execute' button, and a 'Graph' tab. The status bar at the bottom shows the terminal window and the Prometheus Time Series Collector.

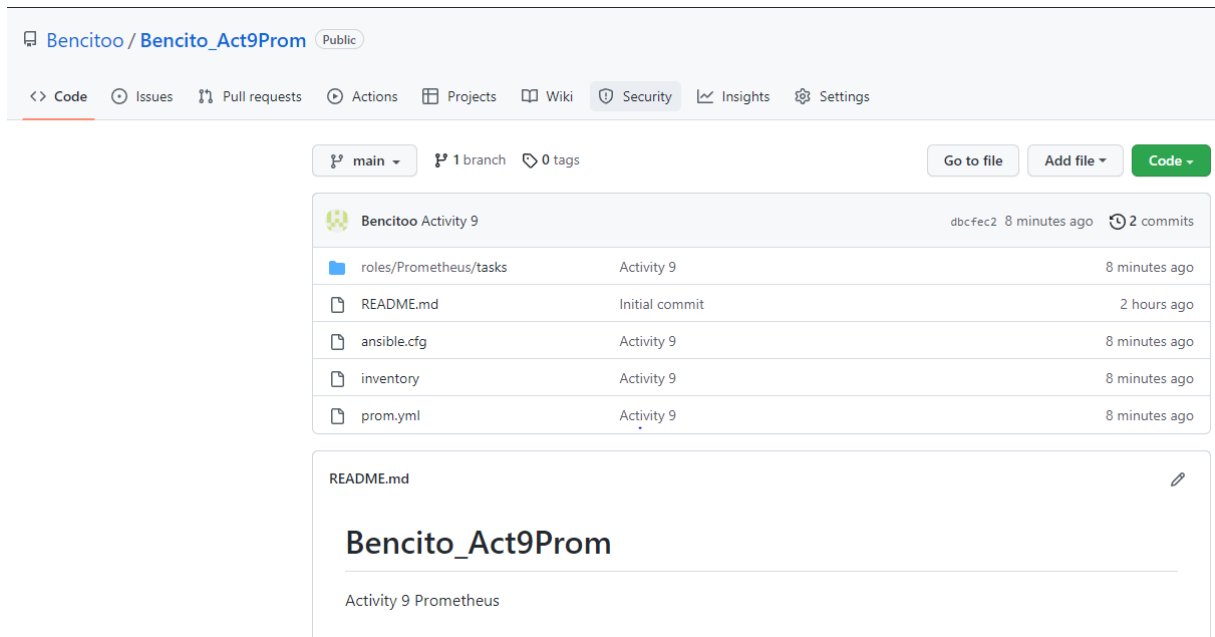
```
bencito@localhost:/var/lib/snapd/snap
File Edit View Search Terminal Help
[bencito@localhost ~]$ promtool --version
promtool, version 2.37.0 (branch: main, revision: b41e0750abf5cc18d8233161560731de05199330)
  build user:      root@lcy02-amd64-061
  build date:      20220816-05:02:48
  go version:      go1.18.5
  platform:        linux/amd64
[bencito@localhost ~]$ cd /var/lib/snapd/snap
[bencito@localhost snap]$ ls
bin core20 prometheus README snapd
[bencito@localhost snap]$
```

Bencito_CentOS [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Firefox Mon 21:44
Prometheus Time Series C x +
192.168.56.105:9090/graph?g0.expr=&g0.tab=1&g0.stacked=0
Centos Wiki Documentation Forums
Prometheus Alerts Graph Status Help
Enable linter
Expression (press Shift+Enter for newlines) Execute
Table Graph
Evaluation time
No data queried yet
Remove Panel
Add Panel
bencito@localhost:/var/lib/snapd/s... Prometheus Time Series Collectio...
Right Ctrl

As you can see. It was successfully installed on my CentOS and the Prometheus is inside of the snapd. Because I use the snapD to install the Prometheus.

Add to Github

```
bencito@workstation:~/Bencito_Act9Prom$ git add inventory
bencito@workstation:~/Bencito_Act9Prom$ git add ansible.cfg
bencito@workstation:~/Bencito_Act9Prom$ git add prom.yml
bencito@workstation:~/Bencito_Act9Prom$ git add roles/
bencito@workstation:~/Bencito_Act9Prom$ git commit -m "Activity 9"
[main dbcfe2] Activity 9
 4 files changed, 69 insertions(+)
 create mode 100644 ansible.cfg
 create mode 100644 inventory
 create mode 100644 prom.yml
 create mode 100644 roles/Prometheus/tasks/main.yml
bencito@workstation:~/Bencito_Act9Prom$ git push
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (9/9), 1.14 KiB | 72.00 KiB/s, done.
Total 9 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:Bencitoo/Bencito_Act9Prom.git
 8d623ff..dbcfe2  main -> main
bencito@workstation:~/Bencito_Act9Prom$
```



Bencitoo / Bencito_Act9Prom Public

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags Go to file Add file Code

Bencitoo Activity 9 dbcfe2 8 minutes ago 2 commits

File	Commit	Time
roles/Prometheus/tasks	Activity 9	8 minutes ago
README.md	Initial commit	2 hours ago
ansible.cfg	Activity 9	8 minutes ago
inventory	Activity 9	8 minutes ago
prom.yml	Activity 9	8 minutes ago

README.md

Bencito_Act9Prom

Activity 9 Prometheus

After successfully installed the Prometheus. I do the git add and git push to make sure it was saved on my new repository on GitHub.

Reflections:

Answer the following:

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

There so many benefits of performance monitoring tool. It helps you to monitor realtime check of the data security and the health of the computer system. It can do accelerate your computer system to sleep and disable the unnecessary application.

Also, I use the performance monitoring tool to check my hardware accessories like my hard drive. If it's running smoothly and if there is a problem.

Conclusions:

In my conclusion. After making this activity I learn that how to create a playbook that will install the given task. Like when I'm doing the installation of prometheus it is the same on the last activity you need to create a roles and input there the playbook code. But I have encountered some error like the installation of the Prometheus using the snapd. Because you need first install the snapd before the Prometheus. We need to wait some minutes to re-run again the playbook for us to read the snapd on our CentOS. After that It was smoothly installed on both control nodes and I realize that you need to familiarize those codes. Because it will help you to the next activity or on your future works. I'm happy when it was successfully run and I don't encounter some error.