Name: Clarence B. Chavez	**Date Performed:** November 6, 2024
Course/Section: CPE 212/CPE31S2	**Date Submitted:** November 6, 2024
Instructor: Engr. Robin Valenzuela	**Semester and SY:** 2nd Sem - 2024-2025

Midterm Skills Exam: Install, Configure, and Manage Log Monitoring tools

1. Objectives

Create and design a workflow that installs, configure and manage enterprise availability, performance and log monitoring tools using Ansible as an Infrastructure as Code (IaC) tool.

2. Instructions

1. Create a repository in your GitHub account and label it **CPE_MIDEXAM_SURNAME**.

2. Clone the repository and do the following:
   2.1. Create an Ansible playbook that does the following with an input of a **config.yaml** file and **arranged Inventory** file:

   2.2. Install and configure **Elastic Stack** in separate hosts (Elastic Search, Kibana, Logstash) • Install Nagios in one host

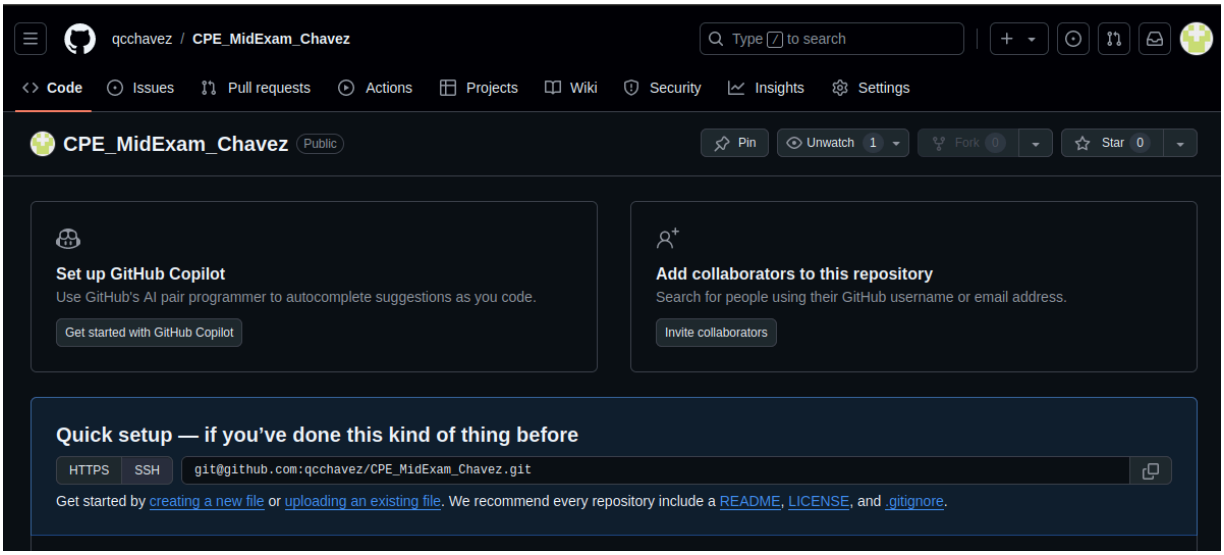
   2.3. Install **Grafana, Prometheus** and **Influxdb** in separate hosts (Influxdb,Grafana,Prometheus)

   2.4. Install **Lamp Stack** in separate hosts (Httpd + Php,Mariadb)

3. Document all your tasks using this document. Provide proofs of all the ansible playbooks codes and successful installations.
4. Document the push and commit from the local repository to GitHub.
5. Finally, paste also the link of your GitHub repository in the documentation.

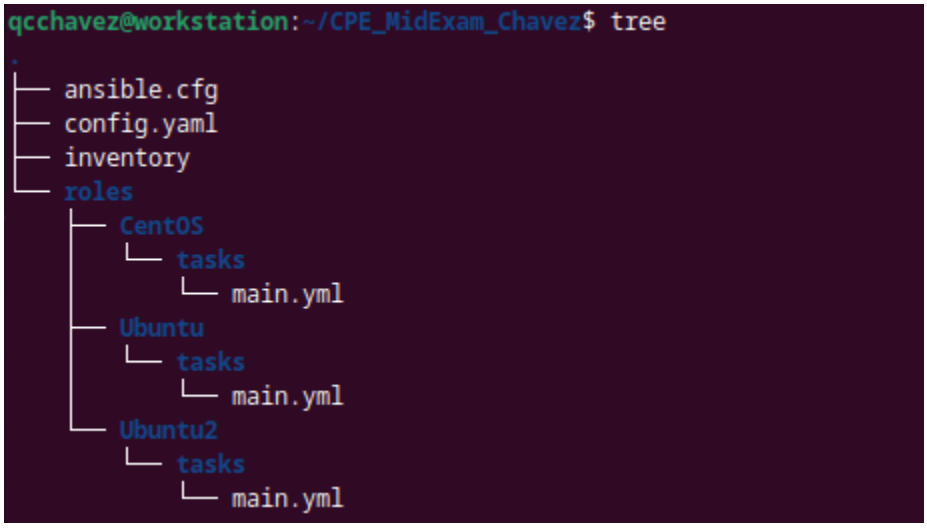
3. **Output** (screenshots and explanations)

Task 1

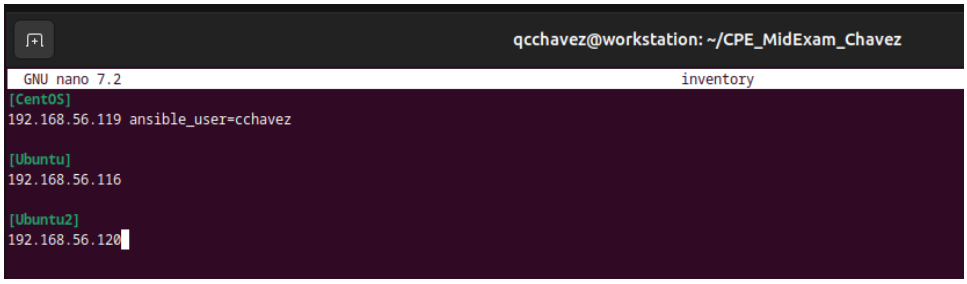


- This screenshot showcases the creation of the GitHub repository before doing the necessary tasks of the Midterm Examination.

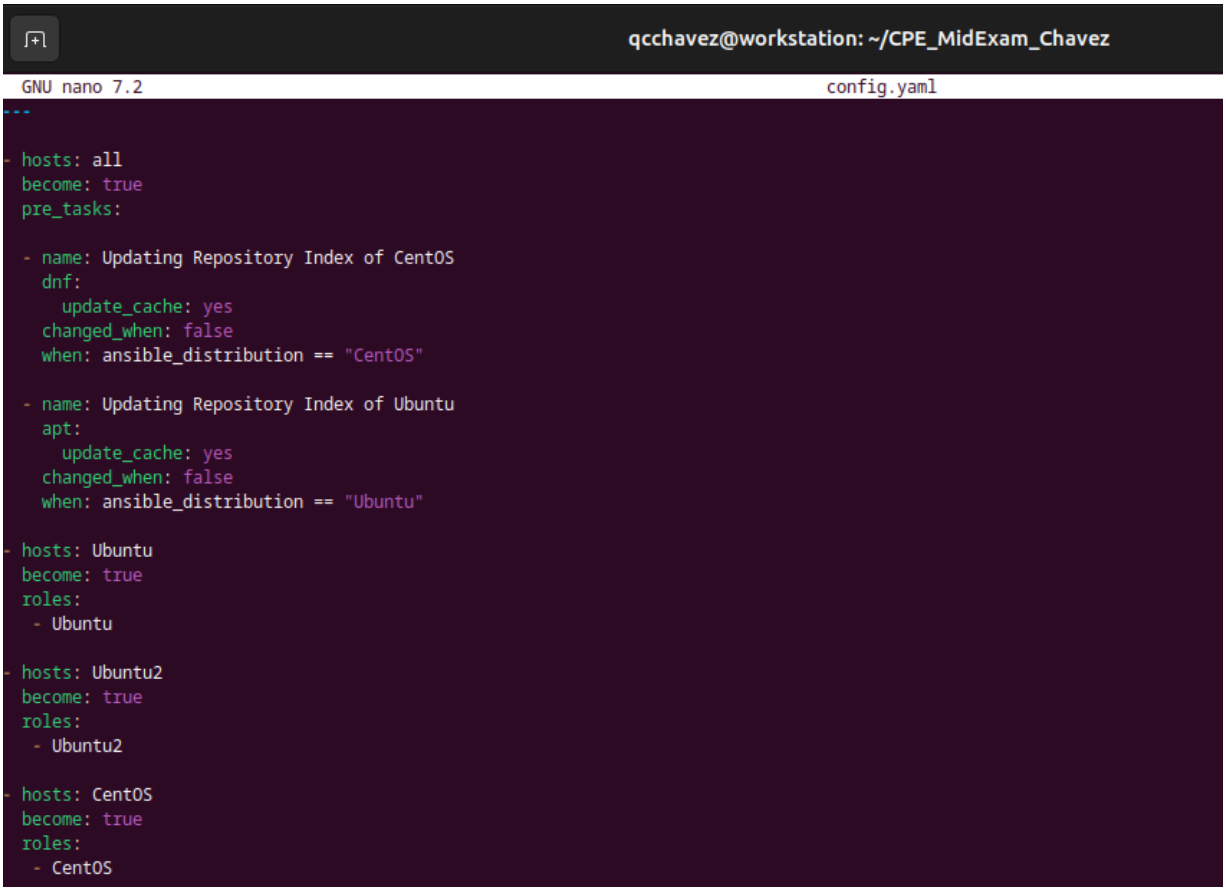
Task 2.1



- This screenshot showcases the file content of the GitHub repository for my Midterm Examination. It includes files like **ansible.cfg**, **config.yaml**, **inventory file**, etc.



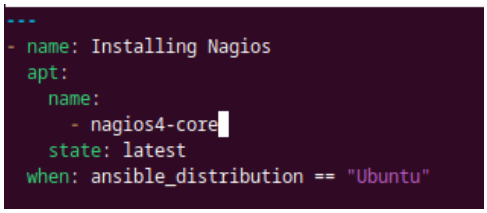
- This screenshot showcases the content for the inventory file, it shows the IP addresses of my remote servers.



- This screenshot showcases the file content of my config.yaml, firstly, it updates the repository indexes and also includes the directory names of my remote servers for roles category.

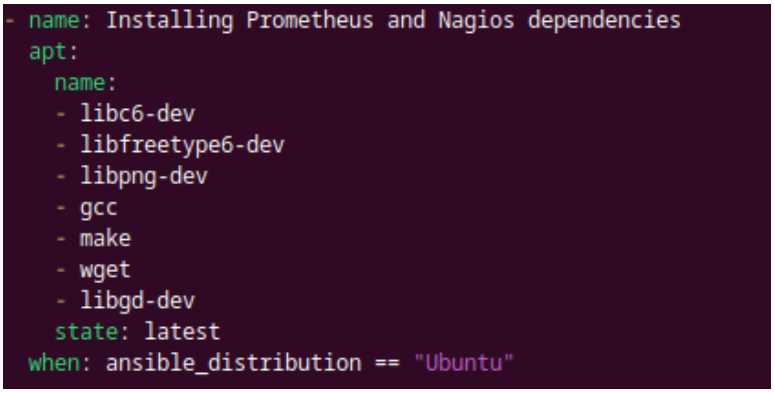
Task 2.2 Install and configure Elastic Stack in separate hosts (Elastic Search, Kibana, Logstash), Install Nagios in one host

Ubuntu (Nagios)



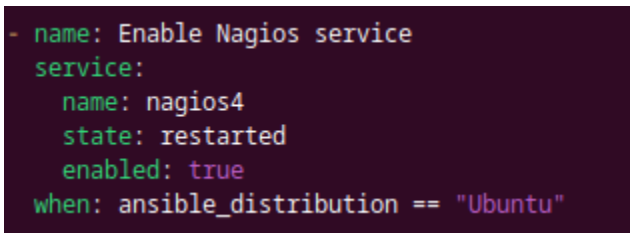
```yaml
---
- name: Installing Nagios
  apt:
    name:
      - nagios4-core
    state: latest
  when: ansible_distribution == "Ubuntu"
```

- This screenshot showcases the required code for installing **Nagios**.



```yaml
- name: Installing Prometheus and Nagios dependencies
  apt:
    name:
    - libc6-dev
    - libfreetype6-dev
    - libpng-dev
    - gcc
    - make
    - wget
    - libgd-dev
    state: latest
  when: ansible_distribution == "Ubuntu"
```

- This screenshot showcases the code for installing the required dependencies for both **Prometheus** and **Nagios**. Without these, neither of the two services would work.



```yaml
- name: Enable Nagios service
  service:
    name: nagios4
    state: restarted
    enabled: true
  when: ansible_distribution == "Ubuntu"
```

- This screenshot showcases the code for enabling the Nagios service on the remote server.

```
TASK [Ubuntu : Installing Nagios] ***********************************************************************************
skipping: [192.168.56.119]
ok: [192.168.56.116]
```

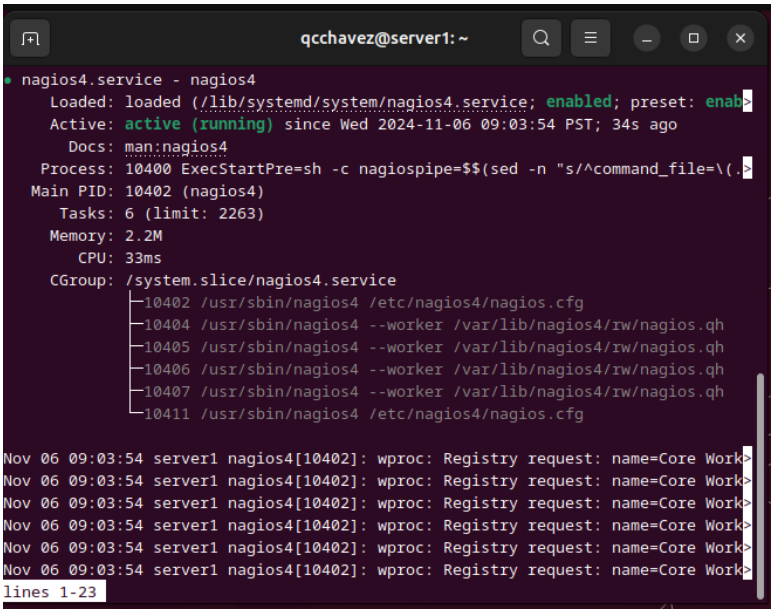
- This screenshot showcases that the task for **Nagios** installation works for the target remote server.

```
TASK [Ubuntu : Installing Prometheus and Nagios dependencies] *******************************************************
skipping: [192.168.56.119]
ok: [192.168.56.116]
```

- This screenshot showcases that the task for **Nagios and Prometheus** dependencies installation works for the target remote server.

```
TASK [Ubuntu : Enable Nagios service] *******************************************************************************
skipping: [192.168.56.119]
changed: [192.168.56.116]
```

- This screenshot showcases that the task for enabling the **Nagios** service works for the target remote server.



- This screenshot showcases that the **Nagios4** is indeed working on the remote server.

Task 2.3 Install Grafana,Prometheus and Influxdb in seperate hosts (Influxdb,Grafana,Prometheus)

Ubuntu (Prometheus)

```
- name: Installing Prometheus
  apt:
    name: prometheus
    state: latest
  when: ansible_distribution == "Ubuntu"
```

- This screenshot showcases the required code for installing **Prometheus**.

```
- name: Installing Prometheus and Nagios dependencies
  apt:
    name:
    - libc6-dev
    - libfreetype6-dev
    - libpng-dev
    - gcc
    - make
    - wget
    - libgd-dev
    state: latest
  when: ansible_distribution == "Ubuntu"
```

- This screenshot showcases the code for installing the required dependencies for both **Prometheus** and **Nagios**. Without these, neither of the two services would work.

```
- name: Enable Prometheus service
  service:
    name: prometheus
    state: restarted
    enabled: true
  when: ansible_distribution == "Ubuntu"
```

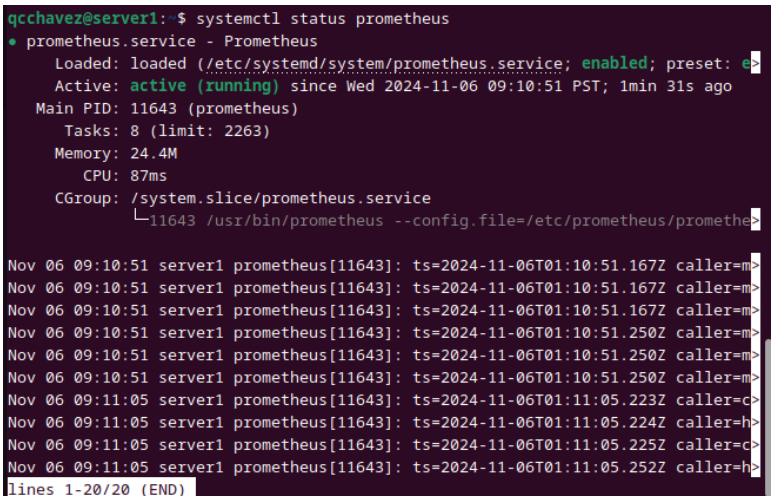
- This screenshot showcases the code for enabling the **Prometheus** service on the remote server.

```
TASK [Ubuntu : Installing Prometheus] ******************************************************************************
skipping: [192.168.56.119]
ok: [192.168.56.116]
```

- This screenshot showcases that the task **Prometheus** installation works for the target remote server.

```
TASK [Ubuntu : Installing Prometheus and Nagios dependencies] ******************************************************
skipping: [192.168.56.119]
ok: [192.168.56.116]
```

- This screenshot showcases that the task **Nagios and Prometheus** dependencies installation works for the target remote server.



- This screenshot showcases that the **Prometheus** is working on the remote server.

Ubuntu2 (Grafana)

```
qcchavez@workstation: ~/CPE_MidExam_Chavez/roles/Ubuntu2/tasks
 GNU nano 7.2                                  main.yml
---
- name: Install Grafana Agent Flow
  apt:
    name: grafana
    state: latest
  when: ansible_distribution == "Ubuntu"

- name: Enable Grafana service
  service:
    name: grafana-server
    state: restarted
    enabled: true
  when: ansible_distribution == "Ubuntu"
```

CentOS (InfluxDB)

Task 2.4 Install Lamp Stack in separate hosts (Httpd + Php,Mariadb)

Ubuntu (HTTPD + PHP)

```
- name: Installing HTTPD and PHP
  package:
    name:
      - apache2
      - libapache2-mod-php
    state: latest
  when: ansible_distribution == "Ubuntu"
```

- This screenshot showcases the code for installing **HTTPD and PHP**.

```
TASK [Ubuntu : Installing HTTPD and PHP] ***************************************************************************
skipping: [192.168.56.119]
ok: [192.168.56.116]
```

- This screenshot showcases that the task for **HTTPD and PHP** installation works for the target remote server.

```
TASK [Ubuntu : Enable HTTPD and PHP service] ***********************************************************************
skipping: [192.168.56.119]
changed: [192.168.56.116]
```

- This screenshot showcases that the task for Enabling **HTTPD and PHP** services works.

```
qcchavez@server1:~$ systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enab>
     Active: active (running) since Wed 2024-11-06 09:10:52 PST; 2min 1s ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 11720 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/S>
   Main PID: 11725 (apache2)
      Tasks: 6 (limit: 2263)
     Memory: 10.3M
        CPU: 39ms
     CGroup: /system.slice/apache2.service
             ├─11725 /usr/sbin/apache2 -k start
             ├─11730 /usr/sbin/apache2 -k start
             ├─11731 /usr/sbin/apache2 -k start
             ├─11732 /usr/sbin/apache2 -k start
             ├─11733 /usr/sbin/apache2 -k start
             └─11734 /usr/sbin/apache2 -k start

Nov 06 09:10:52 server1 systemd[1]: Starting apache2.service - The Apache HTTP >
Nov 06 09:10:52 server1 apachectl[11724]: AH00558: apache2: Could not reliably >
Nov 06 09:10:52 server1 systemd[1]: Started apache2.service - The Apache HTTP S>
lines 1-20/20 (END)
```

- This screenshot showcases that the **HTTPD and PHP** is properly working on the remote server.

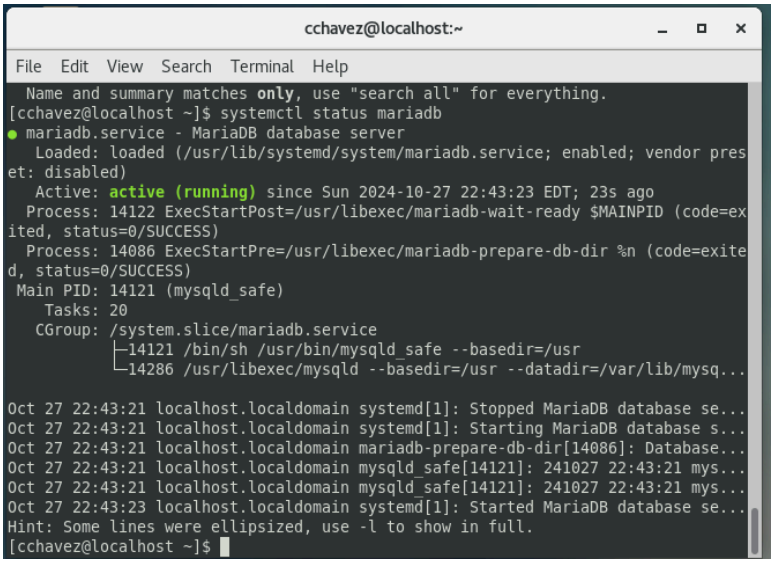
CentOS (MariaDB)

```yaml
---
- name: Installing MariaDB
  yum:
    name: mariadb-server
    state: latest
  when: ansible_distribution == "CentOS"

- name: Enabling MariaDB
  service:
    name: mariadb
    state: restarted
    enabled: true
  when: ansible_distribution == "CentOS"
```

- This screenshot showcases the code for installing and enabling **MariaDB**

```
TASK [CentOS : Installing MariaDB] *********************************************************************************************
skipping: [192.168.56.116]
ok: [192.168.56.119]

TASK [CentOS : Enabling MariaDB] ***********************************************************************************************
skipping: [192.168.56.116]
changed: [192.168.56.119]
```

- This screenshot showcases that the task for **MariaDB** installation and enabling the service works for the target remote server.

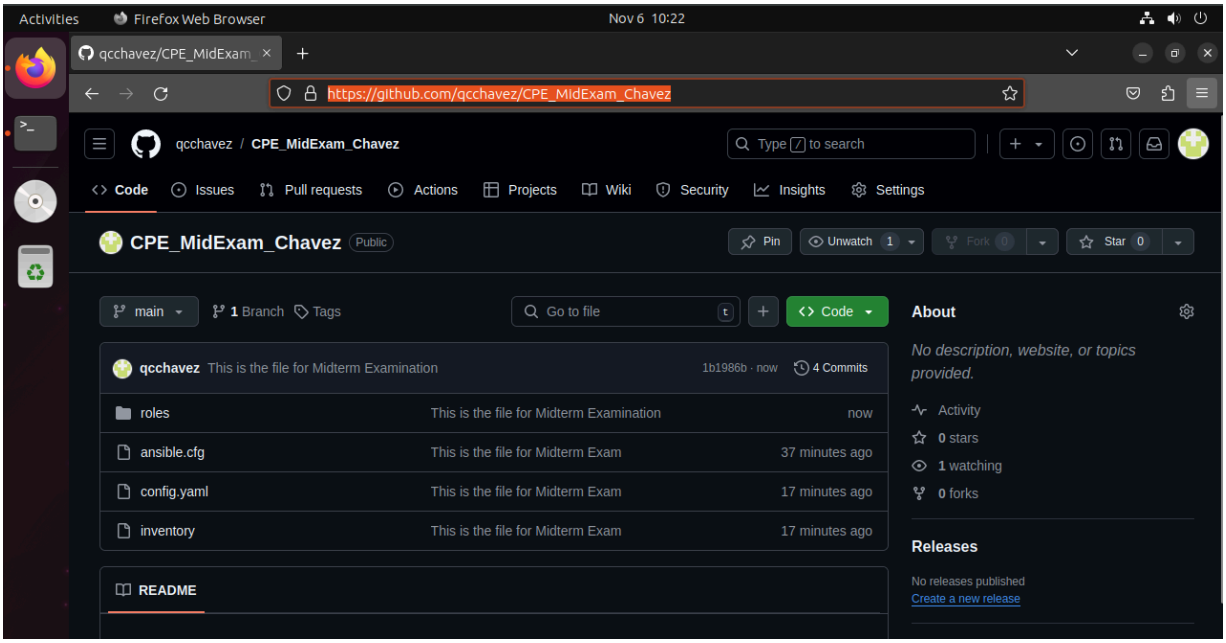


- This screenshot showcases that **MariaDB** is working well on the remote server.

Task 4. Document the push and commit from the local repository to GitHub.

```
qcchavez@workstation:~/CPE_MidExam_Chavez$ git add .
qcchavez@workstation:~/CPE_MidExam_Chavez$ git commit -m "This is the file for Midterm Examination"
[main 1b1986b] This is the file for Midterm Examination
 2 files changed, 14 insertions(+), 16 deletions(-)
qcchavez@workstation:~/CPE_MidExam_Chavez$ git push origin main
Enumerating objects: 17, done.
Counting objects: 100% (17/17), done.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (9/9), 747 bytes | 747.00 KiB/s, done.
Total 9 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:qcchavez/CPE_MidExam_Chavez.git
   258ce6e..1b1986b  main -> main
qcchavez@workstation:~/CPE_MidExam_Chavez$
```

- In this screenshot, I've pushed and committed from my local repository **CPE_Midterm_Exam** to Github.



- In this screenshot, it shows that the necessary files from the local repository were successfully committed to Github.

GitHub link:

https://www.github.com/qcchavez/CPE_MidExam_Chavez

Conclusions: (link your conclusion from the objective)

- In this Midterm Examination, I have learned that installing, configuring, and managing enterprise availability, performance, and log monitoring tools using Ansible is a really great method to install on the remote servers. I can say that this is a great method because for me, it's easier to navigate the files whenever you need to make changes. And since it's easier to navigate the files for me as an administrator, it's much more convenient when making adjustments on the packages that are needed for the remote servers, considering that availability, performance, and log monitoring tools play an important part when managing servers. For example, Nagios is a tool that lets you detect and repair problems as soon as possible, and also, reduces the possible future issues that you might encounter before they will be able to affect the end users. With this tool, you can plan ahead on several infrastructure upgrades, especially updating outdated systems that can possibly provide failures in the near future.