Name: Seruelas, Ronn Kristoper H.	**Date Performed:** 11-06-2023
Course/Section: CPE 232 - CPE31S4	**Date Submitted:** 11-06-2023
Instructor: Dr. Jonathan V. Taylar	**Semester and SY:**

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.

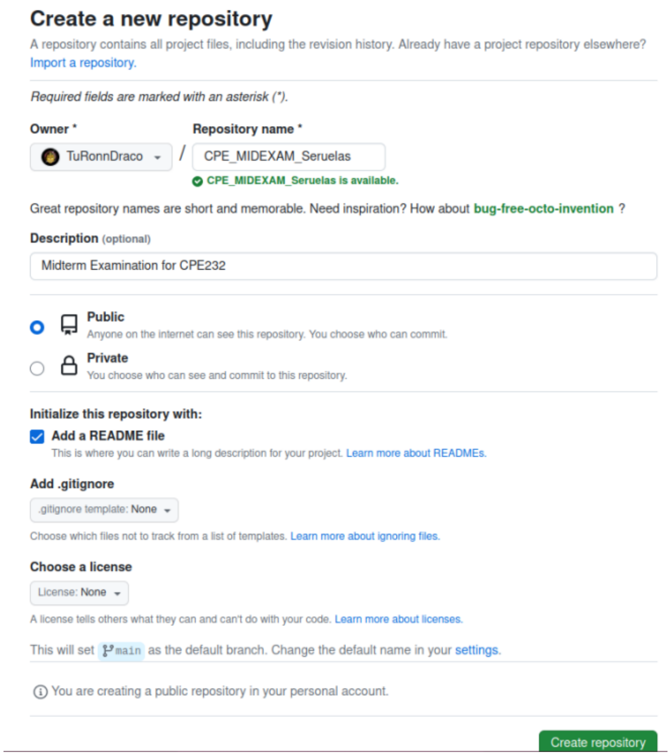
2. Instructions

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 seperate 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: Preparation of Repository

1. Create a Github Repository named **CPE_MIDEXAM_SURNAME**, then clone it to the local repository of the workstation.



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 * TuRonnDraco / **Repository name *** CPE_MIDEXAM_Seruelas

CPE_MIDEXAM_Seruelas is available.

Great repository names are short and memorable. Need inspiration? How about bug-free-octo-invention ?

Description (optional)

Midterm Examination for CPE232

● **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

.gitignore template: None

Choose which files not to track from a list of templates. Learn more about ignoring files.

Choose a license

License: None

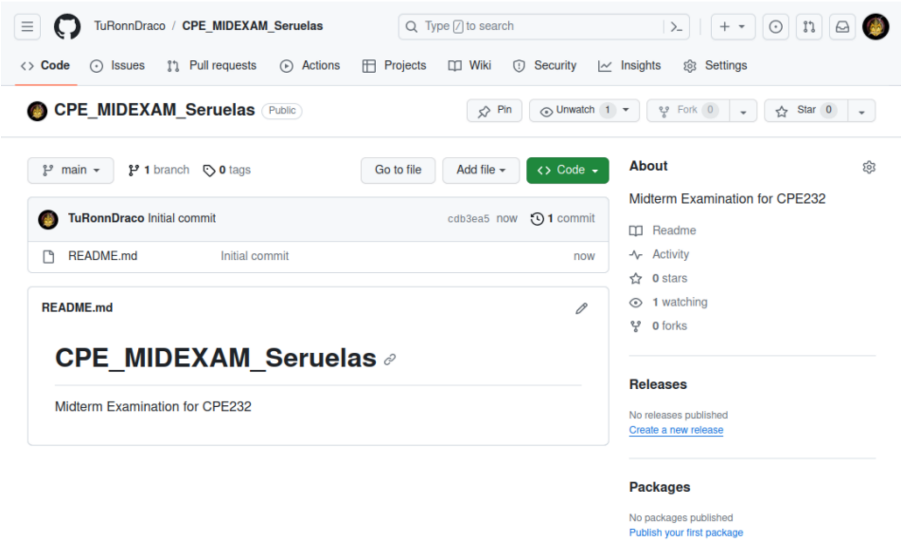
A license tells others what they can and can't do with your code. Learn more about licenses.

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

Figure 1.1.1 - Creation of the Github Repository.



TuRonnDraco / **CPE_MIDEXAM_Seruelas**

Code · Issues · Pull requests · Actions · Projects · Wiki · Security · Insights · Settings

CPE_MIDEXAM_Seruelas Public

main · 1 branch · 0 tags · Go to file · Add file · Code

TuRonnDraco Initial commit · cdb3ea5 now · 1 commit

README.md · Initial commit · now

README.md

CPE_MIDEXAM_Seruelas

Midterm Examination for CPE232

About

Midterm Examination for CPE232

Readme · Activity · 0 stars · 1 watching · 0 forks

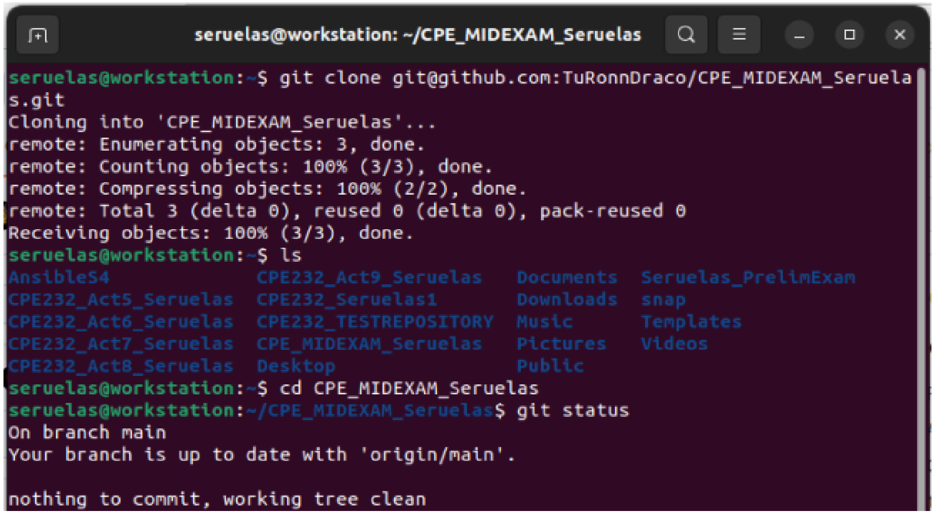
Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

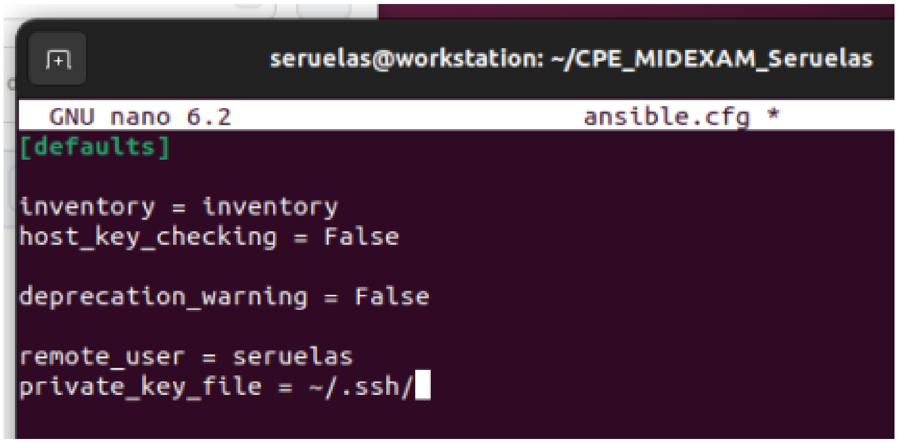
Figure 1.1.2 - Github Repository, CPE_MIDEXAM_Seruelas.



```
seruelas@workstation:~$ git clone git@github.com:TuRonnDraco/CPE_MIDEXAM_Seruela
s.git
Cloning into 'CPE_MIDEXAM_Seruelas'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
seruelas@workstation:~$ ls
AnsibleS4             CPE232_Act9_Seruelas     Documents  Seruelas_PrelimExam
CPE232_Act5_Seruelas  CPE232_Seruelas1         Downloads  snap
CPE232_Act6_Seruelas  CPE232_TESTREPOSITORY    Music      Templates
CPE232_Act7_Seruelas  CPE_MIDEXAM_Seruelas     Pictures   Videos
CPE232_Act8_Seruelas  Desktop                  Public
seruelas@workstation:~$ cd CPE_MIDEXAM_Seruelas
seruelas@workstation:~/CPE_MIDEXAM_Seruelas$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

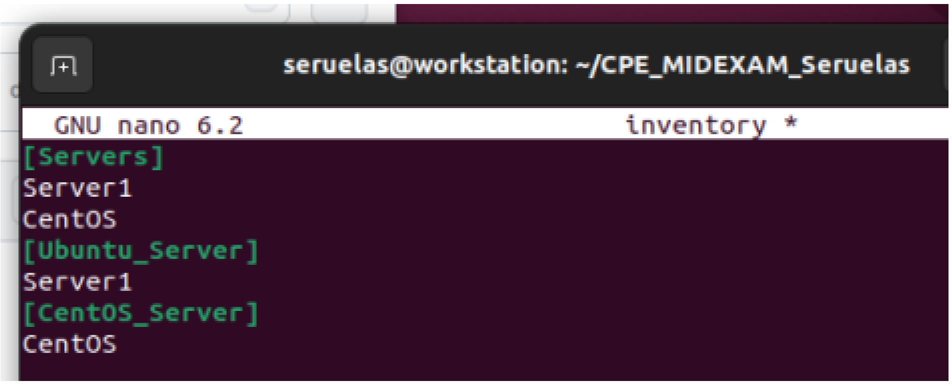
Figure 1.1.3 - Cloning of the Github repository to the local repository of the workstation.

2. Upon the cloning of the repository, configure the **ansible.cfg** and the **inventory** of the local repository.



```
[defaults]

inventory = inventory
host_key_checking = False

deprecation_warning = False

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

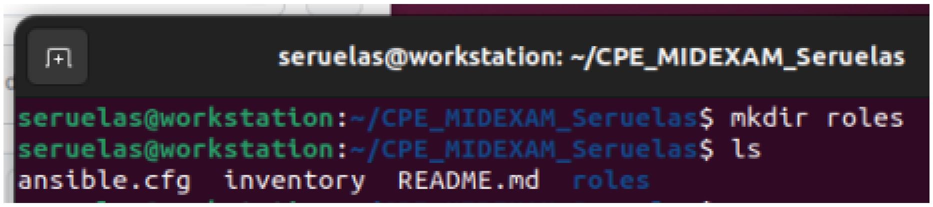
Figure 1.2.1 - ansible.cfg of the repository.



```
[Servers]
Server1
CentOS
[Ubuntu_Server]
Server1
[CentOS_Server]
CentOS
```

Figure 1.2.2 - inventory configuration of the repository.

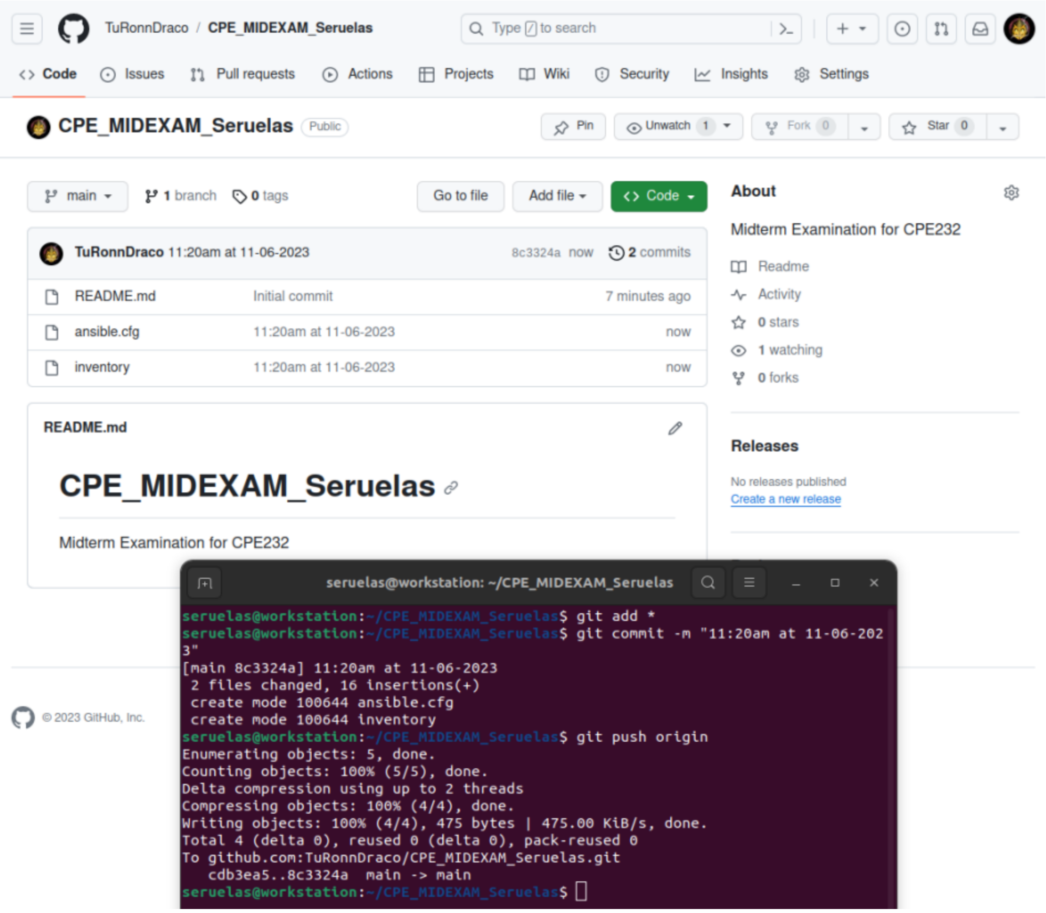
3. Create the **roles** directory that will be used for the organization and the specification of tasks for each server.



```
seruelas@workstation:~/CPE_MIDEXAM_Seruelas$ mkdir roles
seruelas@workstation:~/CPE_MIDEXAM_Seruelas$ ls
ansible.cfg  inventory  README.md  roles
```

Figure 1.3.1 - Creation of the **roles** directory.

4. Save all changes and push to the Github Repository.



Midterm Examination for CPE232

TuRonnDraco 11:20am at 11-06-2023		8c3324a now 2 commits
README.md	Initial commit	7 minutes ago
ansible.cfg	11:20am at 11-06-2023	now
inventory	11:20am at 11-06-2023	now

README.md

CPE_MIDEXAM_Seruelas

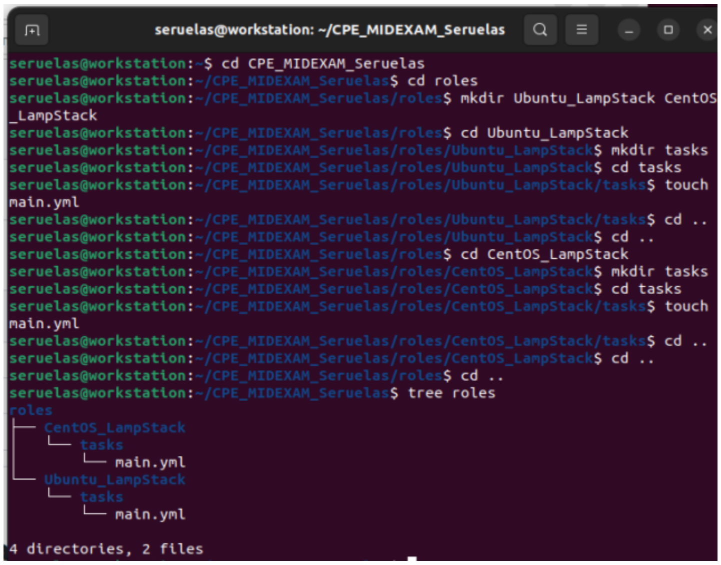
Midterm Examination for CPE232

```
seruelas@workstation:~/CPE_MIDEXAM_Seruelas$ git add *
seruelas@workstation:~/CPE_MIDEXAM_Seruelas$ git commit -m "11:20am at 11-06-202
3"
[main 8c3324a] 11:20am at 11-06-2023
 2 files changed, 16 insertions(+)
 create mode 100644 ansible.cfg
 create mode 100644 inventory
seruelas@workstation:~/CPE_MIDEXAM_Seruelas$ git push origin
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 2 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 475 bytes | 475.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:TuRonnDraco/CPE_MIDEXAM_Seruelas.git
   cdb3ea5..8c3324a  main -> main
seruelas@workstation:~/CPE_MIDEXAM_Seruelas$ 
```

Figure 1.4.1 - Saving all changes done in the local repository and pushing it to the Github Repository.

Task 2: Installation of Lamp Stack in Ubuntu and CentOS

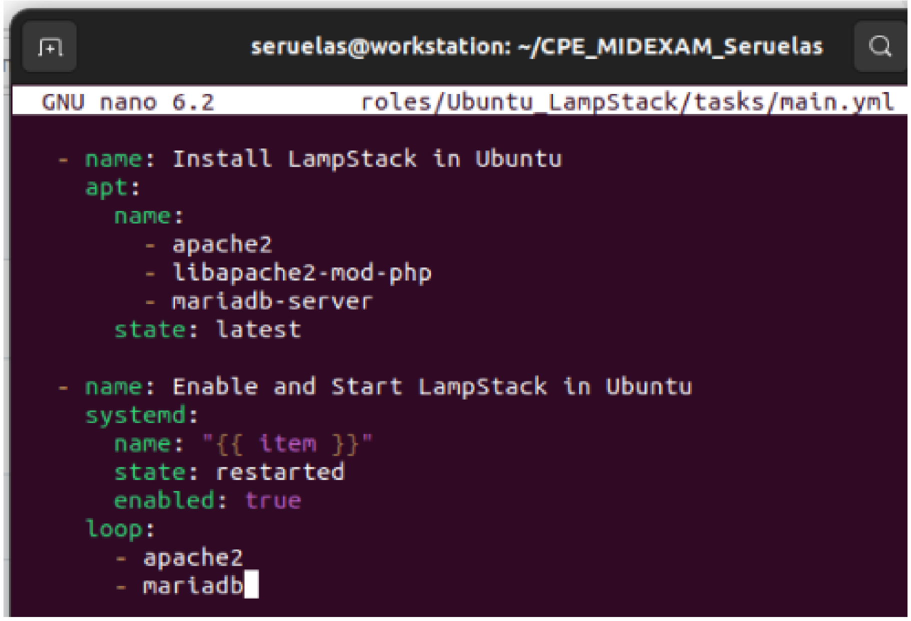
1. For preparation, create the dedicated roles for both the Ubuntu and CentOS for the installation of the LampStack (Httpd, PHP and MariaDB).



```
seruelas@workstation:~$ cd CPE_MIDEXAM_Seruelas
seruelas@workstation:~/CPE_MIDEXAM_Seruelas$ cd roles
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles$ mkdir Ubuntu_LampStack CentOS
_LampStack
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles$ cd Ubuntu_LampStack
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles/Ubuntu_LampStack$ mkdir tasks
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles/Ubuntu_LampStack$ cd tasks
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles/Ubuntu_LampStack/tasks$ touch
main.yml
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles/Ubuntu_LampStack/tasks$ cd ..
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles/Ubuntu_LampStack$ cd ..
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles$ cd CentOS_LampStack
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles/CentOS_LampStack$ mkdir tasks
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles/CentOS_LampStack$ cd tasks
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles/CentOS_LampStack/tasks$ touch
main.yml
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles/CentOS_LampStack/tasks$ cd ..
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles/CentOS_LampStack$ cd ..
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles$ cd ..
seruelas@workstation:~/CPE_MIDEXAM_Seruelas$ tree roles
roles
├── CentOS_LampStack
│   └── tasks
│       └── main.yml
└── Ubuntu_LampStack
    └── tasks
        └── main.yml

4 directories, 2 files
```

Figure 2.1.1 - Creation of the roles for the installation of LampStack of both machines.

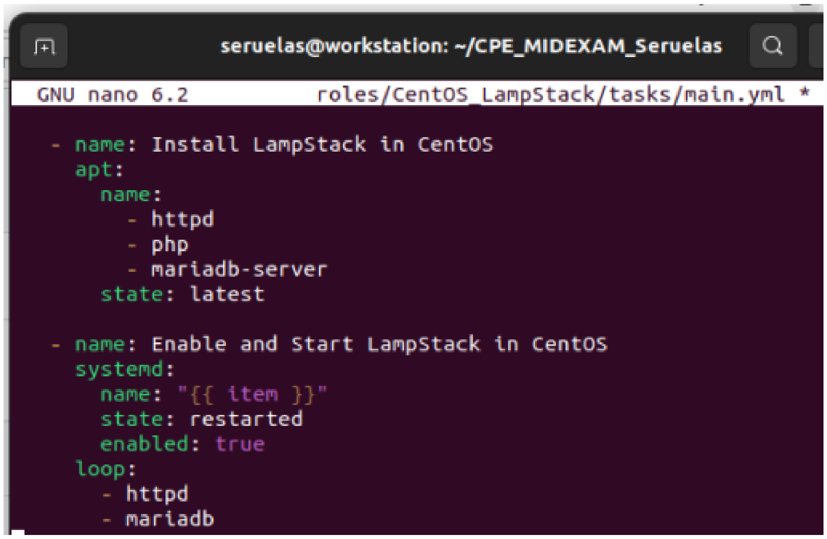
2. Configure the main.yml of the **Ubuntu_LampStack** role so that it will install the LampStack and enable each service after its installation.



```
GNU nano 6.2              roles/Ubuntu_LampStack/tasks/main.yml

 - name: Install LampStack in Ubuntu
   apt:
     name:
       - apache2
       - libapache2-mod-php
       - mariadb-server
     state: latest

 - name: Enable and Start LampStack in Ubuntu
   systemd:
     name: "{{ item }}"
     state: restarted
     enabled: true
   loop:
     - apache2
     - mariadb
```

Figure 2.2.1 - Configuration of the Ubuntu_LampStack role for installation of Lampstack in Ubuntu.

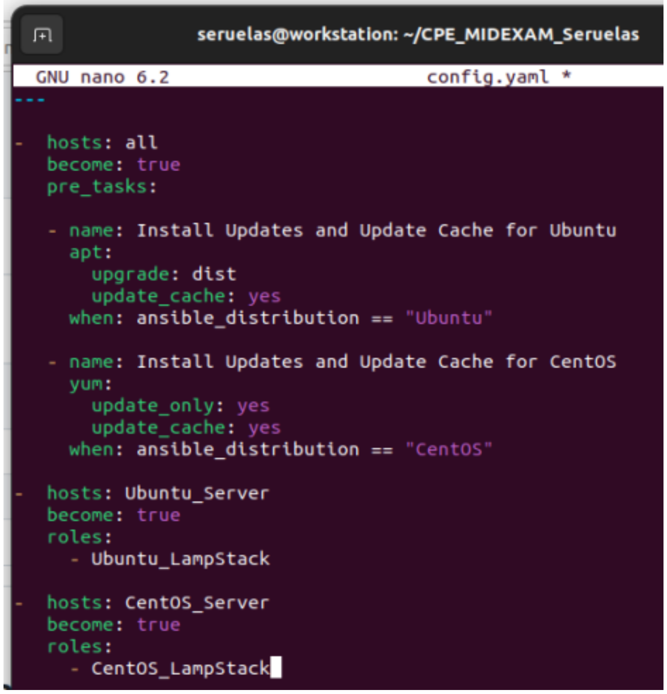
3. Configure the main.yml of the **CentOS_LampStack** role so that it will install the LampStack and enable each service after its installation.



```
seruelas@workstation: ~/CPE_MIDEXAM_Seruelas
GNU nano 6.2              roles/CentOS_LampStack/tasks/main.yml *

  - name: Install LampStack in CentOS
    apt:
      name:
        - httpd
        - php
        - mariadb-server
      state: latest

  - name: Enable and Start LampStack in CentOS
    systemd:
      name: "{{ item }}"
      state: restarted
      enabled: true
    loop:
      - httpd
      - mariadb
```

Figure 2.3.1 - Configuration of the main.yml of the **CentOS_LampStack** role to install LampStack in CentOS.

4. After the configuration of both LampStack roles, create the **config.yaml** playbook in the main directory that will be running all the roles for the installation of the LampStack.

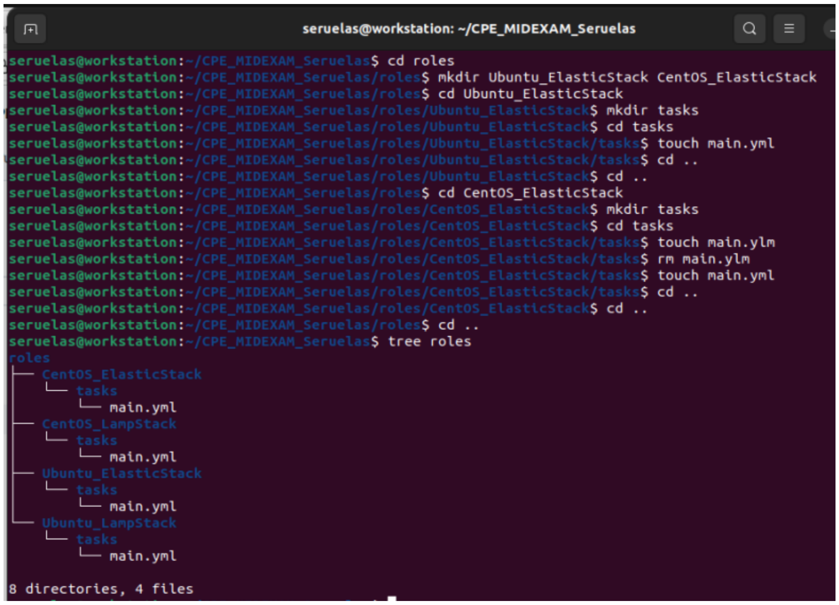


```
seruelas@workstation: ~/CPE_MIDEXAM_Seruelas
GNU nano 6.2                          config.yaml *
---

- hosts: all
  become: true
  pre_tasks:

  - name: Install Updates and Update Cache for Ubuntu
    apt:
      upgrade: dist
      update_cache: yes
    when: ansible_distribution == "Ubuntu"

  - name: Install Updates and Update Cache for CentOS
    yum:
      update_only: yes
      update_cache: yes
    when: ansible_distribution == "CentOS"

- hosts: Ubuntu_Server
  become: true
  roles:
    - Ubuntu_LampStack

- hosts: CentOS_Server
  become: true
  roles:
    - CentOS_LampStack
```

Figure 2.4.1 - Configuration of the **config.yaml** playbook.

Task 3: Installation of ElasticStack in Ubuntu and CentOS

For preparation, create the dedicated roles for both the Ubuntu and CentOS for the installation of the ElasticStack (ElasticSearch, Kibana, Logstash).

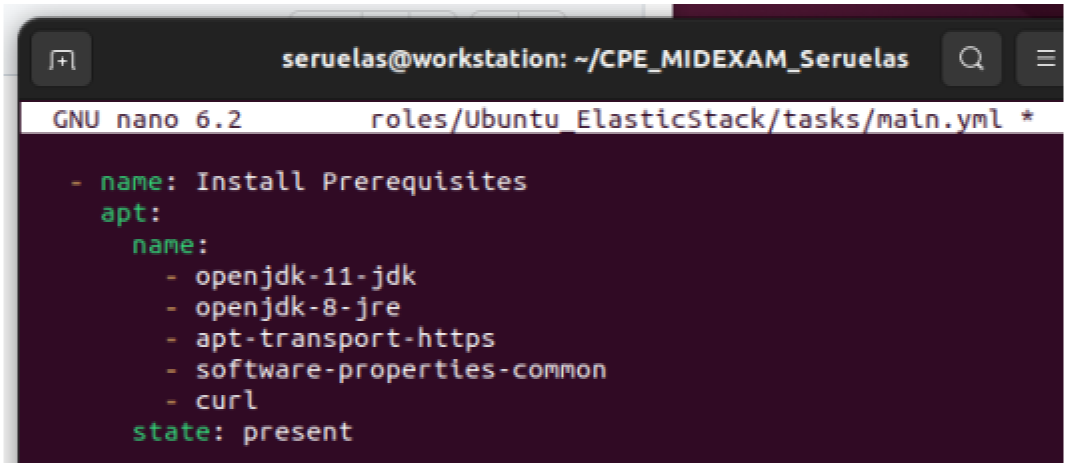


```
seruelas@workstation: ~/CPE_MIDEXAM_Seruelas
seruelas@workstation:~/CPE_MIDEXAM_Seruelas$ cd roles
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles$ mkdir Ubuntu_ElasticStack CentOS_ElasticStack
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles$ cd Ubuntu_ElasticStack
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles/Ubuntu_ElasticStack$ mkdir tasks
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles/Ubuntu_ElasticStack$ cd tasks
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles/Ubuntu_ElasticStack/tasks$ touch main.yml
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles/Ubuntu_ElasticStack/tasks$ cd ..
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles/Ubuntu_ElasticStack$ cd ..
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles$ cd CentOS_ElasticStack
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles/CentOS_ElasticStack$ mkdir tasks
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles/CentOS_ElasticStack$ cd tasks
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles/CentOS_ElasticStack/tasks$ touch main.ylm
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles/CentOS_ElasticStack/tasks$ rm main.ylm
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles/CentOS_ElasticStack/tasks$ touch main.yml
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles/CentOS_ElasticStack/tasks$ cd ..
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles/CentOS_ElasticStack$ cd ..
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles$ cd ..
seruelas@workstation:~/CPE_MIDEXAM_Seruelas$ tree roles
roles
├── CentOS_ElasticStack
│   └── tasks
│       └── main.yml
├── CentOS_LampStack
│   └── tasks
│       └── main.yml
├── Ubuntu_ElasticStack
│   └── tasks
│       └── main.yml
└── Ubuntu_LampStack
    └── tasks
        └── main.yml

8 directories, 4 files
```

Figure 3.0 - Creation of roles for installation of ElasticStack in Ubuntu and CentOS.

Task 3.1: Installation of Elastic Stack in Ubuntu

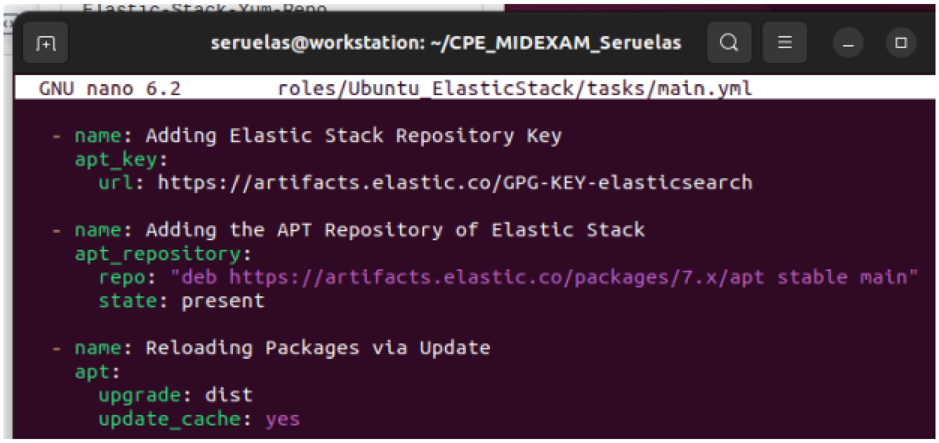
1. To install ElasticStack in Ubuntu, configure the main.yml of the **Ubuntu_ElasticStack** role first to install the prerequisites needed to make the ElasticStack services operable.



```
seruelas@workstation: ~/CPE_MIDEXAM_Seruelas
 GNU nano 6.2          roles/Ubuntu_ElasticStack/tasks/main.yml *

 - name: Install Prerequisites
   apt:
     name:
       - openjdk-11-jdk
       - openjdk-8-jre
       - apt-transport-https
       - software-properties-common
       - curl
     state: present
```

Figure 3.1.1.1 - Module that will install the prerequisites for ElasticStack.

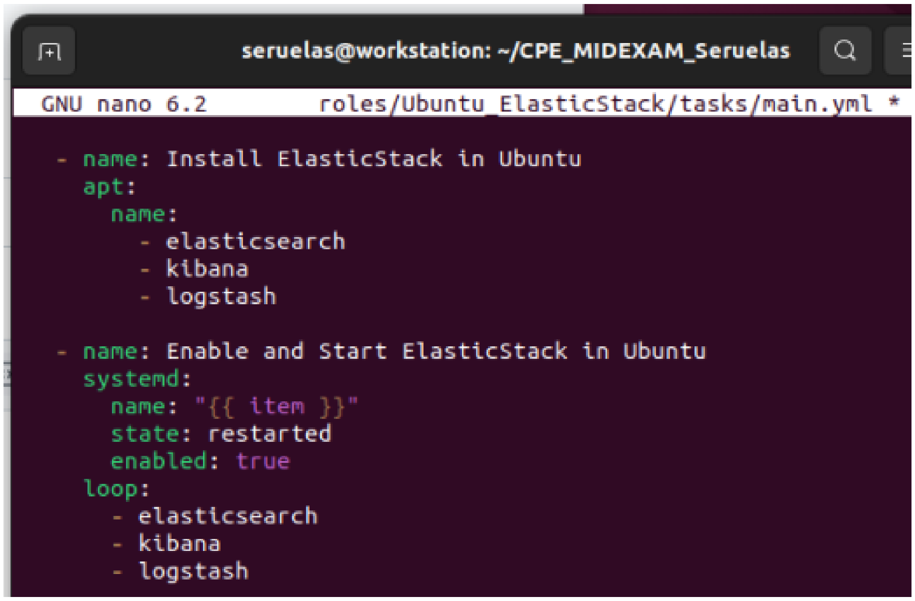
2. Create the module/s that will add the ElasticStack repository in Ubuntu.



```
  GNU nano 6.2          roles/Ubuntu_ElasticStack/tasks/main.yml

  - name: Adding Elastic Stack Repository Key
    apt_key:
      url: https://artifacts.elastic.co/GPG-KEY-elasticsearch

  - name: Adding the APT Repository of Elastic Stack
    apt_repository:
      repo: "deb https://artifacts.elastic.co/packages/7.x/apt stable main"
      state: present

  - name: Reloading Packages via Update
    apt:
      upgrade: dist
      update_cache: yes
```

Figure 3.1.2.1 - Module that will add the repository key, apt repository of ElasticStack and reload the packages in Ubuntu.

3. Create the module/s that will install and enable the ElasticStack in Ubuntu.

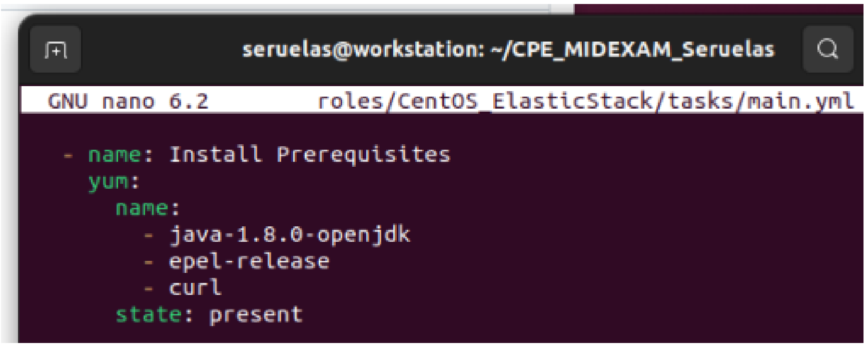


```
  GNU nano 6.2          roles/Ubuntu_ElasticStack/tasks/main.yml *

  - name: Install ElasticStack in Ubuntu
    apt:
      name:
        - elasticsearch
        - kibana
        - logstash

  - name: Enable and Start ElasticStack in Ubuntu
    systemd:
      name: "{{ item }}"
      state: restarted
      enabled: true
    loop:
      - elasticsearch
      - kibana
      - logstash
```

Figure 3.1.5.1 - Module/s that will install and enable ElasticStack in Ubuntu.

Task 3.2: Installation of ElasticStack in CentOS

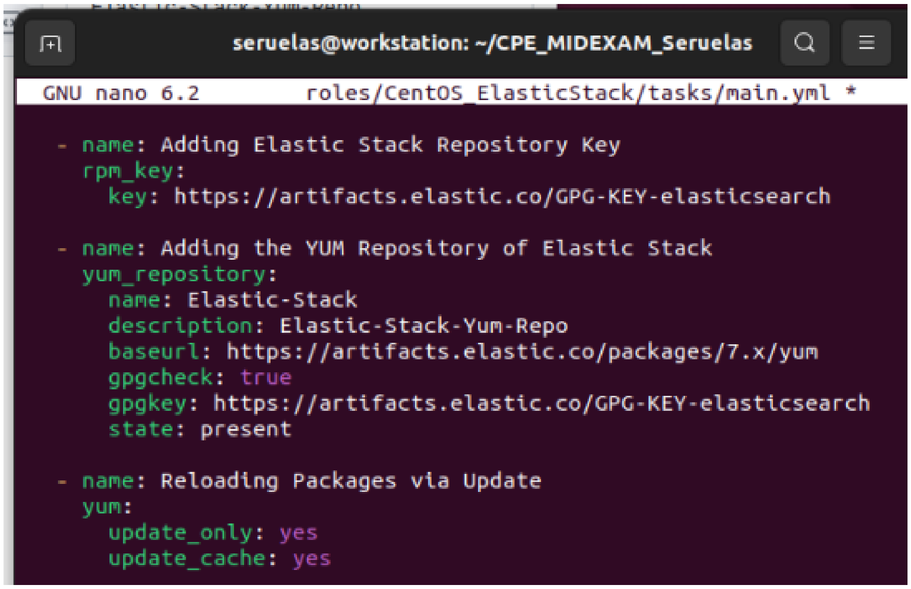
1. To install ElasticStack in CentOS, configure the main.yml of the **Ubuntu_ElasticStack** role first to install the prerequisites needed to make the ElasticStack services operable.



```
  GNU nano 6.2          roles/CentOS_ElasticStack/tasks/main.yml

  - name: Install Prerequisites
    yum:
      name:
        - java-1.8.0-openjdk
        - epel-release
        - curl
      state: present
```

Figure 3.2.1.1 - Module that installs the prerequisites for ElasticStack.

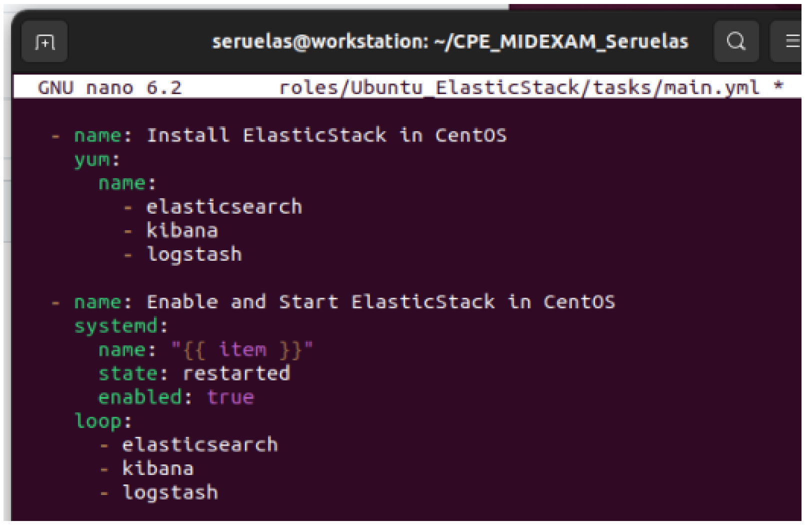
2. Create the module/s that will add the ElasticStack repository to CentOS.



```
  GNU nano 6.2          roles/CentOS_ElasticStack/tasks/main.yml *

  - name: Adding Elastic Stack Repository Key
    rpm_key:
      key: https://artifacts.elastic.co/GPG-KEY-elasticsearch

  - name: Adding the YUM Repository of Elastic Stack
    yum_repository:
      name: Elastic-Stack
      description: Elastic-Stack-Yum-Repo
      baseurl: https://artifacts.elastic.co/packages/7.x/yum
      gpgcheck: true
      gpgkey: https://artifacts.elastic.co/GPG-KEY-elasticsearch
      state: present

  - name: Reloading Packages via Update
    yum:
      update_only: yes
      update_cache: yes
```

Figure 3.2.2.1 - Module/s that adds the key, adds the yum repository and reloads the packages of ElasticStack in CentOS.

3. Create the module/s that will install and enable ElasticStack in CentOS.

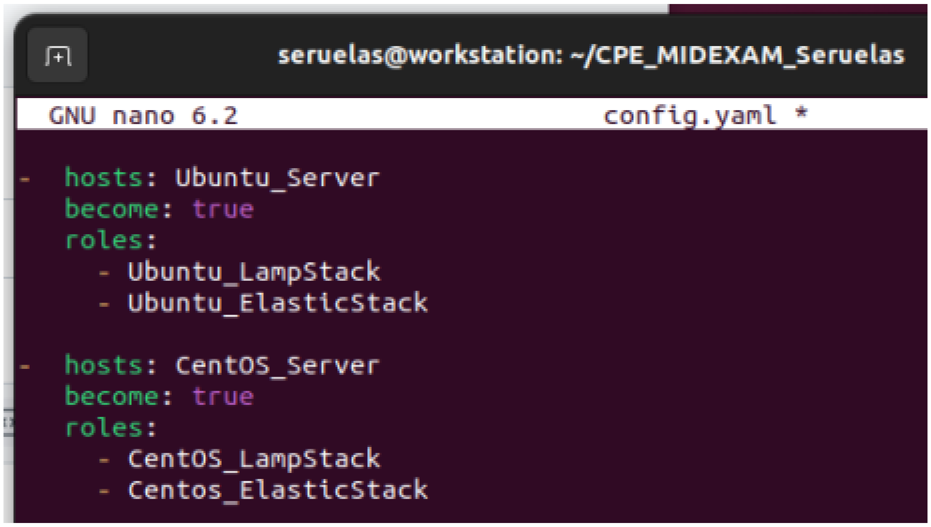


seruelas@workstation: ~/CPE_MIDEXAM_Seruelas
GNU nano 6.2
roles/Ubuntu_ElasticStack/tasks/main.yml *

```
- name: Install ElasticStack in CentOS
  yum:
    name:
      - elasticsearch
      - kibana
      - logstash

- name: Enable and Start ElasticStack in CentOS
  systemd:
    name: "{{ item }}"
    state: restarted
    enabled: true
  loop:
    - elasticsearch
    - kibana
    - logstash
```

Figure 3.2.3.1 - Module/s that installs and enables ElasticStack in CentOS.

4. Modify the **config.yaml** in the main directory of the repository that will execute the two roles.

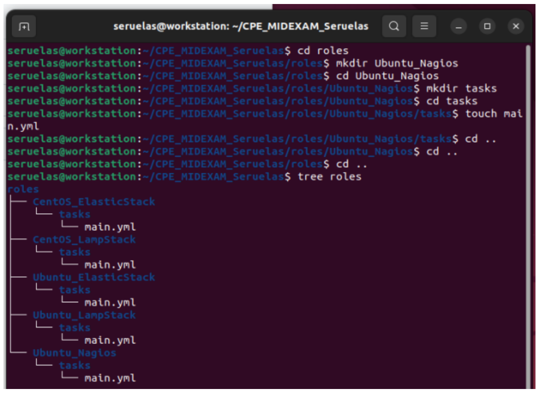


```
- hosts: Ubuntu_Server
  become: true
  roles:
    - Ubuntu_LampStack
    - Ubuntu_ElasticStack

- hosts: CentOS_Server
  become: true
  roles:
    - CentOS_LampStack
    - Centos_ElasticStack
```

Figure 3.2.4.1 - Modified config.yaml that executes the two roles that will install ElasticStack.

Task 4: Installation of Nagios in one host (Chosen host: Ubuntu)

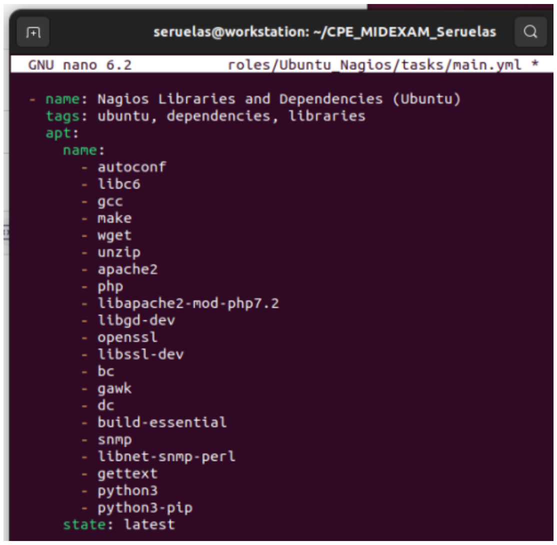
1. Create a dedicated role for the installation of Nagios in Ubuntu.



```
seruelas@workstation:~/CPE_MIDEXAM_Seruelas$ cd roles
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles$ mkdir Ubuntu_Nagios
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles$ cd Ubuntu_Nagios
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles/Ubuntu_Nagios$ mkdir tasks
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles/Ubuntu_Nagios$ cd tasks
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles/Ubuntu_Nagios/tasks$ touch mai
n.yml
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles/Ubuntu_Nagios/tasks$ cd ..
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles/Ubuntu_Nagios$ cd ..
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles$ cd ..
seruelas@workstation:~/CPE_MIDEXAM_Seruelas$ tree roles
roles
├── CentOS_ElasticStack
│   └── tasks
│       └── main.yml
├── CentOS_LampStack
│   └── tasks
│       └── main.yml
├── Ubuntu_ElasticStack
│   └── tasks
│       └── main.yml
├── Ubuntu_LampStack
│   └── tasks
│       └── main.yml
└── Ubuntu_Nagios
    └── tasks
        └── main.yml
```

Figure 4.1.1 - Creation of the **Ubuntu_Nagios** role.

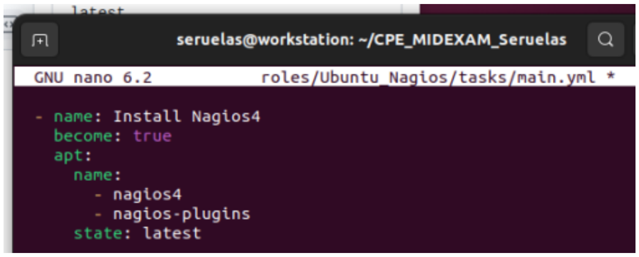
2. Create a module that will install all the pre-requisites for Nagios.



```
  GNU nano 6.2            roles/Ubuntu_Nagios/tasks/main.yml *

 - name: Nagios Libraries and Dependencies (Ubuntu)
   tags: ubuntu, dependencies, libraries
   apt:
     name:
       - autoconf
       - libc6
       - gcc
       - make
       - wget
       - unzip
       - apache2
       - php
       - libapache2-mod-php7.2
       - libgd-dev
       - openssl
       - libssl-dev
       - bc
       - gawk
       - dc
       - build-essential
       - snmp
       - libnet-snmp-perl
       - gettext
       - python3
       - python3-pip
     state: latest
```

Figure 4.2.1 - Module that installs all the prerequisites for Nagios.

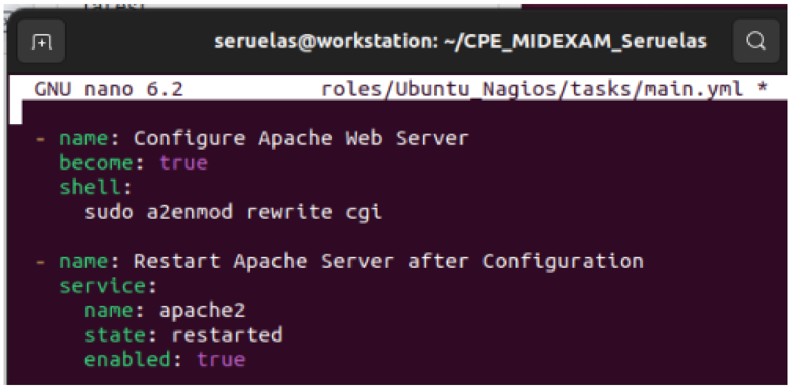
3. Create a module that installs Nagios.



```
  GNU nano 6.2            roles/Ubuntu_Nagios/tasks/main.yml *

 - name: Install Nagios4
   become: true
   apt:
     name:
       - nagios4
       - nagios-plugins
     state: latest
```

Figure 4.3.1 - Module that installs Nagios.

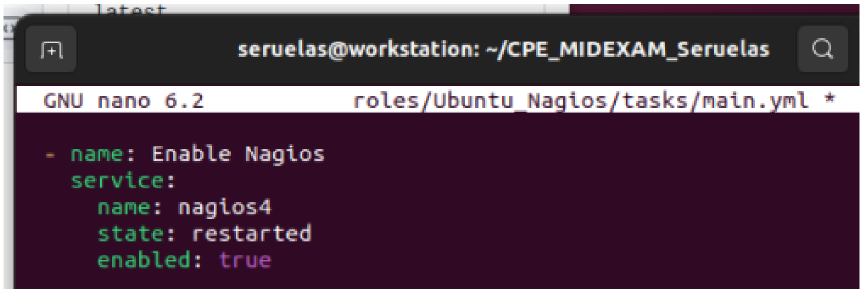
4. Create the module/s that modifies the Apache Web Server to allow operation of Nagios and restarts the apache web server service.



```
GNU nano 6.2              roles/Ubuntu_Nagios/tasks/main.yml *

- name: Configure Apache Web Server
  become: true
  shell:
    sudo a2enmod rewrite cgi

- name: Restart Apache Server after Configuration
  service:
    name: apache2
    state: restarted
    enabled: true
```

Figure 4.4.1 - Module/s that configures the Apache Web Server and restarts the service after configuration.

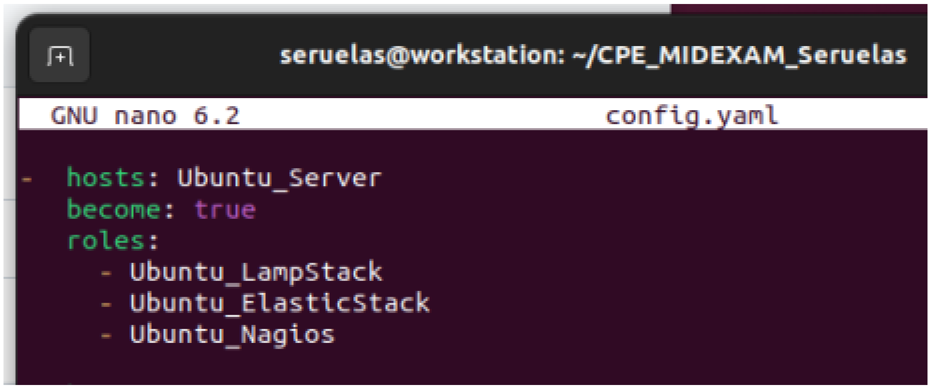
5. Create a module that will enable and start Nagios.



```
GNU nano 6.2              roles/Ubuntu_Nagios/tasks/main.yml *

- name: Enable Nagios
  service:
    name: nagios4
    state: restarted
    enabled: true
```

Figure 4.5.1 - Module that enables and starts the Nagios service.

6. Modify the **config.yaml** in the main directory of the repository that will execute the nagios role.

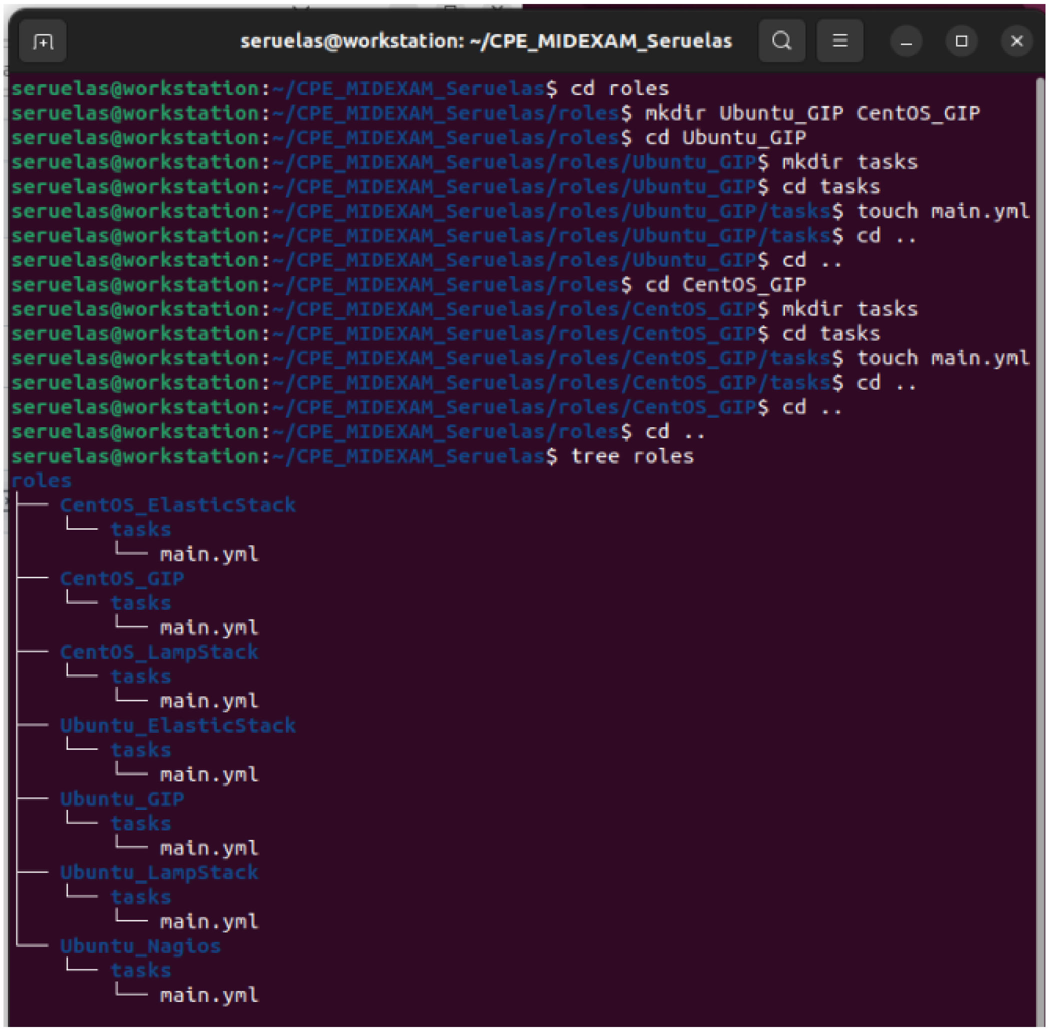


```
GNU nano 6.2                               config.yaml

-  hosts: Ubuntu_Server
   become: true
   roles:
     - Ubuntu_LampStack
     - Ubuntu_ElasticStack
     - Ubuntu_Nagios
```

Figure 4.6.1 - Configuration of the main.yml to execute the **Ubuntu_Nagios** role by the **Ubuntu_Server** host.

Task 5: Installation of Grafana, InfluxDB, and Prometheus in Ubuntu and CentOS.

Create the dedicated roles that will install Grafana, InfluxDB, and Prometheus in each server.

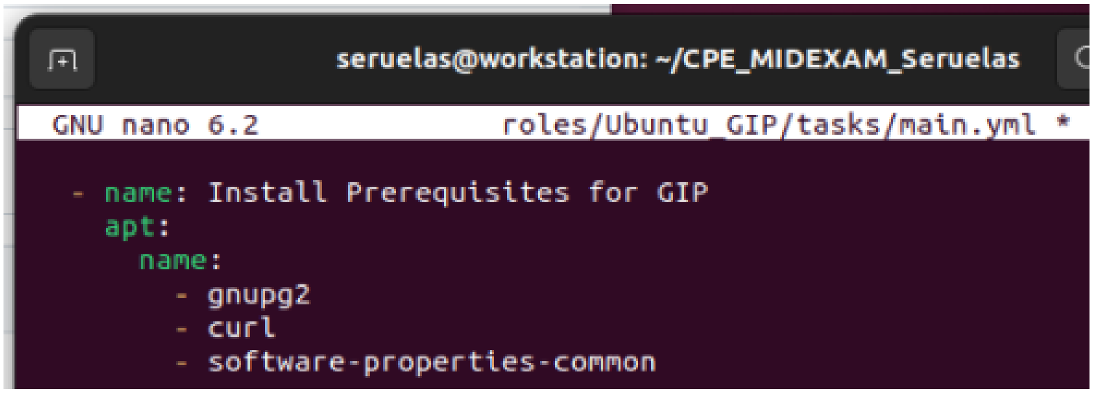


```
seruelas@workstation:~/CPE_MIDEXAM_Seruelas$ cd roles
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles$ mkdir Ubuntu_GIP CentOS_GIP
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles$ cd Ubuntu_GIP
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles/Ubuntu_GIP$ mkdir tasks
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles/Ubuntu_GIP$ cd tasks
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles/Ubuntu_GIP/tasks$ touch main.yml
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles/Ubuntu_GIP/tasks$ cd ..
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles/Ubuntu_GIP$ cd ..
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles$ cd CentOS_GIP
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles/CentOS_GIP$ mkdir tasks
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles/CentOS_GIP$ cd tasks
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles/CentOS_GIP/tasks$ touch main.yml
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles/CentOS_GIP/tasks$ cd ..
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles/CentOS_GIP$ cd ..
seruelas@workstation:~/CPE_MIDEXAM_Seruelas/roles$ cd ..
seruelas@workstation:~/CPE_MIDEXAM_Seruelas$ tree roles
roles
├── CentOS_ElasticStack
│   └── tasks
│       └── main.yml
├── CentOS_GIP
│   └── tasks
│       └── main.yml
├── CentOS_LampStack
│   └── tasks
│       └── main.yml
├── Ubuntu_ElasticStack
│   └── tasks
│       └── main.yml
├── Ubuntu_GIP
│   └── tasks
│       └── main.yml
├── Ubuntu_LampStack
│   └── tasks
│       └── main.yml
└── Ubuntu_Nagios
    └── tasks
        └── main.yml
```

Figure 5.0 - Creation of the **GIP** (Grafana, InfluxDB, Prometheus) roles.

Task 5.1: Installation of Grafana, InfluxDB, and Prometheus in Ubuntu

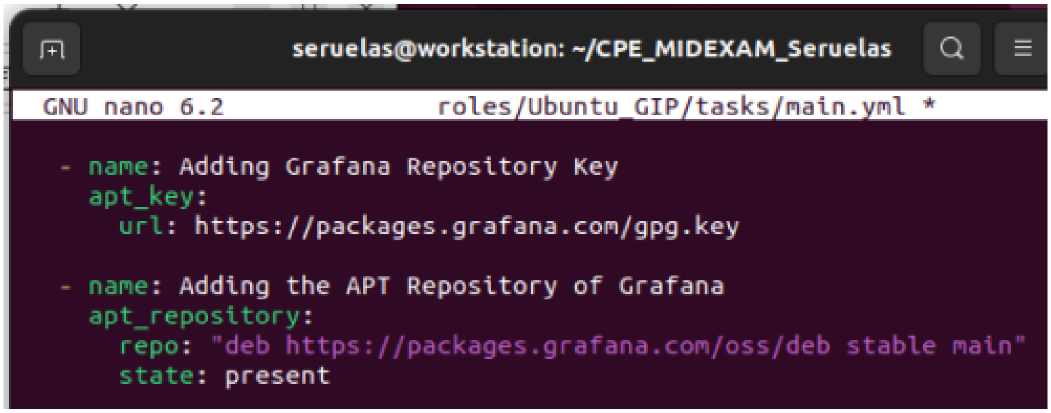
1. Create a module that will install all prerequisites of GIP.



```
GNU nano 6.2                    roles/Ubuntu_GIP/tasks/main.yml *

 - name: Install Prerequisites for GIP
   apt:
     name:
       - gnupg2
       - curl
       - software-properties-common
```

Figure 5.1.1.1 - Module that installs prerequisites for GIP.

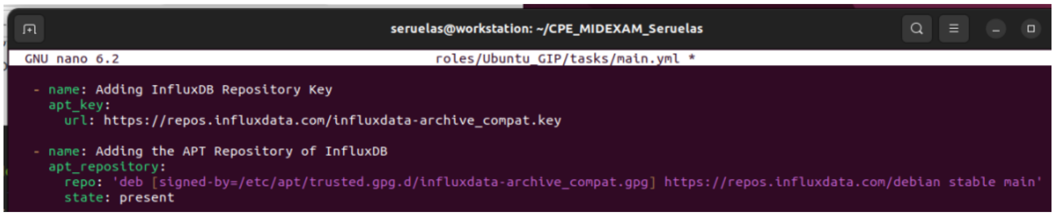
2. Create a module that adds the GPG key and apt repository of Grafana.



```
GNU nano 6.2                  roles/Ubuntu_GIP/tasks/main.yml *

 - name: Adding Grafana Repository Key
   apt_key:
     url: https://packages.grafana.com/gpg.key

 - name: Adding the APT Repository of Grafana
   apt_repository:
     repo: "deb https://packages.grafana.com/oss/deb stable main"
     state: present
```

Figure 5.1.2.1 - Module that adds the apt repository of Grafana.

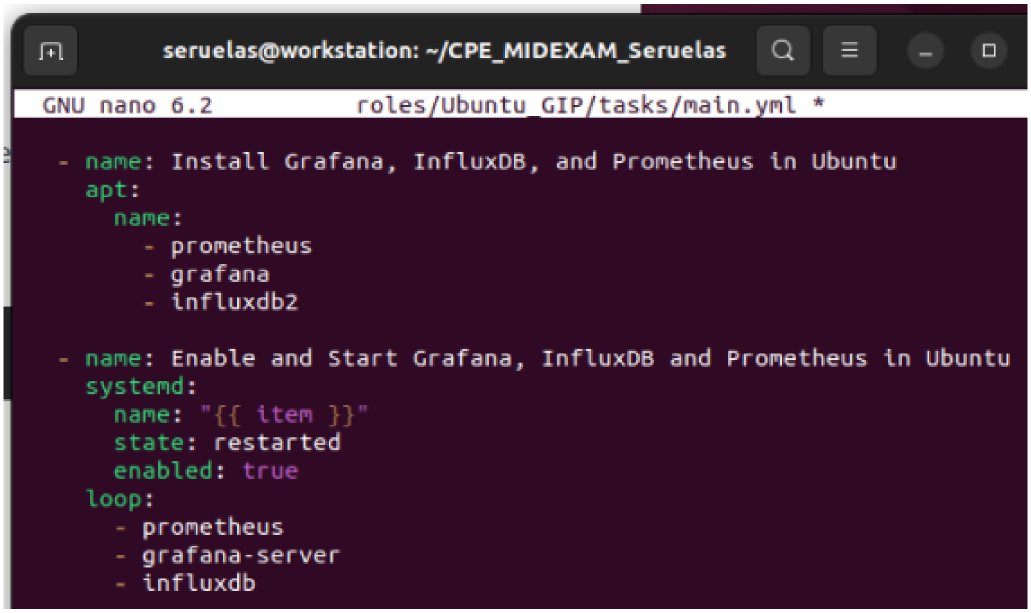
3. Create a module that adds the GPG key and apt repository of InfluxDB.



```
GNU nano 6.2                              roles/Ubuntu_GIP/tasks/main.yml *

 - name: Adding InfluxDB Repository Key
   apt_key:
     url: https://repos.influxdata.com/influxdata-archive_compat.key

 - name: Adding the APT Repository of InfluxDB
   apt_repository:
     repo: 'deb [signed-by=/etc/apt/trusted.gpg.d/influxdata-archive_compat.gpg] https://repos.influxdata.com/debian stable main'
     state: present
```

Figure 5.1.3.1 - Module that adds the apt repository of InfluxDB.

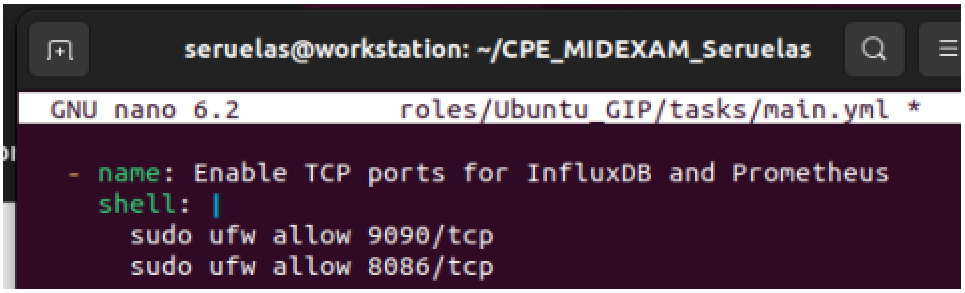
4. Create a module that installs and enables Grafana, InfluxDB, and Prometheus in Ubuntu.



```
 GNU nano 6.2             roles/Ubuntu_GIP/tasks/main.yml *

  - name: Install Grafana, InfluxDB, and Prometheus in Ubuntu
    apt:
      name:
        - prometheus
        - grafana
        - influxdb2

  - name: Enable and Start Grafana, InfluxDB and Prometheus in Ubuntu
    systemd:
      name: "{{ item }}"
      state: restarted
      enabled: true
    loop:
      - prometheus
      - grafana-server
      - influxdb
```

Figure 5.1.4.1 - Module that installs and enables GIP in Ubuntu.

5. Create a module that enables the ports for InfluxDB and Prometheus.

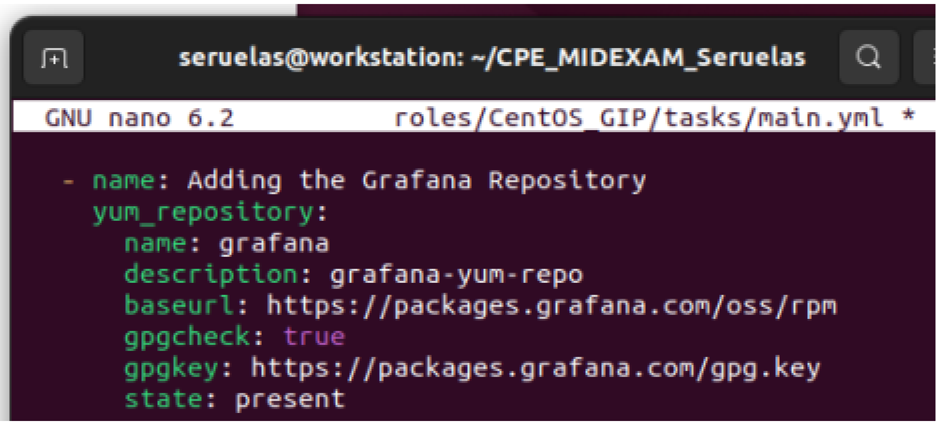


```
 GNU nano 6.2            roles/Ubuntu_GIP/tasks/main.yml *

  - name: Enable TCP ports for InfluxDB and Prometheus
    shell: |
      sudo ufw allow 9090/tcp
      sudo ufw allow 8086/tcp
```

Figure 5.1.6.1 - Module that enables specific TCP ports for InfluxDB and Prometheus.

Task 5.2: Installation of Grafana, InfluxDB, and Prometheus in CentOS

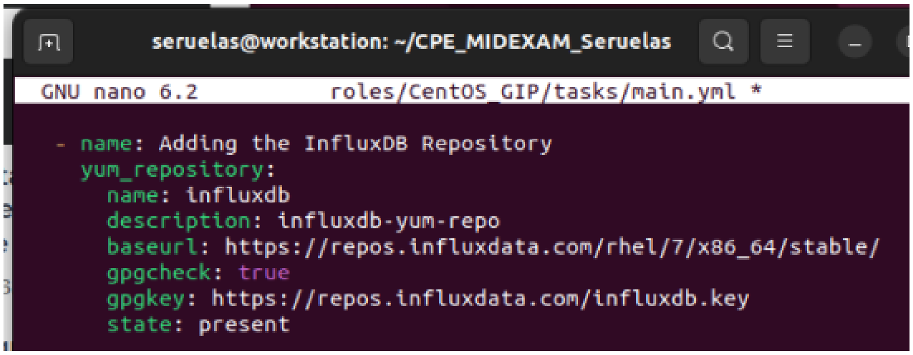
1. Create a module that adds the Grafana repository.



```
GNU nano 6.2             roles/CentOS_GIP/tasks/main.yml *

  - name: Adding the Grafana Repository
    yum_repository:
      name: grafana
      description: grafana-yum-repo
      baseurl: https://packages.grafana.com/oss/rpm
      gpgcheck: true
      gpgkey: https://packages.grafana.com/gpg.key
      state: present
```

Figure 5.2.1.1 - Module that adds the Grafana Repository.

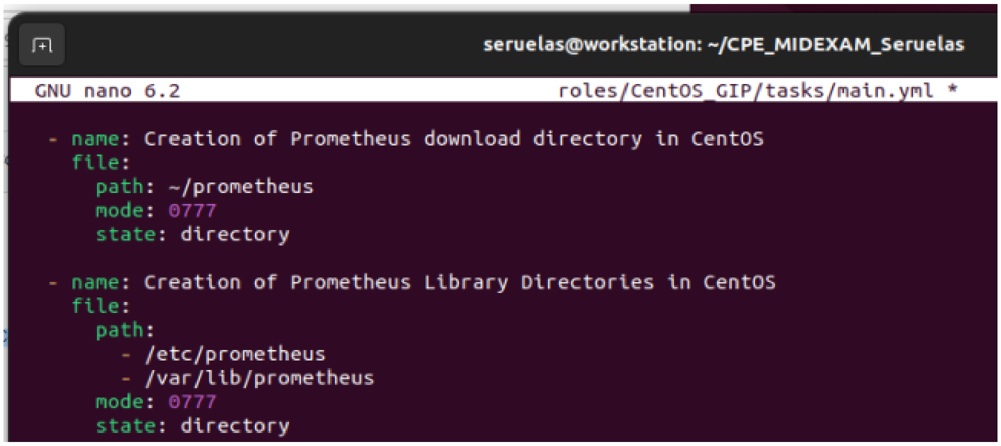
2. Create a module that adds the InfluxDB repository.



```
GNU nano 6.2             roles/CentOS_GIP/tasks/main.yml *

  - name: Adding the InfluxDB Repository
    yum_repository:
      name: influxdb
      description: influxdb-yum-repo
      baseurl: https://repos.influxdata.com/rhel/7/x86_64/stable/
      gpgcheck: true
      gpgkey: https://repos.influxdata.com/influxdb.key
      state: present
```

Figure 5.2.2.1 - Module that adds the InfluxDB repository.

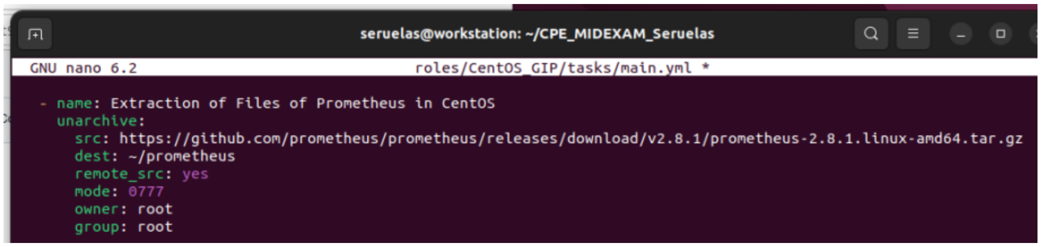
3. Create a module that creates the directories for Prometheus.



```
GNU nano 6.2                         roles/CentOS_GIP/tasks/main.yml *

  - name: Creation of Prometheus download directory in CentOS
    file:
      path: ~/prometheus
      mode: 0777
      state: directory

  - name: Creation of Prometheus Library Directories in CentOS
    file:
      path:
        - /etc/prometheus
        - /var/lib/prometheus
      mode: 0777
      state: directory
```

Figure 5.2.3.1 - Module that creates directories for Prometheus.

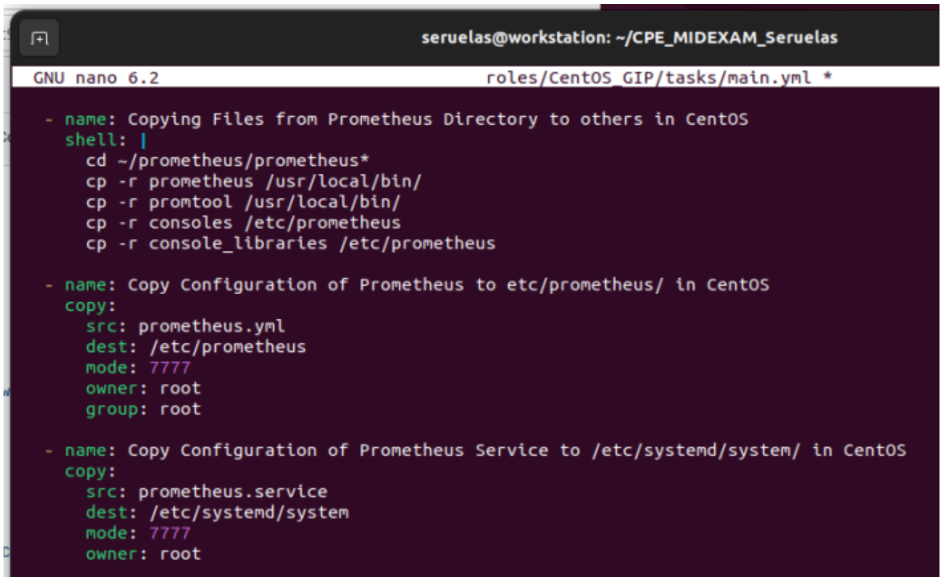
4. Create a module that downloads and extracts Prometheus.


seruelas@workstation: ~/CPE_MIDEXAM_Seruelas
GNU nano 6.2
roles/CentOS_GIP/tasks/main.yml *

```
- name: Extraction of Files of Prometheus in CentOS
  unarchive:
    src: https://github.com/prometheus/prometheus/releases/download/v2.8.1/prometheus-2.8.1.linux-amd64.tar.gz
    dest: ~/prometheus
    remote_src: yes
    mode: 0777
    owner: root
    group: root
```

Figure 5.2.4.1 - Module that downloads and extracts Prometheus.

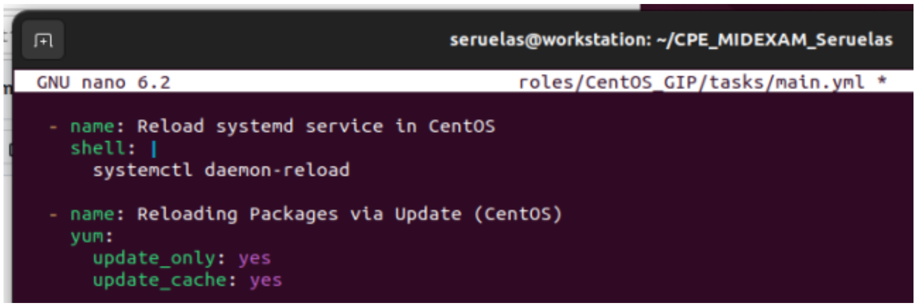
5. Create a module that copies the necessary files of Prometheus to the respective directories.


seruelas@workstation: ~/CPE_MIDEXAM_Seruelas
GNU nano 6.2
roles/CentOS_GIP/tasks/main.yml *

```
- name: Copying Files from Prometheus Directory to others in CentOS
  shell: |
    cd ~/prometheus/prometheus*
    cp -r prometheus /usr/local/bin/
    cp -r promtool /usr/local/bin/
    cp -r consoles /etc/prometheus
    cp -r console_libraries /etc/prometheus

- name: Copy Configuration of Prometheus to etc/prometheus/ in CentOS
  copy:
    src: prometheus.yml
    dest: /etc/prometheus
    mode: 7777
    owner: root
    group: root

- name: Copy Configuration of Prometheus Service to /etc/systemd/system/ in CentOS
  copy:
    src: prometheus.service
    dest: /etc/systemd/system
    mode: 7777
    owner: root
```

Figure 5.2.5.1 - Module that copies the files of Prometheus to respective directories.

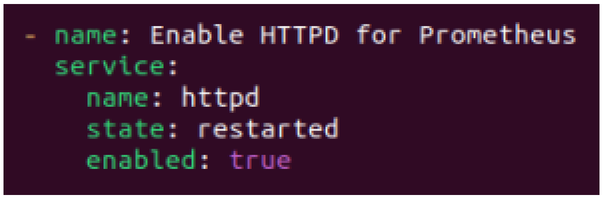
6. Create the module/s that will restart the systemd-daemon service and update the packages of CentOS.


seruelas@workstation: ~/CPE_MIDEXAM_Seruelas
GNU nano 6.2
roles/CentOS_GIP/tasks/main.yml *

```
- name: Reload systemd service in CentOS
  shell: |
    systemctl daemon-reload

- name: Reloading Packages via Update (CentOS)
  yum:
    update_only: yes
    update_cache: yes
```

Figure 5.2.6.1 - Modules that restarts the services and updates the packages.

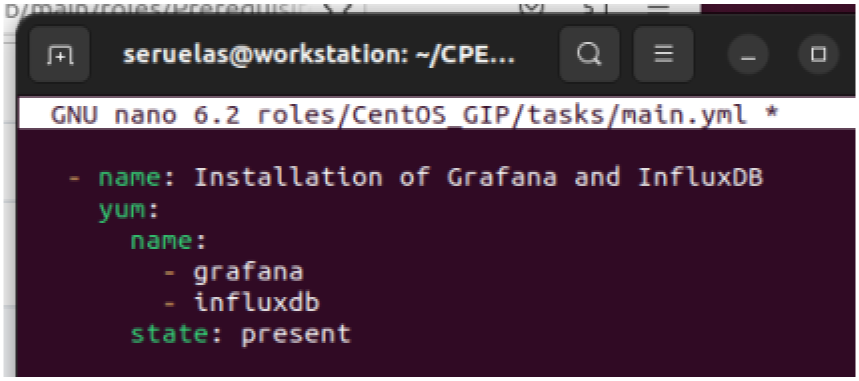
7. Create a module that enabled HTTPD for Prometheus.



```
- name: Enable HTTPD for Prometheus
  service:
    name: httpd
    state: restarted
    enabled: true
```

Figure 5.2.7.1 - Module that enables HTTP for operations of Prometheus.

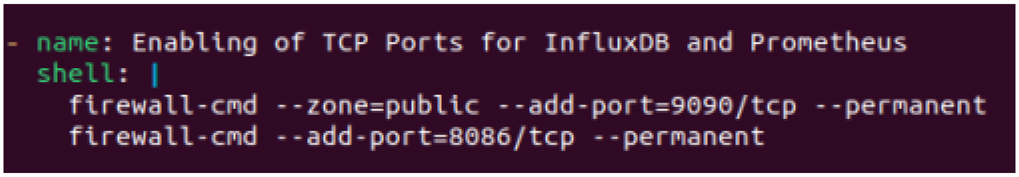
8. Create a module that installs Grafana and InfluxDB in CentOS.



```
 GNU nano 6.2 roles/CentOS_GIP/tasks/main.yml *

  - name: Installation of Grafana and InfluxDB
    yum:
      name:
        - grafana
        - influxdb
      state: present
```

Figure 5.2.8.1 - Module that installs Grafana and InfluxDB in CentOS.

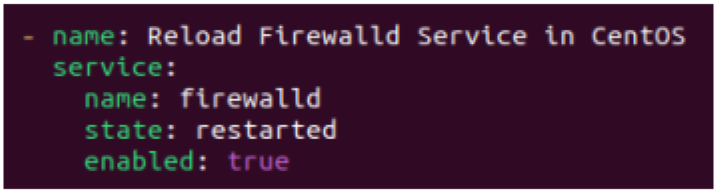
9. Create a module that enables the proper TCP ports for InfluxDB and Prometheus.



```
- name: Enabling of TCP Ports for InfluxDB and Prometheus
  shell: |
    firewall-cmd --zone=public --add-port=9090/tcp --permanent
    firewall-cmd --add-port=8086/tcp --permanent
```

Figure 5.2.9.1 - Module that enables the proper TCP ports for InfluxDB and Prometheus.

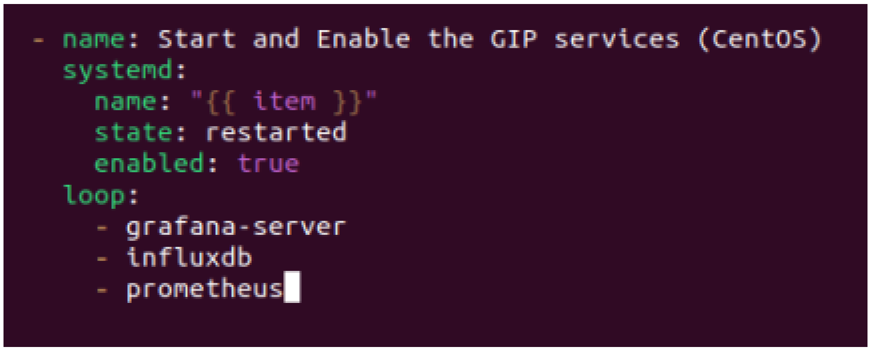
10. Create a module that reloads the firewall services.



```
- name: Reload Firewalld Service in CentOS
  service:
    name: firewalld
    state: restarted
    enabled: true
```

Figure 5.2.10.1 - Module that reloads the firewall services.

11. Create a module that enables Grafana, InfluxDB and Prometheus in CentOS.



```
- name: Start and Enable the GIP services (CentOS)
  systemd:
    name: "{{ item }}"
    state: restarted
    enabled: true
  loop:
    - grafana-server
    - influxdb
    - prometheus
```

Figure 5.2.11.1 - Module that enables and starts Grafana, InfluxDB, and Prometheus in CentOS.

GitHub link:

https://github.com/TuRonnDraco/CPE_MIDEXAM_Seruelas

Conclusions: (link your conclusion from the objective)

In this Midterm Examination, we have utilized roles in specifying our commands for each hosts and that we are able to utilize roles to organize and to execute specific only commands on each groups or specified roles. In this midterm examination, we are able to utilize what we have learned in our Midterm Period and we are able to install Elastic Stack, Lamp Stack, Grafana, InfluxDB, and Prometheus successfully in our nodes.