Name: Keneth Campo	**Date Performed: November 8, 2024**
Course/Section: CPE 212- CPE31S21	**Date Submitted: November 8, 2024**
Instructor: Engr. RObin Valenzuela	**Semester and SY: 2024-2025**

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

1. Objectives

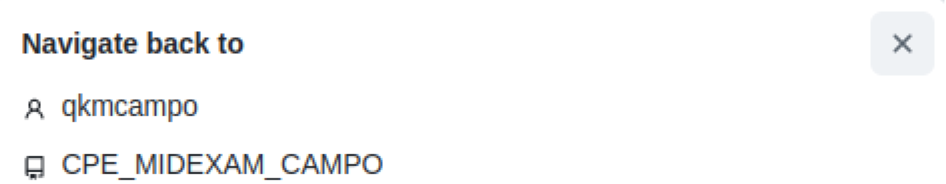
Create and design a workflow that installs, configure and manage enterprise availability, performance and log monitoring tools using Ansible as an Infrastructure as Code (IaC) tool.

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 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. **5.** Finally, paste also the link of your GitHub repository in the documentation.

3. Output (screenshots and explanations)

1.



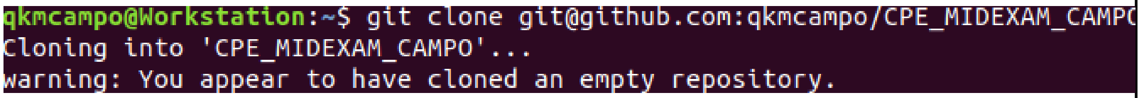
Navigate back to

qkmcampo

CPE_MIDEXAM_CAMPO

I create a repository inside the GitHub

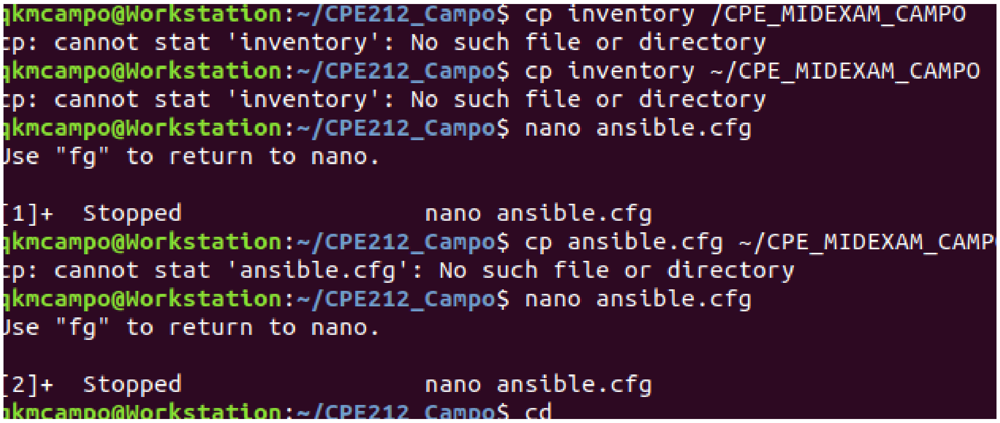
2.



```
qkmcampo@Workstation:~$ git clone git@github.com:qkmcampo/CPE_MIDEXAM_CAMPO
Cloning into 'CPE_MIDEXAM_CAMPO'...
warning: You appear to have cloned an empty repository.
```

I cloned the repository by creating first the ssh key and copy the ssh and run it to the terminal.

2.1.

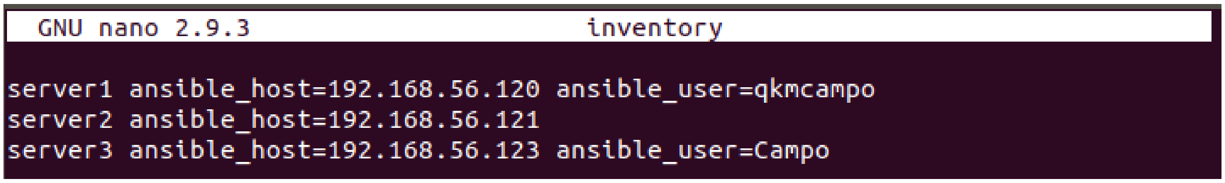


```
qkmcampo@Workstation:~/CPE212_Campo$ cp inventory /CPE_MIDEXAM_CAMPO
cp: cannot stat 'inventory': No such file or directory
qkmcampo@Workstation:~/CPE212_Campo$ cp inventory ~/CPE_MIDEXAM_CAMPO
cp: cannot stat 'inventory': No such file or directory
qkmcampo@Workstation:~/CPE212_Campo$ nano ansible.cfg
Use "fg" to return to nano.

[1]+  Stopped                 nano ansible.cfg
qkmcampo@Workstation:~/CPE212_Campo$ cp ansible.cfg ~/CPE_MIDEXAM_CAMP
cp: cannot stat 'ansible.cfg': No such file or directory
qkmcampo@Workstation:~/CPE212_Campo$ nano ansible.cfg
Use "fg" to return to nano.

[2]+  Stopped                 nano ansible.cfg
qkmcampo@Workstation:~/CPE212_Campo$ cd
```

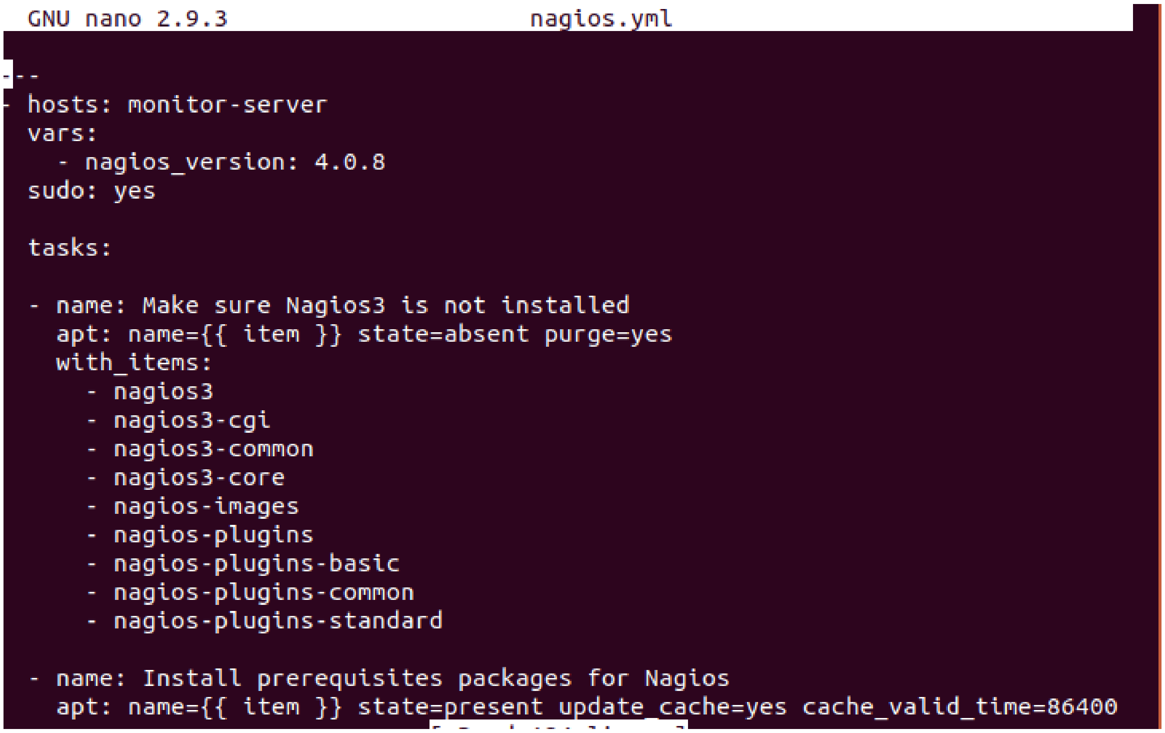
I created a Ansible playbook



```
  GNU nano 2.9.3                          inventory

server1 ansible_host=192.168.56.120 ansible_user=qkmcampo
server2 ansible_host=192.168.56.121
server3 ansible_host=192.168.56.123 ansible_user=Campo
```

2.2.



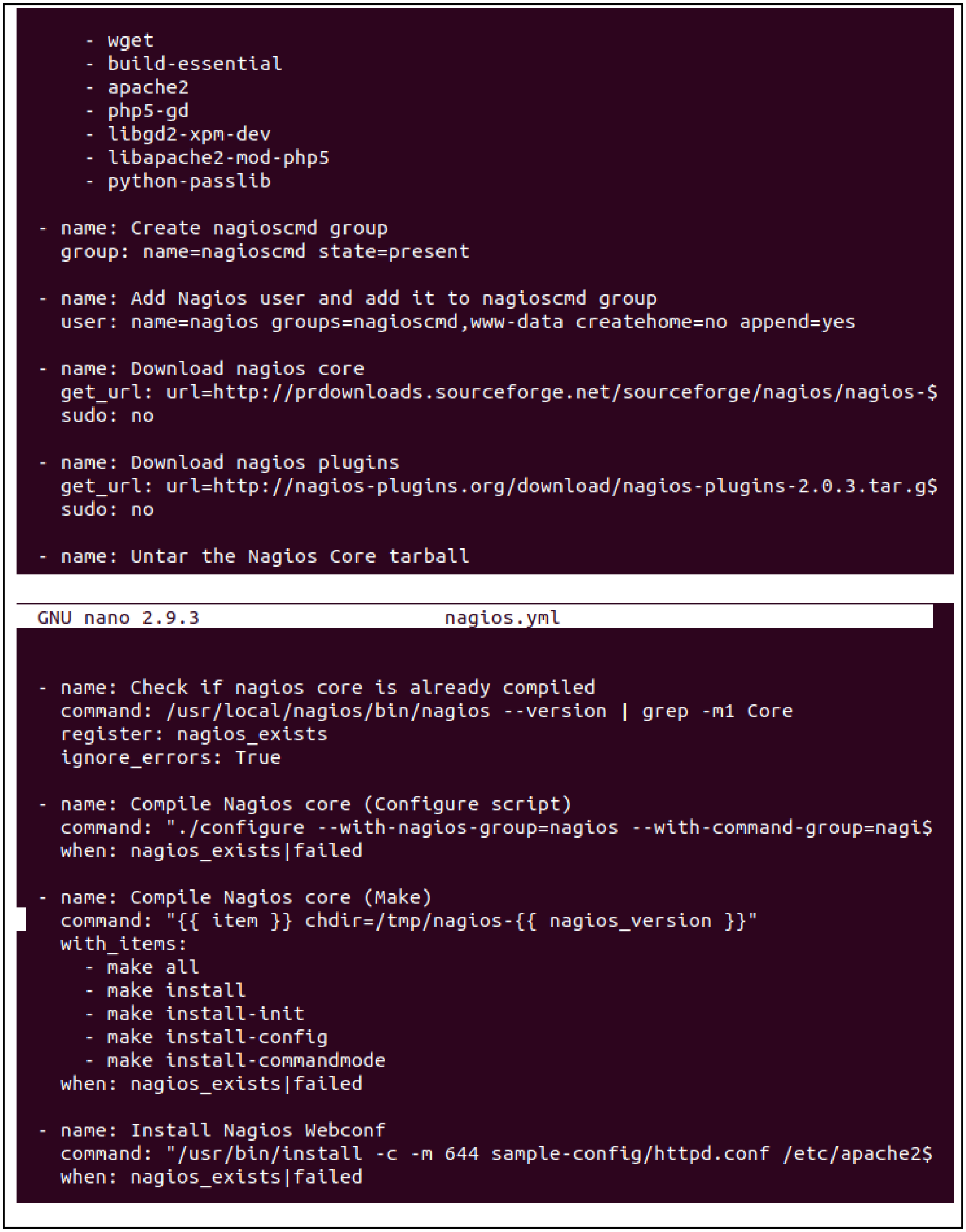
```
  GNU nano 2.9.3                          nagios.yml

---
- hosts: monitor-server
  vars:
    - nagios_version: 4.0.8
  sudo: yes

  tasks:

  - name: Make sure Nagios3 is not installed
    apt: name={{ item }} state=absent purge=yes
    with_items:
      - nagios3
      - nagios3-cgi
      - nagios3-common
      - nagios3-core
      - nagios-images
      - nagios-plugins
      - nagios-plugins-basic
      - nagios-plugins-common
      - nagios-plugins-standard

  - name: Install prerequisites packages for Nagios
    apt: name={{ item }} state=present update_cache=yes cache_valid_time=86400
```



```
    - wget
    - build-essential
    - apache2
    - php5-gd
    - libgd2-xpm-dev
    - libapache2-mod-php5
    - python-passlib

- name: Create nagioscmd group
  group: name=nagioscmd state=present

- name: Add Nagios user and add it to nagioscmd group
  user: name=nagios groups=nagioscmd,www-data createhome=no append=yes

- name: Download nagios core
  get_url: url=http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-$
  sudo: no

- name: Download nagios plugins
  get_url: url=http://nagios-plugins.org/download/nagios-plugins-2.0.3.tar.g$
  sudo: no

- name: Untar the Nagios Core tarball
```

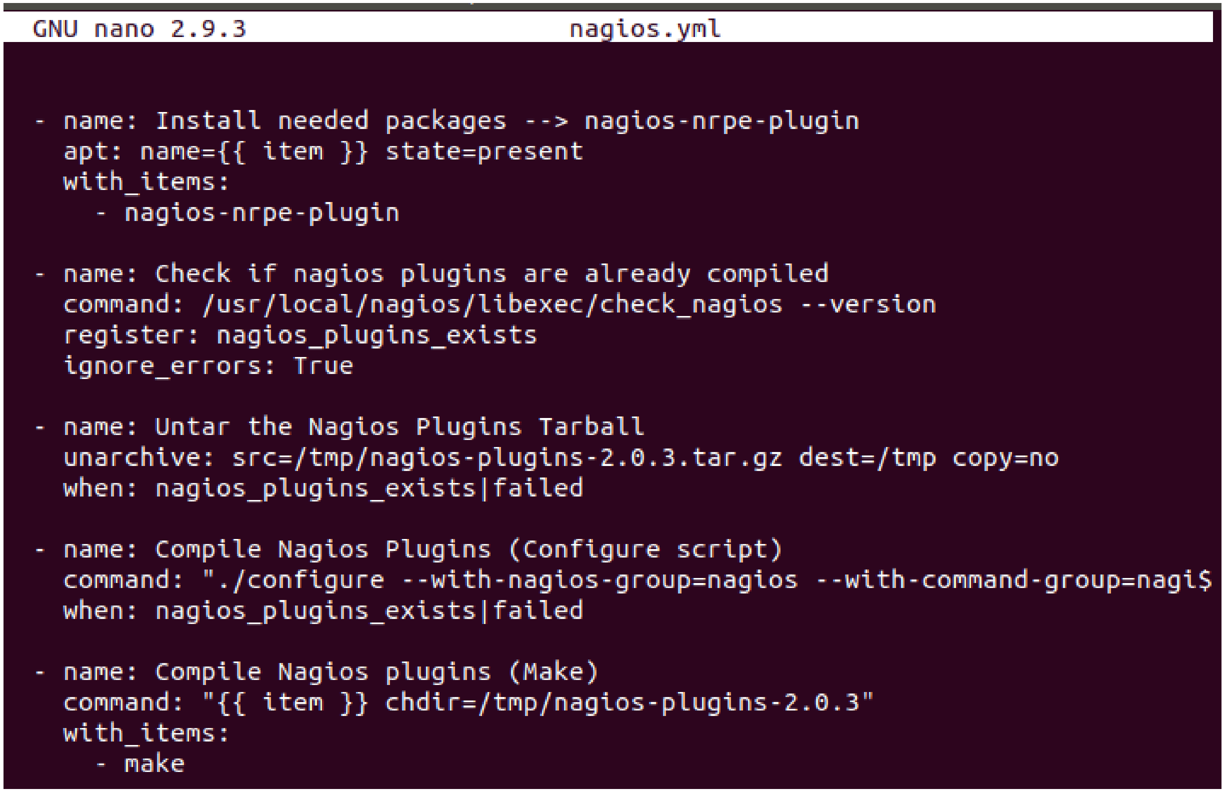
```
 GNU nano 2.9.3                    nagios.yml


- name: Check if nagios core is already compiled
  command: /usr/local/nagios/bin/nagios --version | grep -m1 Core
  register: nagios_exists
  ignore_errors: True

- name: Compile Nagios core (Configure script)
  command: "./configure --with-nagios-group=nagios --with-command-group=nagi$
  when: nagios_exists|failed

- name: Compile Nagios core (Make)
  command: "{{ item }} chdir=/tmp/nagios-{{ nagios_version }}"
  with_items:
    - make all
    - make install
    - make install-init
    - make install-config
    - make install-commandmode
  when: nagios_exists|failed

- name: Install Nagios Webconf
  command: "/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/apache2$
  when: nagios_exists|failed
```



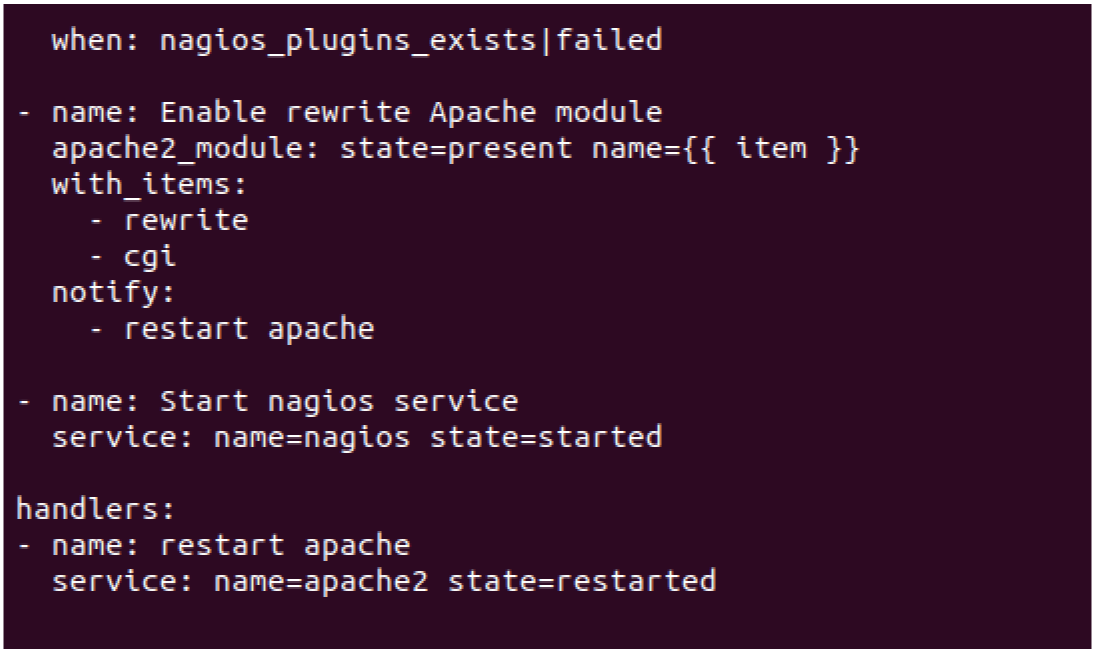
```
  - name: Install needed packages --> nagios-nrpe-plugin
    apt: name={{ item }} state=present
    with_items:
      - nagios-nrpe-plugin

  - name: Check if nagios plugins are already compiled
    command: /usr/local/nagios/libexec/check_nagios --version
    register: nagios_plugins_exists
    ignore_errors: True

  - name: Untar the Nagios Plugins Tarball
    unarchive: src=/tmp/nagios-plugins-2.0.3.tar.gz dest=/tmp copy=no
    when: nagios_plugins_exists|failed

  - name: Compile Nagios Plugins (Configure script)
    command: "./configure --with-nagios-group=nagios --with-command-group=nagi$
    when: nagios_plugins_exists|failed

  - name: Compile Nagios plugins (Make)
    command: "{{ item }} chdir=/tmp/nagios-plugins-2.0.3"
    with_items:
      - make
```



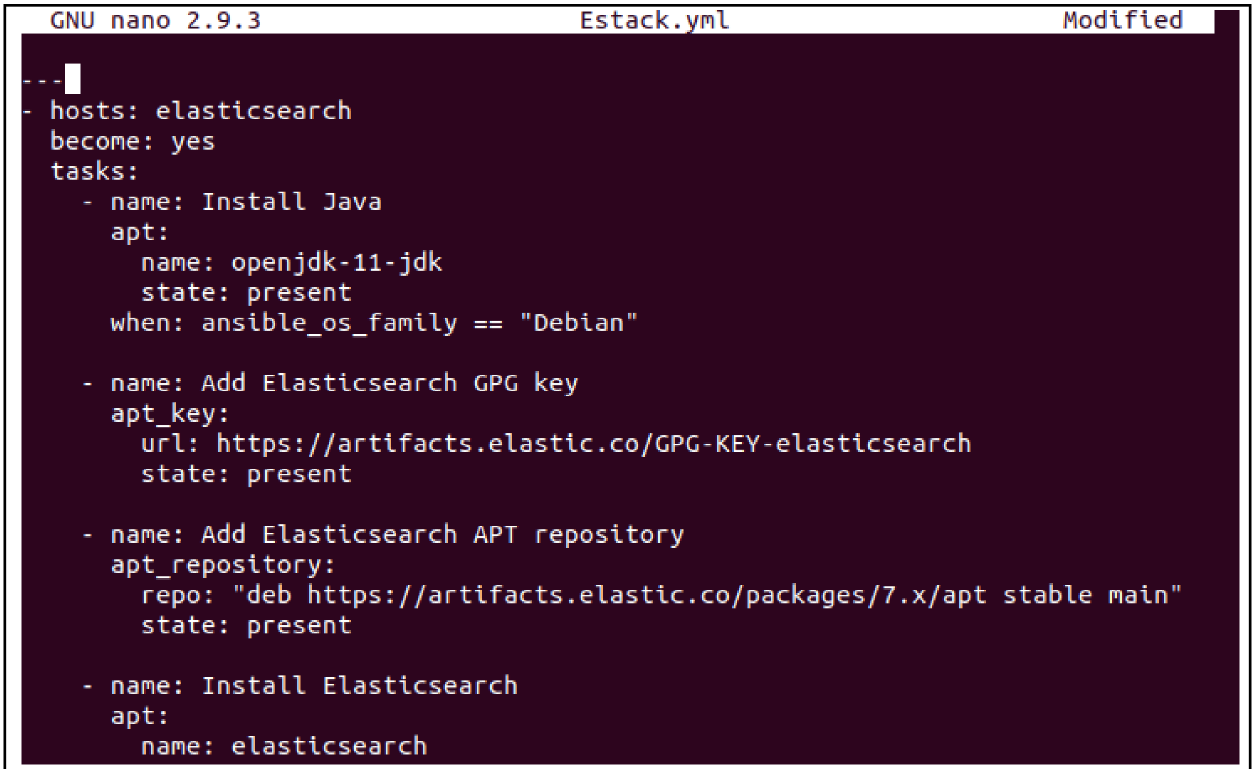
```
  when: nagios_plugins_exists|failed

- name: Enable rewrite Apache module
  apache2_module: state=present name={{ item }}
  with_items:
    - rewrite
    - cgi
  notify:
    - restart apache

- name: Start nagios service
  service: name=nagios state=started

handlers:
- name: restart apache
  service: name=apache2 state=restarted
```

This is the code in installing the nagios inside the file



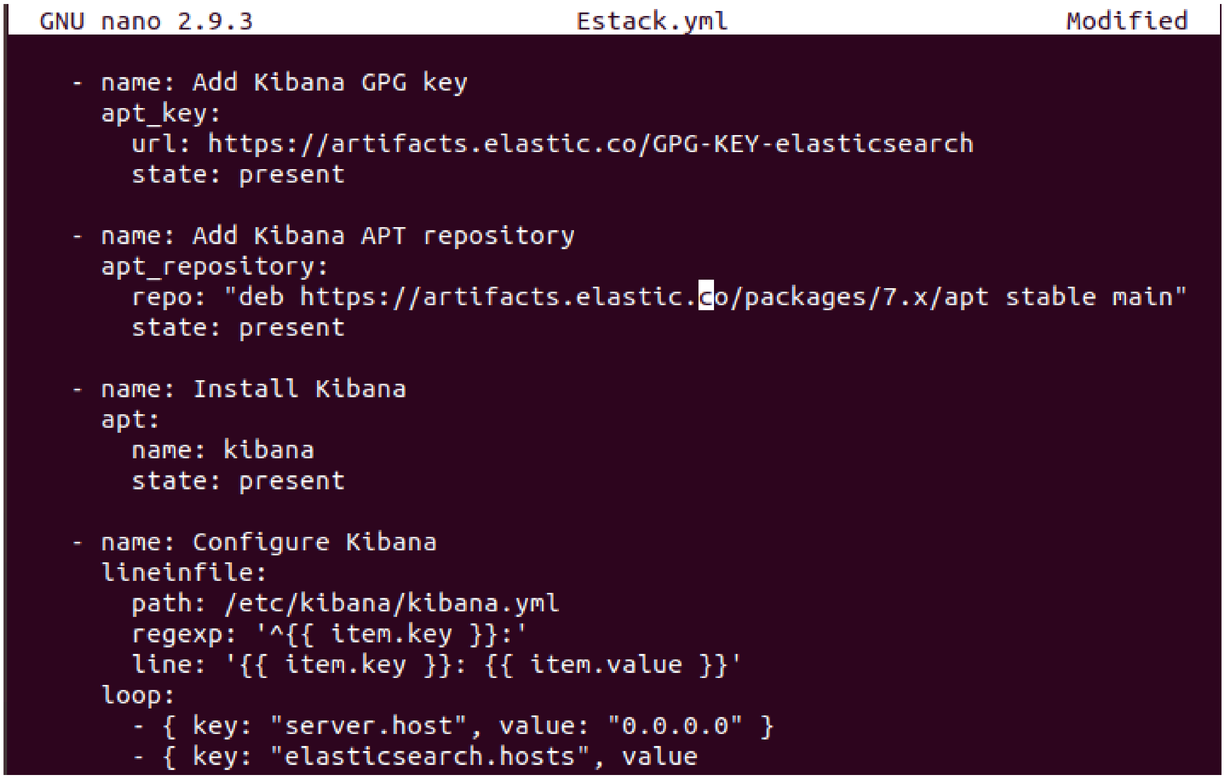
```
  GNU nano 2.9.3                    Estack.yml                    Modified

---
- hosts: elasticsearch
  become: yes
  tasks:
    - name: Install Java
      apt:
        name: openjdk-11-jdk
        state: present
      when: ansible_os_family == "Debian"

    - name: Add Elasticsearch GPG key
      apt_key:
        url: https://artifacts.elastic.co/GPG-KEY-elasticsearch
        state: present

    - name: Add Elasticsearch APT repository
      apt_repository:
        repo: "deb https://artifacts.elastic.co/packages/7.x/apt stable main"
        state: present

    - name: Install Elasticsearch
      apt:
        name: elasticsearch
```



```
  GNU nano 2.9.3                    Estack.yml                    Modified

    - name: Add Kibana GPG key
      apt_key:
        url: https://artifacts.elastic.co/GPG-KEY-elasticsearch
        state: present

    - name: Add Kibana APT repository
      apt_repository:
        repo: "deb https://artifacts.elastic.co/packages/7.x/apt stable main"
        state: present

    - name: Install Kibana
      apt:
        name: kibana
        state: present

    - name: Configure Kibana
      lineinfile:
        path: /etc/kibana/kibana.yml
        regexp: '^{{ item.key }}:'
        line: '{{ item.key }}: {{ item.value }}'
      loop:
        - { key: "server.host", value: "0.0.0.0" }
        - { key: "elasticsearch.hosts", value
```

GitHub link: https://github.com/qkmcampo/CPE_MIDEXAM_CAMPO

Conclusions: In this midterm exam we created and designed a workflow that installs, configures and manages enterprise availability. Unfortunately I haven’t installed the nagios properly. It has an error that I can't define what it is. Despite following the necessary steps to create the repository, clone it, and develop the playbook structure, I unfortunately did not succeed in completing the installation of the services