Name:Calderon Ricardo B.	**Date Performed:April 01, 2024**
Course/Section:CPE232 - CPE31S1	**Date Submitted:April 02, 2024**
Instructor: Dr. Jonathan Taylar	**Semester and SY: 2nd Sem / 2023-2024**

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

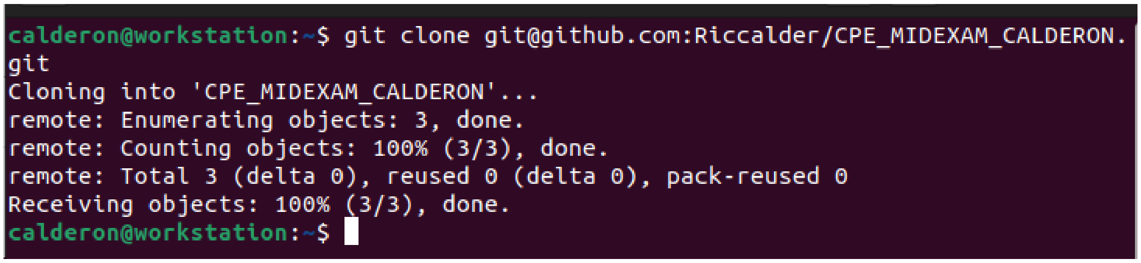
1. Objectives

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. Infrastructure as Code (IaC) and Ansible (unrepo.com)

2. Instructions

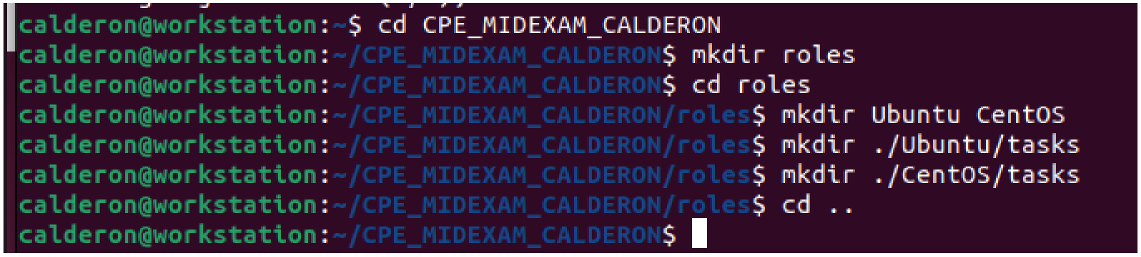
1. Create a repository in your GitHub account and label it CPE_MIDEXAM_SURNAME.
2. Clone the repository and do the following:
   2.1. Create an Ansible playbook that does the following with an input of a config.yaml file and arranged Inventory file:
   2.2. Install and configure Elastic Stack in separate 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)



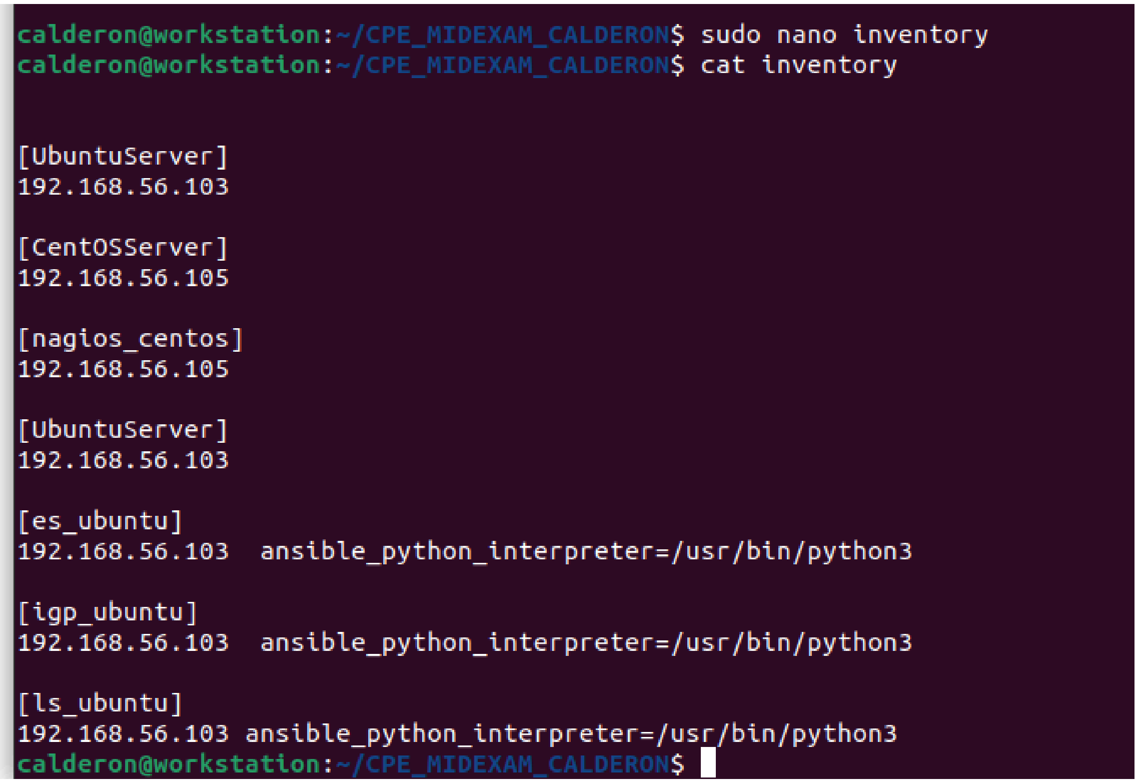
```
calderon@workstation:~$ git clone git@github.com:Riccalder/CPE_MIDEXAM_CALDERON.
git
Cloning into 'CPE_MIDEXAM_CALDERON'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
calderon@workstation:~$ 
```

This command clones a Git repository hosted on GitHub with the URL `git@github.com:Riccalder/calderon_hoa9.git` into a local directory named CPE_MIDEXAM_CALDERON

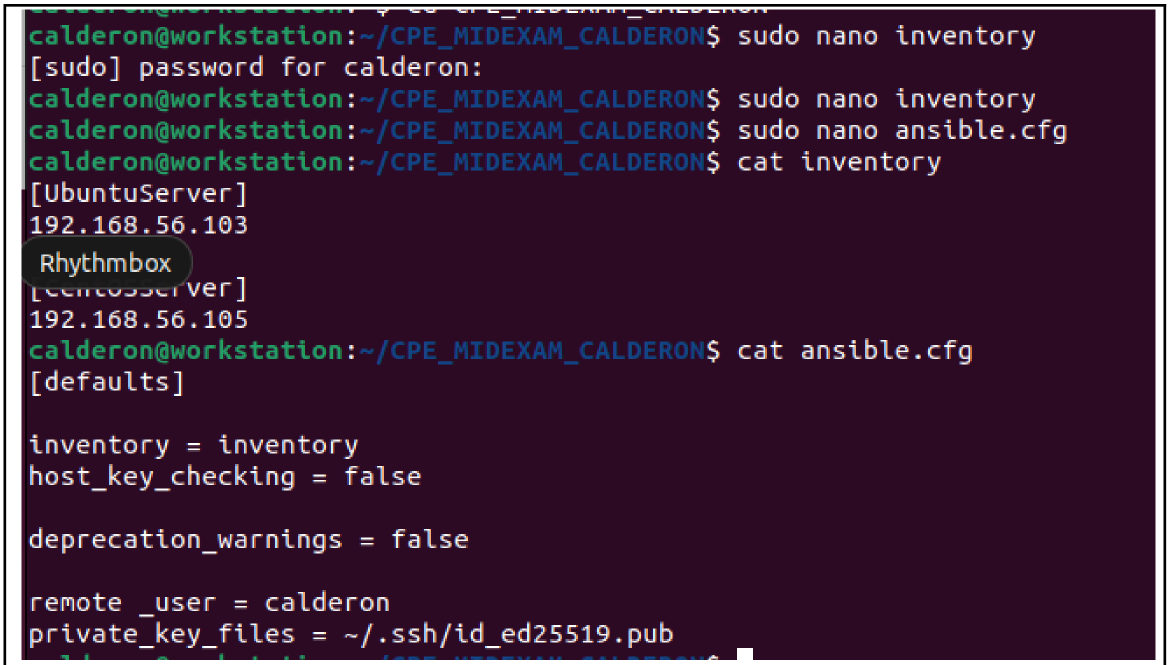


```
calderon@workstation:~$ cd CPE_MIDEXAM_CALDERON
calderon@workstation:~/CPE_MIDEXAM_CALDERON$ mkdir roles
calderon@workstation:~/CPE_MIDEXAM_CALDERON$ cd roles
calderon@workstation:~/CPE_MIDEXAM_CALDERON/roles$ mkdir Ubuntu CentOS
calderon@workstation:~/CPE_MIDEXAM_CALDERON/roles$ mkdir ./Ubuntu/tasks
calderon@workstation:~/CPE_MIDEXAM_CALDERON/roles$ mkdir ./CentOS/tasks
calderon@workstation:~/CPE_MIDEXAM_CALDERON/roles$ cd ..
calderon@workstation:~/CPE_MIDEXAM_CALDERON$ 
```

I've organized Ansible roles into separate directories for Ubuntu and CentOS, each containing a 'tasks' subdirectory for defining server-specific tasks.

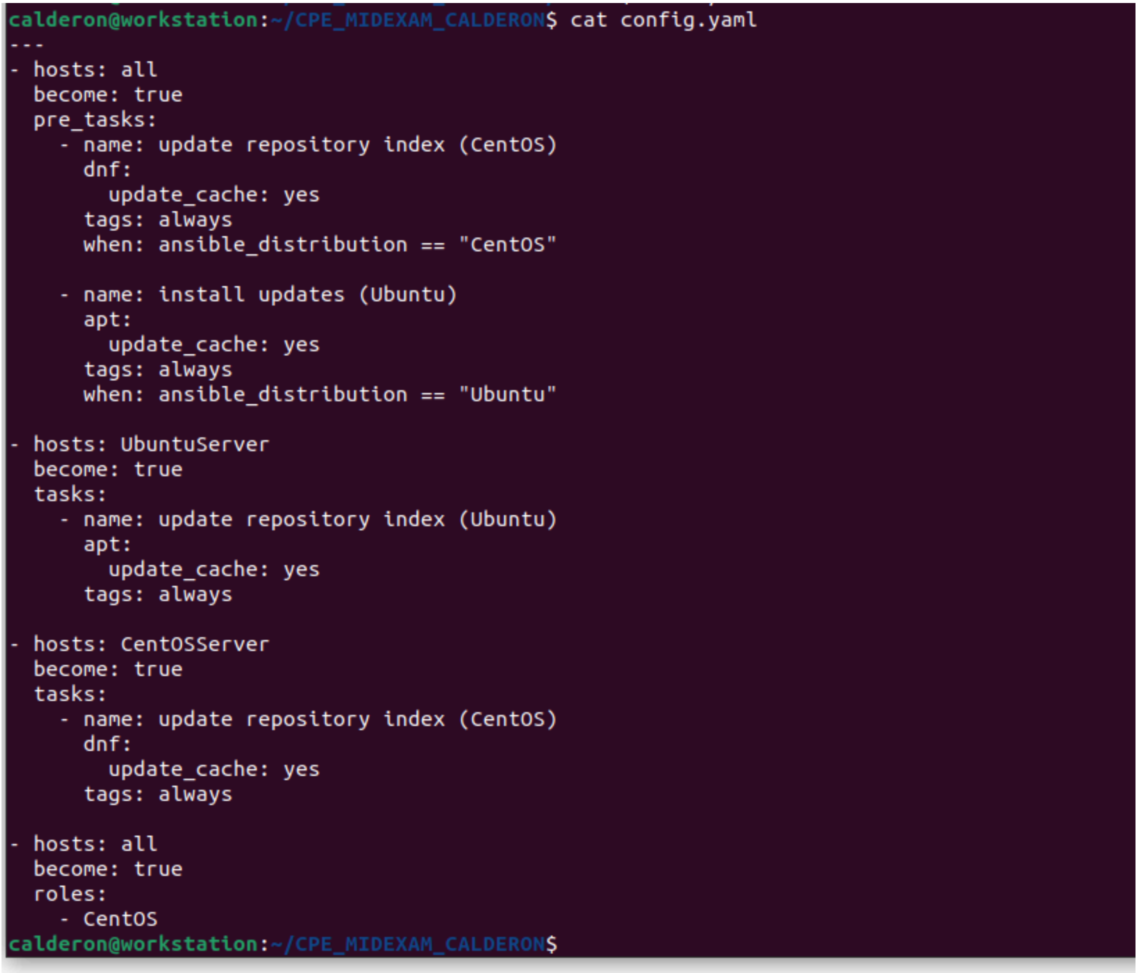


```
calderon@workstation:~/CPE_MIDEXAM_CALDERON$ sudo nano inventory
calderon@workstation:~/CPE_MIDEXAM_CALDERON$ cat inventory


[UbuntuServer]
192.168.56.103

[CentOSServer]
192.168.56.105

[nagios_centos]
192.168.56.105

[UbuntuServer]
192.168.56.103

[es_ubuntu]
192.168.56.103  ansible_python_interpreter=/usr/bin/python3

[igp_ubuntu]
192.168.56.103  ansible_python_interpreter=/usr/bin/python3

[ls_ubuntu]
192.168.56.103 ansible_python_interpreter=/usr/bin/python3
calderon@workstation:~/CPE_MIDEXAM_CALDERON$ 
```



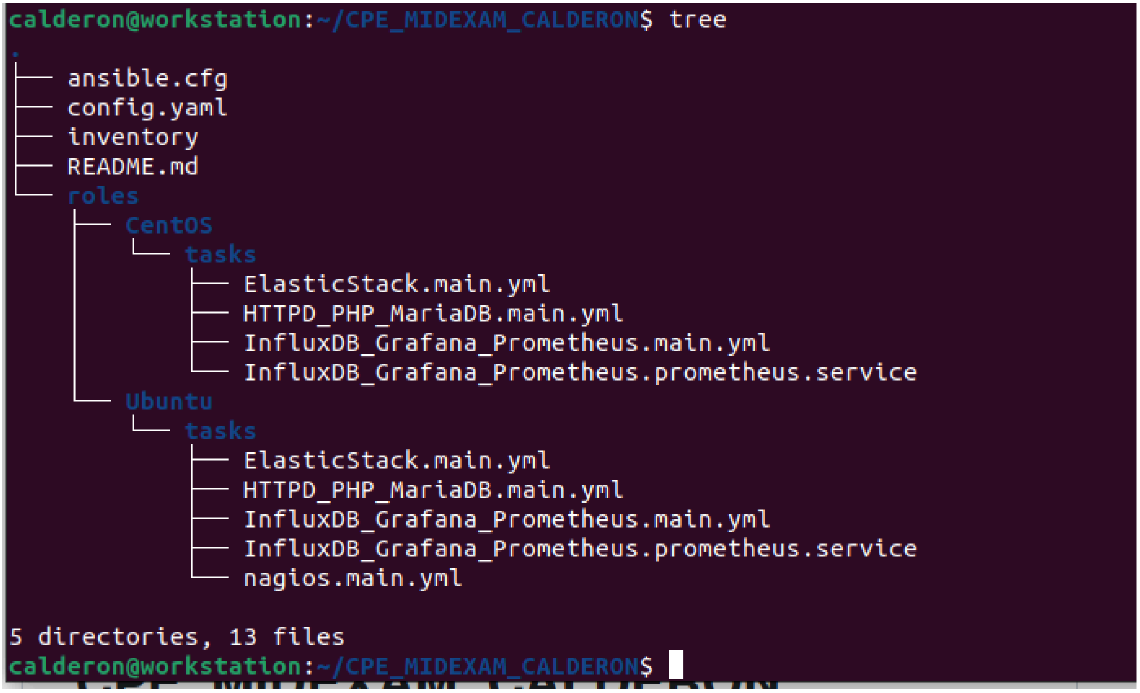
```
calderon@workstation:~$ cd CPE_MIDEXAM_CALDERON
calderon@workstation:~/CPE_MIDEXAM_CALDERON$ sudo nano inventory
[sudo] password for calderon:
calderon@workstation:~/CPE_MIDEXAM_CALDERON$ sudo nano inventory
calderon@workstation:~/CPE_MIDEXAM_CALDERON$ sudo nano ansible.cfg
calderon@workstation:~/CPE_MIDEXAM_CALDERON$ cat inventory
[UbuntuServer]
192.168.56.103

[CentosServer]
192.168.56.105
calderon@workstation:~/CPE_MIDEXAM_CALDERON$ cat ansible.cfg
[defaults]

inventory = inventory
host_key_checking = false

deprecation_warnings = false

remote _user = calderon
private_key_files = ~/.ssh/id_ed25519.pub
calderon@workstation:~/CPE_MIDEXAM_CALDERON$ 
```

I've created an inventory file listing Ubuntu and CentOS servers, and configured Ansible settings in the `ansible.cfg` file, including the inventory path, host key checking status.



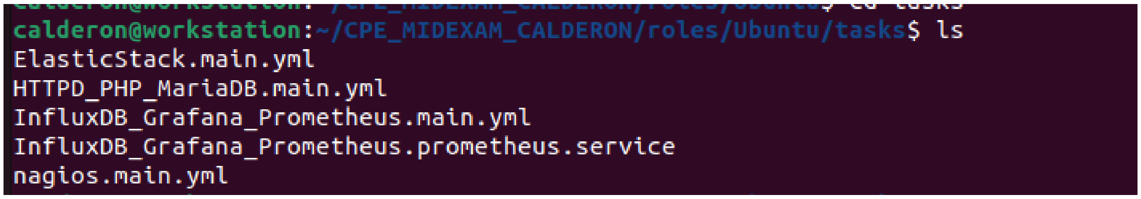
```
calderon@workstation:~/CPE_MIDEXAM_CALDERON$ cat config.yaml
---
- hosts: all
  become: true
  pre_tasks:
    - name: update repository index (CentOS)
      dnf:
        update_cache: yes
      tags: always
      when: ansible_distribution == "CentOS"

    - name: install updates (Ubuntu)
      apt:
        update_cache: yes
      tags: always
      when: ansible_distribution == "Ubuntu"

- hosts: UbuntuServer
  become: true
  tasks:
    - name: update repository index (Ubuntu)
      apt:
        update_cache: yes
      tags: always

- hosts: CentOSServer
  become: true
  tasks:
    - name: update repository index (CentOS)
      dnf:
        update_cache: yes
      tags: always

- hosts: all
  become: true
  roles:
    - CentOS
calderon@workstation:~/CPE_MIDEXAM_CALDERON$
```

I've created an Ansible playbook named `config.yaml` with tasks to update the repository index and install updates on Ubuntu and CentOS servers.

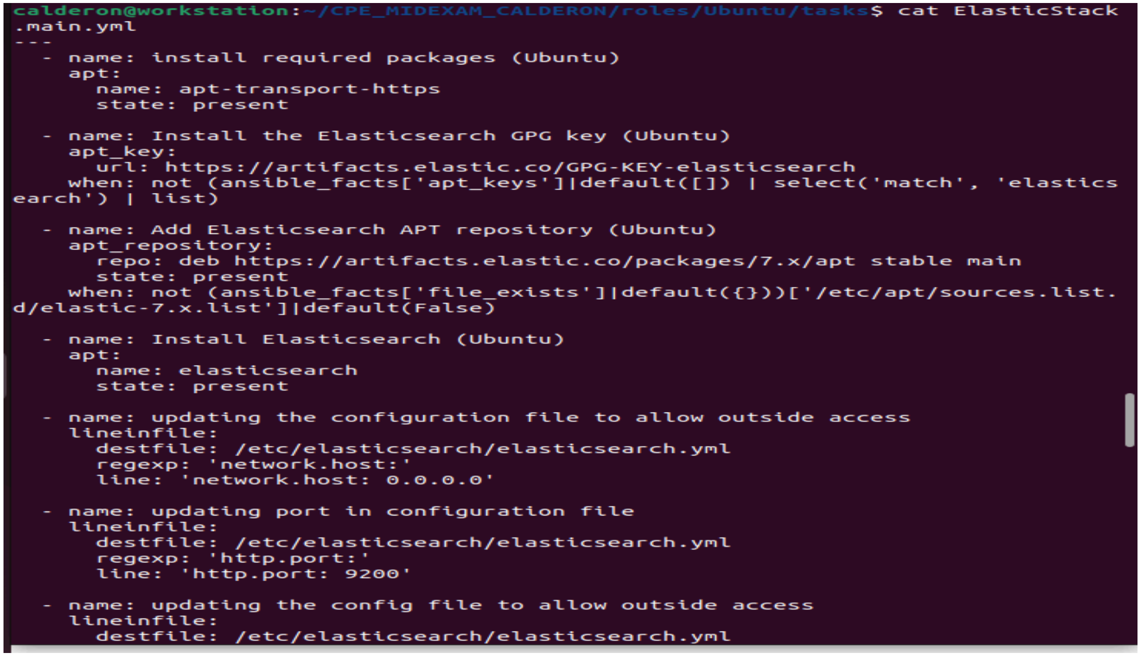


```
calderon@workstation:~/CPE_MIDEXAM_CALDERON$ tree
.
├── ansible.cfg
├── config.yaml
├── inventory
├── README.md
└── roles
    ├── CentOS
    │   └── tasks
    │       ├── ElasticStack.main.yml
    │       ├── HTTPD_PHP_MariaDB.main.yml
    │       ├── InfluxDB_Grafana_Prometheus.main.yml
    │       └── InfluxDB_Grafana_Prometheus.prometheus.service
    └── Ubuntu
        └── tasks
            ├── ElasticStack.main.yml
            ├── HTTPD_PHP_MariaDB.main.yml
            ├── InfluxDB_Grafana_Prometheus.main.yml
            ├── InfluxDB_Grafana_Prometheus.prometheus.service
            └── nagios.main.yml

5 directories, 13 files
calderon@workstation:~/CPE_MIDEXAM_CALDERON$ 
```

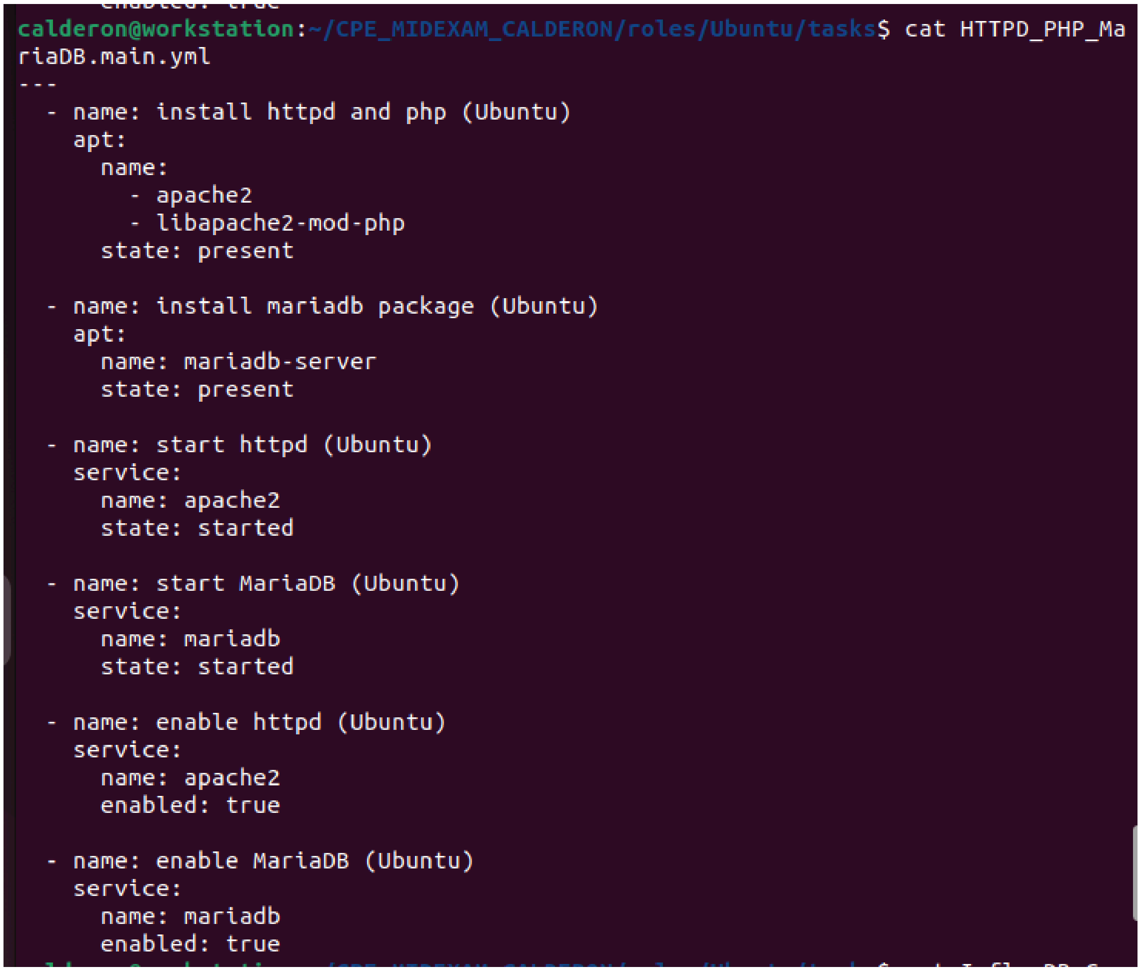
Through tree, viewing directory with subdirectories for roles (CentOS and Ubuntu), each containing task files.



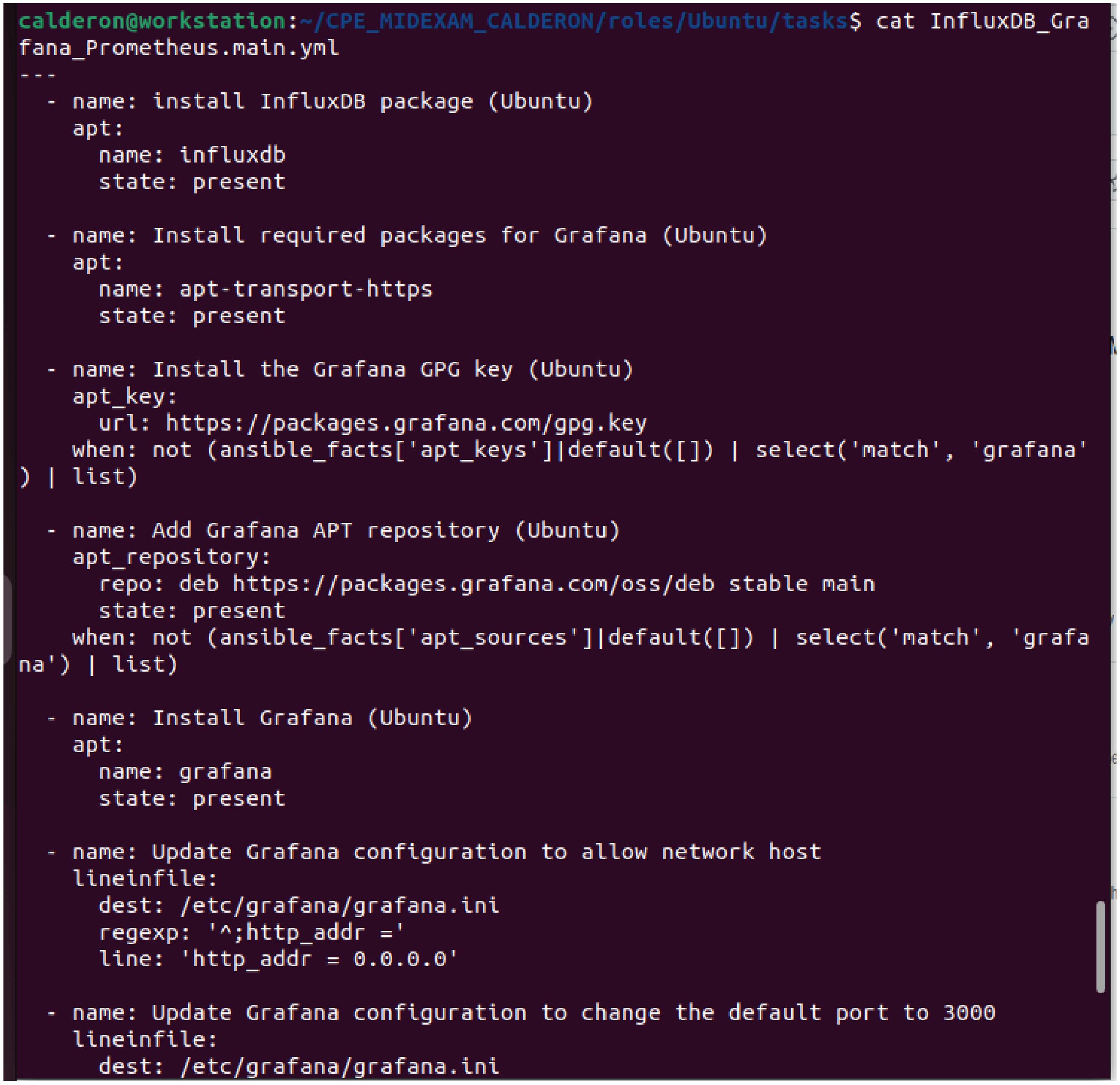
```
calderon@workstation:~/CPE_MIDEXAM_CALDERON/roles/Ubuntu$ cd tasks
calderon@workstation:~/CPE_MIDEXAM_CALDERON/roles/Ubuntu/tasks$ ls
ElasticStack.main.yml
HTTPD_PHP_MariaDB.main.yml
InfluxDB_Grafana_Prometheus.main.yml
InfluxDB_Grafana_Prometheus.prometheus.service
nagios.main.yml
```



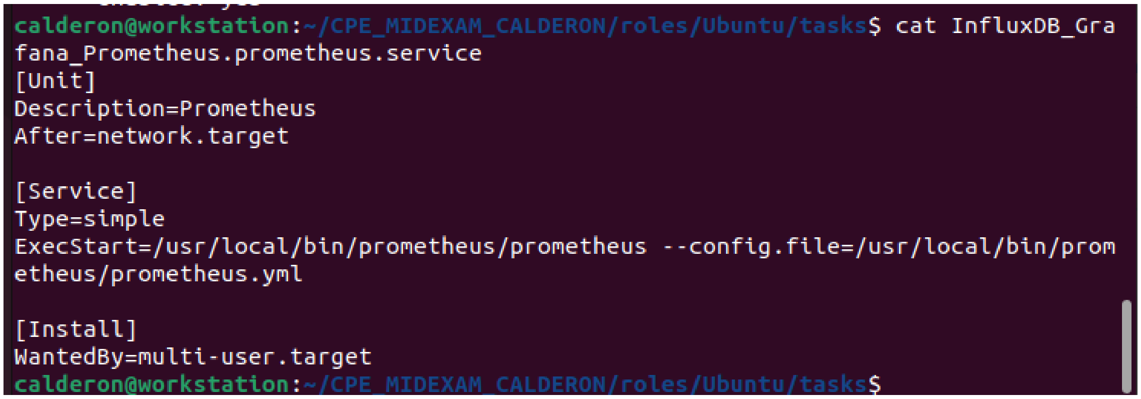
```
calderon@workstation:~/CPE_MIDEXAM_CALDERON/roles/Ubuntu/tasks$ cat ElasticStack
.main.yml
---
  - name: install required packages (Ubuntu)
    apt:
      name: apt-transport-https
      state: present

  - name: Install the Elasticsearch GPG key (Ubuntu)
    apt_key:
      url: https://artifacts.elastic.co/GPG-KEY-elasticsearch
    when: not (ansible_facts['apt_keys']|default([]) | select('match', 'elastics
earch') | list)

  - name: Add Elasticsearch APT repository (Ubuntu)
    apt_repository:
      repo: deb https://artifacts.elastic.co/packages/7.x/apt stable main
      state: present
    when: not (ansible_facts['file_exists']|default({}))['/etc/apt/sources.list.
d/elastic-7.x.list']|default(False)

  - name: Install Elasticsearch (Ubuntu)
    apt:
      name: elasticsearch
      state: present

  - name: updating the configuration file to allow outside access
    lineinfile:
      destfile: /etc/elasticsearch/elasticsearch.yml
      regexp: 'network.host:'
      line: 'network.host: 0.0.0.0'

  - name: updating port in configuration file
    lineinfile:
      destfile: /etc/elasticsearch/elasticsearch.yml
      regexp: 'http.port:'
      line: 'http.port: 9200'

  - name: updating the config file to allow outside access
    lineinfile:
      destfile: /etc/elasticsearch/elasticsearch.yml
```



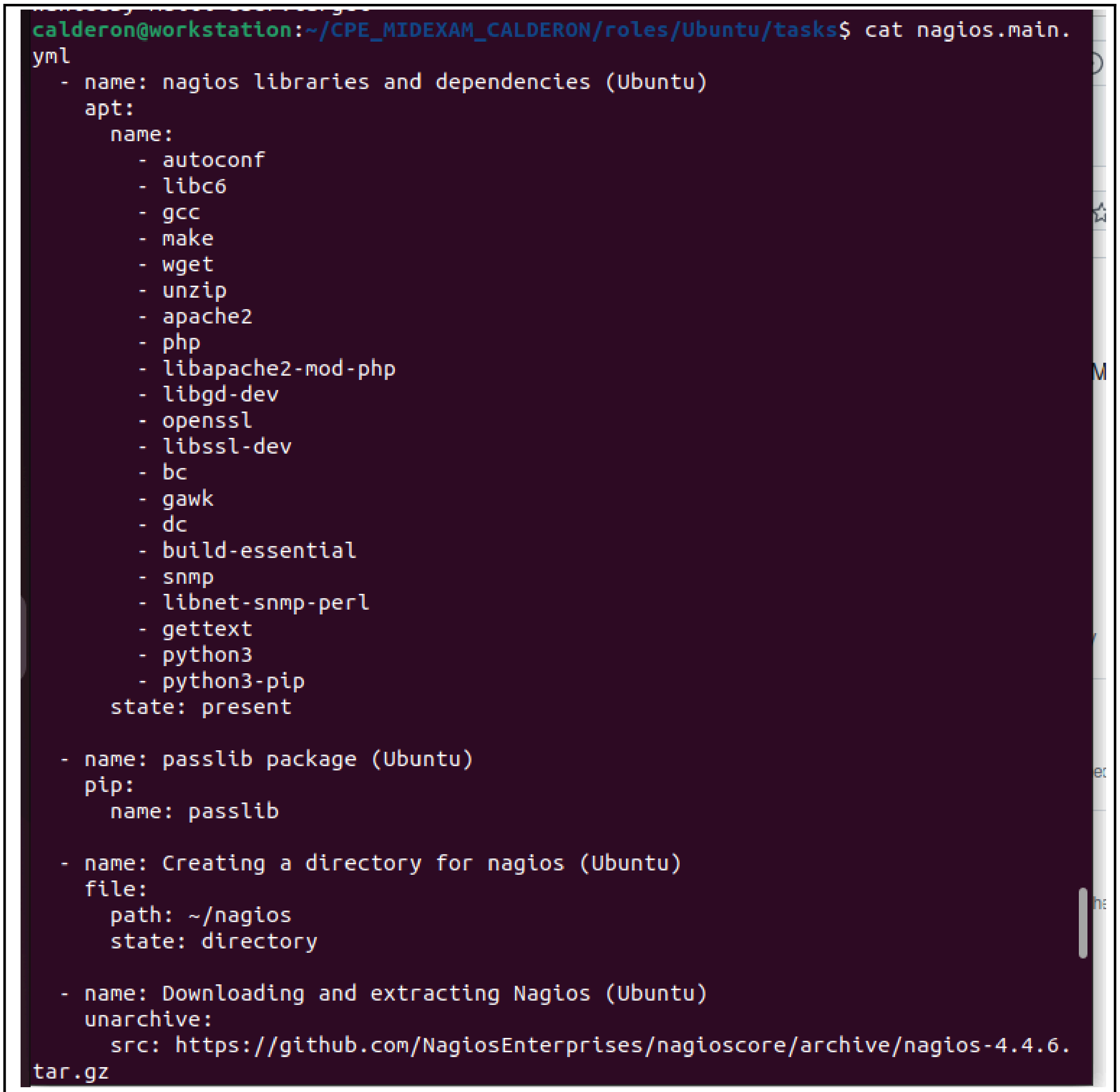
```
      enabled: true
calderon@workstation:~/CPE_MIDEXAM_CALDERON/roles/Ubuntu/tasks$ cat HTTPD_PHP_Ma
riaDB.main.yml
---
  - name: install httpd and php (Ubuntu)
    apt:
      name:
        - apache2
        - libapache2-mod-php
      state: present

  - name: install mariadb package (Ubuntu)
    apt:
      name: mariadb-server
      state: present

  - name: start httpd (Ubuntu)
    service:
      name: apache2
      state: started

  - name: start MariaDB (Ubuntu)
    service:
      name: mariadb
      state: started

  - name: enable httpd (Ubuntu)
    service:
      name: apache2
      enabled: true

  - name: enable MariaDB (Ubuntu)
    service:
      name: mariadb
      enabled: true
```



```
calderon@workstation:~/CPE_MIDEXAM_CALDERON/roles/Ubuntu/tasks$ cat InfluxDB_Gra
fana_Prometheus.main.yml
---
  - name: install InfluxDB package (Ubuntu)
    apt:
      name: influxdb
      state: present

  - name: Install required packages for Grafana (Ubuntu)
    apt:
      name: apt-transport-https
      state: present

  - name: Install the Grafana GPG key (Ubuntu)
    apt_key:
      url: https://packages.grafana.com/gpg.key
    when: not (ansible_facts['apt_keys']|default([]) | select('match', 'grafana'
) | list)

  - name: Add Grafana APT repository (Ubuntu)
    apt_repository:
      repo: deb https://packages.grafana.com/oss/deb stable main
      state: present
    when: not (ansible_facts['apt_sources']|default([]) | select('match', 'grafa
na') | list)

  - name: Install Grafana (Ubuntu)
    apt:
      name: grafana
      state: present

  - name: Update Grafana configuration to allow network host
    lineinfile:
      dest: /etc/grafana/grafana.ini
      regexp: '^;http_addr ='
      line: 'http_addr = 0.0.0.0'

  - name: Update Grafana configuration to change the default port to 3000
    lineinfile:
      dest: /etc/grafana/grafana.ini
```



```
calderon@workstation:~/CPE_MIDEXAM_CALDERON/roles/Ubuntu/tasks$ cat InfluxDB_Gra
fana_Prometheus.prometheus.service
[Unit]
Description=Prometheus
After=network.target

[Service]
Type=simple
ExecStart=/usr/local/bin/prometheus/prometheus --config.file=/usr/local/bin/prom
etheus/prometheus.yml

[Install]
WantedBy=multi-user.target
calderon@workstation:~/CPE_MIDEXAM_CALDERON/roles/Ubuntu/tasks$
```

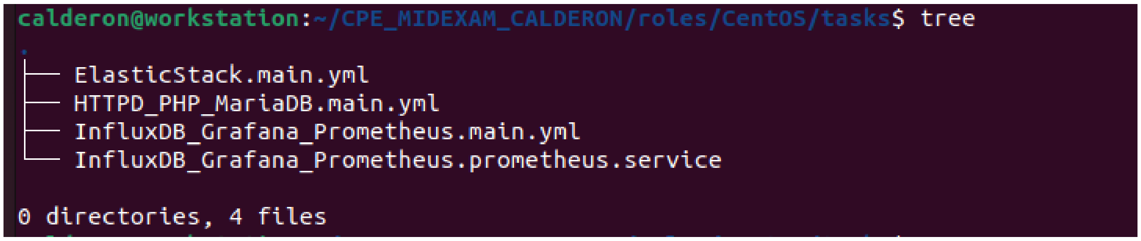


```
calderon@workstation:~/CPE_MIDEXAM_CALDERON/roles/Ubuntu/tasks$ cat nagios.main.
yml
  - name: nagios libraries and dependencies (Ubuntu)
    apt:
      name:
        - autoconf
        - libc6
        - gcc
        - make
        - wget
        - unzip
        - apache2
        - php
        - libapache2-mod-php
        - libgd-dev
        - openssl
        - libssl-dev
        - bc
        - gawk
        - dc
        - build-essential
        - snmp
        - libnet-snmp-perl
        - gettext
        - python3
        - python3-pip
      state: present

  - name: passlib package (Ubuntu)
    pip:
      name: passlib

  - name: Creating a directory for nagios (Ubuntu)
    file:
      path: ~/nagios
      state: directory

  - name: Downloading and extracting Nagios (Ubuntu)
    unarchive:
      src: https://github.com/NagiosEnterprises/nagioscore/archive/nagios-4.4.6.
tar.gz
```

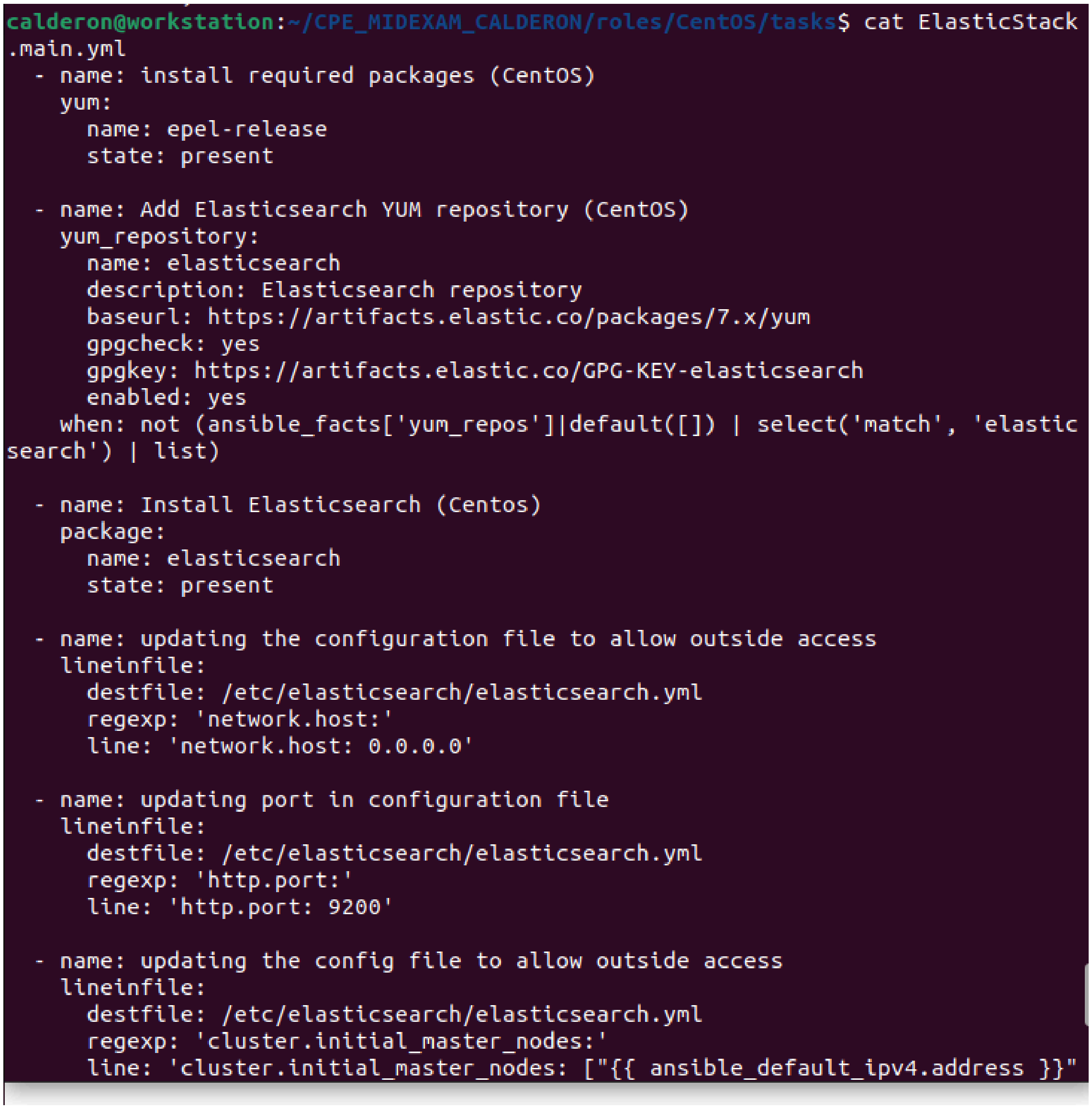
Those screenshots on top are from Ubuntu contain a series of tasks necessary for configuring and managing the corresponding service.

From roles Ubuntu tasks

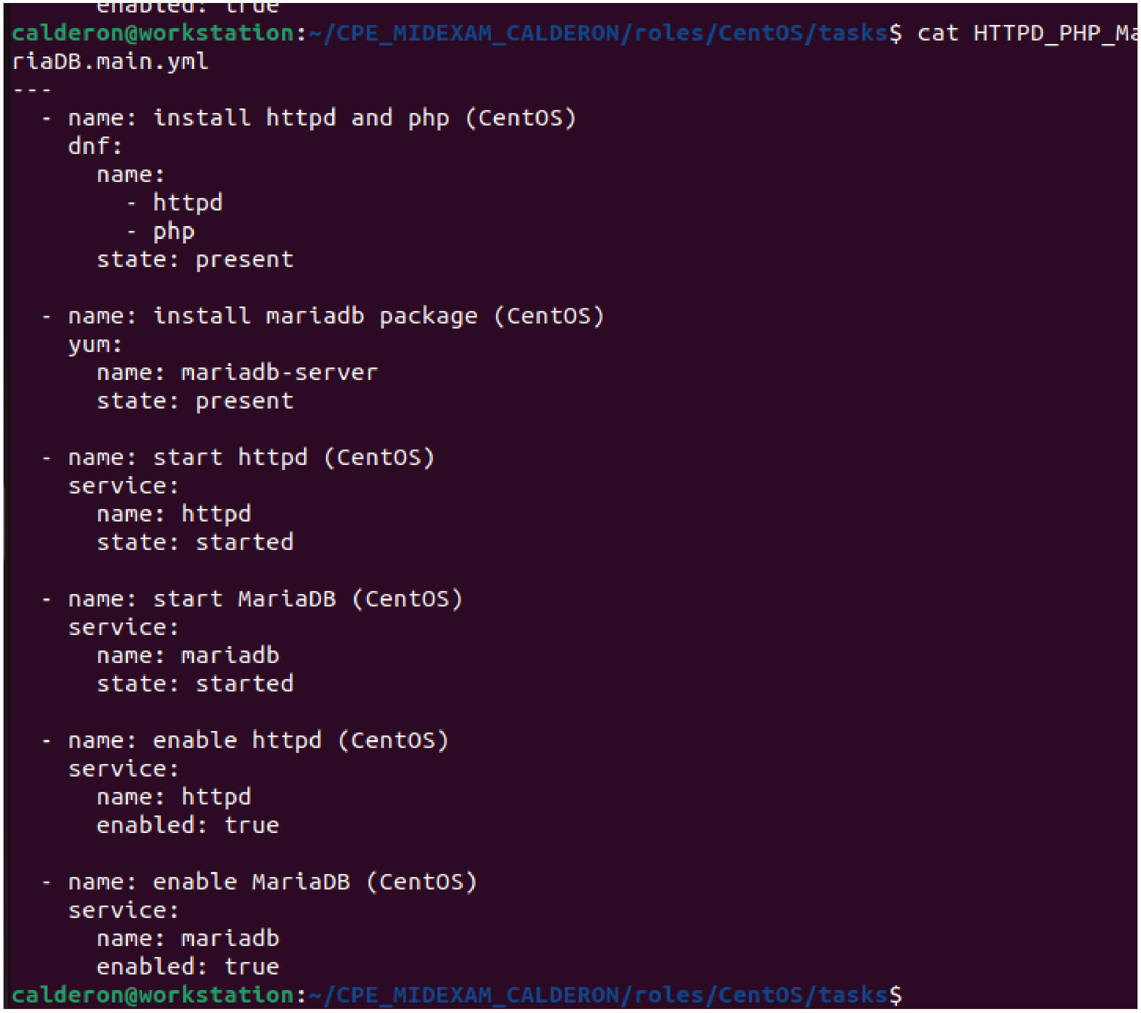
```
├── ElasticStack.main.yml

├── HTTPD_PHP_MariaDB.main.yml

├── InfluxDB_Grafana_Prometheus.main.yml

├── InfluxDB_Grafana_Prometheus.prometheus.service

└── nagios.main.yml
```



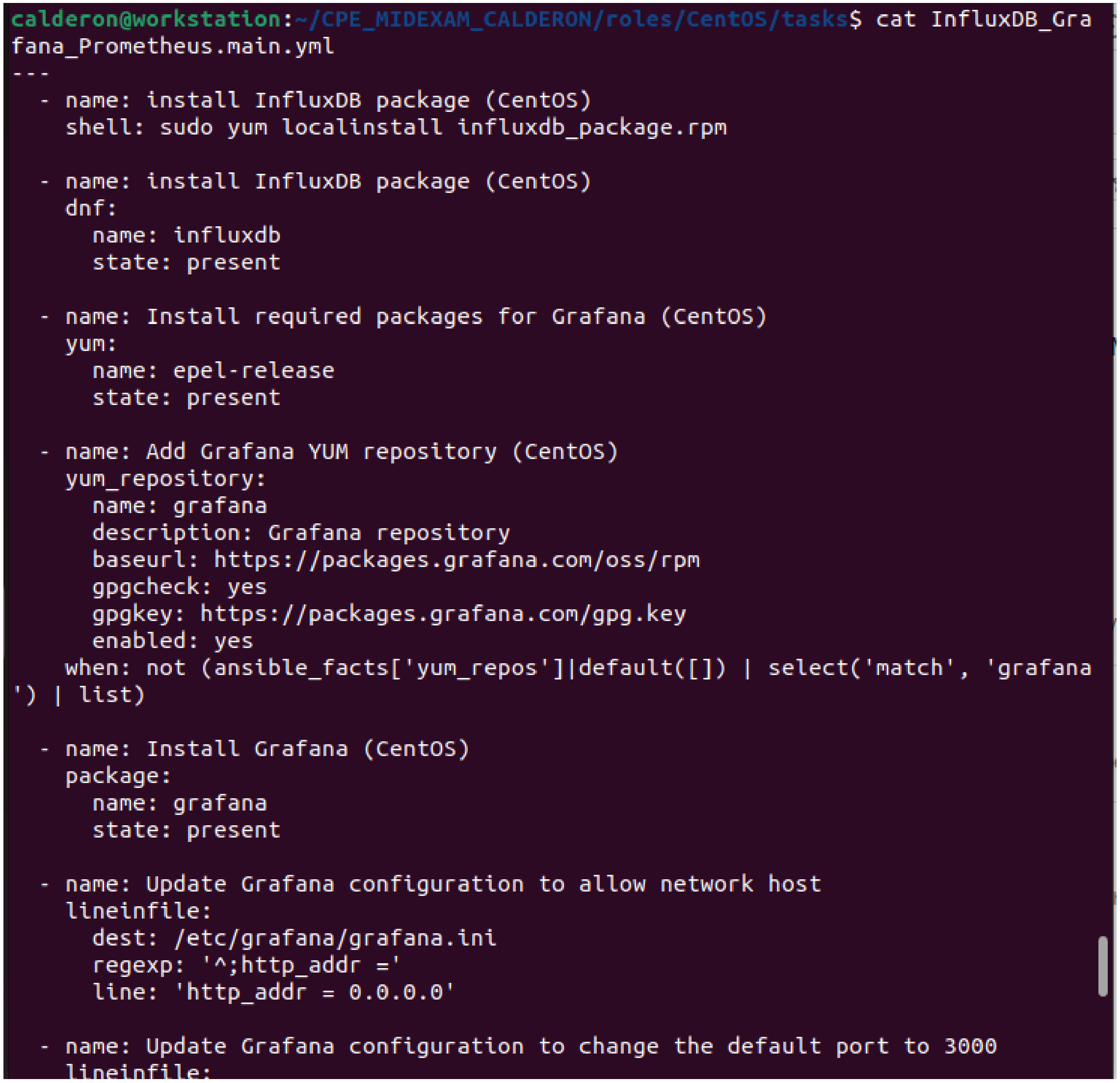
```
calderon@workstation:~/CPE_MIDEXAM_CALDERON/roles/CentOS/tasks$ tree
.
├── ElasticStack.main.yml
├── HTTPD_PHP_MariaDB.main.yml
├── InfluxDB_Grafana_Prometheus.main.yml
└── InfluxDB_Grafana_Prometheus.prometheus.service

0 directories, 4 files
```



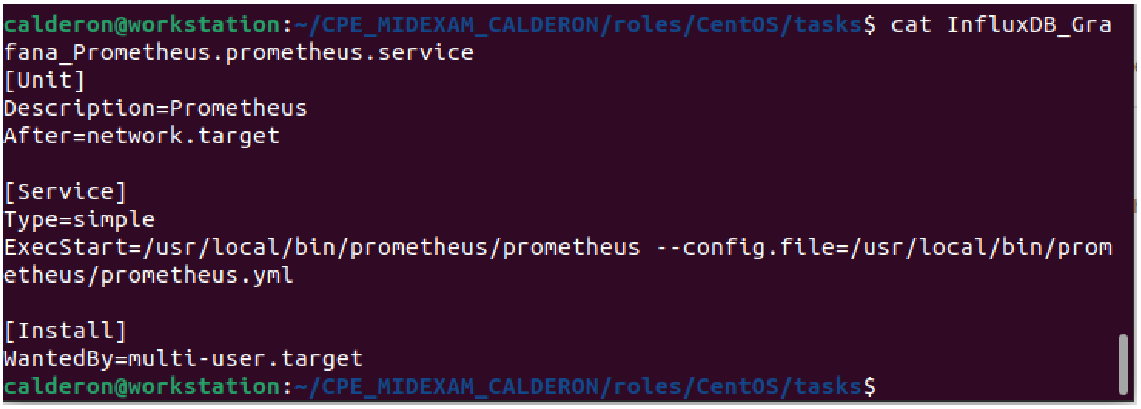
```
calderon@workstation:~/CPE_MIDEXAM_CALDERON/roles/CentOS/tasks$ cat ElasticStack
.main.yml
  - name: install required packages (CentOS)
    yum:
      name: epel-release
      state: present

  - name: Add Elasticsearch YUM repository (CentOS)
    yum_repository:
      name: elasticsearch
      description: Elasticsearch repository
      baseurl: https://artifacts.elastic.co/packages/7.x/yum
      gpgcheck: yes
      gpgkey: https://artifacts.elastic.co/GPG-KEY-elasticsearch
      enabled: yes
    when: not (ansible_facts['yum_repos']|default([]) | select('match', 'elastic
search') | list)

  - name: Install Elasticsearch (Centos)
    package:
      name: elasticsearch
      state: present

  - name: updating the configuration file to allow outside access
    lineinfile:
      destfile: /etc/elasticsearch/elasticsearch.yml
      regexp: 'network.host:'
      line: 'network.host: 0.0.0.0'

  - name: updating port in configuration file
    lineinfile:
      destfile: /etc/elasticsearch/elasticsearch.yml
      regexp: 'http.port:'
      line: 'http.port: 9200'

  - name: updating the config file to allow outside access
    lineinfile:
      destfile: /etc/elasticsearch/elasticsearch.yml
      regexp: 'cluster.initial_master_nodes:'
      line: 'cluster.initial_master_nodes: ["{{ ansible_default_ipv4.address }}"
```



```
      enabled: true
calderon@workstation:~/CPE_MIDEXAM_CALDERON/roles/CentOS/tasks$ cat HTTPD_PHP_Ma
riaDB.main.yml
---
  - name: install httpd and php (CentOS)
    dnf:
      name:
        - httpd
        - php
      state: present

  - name: install mariadb package (CentOS)
    yum:
      name: mariadb-server
      state: present

  - name: start httpd (CentOS)
    service:
      name: httpd
      state: started

  - name: start MariaDB (CentOS)
    service:
      name: mariadb
      state: started

  - name: enable httpd (CentOS)
    service:
      name: httpd
      enabled: true

  - name: enable MariaDB (CentOS)
    service:
      name: mariadb
      enabled: true
calderon@workstation:~/CPE_MIDEXAM_CALDERON/roles/CentOS/tasks$
```



```
calderon@workstation:~/CPE_MIDEXAM_CALDERON/roles/CentOS/tasks$ cat InfluxDB_Gra
fana_Prometheus.main.yml
---
  - name: install InfluxDB package (CentOS)
    shell: sudo yum localinstall influxdb_package.rpm

  - name: install InfluxDB package (CentOS)
    dnf:
      name: influxdb
      state: present

  - name: Install required packages for Grafana (CentOS)
    yum:
      name: epel-release
      state: present

  - name: Add Grafana YUM repository (CentOS)
    yum_repository:
      name: grafana
      description: Grafana repository
      baseurl: https://packages.grafana.com/oss/rpm
      gpgcheck: yes
      gpgkey: https://packages.grafana.com/gpg.key
      enabled: yes
    when: not (ansible_facts['yum_repos']|default([]) | select('match', 'grafana
') | list)

  - name: Install Grafana (CentOS)
    package:
      name: grafana
      state: present

  - name: Update Grafana configuration to allow network host
    lineinfile:
      dest: /etc/grafana/grafana.ini
      regexp: '^;http_addr ='
      line: 'http_addr = 0.0.0.0'

  - name: Update Grafana configuration to change the default port to 3000
    lineinfile:
```

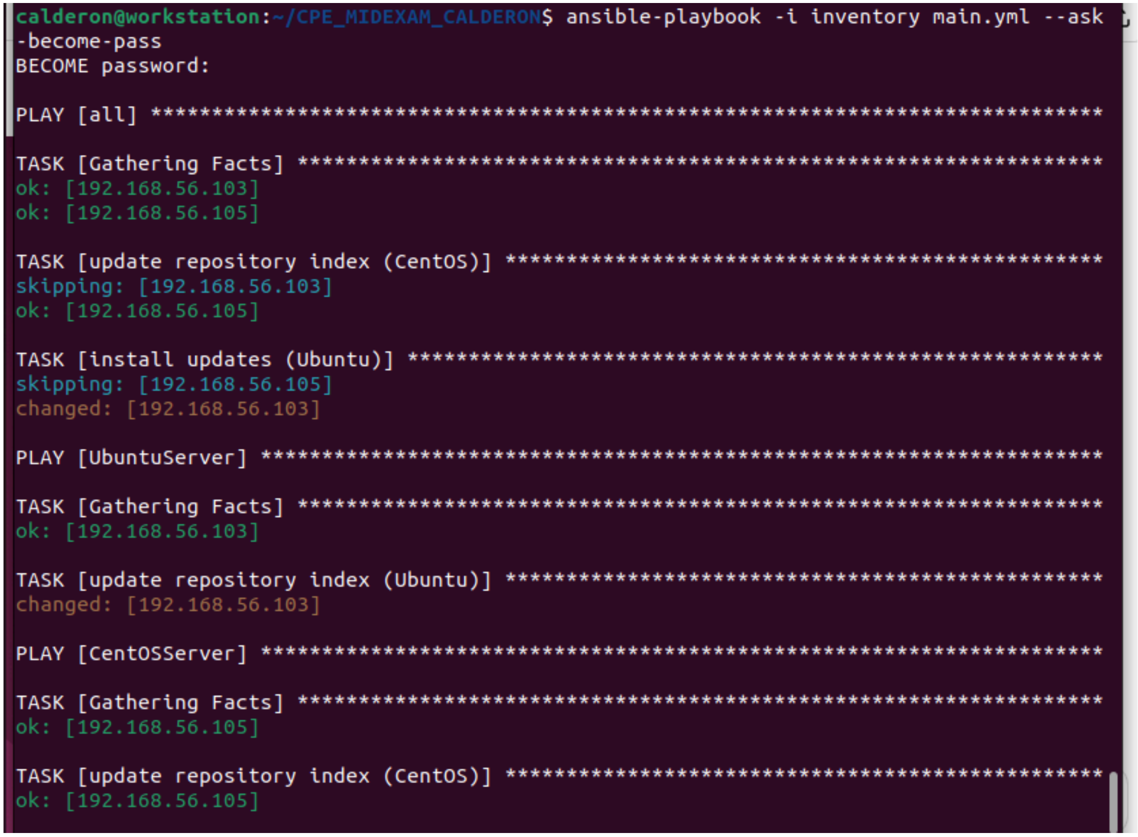


```
calderon@workstation:~/CPE_MIDEXAM_CALDERON/roles/CentOS/tasks$ cat InfluxDB_Gra
fana_Prometheus.prometheus.service
[Unit]
Description=Prometheus
After=network.target

[Service]
Type=simple
ExecStart=/usr/local/bin/prometheus/prometheus --config.file=/usr/local/bin/prom
etheus/prometheus.yml

[Install]
WantedBy=multi-user.target
calderon@workstation:~/CPE_MIDEXAM_CALDERON/roles/CentOS/tasks$
```

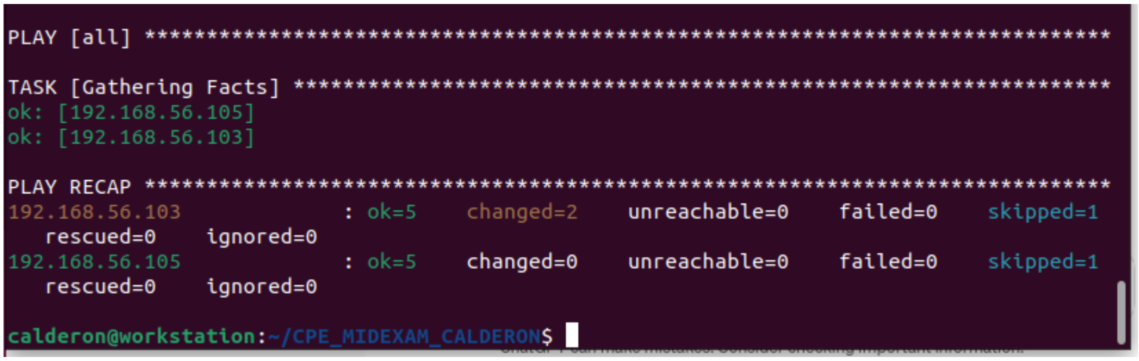
Those screenshots on the top are from centos contain a series of tasks necessary for configuring and managing the corresponding service.

From roles CentOS tasks

```
├── ElasticStack.main.yml
├── HTTPD_PHP_MariaDB.main.yml
├── InfluxDB_Grafana_Prometheus.main.yml
└── InfluxDB_Grafana_Prometheus.prometheus.service
```

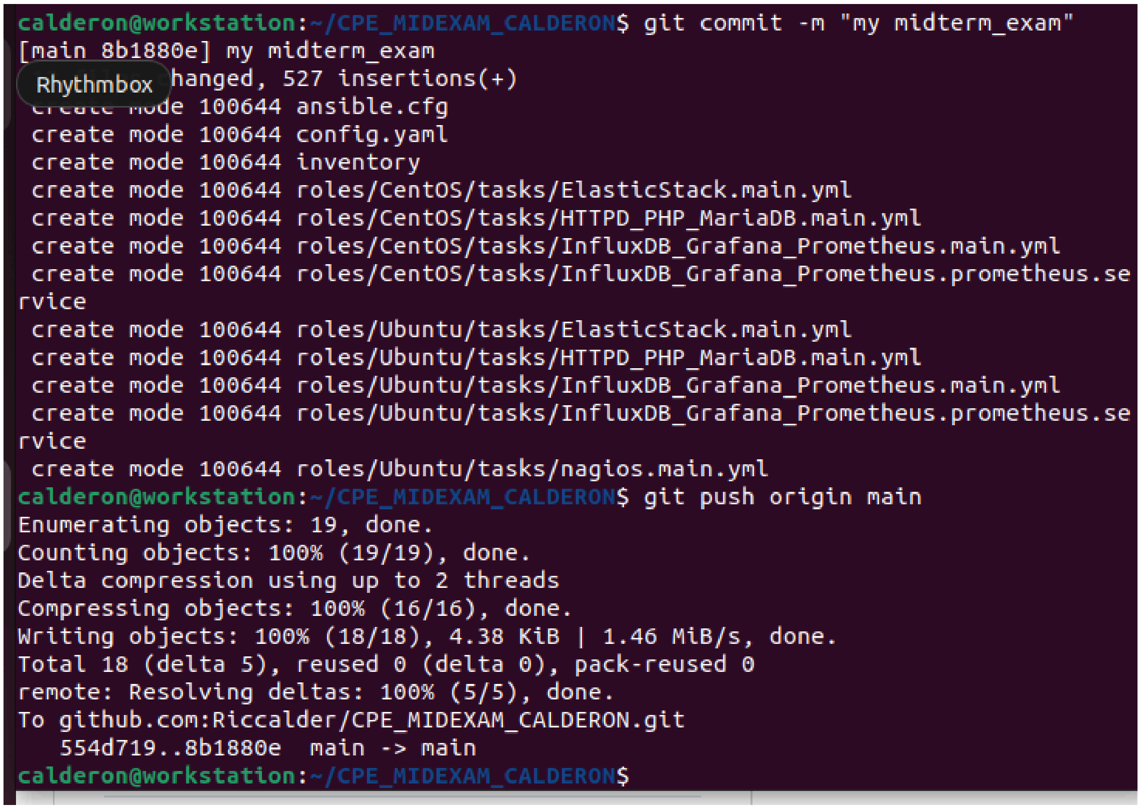


```
calderon@workstation:~/CPE_MIDEXAM_CALDERON$ ansible-playbook -i inventory main.yml --ask
-become-pass
BECOME password:

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

TASK [Gathering Facts] *************************************************************
ok: [192.168.56.103]
ok: [192.168.56.105]

TASK [update repository index (CentOS)] ********************************************
skipping: [192.168.56.103]
ok: [192.168.56.105]

TASK [install updates (Ubuntu)] ****************************************************
skipping: [192.168.56.105]
changed: [192.168.56.103]

PLAY [UbuntuServer] ****************************************************************

TASK [Gathering Facts] *************************************************************
ok: [192.168.56.103]

TASK [update repository index (Ubuntu)] ********************************************
changed: [192.168.56.103]

PLAY [CentOSServer] ****************************************************************

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

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

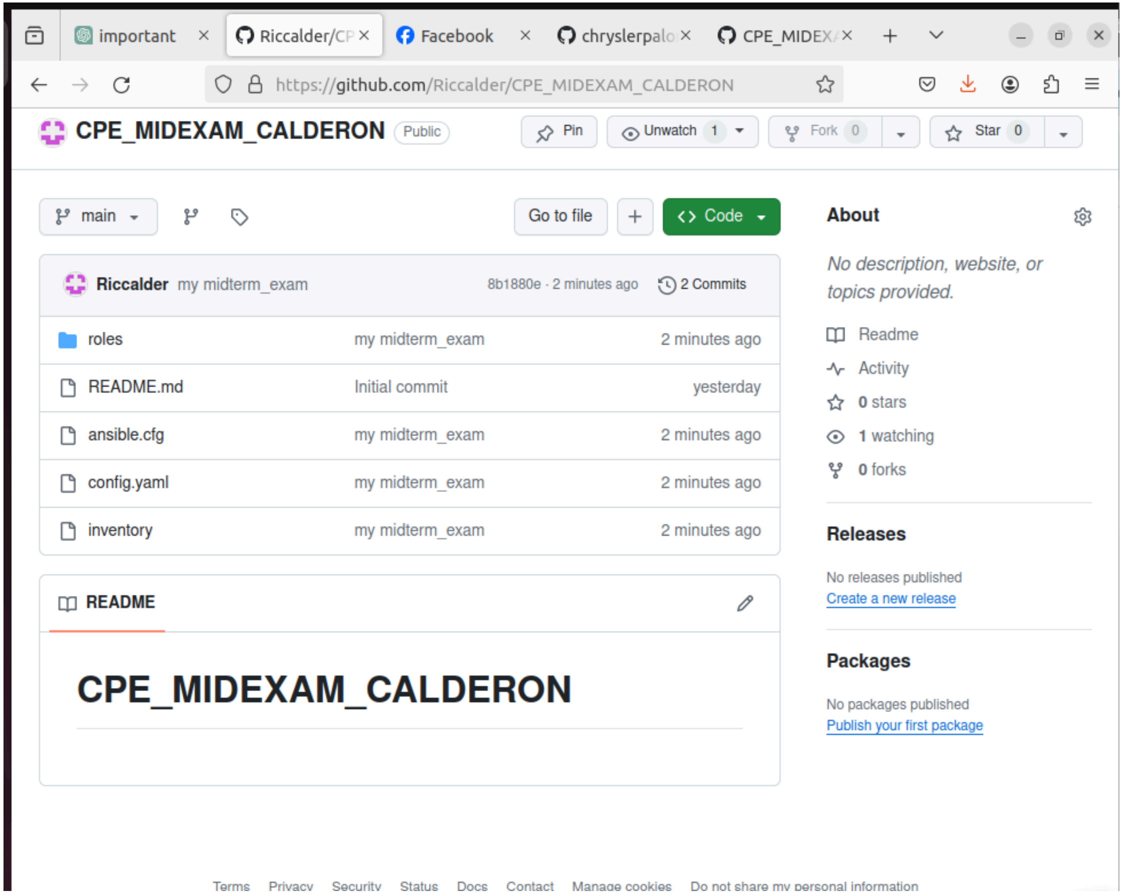


```
PLAY [all] *************************************************************************

TASK [Gathering Facts] *************************************************************
ok: [192.168.56.105]
ok: [192.168.56.103]

PLAY RECAP *************************************************************************
192.168.56.103             : ok=5    changed=2    unreachable=0    failed=0    skipped=1
   rescued=0    ignored=0
192.168.56.105             : ok=5    changed=0    unreachable=0    failed=0    skipped=1
   rescued=0    ignored=0

calderon@workstation:~/CPE_MIDEXAM_CALDERON$
```

I run an Ansible playbook named `main.yml` while specifying an inventory file (`inventory`) that contains the list of hosts to manage. Additionally, the `--ask-become-pass` option prompts the user to provide the password during playbook execution, allowing Ansible to escalate privileges when executing tasks that require elevated permissions.



```
calderon@workstation:~/CPE_MIDEXAM_CALDERON$ git commit -m "my midterm_exam"
[main 8b1880e] my midterm_exam
            hanged, 527 insertions(+)
 create mode 100644 ansible.cfg
 create mode 100644 config.yaml
 create mode 100644 inventory
 create mode 100644 roles/CentOS/tasks/ElasticStack.main.yml
 create mode 100644 roles/CentOS/tasks/HTTPD_PHP_MariaDB.main.yml
 create mode 100644 roles/CentOS/tasks/InfluxDB_Grafana_Prometheus.main.yml
 create mode 100644 roles/CentOS/tasks/InfluxDB_Grafana_Prometheus.prometheus.se
rvice
 create mode 100644 roles/Ubuntu/tasks/ElasticStack.main.yml
 create mode 100644 roles/Ubuntu/tasks/HTTPD_PHP_MariaDB.main.yml
 create mode 100644 roles/Ubuntu/tasks/InfluxDB_Grafana_Prometheus.main.yml
 create mode 100644 roles/Ubuntu/tasks/InfluxDB_Grafana_Prometheus.prometheus.se
rvice
 create mode 100644 roles/Ubuntu/tasks/nagios.main.yml
calderon@workstation:~/CPE_MIDEXAM_CALDERON$ git push origin main
Enumerating objects: 19, done.
Counting objects: 100% (19/19), done.
Delta compression using up to 2 threads
Compressing objects: 100% (16/16), done.
Writing objects: 100% (18/18), 4.38 KiB | 1.46 MiB/s, done.
Total 18 (delta 5), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (5/5), done.
To github.com:Riccalder/CPE_MIDEXAM_CALDERON.git
   554d719..8b1880e  main -> main
calderon@workstation:~/CPE_MIDEXAM_CALDERON$
```


important
Riccalder/CP
Facebook
chryslerpalo
CPE_MIDEX
https://github.com/Riccalder/CPE_MIDEXAM_CALDERON
Pin
Unwatch 1
Fork 0
Star 0
main
Go to file
Code
Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information

CPE_MIDEXAM_CALDERON Public

Riccalder my midterm_exam 8b1880e · 2 minutes ago 2 Commits

roles	my midterm_exam	2 minutes ago
README.md	Initial commit	yesterday
ansible.cfg	my midterm_exam	2 minutes ago
config.yaml	my midterm_exam	2 minutes ago
inventory	my midterm_exam	2 minutes ago

README

CPE_MIDEXAM_CALDERON

About

No description, website, or topics provided.

Readme
Activity
0 stars
1 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

I have successfully committed changes to my local git repository and pushed them to the remote repository on GitHub. By using these commands, developers can keep track of changes made to code over time, collaborate and version control code effectively, and share it with others.

GitHub link: https://github.com/Riccalder/CPE_MIDEXAM_CALDERON

Conclusions: (link your conclusion from the objective)

Our midterm exam demonstrates proficiency in using Ansible as an Infrastructure as Code (IaC) tool to install, configure, and manage various enterprise availability, performance, and log monitoring tools on separate hosts. The task involves creating an Ansible playbook that installs and configures various software applications on separate hosts, documenting all the tasks, and pushing the code to a GitHub repository. This task requires knowledge of

software installation, system configuration, and familiarity with Ansible. The completion of this task will demonstrate the ability to use Ansible to automate infrastructure management tasks and provide valuable insights into enterprise availability, performance, and log monitoring.