Name: Buduan, Christian Aaron C.	**Date Performed:** 11/06/2024
Course/Section: CPE31S2	**Date Submitted:** 11/06/2024
Instructor: Engr. Robin Vlanezuela	**Semester and SY:** 1st sem 2024-2025

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

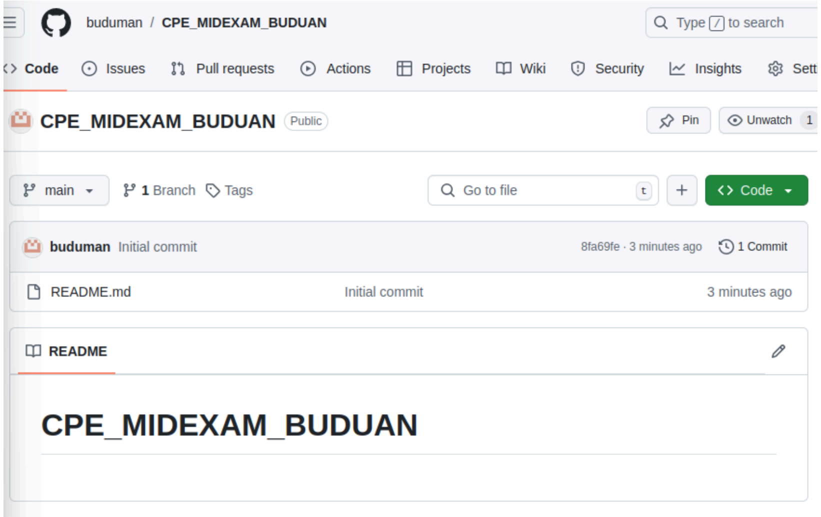
1. Objectives

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)



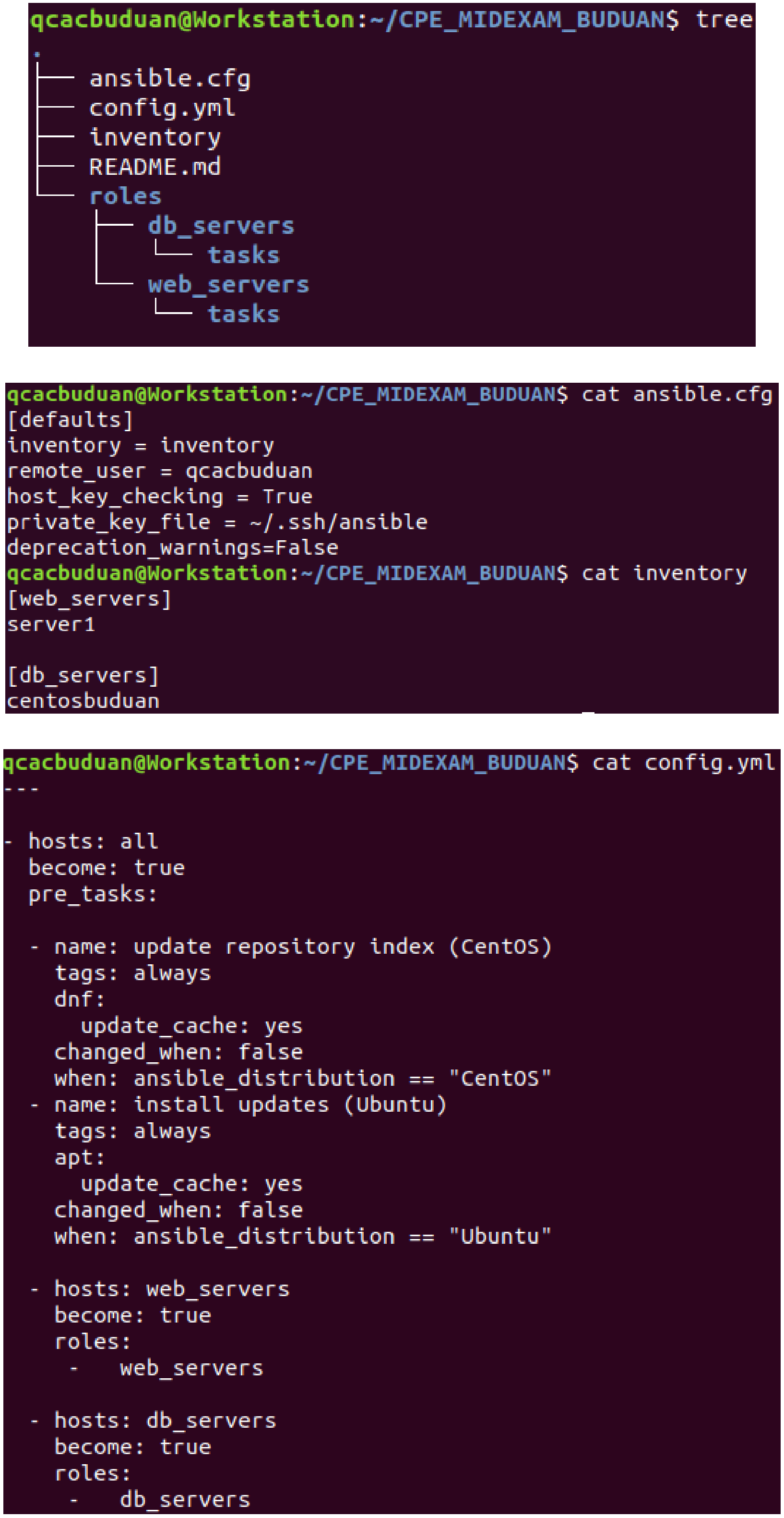
CPE_MIDEXAM_BUDUAN

buduman Initial commit 8fa69fe · 3 minutes ago 1 Commit

README.md Initial commit 3 minutes ago

CPE_MIDEXAM_BUDUAN

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



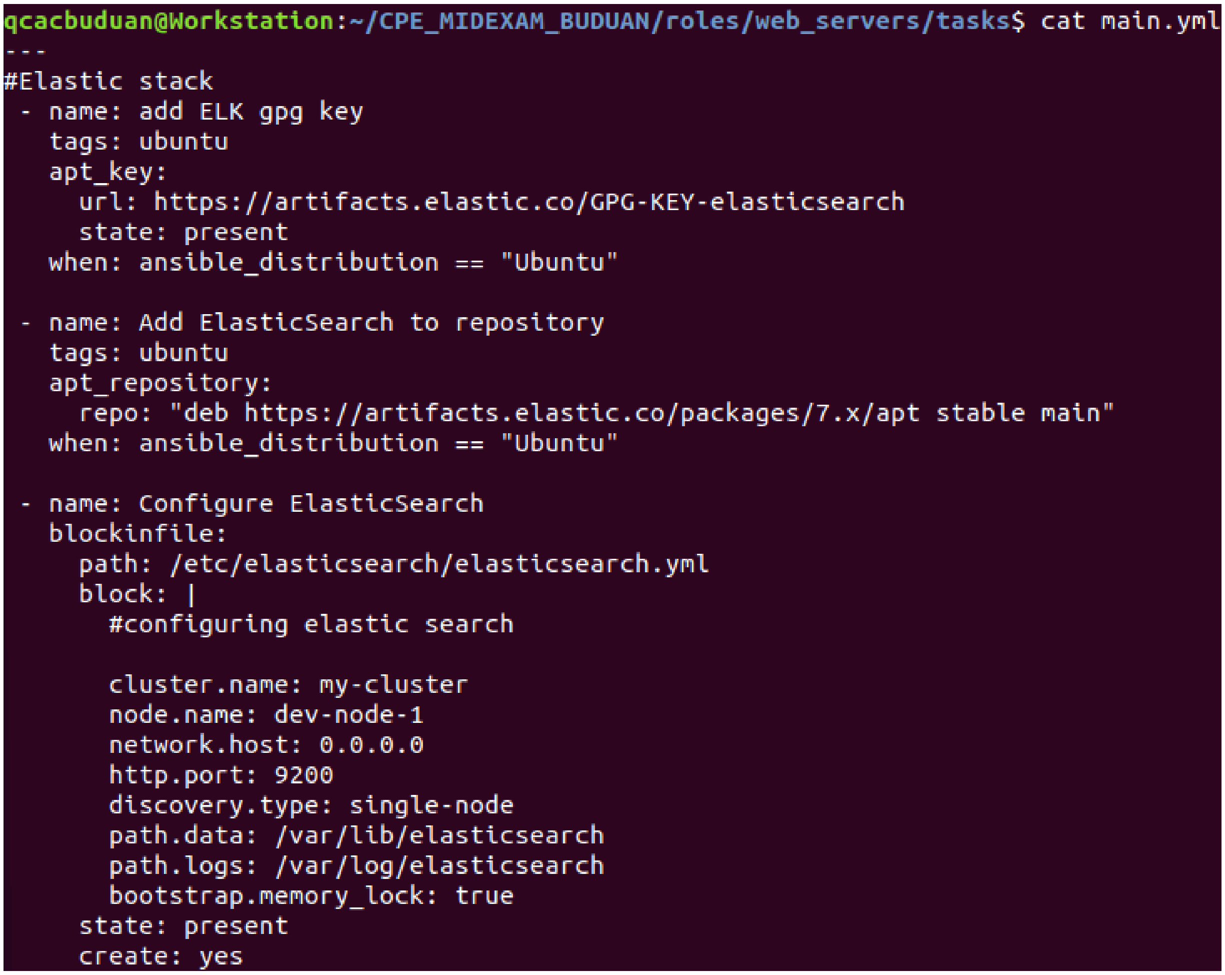
```
qcacbuduan@Workstation:~/CPE_MIDEXAM_BUDUAN$ tree
.
├── ansible.cfg
├── config.yml
├── inventory
├── README.md
└── roles
    ├── db_servers
    │   └── tasks
    └── web_servers
        └── tasks
```

```
qcacbuduan@Workstation:~/CPE_MIDEXAM_BUDUAN$ cat ansible.cfg
[defaults]
inventory = inventory
remote_user = qcacbuduan
host_key_checking = True
private_key_file = ~/.ssh/ansible
deprecation_warnings=False
qcacbuduan@Workstation:~/CPE_MIDEXAM_BUDUAN$ cat inventory
[web_servers]
server1

[db_servers]
centosbuduan
```

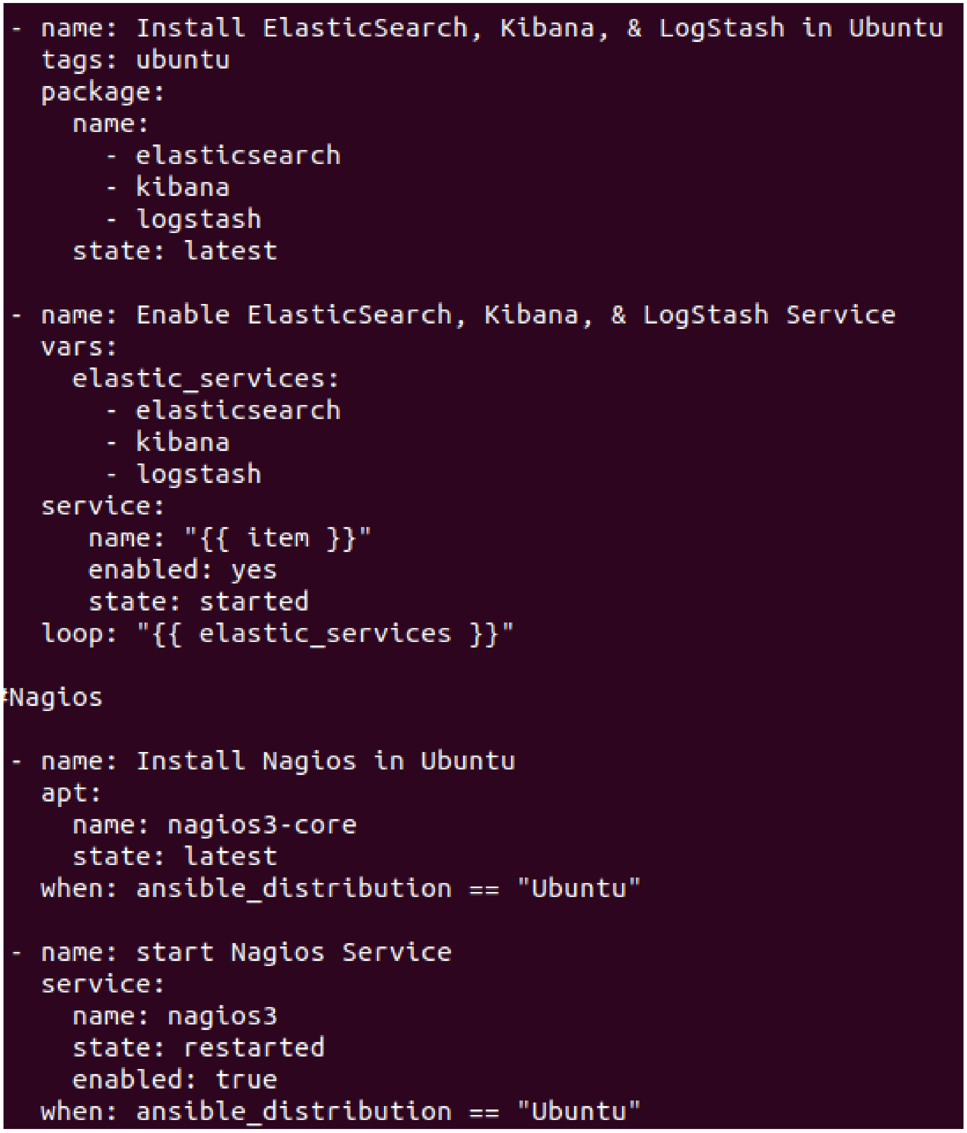
```
qcacbuduan@Workstation:~/CPE_MIDEXAM_BUDUAN$ cat config.yml
---

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

  - hosts: web_servers
    become: true
    roles:
     -   web_servers

  - hosts: db_servers
    become: true
    roles:
     -   db_servers
```

repository contents

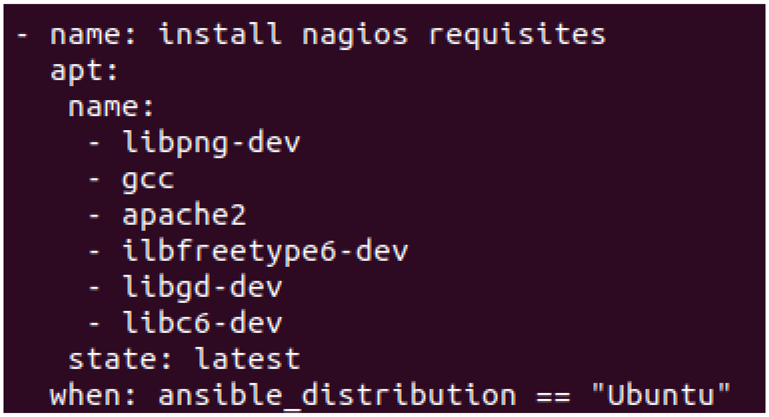
under the web_servers group which consists of one Ubuntu server, I created a playbook that installs ELK stack along with Nagios.



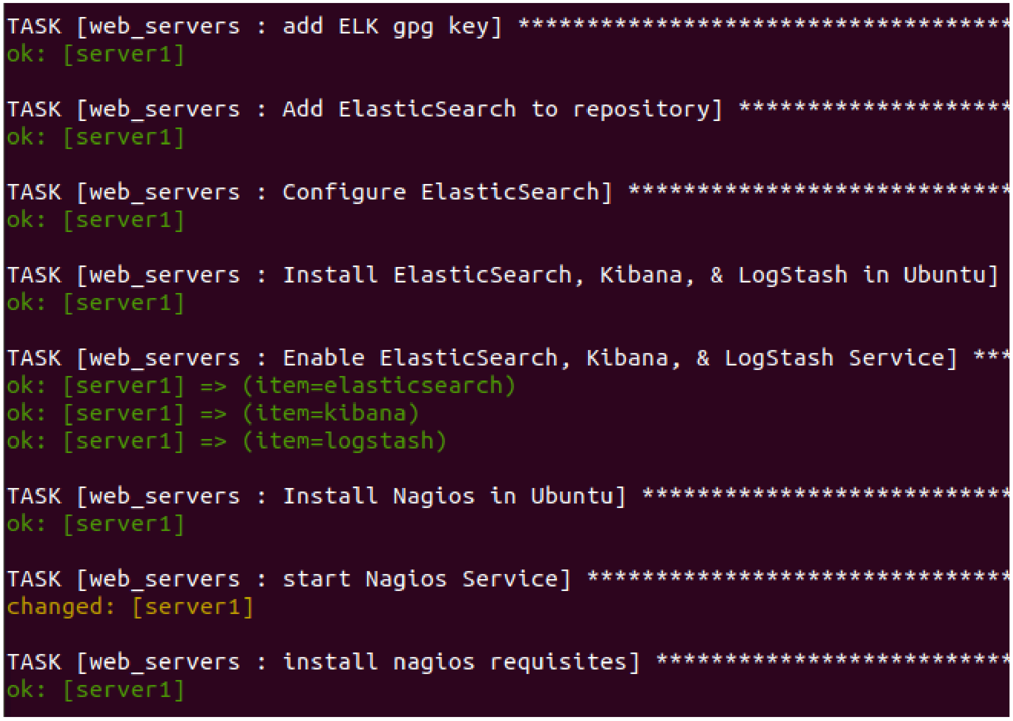
```
qcacbuduan@Workstation:~/CPE_MIDEXAM_BUDUAN/roles/web_servers/tasks$ cat main.yml
---
#Elastic stack
 - name: add ELK gpg key
   tags: ubuntu
   apt_key:
     url: https://artifacts.elastic.co/GPG-KEY-elasticsearch
     state: present
   when: ansible_distribution == "Ubuntu"

 - name: Add ElasticSearch to repository
   tags: ubuntu
   apt_repository:
     repo: "deb https://artifacts.elastic.co/packages/7.x/apt stable main"
   when: ansible_distribution == "Ubuntu"

 - name: Configure ElasticSearch
   blockinfile:
     path: /etc/elasticsearch/elasticsearch.yml
     block: |
       #configuring elastic search

       cluster.name: my-cluster
       node.name: dev-node-1
       network.host: 0.0.0.0
       http.port: 9200
       discovery.type: single-node
       path.data: /var/lib/elasticsearch
       path.logs: /var/log/elasticsearch
       bootstrap.memory_lock: true
     state: present
     create: yes
```



```
- name: Install ElasticSearch, Kibana, & LogStash in Ubuntu
  tags: ubuntu
  package:
    name:
      - elasticsearch
      - kibana
      - logstash
    state: latest

- name: Enable ElasticSearch, Kibana, & LogStash Service
  vars:
    elastic_services:
      - elasticsearch
      - kibana
      - logstash
  service:
     name: "{{ item }}"
     enabled: yes
     state: started
  loop: "{{ elastic_services }}"

#Nagios

- name: Install Nagios in Ubuntu
  apt:
    name: nagios3-core
    state: latest
  when: ansible_distribution == "Ubuntu"

- name: start Nagios Service
  service:
    name: nagios3
    state: restarted
    enabled: true
  when: ansible_distribution == "Ubuntu"
```

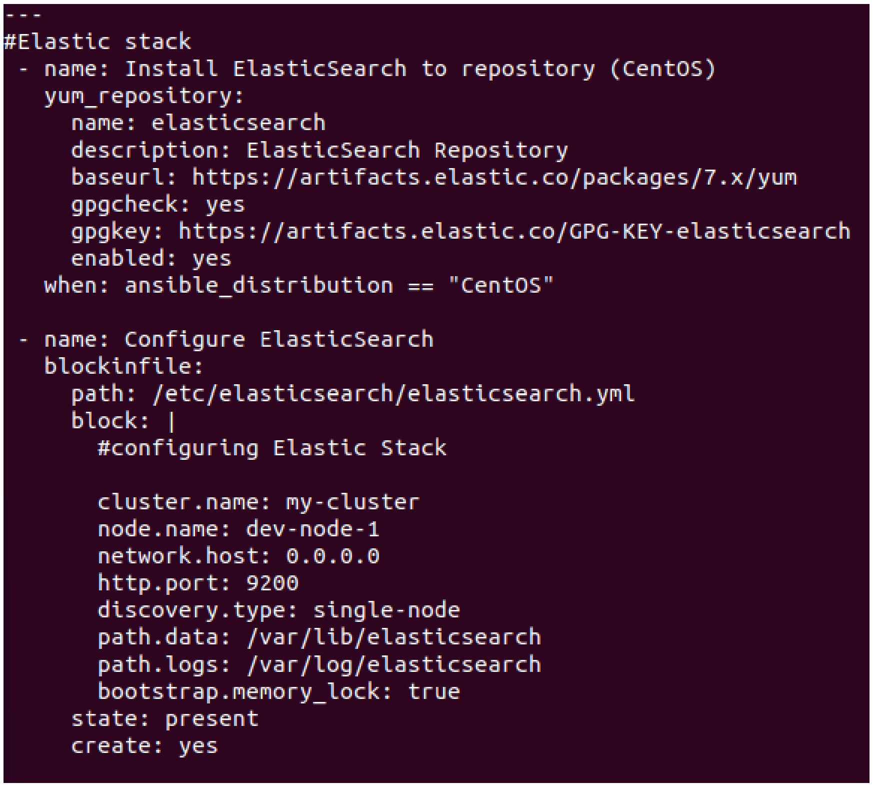


```
- name: install nagios requisites
  apt:
   name:
    - libpng-dev
    - gcc
    - apache2
    - ilbfreetype6-dev
    - libgd-dev
    - libc6-dev
   state: latest
  when: ansible_distribution == "Ubuntu"
```

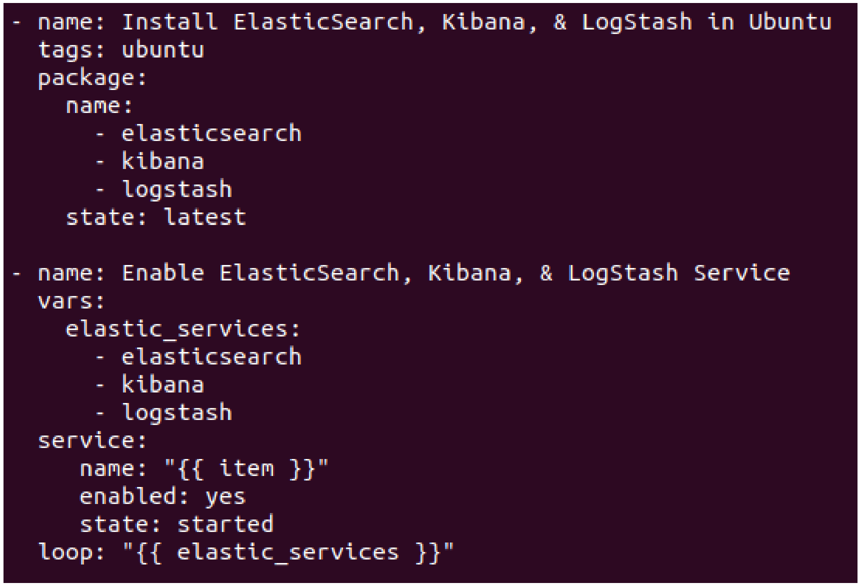


```
TASK [web_servers : add ELK gpg key] ************************************
ok: [server1]

TASK [web_servers : Add ElasticSearch to repository] ********************
ok: [server1]

TASK [web_servers : Configure ElasticSearch] ****************************
ok: [server1]

TASK [web_servers : Install ElasticSearch, Kibana, & LogStash in Ubuntu]
ok: [server1]

TASK [web_servers : Enable ElasticSearch, Kibana, & LogStash Service] ***
ok: [server1] => (item=elasticsearch)
ok: [server1] => (item=kibana)
ok: [server1] => (item=logstash)

TASK [web_servers : Install Nagios in Ubuntu] ***************************
ok: [server1]

TASK [web_servers : start Nagios Service] *******************************
changed: [server1]

TASK [web_servers : install nagios requisites] **************************
ok: [server1]
```

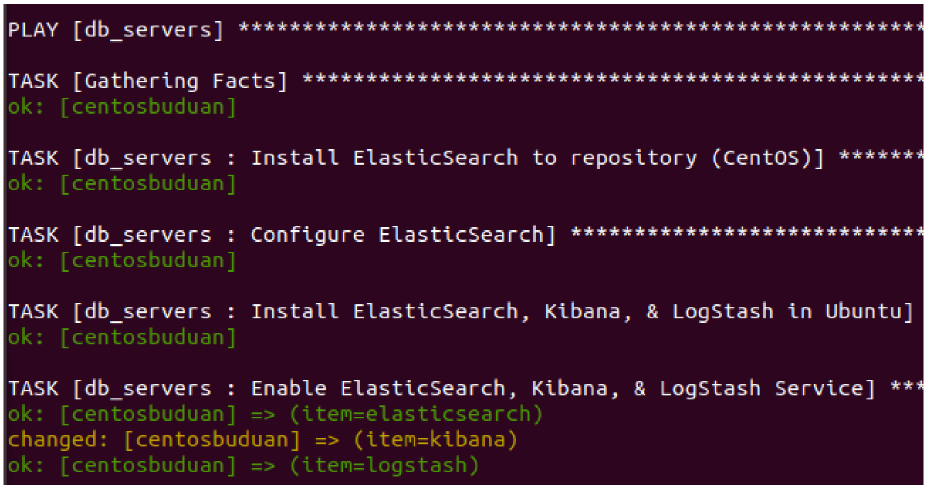
playbook to install Elastic Stack in db_server group which consists of one CentOS node.



```
---
#Elastic stack
 - name: Install ElasticSearch to 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: ansible_distribution == "CentOS"

 - name: Configure ElasticSearch
   blockinfile:
     path: /etc/elasticsearch/elasticsearch.yml
     block: |
       #configuring Elastic Stack

       cluster.name: my-cluster
       node.name: dev-node-1
       network.host: 0.0.0.0
       http.port: 9200
       discovery.type: single-node
       path.data: /var/lib/elasticsearch
       path.logs: /var/log/elasticsearch
       bootstrap.memory_lock: true
     state: present
     create: yes
```

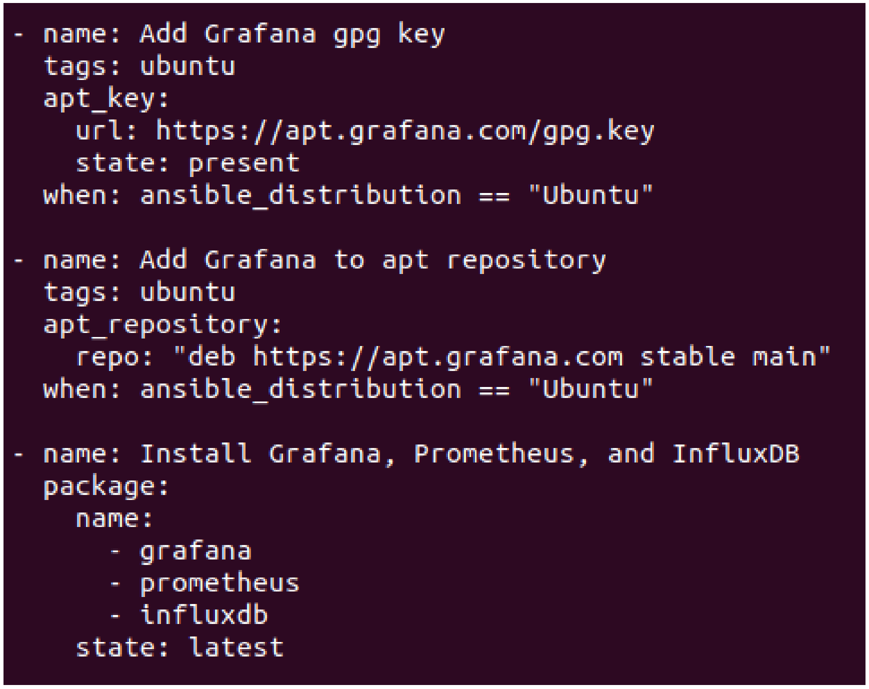


```
- name: Install ElasticSearch, Kibana, & LogStash in Ubuntu
  tags: ubuntu
  package:
    name:
      - elasticsearch
      - kibana
      - logstash
    state: latest

- name: Enable ElasticSearch, Kibana, & LogStash Service
  vars:
    elastic_services:
      - elasticsearch
      - kibana
      - logstash
  service:
     name: "{{ item }}"
     enabled: yes
     state: started
  loop: "{{ elastic_services }}"
```

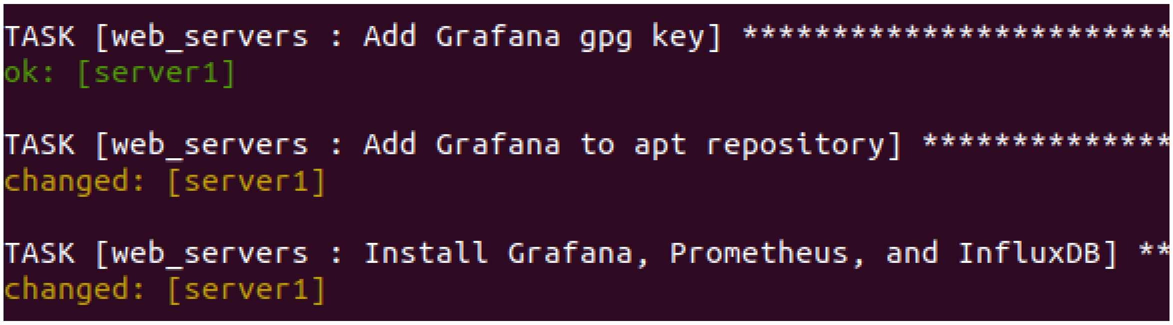


```
PLAY [db_servers] ******************************************************

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

TASK [db_servers : Install ElasticSearch to repository (CentOS)] *******
ok: [centosbuduan]

TASK [db_servers : Configure ElasticSearch] ****************************
ok: [centosbuduan]

TASK [db_servers : Install ElasticSearch, Kibana, & LogStash in Ubuntu]
ok: [centosbuduan]

TASK [db_servers : Enable ElasticSearch, Kibana, & LogStash Service] ***
ok: [centosbuduan] => (item=elasticsearch)
changed: [centosbuduan] => (item=kibana)
ok: [centosbuduan] => (item=logstash)
```

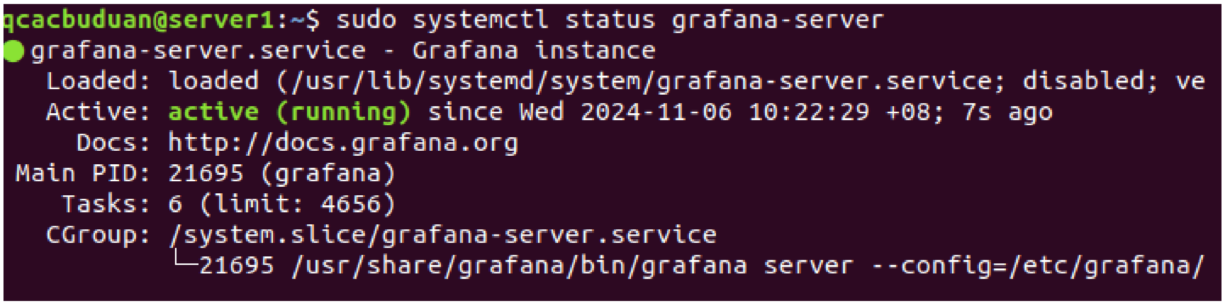
In web-servers group, I modified the playbook so that it would also Install Grafana, Prometheus, and InfluxDB.



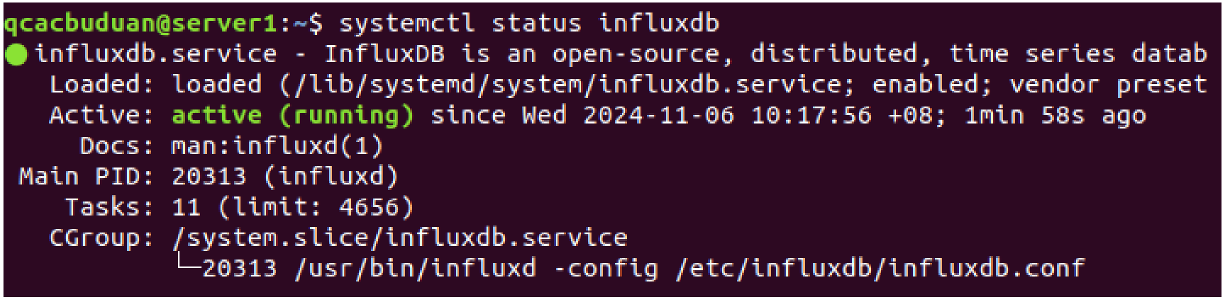
```
- name: Add Grafana gpg key
  tags: ubuntu
  apt_key:
    url: https://apt.grafana.com/gpg.key
    state: present
  when: ansible_distribution == "Ubuntu"

- name: Add Grafana to apt repository
  tags: ubuntu
  apt_repository:
    repo: "deb https://apt.grafana.com stable main"
  when: ansible_distribution == "Ubuntu"

- name: Install Grafana, Prometheus, and InfluxDB
  package:
    name:
      - grafana
      - prometheus
      - influxdb
    state: latest
```



```
TASK [web_servers : Add Grafana gpg key] *************************
ok: [server1]

TASK [web_servers : Add Grafana to apt repository] **************
changed: [server1]

TASK [web_servers : Install Grafana, Prometheus, and InfluxDB] **
changed: [server1]
```

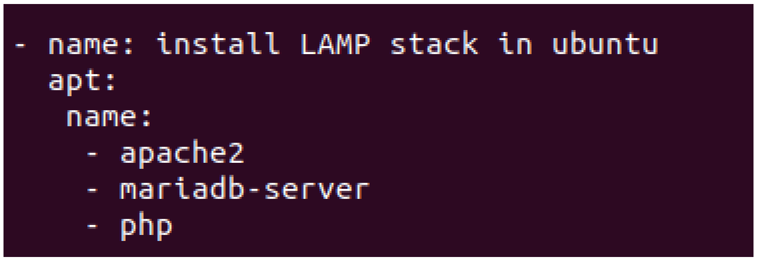


```
qcacbuduan@server1:~$ sudo systemctl status grafana-server
● grafana-server.service - Grafana instance
   Loaded: loaded (/usr/lib/systemd/system/grafana-server.service; disabled; ve
   Active: active (running) since Wed 2024-11-06 10:22:29 +08; 7s ago
     Docs: http://docs.grafana.org
 Main PID: 21695 (grafana)
    Tasks: 6 (limit: 4656)
   CGroup: /system.slice/grafana-server.service
           └─21695 /usr/share/grafana/bin/grafana server --config=/etc/grafana/
```

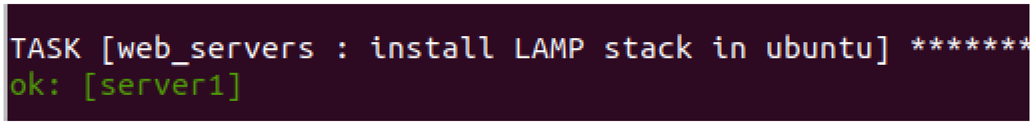


```
qcacbuduan@server1:~$ systemctl status influxdb
● influxdb.service - InfluxDB is an open-source, distributed, time series datab
   Loaded: loaded (/lib/systemd/system/influxdb.service; enabled; vendor preset
   Active: active (running) since Wed 2024-11-06 10:17:56 +08; 1min 58s ago
     Docs: man:influxd(1)
 Main PID: 20313 (influxd)
    Tasks: 11 (limit: 4656)
   CGroup: /system.slice/influxdb.service
           └─20313 /usr/bin/influxd -config /etc/influxdb/influxdb.conf
```

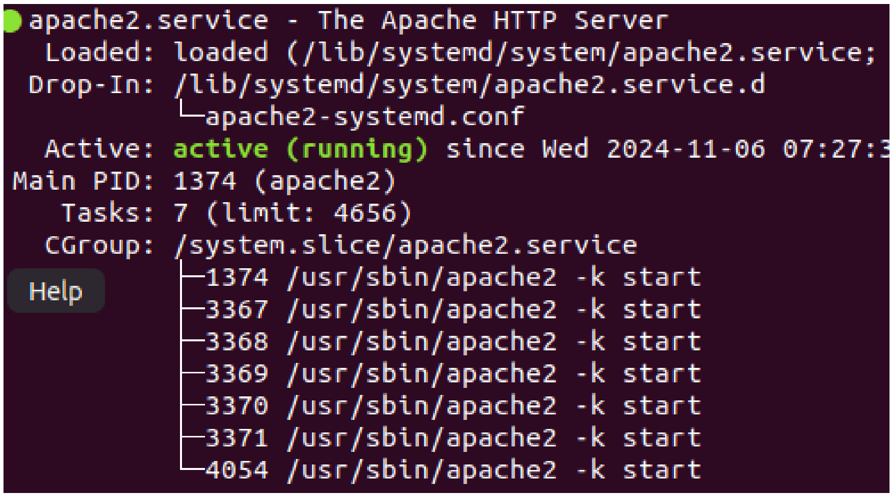
In web-servers group, I modified the playbook so that it would also Install Lamp Stack (Httpd + Php,Mariadb)



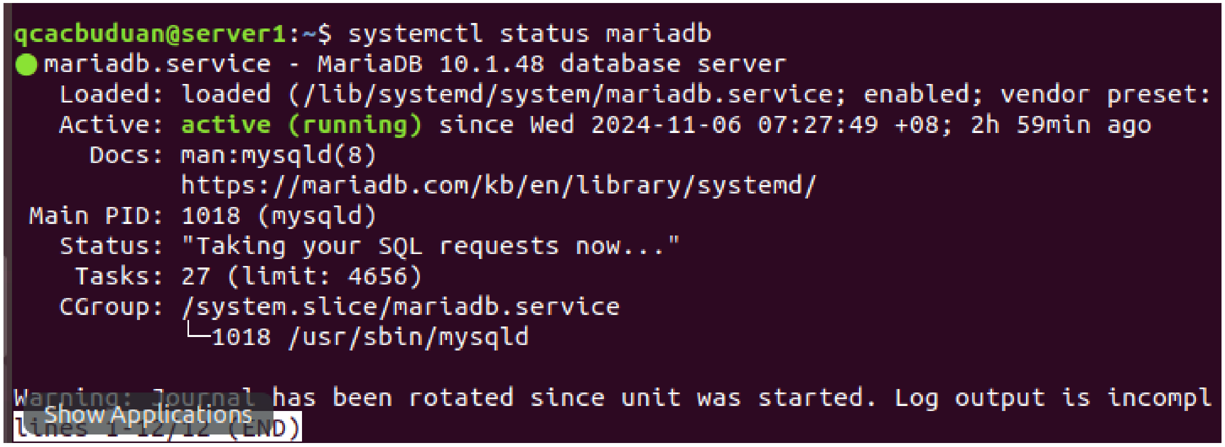
```
- name: install LAMP stack in ubuntu
  apt:
   name:
    - apache2
    - mariadb-server
    - php
```



```
TASK [web_servers : install LAMP stack in ubuntu] *******
ok: [server1]
```



```
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service;
  Drop-In: /lib/systemd/system/apache2.service.d
           └─apache2-systemd.conf
   Active: active (running) since Wed 2024-11-06 07:27:3
 Main PID: 1374 (apache2)
    Tasks: 7 (limit: 4656)
   CGroup: /system.slice/apache2.service
           ├─1374 /usr/sbin/apache2 -k start
           ├─3367 /usr/sbin/apache2 -k start
           ├─3368 /usr/sbin/apache2 -k start
           ├─3369 /usr/sbin/apache2 -k start
           ├─3370 /usr/sbin/apache2 -k start
           ├─3371 /usr/sbin/apache2 -k start
           └─4054 /usr/sbin/apache2 -k start
```



```
qcacbuduan@server1:~$ systemctl status mariadb
● mariadb.service - MariaDB 10.1.48 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset:
   Active: active (running) since Wed 2024-11-06 07:27:49 +08; 2h 59min ago
     Docs: man:mysqld(8)
           https://mariadb.com/kb/en/library/systemd/
 Main PID: 1018 (mysqld)
   Status: "Taking your SQL requests now..."
    Tasks: 27 (limit: 4656)
   CGroup: /system.slice/mariadb.service
           └─1018 /usr/sbin/mysqld

Warning: Journal has been rotated since unit was started. Log output is incompl
lines 1-12/12 (END)
```

GitHub link:

https://github.com/buduman/CPE_MIDEXAM_BUDUAN

Conclusions: (link your conclusion from the objective)

In this practical Exam, I was able to apply everything that I have learned from the past hands-on activities that were given, such as applying the concept of roles so that I can create and organize my playbooks based on the server that I would like to work on. I also learned throughout the activities on how to Install different types of programs which help us manga our servers more. Some of these programs require more