Name: Maxine Audrey D. Pulao	**Date Performed: October 24, 2022**
Course/Section: CPE31S2	**Date Submitted: October 24, 2022**
Instructor: Dr. Jonathan Taylar	**Semester and SY: 2022-2023**

Activity 10: Install, Configure, and Manage Log Monitoring tools

1. Objectives

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

2. Discussion

Log monitoring software scans and monitors log files generated by servers, applications, and networks. By detecting and alerting users to patterns in these log files, log monitoring software helps solve performance and security issues. System administrators use log monitoring software to detect common important events indicated by log files.

Log monitoring software helps maintain IT infrastructure performance and pinpoints issues to prevent downtime and mitigate risks. These tools will often integrate with IT alerting software, log analysis software, and other IT issue resolution products to more aptly flesh out the IT infrastructure maintenance ecosystem.

To qualify for inclusion in the Log Monitoring category, a product must:

- Monitor the log files generated by servers, applications, or networks
- Alert users when important events are detected
- Provide reporting capabilities for log files

Elastic Stack

ELK suite stands for Elasticsearch, Kibana, Beats, and Logstash (also known as the ELK Stack). Source: https://www.elastic.co/elastic-stack

The Elastic Stack is a group of open source products from Elastic designed to help users take data from any type of source and in any format, and search, analyze and visualize that data in real time. The product group was formerly known as the ELK Stack for the core products in the group -- Elasticsearch, Logstash and Kibana -- but has been rebranded as the Elastic Stack. A fourth product, Beats, was subsequently added to the stack. The Elastic Stack can be deployed on premises or made available as software as a service (SaaS). Elasticsearch supports Amazon Web Services (AWS), Google Cloud Platform and Microsoft Azure.

GrayLog

Graylog is a powerful platform that allows for easy log management of both structured and unstructured data along with debugging applications.

It is based on Elasticsearch, MongoDB, and Scala. Graylog has a main server, which receives data from its clients installed on different servers, and a web interface, which visualizes the data and allows to work with logs aggregated by the main server.

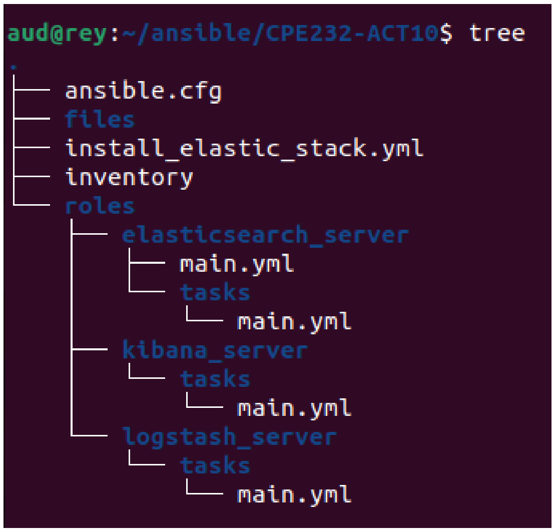
We use Graylog primarily as the stash for the logs of the web applications we build. However, it is also effective when working with raw strings (i.e. syslog): the tool parses it into the structured data we need. It also allows advanced custom search in the logs using structured queries. In other words, when integrated properly with a web app, Graylog helps engineers to analyze the system behavior on almost per code line basis.

Source: https://www.graylog.org/products/open-source

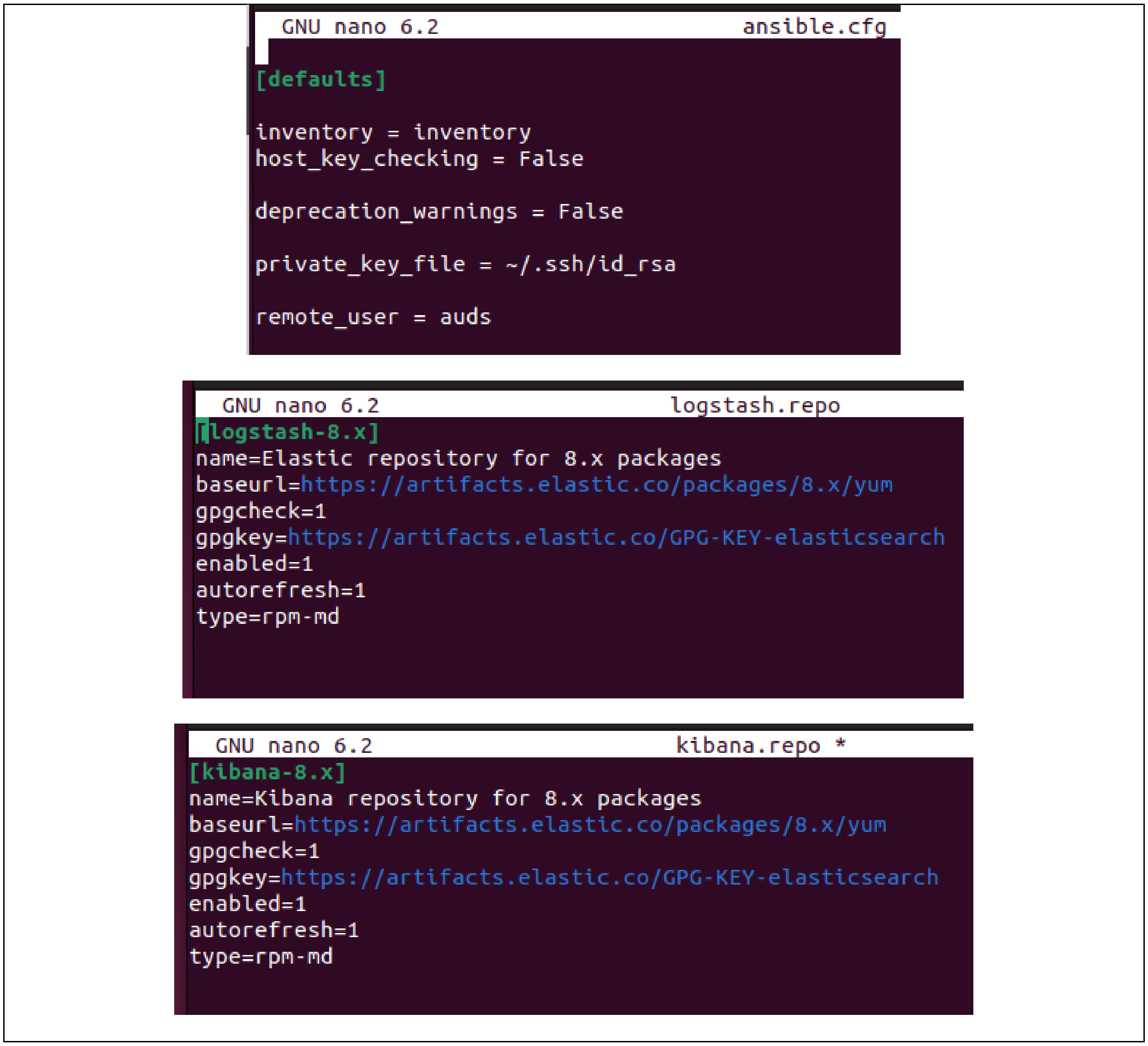
3. Tasks

1. Create a playbook that:
   a. Install and configure Elastic Stack in separate hosts (Elastic Search, Kibana, Logstash)
2. Apply the concept of creating roles.
3. Describe how you did step 1. (Provide screenshots and explanations in your report. Make your report detailed such that it will look like a manual.)
4. Show an output of the installed Elastic Stack for both Ubuntu and CentOS.
5. Make sure to create a new repository in GitHub for this activity.

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



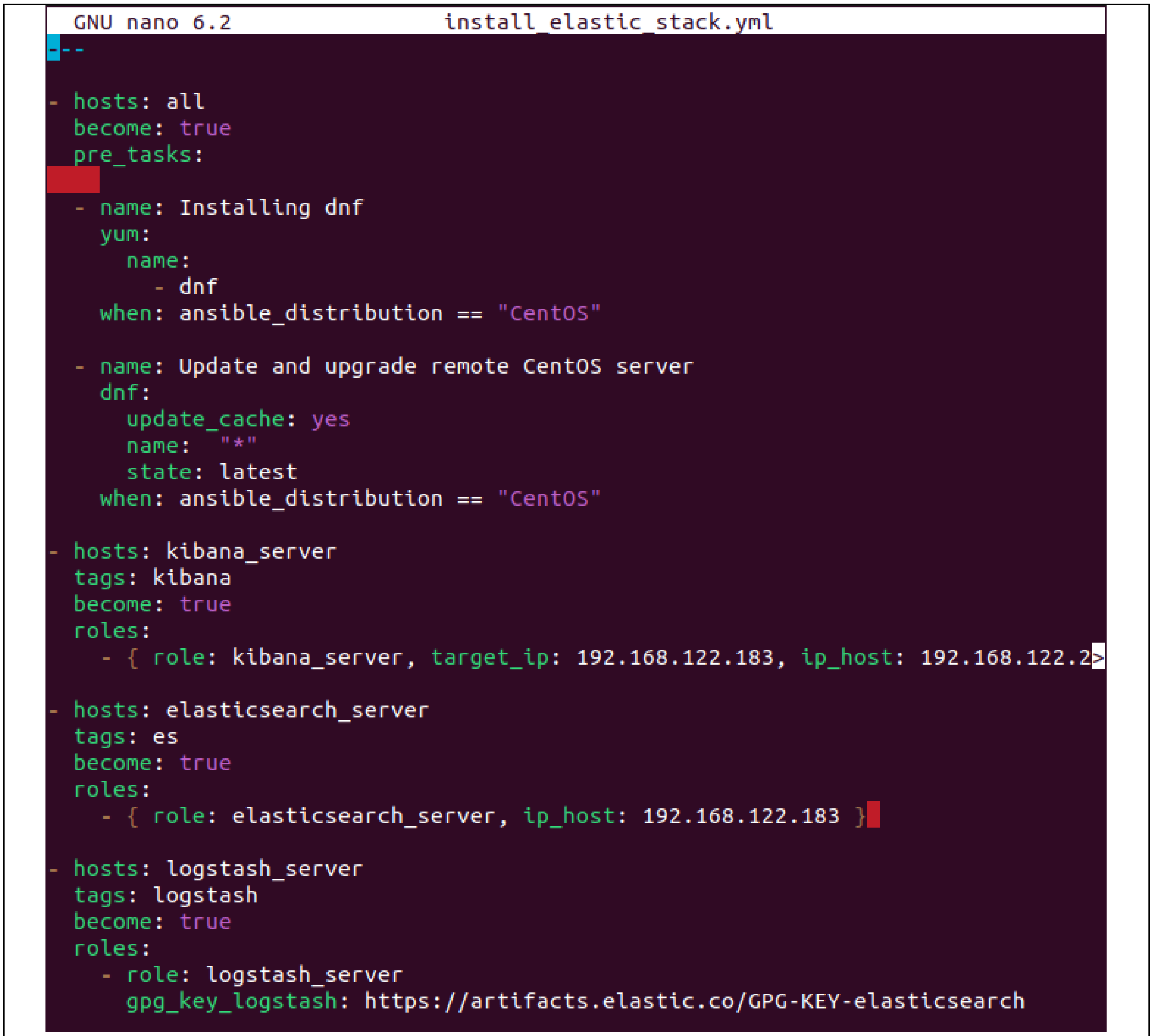
```
aud@rey:~/ansible/CPE232-ACT10$ tree
.
├── ansible.cfg
├── files
├── install_elastic_stack.yml
├── inventory
└── roles
    ├── elasticsearch_server
    │   ├── main.yml
    │   └── tasks
    │       └── main.yml
    ├── kibana_server
    │   └── tasks
    │       └── main.yml
    └── logstash_server
        └── tasks
            └── main.yml
```



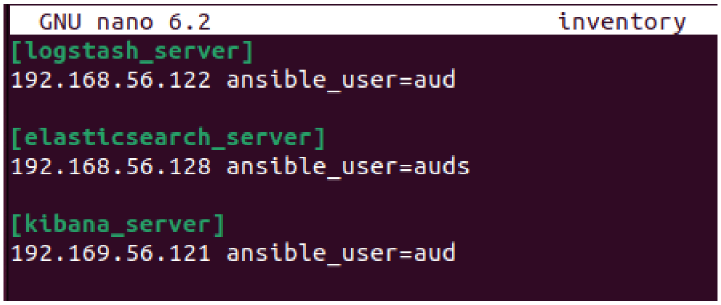
```
  GNU nano 6.2                          ansible.cfg

[defaults]

inventory = inventory
host_key_checking = False

deprecation_warnings = False

private_key_file = ~/.ssh/id_rsa

remote_user = auds
```

```
  GNU nano 6.2                          logstash.repo
[logstash-8.x]
name=Elastic repository for 8.x packages
baseurl=https://artifacts.elastic.co/packages/8.x/yum
gpgcheck=1
gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch
enabled=1
autorefresh=1
type=rpm-md
```

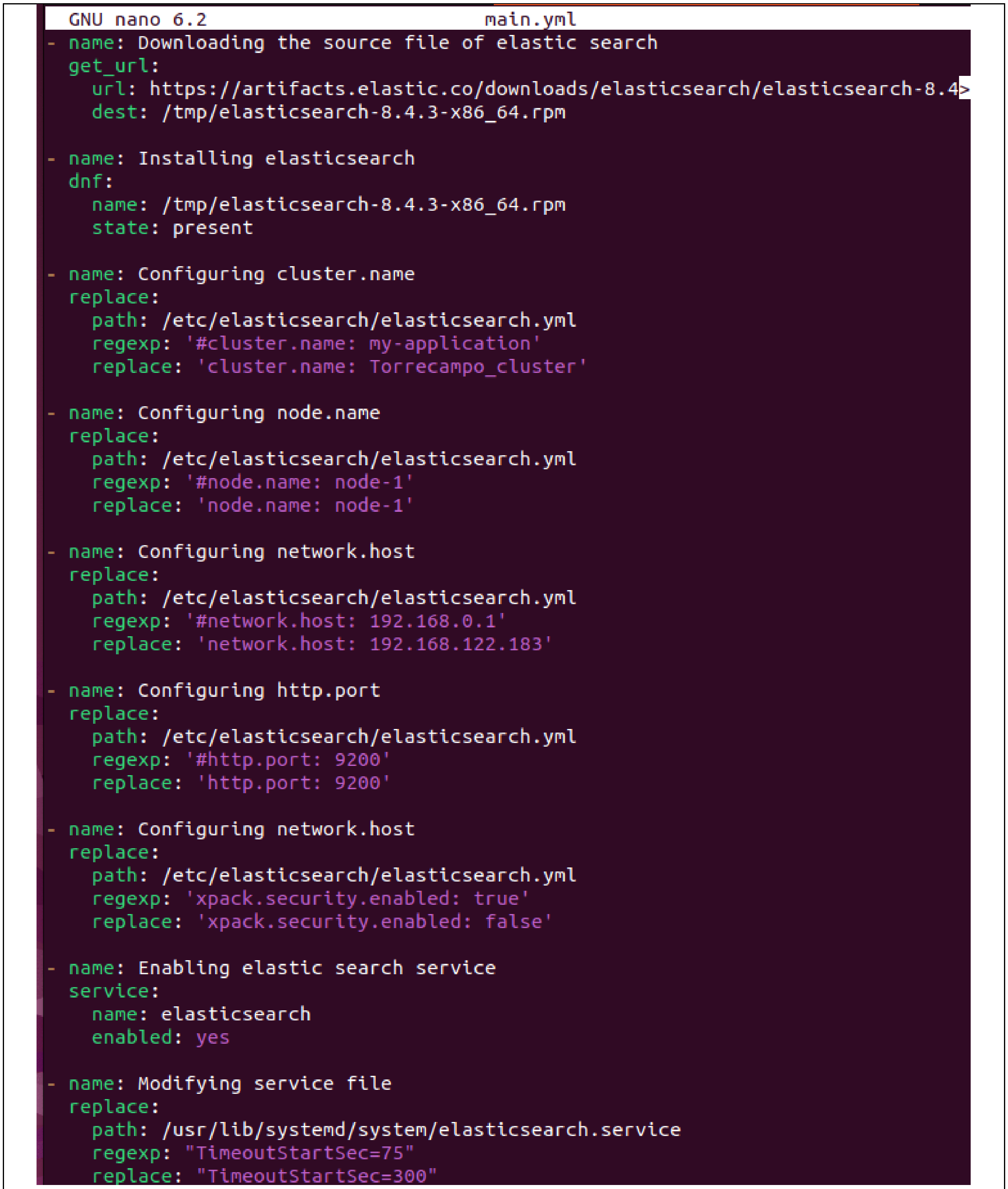
```
  GNU nano 6.2                          kibana.repo *
[kibana-8.x]
name=Kibana repository for 8.x packages
baseurl=https://artifacts.elastic.co/packages/8.x/yum
gpgcheck=1
gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch
enabled=1
autorefresh=1
type=rpm-md
```



```
  GNU nano 6.2                    install_elastic_stack.yml
---

- hosts: all
  become: true
  pre_tasks:

  - name: Installing dnf
    yum:
      name:
        - dnf
    when: ansible_distribution == "CentOS"

  - name: Update and upgrade remote CentOS server
    dnf:
      update_cache: yes
      name:  "*"
      state: latest
    when: ansible_distribution == "CentOS"

- hosts: kibana_server
  tags: kibana
  become: true
  roles:
    - { role: kibana_server, target_ip: 192.168.122.183, ip_host: 192.168.122.2>

- hosts: elasticsearch_server
  tags: es
  become: true
  roles:
    - { role: elasticsearch_server, ip_host: 192.168.122.183 }

- hosts: logstash_server
  tags: logstash
  become: true
  roles:
    - role: logstash_server
      gpg_key_logstash: https://artifacts.elastic.co/GPG-KEY-elasticsearch
```

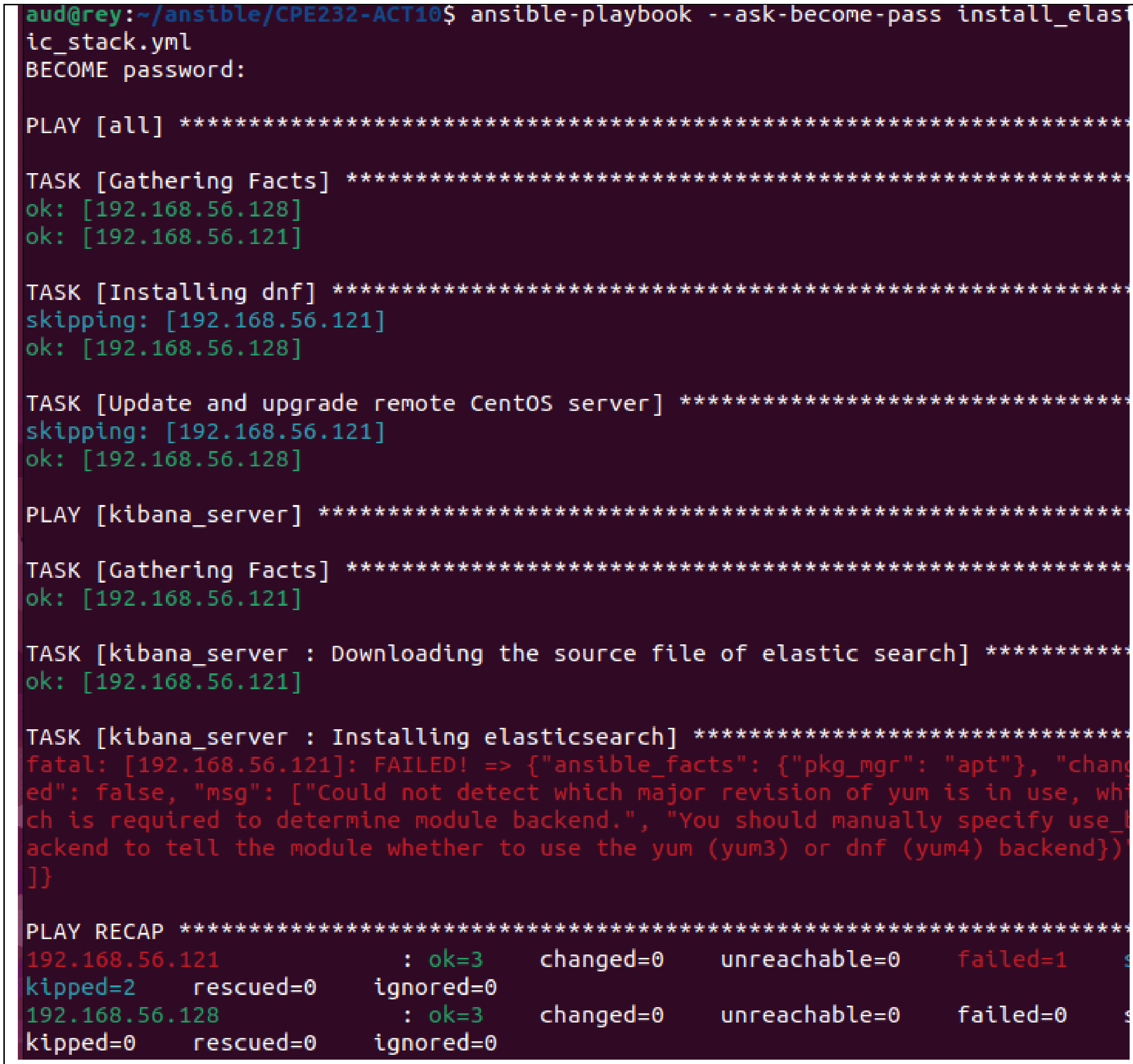


```
  GNU nano 6.2                          inventory
[logstash_server]
192.168.56.122 ansible_user=aud

[elasticsearch_server]
192.168.56.128 ansible_user=auds

[kibana_server]
192.169.56.121 ansible_user=aud
```



```
  GNU nano 6.2                              main.yml
- name: Downloading the source file of elastic search
  get_url:
    url: https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-8.4>
    dest: /tmp/elasticsearch-8.4.3-x86_64.rpm

- name: Installing elasticsearch
  dnf:
    name: /tmp/elasticsearch-8.4.3-x86_64.rpm
    state: present

- name: Configuring cluster.name
  replace:
    path: /etc/elasticsearch/elasticsearch.yml
    regexp: '#cluster.name: my-application'
    replace: 'cluster.name: Torrecampo_cluster'

- name: Configuring node.name
  replace:
    path: /etc/elasticsearch/elasticsearch.yml
    regexp: '#node.name: node-1'
    replace: 'node.name: node-1'

- name: Configuring network.host
  replace:
    path: /etc/elasticsearch/elasticsearch.yml
    regexp: '#network.host: 192.168.0.1'
    replace: 'network.host: 192.168.122.183'

- name: Configuring http.port
  replace:
    path: /etc/elasticsearch/elasticsearch.yml
    regexp: '#http.port: 9200'
    replace: 'http.port: 9200'

- name: Configuring network.host
  replace:
    path: /etc/elasticsearch/elasticsearch.yml
    regexp: 'xpack.security.enabled: true'
    replace: 'xpack.security.enabled: false'

- name: Enabling elastic search service
  service:
    name: elasticsearch
    enabled: yes

- name: Modifying service file
  replace:
    path: /usr/lib/systemd/system/elasticsearch.service
    regexp: "TimeoutStartSec=75"
    replace: "TimeoutStartSec=300"
```

- These are the needed ansible directories and files in order to install the elastic search.



```
aud@rey:~/ansible/CPE232-ACT10$ ansible-playbook --ask-become-pass install_elast
ic_stack.yml
BECOME password:

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

TASK [Gathering Facts] ******************************************************
ok: [192.168.56.128]
ok: [192.168.56.121]

TASK [Installing dnf] *******************************************************
skipping: [192.168.56.121]
ok: [192.168.56.128]

TASK [Update and upgrade remote CentOS server] ******************************
skipping: [192.168.56.121]
ok: [192.168.56.128]

PLAY [kibana_server] ********************************************************

TASK [Gathering Facts] ******************************************************
ok: [192.168.56.121]

TASK [kibana_server : Downloading the source file of elastic search] ********
ok: [192.168.56.121]

TASK [kibana_server : Installing elasticsearch] *****************************
fatal: [192.168.56.121]: FAILED! => {"ansible_facts": {"pkg_mgr": "apt"}, "chang
ed": false, "msg": ["Could not detect which major revision of yum is in use, whi
ch is required to determine module backend.", "You should manually specify use_b
ackend to tell the module whether to use the yum (yum3) or dnf (yum4) backend})"
]}

PLAY RECAP ******************************************************************
192.168.56.121             : ok=3    changed=0    unreachable=0    failed=1    s
kipped=2    rescued=0    ignored=0
192.168.56.128             : ok=3    changed=0    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
```

- Output of install_elastic_stack.yml

Reflections:

Answer the following:

1. What are the benefits of having log monitoring tool?

   For system administrators, having a performance monitoring tool is a lifesaver. System administrators can visit it to check on the servers' present condition. This keeps track of the system's processes, hardware resources, faults, and much more. The Prometheus is one illustration that demonstrates this important feature. While Prometheus is capable of locating the cause of failures, alerting the system administrators, and continuously monitoring services for each individual device, it is not in charge of correcting flaws or errors. Although not mentioned in the activity, Cacti is also thought of as a monitoring tool and has the ability to produce performance management graphs.

Conclusions:

- I may not have successfully installed the log monitoring tool but I however learned how ansible playbook works and managed to troubleshoot somg of its errors and bugs. With that, I will examine this activity again and make this as a lesson to learn more on my mistakes in designing a workflow that installs, configure and manage enterprise log monitoring tools using Ansible as an Infrastructure as Code. With that, I can effectively apply what I have learned in this activity and further help with my future as a system administrator.