Name: Jozshua Amiel Alonzo	**Date Performed: October 26, 2022**
Course/Section: CPE31S23	**Date Submitted:**
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

Github Link: https://github.com/jozshua/HOA10_Alonzo.git

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.

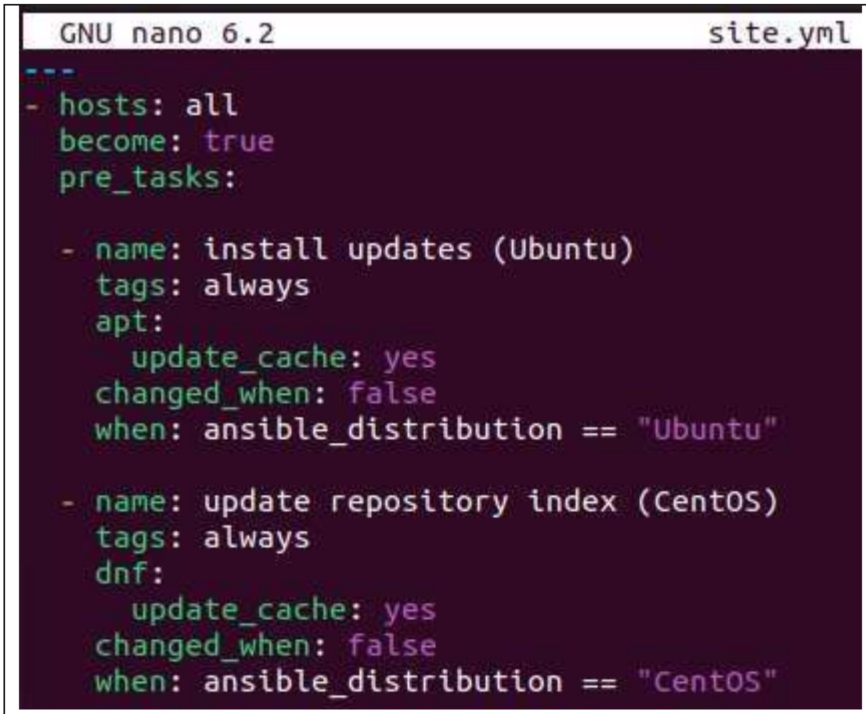
We use Graylog primarily as the stash for the logs of the web applications we build. However, it is also effective when working with raw strings (i.e. syslog): the tool parses it into the structured data we need. It also allows advanced custom search in the logs using structured queries. In other words, 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

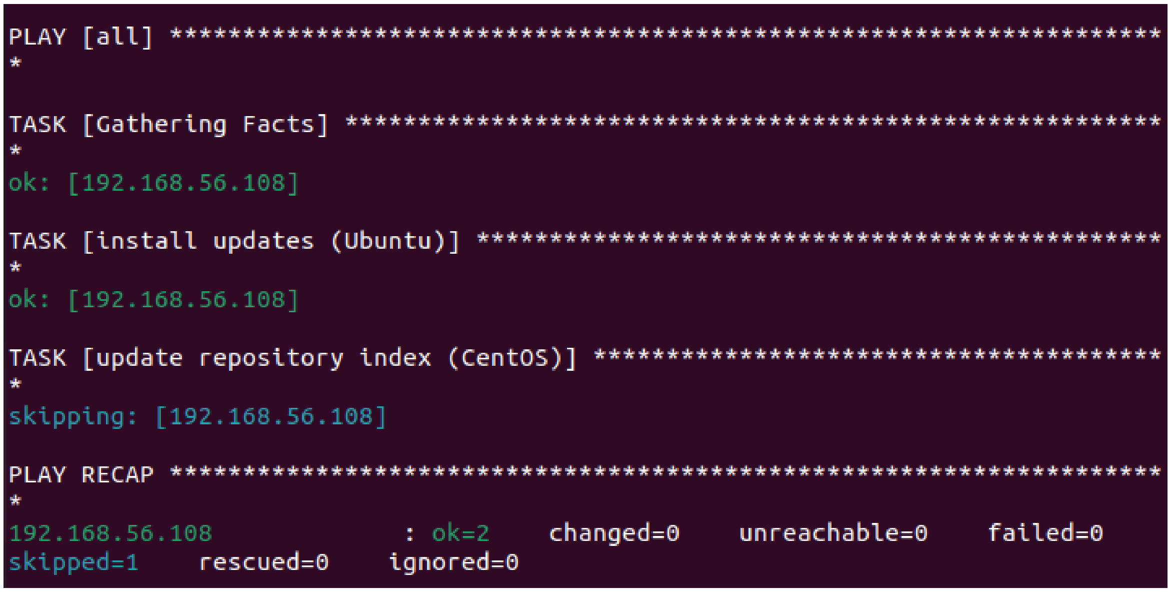
1. Create a playbook that:
   a. Install and configure Elastic Stack in separate hosts (Elastic Search, Kibana, Logstash)
2. Apply the concept of creating roles.
3. Describe how you did step 1. (Provide screenshots and explanations in your report. Make your report detailed such that it will look like 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)

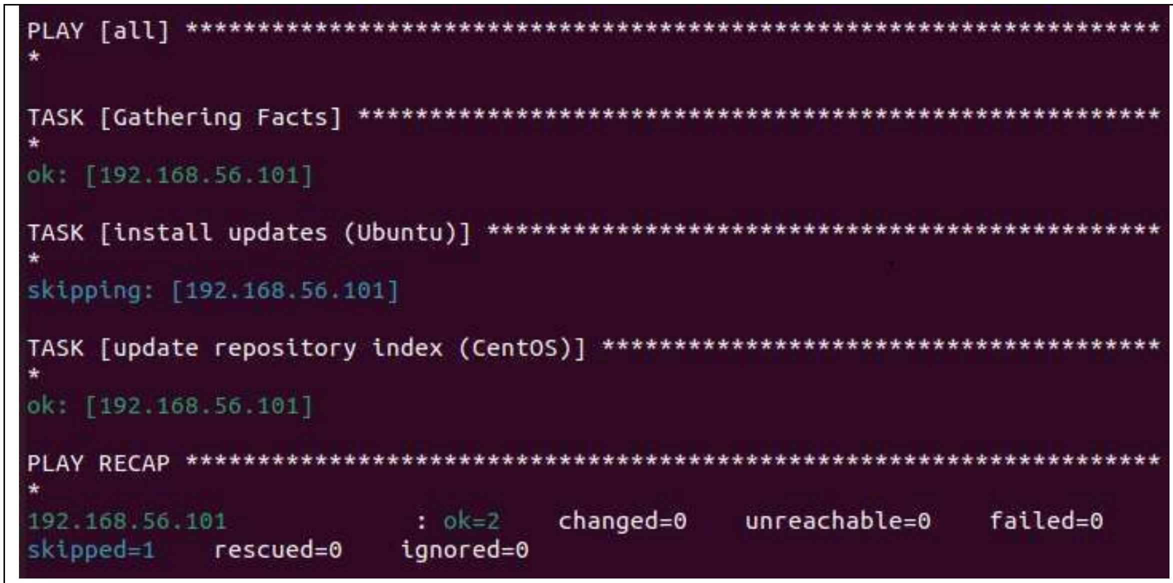


```
  GNU nano 6.2                         site.yml
---
- hosts: all
  become: true
  pre_tasks:

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

  - name: update repository index (CentOS)
    tags: always
    dnf:
      update_cache: yes
    changed_when: false
    when: ansible_distribution == "CentOS"
```

Create the site.yml file and type these codes inside, these codes are for the two server distributions. Run the playbook.



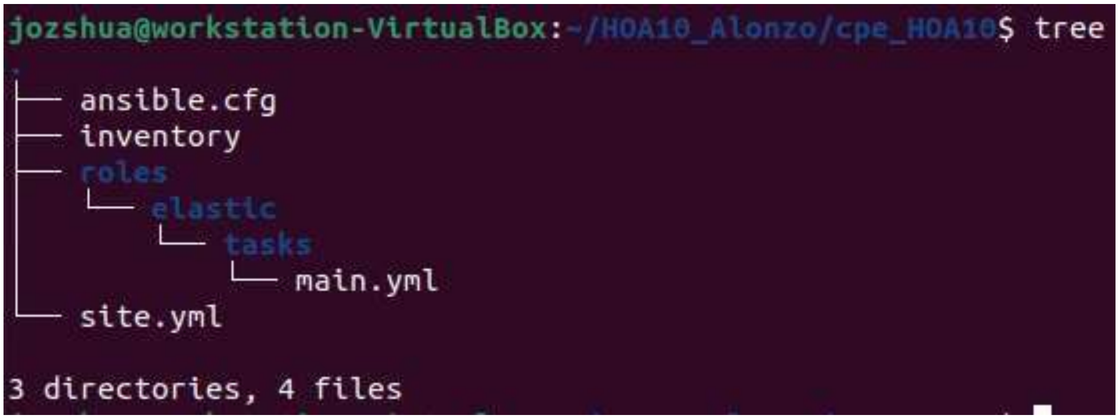
```
PLAY [all] ********************************************************************
*

TASK [Gathering Facts] ********************************************************
*
ok: [192.168.56.108]

TASK [install updates (Ubuntu)] ***********************************************
*
ok: [192.168.56.108]

TASK [update repository index (CentOS)] ***************************************
*
skipping: [192.168.56.108]

PLAY RECAP ********************************************************************
*
192.168.56.108             : ok=2    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
```



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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.101]

TASK [install updates (Ubuntu)] ************************************************
*
skipping: [192.168.56.101]

TASK [update repository index (CentOS)] ****************************************
*
ok: [192.168.56.101]

PLAY RECAP *********************************************************************
*
192.168.56.101             : ok=2    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
```

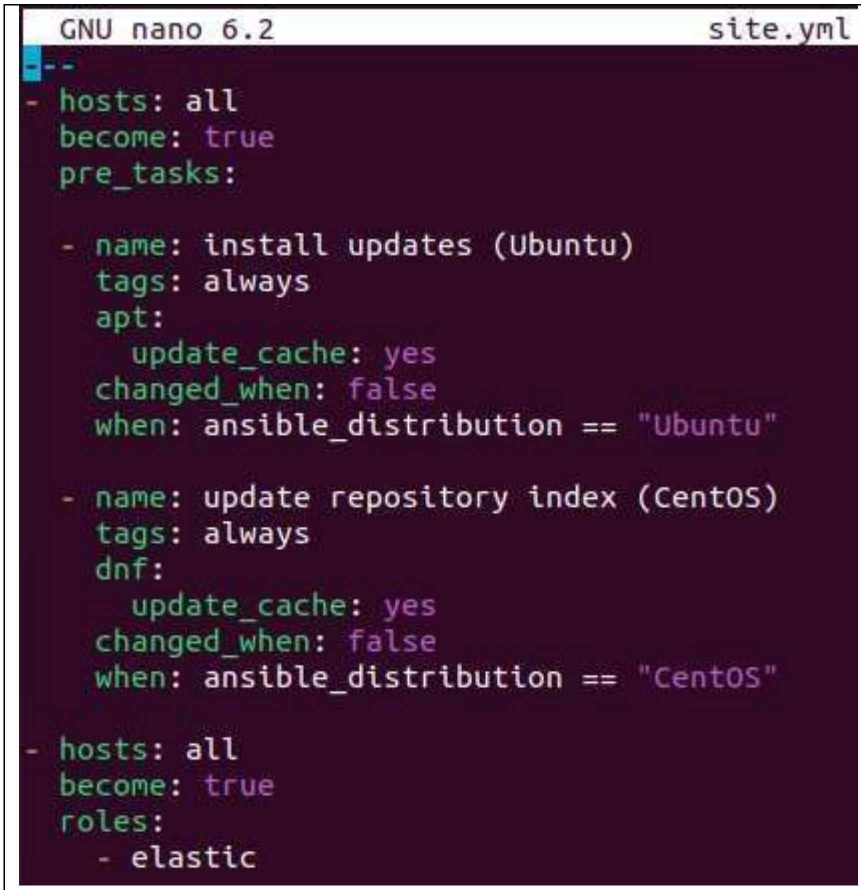
From running this playbook the tasks were successfully applied on each server distribution. There are 2 skipped states in each server because of having a different distribution from the ansible command.

Applying the concept of creating roles:



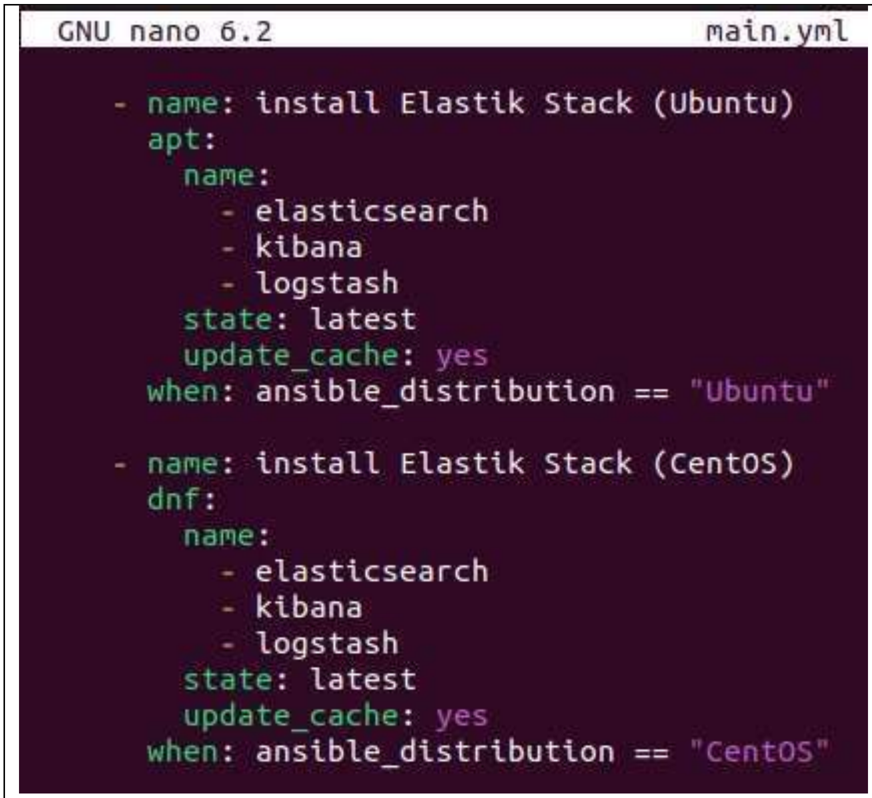
```
jozshua@workstation-VirtualBox:~/HOA10_Alonzo/cpe_HOA10$ tree
.
├── ansible.cfg
├── inventory
├── roles
│   └── elastic
│       └── tasks
│           └── main.yml
└── site.yml

3 directories, 4 files
```

Create the roles, elastic, and tasks directories for creating the main.yml file. Issue the command tree to display the route.



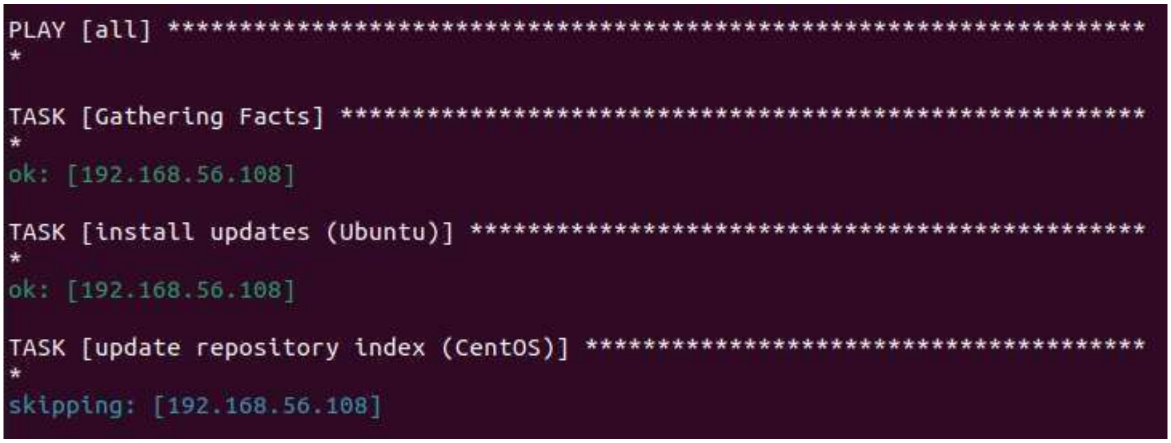
```
  GNU nano 6.2                          site.yml
---
- hosts: all
  become: true
  pre_tasks:

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

  - name: update repository index (CentOS)
    tags: always
    dnf:
      update_cache: yes
    changed_when: false
    when: ansible_distribution == "CentOS"

- hosts: all
  become: true
  roles:
    - elastic
```

Edit the site.yml file and insert these codes below from applying the main.yml file later.

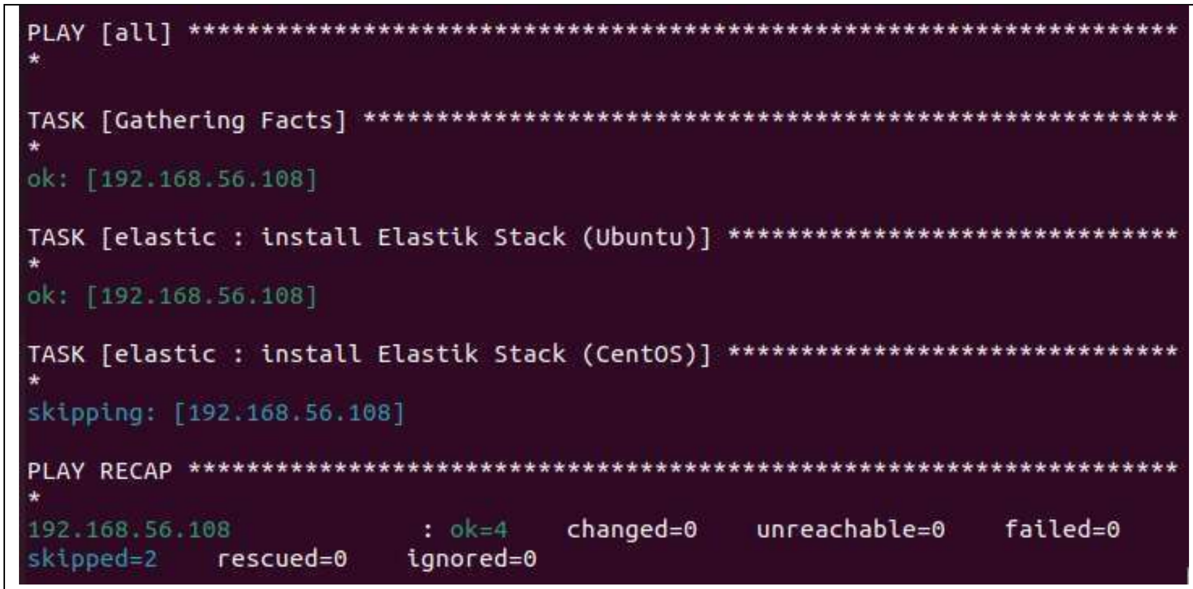


```
  GNU nano 6.2                                    main.yml

     - name: install Elastik Stack (Ubuntu)
       apt:
         name:
           - elasticsearch
           - kibana
           - logstash
         state: latest
         update_cache: yes
       when: ansible_distribution == "Ubuntu"

     - name: install Elastik Stack (CentOS)
       dnf:
         name:
           - elasticsearch
           - kibana
           - logstash
         state: latest
         update_cache: yes
       when: ansible_distribution == "CentOS"
```

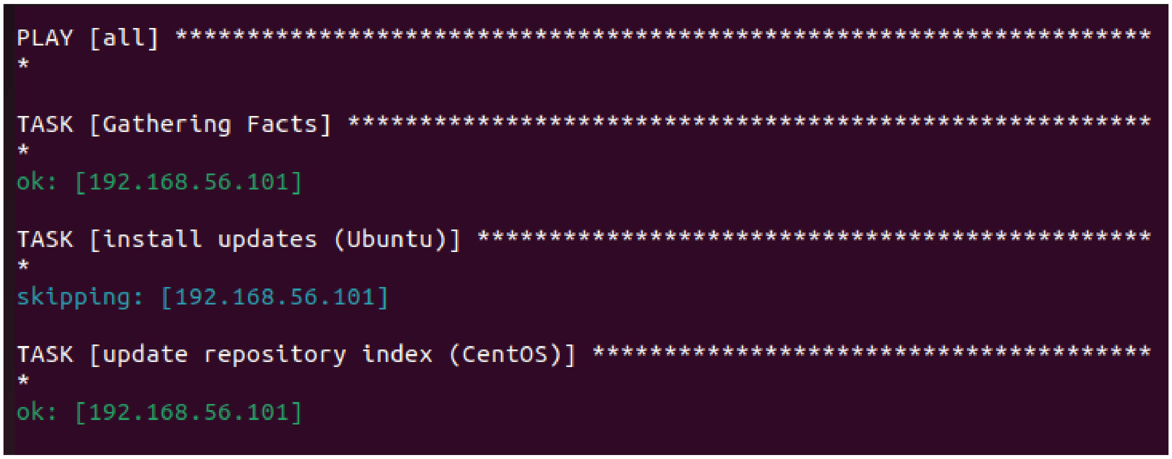
Apply the concept of creating roles and from the tasks, directory create the main.yml file. After that go back to the original directory by issuing the command “cd ..” Run the playbook once again.



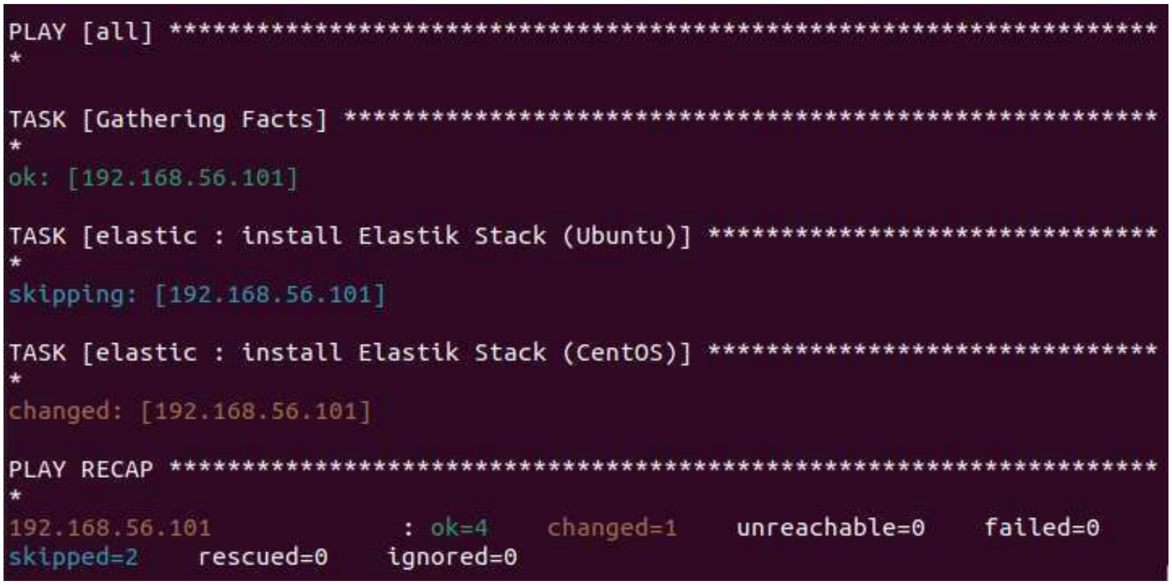
```
PLAY [all] *********************************************************************
*

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.108]

TASK [install updates (Ubuntu)] ************************************************
*
ok: [192.168.56.108]

TASK [update repository index (CentOS)] ****************************************
*
skipping: [192.168.56.108]
```



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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.108]

TASK [elastic : install Elastik Stack (Ubuntu)] ********************************
*
ok: [192.168.56.108]

TASK [elastic : install Elastik Stack (CentOS)] ********************************
*
skipping: [192.168.56.108]

PLAY RECAP *********************************************************************
*
192.168.56.108             : ok=4    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
```



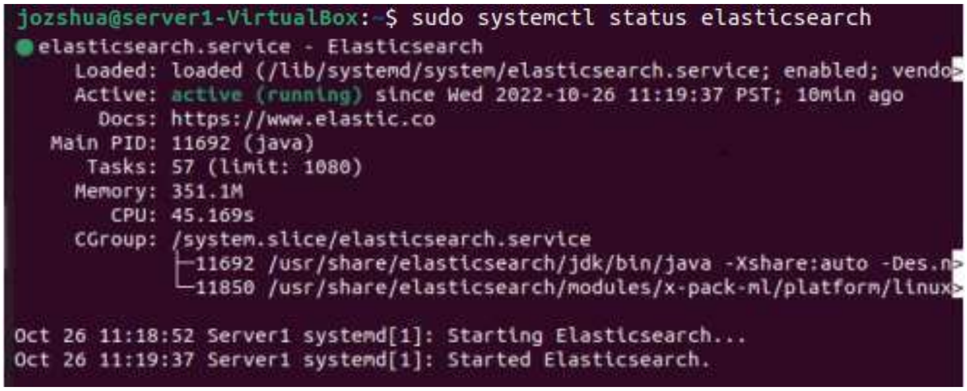
```
PLAY [all] *********************************************************************
*

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.101]

TASK [install updates (Ubuntu)] ************************************************
*
skipping: [192.168.56.101]

TASK [update repository index (CentOS)] ****************************************
*
ok: [192.168.56.101]
```



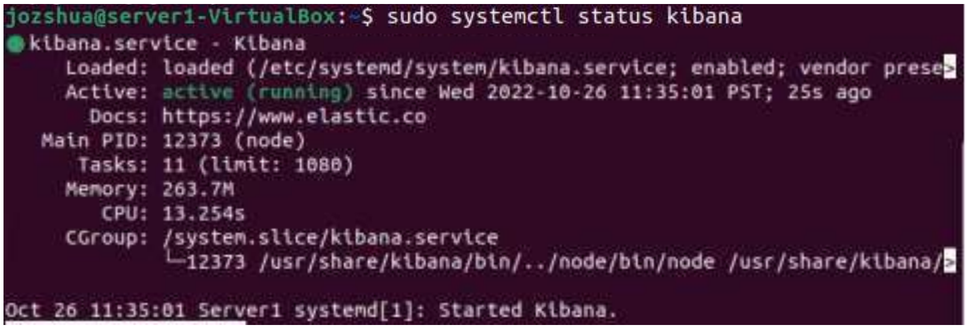
```
PLAY [all] *********************************************************************
*

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.101]

TASK [elastic : install Elastik Stack (Ubuntu)] ********************************
*
skipping: [192.168.56.101]

TASK [elastic : install Elastik Stack (CentOS)] ********************************
*
changed: [192.168.56.101]

PLAY RECAP *********************************************************************
*
192.168.56.101             : ok=4    changed=1    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
```

There are 4 tasks that were successfully executed from running this playbook. There is also a total of 4 skipped state because of having the different ansible distribution for the tasks that were assigned. There is also 1 changed state from the task of having the installation of elastic stack from the CentOS server.

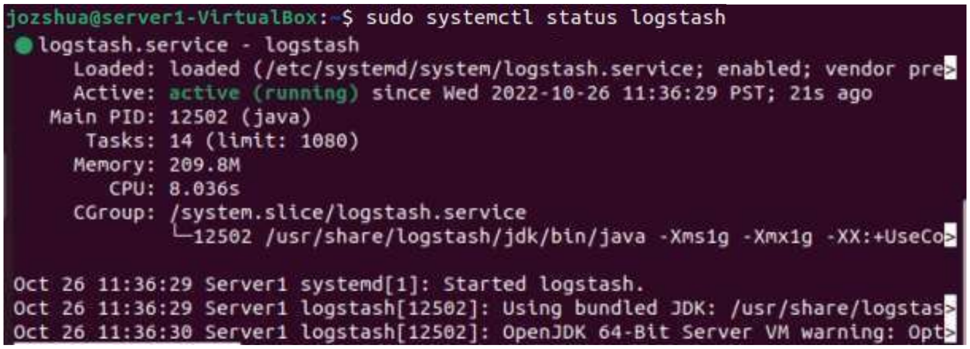
For Ubuntu:



```
jozshua@server1-VirtualBox:~$ sudo systemctl status elasticsearch
● elasticsearch.service - Elasticsearch
     Loaded: loaded (/lib/systemd/system/elasticsearch.service; enabled; vendo>
     Active: active (running) since Wed 2022-10-26 11:19:37 PST; 10min ago
       Docs: https://www.elastic.co
   Main PID: 11692 (java)
      Tasks: 57 (limit: 1080)
     Memory: 351.1M
        CPU: 45.169s
     CGroup: /system.slice/elasticsearch.service
             ├─11692 /usr/share/elasticsearch/jdk/bin/java -Xshare:auto -Des.n>
             └─11850 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux>

Oct 26 11:18:52 Server1 systemd[1]: Starting Elasticsearch...
Oct 26 11:19:37 Server1 systemd[1]: Started Elasticsearch.
```

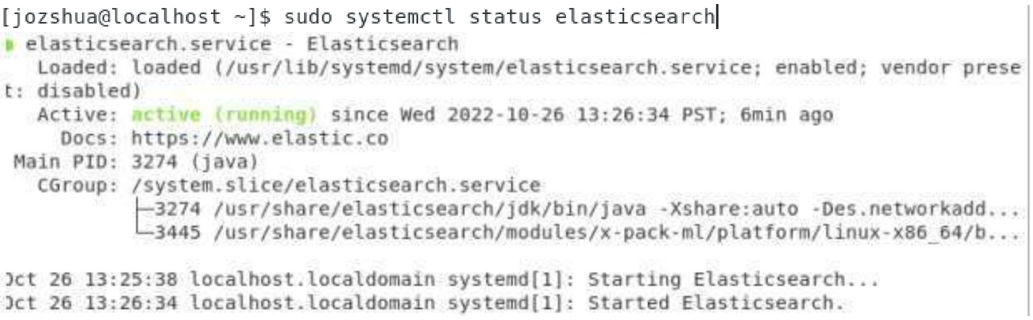


```
jozshua@server1-VirtualBox:~$ sudo systemctl status kibana
● kibana.service - Kibana
     Loaded: loaded (/etc/systemd/system/kibana.service; enabled; vendor prese>
     Active: active (running) since Wed 2022-10-26 11:35:01 PST; 25s ago
       Docs: https://www.elastic.co
   Main PID: 12373 (node)
      Tasks: 11 (limit: 1080)
     Memory: 263.7M
        CPU: 13.254s
     CGroup: /system.slice/kibana.service
             └─12373 /usr/share/kibana/bin/../node/bin/node /usr/share/kibana/>

Oct 26 11:35:01 Server1 systemd[1]: Started Kibana.
```

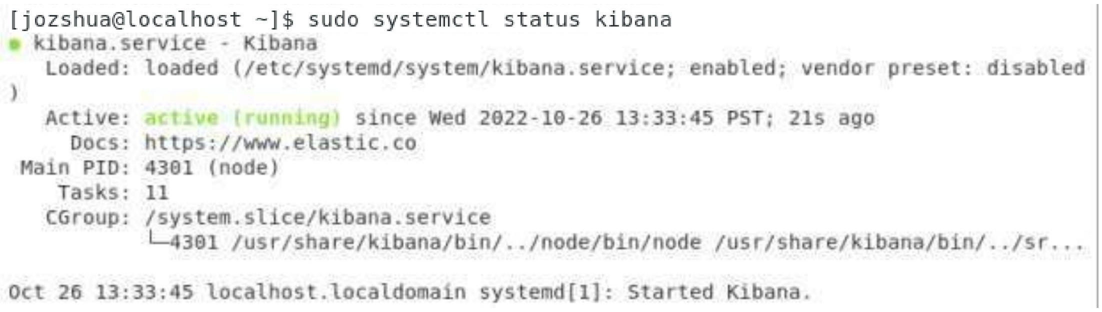


```
jozshua@server1-VirtualBox:~$ sudo systemctl status logstash
● logstash.service - logstash
     Loaded: loaded (/etc/systemd/system/logstash.service; enabled; vendor pre>
     Active: active (running) since Wed 2022-10-26 11:36:29 PST; 21s ago
   Main PID: 12502 (java)
      Tasks: 14 (limit: 1080)
     Memory: 209.8M
        CPU: 8.036s
     CGroup: /system.slice/logstash.service
             └─12502 /usr/share/logstash/jdk/bin/java -Xms1g -Xmx1g -XX:+UseCo>

Oct 26 11:36:29 Server1 systemd[1]: Started logstash.
Oct 26 11:36:29 Server1 logstash[12502]: Using bundled JDK: /usr/share/logstas>
Oct 26 11:36:30 Server1 logstash[12502]: OpenJDK 64-Bit Server VM warning: Opt>
```

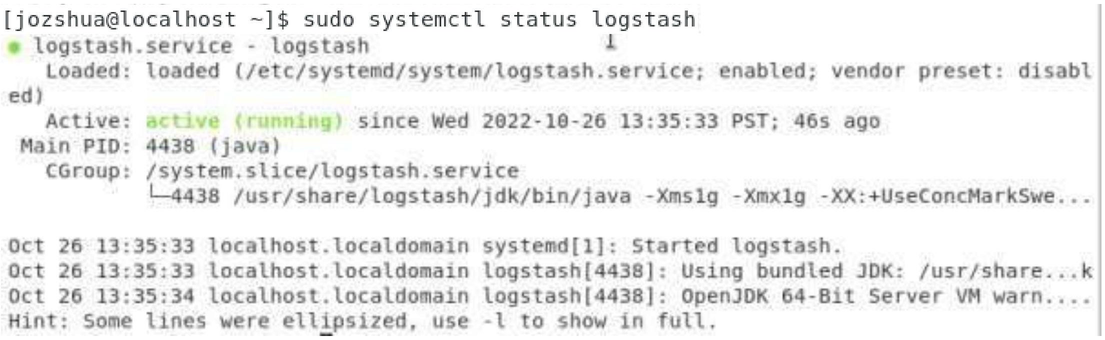
For CentOS:



```
[jozshua@localhost ~]$ sudo systemctl status elasticsearch
● elasticsearch.service - Elasticsearch
   Loaded: loaded (/usr/lib/systemd/system/elasticsearch.service; enabled; vendor prese
t: disabled)
   Active: active (running) since Wed 2022-10-26 13:26:34 PST; 6min ago
     Docs: https://www.elastic.co
 Main PID: 3274 (java)
   CGroup: /system.slice/elasticsearch.service
           ├─3274 /usr/share/elasticsearch/jdk/bin/java -Xshare:auto -Des.networkadd...
           └─3445 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux-x86_64/b...

Oct 26 13:25:38 localhost.localdomain systemd[1]: Starting Elasticsearch...
Oct 26 13:26:34 localhost.localdomain systemd[1]: Started Elasticsearch.
```

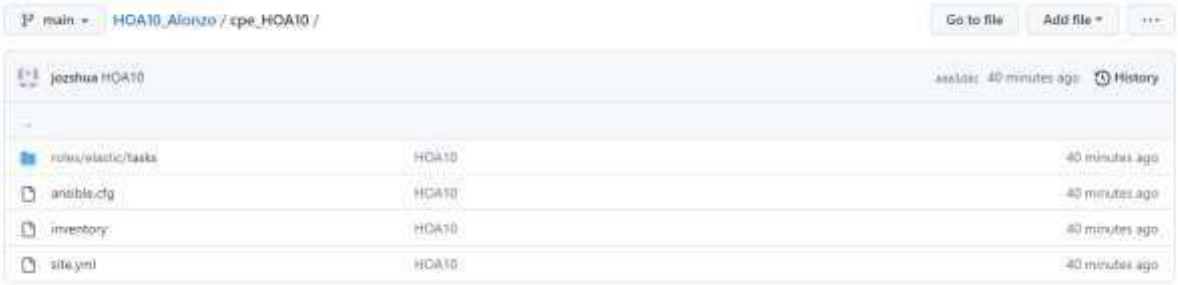


```
[jozshua@localhost ~]$ sudo systemctl status kibana
● kibana.service - Kibana
   Loaded: loaded (/etc/systemd/system/kibana.service; enabled; vendor preset: disabled
)
   Active: active (running) since Wed 2022-10-26 13:33:45 PST; 21s ago
     Docs: https://www.elastic.co
 Main PID: 4301 (node)
    Tasks: 11
   CGroup: /system.slice/kibana.service
           └─4301 /usr/share/kibana/bin/../node/bin/node /usr/share/kibana/bin/../sr...

Oct 26 13:33:45 localhost.localdomain systemd[1]: Started Kibana.
```



```
[jozshua@localhost ~]$ sudo systemctl status logstash
● logstash.service - logstash
   Loaded: loaded (/etc/systemd/system/logstash.service; enabled; vendor preset: disabl
ed)
   Active: active (running) since Wed 2022-10-26 13:35:33 PST; 46s ago
 Main PID: 4438 (java)
   CGroup: /system.slice/logstash.service
           └─4438 /usr/share/logstash/jdk/bin/java -Xms1g -Xmx1g -XX:+UseConcMarkSwe...

Oct 26 13:35:33 localhost.localdomain systemd[1]: Started logstash.
Oct 26 13:35:33 localhost.localdomain logstash[4438]: Using bundled JDK: /usr/share...k
Oct 26 13:35:34 localhost.localdomain logstash[4438]: OpenJDK 64-Bit Server VM warn....
Hint: Some lines were ellipsized, use -l to show in full.
```

Checking the installed output of the installation of elastic stack from issuing the command sudo systemctl status command.



jozshua HOA10 [illegible] 40 minutes ago

..

roles/elastic/tasks	HOA10	40 minutes ago
ansible.cfg	HOA10	40 minutes ago
inventory	HOA10	40 minutes ago
site.yml	HOA10	40 minutes ago

All the files that were created for this activity were committed to the new github repository.

Reflections:

Answer the following:

1. What are the benefits of having log monitoring tool? The benefits of having an log monitoring tool are the availability of monitoring everything at once it also more manageable because it allowing us to keep all the important logs. It also provide a better performance on the system this could make a process from finding and fixing the issue much more reliable. It can also allow you to save time because it is hassle to have the system downtime cause it takes time to fix it.

Conclusions:

In this activity, Using ansible as an Infrastructure as Code tool I created and designed the workflow of installing the elastic stack for both server distributions. Installing the elastic stack for both server distributions needs a site.yml file and main.yml file. Inside these files, there are codes that are necessary for executing the tasks for the ansible playbooks for the output. After running the playbook successfully I am able to display the installation of elastic stack for both servers by issuing the sudo systemctl status with the installed application from the elastic stack. Lastly, I make sure that all files that I created for this activity were committed to my Github repository. Therefore there is no sign of failure from the outputs so I conclude that I had finished this activity well.