Name:Calderon Ricardo B.	Date Performed:Mar 31, 2024
Course/Section:CPE232 - CPE31S1	Date Submitted:april 02, 2024
Instructor: Dr. Jonathan Taylar	Semester and SY: 2nd Sem / 2023-2024

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.

Output (screenshots and explanations)

```
calderon@workstation:~$ git clone git@github.com:Riccalder/cpe232_calderon_hoa10.git
Cloning into 'cpe232_calderon_hoa10'...
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:~$ cd cpe232_calderon_hoa10
```

This command clones a Git repository hosted on GitHub with the URL `git@github.com:Riccalder/cpe232_calderon_hoa10.git` into a local directory.

```
calderon@workstation:~/cpe232_calderon_hoa10$ sudo nano ansible.cfg
calderon@workstation:~/cpe232_calderon_hoa10$ 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:~/cpe232_calderon_hoa10$ sudo nano inventory
calderon@workstation:~/cpe232_calderon_hoa10$ cat inventory
[UbuntuServer]
192.168.56.103

[CentOSServer]
192.168.56.105
calderon@workstation:~/cpe232_calderon_hoa10$
```

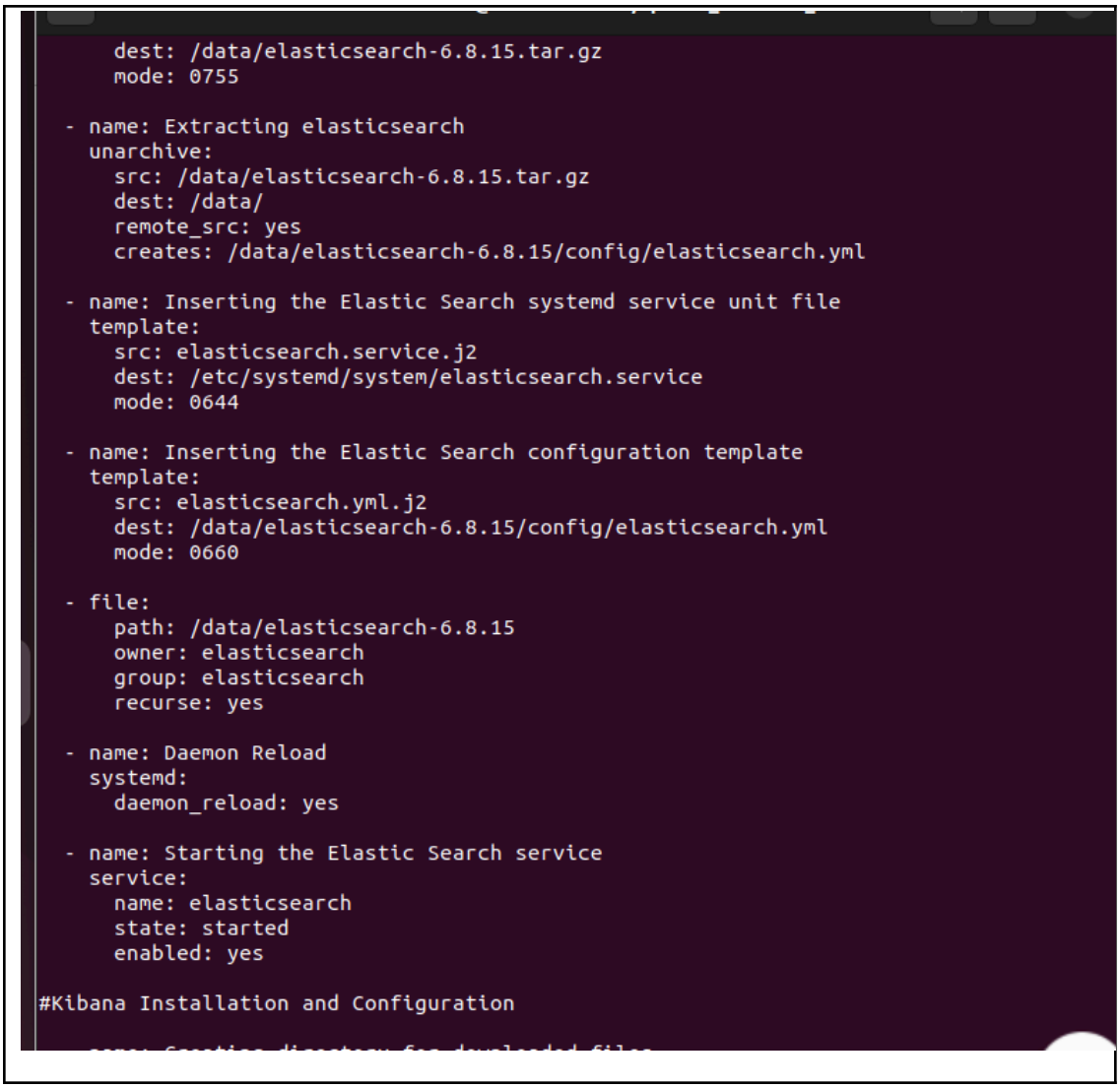
```
calderon@workstation: ~/cpe232_calderon_hoa10
calderon@workstation:~/cpe232_calderon_hoa10$ mkdir roles
calderon@workstation:~/cpe232_calderon_hoa10$ cd roles
calderon@workstation:~/cpe232_calderon_hoa10/roles$ mkdir CentOS Ubuntu
calderon@workstation:~/cpe232_calderon_hoa10/roles$ mkdir ./CentOS/tasks
calderon@workstation:~/cpe232_calderon_hoa10/roles$ mkdir ./Ubuntu/tasks
calderon@workstation:~/cpe232_calderon_hoa10/roles$ cd ../
calderon@workstation:~/cpe232_calderon_hoa10$ tree
.
├── ansible.cfg
├── inventory
├── README.md
└── roles
    ├── CentOS
    │   └── tasks
    └── Ubuntu
        └── tasks

5 directories, 3 files
calderon@workstation:~/cpe232_calderon_hoa10$
```

I created a directory named *roles* that contains 2 roles: *CentOS* and *Ubuntu* ,I also created inventory and ansible.cfg file in the directory where i'll be running Ansible commands or playbooks.

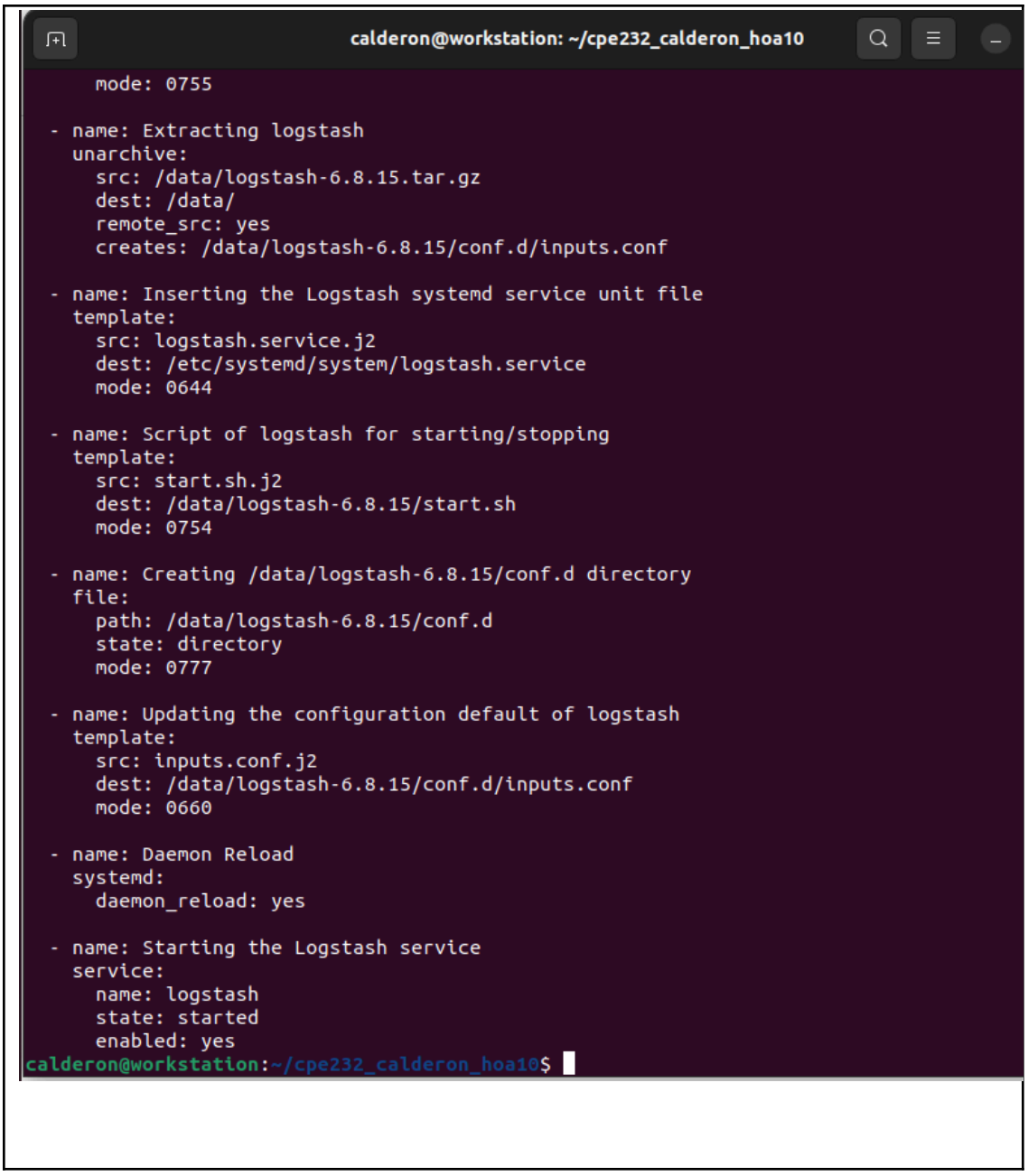
```
calderon@workstation:~/cpe232_calderon_hoa10$ sudo nano elastic_stack.yml
calderon@workstation:~/cpe232_calderon_hoa10$ cat elastic_stack.yml
---
- hosts: all
  become: true
  pre_tasks:
    - name: update repository index (CentOS)
      dnf:
        update_cache: yes
      tags: always
      when: 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

calderon@workstation:~/cpe232_calderon_hoa10$
```

The contents of the Ansible playbook are stored in a file named "elastic_stack.yml". This file contains tasks to update repository indexes and install updates on CentOS and Ubuntu hosts, as well as roles to run against specific hosts.

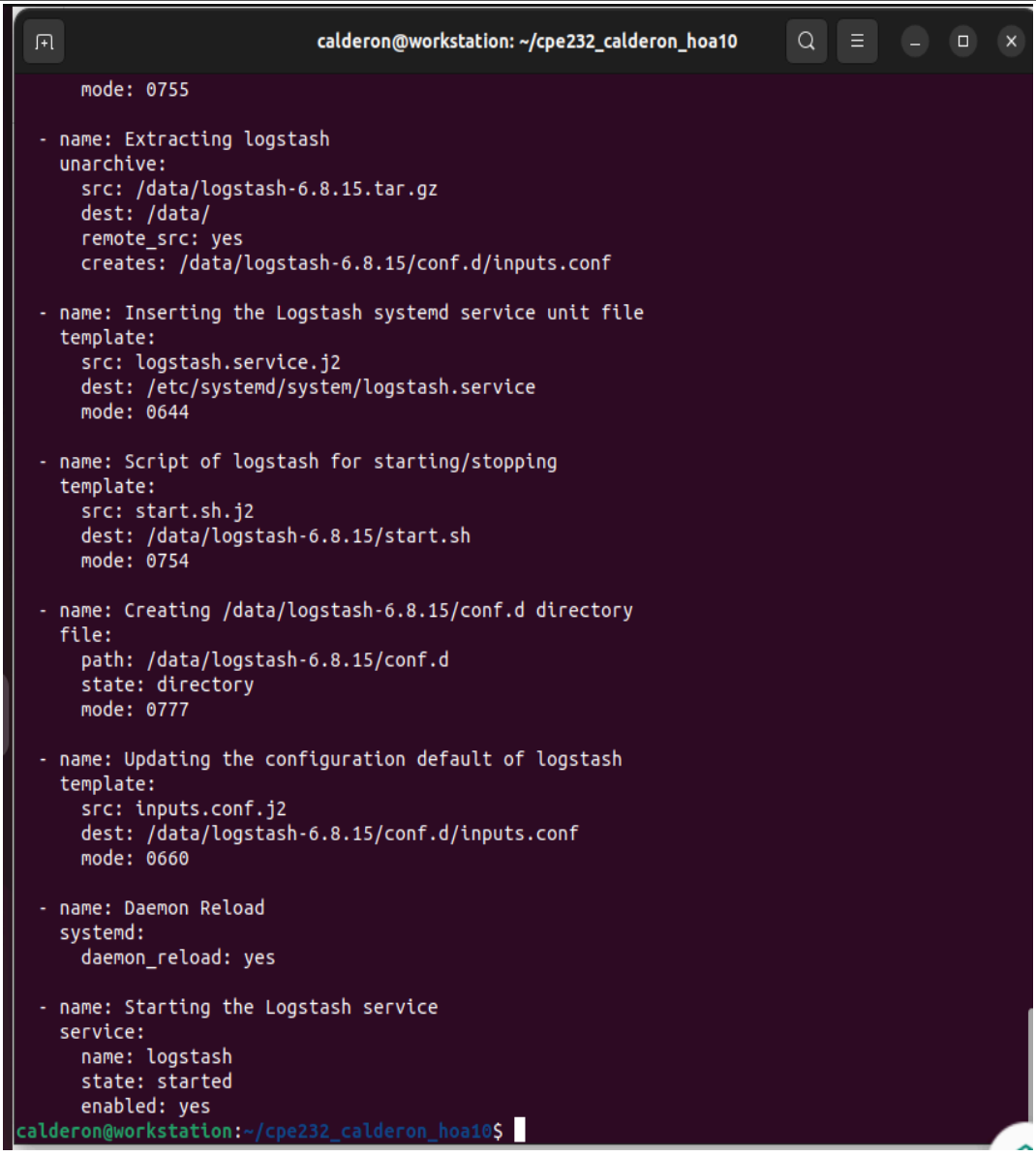
```
calderon@workstation:~/cpe232_calderon_hoa10$ sudo nano ./roles/CentOS/tasks/main.yml
calderon@workstation:~/cpe232_calderon_hoa10$ cat ./roles/CentOS/tasks/main.yml
# Elastic Search Setup

  - name: Termporarily setting the SELINUX of CentOS remote server to permissive
    selinux:
      policy: targeted
      state: permissive
    when: ansible_os_family == 'RedHat'

  - name: Updating sysctl for max_map_count
    sysctl:
      name: vm.max_map_count
      value: "262144"
      sysctl_set: yes

  - name: Adding the user 'elasticsearch'
    user:
      name: elasticsearch
      comment: elasticsearch user

  - name: Creating directory for the downloaded files
    file:
      path: /data
      state: directory
      mode: 0777

  - name: Downloading elasticsearch tar ball
    get_url:
      url: https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-6.8.15.tar.gz
      dest: /data/elasticsearch-6.8.15.tar.gz
      mode: 0755

  - name: Extracting elasticsearch
    unarchive:
      src: /data/elasticsearch-6.8.15.tar.gz
      dest: /data/
      remote_src: yes
      creates: /data/elasticsearch-6.8.15/config/elasticsearch.yml
```



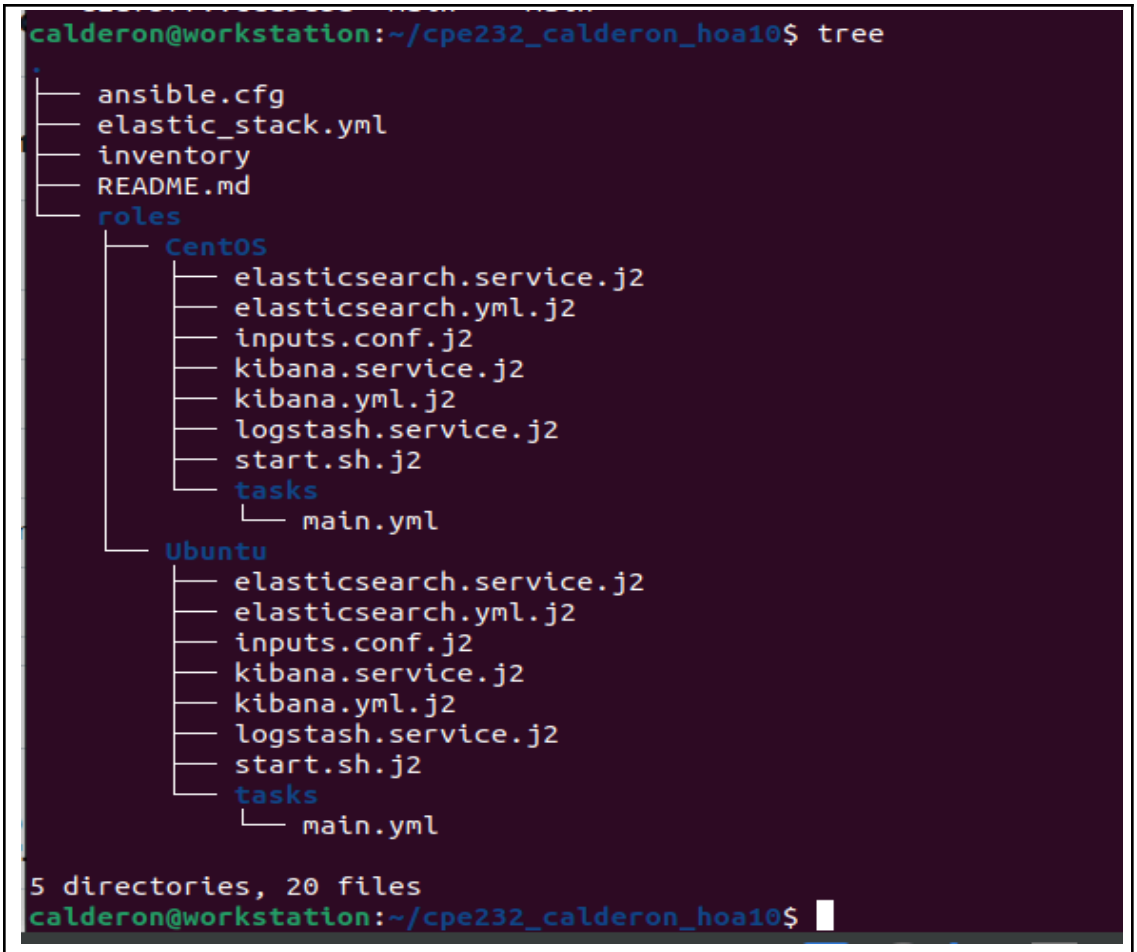
```
    dest: /data/elasticsearch-6.8.15.tar.gz
    mode: 0755

- name: Extracting elasticsearch
  unarchive:
    src: /data/elasticsearch-6.8.15.tar.gz
    dest: /data/
    remote_src: yes
    creates: /data/elasticsearch-6.8.15/config/elasticsearch.yml

- name: Inserting the Elastic Search systemd service unit file
  template:
    src: elasticsearch.service.j2
    dest: /etc/systemd/system/elasticsearch.service
    mode: 0644

- name: Inserting the Elastic Search configuration template
  template:
    src: elasticsearch.yml.j2
    dest: /data/elasticsearch-6.8.15/config/elasticsearch.yml
    mode: 0660

- file:
    path: /data/elasticsearch-6.8.15
    owner: elasticsearch
    group: elasticsearch
    recurse: yes

- name: Daemon Reload
  systemd:
    daemon_reload: yes

- name: Starting the Elastic Search service
  service:
    name: elasticsearch
    state: started
    enabled: yes

#Kibana Installation and Configuration
```


calderon@workstation: ~/cpe232_calderon_hoa10
mode: 0755
- name: Extracting logstash
unarchive:
src: /data/logstash-6.8.15.tar.gz
dest: /data/
remote_src: yes
creates: /data/logstash-6.8.15/conf.d/inputs.conf
- name: Inserting the Logstash systemd service unit file
template:
src: logstash.service.j2
dest: /etc/systemd/system/logstash.service
mode: 0644
- name: Script of logstash for starting/stopping
template:
src: start.sh.j2
dest: /data/logstash-6.8.15/start.sh
mode: 0754
- name: Creating /data/logstash-6.8.15/conf.d directory
file:
path: /data/logstash-6.8.15/conf.d
state: directory
mode: 0777
- name: Updating the configuration default of logstash
template:
src: inputs.conf.j2
dest: /data/logstash-6.8.15/conf.d/inputs.conf
mode: 0660
- name: Daemon Reload
systemd:
daemon_reload: yes
- name: Starting the Logstash service
service:
name: logstash
state: started
enabled: yes
calderon@workstation:~/cpe232_calderon_hoa10$

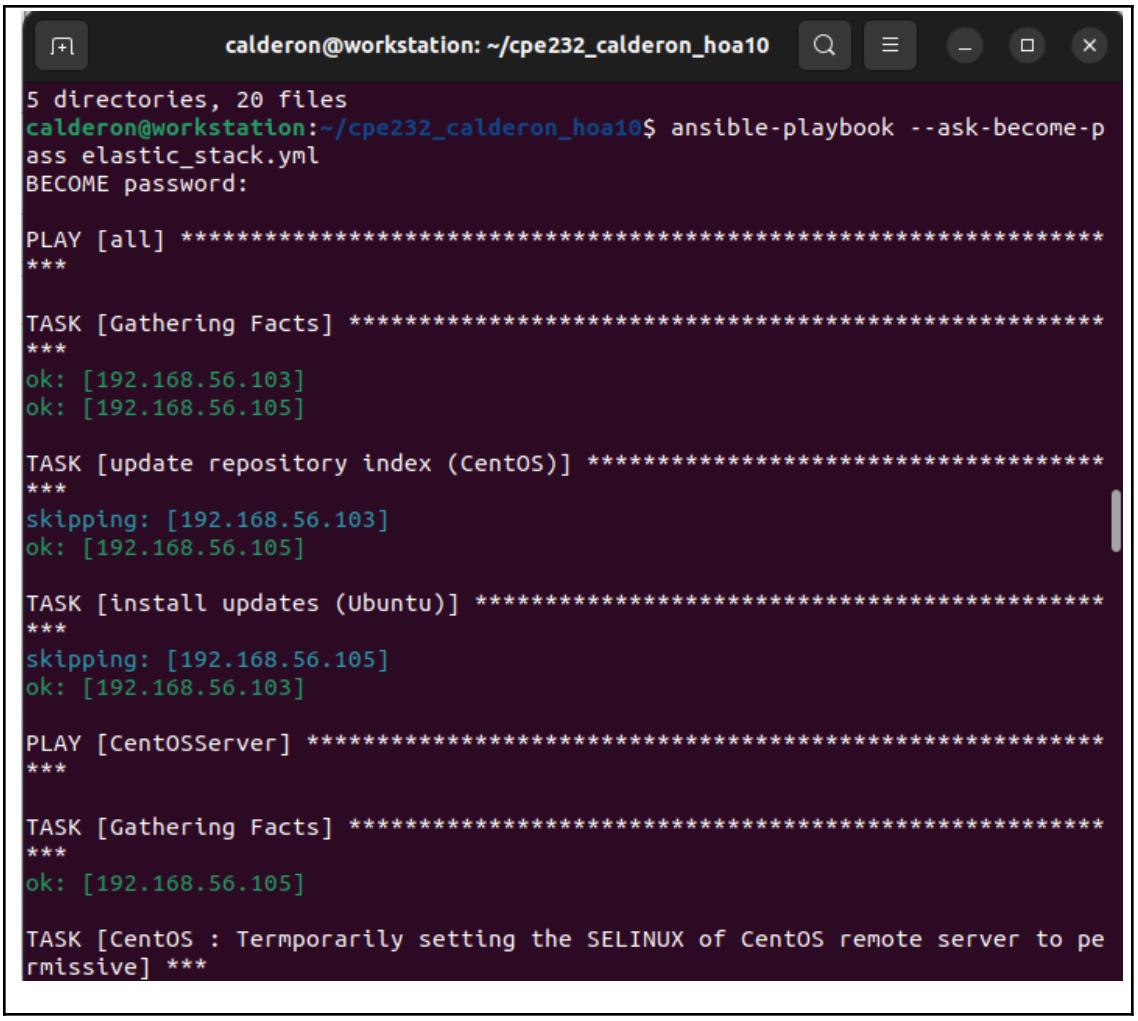
```
calderon@workstation:~/cpe232_calderon_hoa10$ sudo nano ./roles/Ubuntu/tasks/main.yml
calderon@workstation:~/cpe232_calderon_hoa10$ cat ./roles/Ubuntu/tasks/main.yml
# Elastic Search Setup

  - name: Termporarily setting the SELINUX of Ubuntu remote server to permissive
    selinux:
      policy: targeted
      state: permissive
    when: ansible_os_family == 'Ubuntu'

  - name: Updating sysctl for max_map_count
    sysctl:
      name: vm.max_map_count
      value: "262144"
      sysctl_set: yes

  - name: Adding the user 'elasticsearch'
    user:
      name: elasticsearch
      comment: elasticsearch user

  - name: Creating directory for the downloaded files
    file:
      path: /data
      state: directory
      mode: 0777

  - name: Downloading elasticsearch tar ball
    get_url:
      url: https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-6.8.15.ta
      dest: /data/elasticsearch-6.8.15.tar.gz
      mode: 0755

  - name: Extracting elasticsearch
    unarchive:
      src: /data/elasticsearch-6.8.15.tar.gz
      dest: /data/
      remote_src: yes
      creates: /data/elasticsearch-6.8.15/config/elasticsearch.yml
```

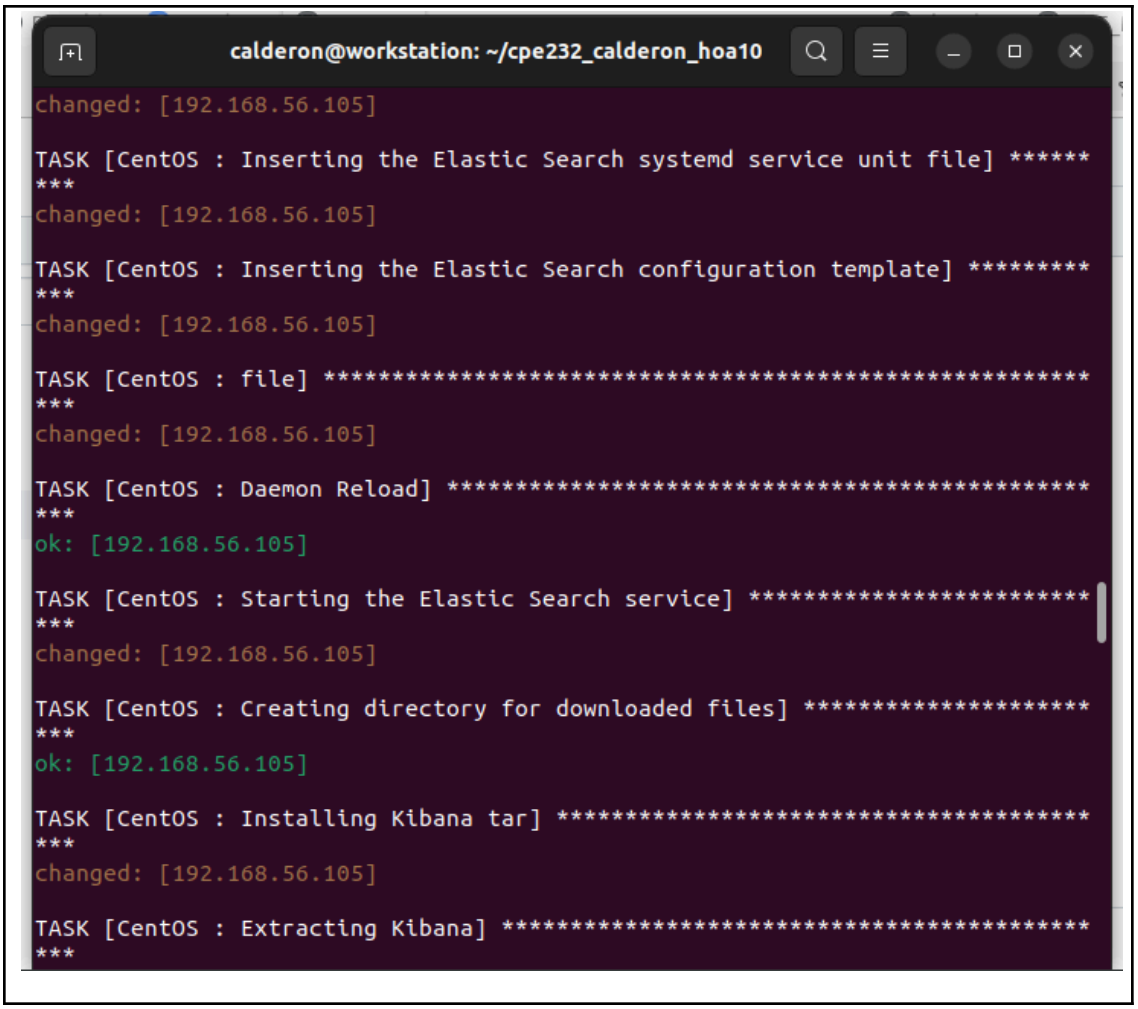


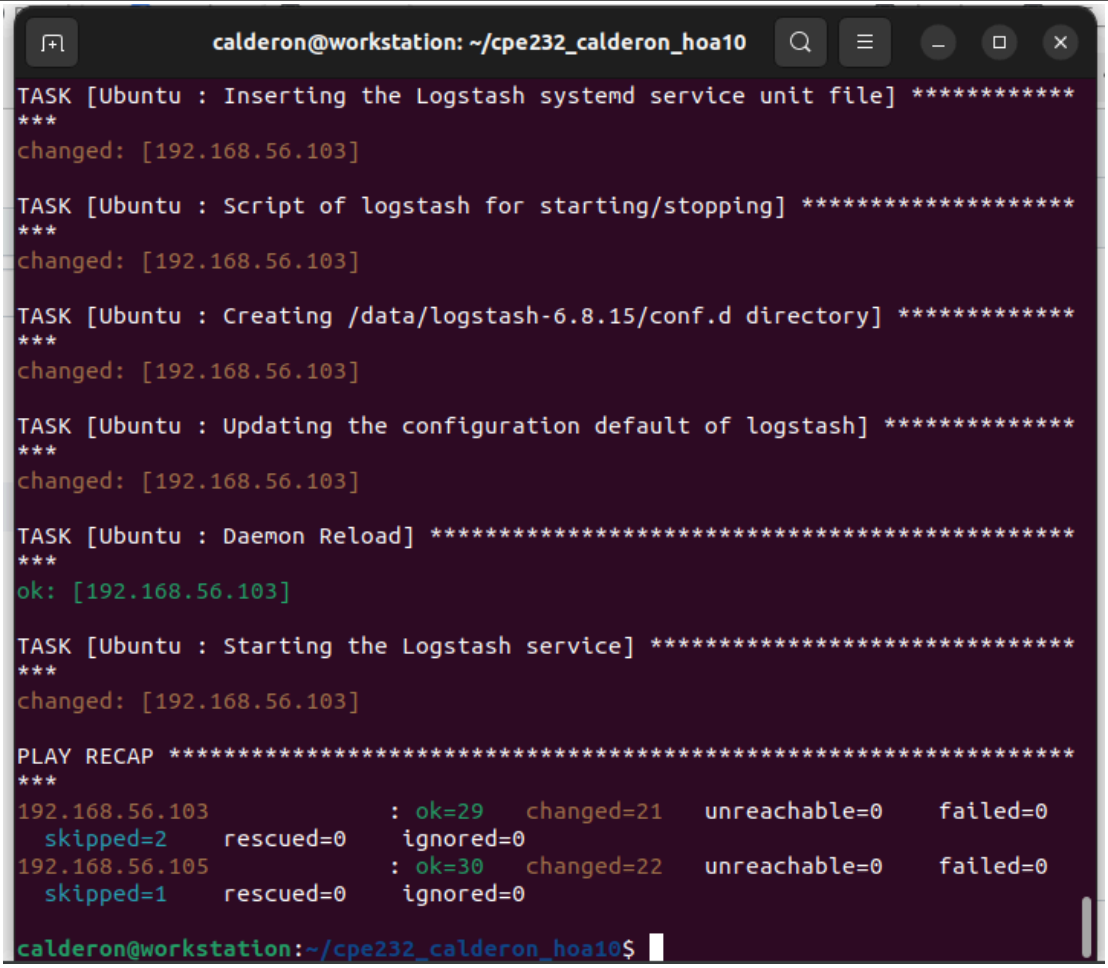
This is an Ansible playbook in YAML format. It contains tasks to update repository indexes and install updates on CentOS and Ubuntu hosts, as well as roles to run against specific hosts. The playbook uses tags for task identification and has elevated privileges with "become: true". Its purpose is to manage multiple hosts with reusable roles.



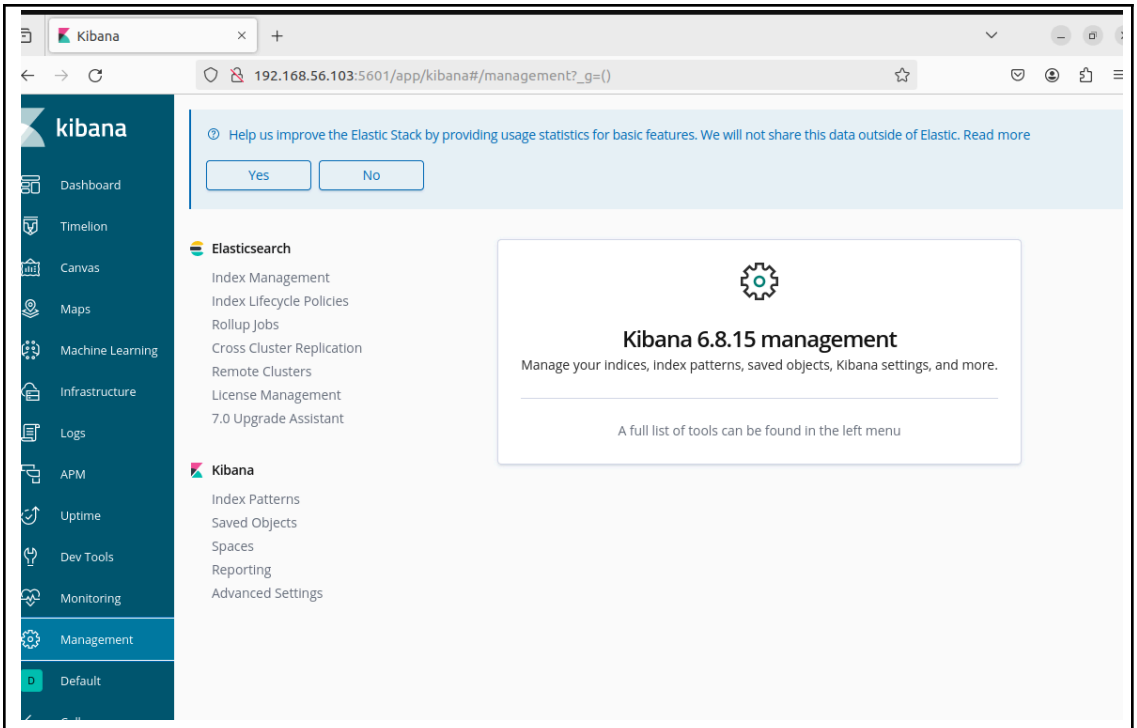
Through tree, viewing directory with subdirectories for roles (CentOS and Ubuntu), each containing task files.


calderon@workstation: ~/cpe232_calderon_hoa10
5 directories, 20 files
calderon@workstation:~/cpe232_calderon_hoa10$ ansible-playbook --ask-become-pass elastic_stack.yml
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]
ok: [192.168.56.103]
PLAY [CentOSServer] ***
TASK [Gathering Facts] ***
ok: [192.168.56.105]
TASK [CentOS : Termporarily setting the SELINUX of CentOS remote server to permissive] ***


calderon@workstation: ~/cpe232_calderon_hoa10
changed: [192.168.56.105]
TASK [CentOS : Inserting the Elastic Search systemd service unit file] ******
***
changed: [192.168.56.105]
TASK [CentOS : Inserting the Elastic Search configuration template] *********
***
changed: [192.168.56.105]
TASK [CentOS : file] ***
***
changed: [192.168.56.105]
TASK [CentOS : Daemon Reload] ***
***
ok: [192.168.56.105]
TASK [CentOS : Starting the Elastic Search service] ************************
***
changed: [192.168.56.105]
TASK [CentOS : Creating directory for downloaded files] *********************
***
ok: [192.168.56.105]
TASK [CentOS : Installing Kibana tar] ***************************************
***
changed: [192.168.56.105]
TASK [CentOS : Extracting Kibana] ***
***

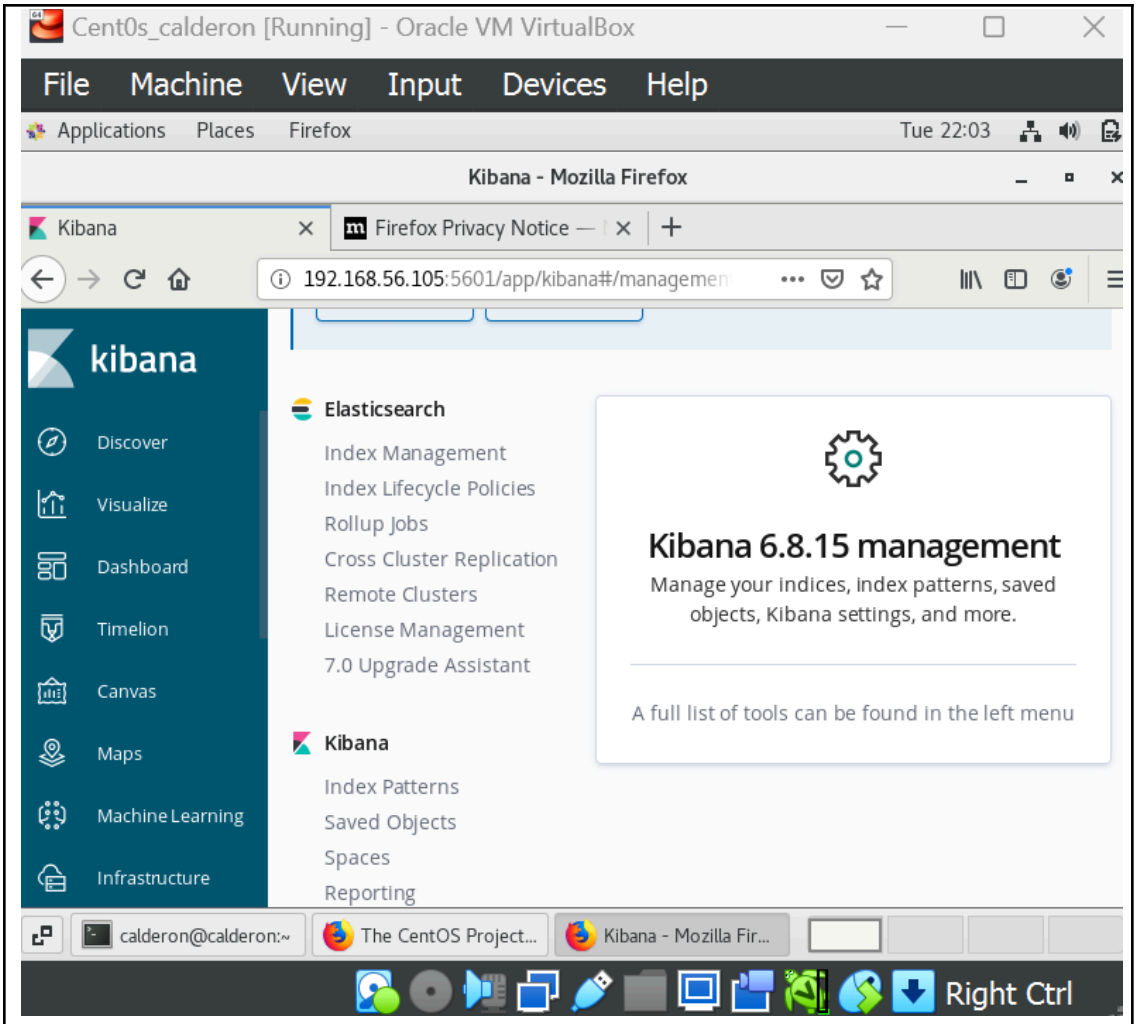


The output indicates that the Ansible playbook executed successfully against the two destinations, with 29 and 30 tasks completed without changes on 192.168.56.103 and 192.168.56.105 respectively.



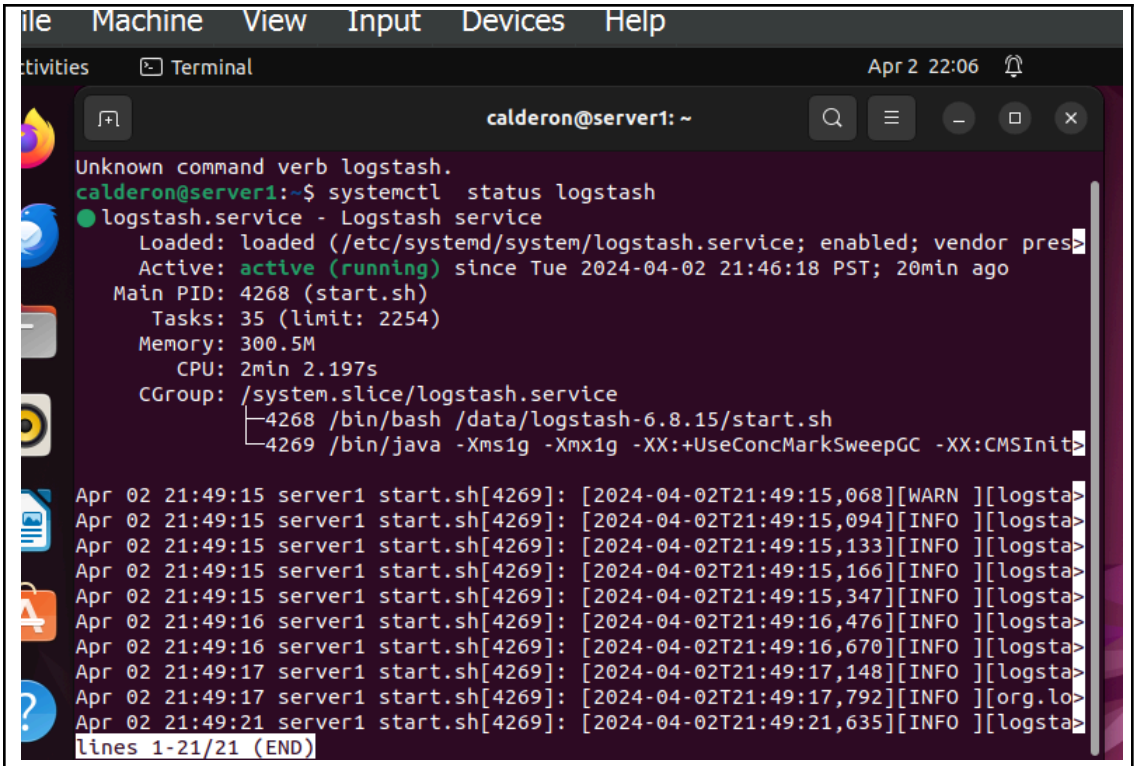
Ubuntu

Elastic Stack is working in *Ubuntu*. I opened a web browser and typed *'192.168.56.103:5601'*. I also opened the terminal and typed the command *'systemctl status logstash'* to check if *Logstash* service is actively running.

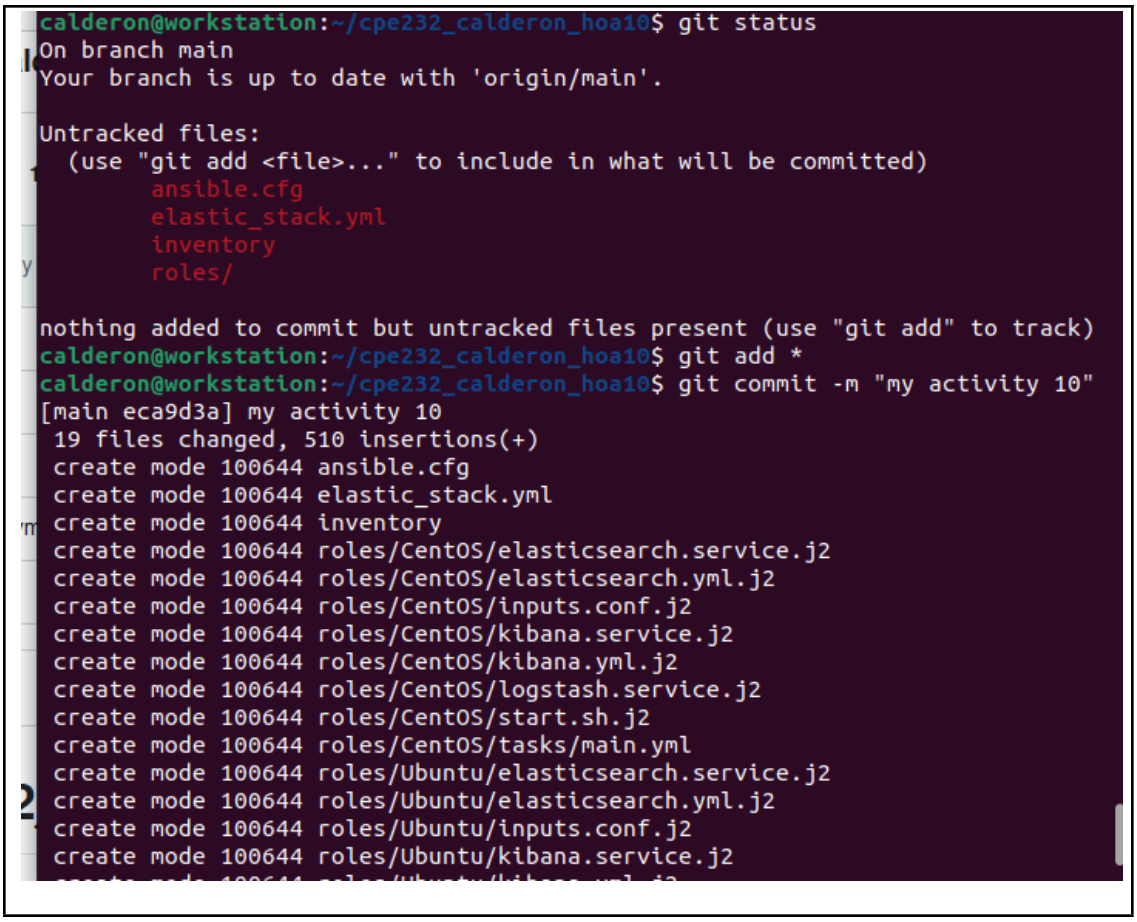


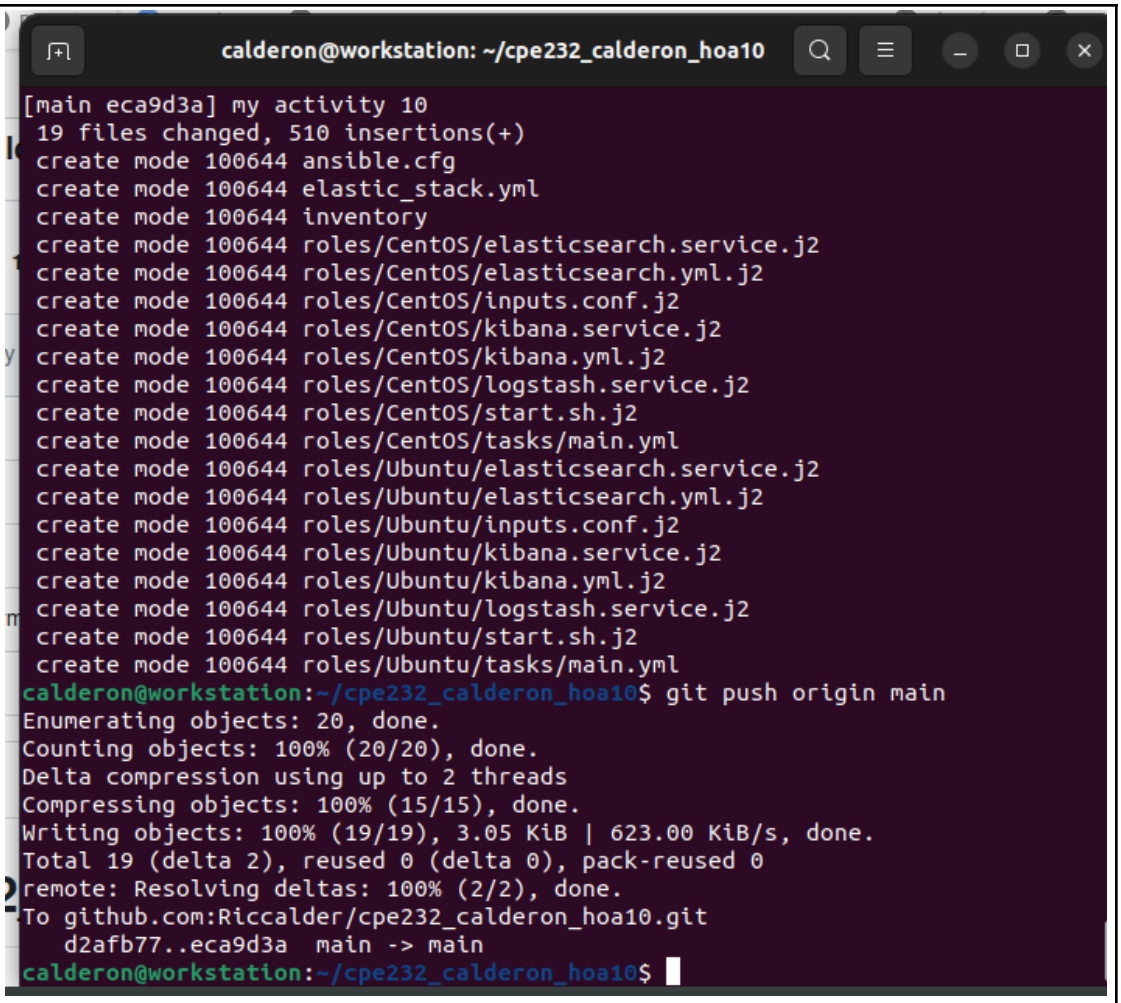
CentOS

Elastic Stack is working in *CentOS*. I opened a web browser and typed *'192.168.56.105:5601'*. I also opened the terminal and typed the command *'systemctl status logstash'* to check if *Logstash* service is actively running.

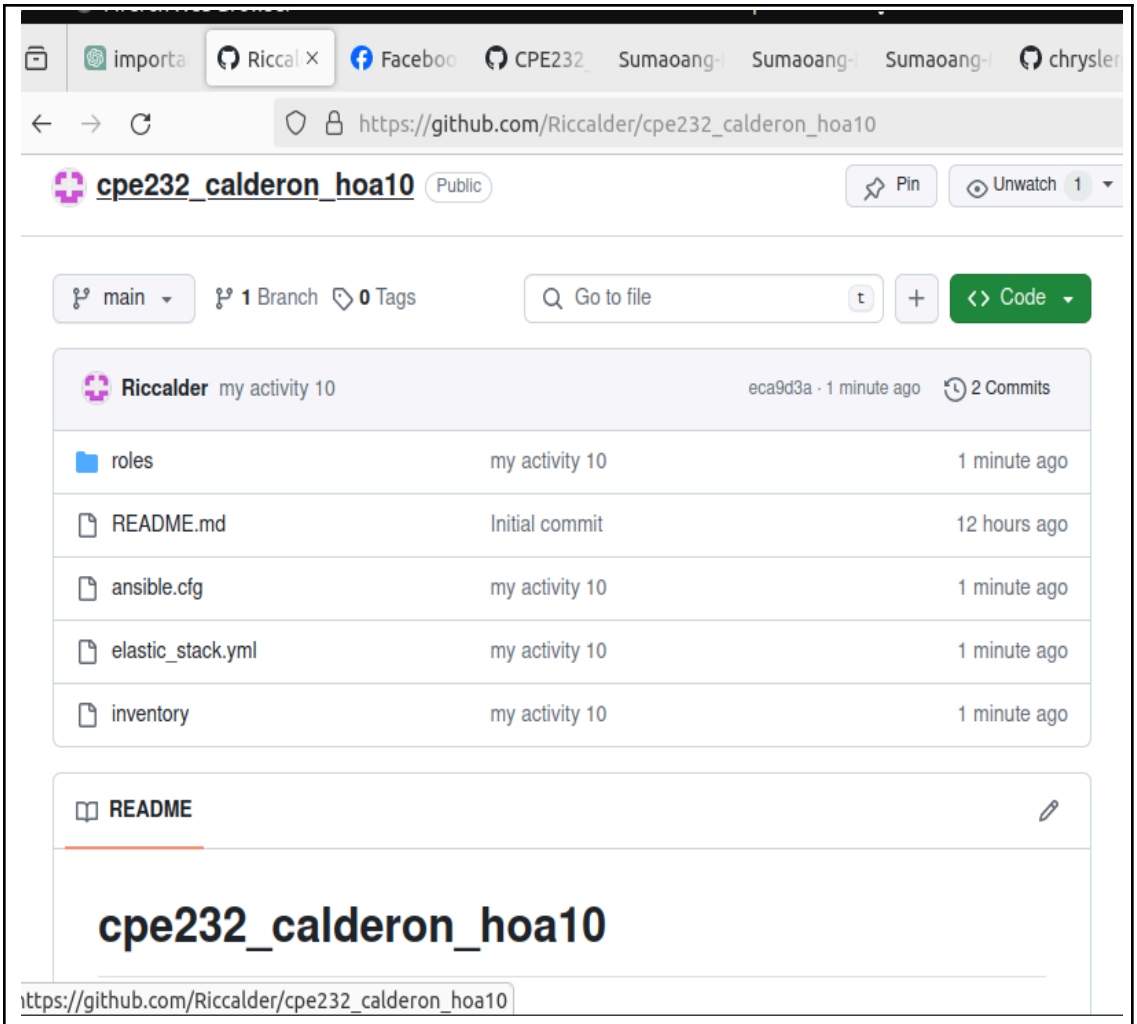


It means that logstash successfully runs without errors or issues. It could be verified by checking the logs or the status of the service to ensure that it is indeed active and running as expected.


calderon@workstation:~/cpe232_calderon_hoa10$ git status
On branch main
Your branch is up to date with 'origin/main'.
Untracked files:
(use "git add <file>..." to include in what will be committed)
ansible.cfg
elastic_stack.yml
inventory
roles/
nothing added to commit but untracked files present (use "git add" to track)
calderon@workstation:~/cpe232_calderon_hoa10$ git add *
calderon@workstation:~/cpe232_calderon_hoa10$ git commit -m "my activity 10"
[main eca9d3a] my activity 10
19 files changed, 510 insertions(+)
create mode 100644 ansible.cfg
create mode 100644 elastic_stack.yml
create mode 100644 inventory
create mode 100644 roles/CentOS/elasticsearch.service.j2
create mode 100644 roles/CentOS/elasticsearch.yml.j2
create mode 100644 roles/CentOS/inputs.conf.j2
create mode 100644 roles/CentOS/kibana.service.j2
create mode 100644 roles/CentOS/kibana.yml.j2
create mode 100644 roles/CentOS/logstash.service.j2
create mode 100644 roles/CentOS/start.sh.j2
create mode 100644 roles/CentOS/tasks/main.yml
create mode 100644 roles/Ubuntu/elasticsearch.service.j2
create mode 100644 roles/Ubuntu/elasticsearch.yml.j2
create mode 100644 roles/Ubuntu/inputs.conf.j2
create mode 100644 roles/Ubuntu/kibana.service.j2



The commands show a series of updates to a Git repository. The "git add" command adds all files for staging, "git commit" creates a commit with a message, and "git push" uploads the changes to the remote repository.



After executing the commands, the changes made to the Git repository are uploaded to the remote GitHub repository.The commit message and the changes made can also be seen in the commit history of the repository.

GitHub Repository link:

https://github.com/Riccalder/cpe232_calderon_hoa10

Reflections:

Answer the following:

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

The ability to monitor log files generated by servers, applications, and networks. With this monitoring, administrators can detect and track important system events and performance metrics.

Conclusions:

This activity involves creating a workflow using Ansible as an Infrastructure as Code (IaC) tool to install, configure, and manage enterprise log monitoring tools. Specifically, the task requires the installation and configuration of Elastic Stack, including ElasticSearch, Kibana, and Logstash, on separate hosts using the concept of creating roles. Graylog is also discussed as an alternative log monitoring tool. The completion of this activity demonstrates proficiency in using Ansible as an IaC tool, system administration skills, and the ability to effectively manage and monitor log files.