Name: Davonn Escobilla	Date Performed: 28/10/2022
Course/Section: CPE31S24	Date Submitted: 28/10/2022
Instructor: Dr. Jonathan Taylar	Semester and SY: 1st sem, 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.

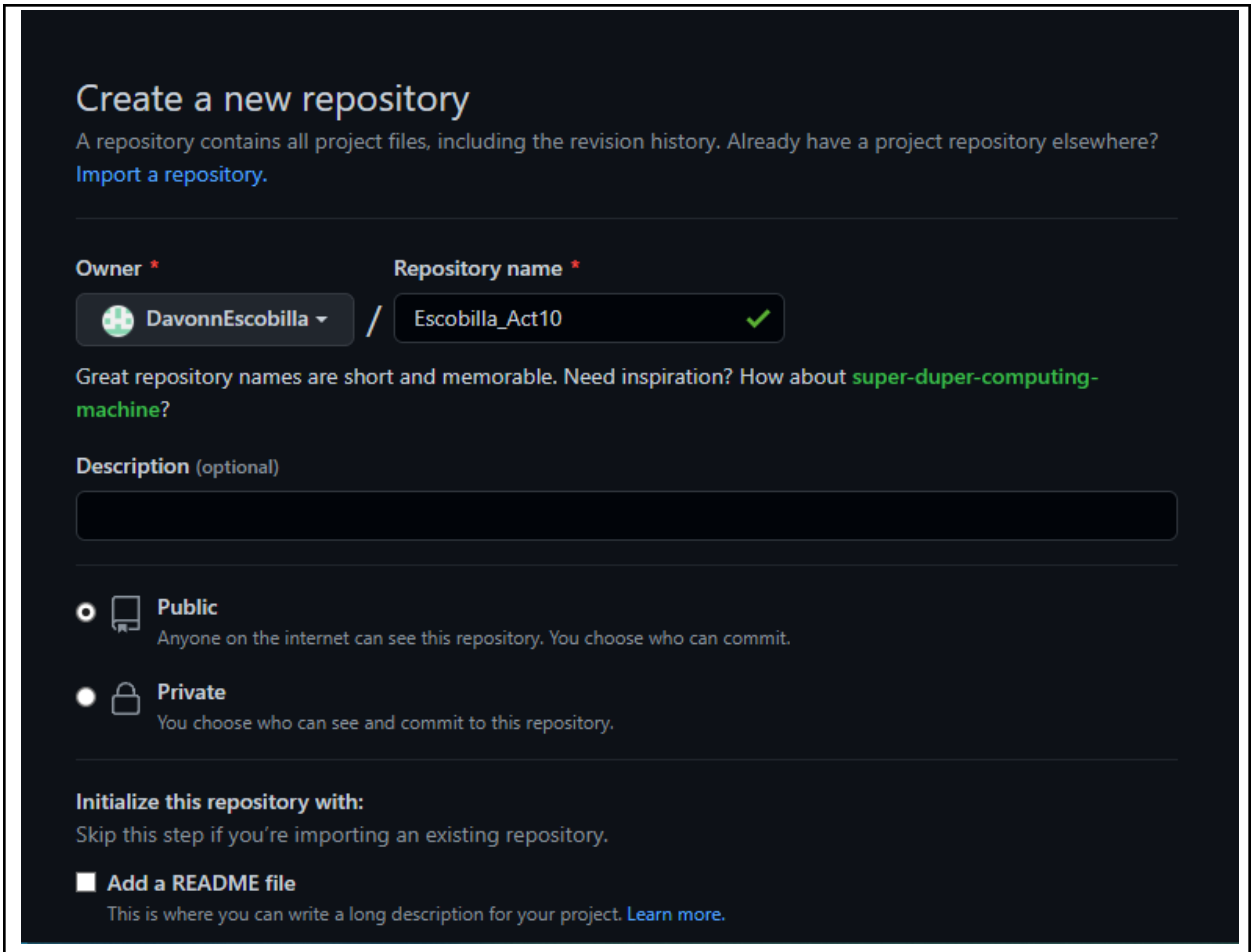
Source: https://www.graylog.org/products/open-source

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)

First step is to create a repository for Activity 10.



Second step, clone the repository to the managed node.

```
davonn@workstation:~$ git clone git@github.com:DavonnEscobilla/Escobilla_Act10.
git
Cloning into 'Escobilla_Act10'...
warning: You appear to have cloned an empty repository.
davonn@workstation:~$ ls
CPE232_Davonn     Downloads                Music      Templates
CPE232_Escobilla  Escobilla_Act10          nano.save  Videos
Desktop           Escobilla_Act8Nagios     Pictures
Documents         Escobilla_Act9Prometheus  Public
```

Third step, create ansible.cfg and inventory file for control nodes.

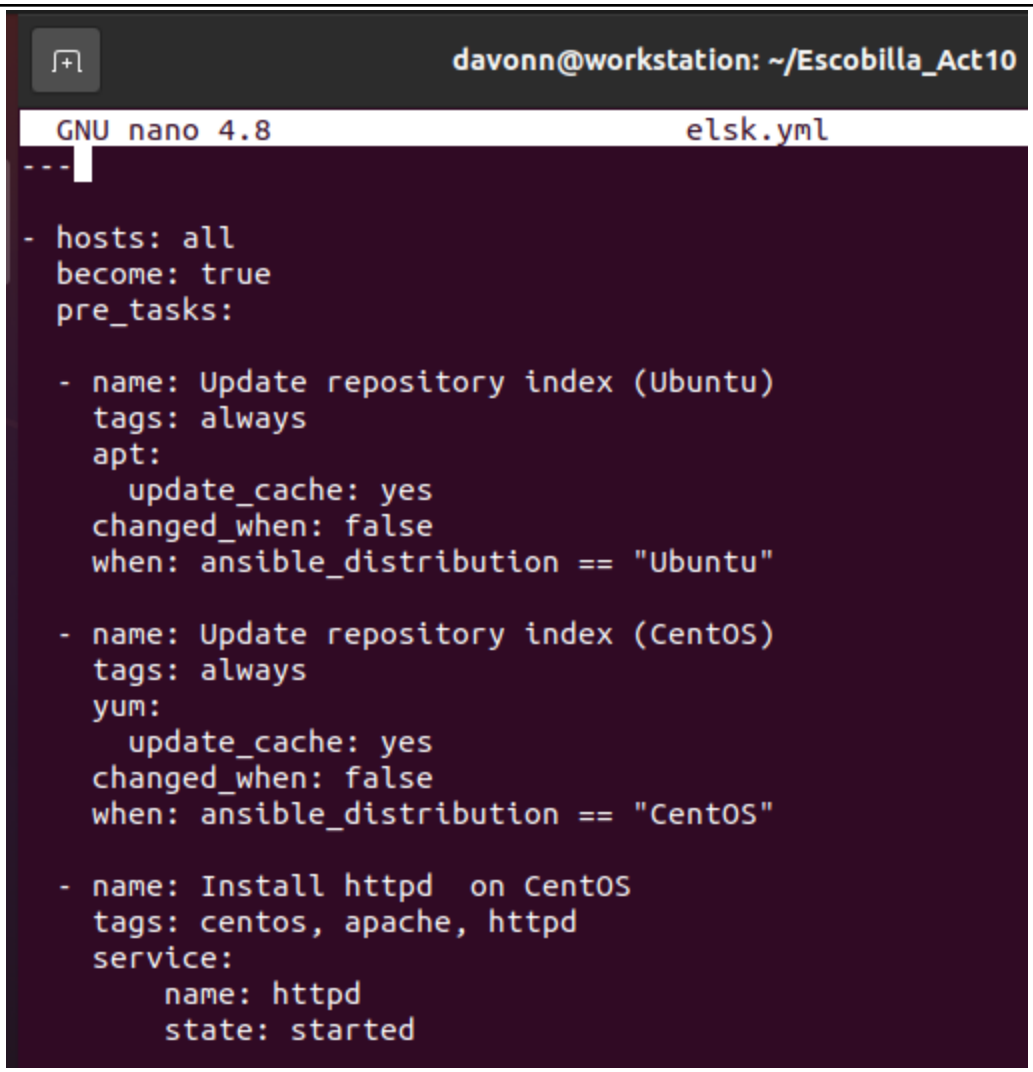
```
davonn@workstation: ~/Escobilla_Act10
  GNU nano 4.8                          ansible.cfg
[defaults]

inventory = inventory
private_key_file = ~/.ssh/ansible
```

```
davonn@workstation: ~/Escobilla_Act10
  GNU nano 4.8                          inventory
[CentOS]
192.168.56.105

[Ubuntu]
192.168.56.103
```

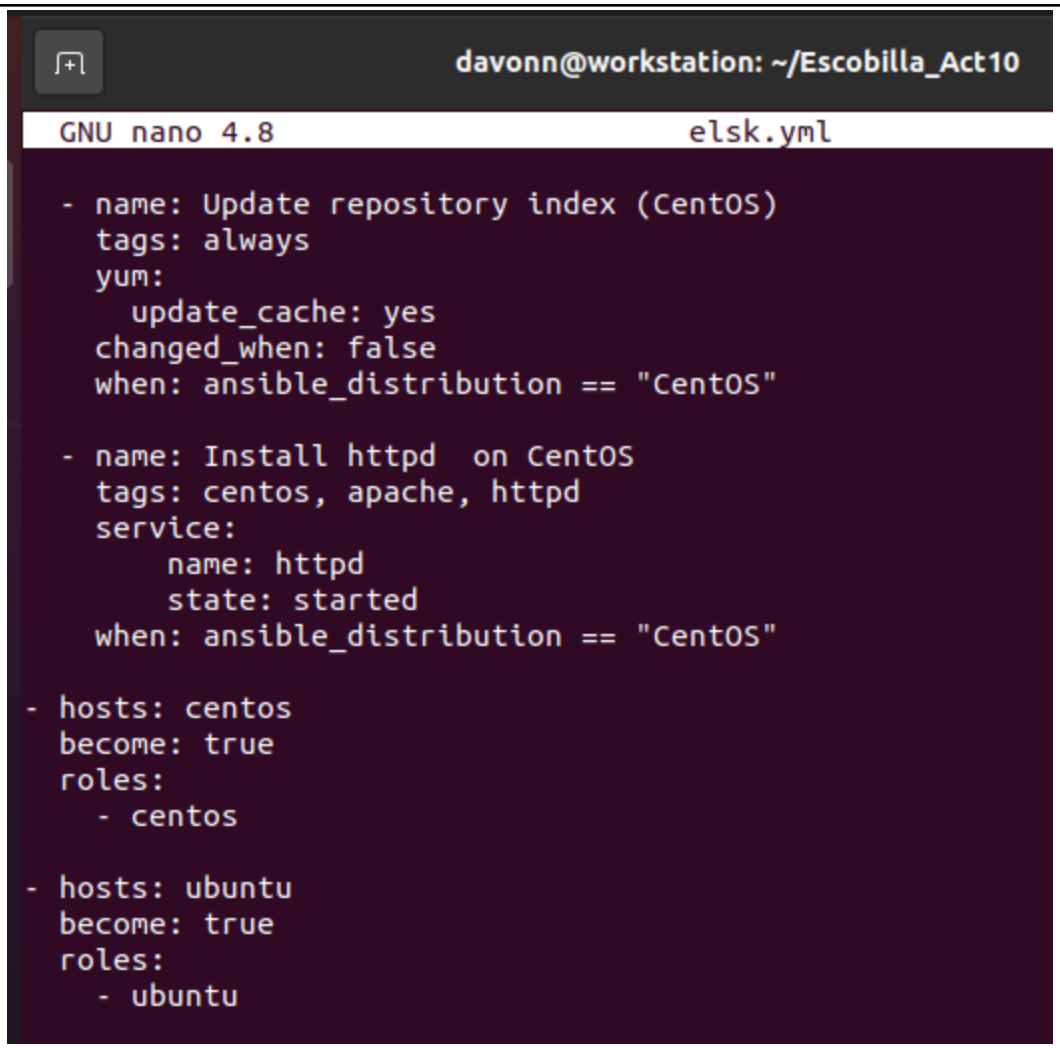
Before proceeding to the next step, check the connectivity of the managed node to the control node.

```
davonn@workstation: ~/Escobilla_Act10
davonn@workstation:~/Escobilla_Act10$ ansible -m ping all
192.168.56.105 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
192.168.56.103 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
```

Fourth step, create the elsk.yml and configure the tasks.

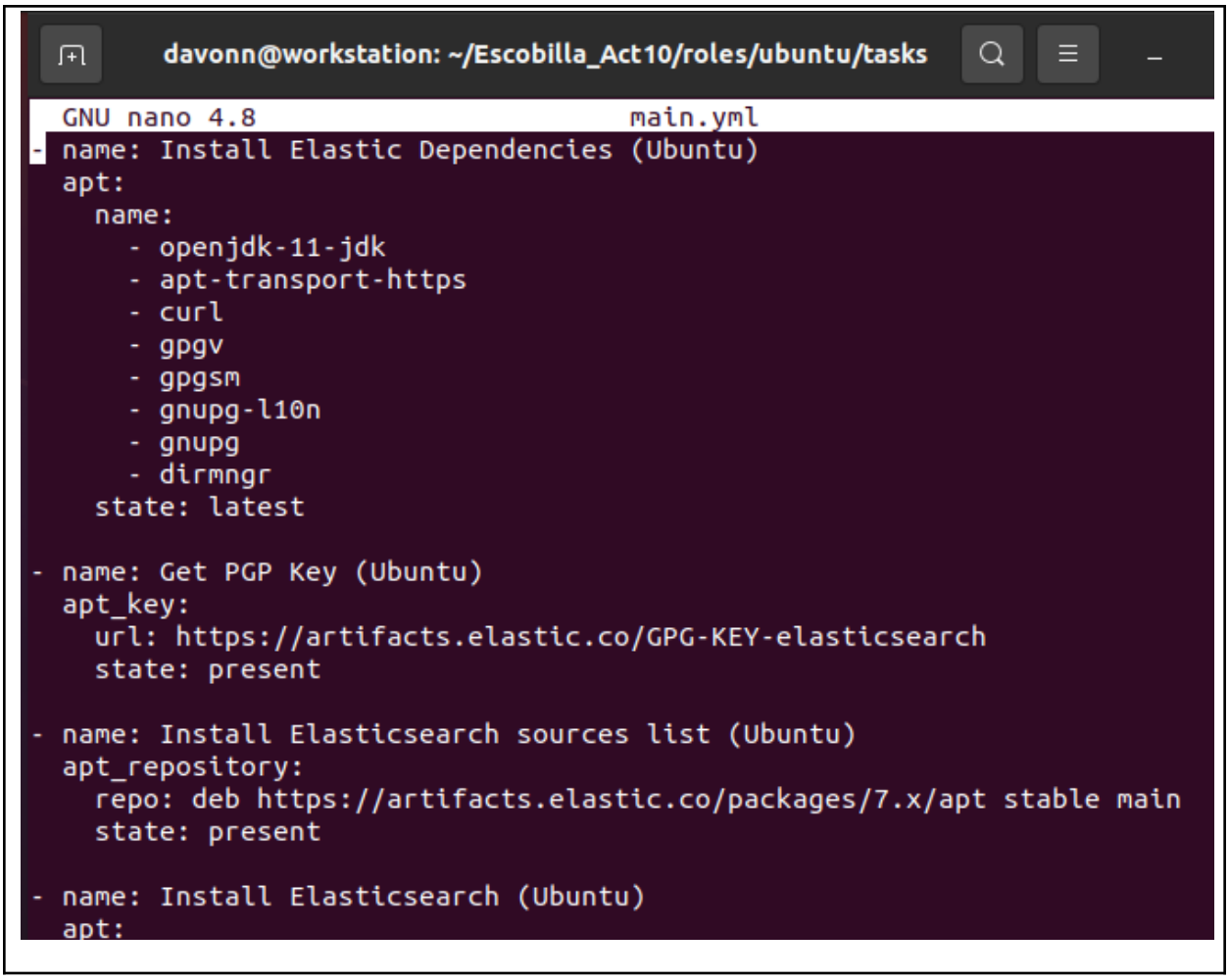

davonn@workstation: ~/Escobilla_Act10
GNU nano 4.8 elsk.yml

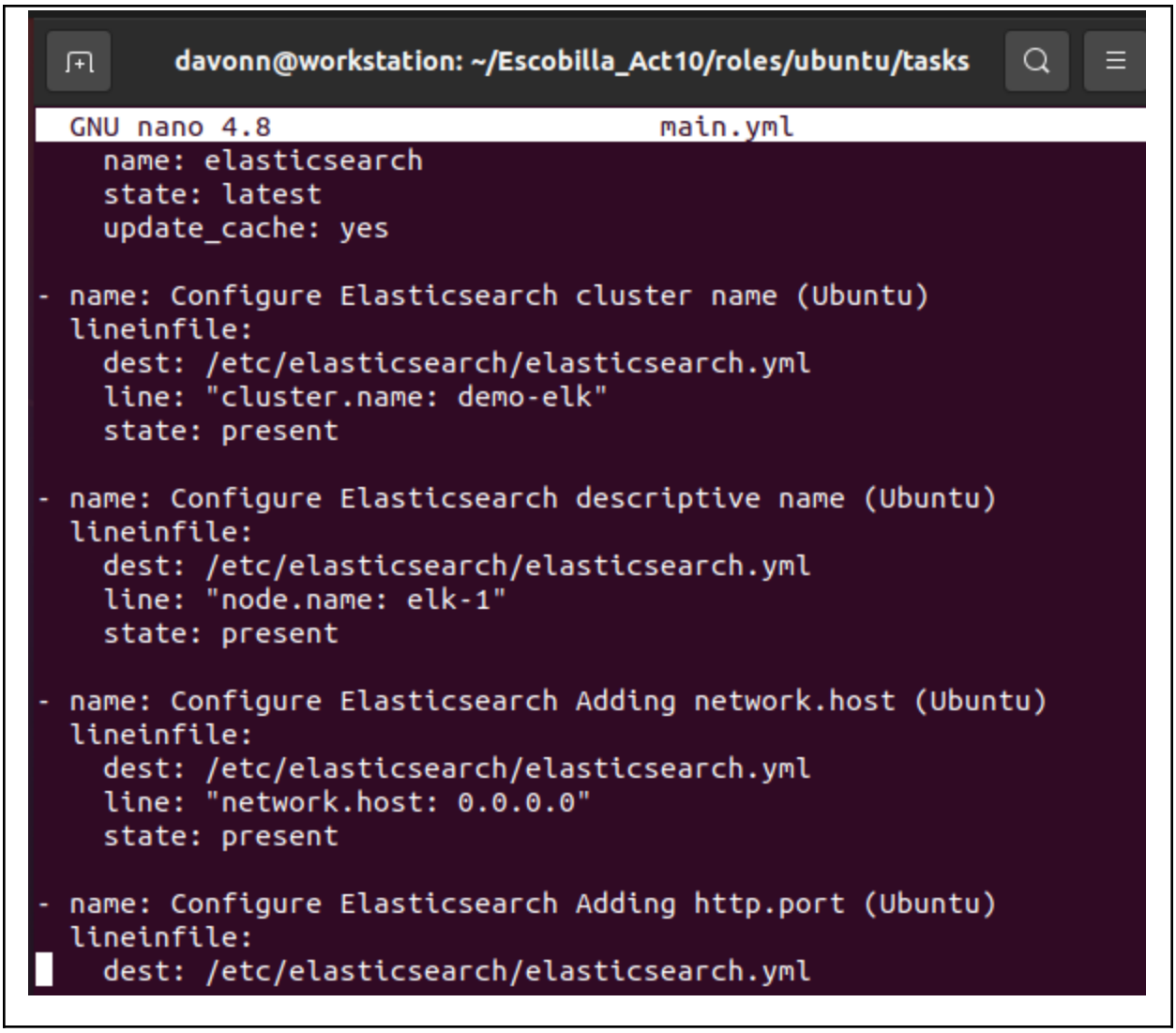
- hosts: all
become: true
pre_tasks:
- name: Update repository index (Ubuntu)
tags: always
apt:
update_cache: yes
changed_when: false
when: ansible_distribution == "Ubuntu"
- name: Update repository index (CentOS)
tags: always
yum:
update_cache: yes
changed_when: false
when: ansible_distribution == "CentOS"
- name: Install httpd on CentOS
tags: centos, apache, httpd
service:
name: httpd
state: started

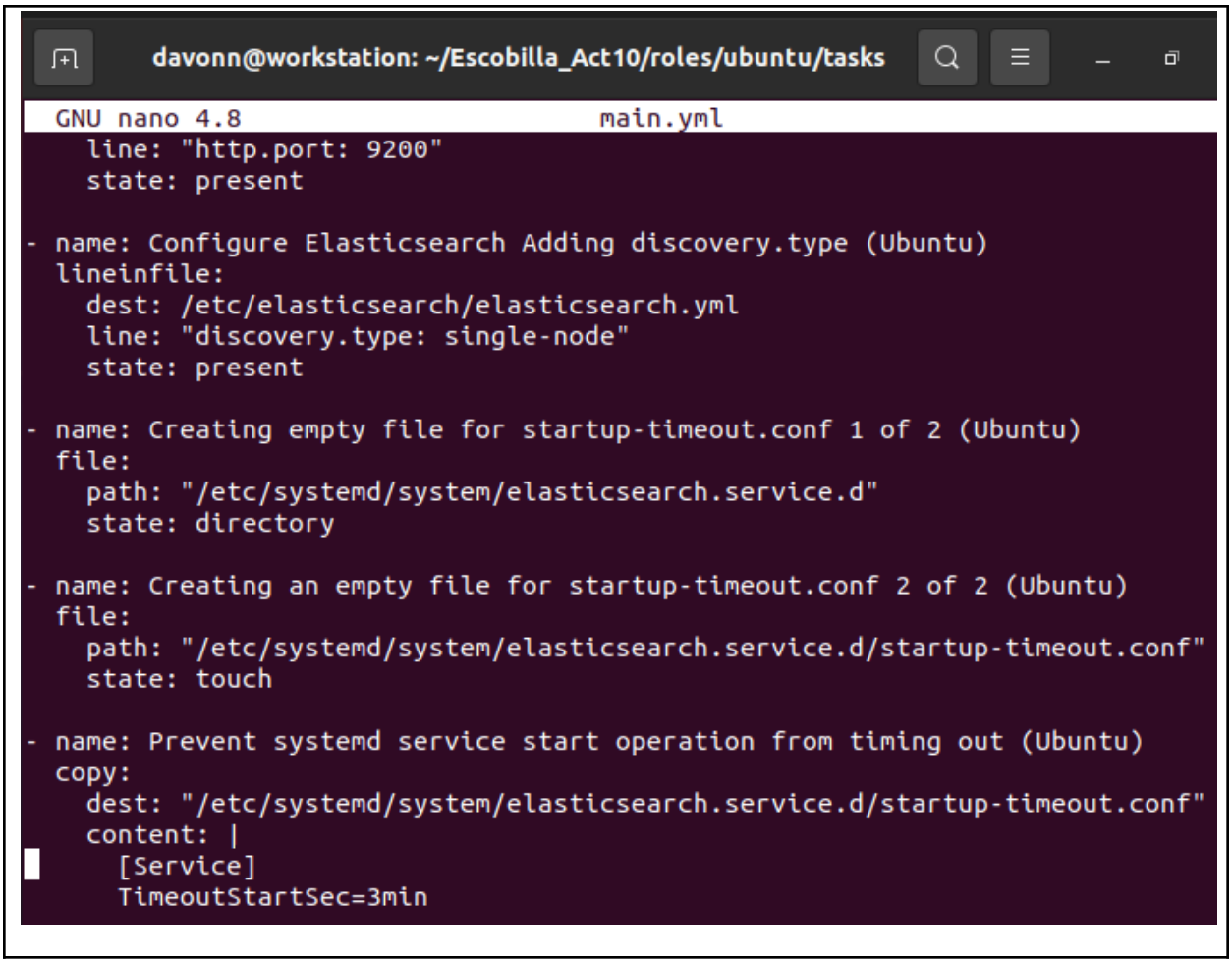


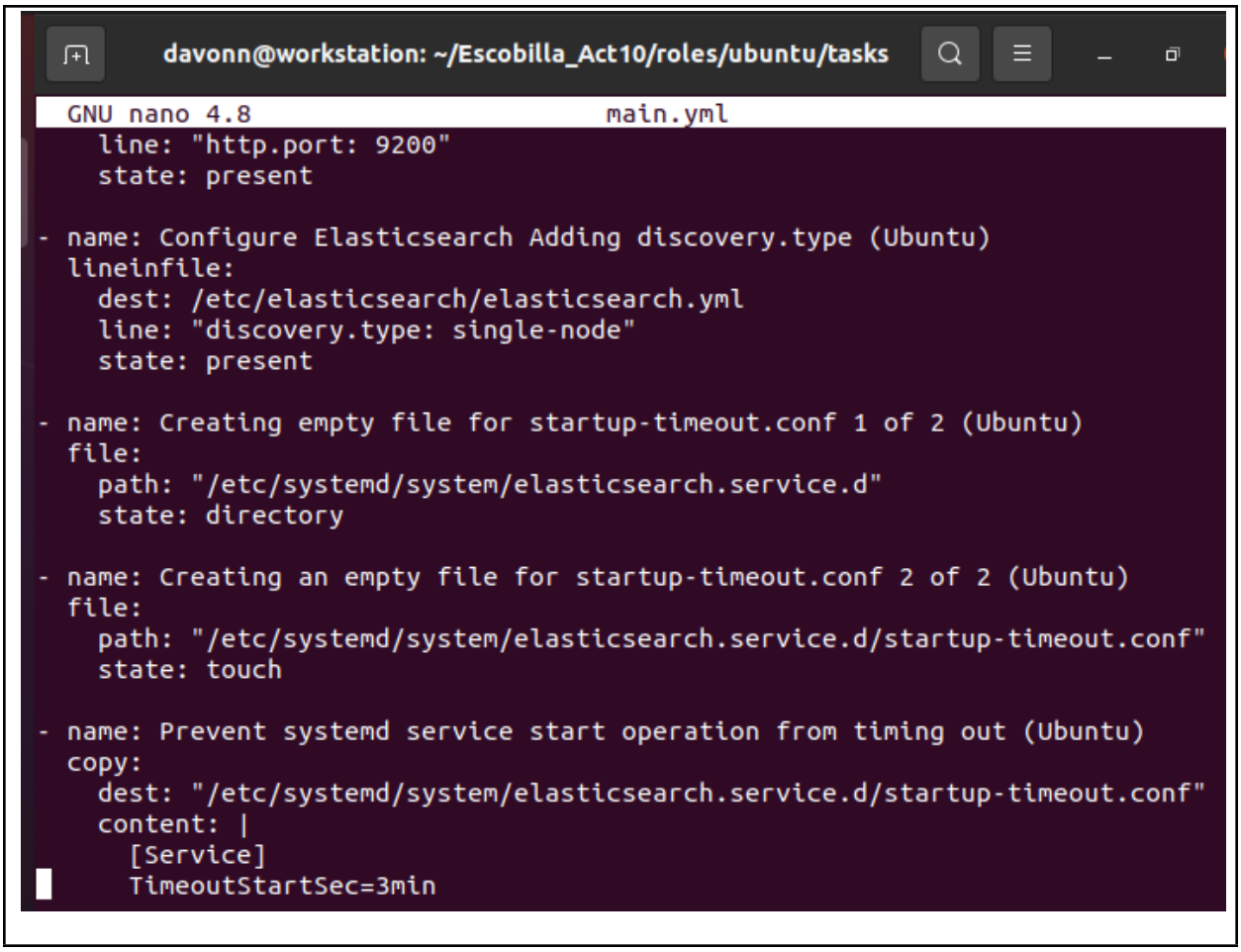
Fifth step is to create a role and directory for Ubuntu and CentOS. Under each directory create a main.yml that contains tasks to be executed.

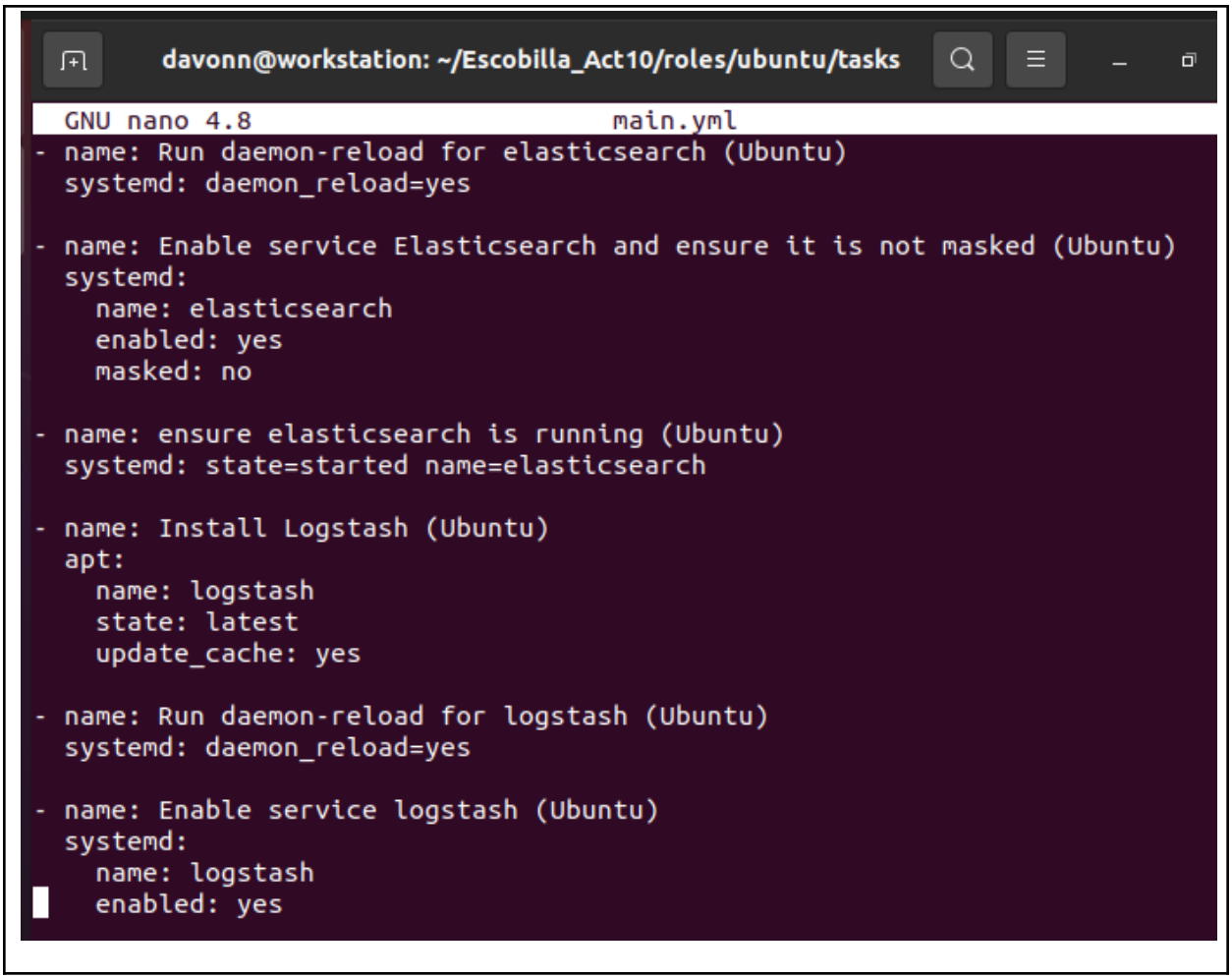
```
davonn@workstation:~/Escobilla_Act10$ mkdir roles
davonn@workstation:~/Escobilla_Act10$ cd roles
davonn@workstation:~/Escobilla_Act10/roles$ mkdir ubuntu
davonn@workstation:~/Escobilla_Act10/roles$ cd ubuntu
davonn@workstation:~/Escobilla_Act10/roles/ubuntu$ mkdir tasks
davonn@workstation:~/Escobilla_Act10/roles/ubuntu$ cd tasks
davonn@workstation:~/Escobilla_Act10/roles/ubuntu/tasks$ nano main.yml
```

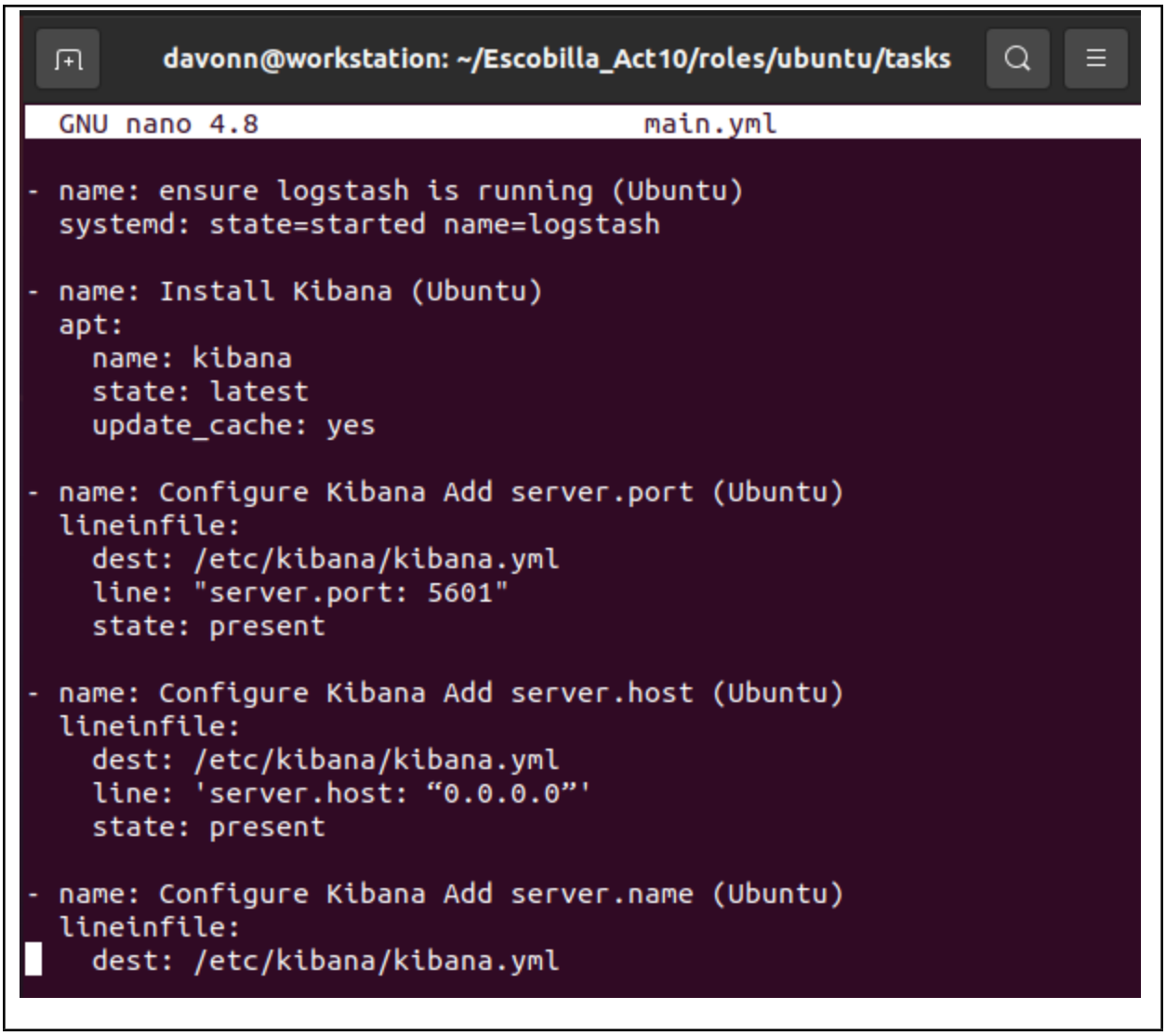

davonn@workstation: ~/Escobilla_Act10/roles/ubuntu/tasks
GNU nano 4.8 main.yml
- name: Install Elastic Dependencies (Ubuntu)
apt:
name:
- openjdk-11-jdk
- apt-transport-https
- curl
- gpgv
- gpgsm
- gnupg-l10n
- gnupg
- dirmngr
state: latest
- name: Get PGP Key (Ubuntu)
apt_key:
url: https://artifacts.elastic.co/GPG-KEY-elasticsearch
state: present
- name: Install Elasticsearch sources list (Ubuntu)
apt_repository:
repo: deb https://artifacts.elastic.co/packages/7.x/apt stable main
state: present
- name: Install Elasticsearch (Ubuntu)
apt:

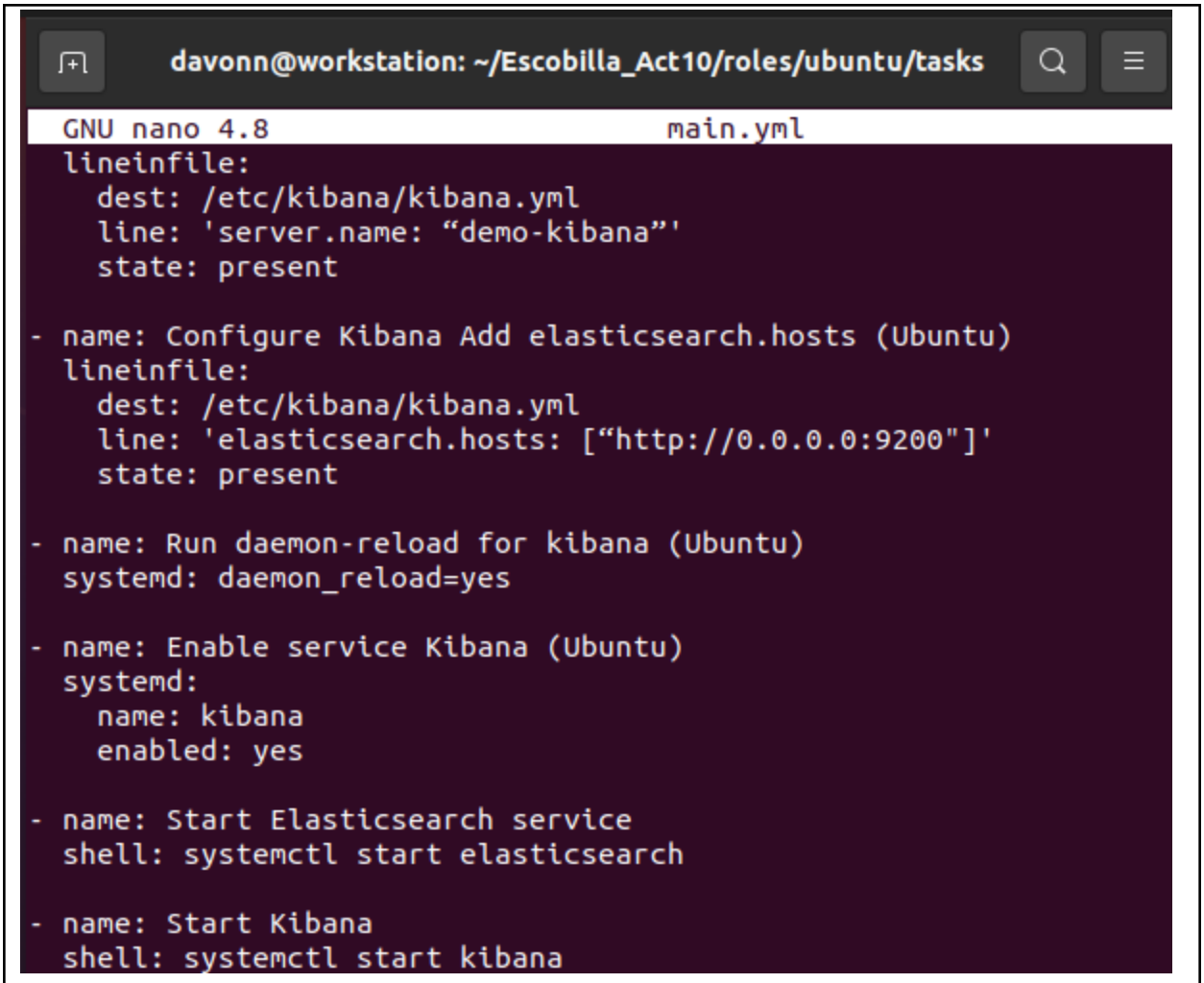

davonn@workstation: ~/Escobilla_Act10/roles/ubuntu/tasks
GNU nano 4.8 main.yml
name: elasticsearch
state: latest
update_cache: yes
- name: Configure Elasticsearch cluster name (Ubuntu)
lineinfile:
dest: /etc/elasticsearch/elasticsearch.yml
line: "cluster.name: demo-elk"
state: present
- name: Configure Elasticsearch descriptive name (Ubuntu)
lineinfile:
dest: /etc/elasticsearch/elasticsearch.yml
line: "node.name: elk-1"
state: present
- name: Configure Elasticsearch Adding network.host (Ubuntu)
lineinfile:
dest: /etc/elasticsearch/elasticsearch.yml
line: "network.host: 0.0.0.0"
state: present
- name: Configure Elasticsearch Adding http.port (Ubuntu)
lineinfile:
dest: /etc/elasticsearch/elasticsearch.yml


davonn@workstation: ~/Escobilla_Act10/roles/ubuntu/tasks
GNU nano 4.8 main.yml
line: "http.port: 9200"
state: present
- name: Configure Elasticsearch Adding discovery.type (Ubuntu)
lineinfile:
dest: /etc/elasticsearch/elasticsearch.yml
line: "discovery.type: single-node"
state: present
- name: Creating empty file for startup-timeout.conf 1 of 2 (Ubuntu)
file:
path: "/etc/systemd/system/elasticsearch.service.d"
state: directory
- name: Creating an empty file for startup-timeout.conf 2 of 2 (Ubuntu)
file:
path: "/etc/systemd/system/elasticsearch.service.d/startup-timeout.conf"
state: touch
- name: Prevent systemd service start operation from timing out (Ubuntu)
copy:
dest: "/etc/systemd/system/elasticsearch.service.d/startup-timeout.conf"
content: |
[Service]
TimeoutStartSec=3min


davonn@workstation: ~/Escobilla_Act10/roles/ubuntu/tasks
GNU nano 4.8 main.yml
line: "http.port: 9200"
state: present
- name: Configure Elasticsearch Adding discovery.type (Ubuntu)
lineinfile:
dest: /etc/elasticsearch/elasticsearch.yml
line: "discovery.type: single-node"
state: present
- name: Creating empty file for startup-timeout.conf 1 of 2 (Ubuntu)
file:
path: "/etc/systemd/system/elasticsearch.service.d"
state: directory
- name: Creating an empty file for startup-timeout.conf 2 of 2 (Ubuntu)
file:
path: "/etc/systemd/system/elasticsearch.service.d/startup-timeout.conf"
state: touch
- name: Prevent systemd service start operation from timing out (Ubuntu)
copy:
dest: "/etc/systemd/system/elasticsearch.service.d/startup-timeout.conf"
content: |
[Service]
TimeoutStartSec=3min

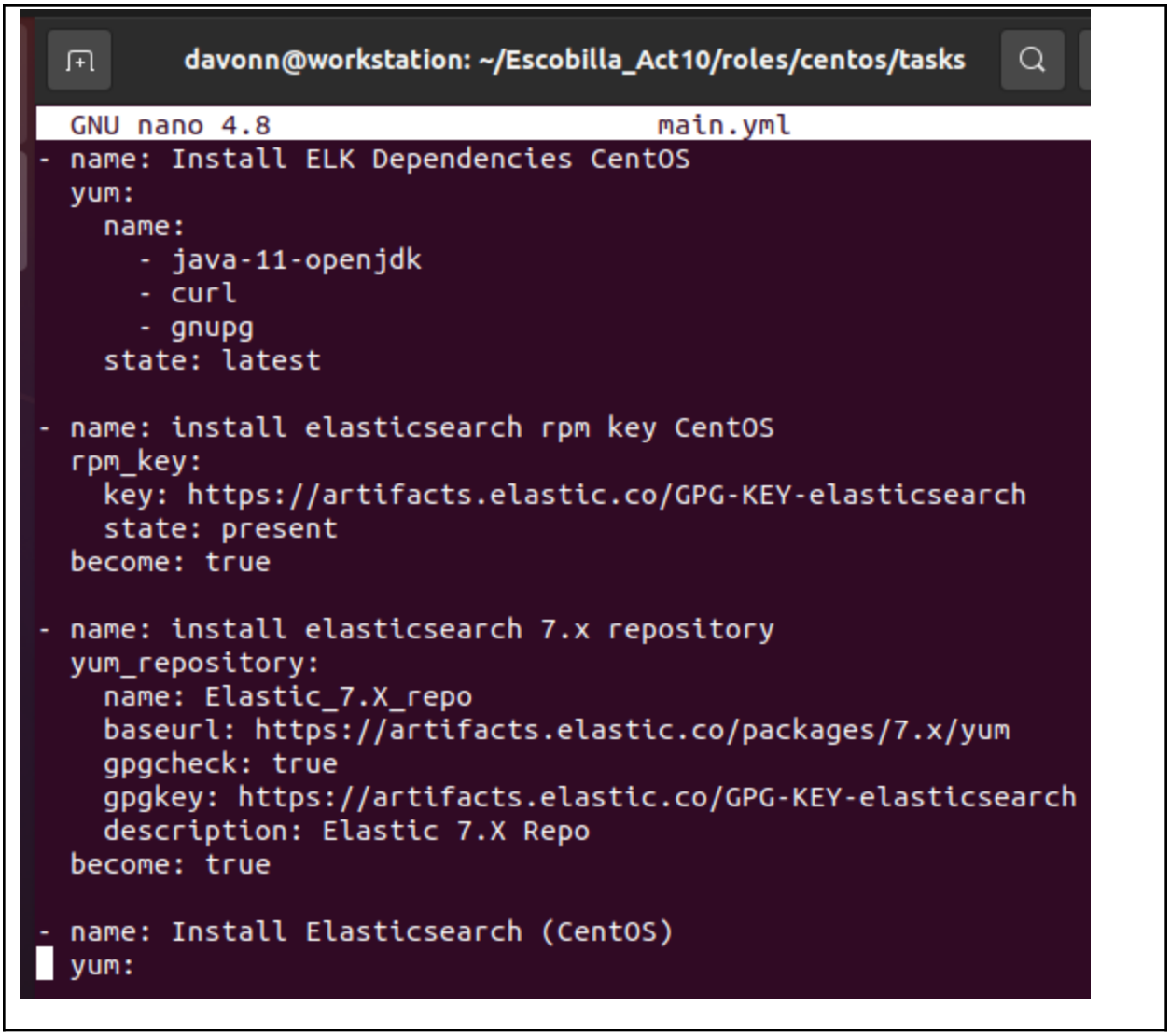

davonn@workstation: ~/Escobilla_Act10/roles/ubuntu/tasks
GNU nano 4.8 main.yml
- name: Run daemon-reload for elasticsearch (Ubuntu)
systemd: daemon_reload=yes
- name: Enable service Elasticsearch and ensure it is not masked (Ubuntu)
systemd:
name: elasticsearch
enabled: yes
masked: no
- name: ensure elasticsearch is running (Ubuntu)
systemd: state=started name=elasticsearch
- name: Install Logstash (Ubuntu)
apt:
name: logstash
state: latest
update_cache: yes
- name: Run daemon-reload for logstash (Ubuntu)
systemd: daemon_reload=yes
- name: Enable service logstash (Ubuntu)
systemd:
name: logstash
enabled: yes

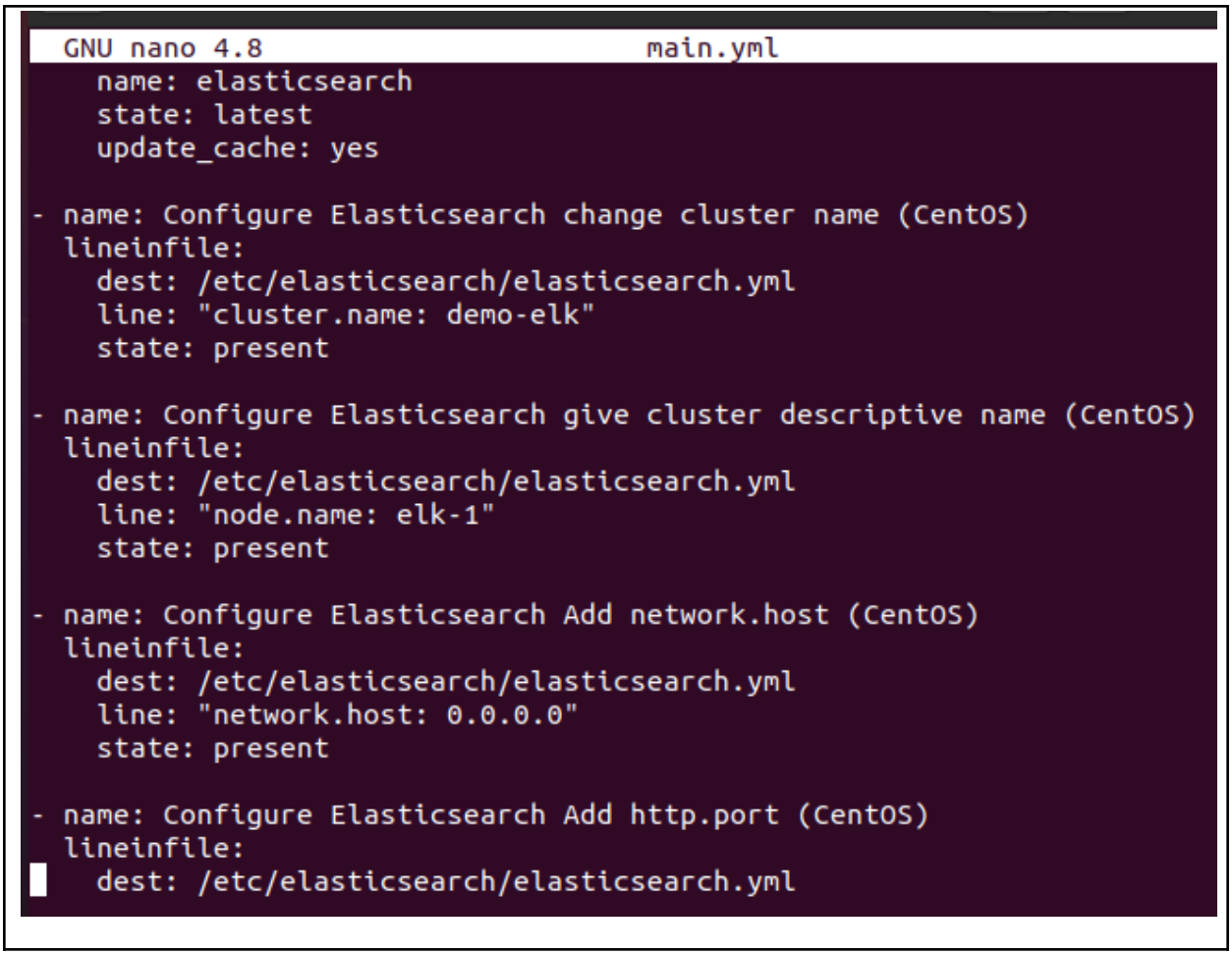

davonn@workstation: ~/Escobilla_Act10/roles/ubuntu/tasks
GNU nano 4.8 main.yml
- name: ensure logstash is running (Ubuntu)
systemd: state=started name=logstash
- name: Install Kibana (Ubuntu)
apt:
name: kibana
state: latest
update_cache: yes
- name: Configure Kibana Add server.port (Ubuntu)
lineinfile:
dest: /etc/kibana/kibana.yml
line: "server.port: 5601"
state: present
- name: Configure Kibana Add server.host (Ubuntu)
lineinfile:
dest: /etc/kibana/kibana.yml
line: 'server.host: “0.0.0.0”'
state: present
- name: Configure Kibana Add server.name (Ubuntu)
lineinfile:
dest: /etc/kibana/kibana.yml

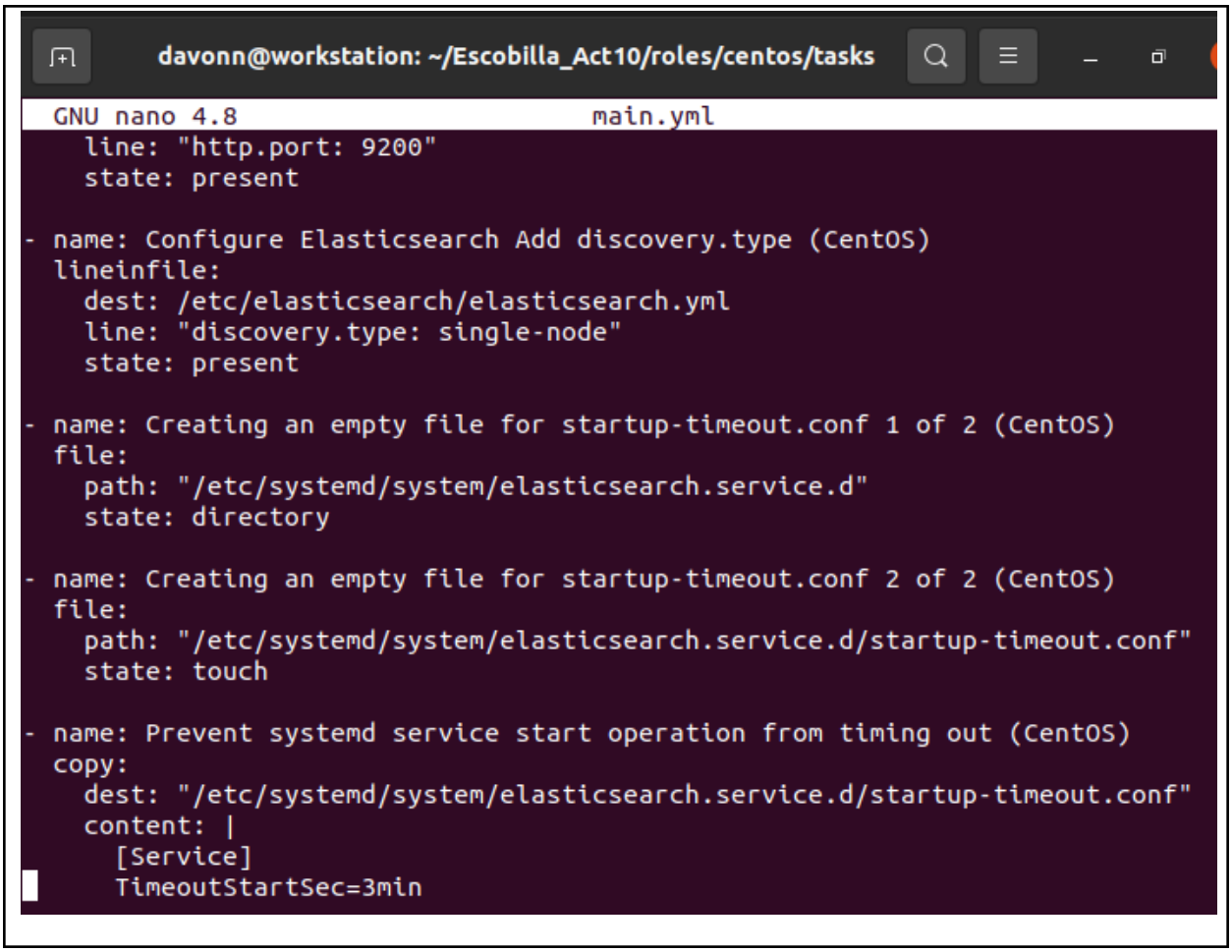


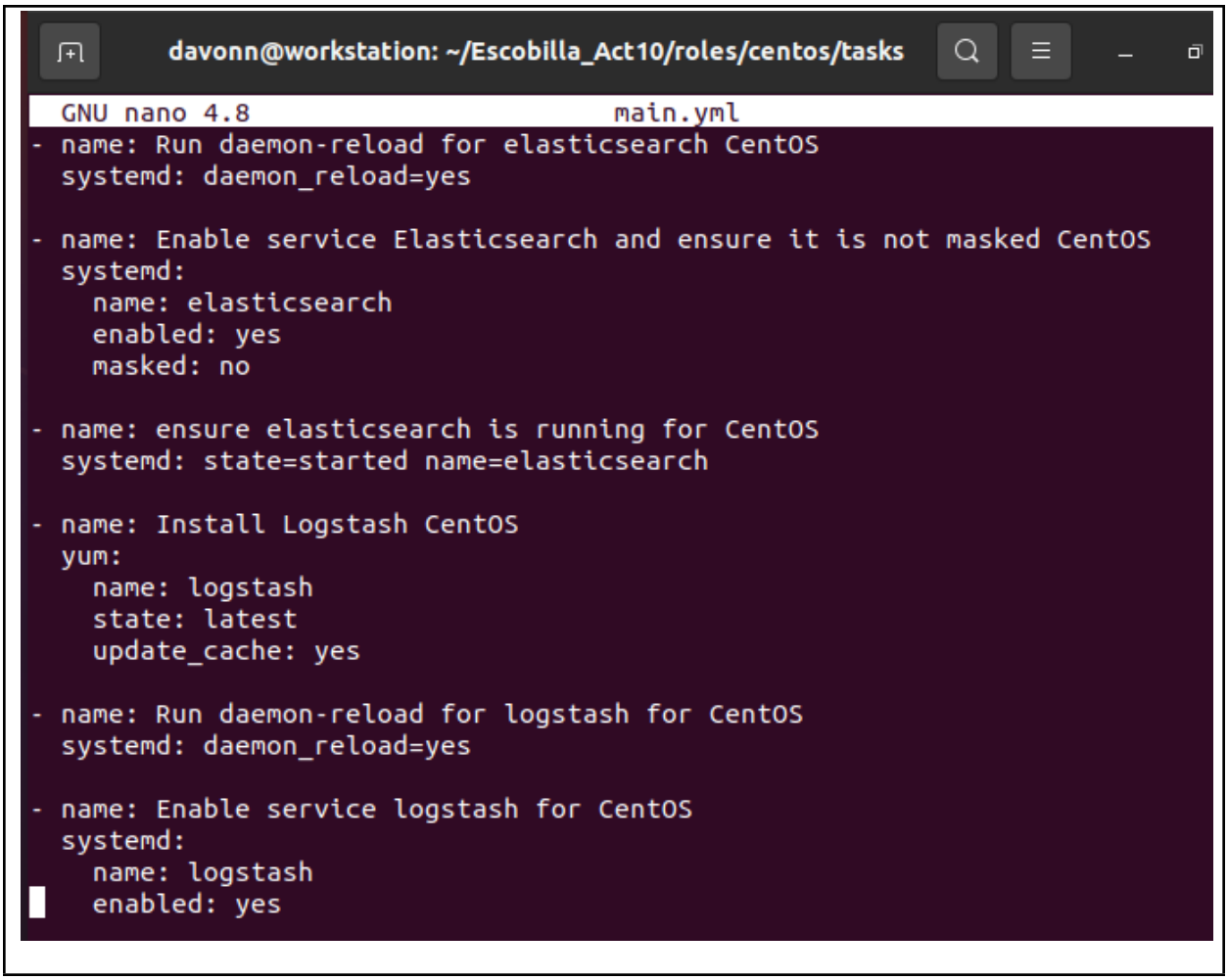
After creating main.yml for Ubuntu, create another for CentOS.

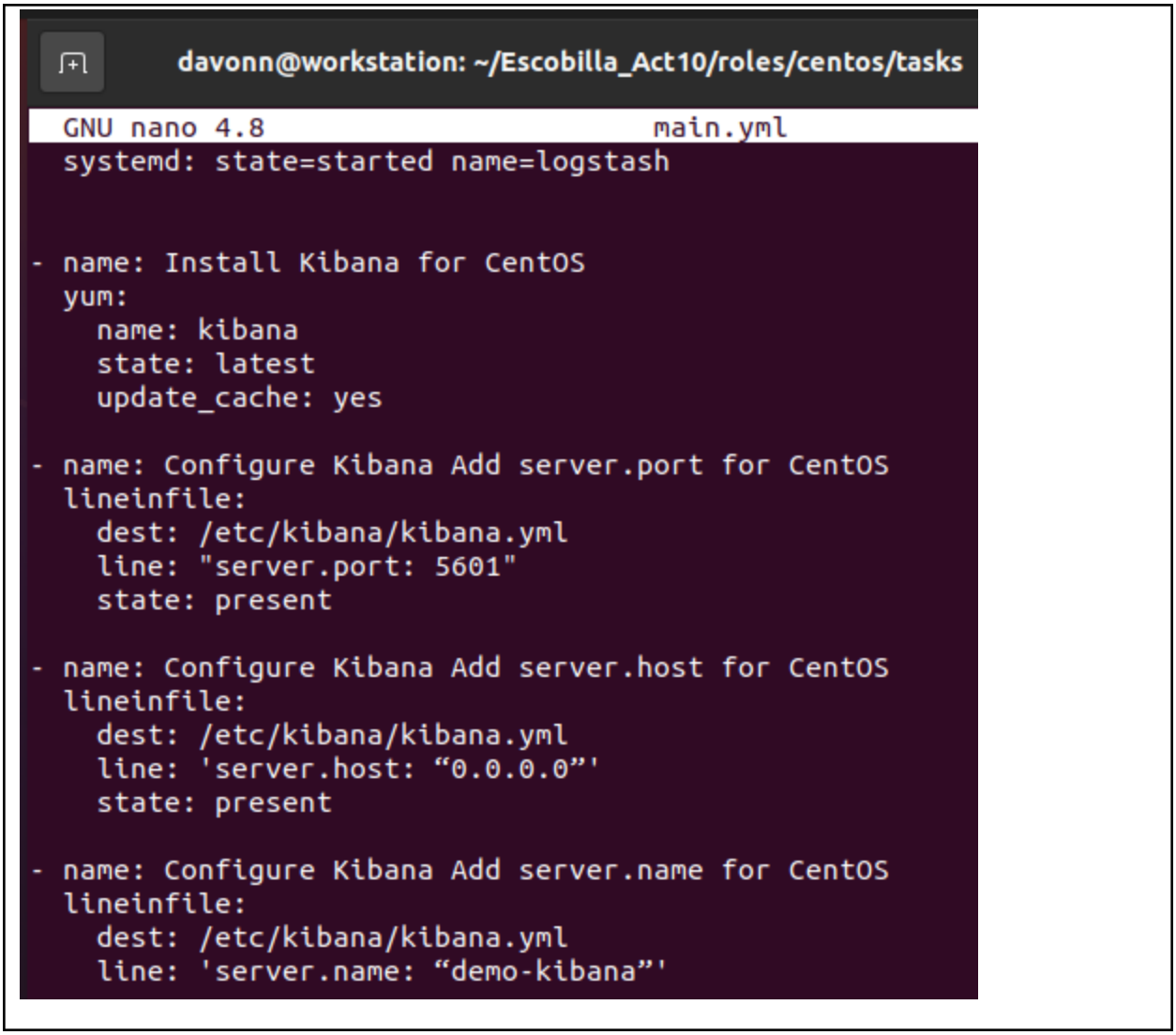
```
davonn@workstation:~/Escobilla_Act10/roles$ ls
centos  ubuntu
davonn@workstation:~/Escobilla_Act10/roles$ cd centos
davonn@workstation:~/Escobilla_Act10/roles/centos$ mkdir tasks
davonn@workstation:~/Escobilla_Act10/roles/centos$ cd tasks
davonn@workstation:~/Escobilla_Act10/roles/centos/tasks$ nano main.yml
```


davonn@workstation: ~/Escobilla_Act10/roles/centos/tasks
GNU nano 4.8 main.yml
- name: Install ELK Dependencies CentOS
yum:
name:
- java-11-openjdk
- curl
- gnupg
state: latest
- name: install elasticsearch rpm key CentOS
rpm_key:
key: https://artifacts.elastic.co/GPG-KEY-elasticsearch
state: present
become: true
- name: install elasticsearch 7.x repository
yum_repository:
name: Elastic_7.X_repo
baseurl: https://artifacts.elastic.co/packages/7.x/yum
gpgcheck: true
gpgkey: https://artifacts.elastic.co/GPG-KEY-elasticsearch
description: Elastic 7.X Repo
become: true
- name: Install Elasticsearch (CentOS)
yum:


GNU nano 4.8 main.yml
name: elasticsearch
state: latest
update_cache: yes
- name: Configure Elasticsearch change cluster name (CentOS)
lineinfile:
dest: /etc/elasticsearch/elasticsearch.yml
line: "cluster.name: demo-elk"
state: present
- name: Configure Elasticsearch give cluster descriptive name (CentOS)
lineinfile:
dest: /etc/elasticsearch/elasticsearch.yml
line: "node.name: elk-1"
state: present
- name: Configure Elasticsearch Add network.host (CentOS)
lineinfile:
dest: /etc/elasticsearch/elasticsearch.yml
line: "network.host: 0.0.0.0"
state: present
- name: Configure Elasticsearch Add http.port (CentOS)
lineinfile:
dest: /etc/elasticsearch/elasticsearch.yml


davonn@workstation: ~/Escobilla_Act10/roles/centos/tasks
GNU nano 4.8 main.yml
line: "http.port: 9200"
state: present
- name: Configure Elasticsearch Add discovery.type (CentOS)
lineinfile:
dest: /etc/elasticsearch/elasticsearch.yml
line: "discovery.type: single-node"
state: present
- name: Creating an empty file for startup-timeout.conf 1 of 2 (CentOS)
file:
path: "/etc/systemd/system/elasticsearch.service.d"
state: directory
- name: Creating an empty file for startup-timeout.conf 2 of 2 (CentOS)
file:
path: "/etc/systemd/system/elasticsearch.service.d/startup-timeout.conf"
state: touch
- name: Prevent systemd service start operation from timing out (CentOS)
copy:
dest: "/etc/systemd/system/elasticsearch.service.d/startup-timeout.conf"
content: |
[Service]
TimeoutStartSec=3min


davonn@workstation: ~/Escobilla_Act10/roles/centos/tasks
GNU nano 4.8 main.yml
- name: Run daemon-reload for elasticsearch CentOS
systemd: daemon_reload=yes
- name: Enable service Elasticsearch and ensure it is not masked CentOS
systemd:
name: elasticsearch
enabled: yes
masked: no
- name: ensure elasticsearch is running for CentOS
systemd: state=started name=elasticsearch
- name: Install Logstash CentOS
yum:
name: logstash
state: latest
update_cache: yes
- name: Run daemon-reload for logstash for CentOS
systemd: daemon_reload=yes
- name: Enable service logstash for CentOS
systemd:
name: logstash
enabled: yes

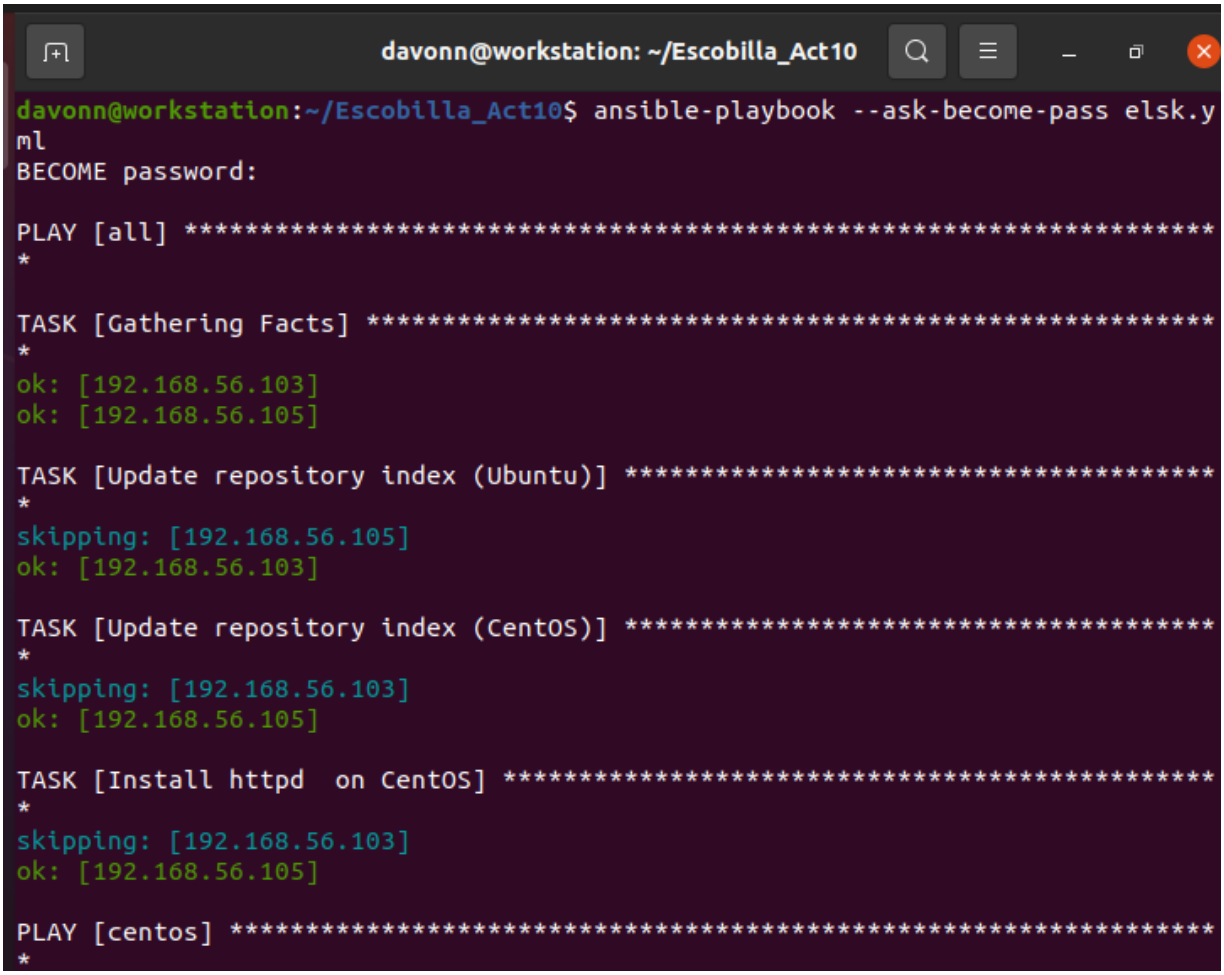

davonn@workstation: ~/Escobilla_Act10/roles/centos/tasks
GNU nano 4.8 main.yml
systemd: state=started name=logstash
- name: Install Kibana for CentOS
yum:
name: kibana
state: latest
update_cache: yes
- name: Configure Kibana Add server.port for CentOS
lineinfile:
dest: /etc/kibana/kibana.yml
line: "server.port: 5601"
state: present
- name: Configure Kibana Add server.host for CentOS
lineinfile:
dest: /etc/kibana/kibana.yml
line: 'server.host: “0.0.0.0”'
state: present
- name: Configure Kibana Add server.name for CentOS
lineinfile:
dest: /etc/kibana/kibana.yml
line: 'server.name: “demo-kibana”'

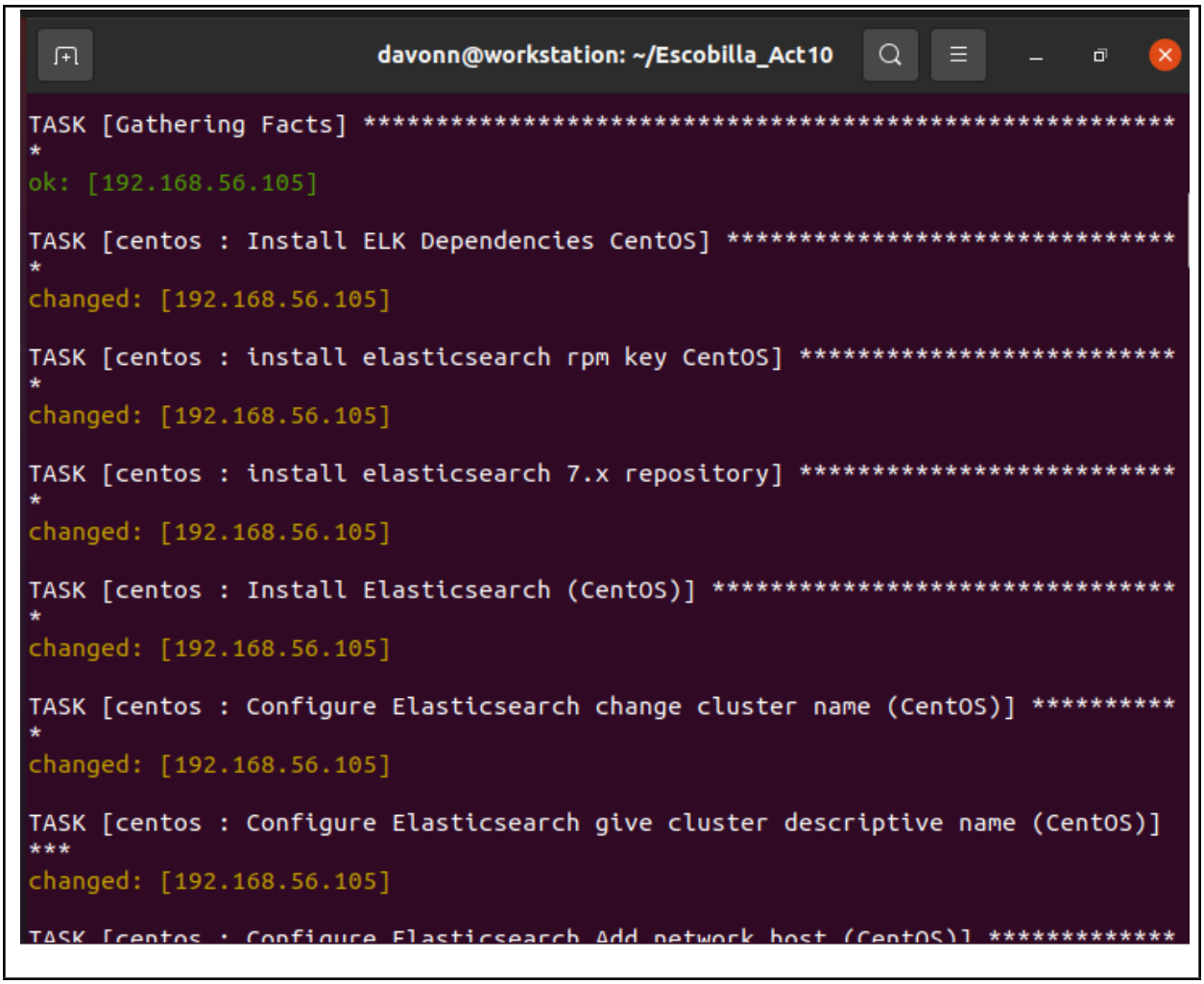
```
davonn@workstation: ~/Escobilla_Act10/roles/centos/tasks
 GNU nano 4.8                                main.yml
lineinfile:
  dest: /etc/kibana/kibana.yml
  line: 'server.name: “demo-kibana”'
  state: present

- name: Configure Kibana Add elasticsearch.hosts for CentOS
lineinfile:
  dest: /etc/kibana/kibana.yml
  line: 'elasticsearch.hosts: [“http://0.0.0.0:9200"]'
  state: present

- name: Run daemon-reload for kibana for CentOS
systemd: daemon_reload=yes

- name: Enable service Kibana for CentOS
systemd:
  name: kibana
  enabled: yes

- name: Start Elasticsearch for CentOS
shell: systemctl start elasticsearch

- name: Start Kibana for CentOS
shell: systemctl start kibana
```

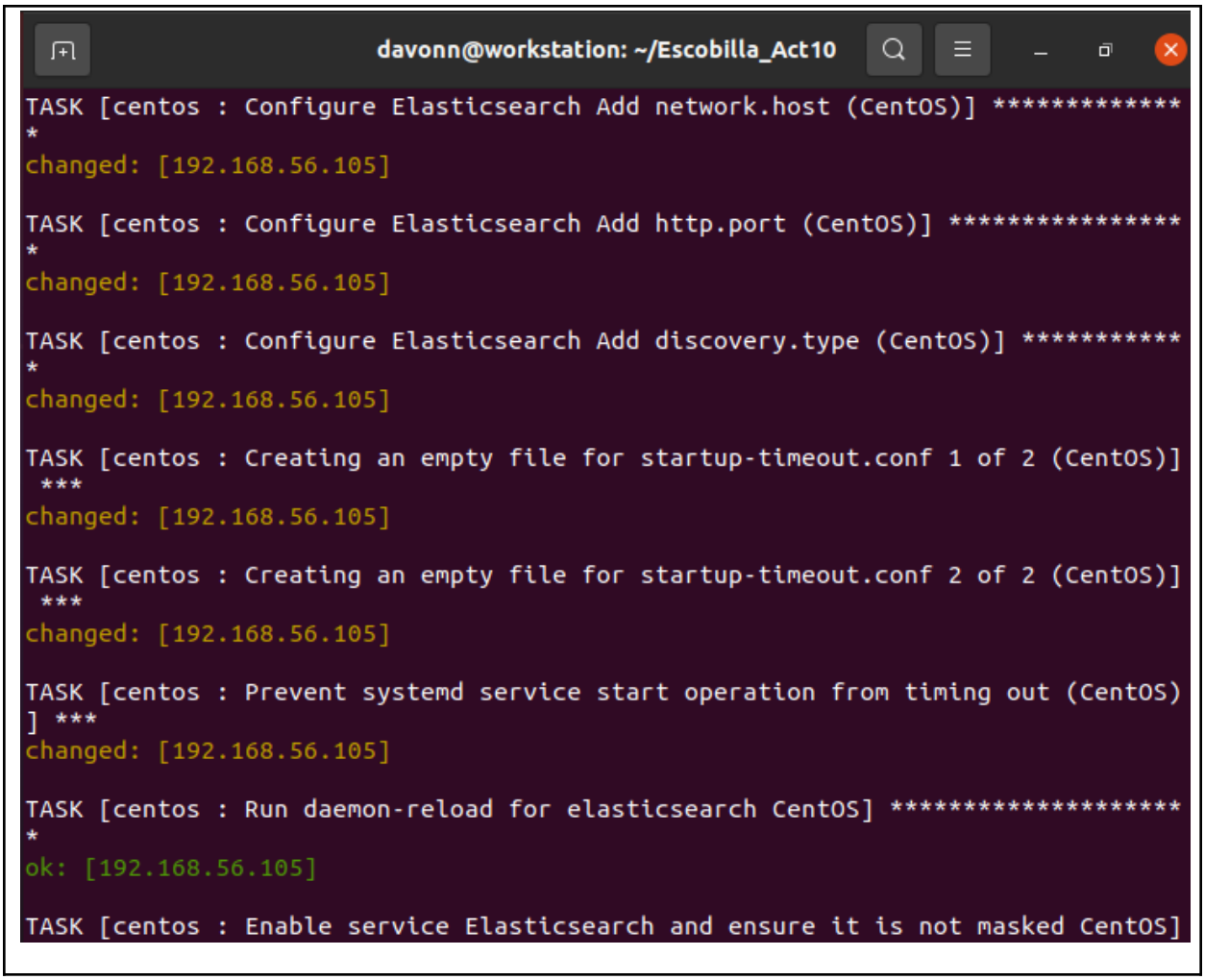
```
davonn@workstation:~/Escobilla_Act10$ tree
.
├── ansible.cfg
├── elsk.yml
├── inventory
└── roles
    ├── centos
    │   └── tasks
    │       └── main.yml
    └── ubuntu
        └── tasks
            └── main.yml

5 directories, 5 files
```

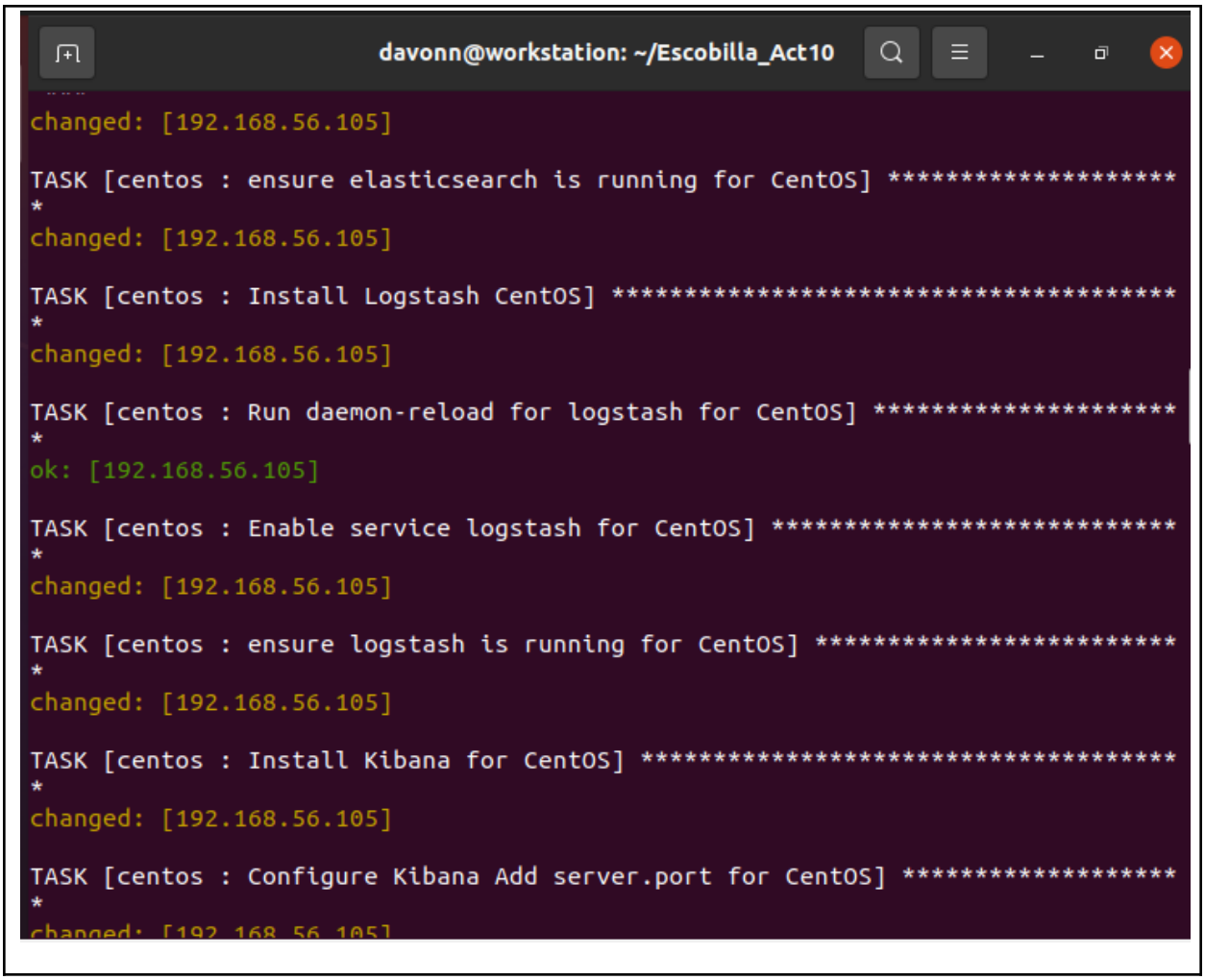
Verification for directories on roles using tree command.

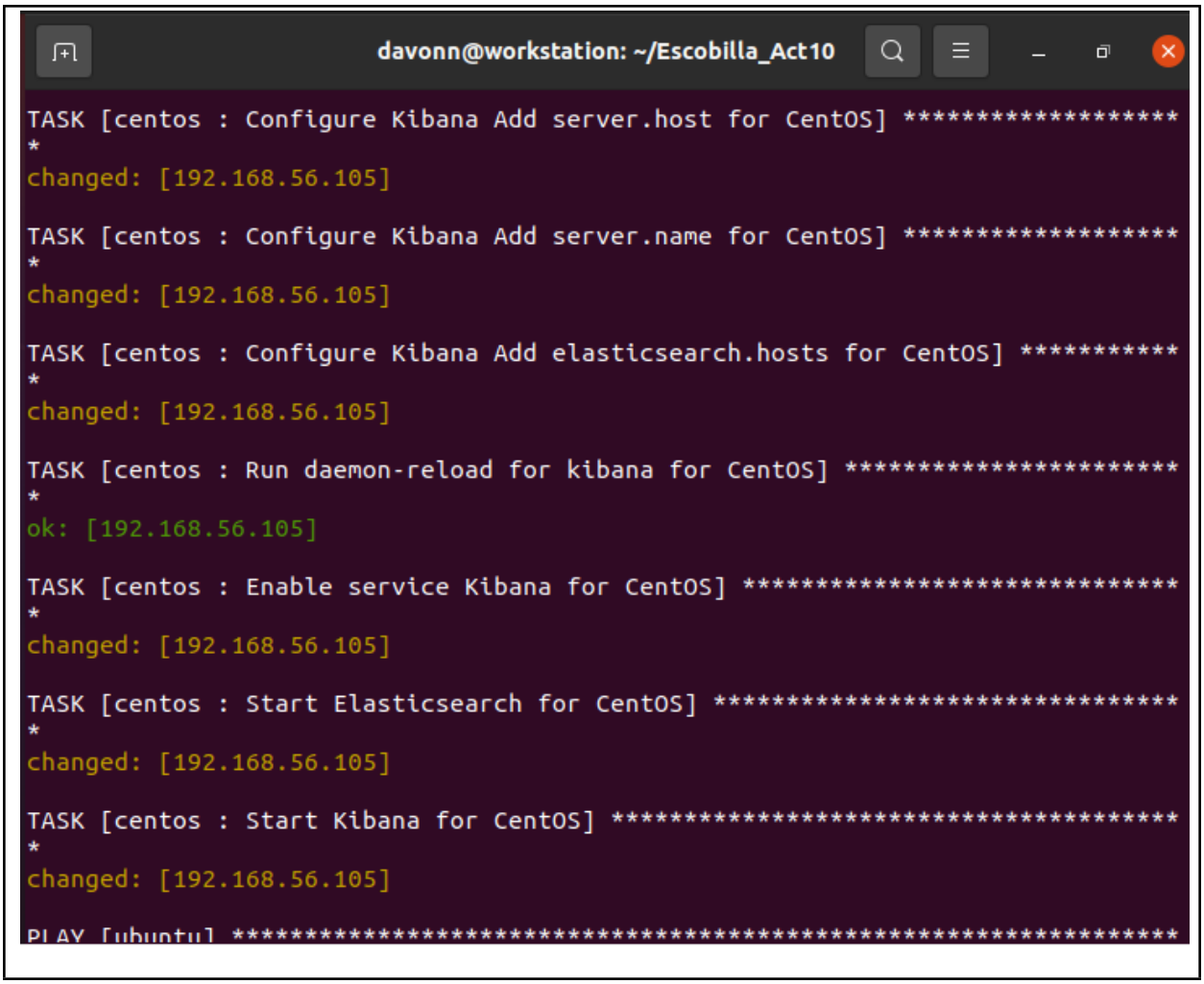
Sixth step, run the playbook and observe the output.

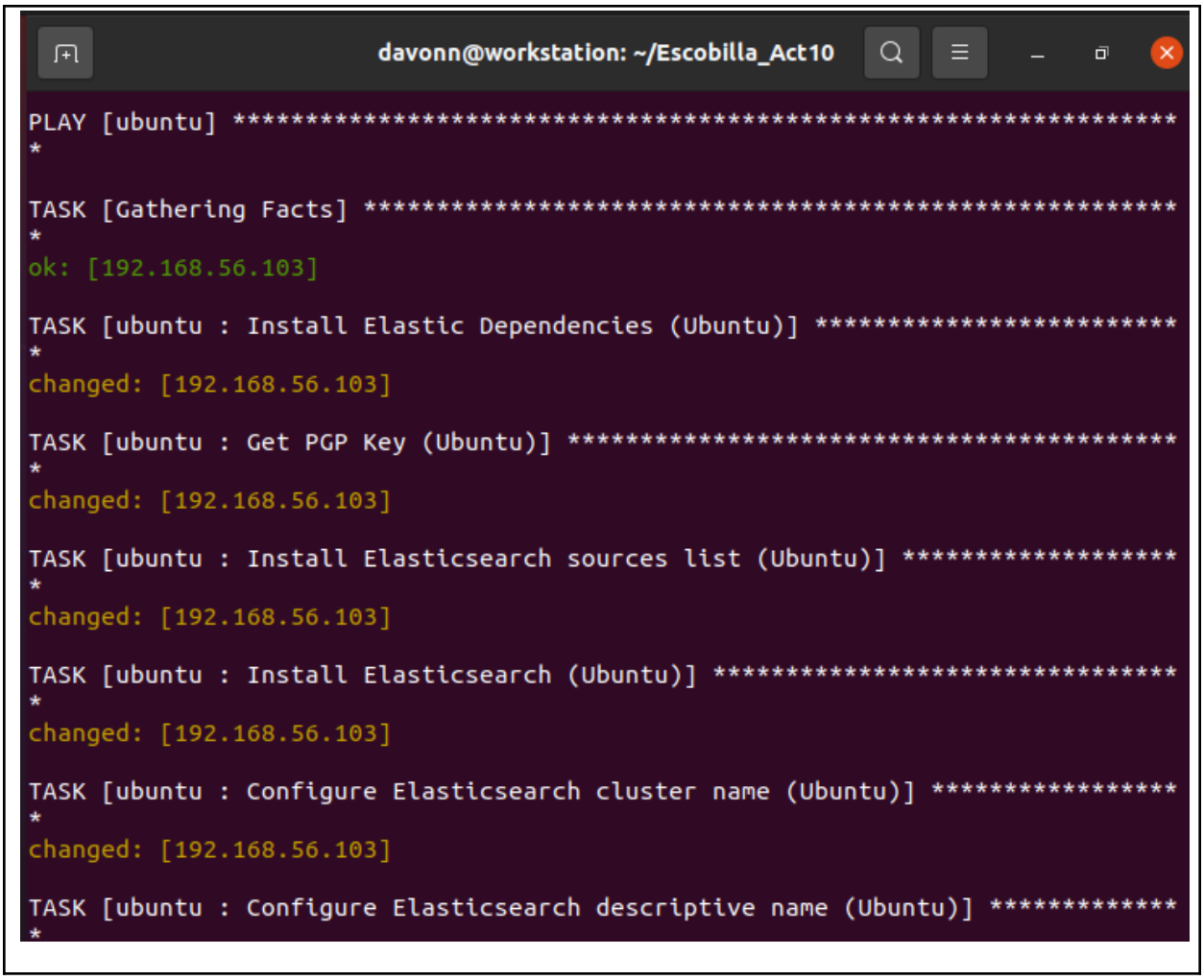


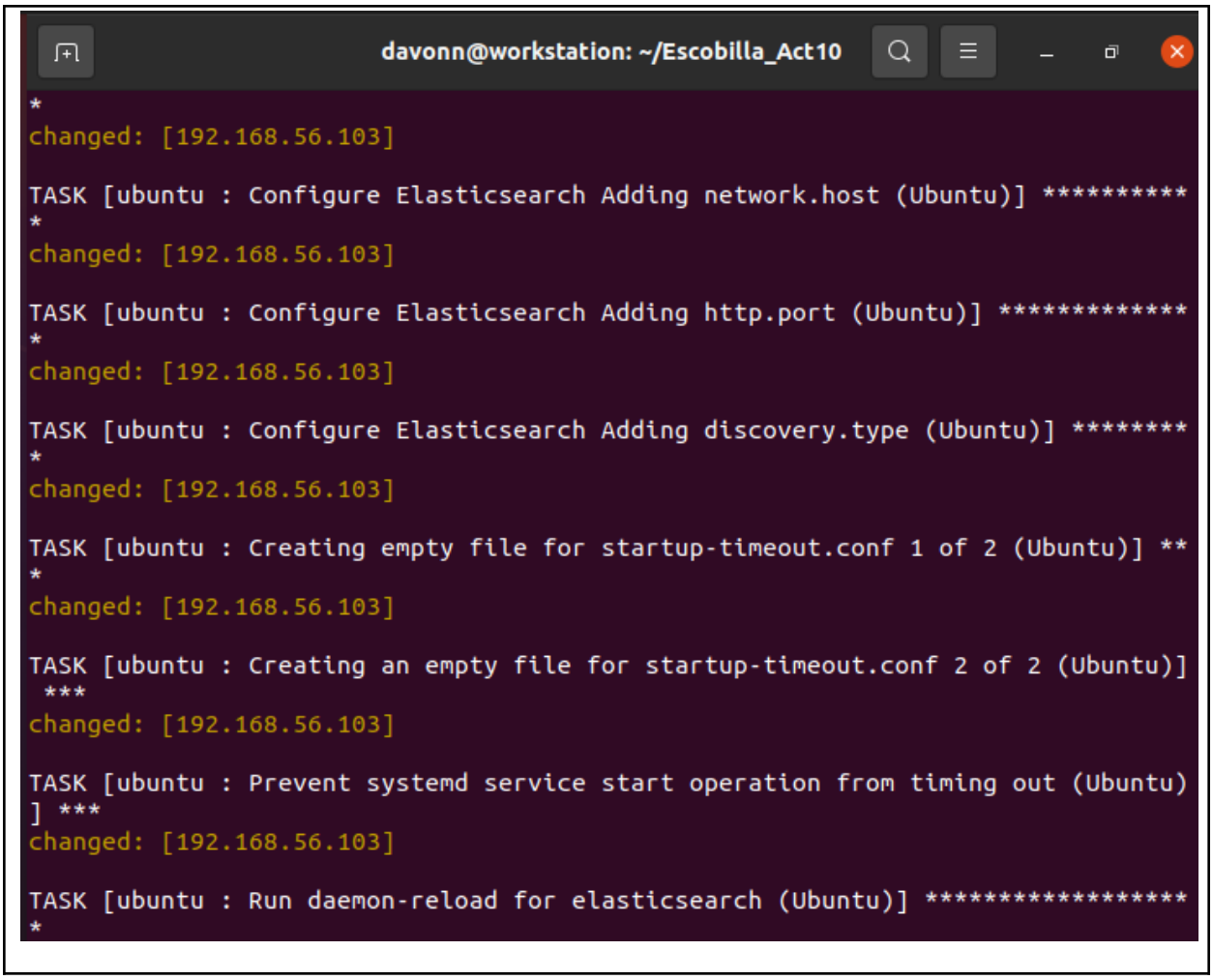

davonn@workstation: ~/Escobilla_Act10
TASK [Gathering Facts] ***
*
ok: [192.168.56.105]
TASK [centos : Install ELK Dependencies CentOS] ********************************
*
changed: [192.168.56.105]
TASK [centos : install elasticsearch rpm key CentOS] ***************************
*
changed: [192.168.56.105]
TASK [centos : install elasticsearch 7.x repository] ***************************
*
changed: [192.168.56.105]
TASK [centos : Install Elasticsearch (CentOS)] *********************************
*
changed: [192.168.56.105]
TASK [centos : Configure Elasticsearch change cluster name (CentOS)] ***********
*
changed: [192.168.56.105]
TASK [centos : Configure Elasticsearch give cluster descriptive name (CentOS)]
***
changed: [192.168.56.105]

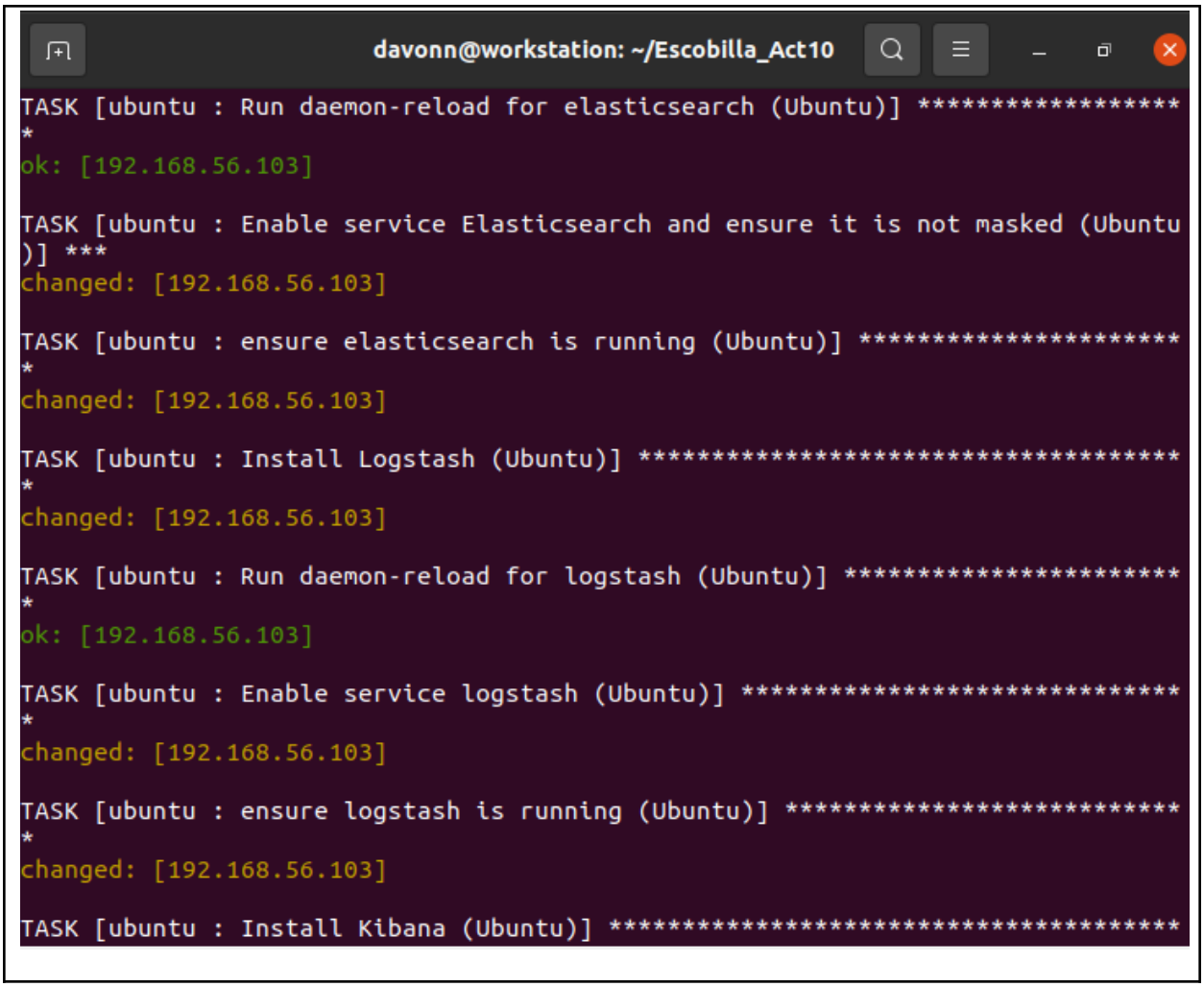

davonn@workstation: ~/Escobilla_Act10
TASK [centos : Configure Elasticsearch Add network.host (CentOS)] ***********
*
changed: [192.168.56.105]
TASK [centos : Configure Elasticsearch Add http.port (CentOS)] ***************
*
changed: [192.168.56.105]
TASK [centos : Configure Elasticsearch Add discovery.type (CentOS)] **********
*
changed: [192.168.56.105]
TASK [centos : Creating an empty file for startup-timeout.conf 1 of 2 (CentOS)]
***
changed: [192.168.56.105]
TASK [centos : Creating an empty file for startup-timeout.conf 2 of 2 (CentOS)]
***
changed: [192.168.56.105]
TASK [centos : Prevent systemd service start operation from timing out (CentOS)
] ***
changed: [192.168.56.105]
TASK [centos : Run daemon-reload for elasticsearch CentOS] ********************
*
ok: [192.168.56.105]
TASK [centos : Enable service Elasticsearch and ensure it is not masked CentOS]

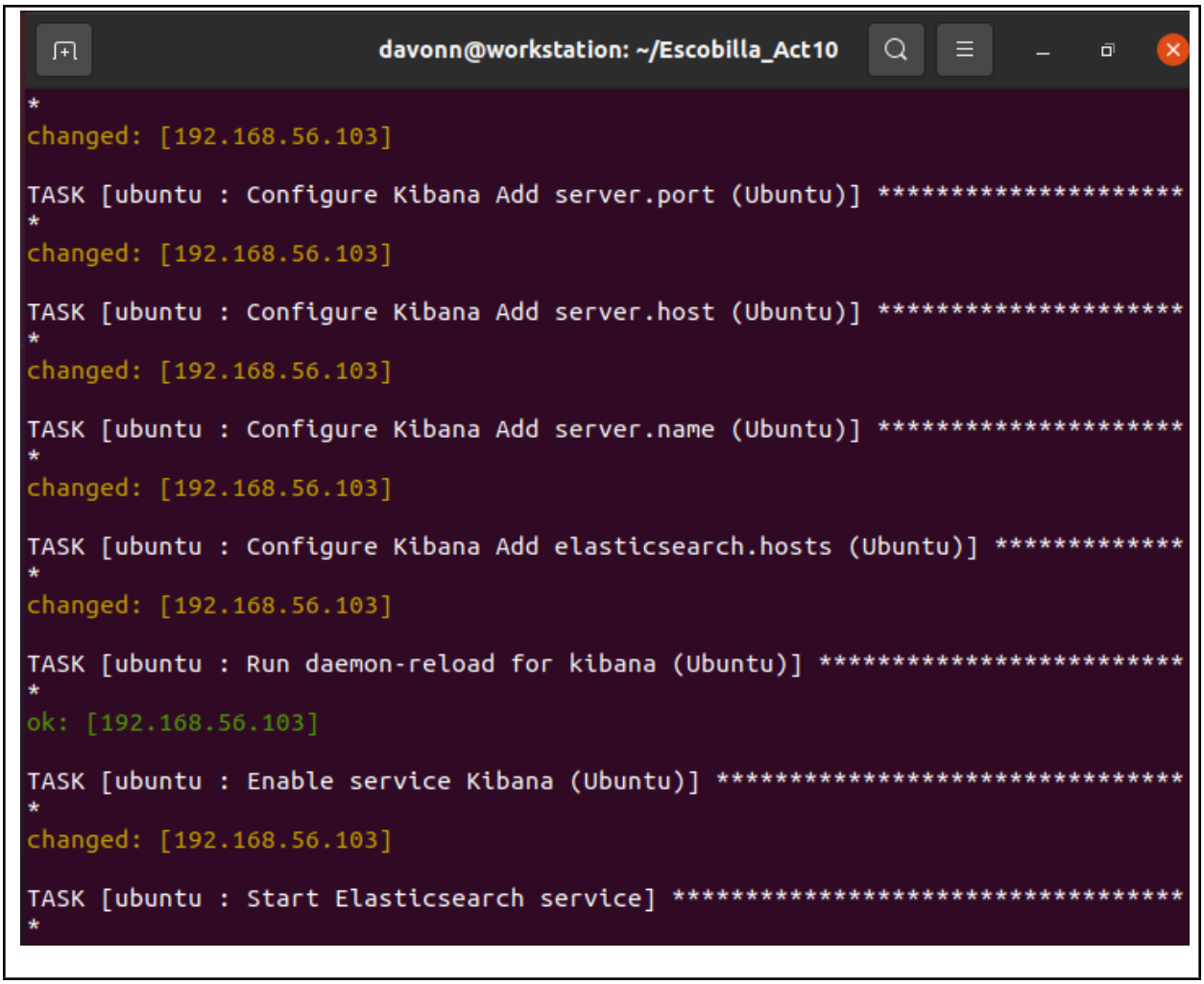

davonn@workstation: ~/Escobilla_Act10
changed: [192.168.56.105]
TASK [centos : ensure elasticsearch is running for CentOS] ********************
*
changed: [192.168.56.105]
TASK [centos : Install Logstash CentOS] ***************************************
*
changed: [192.168.56.105]
TASK [centos : Run daemon-reload for logstash for CentOS] **********************
*
ok: [192.168.56.105]
TASK [centos : Enable service logstash for CentOS] ****************************
*
changed: [192.168.56.105]
TASK [centos : ensure logstash is running for CentOS] *************************
*
changed: [192.168.56.105]
TASK [centos : Install Kibana for CentOS] *************************************
*
changed: [192.168.56.105]
TASK [centos : Configure Kibana Add server.port for CentOS] *******************
*

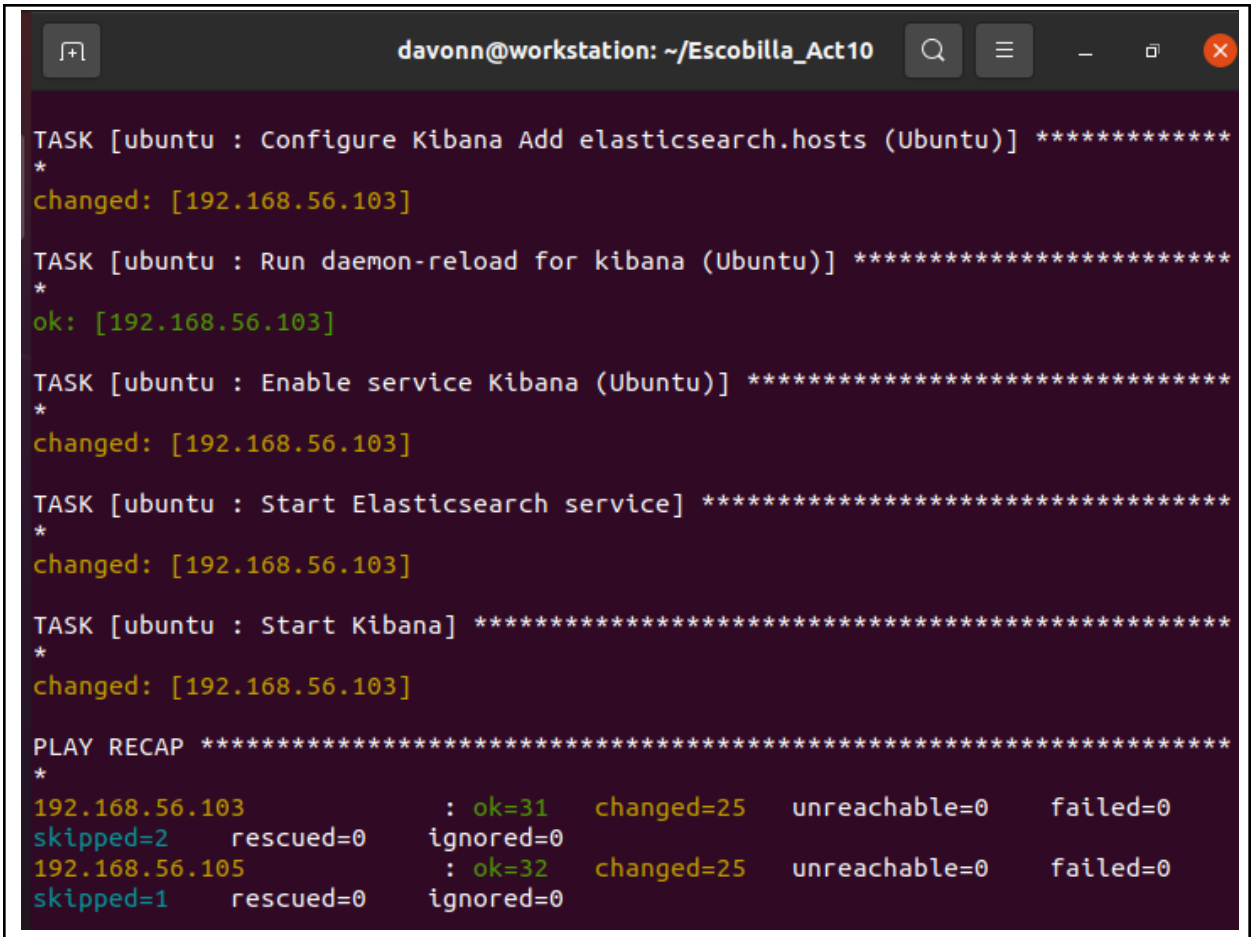

davonn@workstation: ~/Escobilla_Act10
TASK [centos : Configure Kibana Add server.host for CentOS] ********************
*
changed: [192.168.56.105]
TASK [centos : Configure Kibana Add server.name for CentOS] ********************
*
changed: [192.168.56.105]
TASK [centos : Configure Kibana Add elasticsearch.hosts for CentOS] ************
*
changed: [192.168.56.105]
TASK [centos : Run daemon-reload for kibana for CentOS] ************************
*
ok: [192.168.56.105]
TASK [centos : Enable service Kibana for CentOS] *******************************
*
changed: [192.168.56.105]
TASK [centos : Start Elasticsearch for CentOS] *********************************
*
changed: [192.168.56.105]
TASK [centos : Start Kibana for CentOS] **
*
changed: [192.168.56.105]
PLAY [ubuntu] **


davonn@workstation: ~/Escobilla_Act10
PLAY [ubuntu] ***
*
TASK [Gathering Facts] **
*
ok: [192.168.56.103]
TASK [ubuntu : Install Elastic Dependencies (Ubuntu)] *******************
*
changed: [192.168.56.103]
TASK [ubuntu : Get PGP Key (Ubuntu)] ************************************
*
changed: [192.168.56.103]
TASK [ubuntu : Install Elasticsearch sources list (Ubuntu)] *************
*
changed: [192.168.56.103]
TASK [ubuntu : Install Elasticsearch (Ubuntu)] **************************
*
changed: [192.168.56.103]
TASK [ubuntu : Configure Elasticsearch cluster name (Ubuntu)] ***********
*
changed: [192.168.56.103]
TASK [ubuntu : Configure Elasticsearch descriptive name (Ubuntu)] *******
*


davonn@workstation: ~/Escobilla_Act10
*
changed: [192.168.56.103]
TASK [ubuntu : Configure Elasticsearch Adding network.host (Ubuntu)] **********
*
changed: [192.168.56.103]
TASK [ubuntu : Configure Elasticsearch Adding http.port (Ubuntu)] *************
*
changed: [192.168.56.103]
TASK [ubuntu : Configure Elasticsearch Adding discovery.type (Ubuntu)] ********
*
changed: [192.168.56.103]
TASK [ubuntu : Creating empty file for startup-timeout.conf 1 of 2 (Ubuntu)] **
*
changed: [192.168.56.103]
TASK [ubuntu : Creating an empty file for startup-timeout.conf 2 of 2 (Ubuntu)]
***
changed: [192.168.56.103]
TASK [ubuntu : Prevent systemd service start operation from timing out (Ubuntu)
] ***
changed: [192.168.56.103]
TASK [ubuntu : Run daemon-reload for elasticsearch (Ubuntu)] ******************
*


davonn@workstation: ~/Escobilla_Act10
TASK [ubuntu : Run daemon-reload for elasticsearch (Ubuntu)] ******************
*
ok: [192.168.56.103]
TASK [ubuntu : Enable service Elasticsearch and ensure it is not masked (Ubuntu
)] ***
changed: [192.168.56.103]
TASK [ubuntu : ensure elasticsearch is running (Ubuntu)] **********************
*
changed: [192.168.56.103]
TASK [ubuntu : Install Logstash (Ubuntu)] *************************************
*
changed: [192.168.56.103]
TASK [ubuntu : Run daemon-reload for logstash (Ubuntu)] ***********************
*
ok: [192.168.56.103]
TASK [ubuntu : Enable service logstash (Ubuntu)] ******************************
*
changed: [192.168.56.103]
TASK [ubuntu : ensure logstash is running (Ubuntu)] ***************************
*
changed: [192.168.56.103]
TASK [ubuntu : Install Kibana (Ubuntu)] ***************************************

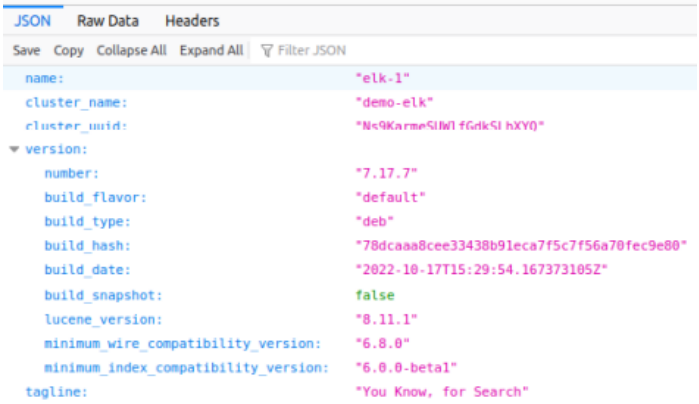

davonn@workstation: ~/Escobilla_Act10
*
changed: [192.168.56.103]
TASK [ubuntu : Configure Kibana Add server.port (Ubuntu)] **********
*
changed: [192.168.56.103]
TASK [ubuntu : Configure Kibana Add server.host (Ubuntu)] **********
*
changed: [192.168.56.103]
TASK [ubuntu : Configure Kibana Add server.name (Ubuntu)] **********
*
changed: [192.168.56.103]
TASK [ubuntu : Configure Kibana Add elasticsearch.hosts (Ubuntu)] *********
*
changed: [192.168.56.103]
TASK [ubuntu : Run daemon-reload for kibana (Ubuntu)] **********
*
ok: [192.168.56.103]
TASK [ubuntu : Enable service Kibana (Ubuntu)] **********
*
changed: [192.168.56.103]
TASK [ubuntu : Start Elasticsearch service] **********
*



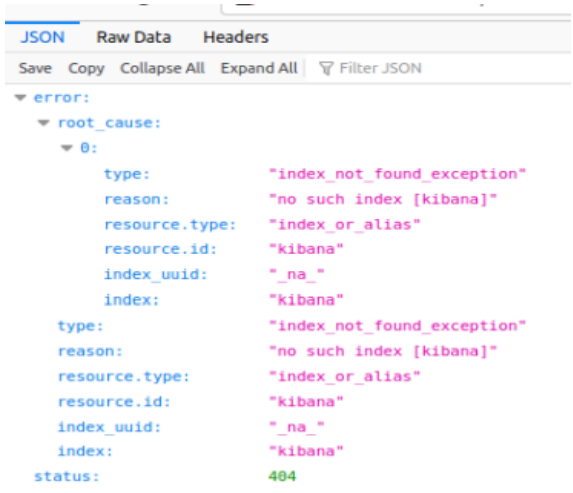
The playbook ran smoothly and successfully executed all the tasks without error. The last procedure is to check all the installed programs on control nodes.

OUTPUT ON UBUNTU:

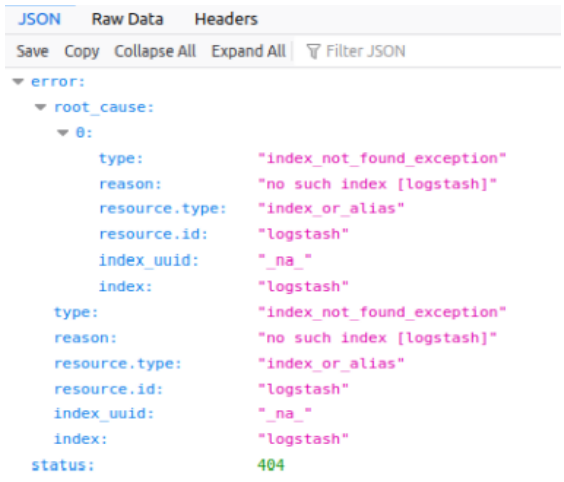
Elastic Search



Kibana

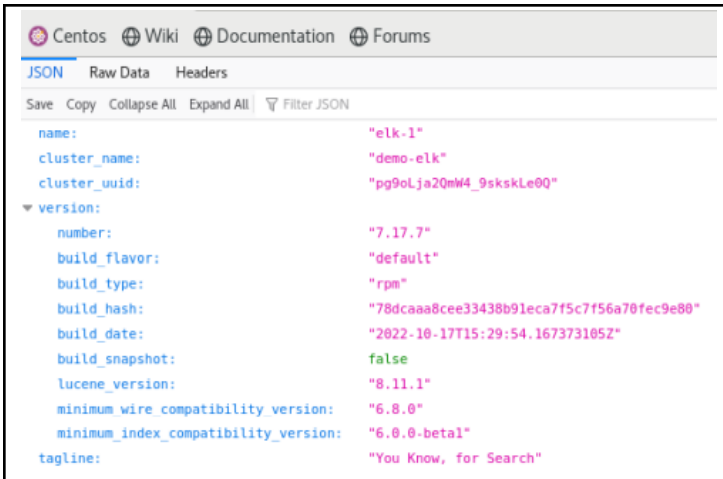


Logstash

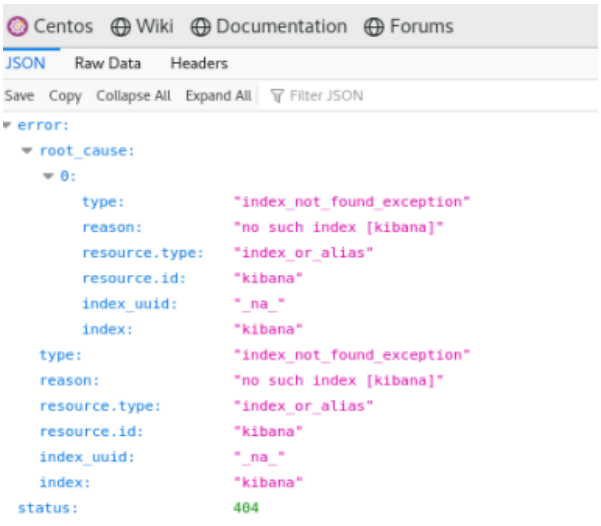


OUTPUT ON CENTOS:

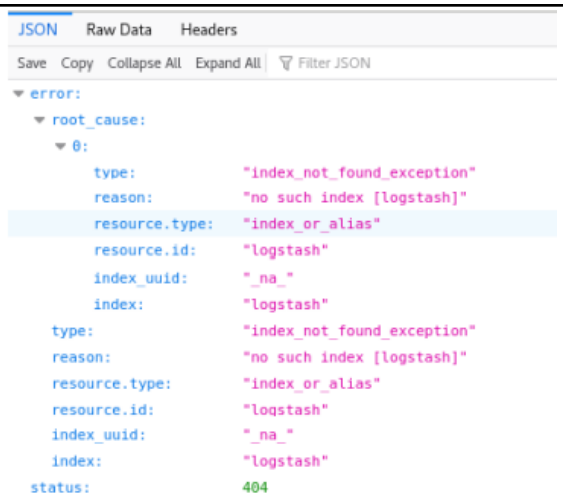
Elastic Search



Kibana



Logstash



All of the output from each control node has successfully appeared, meaning that the installation is finalized.

Do not forget to save the work on the repository by executing the commands.

```
davonn@workstation:~/Escobilla_Act10$ git add -A
davonn@workstation:~/Escobilla_Act10$ git commit -m "Manage Log"
[master (root-commit) b37f0f6] Manage Log
 5 files changed, 340 insertions(+)
 create mode 100644 ansible.cfg
 create mode 100644 elsk.yml
 create mode 100644 inventory
 create mode 100644 roles/centos/tasks/main.yml
 create mode 100644 roles/ubuntu/tasks/main.yml
davonn@workstation:~/Escobilla_Act10$ git push
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Compressing objects: 100% (7/7), done.
Writing objects: 100% (12/12), 2.26 KiB | 772.00 KiB/s, done.
Total 12 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), done.
To github.com:DavonnEscobilla/Escobilla_Act10.git
 * [new branch]      master -> master
```

Reflections:

Answer the following:

1. What are the benefits of having a log monitoring tool?
   Having a log monitoring tool saves time to access the scanned and log files from the servers, application, and networks. It can be used to inform the users about

the log files. It can be also used to properly handle security problems and performance issues. Upon detecting the problems, it can mitigate the downtime and risks.

Conclusions:

Upon performing the activity, I have difficulties finding references to properly handle the tasks implemented on each server. The playbook became very long since there are a lot of tasks that need to be implemented. Also, the installation on the control nodes takes a lot of time to finish since my computer is not that strong. Even so, I have managed to properly perform the activity and get the desired output.