Name: Frias, Abegail L.	**Date Performed: Nov. 11, 2024**
Course/Section: CPE212 - CPE31S21	**Date Submitted: Nov. 12, 2024**
Instructor: Engr. Robin Valenzuela	**Semester and SY: 1st Sem/2024-2025**

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

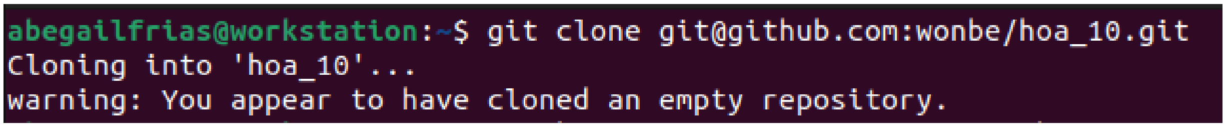
3. Tasks

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.

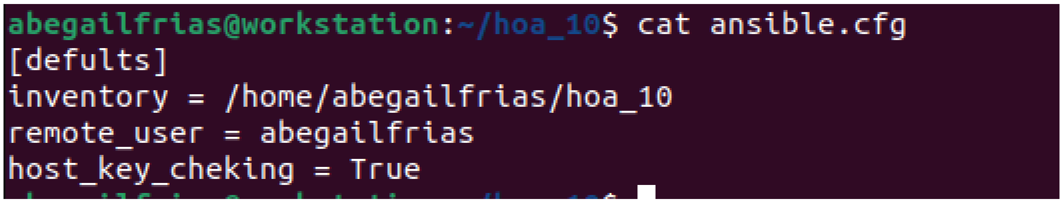
4. Output (screenshots and explanations)

First Step: Establishing a Repository

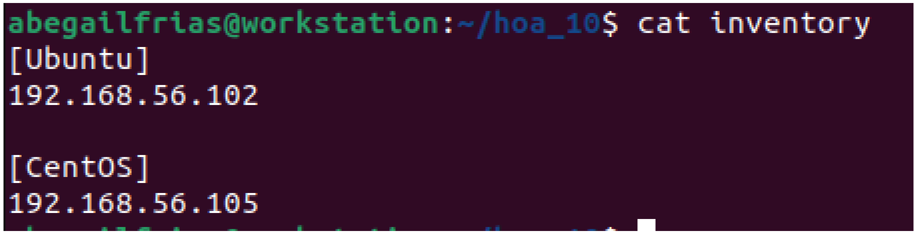
GitHub is a great tool for cloud-based work organization and storage. Create a roles directory with the roles required for particular tasks, an inventory file, and an `ansible.cfg` for this task.



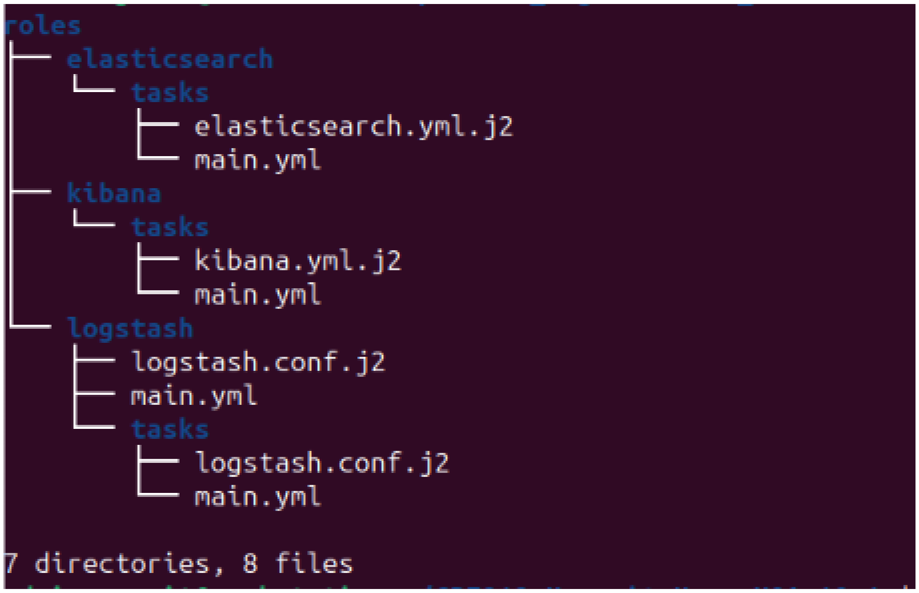
```
abegailfrias@workstation:~$ git clone git@github.com:wonbe/hoa_10.git
Cloning into 'hoa_10'...
warning: You appear to have cloned an empty repository.
```



```
abegailfrias@workstation:~/hoa_10$ cat ansible.cfg
[defults]
inventory = /home/abegailfrias/hoa_10
remote_user = abegailfrias
host_key_cheking = True
```



```
abegailfrias@workstation:~/hoa_10$ cat inventory
[Ubuntu]
192.168.56.102

[CentOS]
192.168.56.105
```

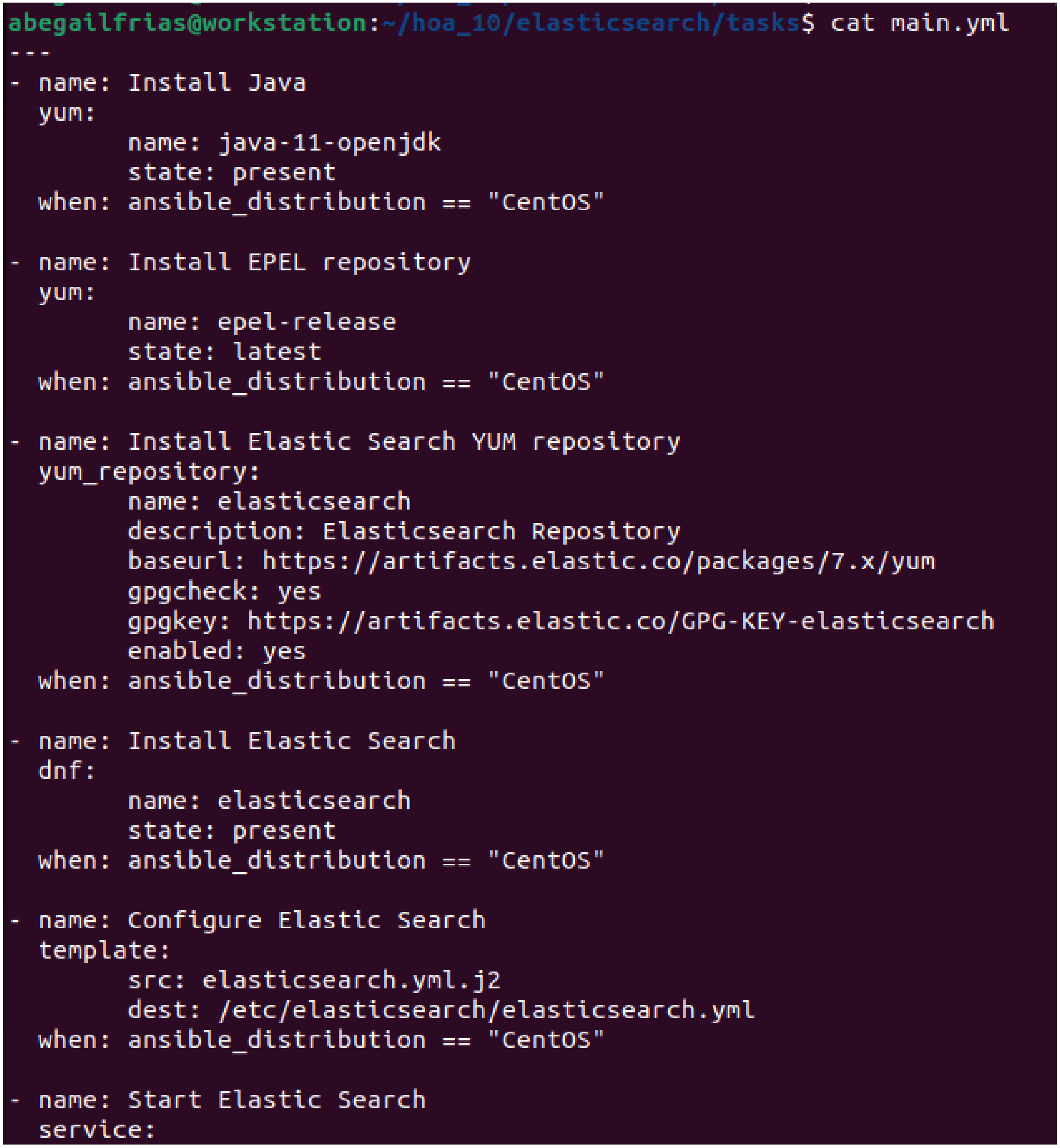


```
roles
├── elasticsearch
│   └── tasks
│       ├── elasticsearch.yml.j2
│       └── main.yml
├── kibana
│   └── tasks
│       ├── kibana.yml.j2
│       └── main.yml
└── logstash
    ├── logstash.conf.j2
    ├── main.yml
    └── tasks
        ├── logstash.conf.j2
        └── main.yml

7 directories, 8 files
```

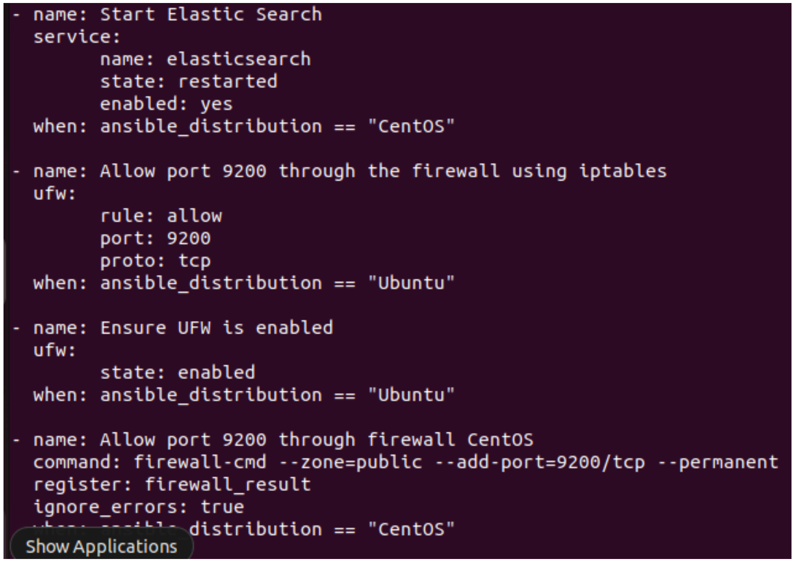
Step 2: Write `main.yml` for every task in every role.

To manage and divide tasks for various Linux distributions, create `main.yml` files in each task folder inside the role directories in Ansible.

Elasticsearch.yml:

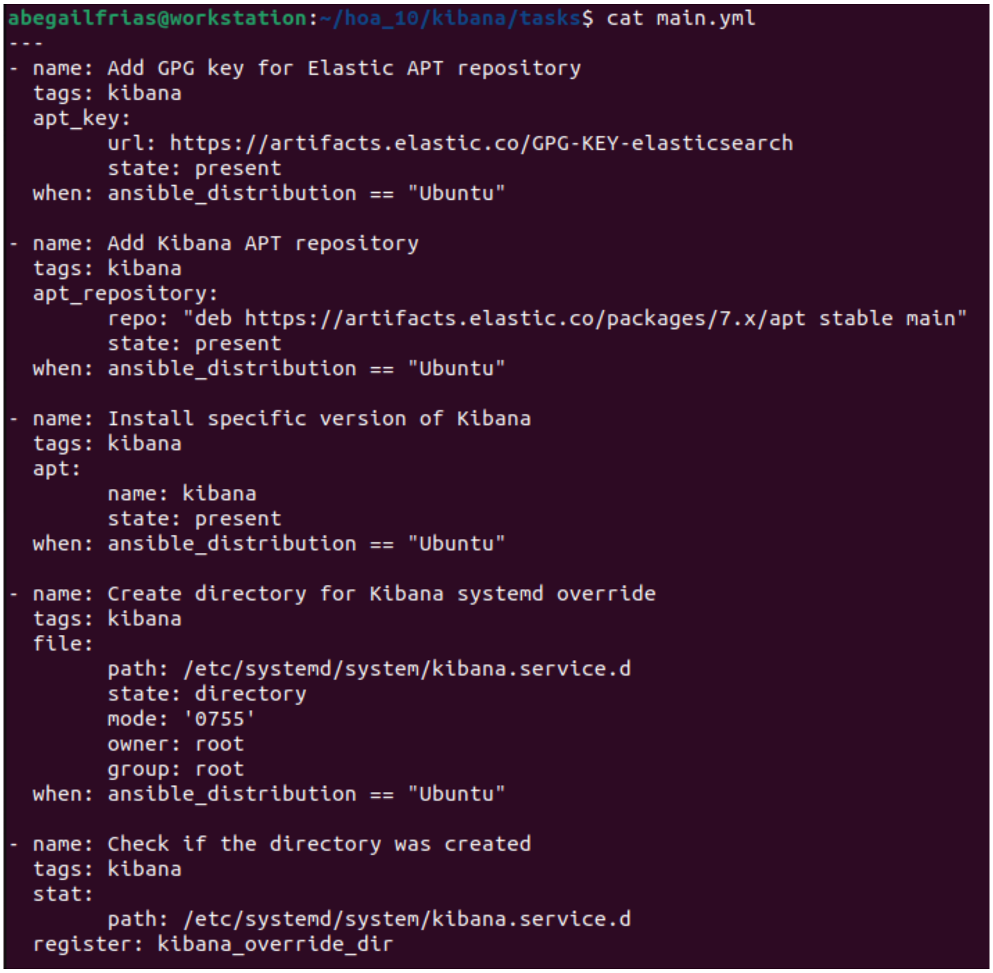


```
abegailfrias@workstation:~/hoa_10/elasticsearch/tasks$ cat main.yml
---
- name: Install Java
  yum:
        name: java-11-openjdk
        state: present
  when: ansible_distribution == "CentOS"

- name: Install EPEL repository
  yum:
        name: epel-release
        state: latest
  when: ansible_distribution == "CentOS"

- name: Install Elastic Search YUM repository
  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: Install Elastic Search
  dnf:
        name: elasticsearch
        state: present
  when: ansible_distribution == "CentOS"

- name: Configure Elastic Search
  template:
        src: elasticsearch.yml.j2
        dest: /etc/elasticsearch/elasticsearch.yml
  when: ansible_distribution == "CentOS"

- name: Start Elastic Search
  service:
```

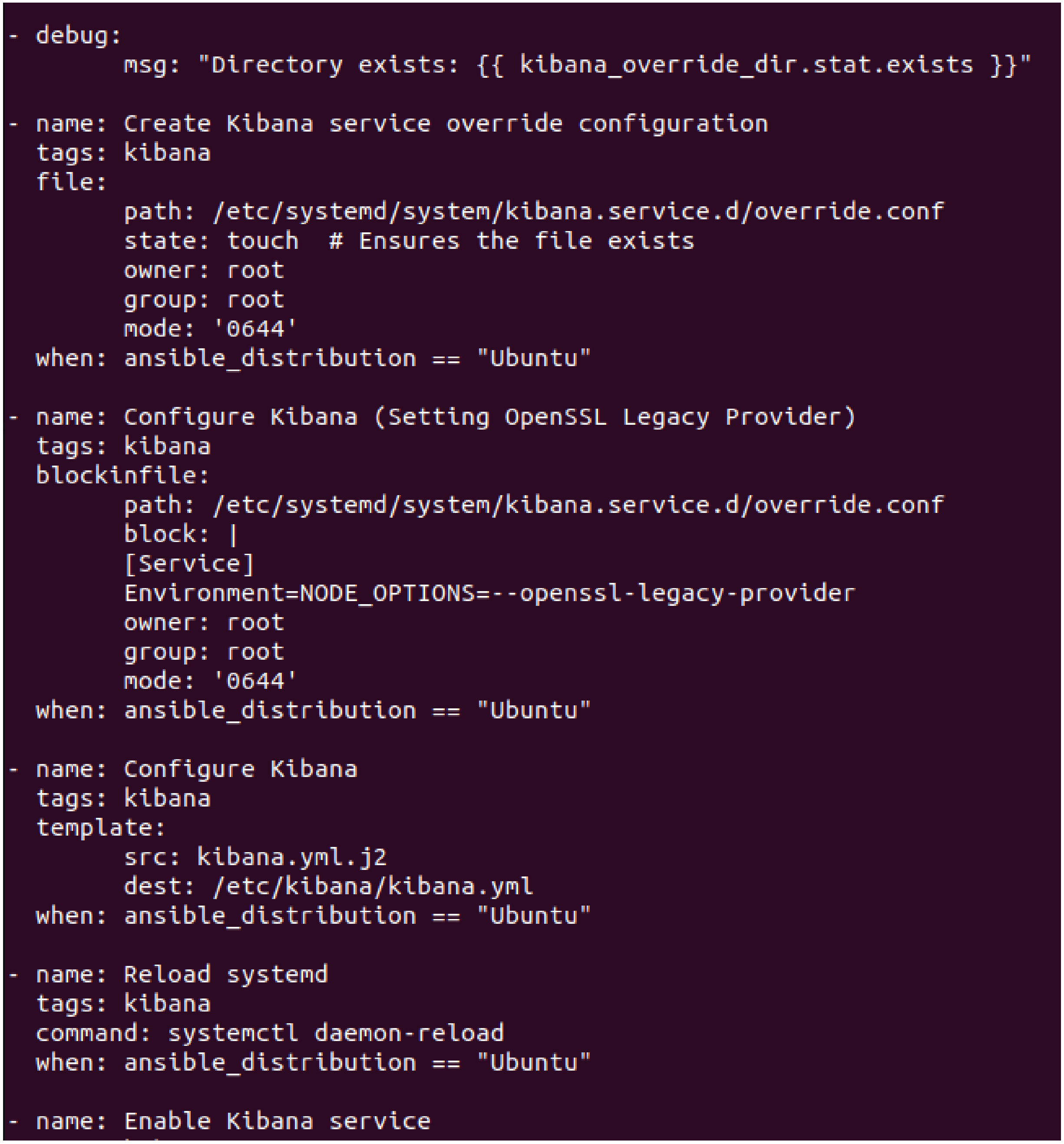


```
- name: Start Elastic Search
  service:
        name: elasticsearch
        state: restarted
        enabled: yes
  when: ansible_distribution == "CentOS"

- name: Allow port 9200 through the firewall using iptables
  ufw:
        rule: allow
        port: 9200
        proto: tcp
  when: ansible_distribution == "Ubuntu"

- name: Ensure UFW is enabled
  ufw:
        state: enabled
  when: ansible_distribution == "Ubuntu"

- name: Allow port 9200 through firewall CentOS
  command: firewall-cmd --zone=public --add-port=9200/tcp --permanent
  register: firewall_result
  ignore_errors: true
  when: ansible_distribution == "CentOS"
```

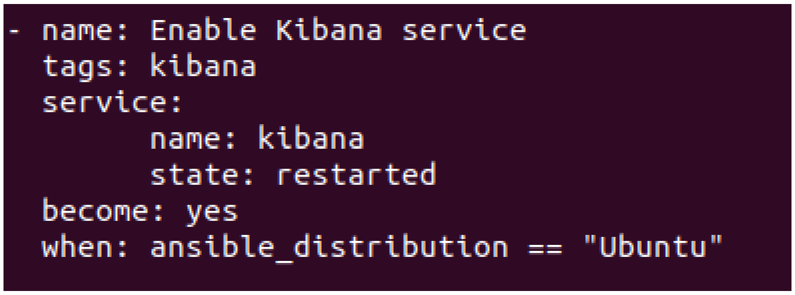
Kibana:



```
abegailfrias@workstation:~/hoa_10/kibana/tasks$ cat main.yml
---
- name: Add GPG key for Elastic APT repository
  tags: kibana
  apt_key:
        url: https://artifacts.elastic.co/GPG-KEY-elasticsearch
        state: present
  when: ansible_distribution == "Ubuntu"

- name: Add Kibana APT repository
  tags: kibana
  apt_repository:
        repo: "deb https://artifacts.elastic.co/packages/7.x/apt stable main"
        state: present
  when: ansible_distribution == "Ubuntu"

- name: Install specific version of Kibana
  tags: kibana
  apt:
        name: kibana
        state: present
  when: ansible_distribution == "Ubuntu"

- name: Create directory for Kibana systemd override
  tags: kibana
  file:
        path: /etc/systemd/system/kibana.service.d
        state: directory
        mode: '0755'
        owner: root
        group: root
  when: ansible_distribution == "Ubuntu"

- name: Check if the directory was created
  tags: kibana
  stat:
        path: /etc/systemd/system/kibana.service.d
  register: kibana_override_dir
```

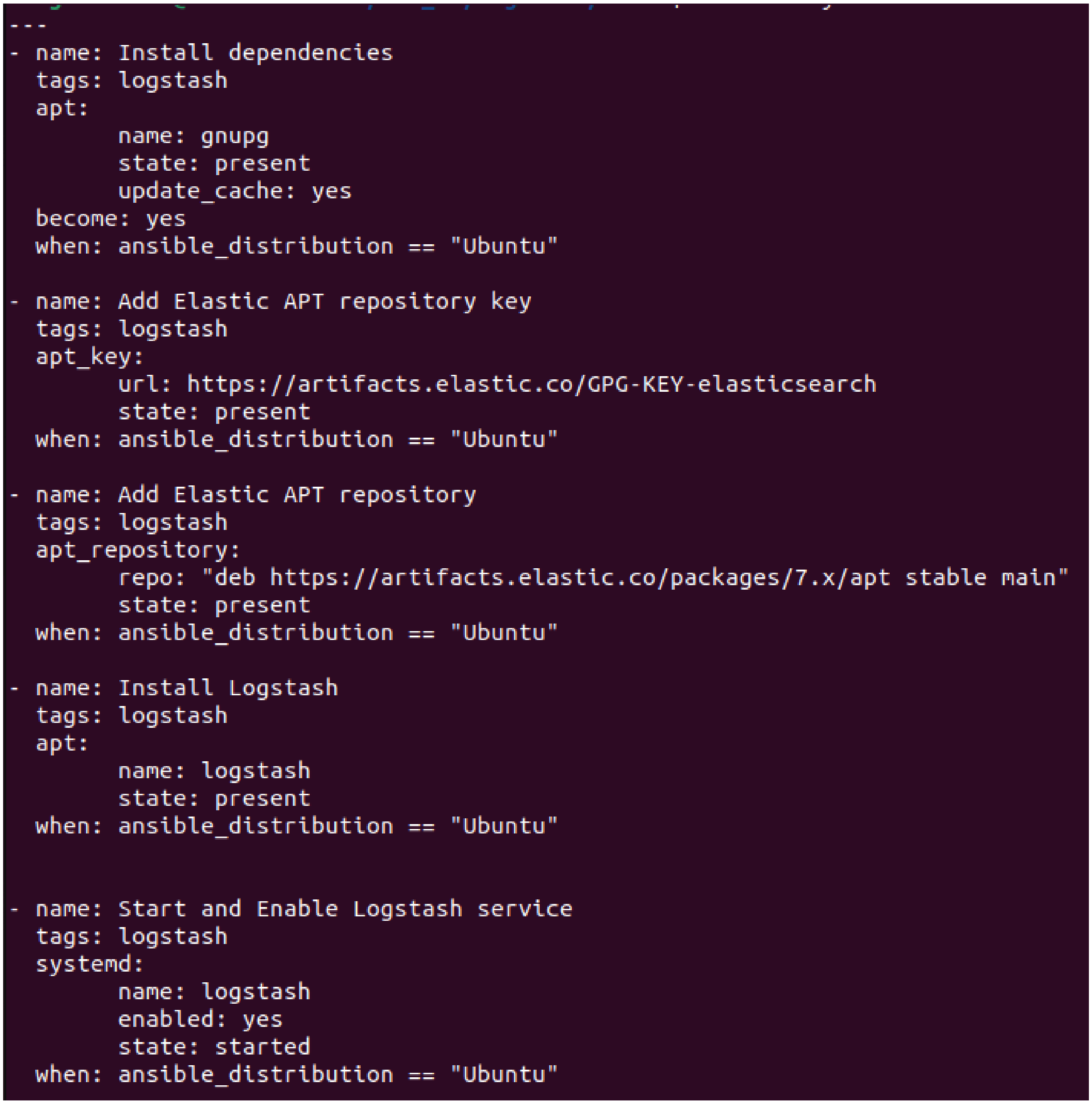


```
- debug:
        msg: "Directory exists: {{ kibana_override_dir.stat.exists }}"

- name: Create Kibana service override configuration
  tags: kibana
  file:
        path: /etc/systemd/system/kibana.service.d/override.conf
        state: touch  # Ensures the file exists
        owner: root
        group: root
        mode: '0644'
  when: ansible_distribution == "Ubuntu"

- name: Configure Kibana (Setting OpenSSL Legacy Provider)
  tags: kibana
  blockinfile:
        path: /etc/systemd/system/kibana.service.d/override.conf
        block: |
        [Service]
        Environment=NODE_OPTIONS=--openssl-legacy-provider
        owner: root
        group: root
        mode: '0644'
  when: ansible_distribution == "Ubuntu"

- name: Configure Kibana
  tags: kibana
  template:
        src: kibana.yml.j2
        dest: /etc/kibana/kibana.yml
  when: ansible_distribution == "Ubuntu"

- name: Reload systemd
  tags: kibana
  command: systemctl daemon-reload
  when: ansible_distribution == "Ubuntu"

- name: Enable Kibana service
```



```
- name: Enable Kibana service
  tags: kibana
  service:
        name: kibana
        state: restarted
  become: yes
  when: ansible_distribution == "Ubuntu"
```

logstash:

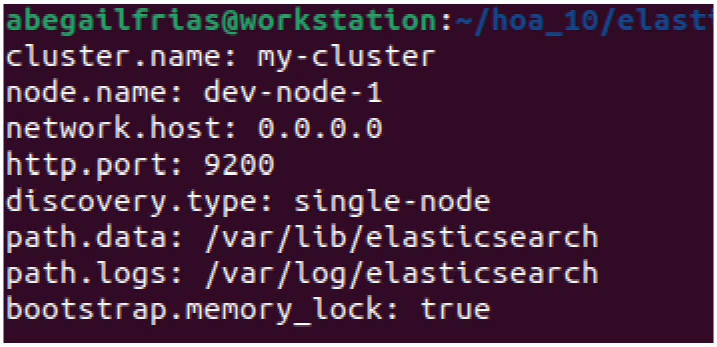


```
---
- name: Install dependencies
  tags: logstash
  apt:
        name: gnupg
        state: present
        update_cache: yes
  become: yes
  when: ansible_distribution == "Ubuntu"

- name: Add Elastic APT repository key
  tags: logstash
  apt_key:
        url: https://artifacts.elastic.co/GPG-KEY-elasticsearch
        state: present
  when: ansible_distribution == "Ubuntu"

- name: Add Elastic APT repository
  tags: logstash
  apt_repository:
        repo: "deb https://artifacts.elastic.co/packages/7.x/apt stable main"
        state: present
  when: ansible_distribution == "Ubuntu"

- name: Install Logstash
  tags: logstash
  apt:
        name: logstash
        state: present
  when: ansible_distribution == "Ubuntu"


- name: Start and Enable Logstash service
  tags: logstash
  systemd:
        name: logstash
        enabled: yes
        state: started
  when: ansible_distribution == "Ubuntu"
```

Step 3: Produce Configuration Documents

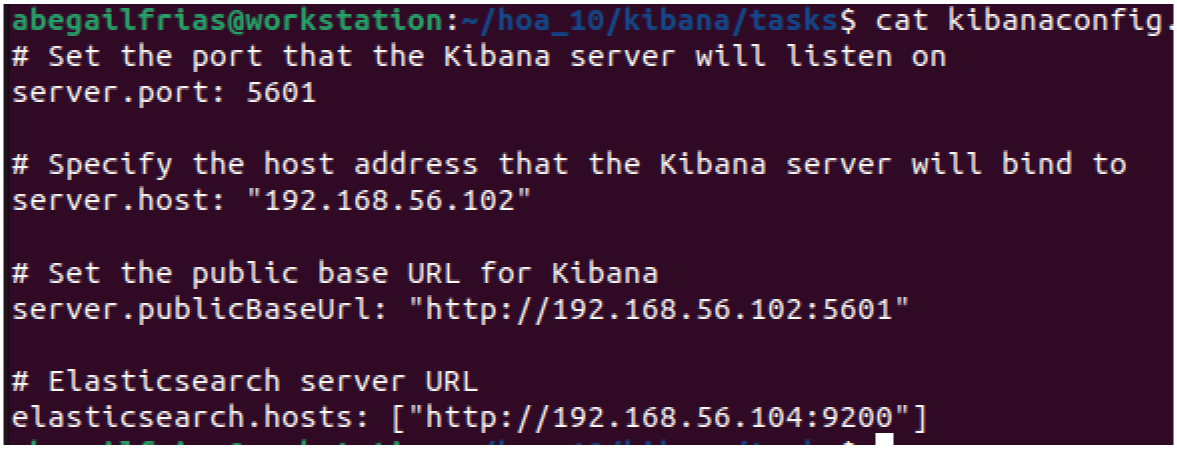
After the services are installed, they must be set up. Use a popular templating engine, `.j2` (Jinja2) files, to generate the required configuration files.

elasticsearch configuration:



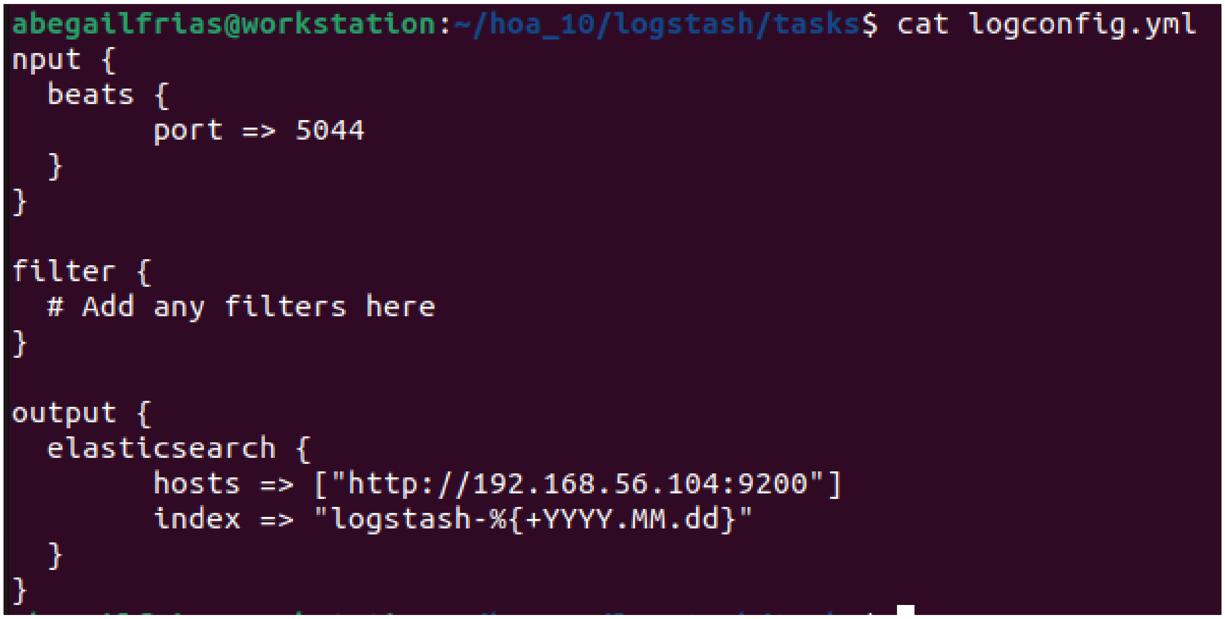
```
abegailfrias@workstation:~/hoa_10/elast
cluster.name: my-cluster
node.name: dev-node-1
network.host: 0.0.0.0
http.port: 9200
discovery.type: single-node
path.data: /var/lib/elasticsearch
path.logs: /var/log/elasticsearch
bootstrap.memory_lock: true
```

kibana configuration:



```
abegailfrias@workstation:~/hoa_10/kibana/tasks$ cat kibanaconfig.
# Set the port that the Kibana server will listen on
server.port: 5601

# Specify the host address that the Kibana server will bind to
server.host: "192.168.56.102"

# Set the public base URL for Kibana
server.publicBaseUrl: "http://192.168.56.102:5601"

# Elasticsearch server URL
elasticsearch.hosts: ["http://192.168.56.104:9200"]
```

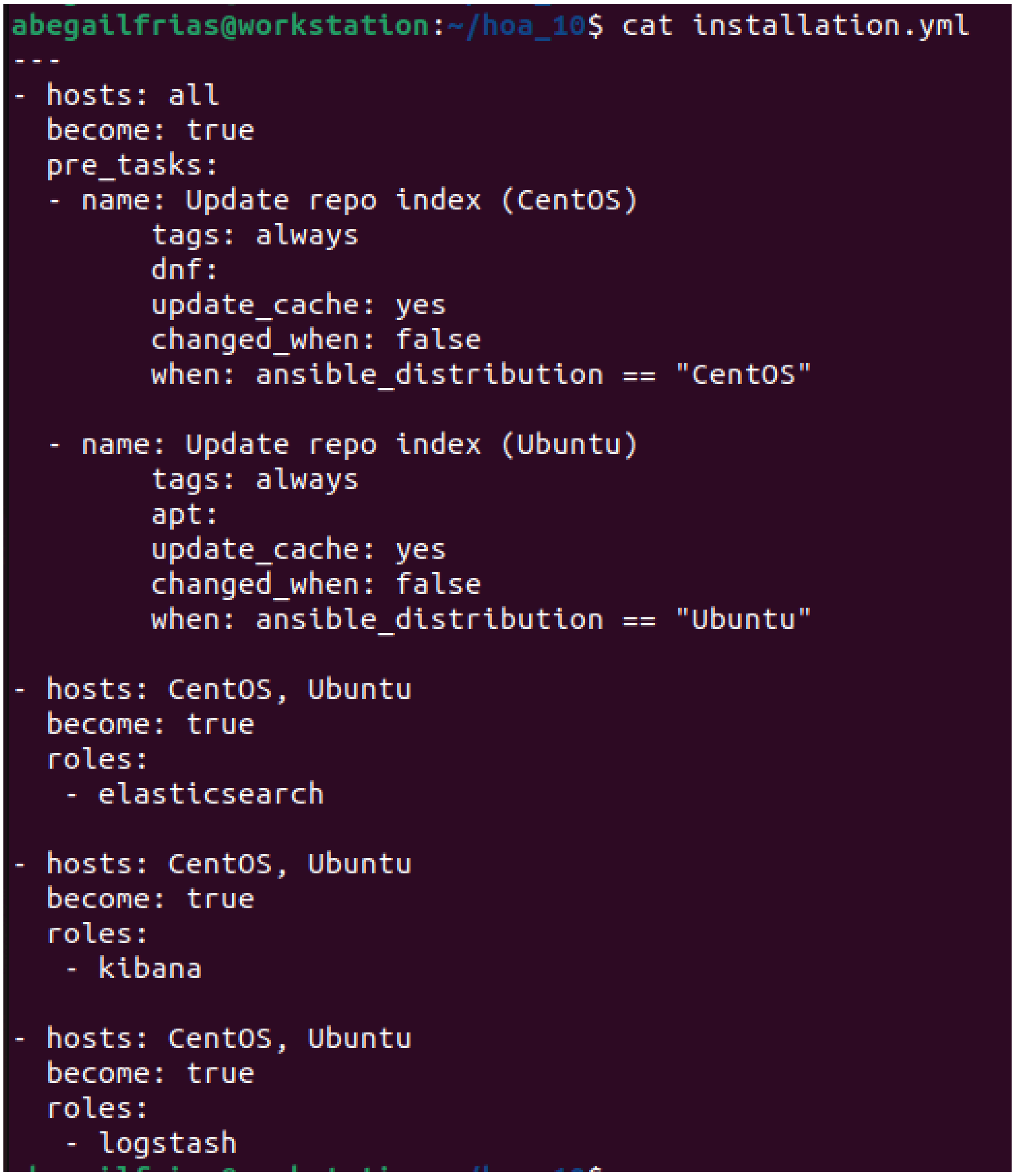
logstash configuration:



```
abegailfrias@workstation:~/hoa_10/logstash/tasks$ cat logconfig.yml
nput {
  beats {
        port => 5044
  }
}

filter {
  # Add any filters here
}

output {
  elasticsearch {
        hosts => ["http://192.168.56.104:9200"]
        index => "logstash-%{+YYYY.MM.dd}"
  }
}
```

Step 4: Create the Main Installation File

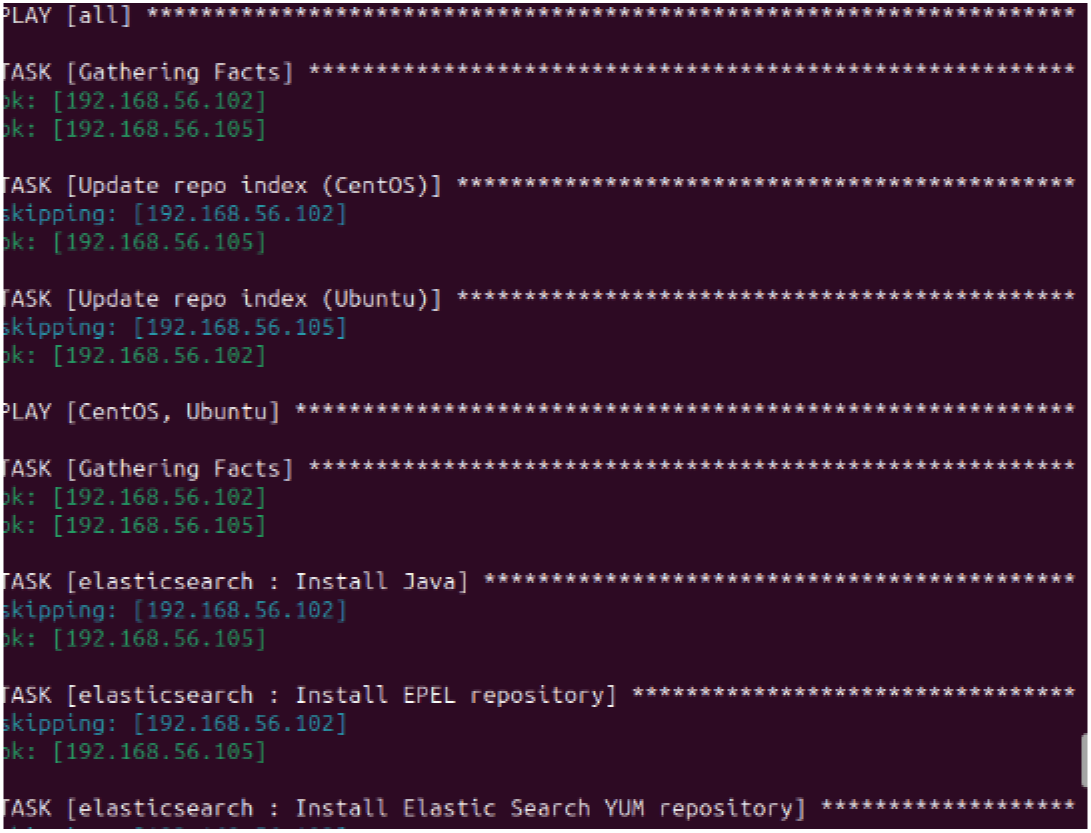
In the repository's main directory, create a `.yml` file to handle basic configurations and execute the `main.yml` from each role in a single run.



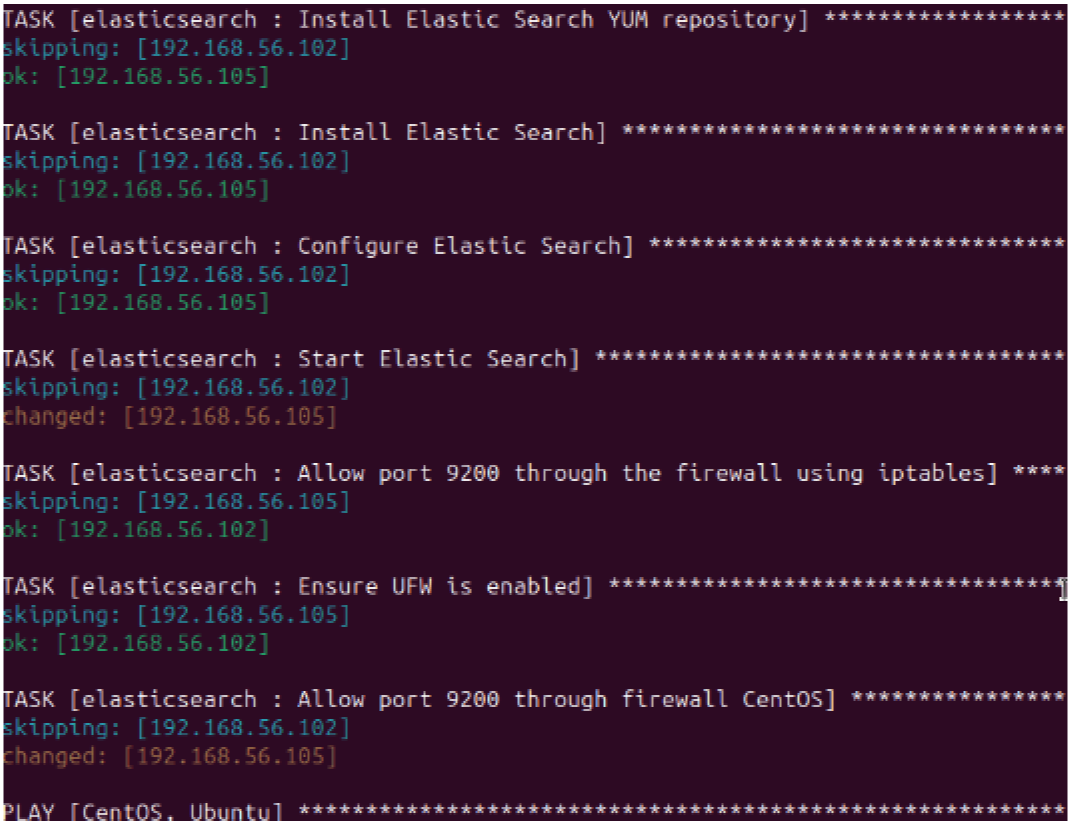
```
abegailfrias@workstation:~/hoa_10$ cat installation.yml
---
- hosts: all
  become: true
  pre_tasks:
  - name: Update repo index (CentOS)
        tags: always
        dnf:
        update_cache: yes
        changed_when: false
        when: ansible_distribution == "CentOS"

  - name: Update repo index (Ubuntu)
        tags: always
        apt:
        update_cache: yes
        changed_when: false
        when: ansible_distribution == "Ubuntu"

- hosts: CentOS, Ubuntu
  become: true
  roles:
   - elasticsearch

- hosts: CentOS, Ubuntu
  become: true
  roles:
   - kibana

- hosts: CentOS, Ubuntu
  become: true
  roles:
   - logstash
```

Step 5: Run the Main Installation Playbook and Verify

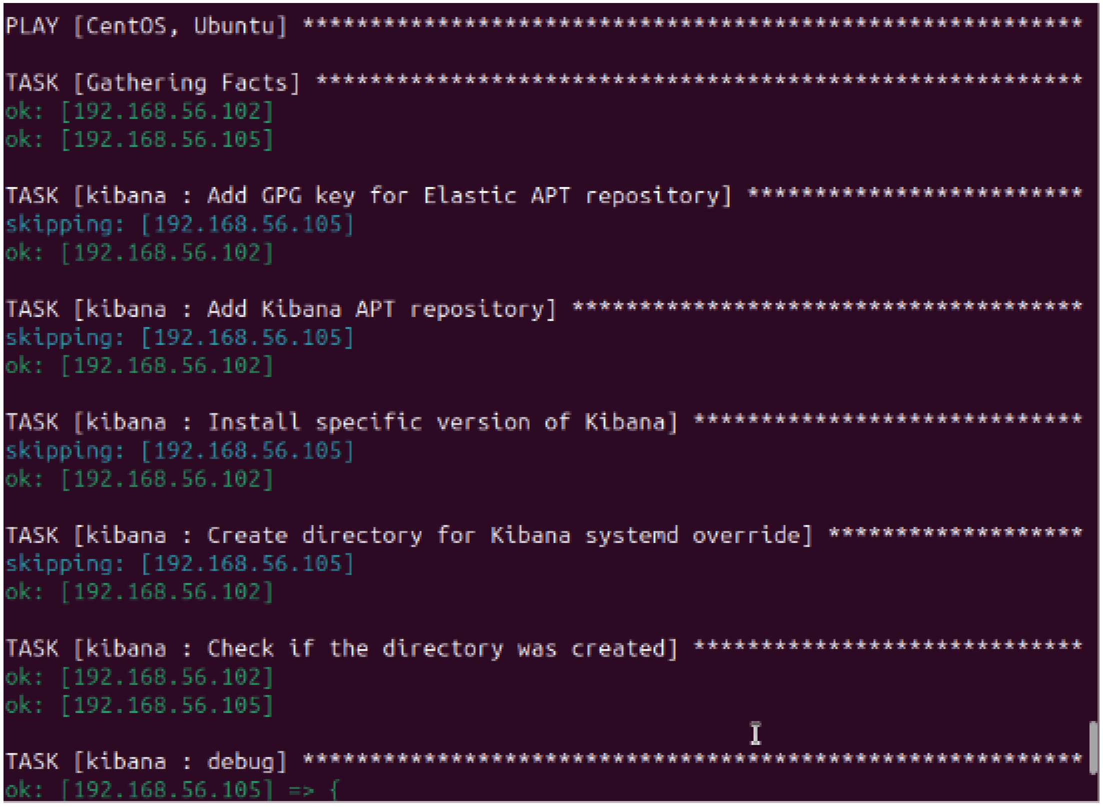
Run the command `ansible-playbook --ask-become-pass <main installation file>` to prompt for the admin or root password on each system. This will execute the playbook, showing progress and any errors encountered.



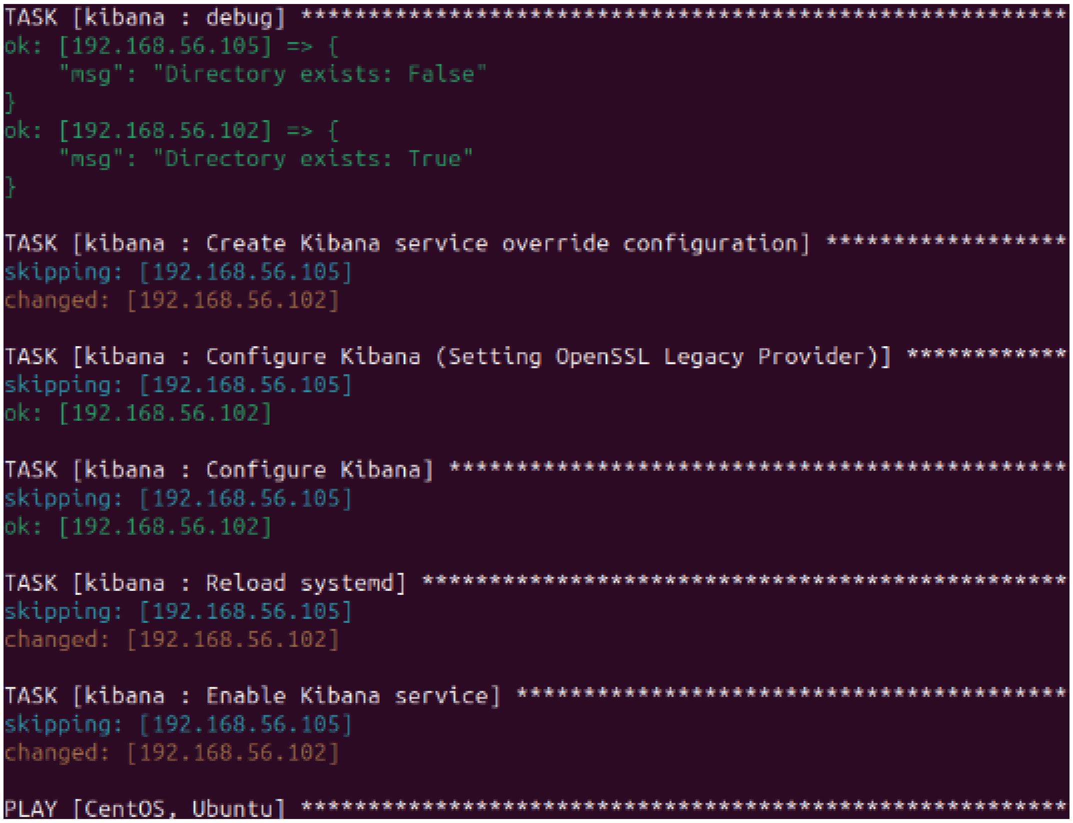
```
PLAY [all] *********************************************************************

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.102]
ok: [192.168.56.105]

TASK [Update repo index (CentOS)] **********************************************
skipping: [192.168.56.102]
ok: [192.168.56.105]

TASK [Update repo index (Ubuntu)] **********************************************
skipping: [192.168.56.105]
ok: [192.168.56.102]

PLAY [CentOS, Ubuntu] **********************************************************

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.102]
ok: [192.168.56.105]

TASK [elasticsearch : Install Java] ********************************************
skipping: [192.168.56.102]
ok: [192.168.56.105]

TASK [elasticsearch : Install EPEL repository] *********************************
skipping: [192.168.56.102]
ok: [192.168.56.105]

TASK [elasticsearch : Install Elastic Search YUM repository] *******************
```



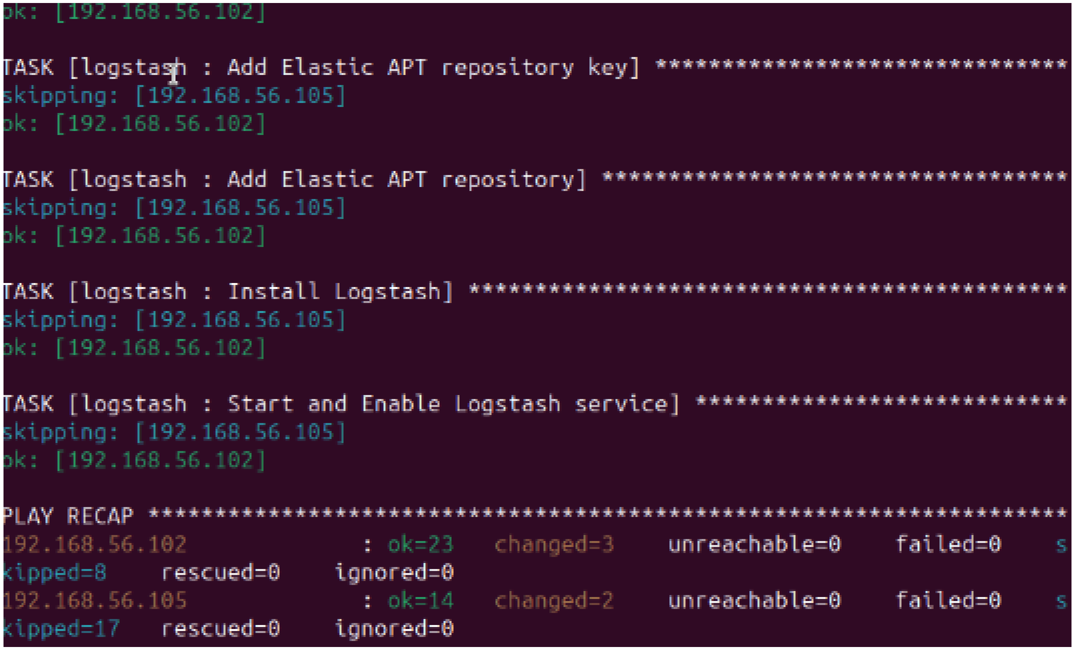
```
TASK [elasticsearch : Install Elastic Search YUM repository] ******************
skipping: [192.168.56.102]
ok: [192.168.56.105]

TASK [elasticsearch : Install Elastic Search] *********************************
skipping: [192.168.56.102]
ok: [192.168.56.105]

TASK [elasticsearch : Configure Elastic Search] *******************************
skipping: [192.168.56.102]
ok: [192.168.56.105]

TASK [elasticsearch : Start Elastic Search] ***********************************
skipping: [192.168.56.102]
changed: [192.168.56.105]

TASK [elasticsearch : Allow port 9200 through the firewall using iptables] ****
skipping: [192.168.56.105]
ok: [192.168.56.102]

TASK [elasticsearch : Ensure UFW is enabled] **********************************
skipping: [192.168.56.105]
ok: [192.168.56.102]

TASK [elasticsearch : Allow port 9200 through firewall CentOS] ****************
skipping: [192.168.56.102]
changed: [192.168.56.105]

PLAY [CentOS, Ubuntu] *********************************************************
```



```
PLAY [CentOS, Ubuntu] ***********************************************************

TASK [Gathering Facts] **********************************************************
ok: [192.168.56.102]
ok: [192.168.56.105]

TASK [kibana : Add GPG key for Elastic APT repository] **************************
skipping: [192.168.56.105]
ok: [192.168.56.102]

TASK [kibana : Add Kibana APT repository] ***************************************
skipping: [192.168.56.105]
ok: [192.168.56.102]

TASK [kibana : Install specific version of Kibana] ******************************
skipping: [192.168.56.105]
ok: [192.168.56.102]

TASK [kibana : Create directory for Kibana systemd override] ********************
skipping: [192.168.56.105]
ok: [192.168.56.102]

TASK [kibana : Check if the directory was created] ******************************
ok: [192.168.56.102]
ok: [192.168.56.105]

TASK [kibana : debug] ***********************************************************
ok: [192.168.56.105] => {
```



```
TASK [kibana : debug] ***********************************************************
ok: [192.168.56.105] => {
    "msg": "Directory exists: False"
}
ok: [192.168.56.102] => {
    "msg": "Directory exists: True"
}

TASK [kibana : Create Kibana service override configuration] ********************
skipping: [192.168.56.105]
changed: [192.168.56.102]

TASK [kibana : Configure Kibana (Setting OpenSSL Legacy Provider)] **************
skipping: [192.168.56.105]
ok: [192.168.56.102]

TASK [kibana : Configure Kibana] ************************************************
skipping: [192.168.56.105]
ok: [192.168.56.102]

TASK [kibana : Reload systemd] **************************************************
skipping: [192.168.56.105]
changed: [192.168.56.102]

TASK [kibana : Enable Kibana service] *******************************************
skipping: [192.168.56.105]
changed: [192.168.56.102]

PLAY [CentOS, Ubuntu] ***********************************************************
```



```
ok: [192.168.56.102]

TASK [logstash : Add Elastic APT repository key] *********************************
skipping: [192.168.56.105]
ok: [192.168.56.102]

TASK [logstash : Add Elastic APT repository] *************************************
skipping: [192.168.56.105]
ok: [192.168.56.102]

TASK [logstash : Install Logstash] ***********************************************
skipping: [192.168.56.105]
ok: [192.168.56.102]

TASK [logstash : Start and Enable Logstash service] ******************************
skipping: [192.168.56.105]
ok: [192.168.56.102]

PLAY RECAP ***********************************************************************
192.168.56.102             : ok=23   changed=3    unreachable=0    failed=0    s
kipped=8    rescued=0    ignored=0
192.168.56.105             : ok=14   changed=2    unreachable=0    failed=0    s
kipped=17   rescued=0    ignored=0
```

github link:

https://github.com/wonbe/hoa_10

Reflections:

Answer the following:

1. What are the benefits of having log monitoring tool?
   By identifying problems and suspicious activity, log monitoring tools provide real-time monitoring, quicker troubleshooting, and enhanced security. They offer centralized log management, assist in ensuring adherence to regulatory standards, and send alerts for prompt action. Additionally, by providing information about system performance, these tools facilitate proactive maintenance and optimization to avert future issues.

Conclusions:

System administrators can manage servers and systems more effectively by using log monitoring, which is a powerful tool that tracks log data for security threats and performance problems. Administrators can promptly resolve issues by spotting trends and abnormalities. Application debugging is made simpler by platforms such as GrayLog, which help manage both structured and unstructured logs. Furthermore, real-time data collection, analysis, and visualization across multiple data formats and sources are provided by tools like Elasticsearch, Kibana, Beats, and Logstash.