Name: Ballesteros, John Erwin S.	**Date Performed: 28/10/2024**
Course/Section: CpE31s2	**Date Submitted: 4/11/2024**
Instructor: Engr. Robin Valenzuela	**Semester and SY: 1st Sem, ‘24-’25**

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.

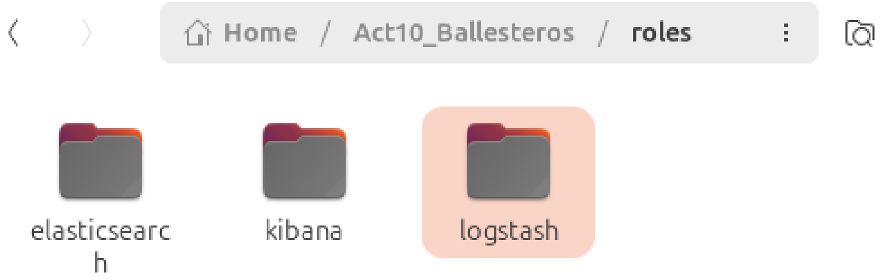
Source: https://www.graylog.org/products/open-source

3. Tasks

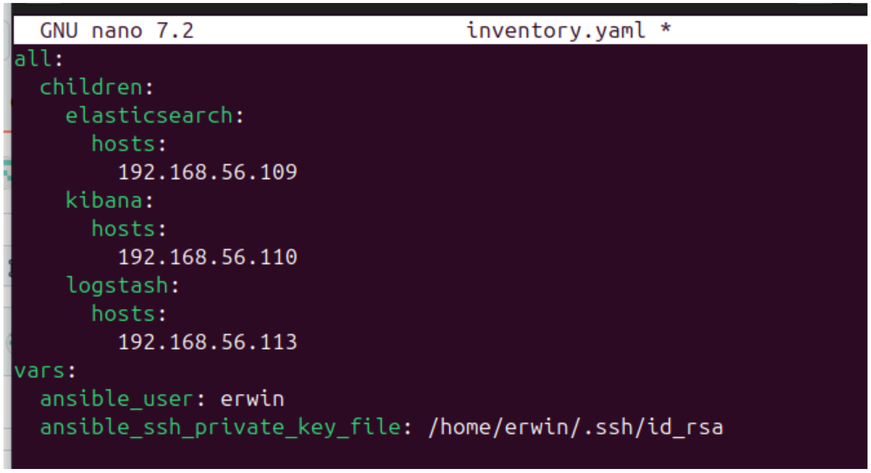
1. Create a playbook that:
   a. Install and configure Elastic Stack in separate hosts (Elastic Search, Kibana, Logstash)
2. Apply the concept of creating roles.
3. Describe how you did step 1. (Provide screenshots and explanations in your report. Make your report detailed such that it will look like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

Step 1: Create or clone a github repository a directory for you playbook that contains the appropriate roles directory (elasticsearch, kibana and logstash) and inside them contains tasks and templates inside it

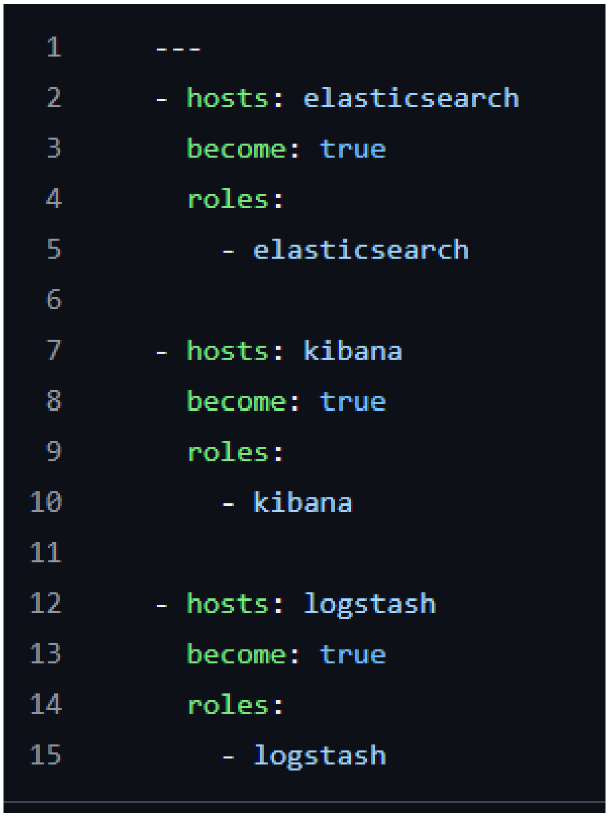


Step 2: Modify your existing ansible inventory, and ansible hosts with the appropriate roles



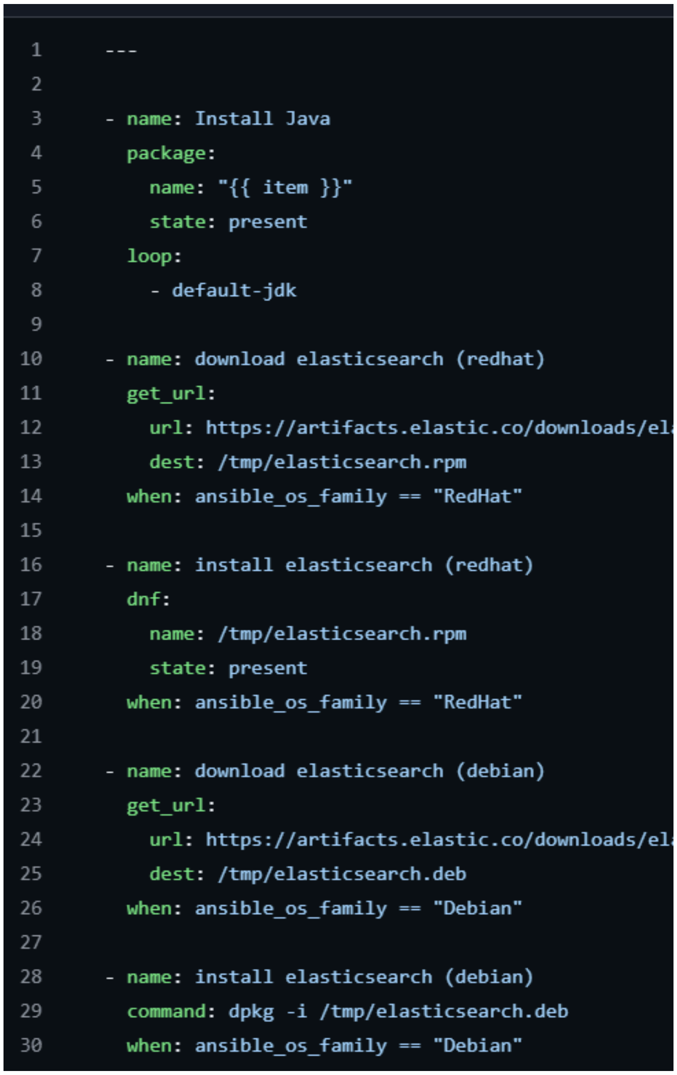
```
  GNU nano 7.2                                inventory.yaml *
all:
  children:
    elasticsearch:
      hosts:
        192.168.56.109
    kibana:
      hosts:
        192.168.56.110
    logstash:
      hosts:
        192.168.56.113
vars:
  ansible_user: erwin
  ansible_ssh_private_key_file: /home/erwin/.ssh/id_rsa
```

Step 4: In your directory create a site.yml file with the following code:

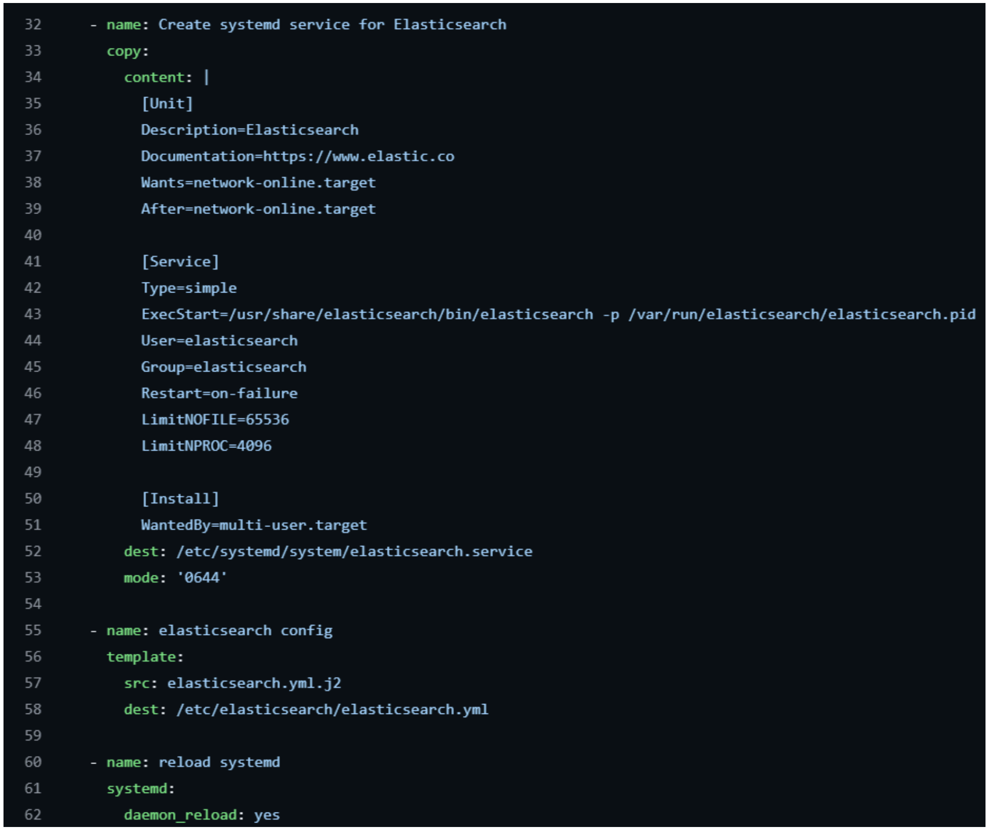


```
---
- hosts: elasticsearch
  become: true
  roles:
    - elasticsearch

- hosts: kibana
  become: true
  roles:
    - kibana

- hosts: logstash
  become: true
  roles:
    - logstash
```

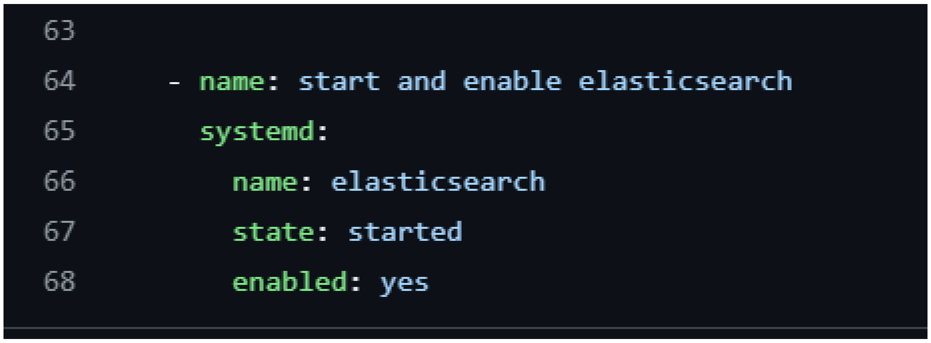
Step 5: Create a main.yml for elasticsearch/tasks with the following code:



```
---

- name: Install Java
  package:
    name: "{{ item }}"
    state: present
  loop:
    - default-jdk

- name: download elasticsearch (redhat)
  get_url:
    url: https://artifacts.elastic.co/downloads/el
    dest: /tmp/elasticsearch.rpm
  when: ansible_os_family == "RedHat"

- name: install elasticsearch (redhat)
  dnf:
    name: /tmp/elasticsearch.rpm
    state: present
  when: ansible_os_family == "RedHat"

- name: download elasticsearch (debian)
  get_url:
    url: https://artifacts.elastic.co/downloads/el
    dest: /tmp/elasticsearch.deb
  when: ansible_os_family == "Debian"

- name: install elasticsearch (debian)
  command: dpkg -i /tmp/elasticsearch.deb
  when: ansible_os_family == "Debian"
```

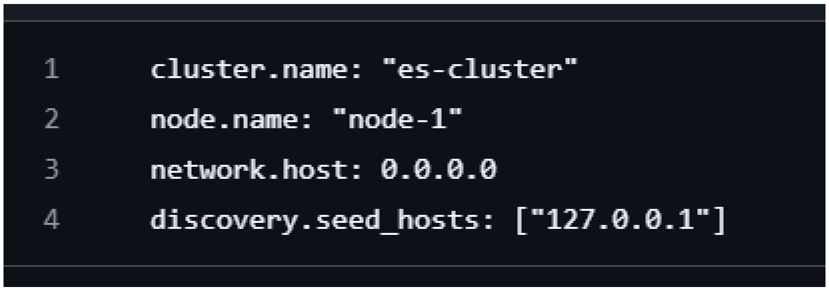


```
    - name: Create systemd service for Elasticsearch
      copy:
        content: |
          [Unit]
          Description=Elasticsearch
          Documentation=https://www.elastic.co
          Wants=network-online.target
          After=network-online.target

          [Service]
          Type=simple
          ExecStart=/usr/share/elasticsearch/bin/elasticsearch -p /var/run/elasticsearch/elasticsearch.pid
          User=elasticsearch
          Group=elasticsearch
          Restart=on-failure
          LimitNOFILE=65536
          LimitNPROC=4096

          [Install]
          WantedBy=multi-user.target
        dest: /etc/systemd/system/elasticsearch.service
        mode: '0644'

    - name: elasticsearch config
      template:
        src: elasticsearch.yml.j2
        dest: /etc/elasticsearch/elasticsearch.yml

    - name: reload systemd
      systemd:
        daemon_reload: yes
```



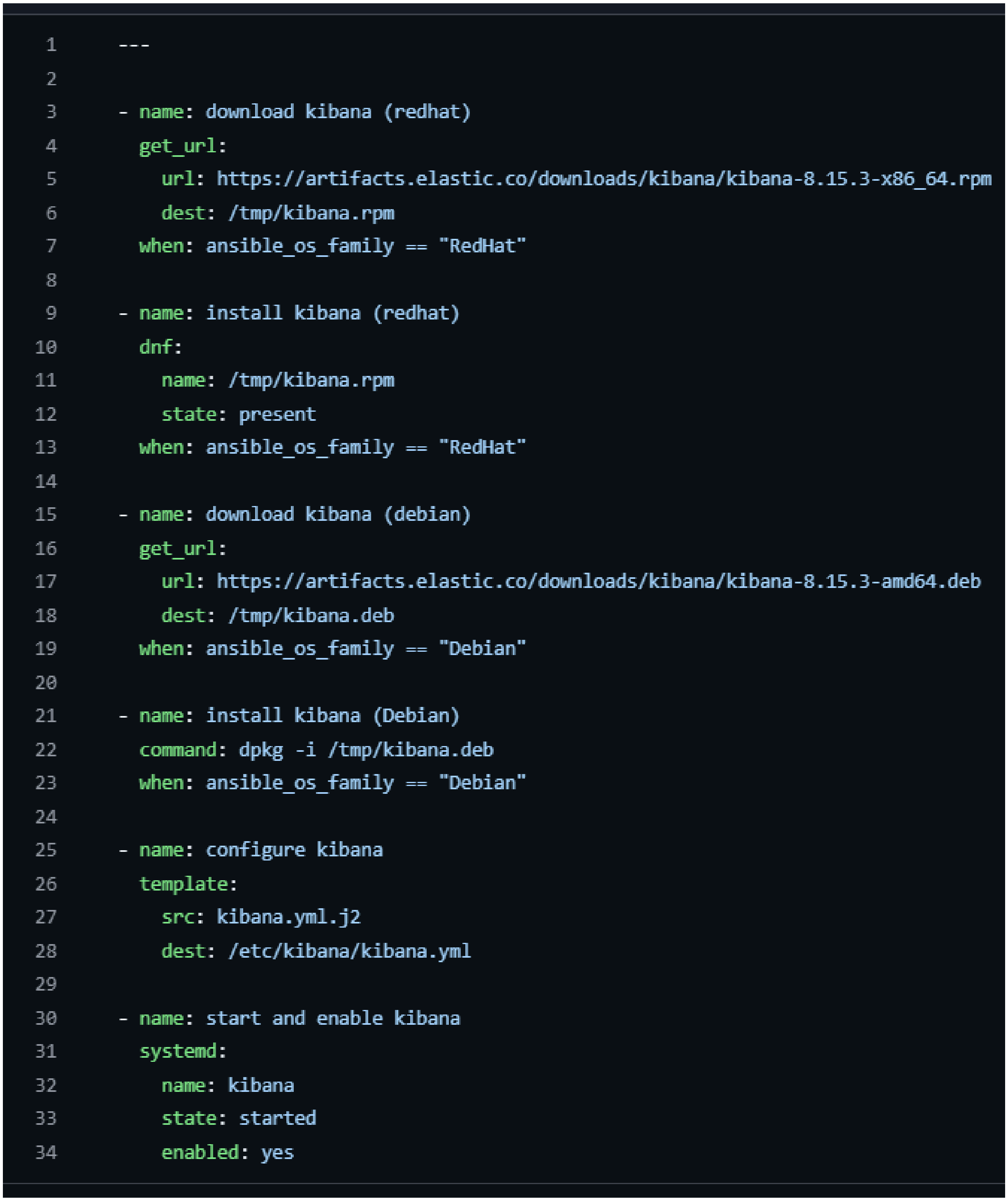
```

    - name: start and enable elasticsearch
      systemd:
        name: elasticsearch
        state: started
        enabled: yes
```

Step 6: Create a template file in elasticsearch/templates named as elasticsearch.yml.j2 with the code:



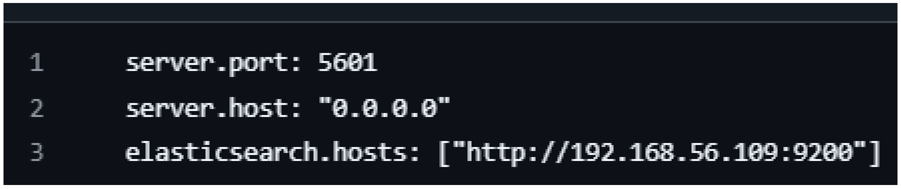
```
cluster.name: "es-cluster"
node.name: "node-1"
network.host: 0.0.0.0
discovery.seed_hosts: ["127.0.0.1"]
```

Step 7: Create a main.yml in kibana/tasks with the following code



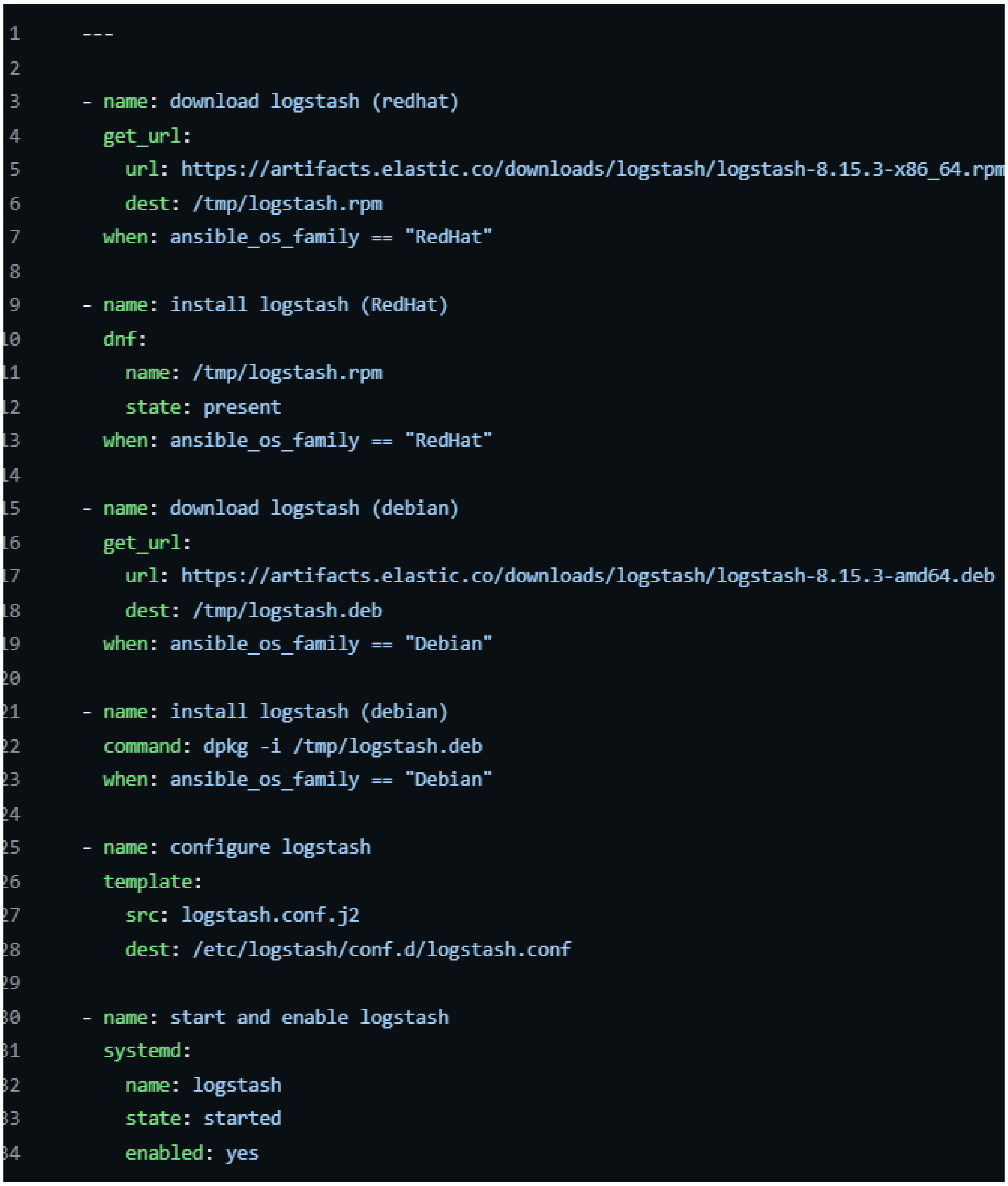
```
---

- name: download kibana (redhat)
  get_url:
    url: https://artifacts.elastic.co/downloads/kibana/kibana-8.15.3-x86_64.rpm
    dest: /tmp/kibana.rpm
  when: ansible_os_family == "RedHat"

- name: install kibana (redhat)
  dnf:
    name: /tmp/kibana.rpm
    state: present
  when: ansible_os_family == "RedHat"

- name: download kibana (debian)
  get_url:
    url: https://artifacts.elastic.co/downloads/kibana/kibana-8.15.3-amd64.deb
    dest: /tmp/kibana.deb
  when: ansible_os_family == "Debian"

- name: install kibana (Debian)
  command: dpkg -i /tmp/kibana.deb
  when: ansible_os_family == "Debian"

- name: configure kibana
  template:
    src: kibana.yml.j2
    dest: /etc/kibana/kibana.yml

- name: start and enable kibana
  systemd:
    name: kibana
    state: started
    enabled: yes
```

Step 8: Create a template file in kibana/templates named kibana.yml.j2:



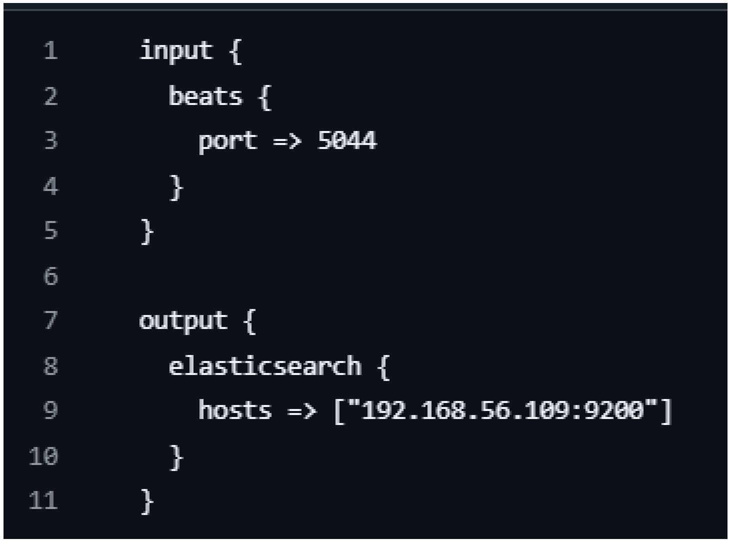
```
server.port: 5601
server.host: "0.0.0.0"
elasticsearch.hosts: ["http://192.168.56.109:9200"]
```

Step 9: Lastly, create a main.yml as well for logstash:



```
---

- name: download logstash (redhat)
  get_url:
    url: https://artifacts.elastic.co/downloads/logstash/logstash-8.15.3-x86_64.rpm
    dest: /tmp/logstash.rpm
  when: ansible_os_family == "RedHat"

- name: install logstash (RedHat)
  dnf:
    name: /tmp/logstash.rpm
    state: present
  when: ansible_os_family == "RedHat"

- name: download logstash (debian)
  get_url:
    url: https://artifacts.elastic.co/downloads/logstash/logstash-8.15.3-amd64.deb
    dest: /tmp/logstash.deb
  when: ansible_os_family == "Debian"

- name: install logstash (debian)
  command: dpkg -i /tmp/logstash.deb
  when: ansible_os_family == "Debian"

- name: configure logstash
  template:
    src: logstash.conf.j2
    dest: /etc/logstash/conf.d/logstash.conf

- name: start and enable logstash
  systemd:
    name: logstash
    state: started
    enabled: yes
```

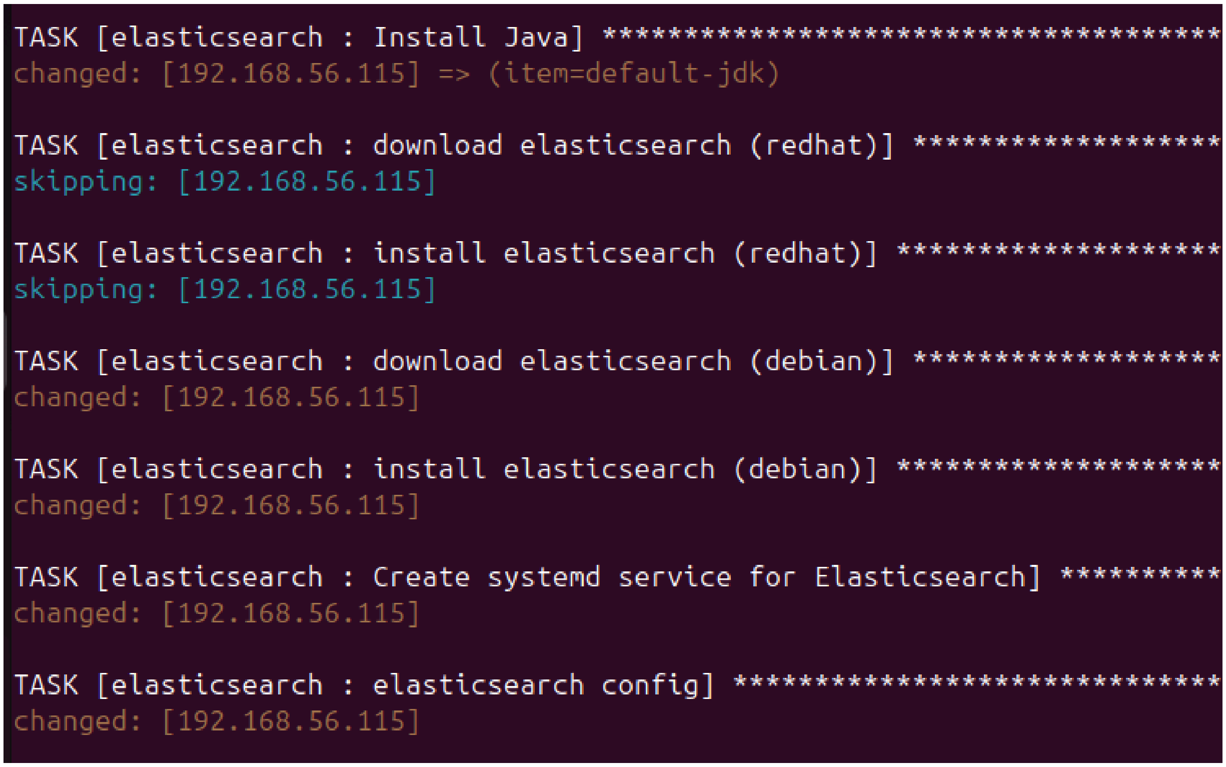
Step 10: Create the template for logstash named logstash.conf.j2



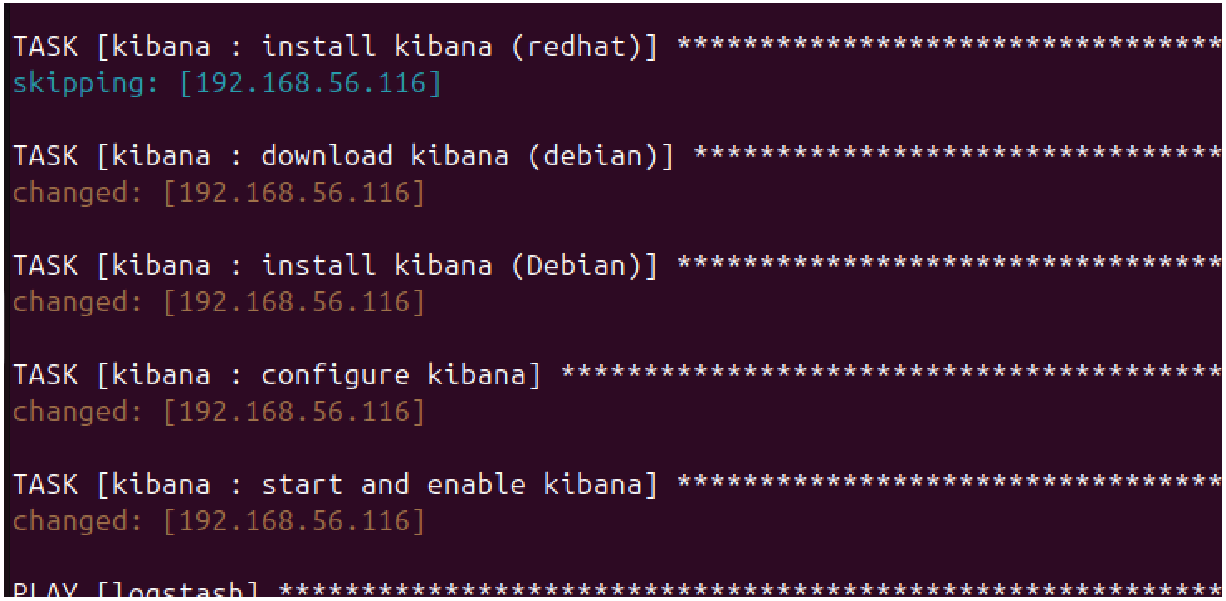
```
input {
  beats {
    port => 5044
  }
}

output {
  elasticsearch {
    hosts => ["192.168.56.109:9200"]
  }
}
```

Step 11: Run the ansible playbook

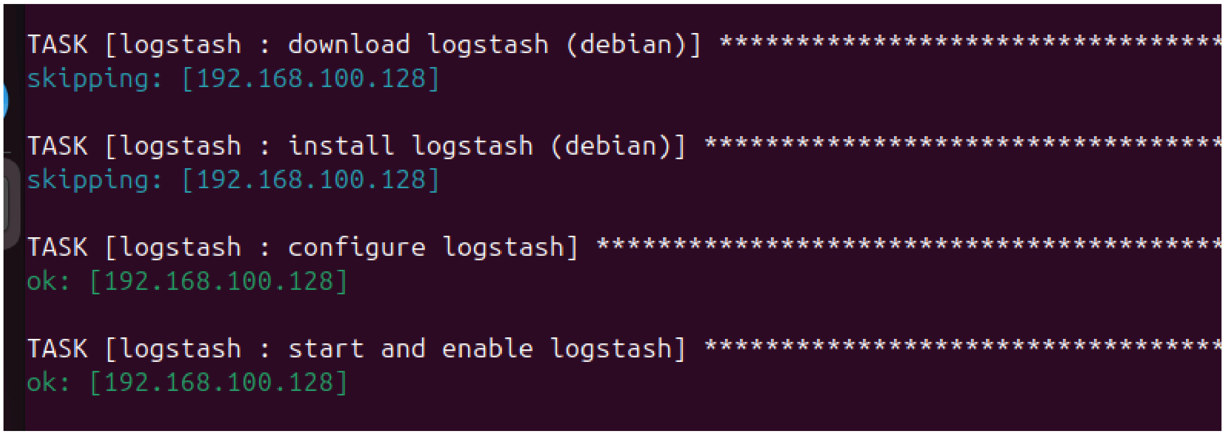
Results:



```
TASK [elasticsearch : Install Java] *************************************
changed: [192.168.56.115] => (item=default-jdk)

TASK [elasticsearch : download elasticsearch (redhat)] *******************
skipping: [192.168.56.115]

TASK [elasticsearch : install elasticsearch (redhat)] ********************
skipping: [192.168.56.115]

TASK [elasticsearch : download elasticsearch (debian)] *******************
changed: [192.168.56.115]

TASK [elasticsearch : install elasticsearch (debian)] ********************
changed: [192.168.56.115]

TASK [elasticsearch : Create systemd service for Elasticsearch] **********
changed: [192.168.56.115]

TASK [elasticsearch : elasticsearch config] ******************************
changed: [192.168.56.115]
```

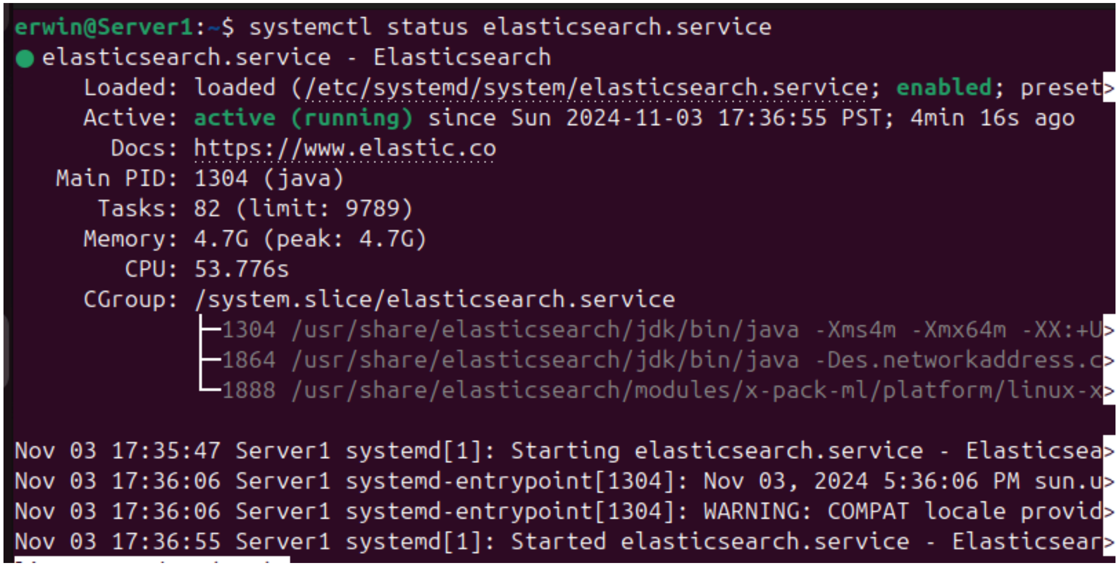


```
TASK [kibana : install kibana (redhat)] **********************************
skipping: [192.168.56.116]

TASK [kibana : download kibana (debian)] *********************************
changed: [192.168.56.116]

TASK [kibana : install kibana (Debian)] **********************************
changed: [192.168.56.116]

TASK [kibana : configure kibana] *****************************************
changed: [192.168.56.116]

TASK [kibana : start and enable kibana] **********************************
changed: [192.168.56.116]

PLAY [logstash] **********************************************************
```

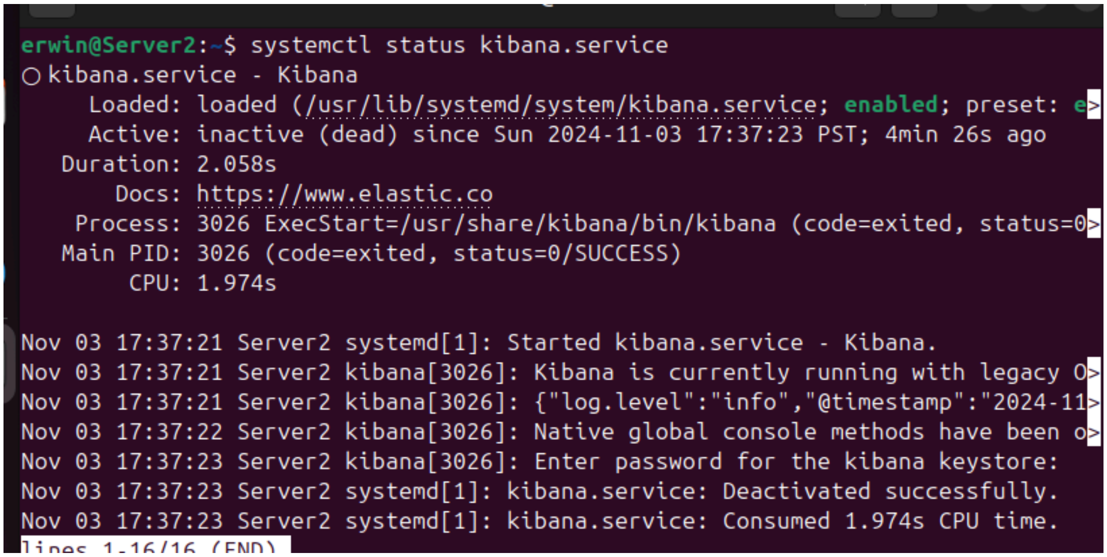


```
TASK [logstash : download logstash (debian)] ****************************
skipping: [192.168.100.128]

TASK [logstash : install logstash (debian)] *****************************
skipping: [192.168.100.128]

TASK [logstash : configure logstash] ************************************
ok: [192.168.100.128]

TASK [logstash : start and enable logstash] *****************************
ok: [192.168.100.128]
```

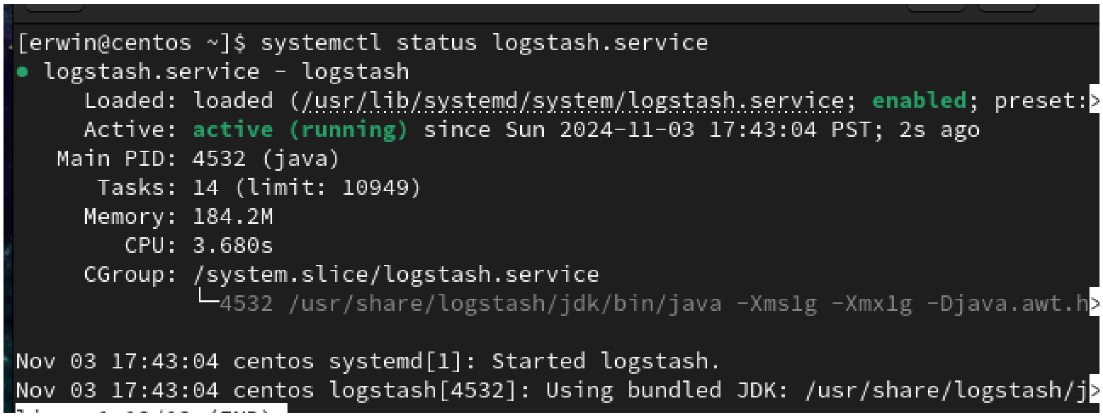
Step 12: Verify if all the services are running



```
erwin@Server1:~$ systemctl status elasticsearch.service
● elasticsearch.service - Elasticsearch
     Loaded: loaded (/etc/systemd/system/elasticsearch.service; enabled; preset>
     Active: active (running) since Sun 2024-11-03 17:36:55 PST; 4min 16s ago
       Docs: https://www.elastic.co
   Main PID: 1304 (java)
      Tasks: 82 (limit: 9789)
     Memory: 4.7G (peak: 4.7G)
        CPU: 53.776s
     CGroup: /system.slice/elasticsearch.service
             ├─1304 /usr/share/elasticsearch/jdk/bin/java -Xms4m -Xmx64m -XX:+U>
             ├─1864 /usr/share/elasticsearch/jdk/bin/java -Des.networkaddress.c>
             └─1888 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux-x>

Nov 03 17:35:47 Server1 systemd[1]: Starting elasticsearch.service - Elasticsea>
Nov 03 17:36:06 Server1 systemd-entrypoint[1304]: Nov 03, 2024 5:36:06 PM sun.u>
Nov 03 17:36:06 Server1 systemd-entrypoint[1304]: WARNING: COMPAT locale provid>
Nov 03 17:36:55 Server1 systemd[1]: Started elasticsearch.service - Elasticsear>
```



```
erwin@Server2:~$ systemctl status kibana.service
○ kibana.service - Kibana
     Loaded: loaded (/usr/lib/systemd/system/kibana.service; enabled; preset: e>
     Active: inactive (dead) since Sun 2024-11-03 17:37:23 PST; 4min 26s ago
   Duration: 2.058s
       Docs: https://www.elastic.co
    Process: 3026 ExecStart=/usr/share/kibana/bin/kibana (code=exited, status=0>
   Main PID: 3026 (code=exited, status=0/SUCCESS)
        CPU: 1.974s

Nov 03 17:37:21 Server2 systemd[1]: Started kibana.service - Kibana.
Nov 03 17:37:21 Server2 kibana[3026]: Kibana is currently running with legacy O>
Nov 03 17:37:21 Server2 kibana[3026]: {"log.level":"info","@timestamp":"2024-11>
Nov 03 17:37:22 Server2 kibana[3026]: Native global console methods have been o>
Nov 03 17:37:23 Server2 kibana[3026]: Enter password for the kibana keystore:
Nov 03 17:37:23 Server2 systemd[1]: kibana.service: Deactivated successfully.
Nov 03 17:37:23 Server2 systemd[1]: kibana.service: Consumed 1.974s CPU time.
lines 1-16/16 (END)
```



```
[erwin@centos ~]$ systemctl status logstash.service
● logstash.service - logstash
     Loaded: loaded (/usr/lib/systemd/system/logstash.service; enabled; preset:>
     Active: active (running) since Sun 2024-11-03 17:43:04 PST; 2s ago
   Main PID: 4532 (java)
      Tasks: 14 (limit: 10949)
     Memory: 184.2M
        CPU: 3.680s
     CGroup: /system.slice/logstash.service
             └─4532 /usr/share/logstash/jdk/bin/java -Xms1g -Xmx1g -Djava.awt.h>

Nov 03 17:43:04 centos systemd[1]: Started logstash.
Nov 03 17:43:04 centos logstash[4532]: Using bundled JDK: /usr/share/logstash/j>
```

Reflections:

Answer the following:

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

   The benefit of a log monitoring tool is that it allows for a real time threat detection which for big companies is very important to respond to a security threat as fast as possible, this also allows to have a more improved system performance as it allows us to analyze potential issues ahead of time.

Conclusions:

This activity teaches us on how to streamline the downloading and installation of a log monitoring tool with their appropriate configuration using the template folder on the playbook. Utilizing the templates folder on ansible allows us to preset many things after installation of tools so that we do not need to individually set it one by one.

Github Link: **Act10/roles/logstash/templates/logstash.conf.j2 at main · Moznaim/Act10**