Name: Khelvin P. Nicolas	**Date Performed: 10/28/2024**
Course/Section: CPE 212 - CPE31S2	**Date Submitted: 11/1/2024**
Instructor: Engr. Robin Valenzuela	**Semester and SY: 1st Sem 3rd Year**

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 it 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)

Assume that we've already set up our sample environment consisting of 3 nodes (1 Ubuntu Control Node, 2 CentOS and Ubuntu Managed Nodes). For the roles, we decided to use only one role which is the db for both Managed Nodes.

Installation of ELK (ElasticSearch, Kibana, LogStash)

To install ELK Stack using Ansible playbook, we must first configure several tasks to ensure that the package services are successfully working.

Create a new YAML file named routine.yml with the following content:

routine.yml

```
---
 - hosts: all
   become: true
   pre_tasks:

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

   - name: update repository index (Ubuntu)
     apt:
       update_cache: yes
     changed_when: false
     when: ansible_distribution == "Ubuntu"

 - hosts: db
   become: true
   roles:
     - db
```

Create a new directory roles/db/tasks and add the following main.yml file:

main.yml

```
---
 - name: Add GPG key for ElasticSearch (Ubuntu)
   tags: ubuntu
   apt_key:
     url: https://artifacts.elastic.co/GPG-KEY-elasticsearch
     state: present
   when: ansible_distribution == "Ubuntu"

 - name: Allow Port 9200 through Firewall (CentOS)
   firewalld:
     zone: public
     port: 9200/tcp
     permanent: yes
     state: enabled
     immediate: yes
   when: ansible_distribution == "CentOS"

 - name: Allow Port 9200 through Firewall (Ubuntu)
```

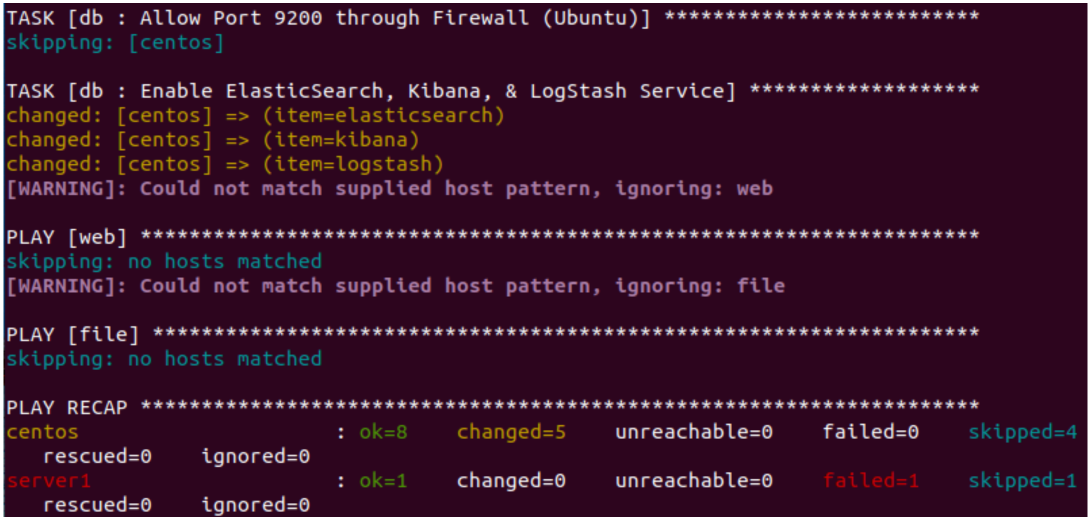
```
  ufw:
    rule: allow
    port: 9200
    proto: tcp
  when: ansible_distribution == "Ubuntu"

- name: Add ElasticSearch to APT repository (Ubuntu)
  tags: ubuntu
  apt_repository:
    repo: "deb https://artifacts.elastic.co/packages/7.x/apt stable main"
    #filename: 'elastic-7.x'
  when: ansible_distribution == "Ubuntu"

- name: Install ElasticSearch to Yum repository (CentOS)
  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: Configure ElasticSearch
  blockinfile:
    path: /etc/elasticsearch/elasticsearch.yml
    block: |
      # ElasticSearch Configuration

      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
    state: present
    create: yes

- name: Install ElasticSearch Kibana LogStash
  tags: ubuntu
  package:
    name:
      - elasticsearch
      - kibana
      - logstash
    state: latest

- name: Enable ElasticSearch, Kibana, & LogStash Service
  vars:
    elastic_services:
      - elasticsearch
      - kibana
      - logstash
  service:
     name: "{{ item }}"
     enabled: yes
     state: started
  loop: "{{ elastic_services }}"
```

This will install the ELK Stack alongside setting the correct configurations to correctly run the packages.

Add the following nodes to your inventory file
(for example, my two managed nodes uses hostname instead of IP Address):

```
inventory

[db]
server1
centos  ansible_user=<insert-user-name>
```

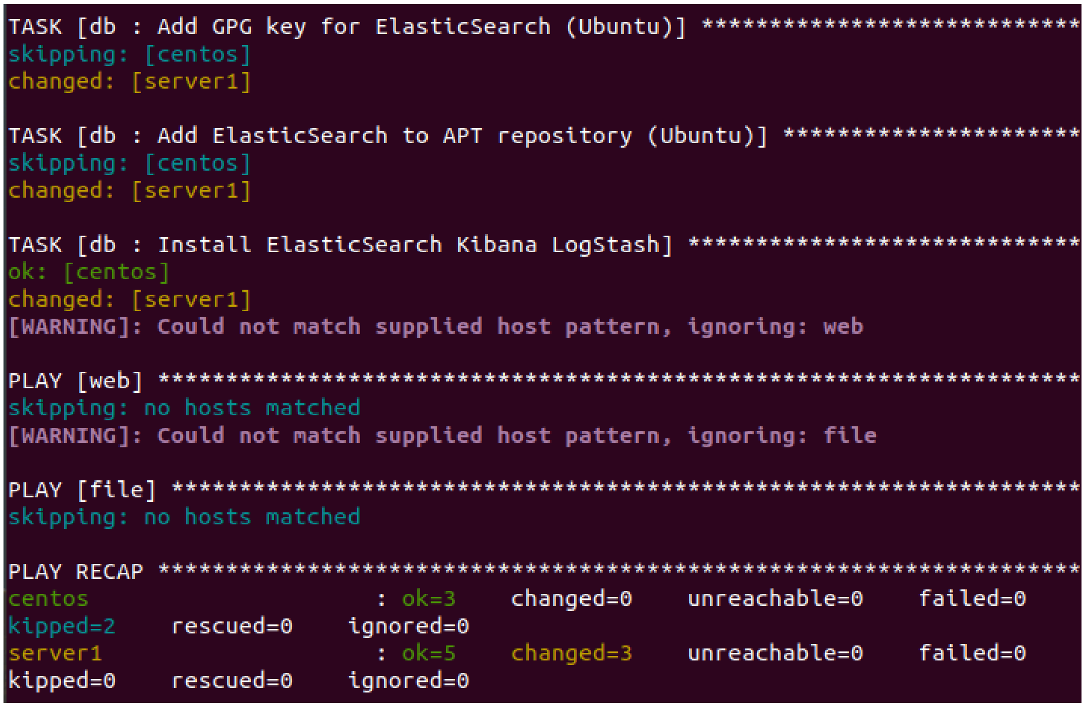
After we’ve set the correct configurations and files to their appropriate directory, we can now run the **routine.yml** playbook.



```
TASK [db : Allow Port 9200 through Firewall (Ubuntu)] **************************
skipping: [centos]

TASK [db : Enable ElasticSearch, Kibana, & LogStash Service] *******************
changed: [centos] => (item=elasticsearch)
changed: [centos] => (item=kibana)
changed: [centos] => (item=logstash)
[WARNING]: Could not match supplied host pattern, ignoring: web

PLAY [web] *********************************************************************
skipping: no hosts matched
[WARNING]: Could not match supplied host pattern, ignoring: file

PLAY [file] ********************************************************************
skipping: no hosts matched

PLAY RECAP *********************************************************************
centos                     : ok=8    changed=5    unreachable=0    failed=0    skipped=4
   rescued=0    ignored=0
server1                    : ok=1    changed=0    unreachable=0    failed=1    skipped=1
   rescued=0    ignored=0
```

Figure 1.1: Trial 1 of running the routine.yml playbook

As you can see in the summary that server1 failed one task. As we go on we may encounter errors after running the playbook, we can check them and try again with no worry of any duplicated installation issues since Ansible uses idempotency when running playbooks.



```
TASK [db : Add GPG key for ElasticSearch (Ubuntu)] ****************************
skipping: [centos]
changed: [server1]

TASK [db : Add ElasticSearch to APT repository (Ubuntu)] **********************
skipping: [centos]
changed: [server1]

TASK [db : Install ElasticSearch Kibana LogStash] *****************************
ok: [centos]
changed: [server1]
[WARNING]: Could not match supplied host pattern, ignoring: web

PLAY [web] ********************************************************************
skipping: no hosts matched
[WARNING]: Could not match supplied host pattern, ignoring: file

PLAY [file] *******************************************************************
skipping: no hosts matched

PLAY RECAP ********************************************************************
centos                     : ok=3    changed=0    unreachable=0    failed=0
kipped=2    rescued=0    ignored=0
server1                    : ok=5    changed=3    unreachable=0    failed=0
kipped=0    rescued=0    ignored=0
```

Figure 1.2: Trial 2 of running the routine.yml playbook

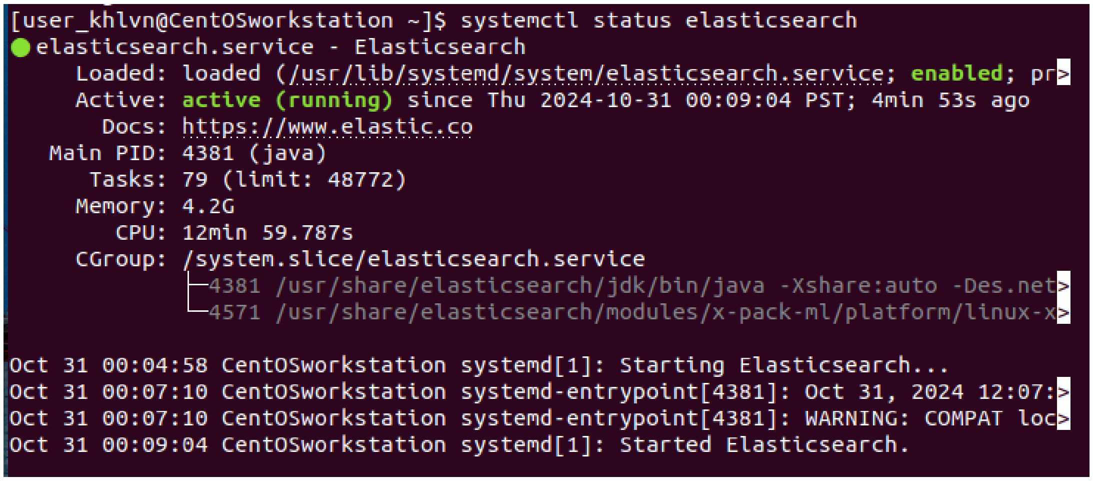
After fixing the errors and running the playbook again, the playbook successfully installed and configured ELK Stack to run in both managed nodes. Since we've already installed the packages, we can verify it in the terminal using the commands:

systemctl status elasticsearch
systemctl status kibana
systemctl status logstash

We can verify them on each of the nodes using the ***ssh user@hostname*** command on the terminal of the control node:

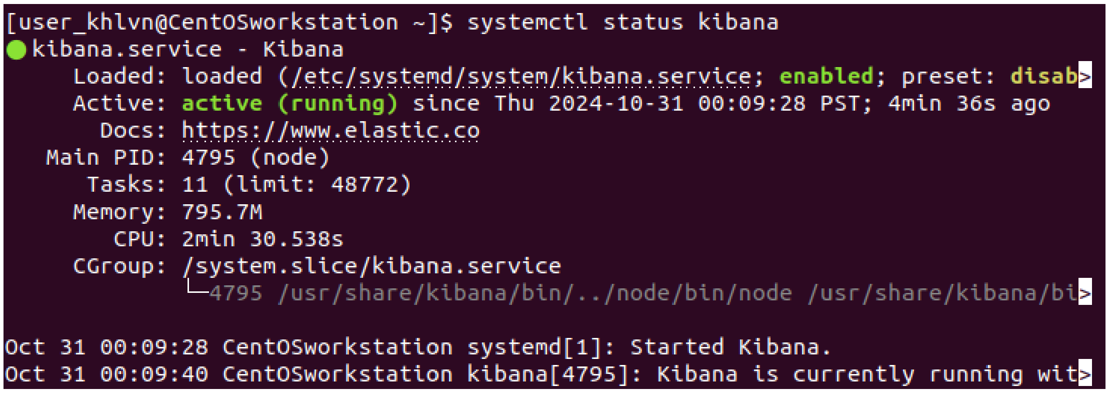
ssh punopaughey@server1
ssh user_khlvn@centos

CentOS (centos):



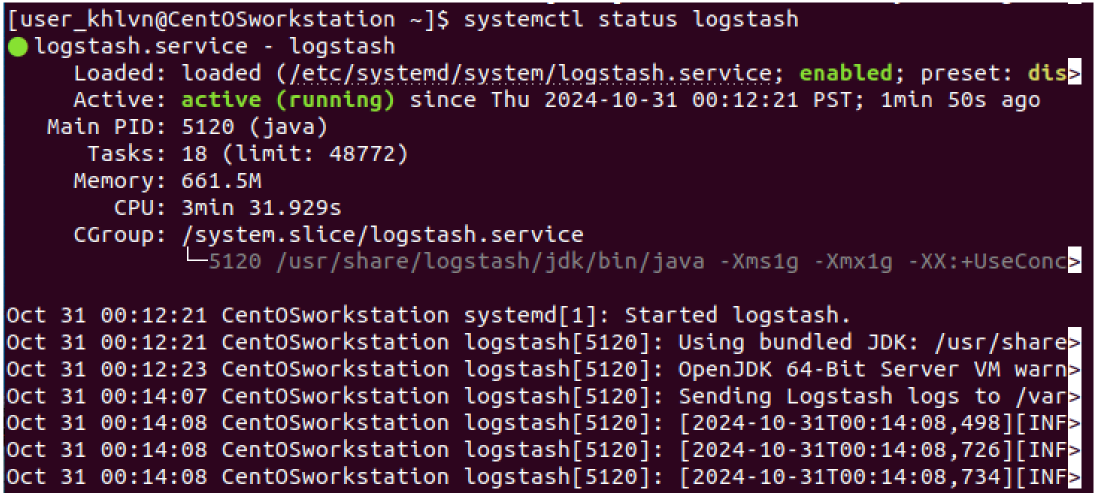
```
[user_khlvn@CentOSworkstation ~]$ systemctl status elasticsearch
● elasticsearch.service - Elasticsearch
     Loaded: loaded (/usr/lib/systemd/system/elasticsearch.service; enabled; pr>
     Active: active (running) since Thu 2024-10-31 00:09:04 PST; 4min 53s ago
       Docs: https://www.elastic.co
   Main PID: 4381 (java)
      Tasks: 79 (limit: 48772)
     Memory: 4.2G
        CPU: 12min 59.787s
     CGroup: /system.slice/elasticsearch.service
             ├─4381 /usr/share/elasticsearch/jdk/bin/java -Xshare:auto -Des.net>
             └─4571 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux-x>

Oct 31 00:04:58 CentOSworkstation systemd[1]: Starting Elasticsearch...
Oct 31 00:07:10 CentOSworkstation systemd-entrypoint[4381]: Oct 31, 2024 12:07:>
Oct 31 00:07:10 CentOSworkstation systemd-entrypoint[4381]: WARNING: COMPAT loc>
Oct 31 00:09:04 CentOSworkstation systemd[1]: Started Elasticsearch.
```

Figure 2.1: Verify ElasticSearch on CentOS



```
[user_khlvn@CentOSworkstation ~]$ systemctl status kibana
● kibana.service - Kibana
     Loaded: loaded (/etc/systemd/system/kibana.service; enabled; preset: disab>
     Active: active (running) since Thu 2024-10-31 00:09:28 PST; 4min 36s ago
       Docs: https://www.elastic.co
   Main PID: 4795 (node)
      Tasks: 11 (limit: 48772)
     Memory: 795.7M
        CPU: 2min 30.538s
     CGroup: /system.slice/kibana.service
             └─4795 /usr/share/kibana/bin/../node/bin/node /usr/share/kibana/bi>

Oct 31 00:09:28 CentOSworkstation systemd[1]: Started Kibana.
Oct 31 00:09:40 CentOSworkstation kibana[4795]: Kibana is currently running wit>
```

Figure 2.2: Verify Kibana on CentOS

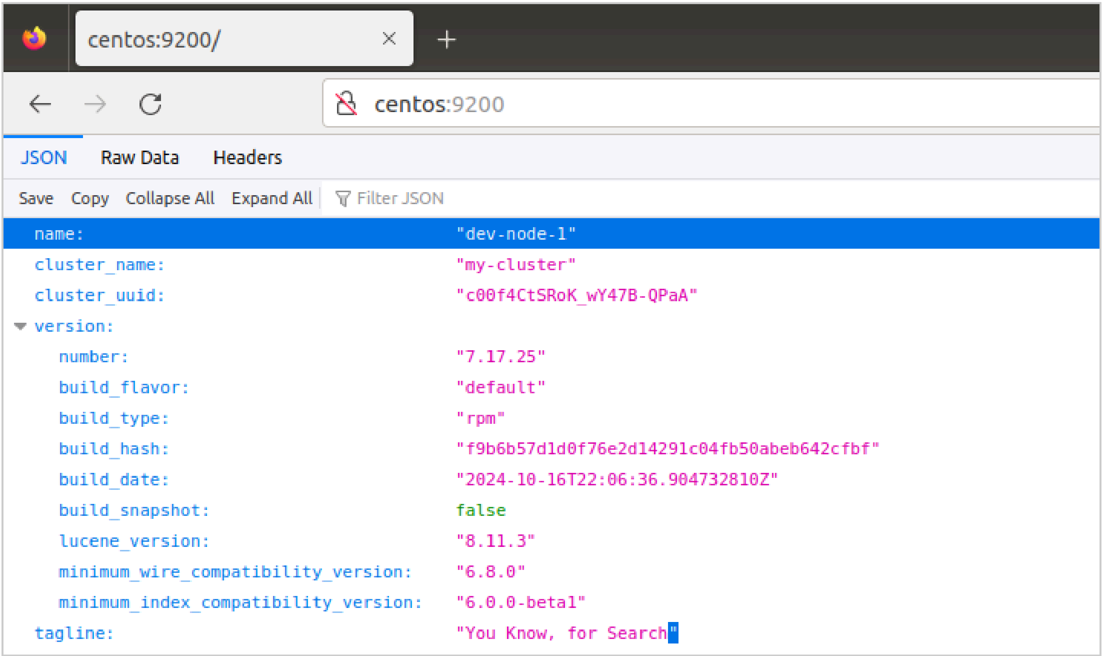


```
[user_khlvn@CentOSworkstation ~]$ systemctl status logstash
● logstash.service - logstash
     Loaded: loaded (/etc/systemd/system/logstash.service; enabled; preset: dis>
     Active: active (running) since Thu 2024-10-31 00:12:21 PST; 1min 50s ago
   Main PID: 5120 (java)
      Tasks: 18 (limit: 48772)
     Memory: 661.5M
        CPU: 3min 31.929s
     CGroup: /system.slice/logstash.service
             └─5120 /usr/share/logstash/jdk/bin/java -Xms1g -Xmx1g -XX:+UseConc>

Oct 31 00:12:21 CentOSworkstation systemd[1]: Started logstash.
Oct 31 00:12:21 CentOSworkstation logstash[5120]: Using bundled JDK: /usr/share>
Oct 31 00:12:23 CentOSworkstation logstash[5120]: OpenJDK 64-Bit Server VM warn>
Oct 31 00:14:07 CentOSworkstation logstash[5120]: Sending Logstash logs to /var>
Oct 31 00:14:08 CentOSworkstation logstash[5120]: [2024-10-31T00:14:08,498][INF>
Oct 31 00:14:08 CentOSworkstation logstash[5120]: [2024-10-31T00:14:08,726][INF>
Oct 31 00:14:08 CentOSworkstation logstash[5120]: [2024-10-31T00:14:08,734][INF>
```

Figure 2.3: Verify LogStash on CentOS

After verifying, we can also check if the ELK Stack can be accessed on port 9200 using any Web Browser (typically Firefox, etc.) by typing the hostname and port separated with colon.

centos:9200

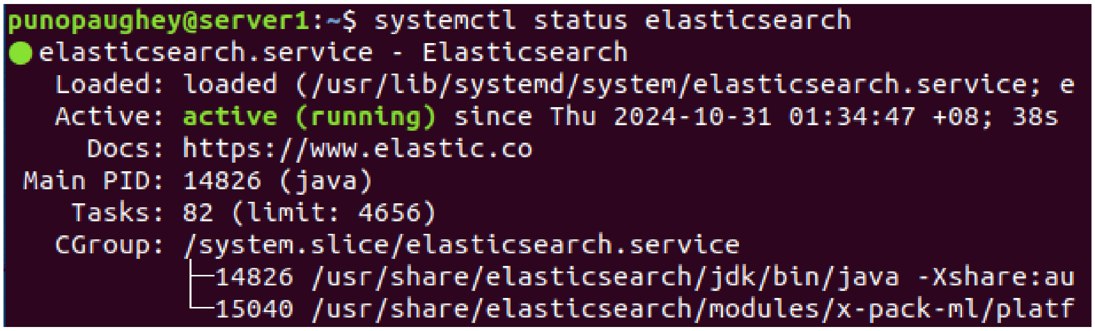


```
name:                               "dev-node-1"
cluster_name:                       "my-cluster"
cluster_uuid:                       "c00f4CtSRoK_wY47B-QPaA"
▼ version:
  number:                           "7.17.25"
  build_flavor:                     "default"
  build_type:                       "rpm"
  build_hash:                       "f9b6b57d1d0f76e2d14291c04fb50abeb642cfbf"
  build_date:                       "2024-10-16T22:06:36.904732810Z"
  build_snapshot:                   false
  lucene_version:                   "8.11.3"
  minimum_wire_compatibility_version:  "6.8.0"
  minimum_index_compatibility_version: "6.0.0-beta1"
tagline:                            "You Know, for Search"
```

Figure 2.4: Verify ELK Stack using Firefox on CentOS

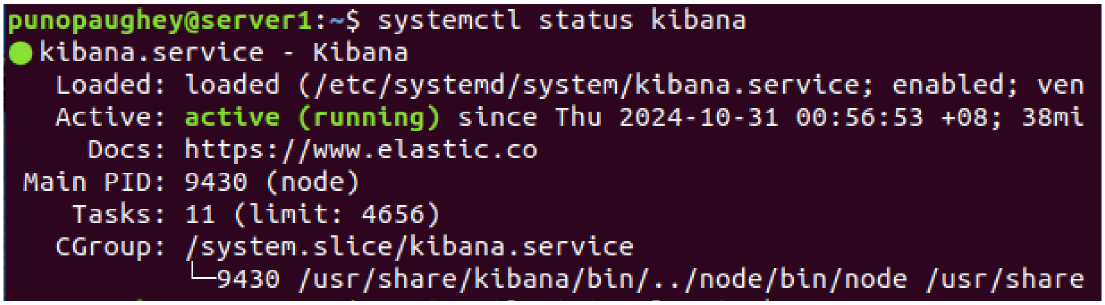
Ubuntu (server1):

We can deploy the same commands that we've used in verifying the ELK stack in centos server in Ubuntu.



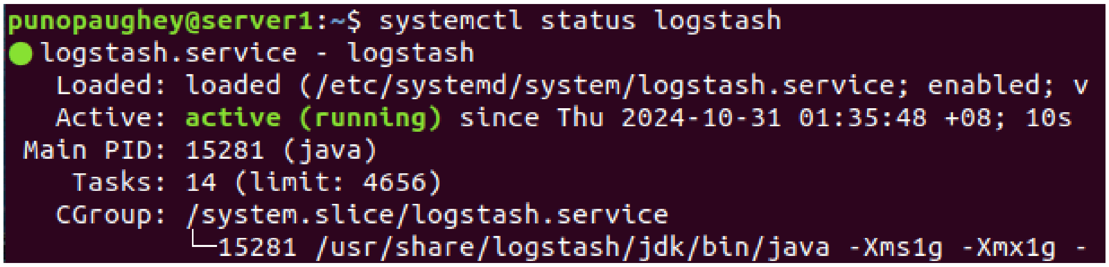
```
punopaughey@server1:~$ systemctl status elasticsearch
● elasticsearch.service - Elasticsearch
   Loaded: loaded (/usr/lib/systemd/system/elasticsearch.service; e
   Active: active (running) since Thu 2024-10-31 01:34:47 +08; 38s
     Docs: https://www.elastic.co
 Main PID: 14826 (java)
    Tasks: 82 (limit: 4656)
   CGroup: /system.slice/elasticsearch.service
           ├─14826 /usr/share/elasticsearch/jdk/bin/java -Xshare:au
           └─15040 /usr/share/elasticsearch/modules/x-pack-ml/platf
```

Figure 3.1: Verify ElasticSearch on Ubuntu



```
punopaughey@server1:~$ systemctl status kibana
● kibana.service - Kibana
   Loaded: loaded (/etc/systemd/system/kibana.service; enabled; ven
   Active: active (running) since Thu 2024-10-31 00:56:53 +08; 38mi
     Docs: https://www.elastic.co
 Main PID: 9430 (node)
    Tasks: 11 (limit: 4656)
   CGroup: /system.slice/kibana.service
           └─9430 /usr/share/kibana/bin/../node/bin/node /usr/share
```

Figure 3.2: Verify Kibana on Ubuntu

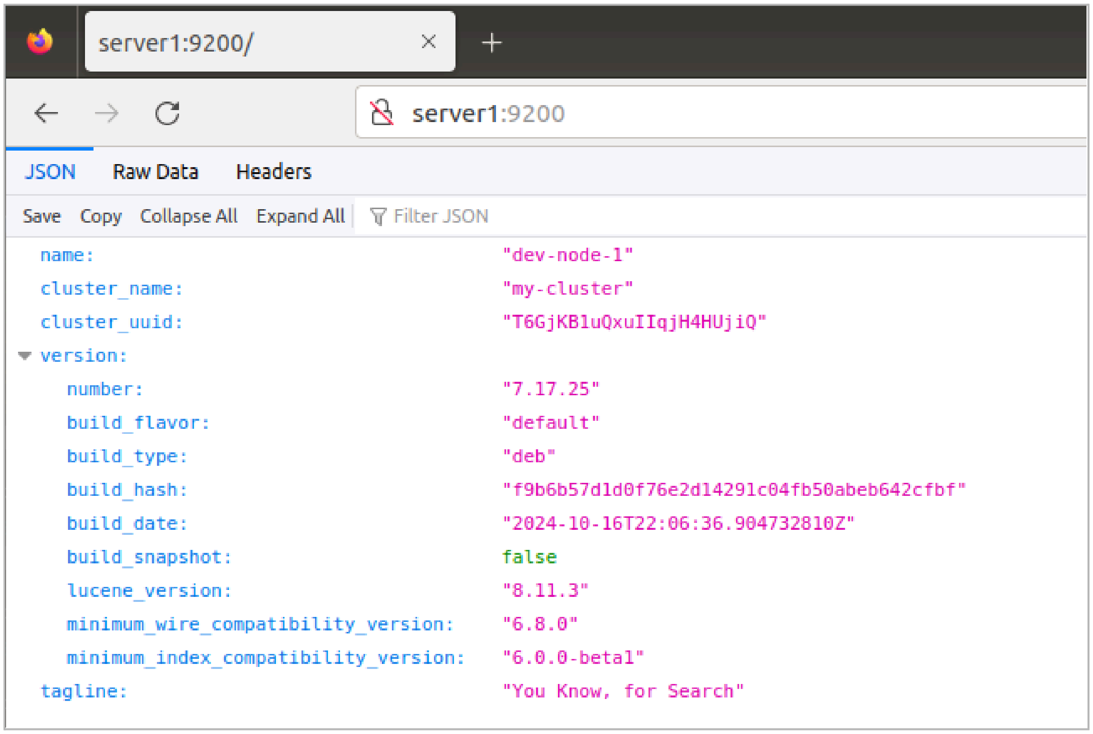


```
punopaughey@server1:~$ systemctl status logstash
● logstash.service - logstash
   Loaded: loaded (/etc/systemd/system/logstash.service; enabled; v
   Active: active (running) since Thu 2024-10-31 01:35:48 +08; 10s
 Main PID: 15281 (java)
    Tasks: 14 (limit: 4656)
   CGroup: /system.slice/logstash.service
           └─15281 /usr/share/logstash/jdk/bin/java -Xms1g -Xmx1g -
```

Figure 3.3: Verify LogStash on Ubuntu

After verifying, we can also check if the ELK Stack can be accessed on port 9200 using any Web Browser (typically Firefox, etc.) same as centos by typing the hostname and port separated with colon.

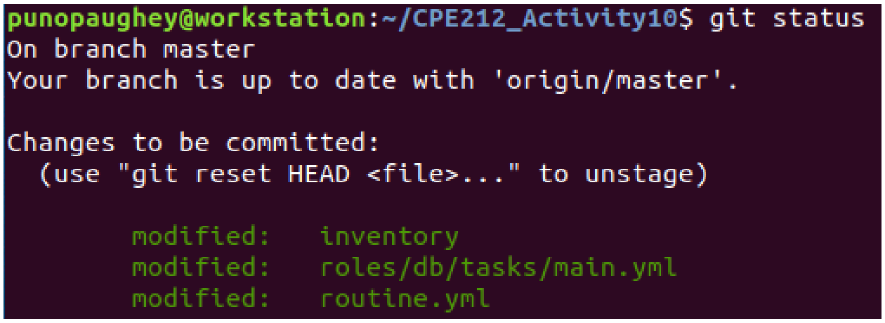
server1:9200



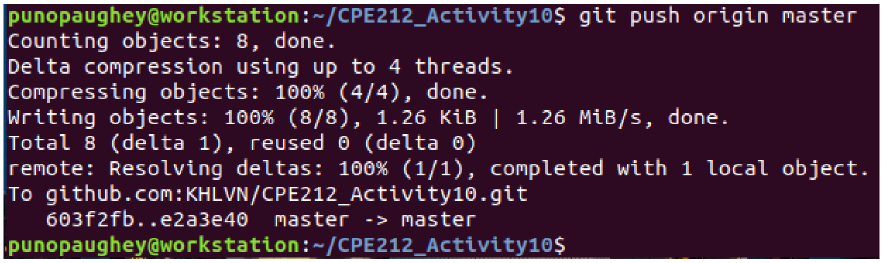
```
name:                                      "dev-node-1"
cluster_name:                              "my-cluster"
cluster_uuid:                              "T6GjKB1uQxuIIqjH4HUjiQ"
version:
  number:                                  "7.17.25"
  build_flavor:                            "default"
  build_type:                              "deb"
  build_hash:                              "f9b6b57d1d0f76e2d14291c04fb50abeb642cfbf"
  build_date:                              "2024-10-16T22:06:36.904732810Z"
  build_snapshot:                          false
  lucene_version:                          "8.11.3"
  minimum_wire_compatibility_version:      "6.8.0"
  minimum_index_compatibility_version:     "6.0.0-beta1"
tagline:                                   "You Know, for Search"
```

Figure 3.4: Verify ELK Stack using Firefox on Ubuntu

To conclude this activity, we must commit these changes and push it in our GitHub Repository.



```
punopaughey@workstation:~/CPE212_Activity10$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   inventory
        modified:   roles/db/tasks/main.yml
        modified:   routine.yml
```



```
punopaughey@workstation:~/CPE212_Activity10$ git push origin master
Counting objects: 8, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (8/8), 1.26 KiB | 1.26 MiB/s, done.
Total 8 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:KHLVN/CPE212_Activity10.git
   603f2fb..e2a3e40  master -> master
punopaughey@workstation:~/CPE212_Activity10$
```

Figure 4: committing and pushing to GitHub repo

Reflections:

Answer the following:

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

Implementing a log monitoring tool is crucial for any company that is seeking to enhance security, improve operational efficiency. Among the numerous log monitoring solutions available, the ELK Stack (Elasticsearch, Logstash, Kibana) stands out as a powerful and versatile tool. This popular open-source stack offers a robust framework for collecting, storing, and analyzing log data from diverse sources.

At the heart of the ELK Stack are three essential components: Elasticsearch, Logstash, and Kibana. Think of them as a powerful trio, working together to unlock the value in your log data. Elasticsearch is the brains of the operation, providing fast search and analytics capabilities. Logstash is the logistics expert, collecting logs from various sources, transforming them into a standardized format, and feeding them into Elasticsearch. And Kibana is like the storyteller, offering an interactive interface to explore, analyze, and visualize log data.

Together, these components enable organizations to centralize log management, automate analysis and alerting, and gain real-time visibility into system performance, security threats, and user behavior. With ELK Stack, the teams can identify potential security threats, troubleshoot application issues, and inform business decisions with data-driven insights.

Conclusions:

In this activity, I performed installation of a Log Monitoring Tool named ELK Stack into our managed nodes using Ansible only, performing this activity solidifies our knowledge about Ansible and meeting the objective of creating and designing a workflow that installs, configures and manages these tools into our network of systems. At first, I encountered minimal errors and easily fixed them. After that the workflow is smooth after running the playbook for the second time since it worked successfully.

GitHub Repository for this Activity:

https://github.com/KHLVN/CPE212_Activity10

https://github.com/KHLVN/CPE212_Activity10