Name: Ilagan, Carlo Hideki D.	**Date Performed: 10-28-2024**
Course/Section: CPE31S2	**Date Submitted: 11-3-2024**
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.

Github link: https://github.com/chilagan-github/HOA10.1-chilagan

4. **Output** (screenshots and explanations)

Tree directory:

```
hideki@workstation:~/HOA10.1-chilagan$ tree
.
├── ansible.cfg
├── elk.yml
├── inventory
├── README.md
└── roles
    ├── elasticsearch
    │   └── tasks
    │       ├── elasticsearch.yml.j2
    │       └── main.yml
    ├── kibana
    │   └── tasks
    │       ├── kibana.yml.j2
    │       └── main.yml
    └── logstash
        └── tasks
            ├── logstash.conf.j2
            └── main.yml
```

Ansible Playbook

Elasticsearch main.yml:

```
hideki@workstation:~/HOA10.1-chilagan$ cat roles/elasticsearch/tasks/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: elasticsearch
      state: restarted
```

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

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

This was the ansible used in order to install elasticsearch for the servers.

```
# 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
```

The elastic configuration after the installment of elastic search

Kibana yml files:

```yaml
---
- 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 }}"
```

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

```
# Kibana Configuration

# 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"]
```

In order to run the kibana.service, these are the queries that should be included, from the installment of kibana up to the restart of systemd for proper running of the said software. Configuration was also provided above.

Logstash yml file and configuration:

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

filter {
  # Add any filters here
}

output {
  elasticsearch {
    hosts => ["http://192.168.56.102:9200"]
    index => "logstash-%{+YYYY.MM.dd}"
  }
}
```

```yaml
- name: Install dependencies
  tags: logstash
  apt:
    name: gnupg
    state: present
    update_cache: yes
  become: yes

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

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

- name: Install Logstash
  tags: logstash
  apt:
    name: logstash
    state: present

- name: Start and Enable Logstash service
  tags: logstash
  systemd:
    name: logstash
    enabled: yes
    state: started
```

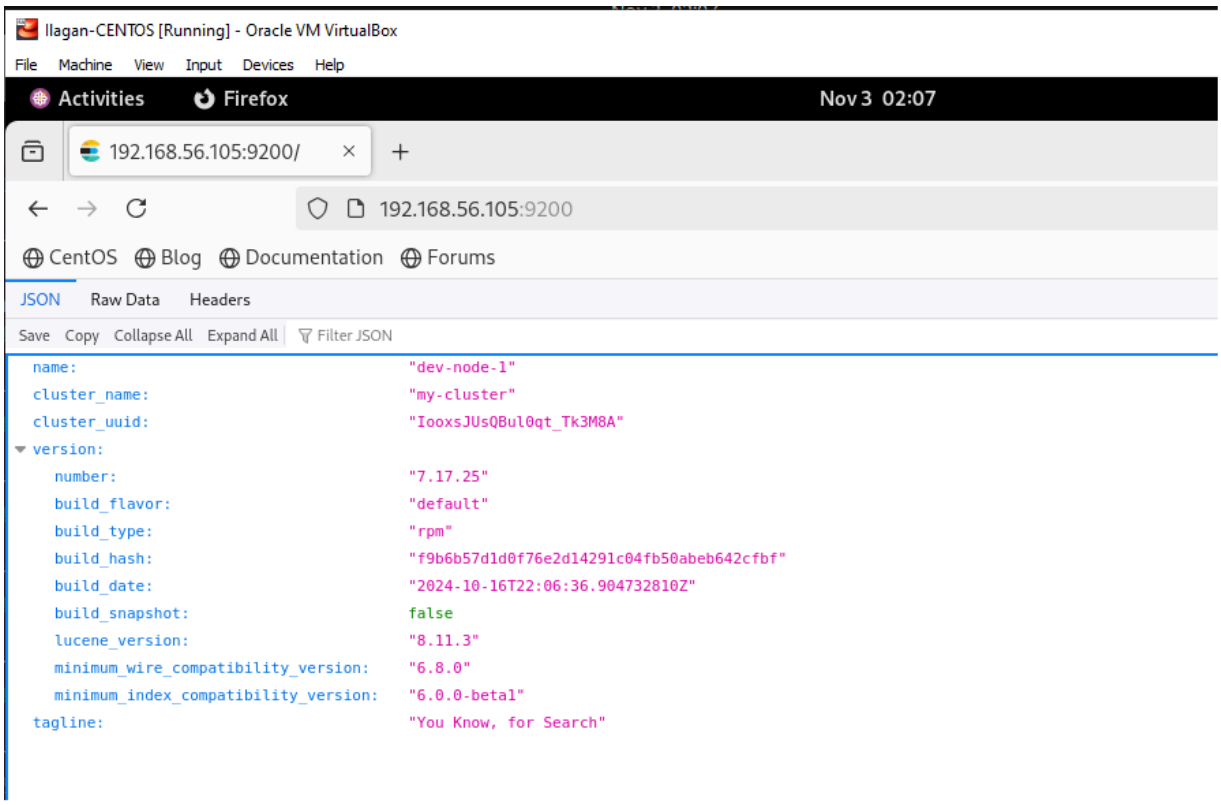
Lastly, these are the queries that were used in order to make the logstash available for the ubuntu. Note that these are two separate files, the first one is a j2 configuration file while the other one was the yml file.

Elk yml file:

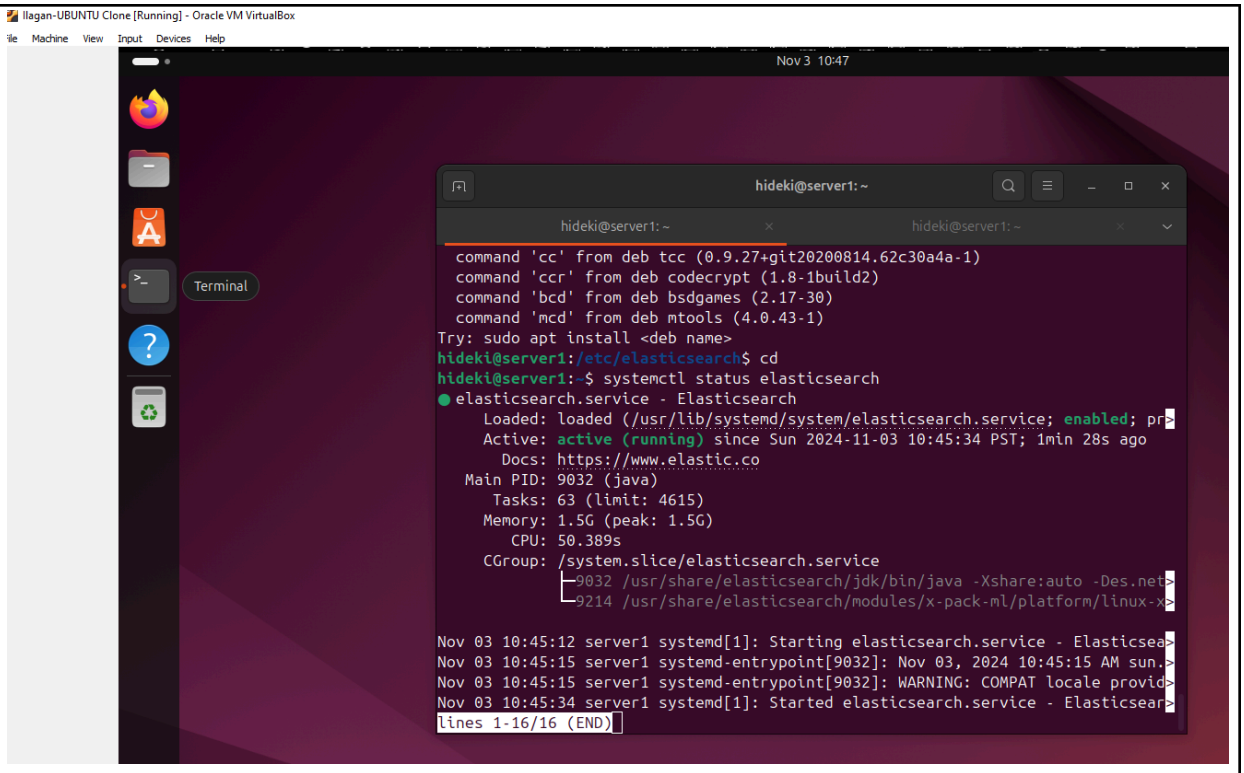
```yaml
---
- hosts: all
  become: true
  pre_tasks:

  - name: update repository index / install Updates (CentOS)
    tags: always
    dnf:
      update_cache: yes
    changed_when: false
    when: ansible_distribution == "CentOS"

  - name: update repository index / install Updates (Ubuntu)
    tags: always
    apt:
      update_cache: yes
    changed_when: false
    when: ansible_distribution == "Ubuntu"

- hosts: elasticsearch
  become: true
  roles:
   - elasticsearch

- hosts: kibana
  become: true
  roles:
   - kibana

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

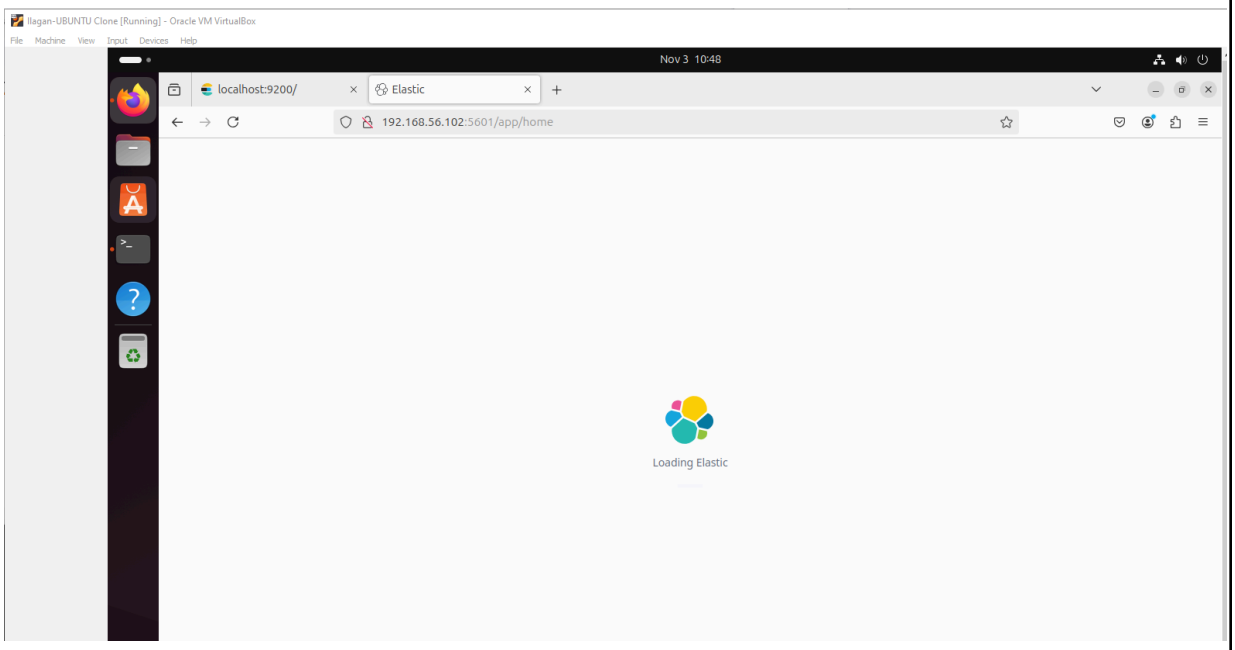
Elk yml file is one of the most important files in order to run the playbook, without this file, we will not be able to run all the activities performed above in just one command.

Output:



For the CentOS, we were able to successfully verify that the software was installed by using the IP address with the 9200 port which was the port for the elasticsearch.





As observed on the provided screenshots above, the elasticsearch software was successfully installed for the ubuntu. We were also able to access kibana using the IP address of the server together with the port used which was 5601.

```
hideki@server1:~$ systemctl status kibana
● kibana.service - Kibana
     Loaded: loaded (/etc/systemd/system/kibana.service; enabled; preset: enabl>
    Drop-In: /etc/systemd/system/kibana.service.d
             └override.conf
     Active: active (running) since Sun 2024-11-03 01:26:27 PST; 2min 21s ago
       Docs: https://www.elastic.co
   Main PID: 1508 (node)
      Tasks: 11 (limit: 4615)
     Memory: 383.0M (peak: 444.6M)
        CPU: 16.318s
     CGroup: /system.slice/kibana.service
             └1508 /usr/share/kibana/bin/../node/bin/node /usr/share/kibana/bi>

Nov 03 01:26:27 server1 systemd[1]: Started kibana.service - Kibana.
Nov 03 01:26:39 server1 kibana[1508]: Kibana is currently running with legacy O>

[1]+  Stopped                 systemctl status kibana
hideki@server1:~$ systemctl status logstash
● logstash.service - logstash
     Loaded: loaded (/etc/systemd/system/logstash.service; enabled; preset: ena>
     Active: active (running) since Sun 2024-11-03 01:29:14 PST; 4s ago
   Main PID: 4082 (java)
      Tasks: 15 (limit: 4615)
     Memory: 254.1M (peak: 254.3M)
        CPU: 7.605s
     CGroup: /system.slice/logstash.service
             └4082 /usr/share/logstash/jdk/bin/java -Xms1g -Xmx1g -XX:+UseConc>
```

Another screenshot provided to show that both logstash and Kibana were properly installed inside the ubuntu server.

Reflections:

Answer the following:

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

Log monitoring provides many benefits that help organizations and IT teams stay proactive in maintaining system health, security, and performance. One of the most obvious advantages is real-time incident detection. By monitoring logs continuously running, administrators can receive alerts for unusual or suspicious activity, allowing for quick responses to prevent potential outages, data breaches, or other issues. Enhanced security is another major benefit, as log monitoring can detect abnormal login attempts, unauthorized access, and unusual system activity, providing early warnings of possible security risks.

Conclusions:

Concluding this activity, I was able to download all the needed software for both Ubuntu and CentOS. Although the process is much harder than the previous activities. . While the installation processes are similar on both operating systems, there are minor differences in commands and configurations due to Ubuntu's use of apt and CentOS reliance on yum or dnf. I also learned the importance of having these kinds of tools especially as a student who is currently pursuing a degree majoring as an administrator. This kind of tool will pretty much help not only the company to reduce damage cost but also the team working in the field to prevent major issues such as outages and data breaches.