Name: VINCENT BRYAN BOSE	**Date Performed: 11/04/2024**
Course/Section: CPE31S2	**Date Submitted: 1/04/2024**
Instructor: ENGR ROBIN VALENZUELA	**Semester and SY: 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)

STEP 1:

```
vbbose@workstation:~/BOSE-HOA-10.1$ 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
```

STEP 2:

MAKE AN ELASTICSEARCH FOLDER

Open

elasticsearch.yml.j2
~/BOSE-HOA-10.1/roles/elasticsearch/tasks

```
# 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
```

Open

main.yml
~/BOSE-HOA-10.1/roles/elasticsearch/tasks

elasticsearch.yml.j2

```yaml
---
- 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
    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
```

STEP 3:

MAKE A KIBANA FOLDER

Open

kibana.yml.j2
~/BOSE-HOA-10.1/roles/kibana/tasks

elasticsearch.yml.j2 × main.yml

```
# 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: "192.168.56.106"

# Set the public base URL for Kibana
server.publicBaseUrl: "http://192.168.56.106:5601"

# Elasticsearch server URL
elasticsearch.hosts: ["http://192.168.56.113:9200"]
```

```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 }}"

- 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"
```

```yaml
- 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"
```

YAM

STEP 4:

MAKE A LOGSTASH FOLDER

Open ▾

logstash.conf.j2
~/BOSE-HOA-10.1/roles/logstash/ta

main.yml ×

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

filter {
  # Add any filters here
}

output {
  elasticsearch {
    hosts => ["http://192.168.56.113:9200"]
    index => "logstash-%{+YYYY.MM.dd}"
  }
}
```

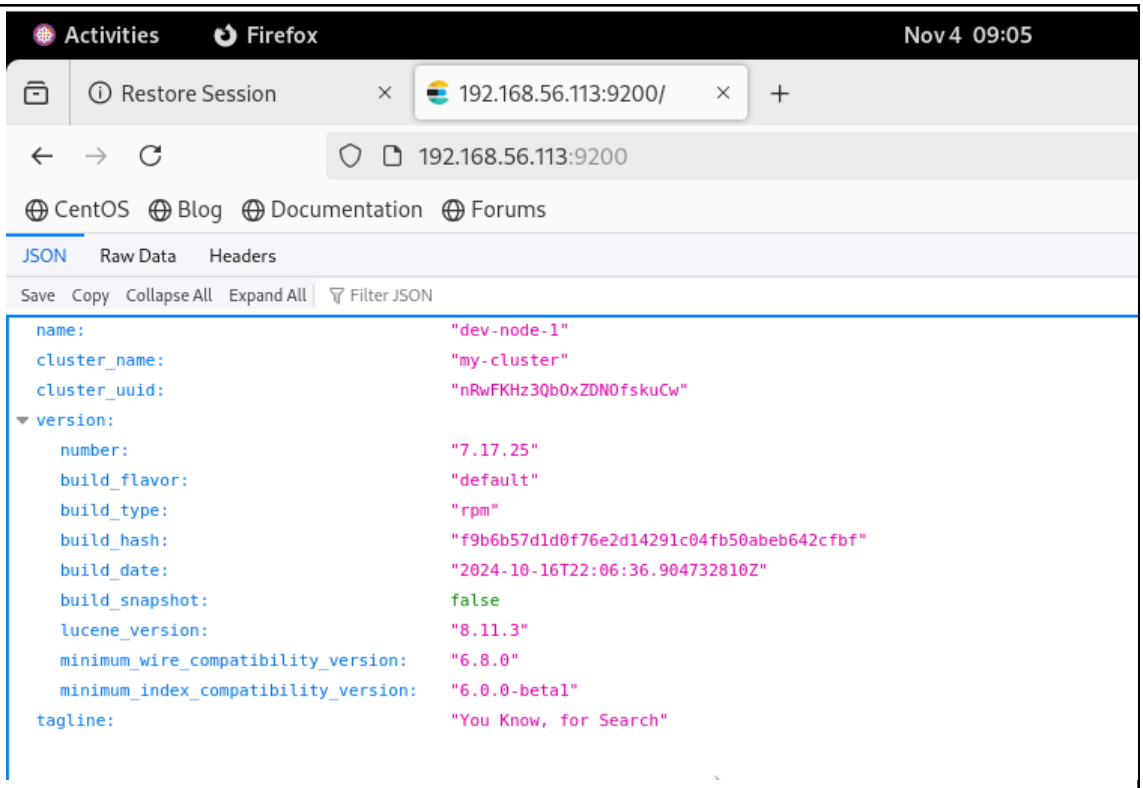
Open

main.yml
~/BOSE-HOA-10.1/roles/logstash/tasks

```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
```

STEP 5:

OUTPUT:



Reflections:

Answer the following:

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

   Integrating a log monitoring tool with an Ubuntu playbook offers numerous benefits that significantly enhance system reliability, security, and performance. The real-time tracking of logs enables immediate detection of issues and errors, while centralized logging simplifies management by aggregating data from multiple sources. Alerting and notification features allow for quick responses to potential problems, and continuous monitoring aids in identifying security incidents, assisting with regulatory compliance. Detailed log histories support forensic analysis and trend assessment, enhancing resilience and resource allocation. Furthermore, performance monitoring identifies inefficiencies, and automation with playbooks streamlines log collection and processing. This integration also improves troubleshooting efficiency by providing a centralized view of logs, accommodates scalability for growing log volumes, and offers customizable insights through tailored dashboards. In summary, using a log monitoring tool alongside an Ubuntu playbook greatly enhances system monitoring and operational efficiency, enabling proactive incident management and overall performance optimization.

Conclusions:

The experience of creating a workflow for managing enterprise log monitoring tools with Ansible as an Infrastructure as Code (IaC) tool has highlighted the advantages of automation, including improved workflow efficiency, deployment consistency, and scalable log management. Automation enhances system reliability and security while saving time and reducing human error, which is essential in complex enterprise settings. Ansible facilitates proactive incident response through continuous log monitoring and timely alerts for anomalies, crucial for maintaining performance in a fast-paced environment. This experience underscores the importance of automation in operational efficiency and security compliance, especially as organizations embrace digital transformation, with valuable insights for future projects.