Name: Baltazar, Paul Eimar R.	**Date Performed:** Oct 30, 2024
Course/Section: CPE212 – CPE31S2	**Date Submitted:** Nov 4, 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.

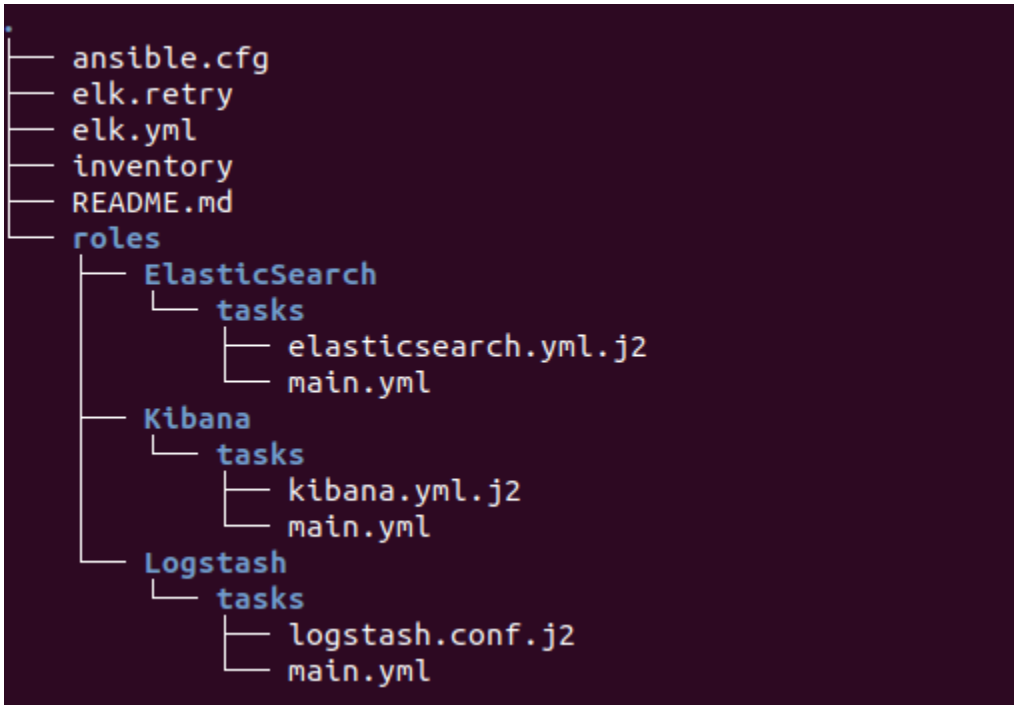
We use Graylog primarily as the stash for the logs of the web applications we build. However, it is also effective when working with raw strings (i.e. syslog): the tool parses it into the structured data we need. It also allows advanced custom search in the logs using structured queries. In other words, 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)



Create a directory which contains the following files shown above. Modify the ansible.cfg file to contain the following:

```
Unset
[defaults]
inventory = inventory
remote_user = paul_eimar
host_key_checking = True
```

Create an Inventory file and it should contain the following:

```
Unset
[kibana]
192.168.56.106
[logstash]
192.168.56.106
[elasticsearch]
192.168.56.109 ansible_user=pbaltazar
```

The inventory file contains the addresses of the remote servers which are grouped according to their assigned services

Create a yml file, this is going to be the main playbook that you will run. For this activity, I will name it elk.yml. It should contain the following:

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

- hosts: kibana
  become: true
  roles:
    - Kibana

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

We now proceed to the roles directory. Under the directory ElasticSearch/tasks/ There should be two files: main.yml and elasticsearch.yml.j2.

Unset

```
- 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
  yum:
    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
```

main.yml

```
Unset
# 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
```

elasticsearch.yml.j2

We proceed to the second service, Kibana. Under the directory roles/Kibana/tasks, There should be two files: main.yml and kibana.yml.j2

```
Unset
- 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
  tags: kibana
  service:
    name: kibana
    state: restarted
  become: yes
```

```
  when: ansible_distribution == "Ubuntu"
```

main.yml

```
Unset
# 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://192.168.56.109:9200"]
```

kibana.yml.j2

Finally, we have the final service, Logstash. Just like the two services, we also need to have two files inside its separate directory, which is in roles/Logstash/tasks. The two files are main.yml and logstash.yml.j2.

```
Unset
- 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
```

main.yml

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

filter {
  # Add any filters here
}

output {
  elasticsearch {
    hosts => ["http://192.168.56.109:9200"]
    index => "logstash-%{+YYYY.MM.dd}"
  }
}
```

logstash.yml.j2

After this, you will have to run the main playbook which is named elk.yml in the main directory. Run it with the command -ansible-playbook --ask-become-pass elk.yml. It should run without errors. To check if the services were installed and running, you can go to the remote computer and execute the command systemctl status <service name>

```
[pbaltazar@localhost ~]$ systemctl status elasticsearch
● elasticsearch.service - Elasticsearch
   Loaded: loaded (/usr/lib/systemd/system/elasticsearch.service; enabled; vendor preset: disabled)
   Active: active (running) since Sun 2024-11-03 19:58:59 EST; 4min 24s ago
     Docs: https://www.elastic.co
 Main PID: 1182 (java)
    Tasks: 75
   CGroup: /system.slice/elasticsearch.service
           ├─1182 /usr/share/elasticsearch/jdk/bin/java -Xshare:auto -Des.net...
           └─2446 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux-x...

Nov 03 19:58:18 localhost.localdomain systemd[1]: Starting Elasticsearch...
Nov 03 19:58:40 localhost.localdomain systemd-entrypoint[1182]: Nov 03, 2024 ...
Nov 03 19:58:40 localhost.localdomain systemd-entrypoint[1182]: WARNING: COMP...
Nov 03 19:58:59 localhost.localdomain systemd[1]: Started Elasticsearch.
Hint: Some lines were ellipsized, use -l to show in full.
```

Elasticsearch

```
paul_eimar@Server1:~$ systemctl status kibana
● kibana.service - Kibana
   Loaded: loaded (/etc/systemd/system/kibana.service; disabled; vendor preset:
  Drop-In: /etc/systemd/system/kibana.service.d
           └─override.conf
   Active: active (running) since Mon 2024-11-04 08:13:17 +08; 14min ago
     Docs: https://www.elastic.co
 Main PID: 6196 (node)
    Tasks: 11 (limit: 4915)
   CGroup: /system.slice/kibana.service
           └─6196 /usr/share/kibana/bin/../node/bin/node /usr/share/kibana/bin/.
```

Kibana

```
paul_eimar@Server1:~$ systemctl status logstash
● logstash.service - logstash
   Loaded: loaded (/etc/systemd/system/logstash.service; enabled; vendor preset:
   Active: active (running) since Mon 2024-11-04 08:27:30 +08; 7s ago
 Main PID: 10368 (java)
    Tasks: 22 (limit: 4915)
   CGroup: /system.slice/logstash.service
           └─10368 /usr/share/logstash/jdk/bin/java -Xms1g -Xmx1g -XX:+UseConcMa
```

Logstash

Alternatively, you can also check if the services are running by going into the web browser, typing the computer's ip address with the port 9200 (for Elasticsearch) and 5601 (for Kibana)

192.168.56.109:9200

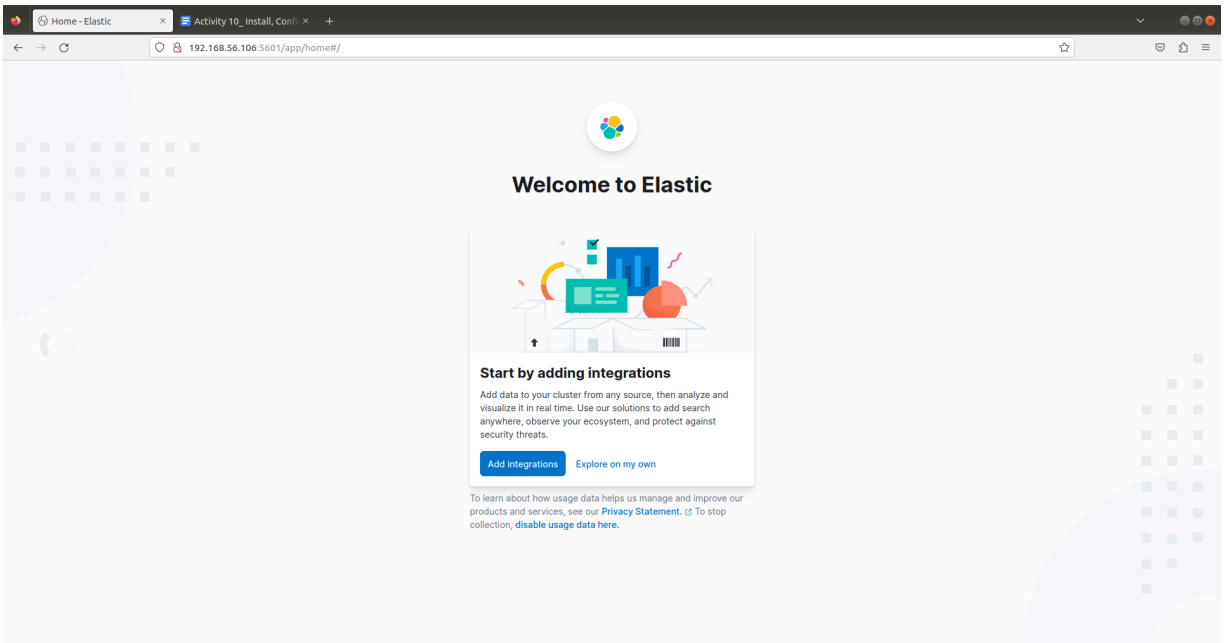
Centos Wiki Documentation Forums

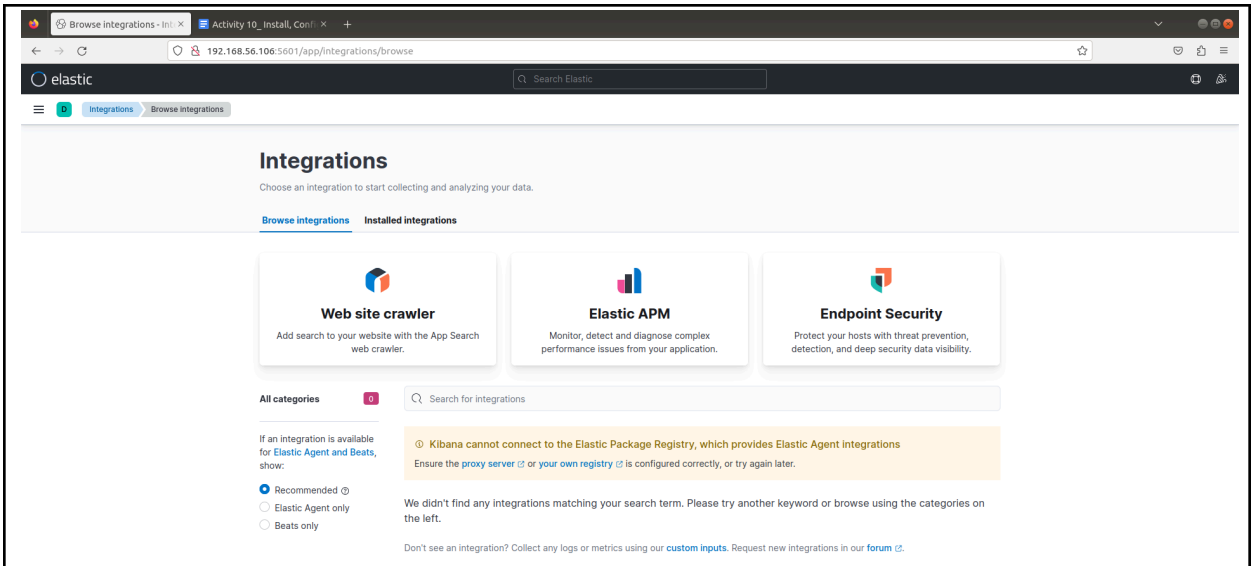
JSON Raw Data Headers

Save Copy Collapse All Expand All Filter JSON


```
name:                                    "dev-node-1"
cluster_name:                            "my-cluster"
cluster_uuid:                            "Q-Vb7jbWRzGP-57Fjc8IBw"
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"
```


Elastic Search





Kibana

Reflections:

Answer the following:

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

   Using an availability monitoring solution for managing Ubuntu servers offers several benefits, including early issue detection, access to real-time performance data, and prompt alerts that help minimize downtime. These tools aid in capacity planning and resource optimization by analyzing historical data, which in turn enhances the user experience. They provide a comprehensive approach to maintaining server performance and reliability, simplify compliance reporting, and often integrate seamlessly with other management tools.

Conclusions:

In this exercise, I successfully installed and set up the Elastic Stack, which includes Kibana, Logstash, and Elasticsearch. This powerful toolkit provides an all-in-one solution for managing logs and metrics in a centralized way.

Elasticsearch, the main data store, efficiently organizes and saves large amounts of time-based data, making it ideal for troubleshooting and analyzing system performance. Kibana, the tool's visualization layer, lets users create interactive dashboards and charts to gain useful insights

from collected data. Logstash, which serves as the data pipeline, gathers, processes, and enriches log data before sending it to Elasticsearch.