UNIVERSITY OF SCIENCE AND TECHNOLOGY OF HANOI

CYBER SECURITY

Web security

Report

Study ELK Stack for log management: overview, architecture, components, functionality. Install ELK Stack and carry out experiments on log collection and analysis.

Submitter by

Group 7

BA12-081: Phạm Phú Hưng

Lectures: Prof. Hoàng Xuân Dậu

Table of Contents

1.Introduction

1.1 Purpose and Scope

Logs are a critical component of any IT environment, capturing events and metrics from systems, applications, and networks. This report provides an in-depth look at the **ELK Stack**—a popular open-source solution for log management and analysis. We will examine its architecture, features, pros and cons, and walk through a practical demonstration of its installation and usage. By the end of this report, readers will have a thorough understanding of how ELK can be used to centralize logs, perform real-time analysis, and derive insights to enhance system reliability and security.

1.2 Importance of Log Management

Modern organizations increasingly rely on software applications and distributed systems to conduct their business. Logs generated by these applications and systems are vital for:

Troubleshooting: Quickly identifying the root cause of errors and performance issues.

Security and Compliance: Detecting malicious activity, unauthorized access, and meeting auditing requirements.

Performance Optimization: Analyzing system usage and resource bottlenecks to improve throughput and stability.

Without centralized and efficient log management, organizations face data silos, inconsistent log structures, and missed opportunities for proactive monitoring.

1.3 Overview of the ELK Stack

The **ELK Stack** is composed of three main components:

Elasticsearch: A powerful search and analytics engine for storing and querying data.

Logstash: A data pipeline tool for collecting, parsing, and transforming log data from various sources.

Kibana: A visualization and dashboard interface to explore and analyze data stored in Elasticsearch.

Additionally, the **Beats** platform (Filebeat, Metricbeat, Packetbeat, etc.) often complements the ELK Stack by providing lightweight data shippers for different types of data and environments.

2. Architecture

2.1 Core Components of ELK

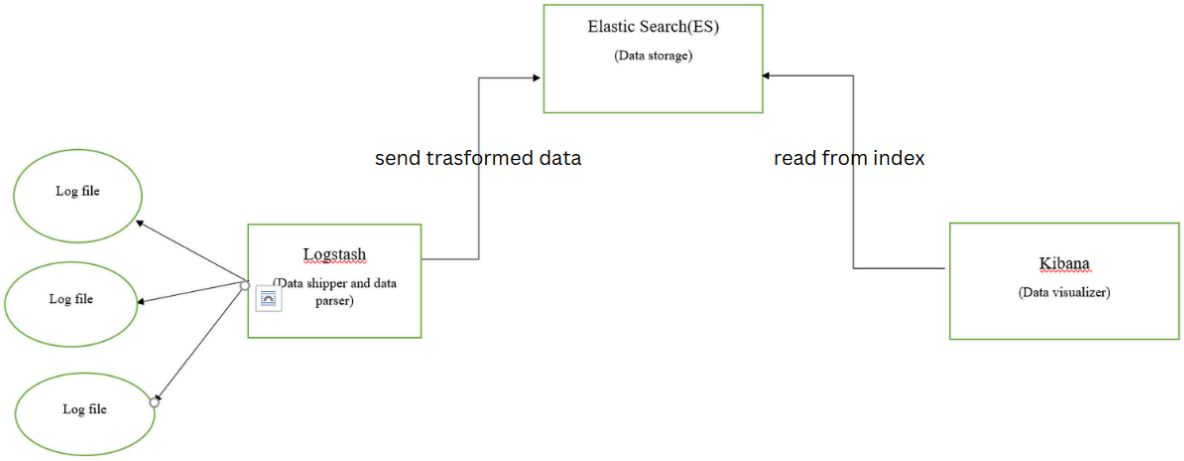
2.1.1 Elasticsearch

Role: Centralized storage and indexing engine.

Key Features:

Distributed storage enabling horizontal scaling.
Near real-time search and analytics.
RESTful APIs using JSON for data interchange.
Resilient clustering for fault tolerance.

Data flow in ELK stack



Usage: Receives data from Logstash (or Beats), indexes documents, and makes them available for complex queries.

2.1.2 Logstash

Role: Ingestion and transformation pipeline.

Key Features:

A plugin-based architecture (input, filter, output).
Supports various protocols (TCP, UDP, HTTP, Syslog, Beats, etc.).
Rich data processing capabilities (grok, mutate, geoIP, etc.).
Buffering and retry mechanisms for reliability.

Usage: Receives raw data from different sources, parses and enriches it, then outputs transformed data to Elasticsearch or other destinations.

2.1.3 Kibana

Role: Visualization and exploration interface.

Key Features:

Interactive dashboards with real-time charts and graphs.
Advanced analytics through time-series visualizations, maps, and custom queries.
Developer tools for experimenting with Elasticsearch queries.
Alerting and reporting (with additional plugins/X-Pack features).

Usage: Presents data in an intuitive visual format to help users discover trends, anomalies, and actionable insights.

2.2 Optional Components (Beats)

Filebeat: Tracks and ships log files in near real-time.

Metricbeat: Collects system and service metrics.

Packetbeat: Monitors network traffic.

Winlogbeat: Collects Windows event logs.

Auditbeat: Collects data from the Linux Audit Framework.

Beats are specialized shippers designed to run on endpoints, pushing data directly to Elasticsearch or via Logstash.

2.3 Data Flow in the ELK Stack

Data Generation: Applications, operating systems, or network devices generate logs.
Collection (Logstash or Beats): Logs are sent to Logstash or directly to Elasticsearch using Beats.
Processing (Logstash Filters): Logstash applies parsing, enrichment (e.g., geoIP lookups, user agent analysis), and metadata extraction.
Storage (Elasticsearch): Parsed logs are indexed and stored for search and retrieval.
Visualization (Kibana): Users create dashboards and run queries to analyze the logs, detect patterns, and set alerts.

3. Features

3.1 Centralized Log Management

Consolidates data from numerous servers, applications, and devices into one searchable repository.

Eliminates the need to log into multiple machines to troubleshoot issues.

Simplifies compliance by providing a single source of truth.

3.2 Real-Time Processing and Analysis

Elasticsearch delivers near real-time indexing and retrieval.

Dashboards in Kibana reflect new data almost immediately.

Rapid feedback loop for operations and security teams to detect and respond to events.

3.3 Scalability and High Availability

Elasticsearch can scale horizontally by adding more nodes to the cluster.

Logstash pipelines can be load-balanced to handle increased ingestion rates.

Built-in replication in Elasticsearch ensures data availability even if individual nodes fail.

3.4 Extensibility and Customization

Logstash plugins provide a range of input, filter, and output options (e.g., AWS S3, JDBC, Kafka).

Kibana plugins can add custom visualizations or integrations.

Beats allow tailored data ingestion for different operating systems, log formats, and metrics.

Advanced users can develop custom Beats or Logstash plugins.

4. Pros and Cons

4.1 Advantages

1. Open Source and Community-Driven
   Wide adoption, extensive documentation, and rapid feature development.

2. Flexible and Extensible
   Plugin-based architecture for ingesting, filtering, and outputting data from various sources.

3. Powerful Real-Time Analytics
   Enables quick troubleshooting and immediate insights.

4. Horizontal Scalability
   Grow the cluster as data volume and user demand increase.

5. Large Ecosystem of Tools and Integrations
   Compatible with DevOps tools (Docker, Kubernetes, Ansible, etc.) for smoother deployments.

4.2 Limitations

1. Resource Intensive
   Elasticsearch and Logstash can consume significant CPU, RAM, and storage resources at scale.

2. Complex Configuration
   Requires knowledge of YAML or configuration files; tuning for performance can be challenging.

3. Security Features
   Some advanced security and monitoring features (e.g., fine-grained access control, auditing) may require additional licensing or plugins.

4. Index Management
   Without proper lifecycle policies (e.g., rollover, deletion), indexes can become very large and impact performance.

5. Upgrades Can Be Tricky
   Version incompatibilities can occur; maintenance windows need careful planning.

5. Installation

5.1 Prerequisites

Hardware: Sufficient RAM (at least 4GB for test setups, more for production) and disk space.

Java Version: Logstash and Elasticsearch run on Java (bundled by default in most recent distributions).

OS Compatibility: ELK supports Windows, Linux, and macOS; Linux is most common for production.

Network Accessibility: Ensure ports are open (e.g., 9200 for Elasticsearch, 5601 for Kibana, 5044 or others for Logstash Beats input).

5.2 Installing And Configure Elasticsearch

Install

We will start adding the repository by importing the GPG key.

Open a terminal window and type the following command to import the Elastic GPG key:

curl -fsSL https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -

The output would display **OK** as shown in the screenshot below.

```
root@ubuntu:/home/ubuntu# curl -fsSL https://artifacts.elastic.co/GPG-KEY-elasti
csearch | sudo apt-key add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (s
ee apt-key(8)).
OK
```

Once the Elastic GPG key is imported, install the **apt-transport-https** package with the following **apt-get** command:

sudo apt-get install apt-transport-https

```
root@ubuntu:/home/ubuntu# sudo apt-get install apt-transport-https
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apt-transport-https is already the newest version (2.7.14build2).
0 upgraded, 0 newly installed, 0 to remove and 398 not upgraded.
```

Next, add the Elastic repository to your system's repository list by executing the following command:

echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" | sudo tee /etc/apt/sources.list.d/elastic-7.x.list

```
root@ubuntu:/home/ubuntu# echo "deb https://artifacts.elastic.co/packages/7.x/ap
t stable main" | sudo tee /etc/apt/sources.list.d/elastic-7.x.list
deb https://artifacts.elastic.co/packages/7.x/apt stable main
```

sudo apt-get update && sudo apt-get install elasticsearch

```
root@ubuntu:/home/ubuntu# echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" | sudo tee /etc/apt/sourc
es.list.d/elastic-7.x.list
deb https://artifacts.elastic.co/packages/7.x/apt stable main
root@ubuntu:/home/ubuntu# sudo apt-get update && sudo apt-get install elasticsearch
Ign:1 cdrom://Ubuntu 24.04.1 LTS _Noble Numbat_ - Release amd64 (20240827.1) noble InRelease
Hit:2 cdrom://Ubuntu 24.04.1 LTS _Noble Numbat_ - Release amd64 (20240827.1) noble Release
Hit:3 https://artifacts.elastic.co/packages/7.x/apt stable InRelease
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Hit:5 http://archive.ubuntu.com/ubuntu noble InRelease
Get:6 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:8 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [615 kB]
Get:10 http://archive.ubuntu.com/ubuntu noble-updates/main i386 Packages [410 kB]
```

Configure

Edit **/etc/elasticsearch/elasticsearch.yml:**
yaml

network.host: localhost

Http.port: 9200

discovery.type: single-node

```
GNU nano 7.2                                  /etc/elasticsearch/elasticsearch.yml
#
# ---------------------------------- Network -----------------------------------
#
# By default Elasticsearch is only accessible on localhost. Set a different
# address here to expose this node on the network:
#
network.host: localhost
#
# By default Elasticsearch listens for HTTP traffic on the first free port it
# finds starting at 9200. Set a specific HTTP port here:
#
http.port: 9200
#
# For more information, consult the network module documentation.
#
# --------------------------------- Discovery ----------------------------------
#

# Pass an initial list of hosts to perform discovery when this node is started:
# The default list of hosts is ["127.0.0.1", "[::1]"]
#
#discovery.seed_hosts: ["host1", "host2"]
#
# Bootstrap the cluster using an initial set of master-eligible nodes:
#
#cluster.initial_master_nodes: ["node-1", "node-2"]
#
# For more information, consult the discovery and cluster formation module documentation.
#
discovery.type: single-node
# ---------------------------------- Various -----------------------------------
```

Start and Enable Service

Now start and enable the Elasticsearch service using this systemctl command:
sudo systemctl enable elasticsearch

sudo systemctl start elasticsearch

```
root@ubuntu:/home/ubuntu# sudo systemctl enable elasticsearch
Synchronizing state of elasticsearch.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable elasticsearch
```

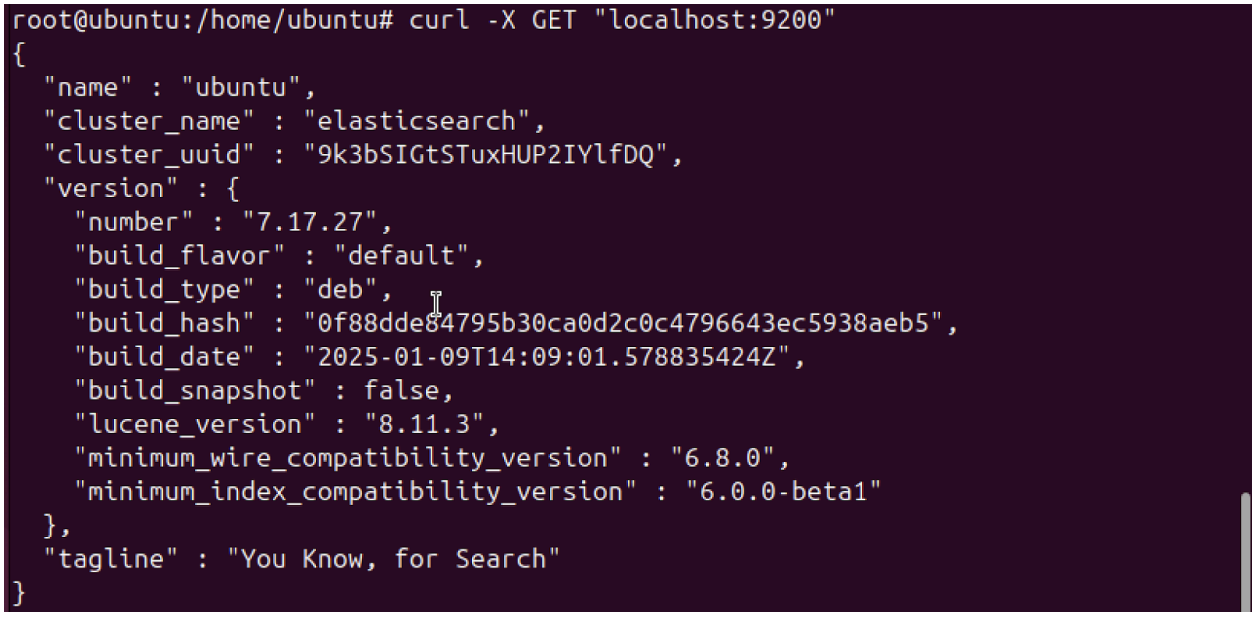
sudo systemctl status elasticsearch

```
root@ubuntu:/home/ubuntu# sudo systemctl status elasticsearch
● elasticsearch.service - Elasticsearch
     Loaded: loaded (/usr/lib/systemd/system/elasticsearch.service; enabled; preset: enabled)
     Active: active (running) since Tue 2025-01-28 18:29:33 UTC; 2 days ago
       Docs: https://www.elastic.co
   Main PID: 8108 (java)
      Tasks: 151 (limit: 18751)
     Memory: 1.4G (peak: 1.4G)
        CPU: 56min 26.399s
     CGroup: /system.slice/elasticsearch.service
             ├─8108 /usr/share/elasticsearch/jdk/bin/java -Xshare:auto -Des.networkaddress.cache.ttl=60 -Des.networkadd>
             └─8364 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux-x86_64/bin/controller

Jan 28 18:27:12 ubuntu systemd[1]: Starting elasticsearch.service - Elasticsearch...
Jan 28 18:27:46 ubuntu systemd-entrypoint[8108]: Jan 28, 2025 6:27:46 PM sun.util.locale.provider.LocaleProviderAdapter>
Jan 28 18:27:46 ubuntu systemd-entrypoint[8108]: WARNING: COMPAT locale provider will be removed in a future release
Jan 28 18:29:33 ubuntu systemd[1]: Started elasticsearch.service - Elasticsearch.
lines 1-16/16 (END)
```

Verify

Check curl -X GET http://localhost:9200 to confirm a JSON response indicating Elasticsearch is running.



5.3 Installing And Configure Kibana

Install

The next component we will install for the ELK stack is Kibana that serves as a user-friendly interface for analyzing and visualizing the data in the system.

Start by executing the following command to install Kibana:
sudo apt-get install kibana

Configure

open the kibana.yml file in your preferred text editor:

sudo nano /etc/kibana/kibana.yml

Locate the following lines in the file and remove the # sign to uncomment them.

server.port: 5601

server.host: "0.0.0.0"

elasticsearch.hosts: ["http://localhost:9200"]

```
  GNU nano 7.2                  /etc/kibana/kibana.yml *
# Kibana is served by a back end server. This setting specifies the port to use.
server.port: 5601

# Specifies the address to which the Kibana server will bind. IP addresses and >
# The default is 'localhost', which usually means remote machines will not be a>
# To allow connections from remote users, set this parameter to a non-loopback >
server.host: "localhost"
```

```
# The Kibana server's name.  This is used for display purposes.
server.name: "your-hostname"

# The URLs of the Elasticsearch instances to use for all your queries.
elasticsearch.hosts: ["http://localhost:9200"]
```

Note: This setup permits traffic from the same system where Elastic Stack is set up. You can adjust the **server.host** value to match the address of a remote server if needed.

Start and Enable Service
Next start and enable Kibana.

Run the following command to start the Kibana service:

sudo systemctl start kibana

No output will be produced if you have successfully started the service.

Next, enable Kibana to start automatically when the system boots:
sudo systemctl enable kibana

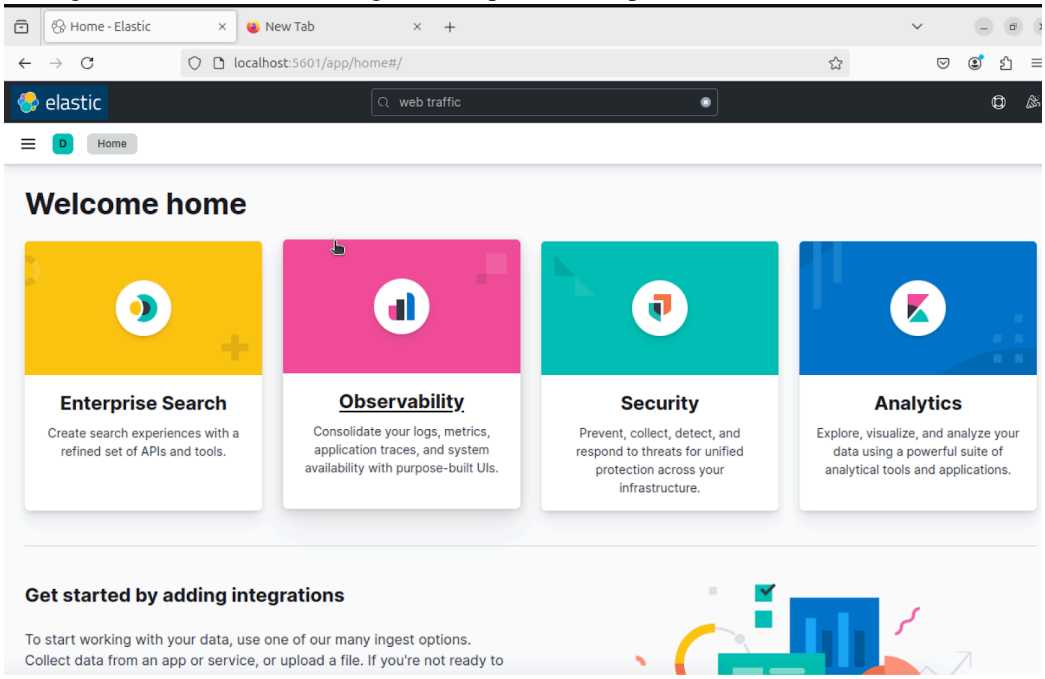
```
root@ubuntu:/home/ubuntu# sudo systemctl enable kibana
Synchronizing state of kibana.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable kibana
```

If the UFW firewall is active on your Ubuntu system, you must allow traffic on port 5601 to access the Kibana dashboard. For this, we recommend the following command:

sudo ufw allow 5601/tcp

```
root@ubuntu:/home/ubuntu# sudo ufw allow 5601/tcp
Rules updated
Rules updated (v6)
```

Next, open a web browser and navigate to **http://<server-ip>:5601** to access the Kibana interface.



5.4 Installing And Configure Logstash

Install

Logstash is a server-side, open-source data processing pipeline that collects data from several sources at once. Elasticsearch keeps this data after Kibana processes it.

To install Logstash, execute the following command:
sudo apt-get install logstash

Basic Pipeline Configuration

Create a config file **/etc/logstash/conf.d/01-logstash.conf:**

```
input {
  beats {
    port => 5044
  }
}
filter {
  grok {
    match => { "message" => "%{COMBINEDAPACHELOG}" }
```

```
  }
}
output {
  elasticsearch {
    hosts => ["localhost:9200"]
    index => "logs-%{+YYYY.MM.dd}"
  }
}
```

Start and Enable the Logstash with the following commands:

sudo systemctl enable logstash

```
root@ubuntu:/home/ubuntu# sudo systemctl enable logstash
Created symlink /etc/systemd/system/multi-user.target.wants/logstash.service → /
etc/systemd/system/logstash.service.
```

sudo systemctl start logstash

To check if it is installed and operating successfully, run the following command as no output would be generated:

sudo systemctl status logstash

```
root@ubuntu:/home/ubuntu# sudo systemctl status logstash
● logstash.service - logstash
     Loaded: loaded (/etc/systemd/system/logstash.service; enabled; preset: ena>
     Active: active (running) since Sat 2025-02-01 11:23:18 UTC; 2min 11s ago
   Main PID: 12652 (java)
      Tasks: 32 (limit: 18751)
     Memory: 754.2M (peak: 758.5M)
        CPU: 7min 40.579s
     CGroup: /system.slice/logstash.service
             └─12652 /usr/share/logstash/jdk/bin/java -Xms1g -Xmx1g -XX:+UseCon>

Feb 01 11:23:18 ubuntu systemd[1]: Started logstash.service - logstash.
Feb 01 11:23:18 ubuntu logstash[12652]: Using bundled JDK: /usr/share/logstash/>
Feb 01 11:23:20 ubuntu logstash[12652]: OpenJDK 64-Bit Server VM warning: Optio>
lines 1-13/13 (END)
```

5.5 Installing Beats (Optional)

A simple tool called Filebeat is used to collect log files and transmit them to Elasticsearch or Logstash. The most often used Beats module is this one. One of Filebeat's main features is its ability to slow down processing in the event that the Logstash receives a lot of data.

Install:

sudo apt-get install filebeat

Note: While installing and configuring Kibana, make sure the service is up and running.

6. Demo

6.1 Experiment 1: Collecting System Logs

6.1.1 Objective

Collect system logs (e.g., **/var/log/syslog** on Ubuntu or **/var/log/messages** on CentOS).

Parse basic fields (timestamps, log levels, etc.) using Logstash filters.

Visualize the logs in Kibana.

6.1.2 Steps

Use our favourite text editor to alter the filebeat.yml configuration file in order to accomplish this:

sudo nano /etc/filebeat/filebeat.yml

Configure /etc/filebeat/filebeat.yml:

```
filebeat.inputs:
  - type: log
    enabled: true
    paths:
      - /var/log/*.log
output.logstash:
  hosts: ["localhost:5044"]
```

Activate the Filebeat system module to analyze local system logs:

sudo filebeat modules enable system

```
root@ubuntu:/home/ubuntu# sudo filebeat modules enable system
Module system is already enabled
```

Once you have activated, load the index template:

sudo filebeat setup --index-management -E output.logstash.enabled=false -E 'output.elasticsearch.hosts=["localhost:9200"]'

Tasks like scanning our system and connecting to our Kibana dashboard will be carried out by the system.
Next, start and enable Filebeat by executing the following command:

sudo systemctl start filebeat

sudo systemctl enable filebeat

Since there would be no output, use the curl command to make sure Filebeat is operating effectively and send log files to Logstash for processing before sending them to Elasticsearch.

curl -XGET http://localhost:9200/_cat/indices?v

```
root@ubuntu:/home/ubuntu# curl -XGET http://localhost:9200/_cat/indices?v
health status index                            uuid                   pri rep docs.count docs.deleted store.size pri.s
tore.size
green  open   .geoip_databases                  vrtYauC0TnWX3Sj4xtFggg   1   0         37            0     35.5mb
   35.5mb
yellow open   filebeat-7.17.27-2025.01.31       i8Fu41VOQreh-O01KK4IzQ   1   1       1022            0    933.6kb
  933.6kb
green  open   .apm-custom-link                  jISaY2_ZQUa5gO1KSP3MpQ   1   0          0            0       227b
     227b
green  open   .apm-agent-configuration          fBLX7XT8SBSOWnR0d8o0JQ   1   0          0            0       227b
     227b
green  open   .kibana_7.17.27_001               x88lMQa0TNSVlsdDarRAtw   1   0       2510          164      3.9mb
    3.9mb
yellow open   filebeat-7.17.27-2025.01.28       HJsU-pKARAG2L3pFevQOOQ   1   1      20501            0      3.5mb
    3.5mb
yellow open   filebeat-7.17.27-2025.01.28-000001 Gi9rkUDbTIC0dsm5eko63A   1   1          0            0       227b
     227b
green  open   .async-search                     sqw4JKxSSpC_O4PotjO7ng   1   0          2          265     82.2kb
   82.2kb
yellow open   filebeat-7.17.27-2025.01.29       Wa9oSehkTqCpECDBuUX7fw   1   1        993            0    365.6kb
  365.6kb
green  open   .kibana_task_manager_7.17.27_001  TVbT4R7PR8aL92hLIHaGgQ   1   0         17         6612    877.7kb
  877.7kb
```

6.1.3 Analysis

Data Ingestion and Overview

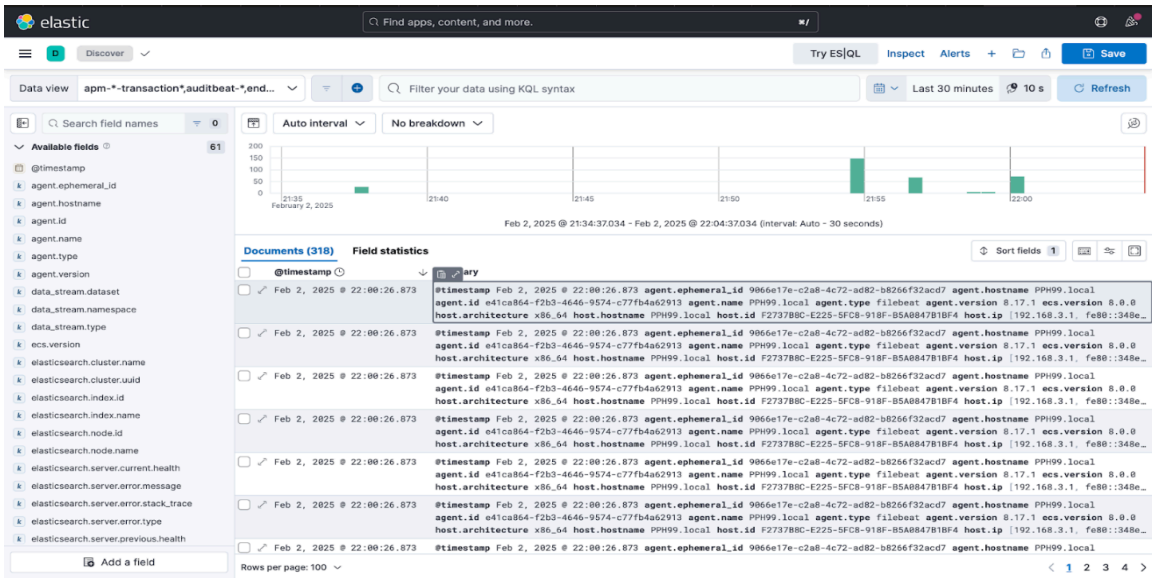
The logs confirm that Filebeat is **successfully parsing** system events and shipping them to Elasticsearch.

Documents display fields such as agent.hostname (PPH99), agent.type (filebeat), host.architecture (x86_64), and standard ECS fields (@timestamp, message, process.name, etc.).

Data Source: System logs (syslogd, login, etc.) collected from host **PPH99**.

Volume: Over 200k documents ingested, displayed in Discover. A subset (125 documents) matches the pre-built "Syslog logs [Filebeat System] ECS" dashboard.

Time Range: Logs span a window from about **Feb 1, 2025 @ 21:35** to **Feb 2, 2025 @ 22:25**.



Visualizations in Kibana

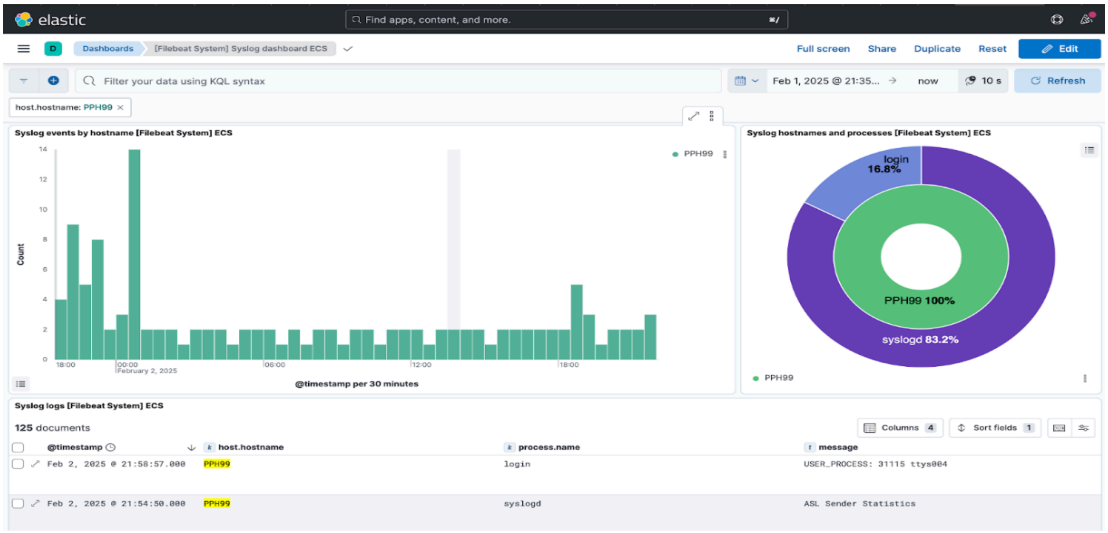
Two main charts are visible in the provided dashboard screenshot:

Bar Chart (left):

> **X-axis**: Time, aggregated in 30-minute intervals.
>
> **Y-axis**: Count of system log events.
>
> Shows varying event rates with a notable spike around midnight. These spikes often occur when multiple system processes rotate logs, user sessions change, or daily cron jobs run.

Donut Chart (right):

> Segments by **hostname** (outer ring) and **process** (inner ring).
>
> Indicates **syslogd** and **login** as the most frequent log-generating processes on host PPH99.
>
> The donut reveals that 83.2% of these particular syslog events relate to **syslogd**, while the rest are from **login**.



6.2 Experiment 2: Collecting Elasticsearch Logs

6.2.1 Objective

Collect logs from a running Elasticsearch instance.
Parse logs (if needed) and ship them to Elasticsearch or Logstash.
Analyze and visualize these logs within Kibana to gain insights into Elasticsearch's internal operations, errors, and performance.

6.2.2 Steps

Enable Elasticsearch Module

This command activates the built-in module for Elasticsearch logs.

sudo filebeat modules enable elasticsearch

Edit the Module Configuration

Open the Elasticsearch module configuration file in Filebeat. Typically located at:

sudo nano filebeat/modules.d/elasticsearch.yml

Update var.paths to match your real Elasticsearch log directory.

```
- module: elasticsearch
server:
  enabled: true
  var.paths:
    - /path/to/elasticsearch/logs/*.json
```

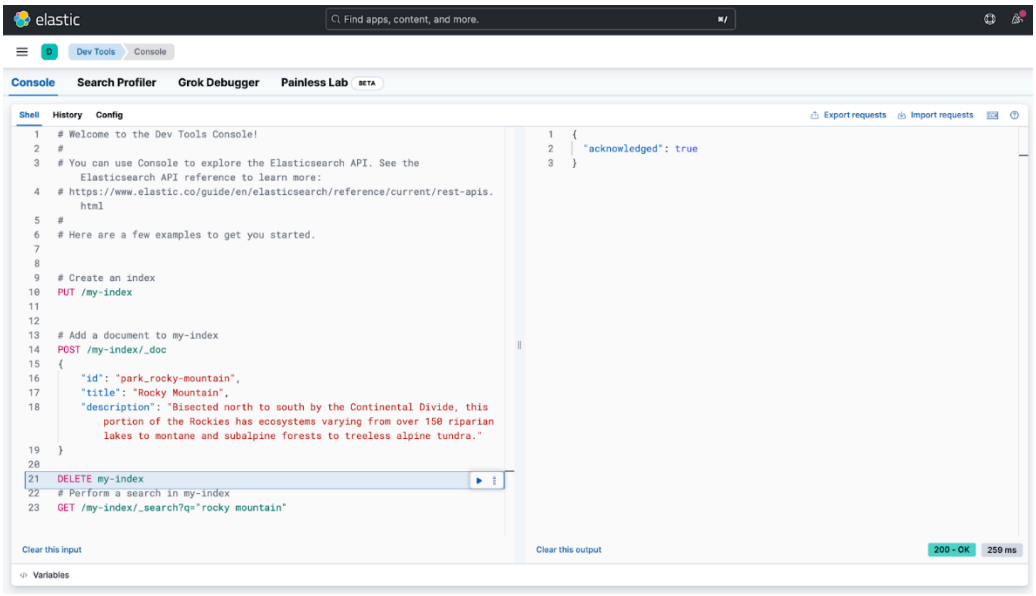
- /path/to/elasticsearch/logs/*_server.json

Restart Filebeat

./filebeat -e

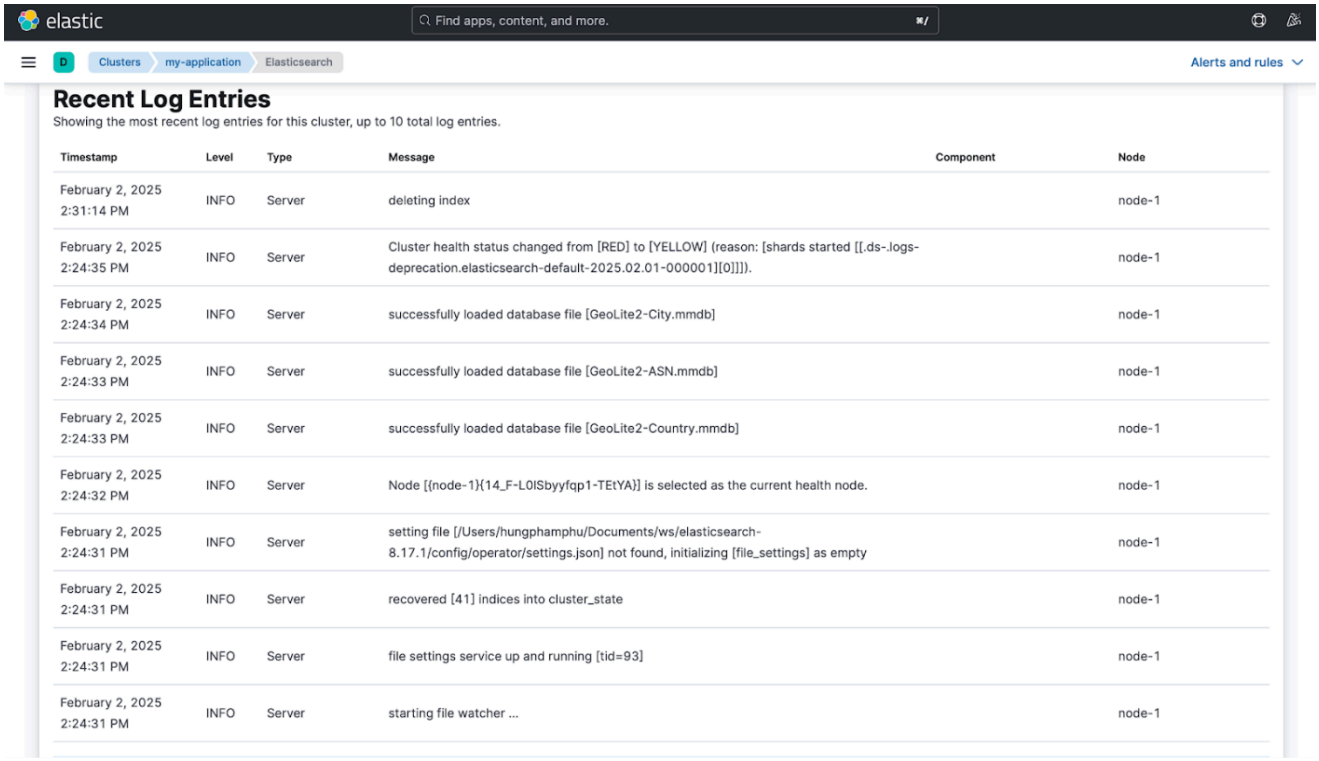
Open Dev Tools in Kibana

Click "Play" (the green arrow) to run each command.



6.2.3 Analysis

Shows the **internal logs** that record each operation initiated in Dev Tools—verifying that our create, update, and delete actions took place and how they affected the cluster.



The “Recent Log Entries” confirm what happened:

Deleting index at 2:31:14 PM directly corresponds to the DELETE /my-index command in Dev Tools.

Cluster health changing from RED to YELLOW reflects shard allocation after creating or removing an index.

Other entries (e.g., loading GeoLite2 databases) typically appear during startup or routine background processes.

7.Conclusion

Through this report, we explored the ELK Stack's critical role in modern log management and analysis. We covered its architecture, core components (Elasticsearch, Logstash, Kibana, and optional Beats), and demonstrated how it enhances system reliability, security, and operational insights. The ELK Stack's real-time processing, centralized log management, and scalability make it a powerful solution for organizations seeking improved monitoring and troubleshooting capabilities.

The experiments showcased how to configure and use the stack effectively for collecting and visualizing both system and Elasticsearch logs, providing valuable insights into system performance and security events.

Despite its many advantages, such as open-source flexibility and extensive documentation, we noted the challenges associated with its resource-intensive nature, complex configuration, and potential upgrade issues. Proper tuning and maintenance are essential for maximizing performance and scalability.

Overall, the ELK Stack remains a comprehensive and adaptable solution for log management, enabling businesses to maintain high levels of system uptime, security, and compliance.

8. References

1. Logz.io, "Complete Guide to the ELK Stack." [Online]. Available: https://logz.io/learn/complete-guide-elk-stack/
2. Elastic, "Elastic Stack Overview." [Online]. Available: https://www.elastic.co/elastic-stack
3. AWS, "What is the ELK Stack?" [Online]. Available: https://aws.amazon.com/vi/what-is/elk-stack/