REDTEAM®
HACKER ACADEMY

END-TO-END ELK STACK PROJECT

LOG COLLECTION, CONFIGURATION, ANALYSIS & VISUALIZATION

Prepared by :

Amal P P

+91 8281506341 amalpp42@gmail.com

Contents

What is ELK?

ELK is a powerful open-source log management and analysis stack used for searching, analyzing, and visualizing log data in real time.
The name ELK comes from the initials of its three core components: Elasticsearch, Logstash, and Kibana. These tools work together to provide a powerful and flexible system for centralized log management and operational intelligence.

Components of ELK Stack

Elasticsearch acts as the central data store and search engine. It allows for fast storage, indexing, and querying of structured and unstructured data. When log data is sent to Elasticsearch, it is stored as JSON documents that can be easily searched and analyzed using RESTful APIs or queries.

Logstash sits at the center of the data processing pipeline. It collects data from various sources such as files, syslogs, or message queues. Before passing data to Elasticsearch, Logstash can parse, transform, and enrich the logs through its powerful filtering capabilities. This ensures that the data is clean, well-structured, and enriched with useful information like timestamps, geo-location, or tags, making analysis much easier downstream.

Kibana, serves as the visualization interface of the ELK Stack. It connects directly to Elasticsearch and provides interactive dashboards, charts, and graphs to help users explore their log data. With Kibana, users can monitor real-time trends, identify security incidents, debug application errors, or analyze performance metrics, all through an intuitive web interface. These visualizations make complex data easy to understand and are crucial for decision-making and operational monitoring.

Logstash Configuration

configure Logstash by editing its pipeline configuration files, typically found in */etc/logstash/conf.d/*. A basic config file has three sections: input, filter, and output. In the input block, you can define how Logstash receives data, for example, from Filebeat or local files. The filter section allows you to parse and transform data (e.g., using *grok* patterns to extract fields from logs), and the output section typically sends the processed data to Elasticsearch. You can test Logstash config with logstash *--config.test_and_exit -f yourfile.conf* and then start the service.

Set up the Logstash configuration file (.conf) to control how log data flows through Logstash from input to processing to output. This file acts like a blueprint or workflow that tells Logstash:
Where the data is coming from
How to interpret or transform the data
Where to send the processed data

Input Configuration
The Input section in a Logstash configuration file defines where Logstash will collect data from. This can be files, syslog, beats, Kafka, or other data sources.

```
input {
 file {
 path => "/var/log/syslog"
 start_position => "beginning"
 }
}
```

This tells Logstash which log or data source to read and begin the pipeline process.

Filter configuration
The Filter section is used to process, parse, and transform the input data before it is sent to the output. You can extract fields, clean data, convert formats, and apply conditional logic.

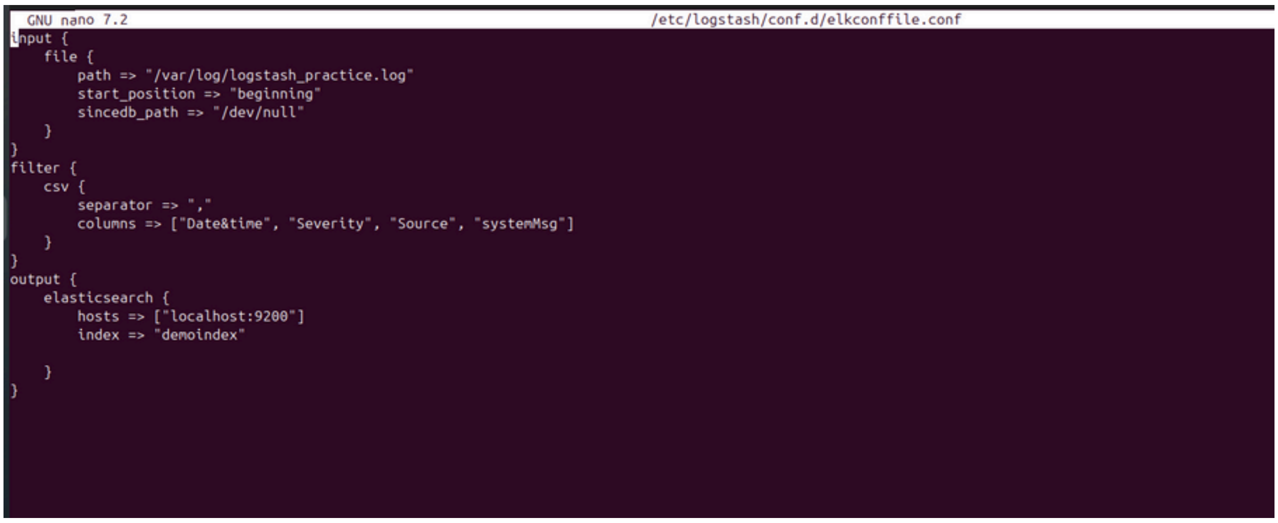
```
filter {
 grok {
 match => { "message" => "%{COMMONAPACHELOG}" }
 }
}
```

Applies data enrichment or parsing logic to structure the logs for easier analysis.

Output Configuration
The Output section defines where to send the processed data, such as to Elasticsearch, files, stdout (console), or other systems.

```
output {
  elasticsearch {
    hosts => ["http://localhost:9200"]
    index => "system-logs"
  }
}
```

Directs the processed logs to a destination (like Elasticsearch for storage and analysis in Kibana).

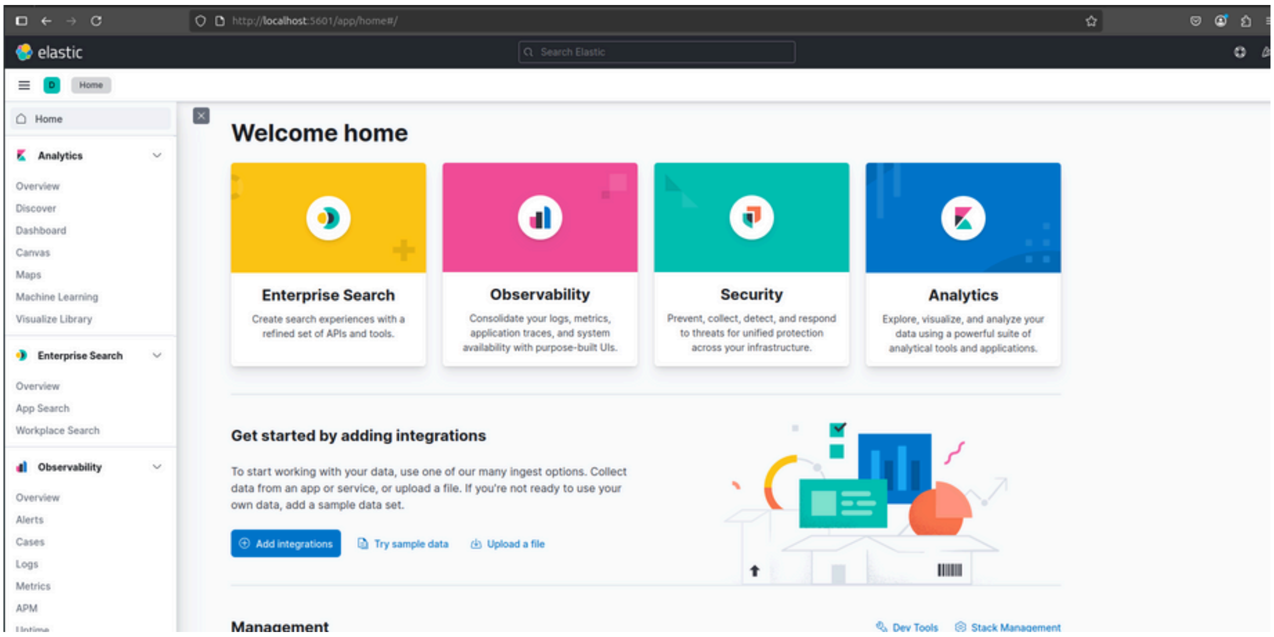


```
  GNU nano 7.2                                              /etc/logstash/conf.d/elkconffile.conf
input {
    file {
        path => "/var/log/logstash_practice.log"
        start_position => "beginning"
        sincedb_path => "/dev/null"
    }
}
filter {
    csv {
        separator => ","
        columns => ["Date&time", "Severity", "Source", "systemMsg"]
    }
}
output {
    elasticsearch {
        hosts => ["localhost:9200"]
        index => "demoindex"

    }
}
```

sudo /usr/share/logstash/bin/logstash -f /etc/logstash/conf.d/elkconffile .conf

run this command to check if there is any error in the conf file.

kibana



http://localhost:5601/app/home#/

Welcome home

Enterprise Search Create search experiences with a refined set of APIs and tools.

Observability Consolidate your logs, metrics, application traces, and system availability with purpose-built UIs.

Security Prevent, collect, detect, and respond to threats for unified protection across your infrastructure.

Analytics Explore, visualize, and analyze your data using a powerful suite of analytical tools and applications.

Get started by adding integrations

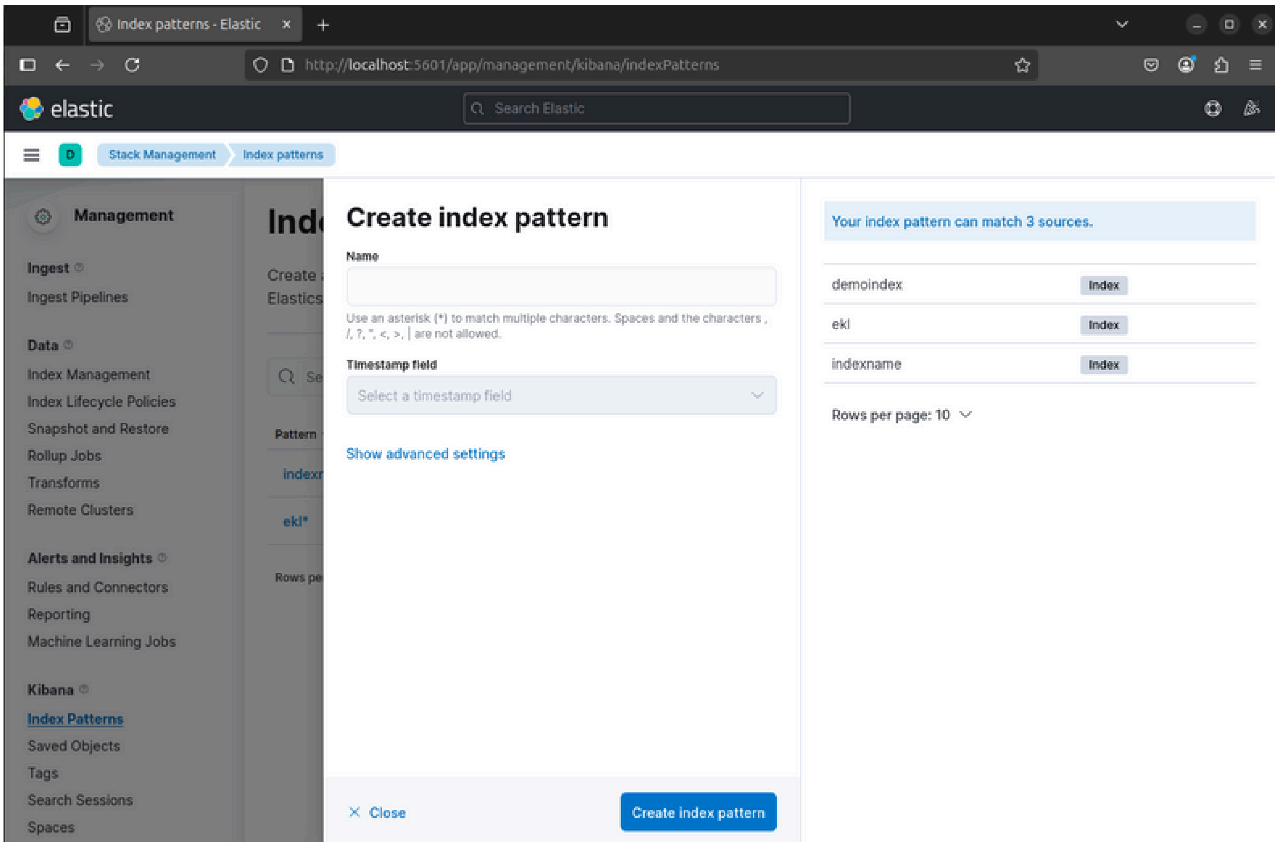
To start working with your data, use one of our many ingest options. Collect data from an app or service, or upload a file. If you're not ready to use your own data, add a sample data set.

index pattern

What is an Index Pattern in Kibana?

An index pattern in Kibana is a configuration that tells Kibana which Elasticsearch indices it should read data from. It acts as a bridge between Kibana and Elasticsearch so that Kibana can query, visualize, and analyze your log data.

How to Create an Index Pattern in Kibana:
Go to Kibana → Stack Management → Index Patterns
Click "Create index pattern"
Enter your index pattern (e.g., logstash-*)
Select the timestamp field (e.g., @timestamp) if available
Click Create index pattern
Once created, you can use this pattern in:
Discover (search and explore logs)
Visualize (build charts)
Dashboards



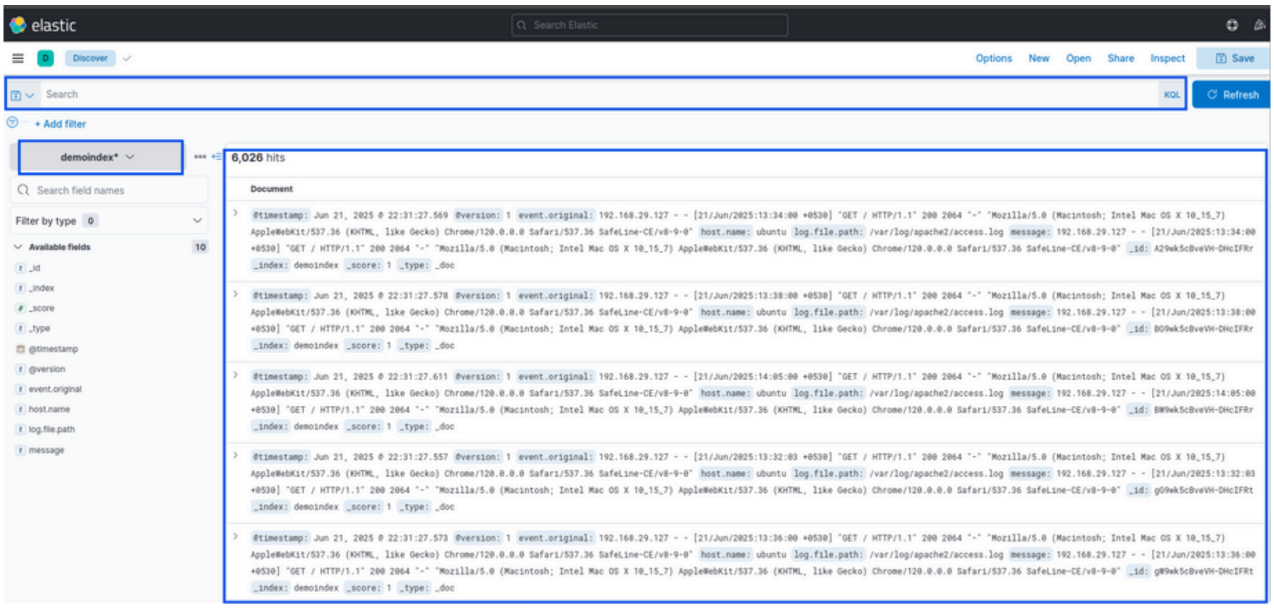
Use an asterisk (*) to match multiple characters. Spaces and the characters , /, ?, ", <, >, | are not allowed.

Your index pattern can match 3 sources.

demoindex	Index
ekl	Index
indexname	Index

What is Discover in Kibana?

Discover is a feature in Kibana that allows you to search, explore, and analyze raw log data stored in Elasticsearch indices. It's the starting point for interacting with your data.



```
6,026 hits
Document
@timestamp: Jun 21, 2025 @ 22:31:27.569 @version: 1 event.original: 192.168.29.127 - - [21/Jun/2025:13:34:00 +0530] "GET / HTTP/1.1" 200 2064 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 SafeLine-CE/v8-9-0" host.name: ubuntu log.file.path: /var/log/apache2/access.log message: 192.168.29.127 - - [21/Jun/2025:13:34:00
+0530] "GET / HTTP/1.1" 200 2064 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 SafeLine-CE/v8-9-0" _id: A29wk5cBveVH-DHcIFRr
_index: demoindex _score: 1 _type: _doc
@timestamp: Jun 21, 2025 @ 22:31:27.578 @version: 1 event.original: 192.168.29.127 - - [21/Jun/2025:13:38:00 +0530] "GET / HTTP/1.1" 200 2064 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 SafeLine-CE/v8-9-0" host.name: ubuntu log.file.path: /var/log/apache2/access.log message: 192.168.29.127 - - [21/Jun/2025:13:38:00
+0530] "GET / HTTP/1.1" 200 2064 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 SafeLine-CE/v8-9-0" _id: BG9wk5cBveVH-DHcIFRr
_index: demoindex _score: 1 _type: _doc
@timestamp: Jun 21, 2025 @ 22:31:27.611 @version: 1 event.original: 192.168.29.127 - - [21/Jun/2025:14:05:00 +0530] "GET / HTTP/1.1" 200 2064 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 SafeLine-CE/v8-9-0" host.name: ubuntu log.file.path: /var/log/apache2/access.log message: 192.168.29.127 - - [21/Jun/2025:14:05:00
+0530] "GET / HTTP/1.1" 200 2064 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 SafeLine-CE/v8-9-0" _id: BW9wk5cBveVH-DHcIFRr
_index: demoindex _score: 1 _type: _doc
@timestamp: Jun 21, 2025 @ 22:31:27.557 @version: 1 event.original: 192.168.29.127 - - [21/Jun/2025:13:32:03 +0530] "GET / HTTP/1.1" 200 2064 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 SafeLine-CE/v8-9-0" host.name: ubuntu log.file.path: /var/log/apache2/access.log message: 192.168.29.127 - - [21/Jun/2025:13:32:03
+0530] "GET / HTTP/1.1" 200 2064 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 SafeLine-CE/v8-9-0" _id: gG9wk5cBveVH-DHcIFRt
_index: demoindex _score: 1 _type: _doc
@timestamp: Jun 21, 2025 @ 22:31:27.573 @version: 1 event.original: 192.168.29.127 - - [21/Jun/2025:13:36:00 +0530] "GET / HTTP/1.1" 200 2064 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 SafeLine-CE/v8-9-0" host.name: ubuntu log.file.path: /var/log/apache2/access.log message: 192.168.29.127 - - [21/Jun/2025:13:36:00
+0530] "GET / HTTP/1.1" 200 2064 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 SafeLine-CE/v8-9-0" _id: gW9wk5cBveVH-DHcIFRt
_index: demoindex _score: 1 _type: _doc
```

Select the index pattern created before and see the logs are showing in.
here logs are from the apache server running in ubuntu operating system

Querying Logs in Kibana

Searching or finding particular logs we can query inside the search bar in kibana,here is where Kibana Querying language(KQL) comes in

Introduction to Kibana Query Language (KQL)

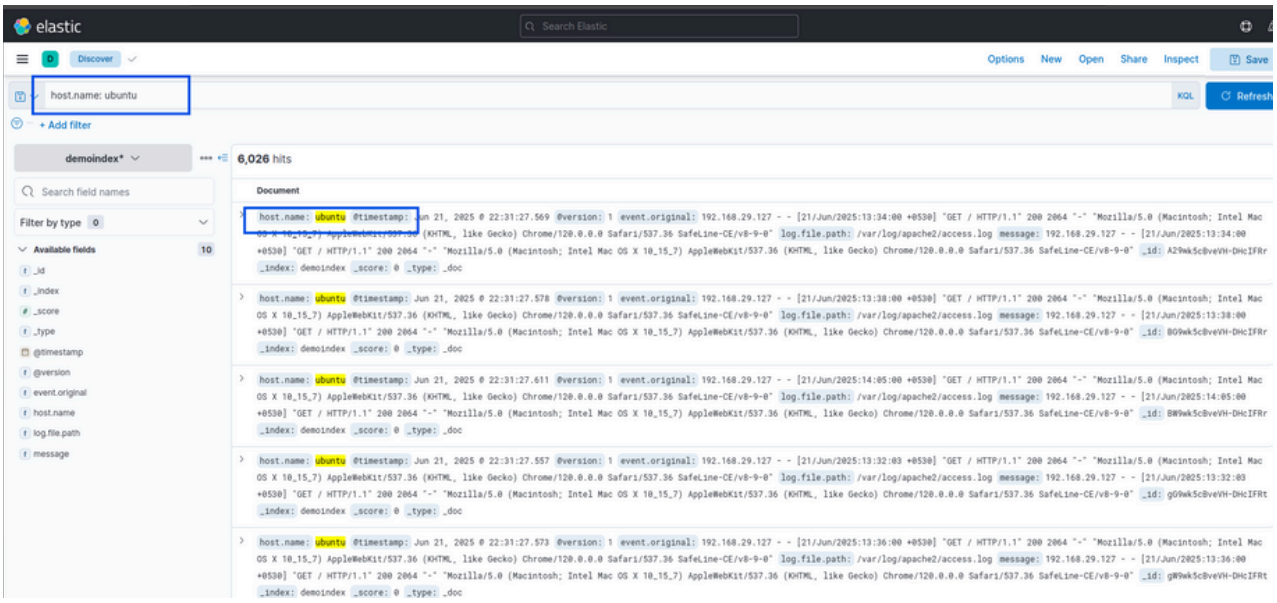
KQL (Kibana Query Language) is a search and filter language used in Kibana to query data in Discover, visualizations, and dashboards. It's a simple, user-friendly syntax designed to help users search and filter logs stored in Elasticsearch indices.

Basic Queries

- Search for a field value → status: 404
- Match multiple values (OR) → status:404 OR status: 500
- Match all conditions (AND) → status:404 AND method: GET
- Negate a query → NOT status:200
- Phrase search → url: "/login"
- Wildcard search → user.name: "admin*"
- Range → search bytes > 1000
- Grouping → (status: 500 AND method: POST)
- Existence check → _exists_: user.name

Filtering and Searching

Filter by host : Ubuntu



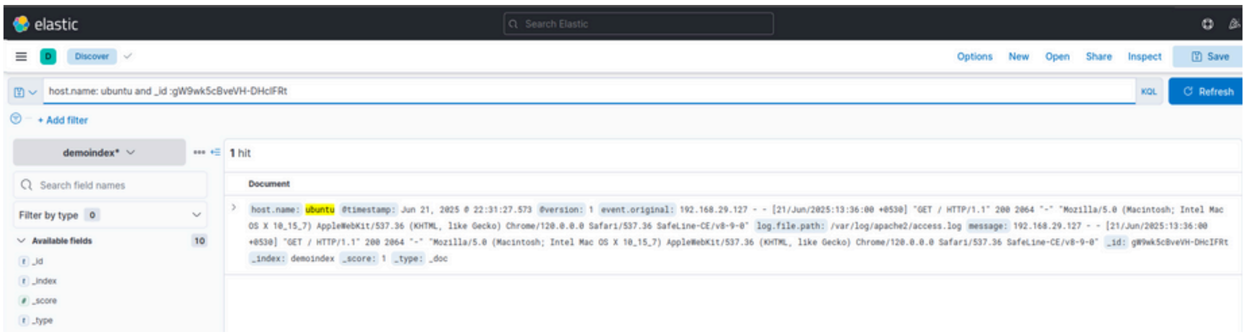
```
host.name: ubuntu
```

6,026 hits

Document

```
host.name: ubuntu @timestamp: Jun 21, 2025 @ 22:31:27.569 @version: 1 event.original: 192.168.29.127 - - [21/Jun/2025:13:34:00 +0530] "GET / HTTP/1.1" 200 2064 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 SafeLine-CE/v8-9-0" log.file.path: /var/log/apache2/access.log message: 192.168.29.127 - - [21/Jun/2025:13:34:00 +0530] "GET / HTTP/1.1" 200 2064 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 SafeLine-CE/v8-9-0" _id: A29wk5cBveVH-DHcIFRr _index: demoindex _score: 0 _type: _doc

host.name: ubuntu @timestamp: Jun 21, 2025 @ 22:31:27.578 @version: 1 event.original: 192.168.29.127 - - [21/Jun/2025:13:38:00 +0530] "GET / HTTP/1.1" 200 2064 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 SafeLine-CE/v8-9-0" log.file.path: /var/log/apache2/access.log message: 192.168.29.127 - - [21/Jun/2025:13:38:00 +0530] "GET / HTTP/1.1" 200 2064 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 SafeLine-CE/v8-9-0" _id: BG9wk5cBveVH-DHcIFRr _index: demoindex _score: 0 _type: _doc

host.name: ubuntu @timestamp: Jun 21, 2025 @ 22:31:27.611 @version: 1 event.original: 192.168.29.127 - - [21/Jun/2025:14:05:00 +0530] "GET / HTTP/1.1" 200 2064 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 SafeLine-CE/v8-9-0" log.file.path: /var/log/apache2/access.log message: 192.168.29.127 - - [21/Jun/2025:14:05:00 +0530] "GET / HTTP/1.1" 200 2064 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 SafeLine-CE/v8-9-0" _id: BW9wk5cBveVH-DHcIFRr _index: demoindex _score: 0 _type: _doc

host.name: ubuntu @timestamp: Jun 21, 2025 @ 22:31:27.557 @version: 1 event.original: 192.168.29.127 - - [21/Jun/2025:13:32:03 +0530] "GET / HTTP/1.1" 200 2064 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 SafeLine-CE/v8-9-0" log.file.path: /var/log/apache2/access.log message: 192.168.29.127 - - [21/Jun/2025:13:32:03 +0530] "GET / HTTP/1.1" 200 2064 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 SafeLine-CE/v8-9-0" _id: gG9wk5cBveVH-DHcIFRt _index: demoindex _score: 0 _type: _doc

host.name: ubuntu @timestamp: Jun 21, 2025 @ 22:31:27.573 @version: 1 event.original: 192.168.29.127 - - [21/Jun/2025:13:36:00 +0530] "GET / HTTP/1.1" 200 2064 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 SafeLine-CE/v8-9-0" log.file.path: /var/log/apache2/access.log message: 192.168.29.127 - - [21/Jun/2025:13:36:00 +0530] "GET / HTTP/1.1" 200 2064 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 SafeLine-CE/v8-9-0" _id: gW9wk5cBveVH-DHcIFRt _index: demoindex _score: 0 _type: _doc
```

Host and id:



```
host.name: ubuntu and _id :gW9wk5cBveVH-DHcIFRt
```

1 hit

Document

```
host.name: ubuntu @timestamp: Jun 21, 2025 @ 22:31:27.573 @version: 1 event.original: 192.168.29.127 - - [21/Jun/2025:13:36:00 +0530] "GET / HTTP/1.1" 200 2064 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 SafeLine-CE/v8-9-0" log.file.path: /var/log/apache2/access.log message: 192.168.29.127 - - [21/Jun/2025:13:36:00 +0530] "GET / HTTP/1.1" 200 2064 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 SafeLine-CE/v8-9-0" _id: gW9wk5cBveVH-DHcIFRt _index: demoindex _score: 1 _type: _doc
```

What is grok

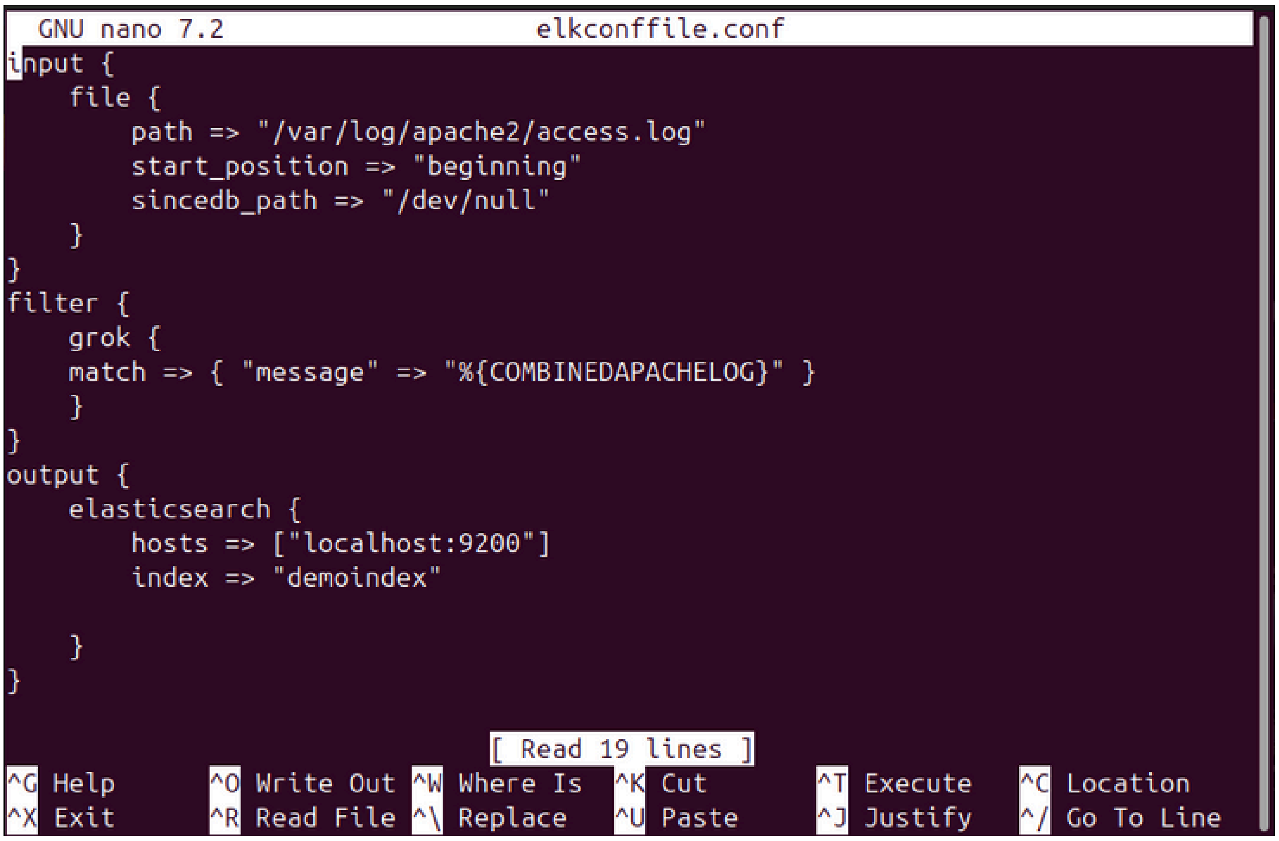
The Grok filter in Logstash is used to extract structured fields from unstructured log data (like your Apache access logs).
It uses regex-like patterns to identify and name parts of the log, so fields like client_ip, method, url, and status become accessible in Elasticsearch.

Use grok in logstash

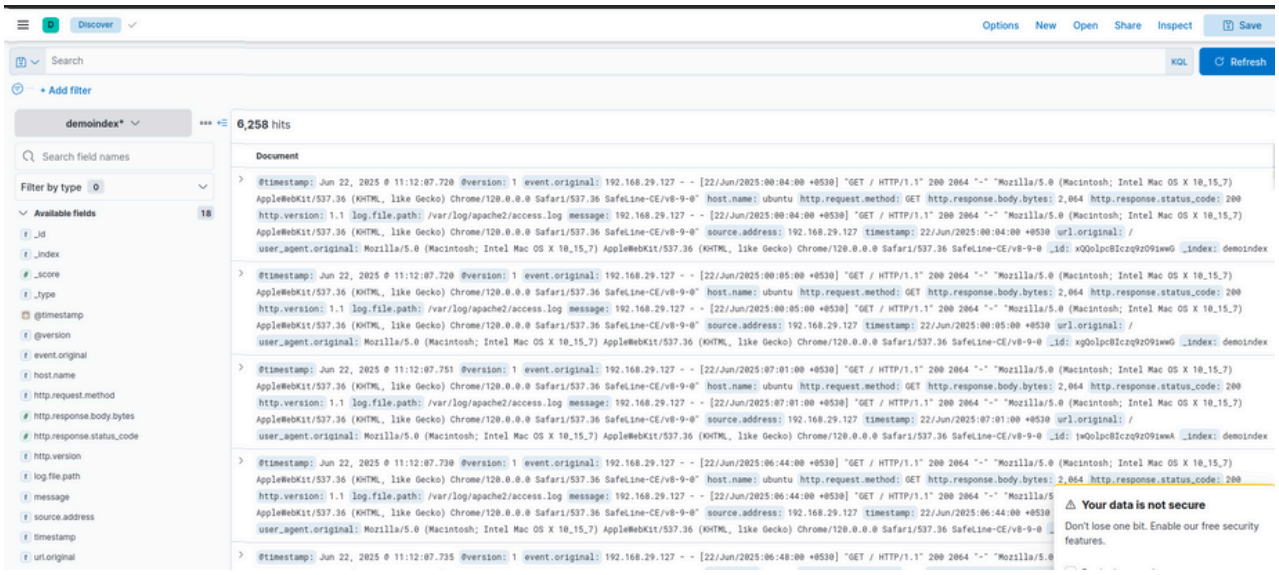
Use Predefined Grok Pattern: COMBINEDAPACHELOG
This matches the Apache combined log format and parses fields like:

- *clientip*
- *ident*
- *auth*
- *timestamp*
- *verb* (HTTP method)
- *request* (URL)
- *httpversion*
- *response* (status code)
- *bytes* (response size)

- *referrer*
- *agent* (User-Agent)



```
  GNU nano 7.2                     elkconffile.conf
input {
    file {
        path => "/var/log/apache2/access.log"
        start_position => "beginning"
        sincedb_path => "/dev/null"
    }
}
filter {
    grok {
    match => { "message" => "%{COMBINEDAPACHELOG}" }
    }
}
output {
    elasticsearch {
        hosts => ["localhost:9200"]
        index => "demoindex"

    }
}
                               [ Read 19 lines ]
^G Help        ^O Write Out  ^W Where Is  ^K Cut        ^T Execute    ^C Location
^X Exit        ^R Read File  ^\ Replace   ^U Paste      ^J Justify    ^/ Go To Line
```



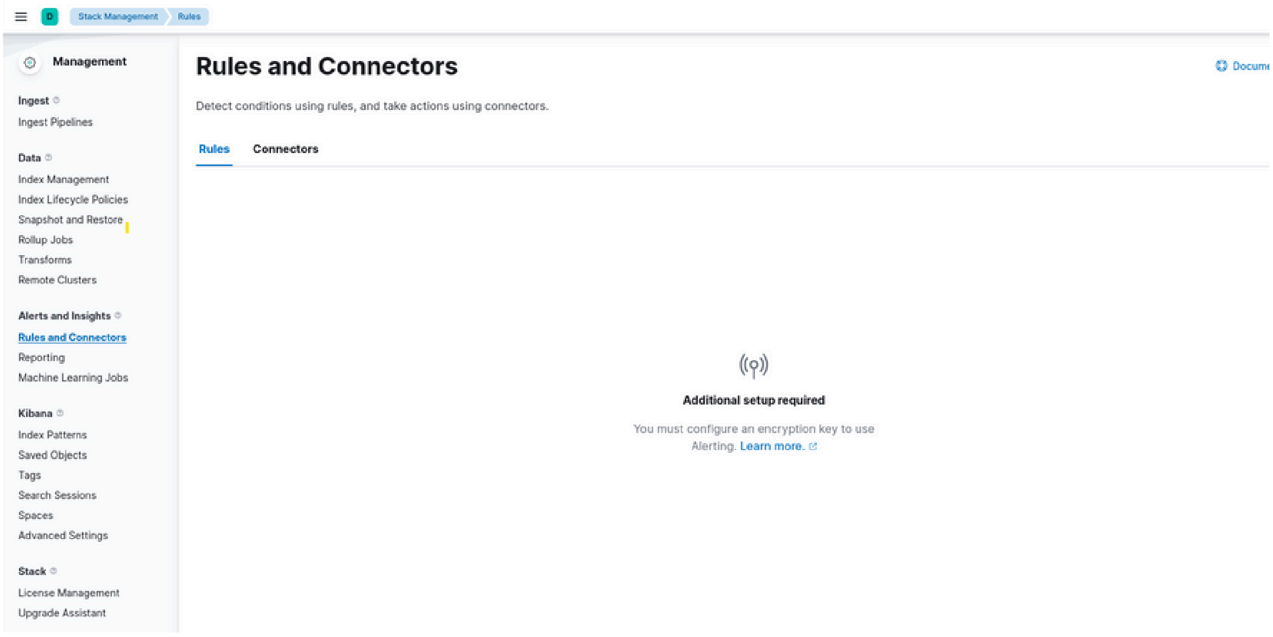
6,258 hits

```
Document
@timestamp: Jun 22, 2025 @ 11:12:07.720 @version: 1 event.original: 192.168.29.127 - - [22/Jun/2025:00:04:00 +0530] "GET / HTTP/1.1" 200 2064 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 SafeLine-CE/v8-9-0" host.name: ubuntu http.request.method: GET http.response.body.bytes: 2,064 http.response.status_code: 200
http.version: 1.1 log.file.path: /var/log/apache2/access.log message: 192.168.29.127 - - [22/Jun/2025:00:04:00 +0530] "GET / HTTP/1.1" 200 2064 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 SafeLine-CE/v8-9-0" source.address: 192.168.29.127 timestamp: 22/Jun/2025:00:04:00 +0530 url.original: /
user_agent.original: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 SafeLine-CE/v8-9-0 _id: xQQolpcBIczq9zO91wwG _index: demoindex

@timestamp: Jun 22, 2025 @ 11:12:07.720 @version: 1 event.original: 192.168.29.127 - - [22/Jun/2025:00:05:00 +0530] "GET / HTTP/1.1" 200 2064 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 SafeLine-CE/v8-9-0" host.name: ubuntu http.request.method: GET http.response.body.bytes: 2,064 http.response.status_code: 200
http.version: 1.1 log.file.path: /var/log/apache2/access.log message: 192.168.29.127 - - [22/Jun/2025:00:05:00 +0530] "GET / HTTP/1.1" 200 2064 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 SafeLine-CE/v8-9-0" source.address: 192.168.29.127 timestamp: 22/Jun/2025:00:05:00 +0530 url.original: /
user_agent.original: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 SafeLine-CE/v8-9-0 _id: xgQolpcBIczq9zO91wwG _index: demoindex

@timestamp: Jun 22, 2025 @ 11:12:07.751 @version: 1 event.original: 192.168.29.127 - - [22/Jun/2025:07:01:00 +0530] "GET / HTTP/1.1" 200 2064 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 SafeLine-CE/v8-9-0" host.name: ubuntu http.request.method: GET http.response.body.bytes: 2,064 http.response.status_code: 200
http.version: 1.1 log.file.path: /var/log/apache2/access.log message: 192.168.29.127 - - [22/Jun/2025:07:01:00 +0530] "GET / HTTP/1.1" 200 2064 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 SafeLine-CE/v8-9-0" source.address: 192.168.29.127 timestamp: 22/Jun/2025:07:01:00 +0530 url.original: /
user_agent.original: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 SafeLine-CE/v8-9-0 _id: jwQolpcBIczq9zO91wwA _index: demoindex

@timestamp: Jun 22, 2025 @ 11:12:07.730 @version: 1 event.original: 192.168.29.127 - - [22/Jun/2025:06:44:00 +0530] "GET / HTTP/1.1" 200 2064 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 SafeLine-CE/v8-9-0" host.name: ubuntu http.request.method: GET http.response.body.bytes: 2,064 http.response.status_code: 200
http.version: 1.1 log.file.path: /var/log/apache2/access.log message: 192.168.29.127 - - [22/Jun/2025:06:44:00 +0530] "GET / HTTP/1.1" 200 2064 "-" "Mozilla/5
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 SafeLine-CE/v8-9-0" source.address: 192.168.29.127 timestamp: 22/Jun/2025:06:44:00 +0530
user_agent.original: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 SafeLine-CE/v8-9-0

@timestamp: Jun 22, 2025 @ 11:12:07.735 @version: 1 event.original: 192.168.29.127 - - [22/Jun/2025:06:48:00 +0530] "GET / HTTP/1.1" 200 2064 "-" "Mozilla/5.0
```

How to Create alert?

Creating alerts in Kibana allows you to monitor your data continuously and get notified when certain conditions are met like brute force attempts, too many 404s, or specific keywords like sqlmap or gobuster.

Go to Stack Management → Rules and Connectors


Stack Management / Rules

Management

Ingest
Ingest Pipelines

Data
Index Management
Index Lifecycle Policies
Snapshot and Restore
Rollup Jobs
Transforms
Remote Clusters

Alerts and Insights
Rules and Connectors
Reporting
Machine Learning Jobs

Kibana
Index Patterns
Saved Objects
Tags
Search Sessions
Spaces
Advanced Settings

Stack
License Management
Upgrade Assistant

Rules and Connectors

Docume

Detect conditions using rules, and take actions using connectors.

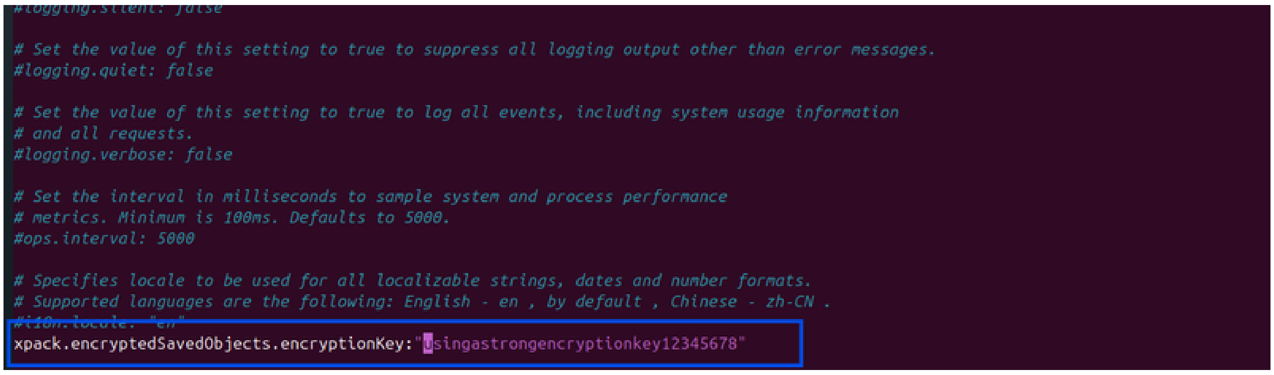
Rules Connectors

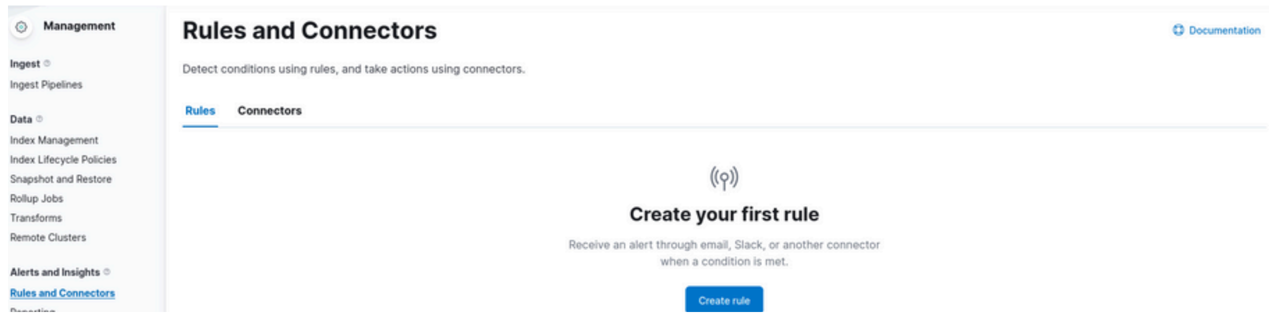
Additional setup required

You must configure an encryption key to use Alerting. Learn more.

To use alerts in Kibana, especially for actions like email, Slack, or indexing, Kibana must be configured with encryption keys. Kibana stores sensitive data in alerting (like API tokens, email configs). For security, it requires a key to encrypt and decrypt those values.
The config file is loacted at : ***/etc/kibana/kibana.yml***
add this configuration at the end of the file

xpack.encryptedSavedObjects.encryptionKey: "aVeryStrongEncryptionKey1234567890!"
 you can use any random string and it must me at least of 32 characters, Remember that the key must be kept private and secure.


```
#logging.silent: false

# Set the value of this setting to true to suppress all logging output other than error messages.
#logging.quiet: false

# Set the value of this setting to true to log all events, including system usage information
# and all requests.
#logging.verbose: false

# Set the interval in milliseconds to sample system and process performance
# metrics. Minimum is 100ms. Defaults to 5000.
#ops.interval: 5000

# Specifies locale to be used for all localizable strings, dates and number formats.
# Supported languages are the following: English - en , by default , Chinese - zh-CN .
#i18n.locale: "en"
xpack.encryptedSavedObjects.encryptionKey:"usingastrongencryptionkey12345678"
```


Management

Ingest
Ingest Pipelines

Data
Index Management
Index Lifecycle Policies
Snapshot and Restore
Rollup Jobs
Transforms
Remote Clusters

Alerts and Insights
Rules and Connectors

Rules and Connectors

Documentation

Detect conditions using rules, and take actions using connectors.

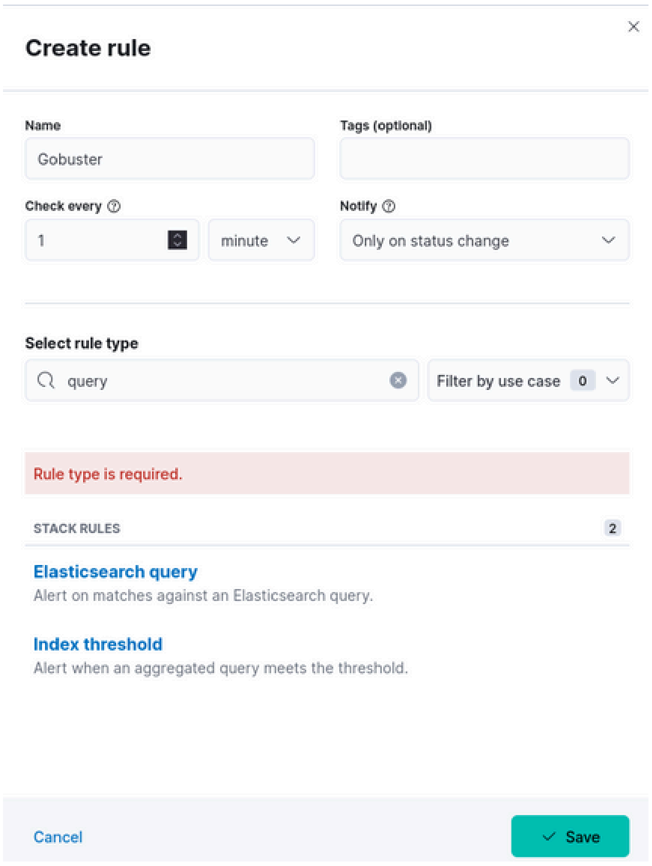
Rules Connectors

Create your first rule

Receive an alert through email, Slack, or another connector when a condition is met.

Create rule

Let’s search for directory enumeration attack using “gobuster”
click on Create new rule
Give a name to the rule and in the search bar search for “query” and select Elasticsearch query


Create rule

Name: Gobuster
Tags (optional):
Check every: 1 minute
Notify: Only on status change

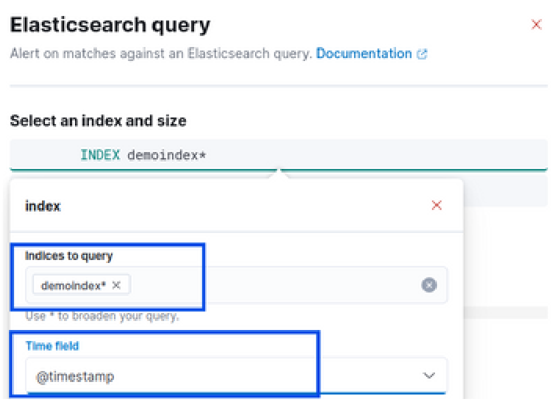
Select rule type: query
Filter by use case 0

Rule type is required.

STACK RULES 2

Elasticsearch query
Alert on matches against an Elasticsearch query.

Index threshold
Alert when an aggregated query meets the threshold.

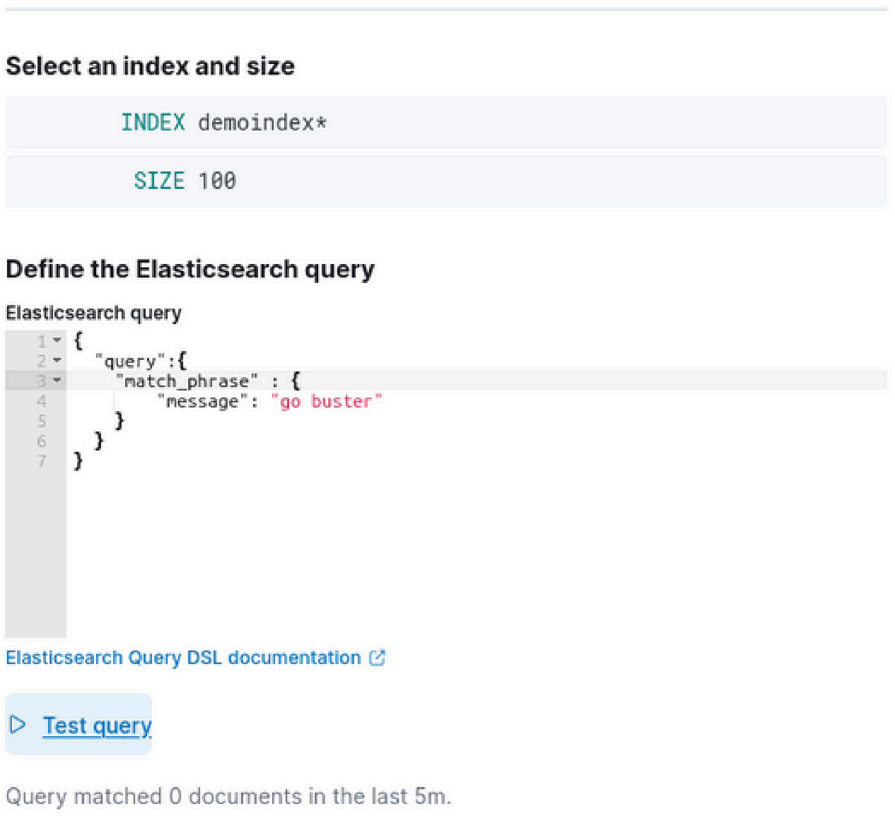

Elasticsearch query
Alert on matches against an Elasticsearch query. Documentation

Select an index and size
INDEX demoindex*

index
Indices to query: demoindex*
Use * to broaden your query.
Time field: @timestamp

select the index and set time field,, this is for realtime checking inside the Elastic search query section add the query.
add the query:

```
{
 "query": {
   "match_phrase": {
     "message": "go buster"
   }
 }
}
```



Select an index and size

```
INDEX demoindex*
SIZE 100
```

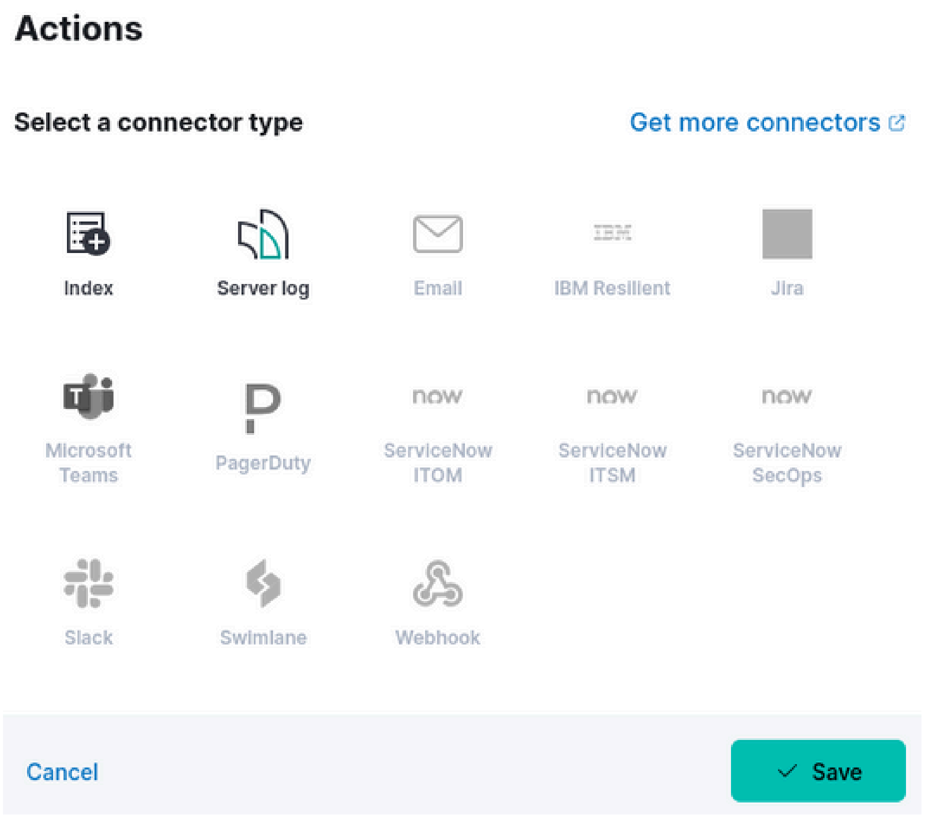
Define the Elasticsearch query

Elasticsearch query

```
{
  "query":{
    "match_phrase" : {
        "message": "go buster"
    }
  }
}
```

Query matched 0 documents in the last 5m.

Actions

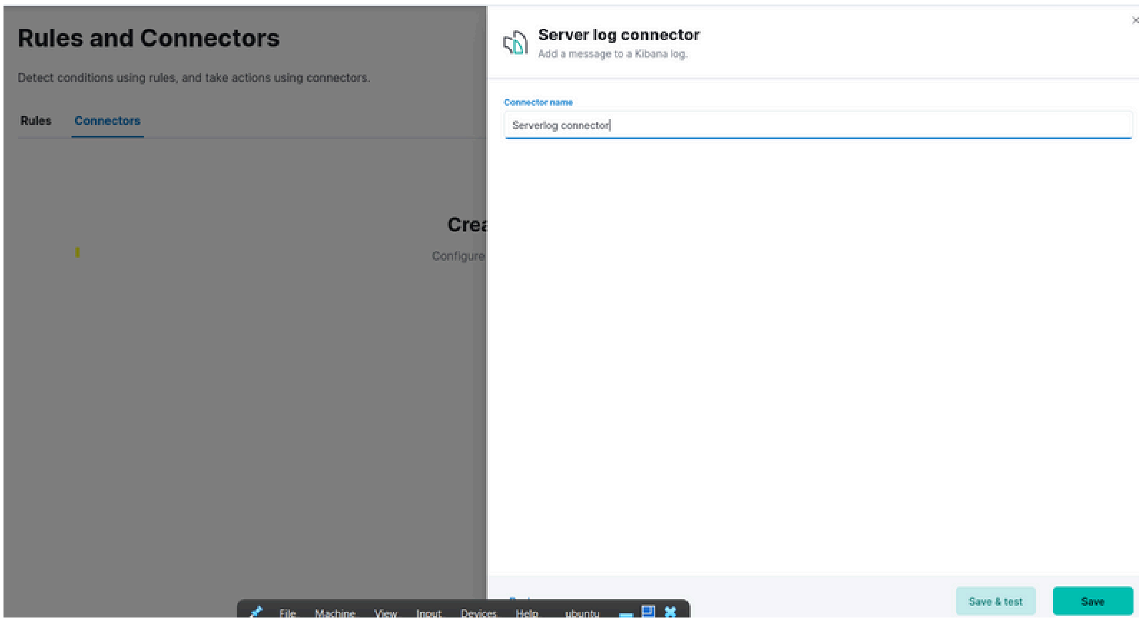


Actions

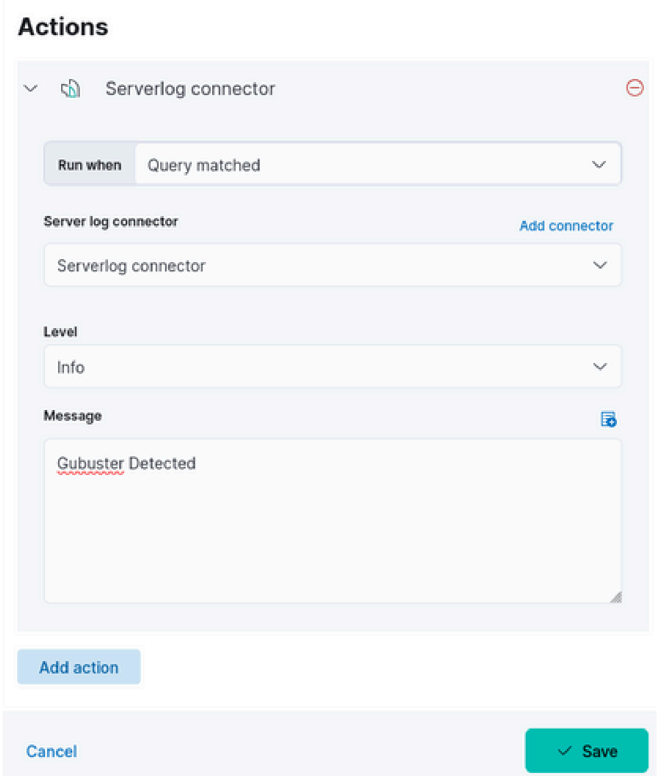
Select a connector type

In Kibana alerts, Actions define what should happen when the alert condition is met. Once your alert rule (condition) is triggered (e.g., based on a query match), the Action sends a notification or takes an automated step.
Connector is necessary when using Actions like Email, Slack, Webhook, Index, Server log, PagerDuty, jira etc. In these cases, you must create a connector first, which provides the credentials, URL, or settings needed to send the alert to that service.
Example:
If you want to send an email when an alert triggers:

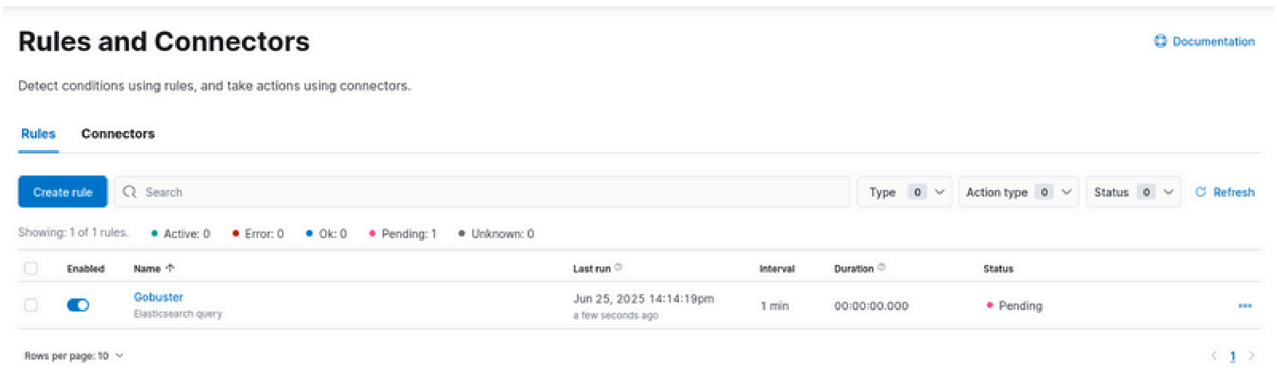
- You must first create an Email connector with SMTP settings.
- Then attach that connector as an action in your alert.


Rules and Connectors
Detect conditions using rules, and take actions using connectors.
Rules
Connectors
Server log connector
Add a message to a Kibana log.
Save & test
Save

Connector name: Serverlog connector



Run when: Query matched
Server log connector: Serverlog connector
Level: Info
Message: Gubuster Detected



Rules and Connectors

Documentation

Detect conditions using rules, and take actions using connectors.

Rules Connectors

Create rule | Search | Type 0 | Action type 0 | Status 0 | Refresh

Showing: 1 of 1 rules. ● Active: 0 ● Error: 0 ● Ok: 0 ● Pending: 1 ● Unknown: 0

	Enabled	Name	Last run	Interval	Duration	Status
	✓	Gobuster Elasticsearch query	Jun 25, 2025 14:14:19pm a few seconds ago	1 min	00:00:00.000	● Pending

Rows per page: 10

Perform an directory enumeration attack on the target to get alert