.

Lesson 05 Demo 02

Building Logstash Pipelines for Centralized Log Collection

Objective: To implement Logstash to store logs in Elasticsearch for efficient management, analysis, and real-time monitoring of application logs

Tools required: Logstash

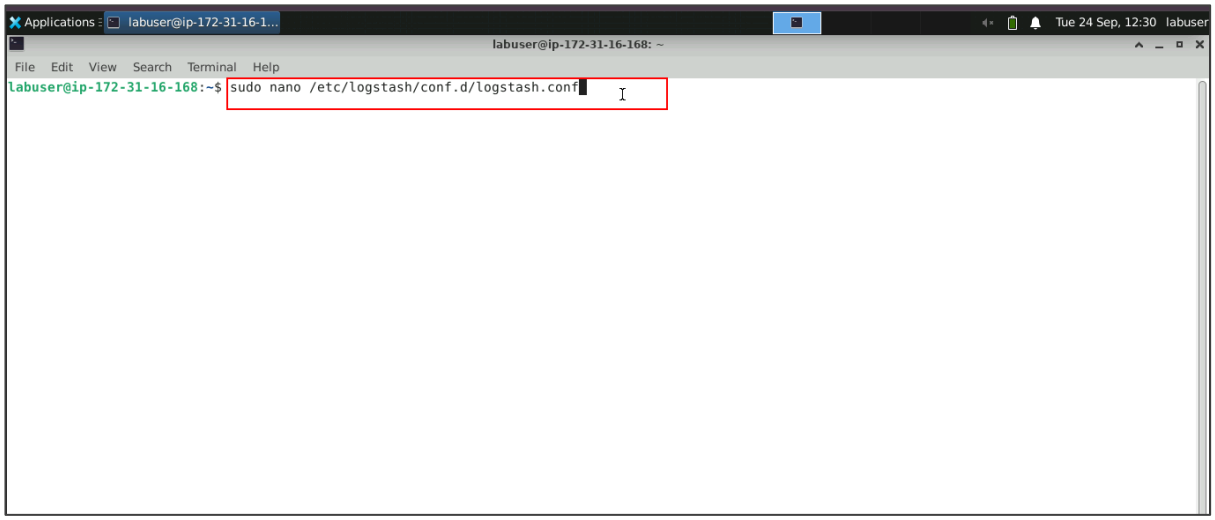
Prerequisites: None

Steps to be followed:

1. Configure the Logstash pipeline

Step 1: Configure the Logstash pipeline

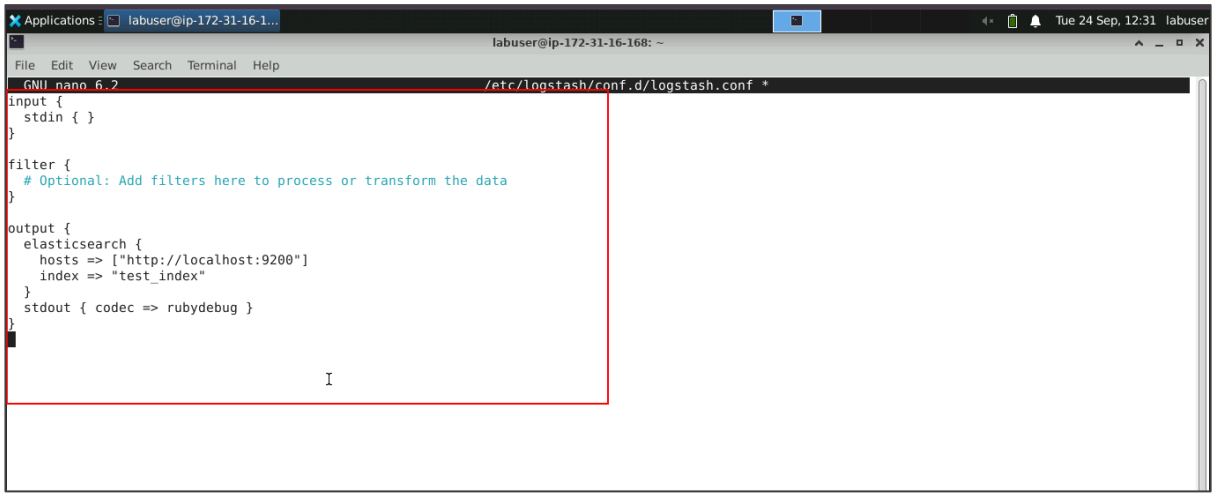
1.1 Open the terminal and run the command below to make changes to the following Logstash configuration file:
sudo nano /etc/logstash/conf.d/logstash.conf



.

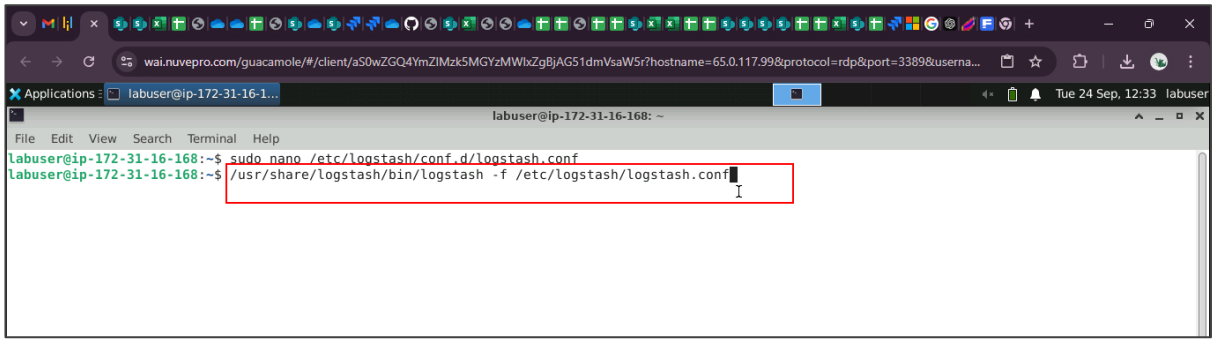
1.2 Enter the following configuration, which contains input, filter, and output blocks in the Logstash configuration file:

```
input {
  stdin { }
}

filter {
  # Optional: Add filters here to process or transform the data
}

output {
  elasticsearch {
    hosts => ["http://localhost:9200"]
    index => "test_index"
  }
  stdout { codec => rubydebug }
}
```



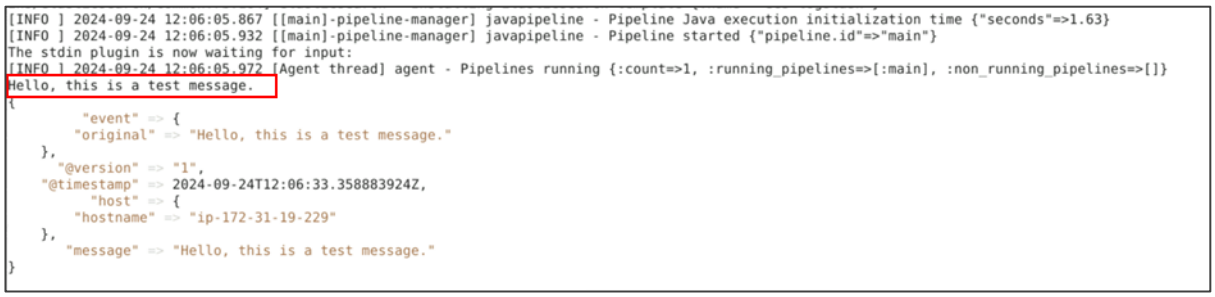
.

1.3 Run the command below to manually test the connectivity between Logstash and Elasticsearch:
/usr/share/logstash/bin/logstash -f /etc/logstash/logstash.conf

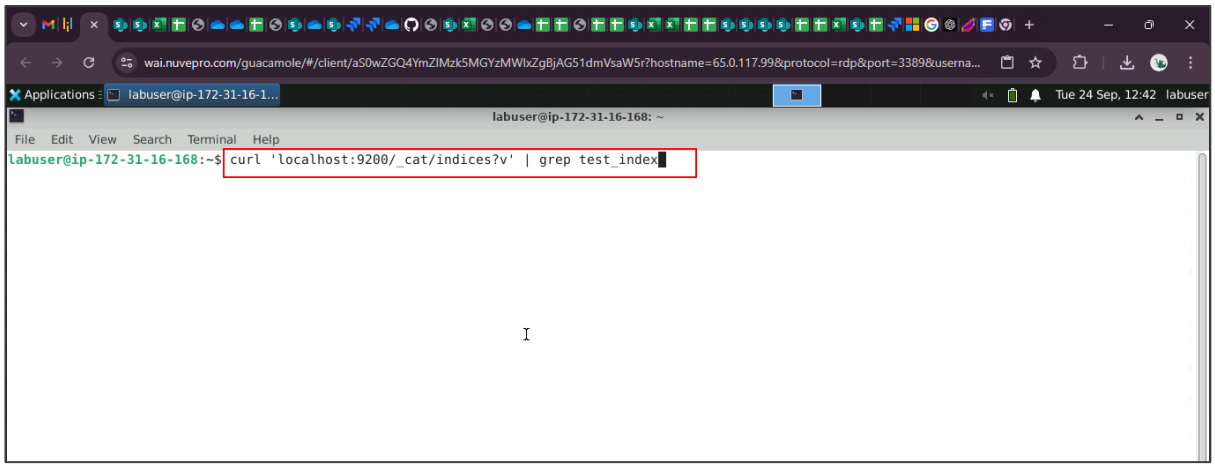


Logstash will start and wait for input.

1.4 Type some dummy data, like the one given below, into the Logstash console and press Enter:
Hello, this is a test message.

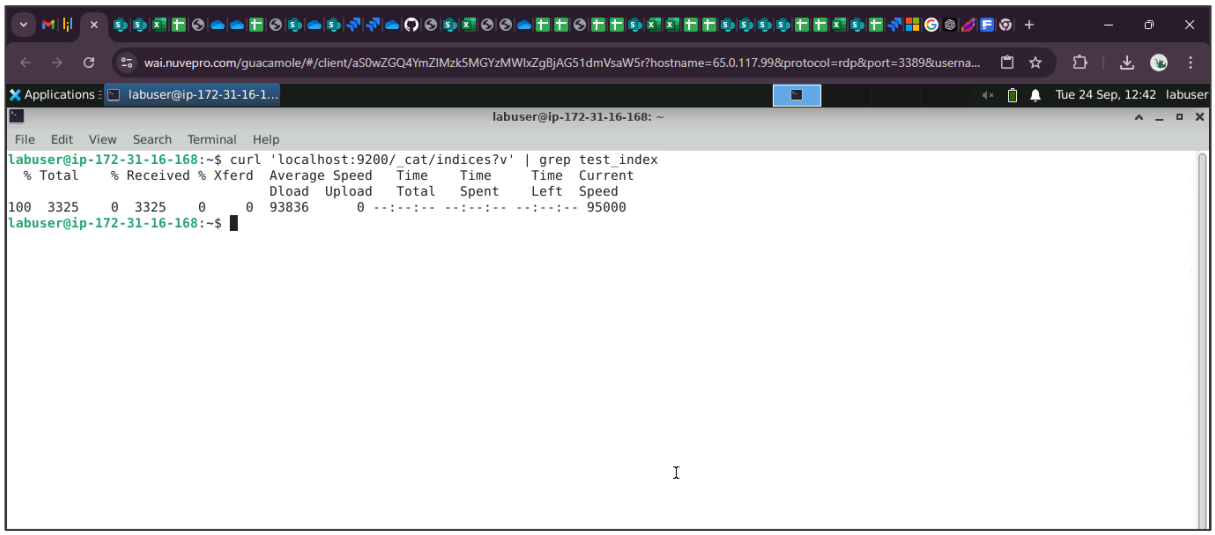


1.5 Run the command below in a new terminal tab to validate whether **test_index** is created after the message is processed:
curl 'localhost:9200/_cat/indices?v' | grep test_index

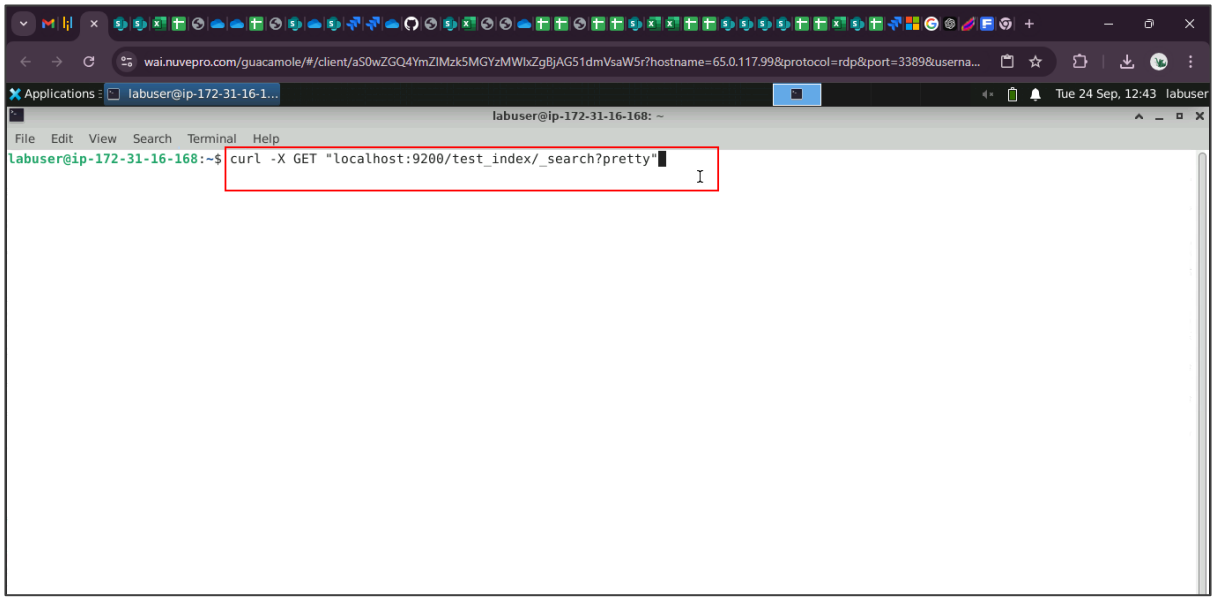


.

You will see the following output:



1.6 Run the command below to check if the data has been indexed in Elasticsearch by querying the **test_index**:
curl -X GET "localhost:9200/test_index/_search?pretty"



.

The output of the above command is as shown below:

By following these steps, you have successfully implemented Logstash for efficient log processing, transforming, and storing in Elasticsearch, enabling streamlined log management and real-time monitoring.