

## 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

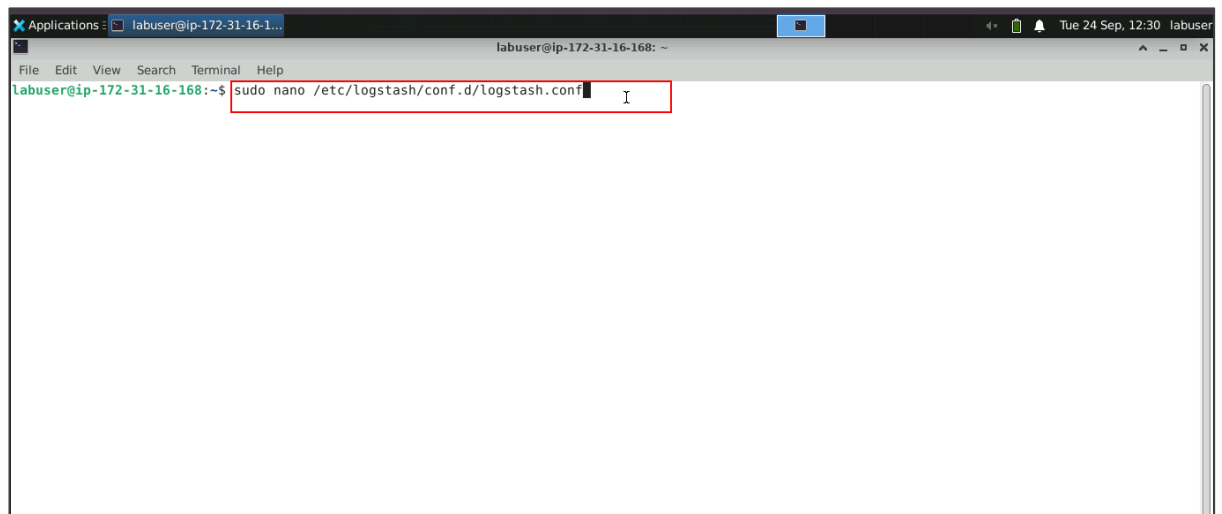
Steps to be followed:

1. Configure the Logstash pipeline

#### Step 1: Configure the Logstash pipeline

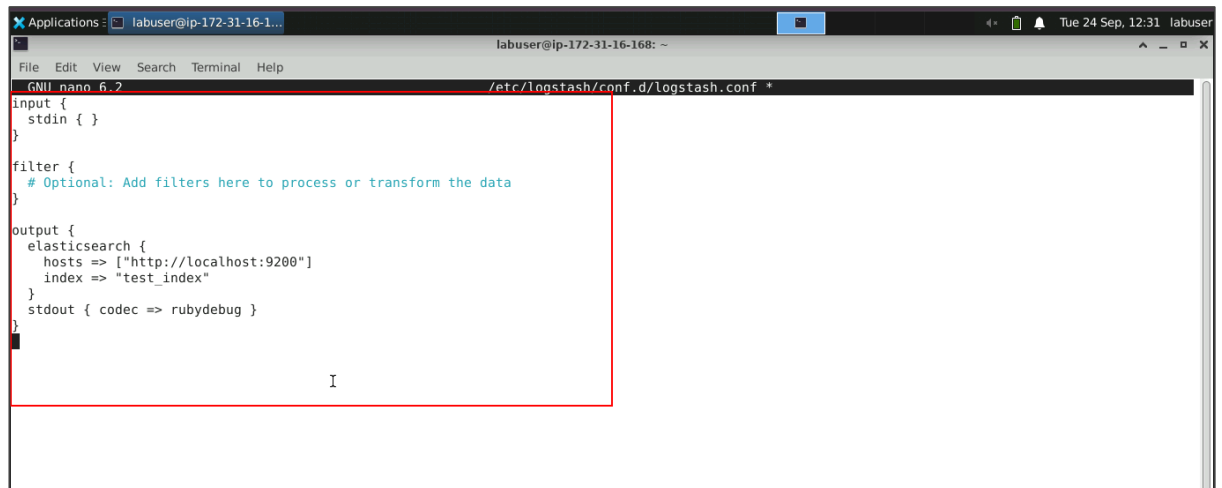
- 1.1 Open the terminal and run the command below to make changes to the following Logstash configuration file:

**`sudo nano /etc/logstash/conf.d/logstash.conf`**

A screenshot of a terminal window. The window title is 'Applications : labuser@ip-172-31-16-1...'. The terminal shows the command 'sudo nano /etc/logstash/conf.d/logstash.conf' being entered at the prompt 'labuser@ip-172-31-16-168:~\$'. The command is highlighted with a red rectangular box. The terminal also shows a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The system clock in the top right corner indicates 'Tue 24 Sep, 12:30'.

- 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 }  
}
```



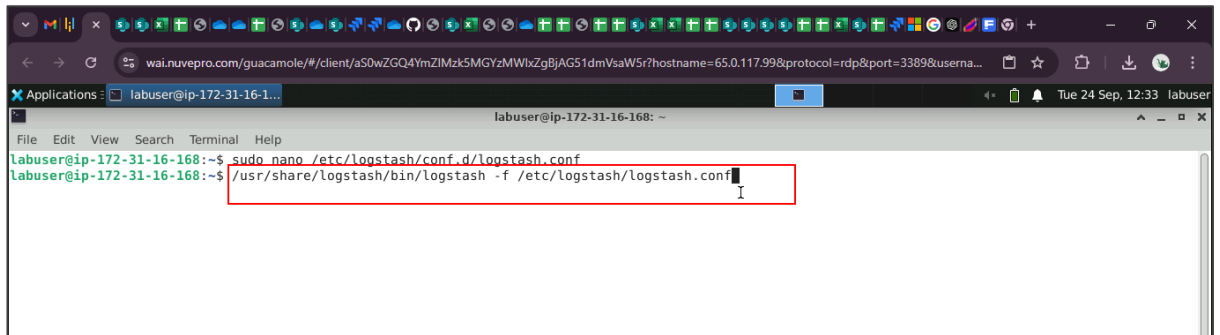
The screenshot shows a terminal window with the nano text editor open. The file being edited is `/etc/logstash/conf.d/logstash.conf`. The configuration content is as follows:

```
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 }  
}
```

The terminal window title bar indicates the user is `labuser` on a system with IP `172-31-16-168`. The nano editor status bar at the bottom shows `GNU nano 6.2` and the current file path.

- 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**

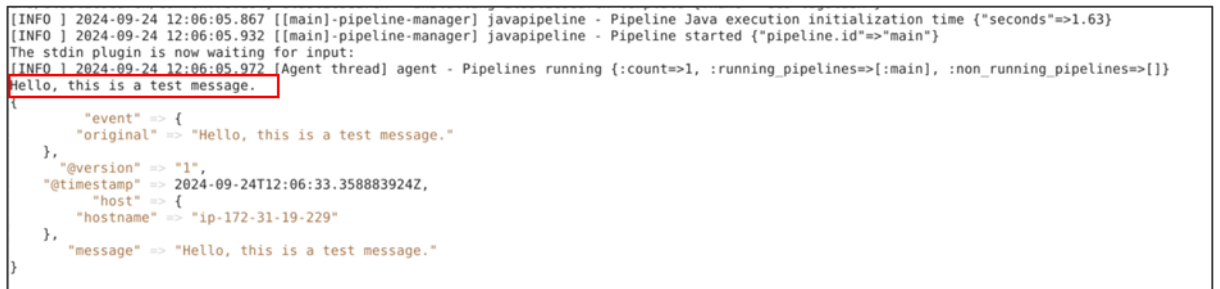


A screenshot of a terminal window. The terminal title is 'labuser@ip-172-31-16-168: ~'. The prompt is 'labuser@ip-172-31-16-168:~\$'. The command entered is 'sudo nano /etc/logstash/conf.d/logstash.conf'. The second line of the command is 'labuser@ip-172-31-16-168:~\$ /usr/share/logstash/bin/logstash -f /etc/logstash/logstash.conf'. The command is highlighted with a red box.

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:

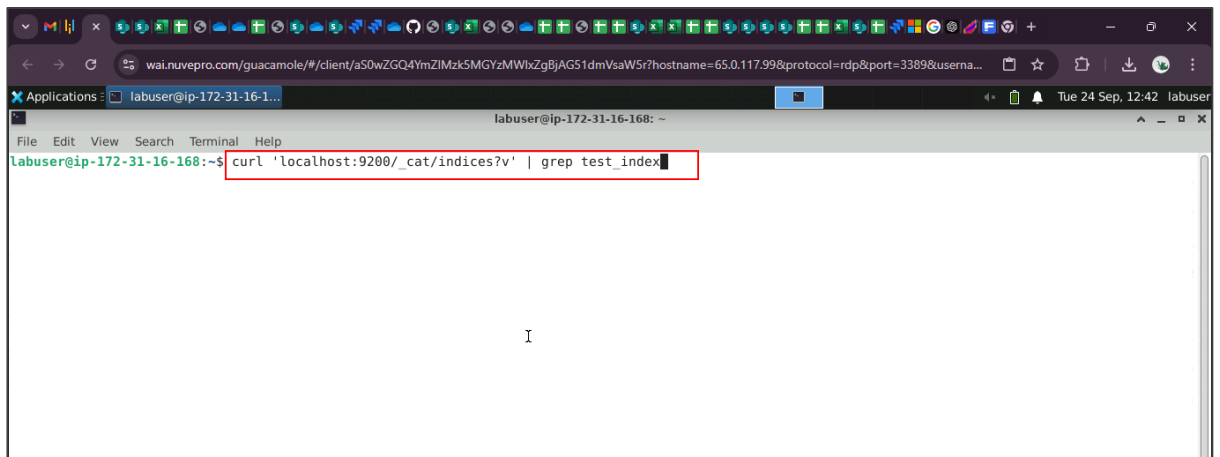
**Hello, this is a test message.**



A screenshot of the Logstash console output. The output shows several log messages. The first two are '[INFO ] 2024-09-24 12:06:05.867 [[main]-pipeline-manager] jvapiipeline - Pipeline Java execution initialization time {"seconds">=>1.63}' and '[INFO ] 2024-09-24 12:06:05.932 [[main]-pipeline-manager] jvapiipeline - Pipeline started {"pipeline.id">=>"main"}'. The third message is '[INFO ] 2024-09-24 12:06:05.972 (Agent thread) agent - Pipelines running {count=>1, :running\_pipelines=>[:main], :non\_running\_pipelines=>[]}'. The fourth message is 'Hello, this is a test message.' and is highlighted with a red box. Below this message is a JSON object: '{ "event" => { "original" => "Hello, this is a test message." }, "@version" => "1", "@timestamp" => 2024-09-24T12:06:33.358883924Z, "host" => { "hostname" => "ip-172-31-19-229" }, "message" => "Hello, this is a test message." }'.

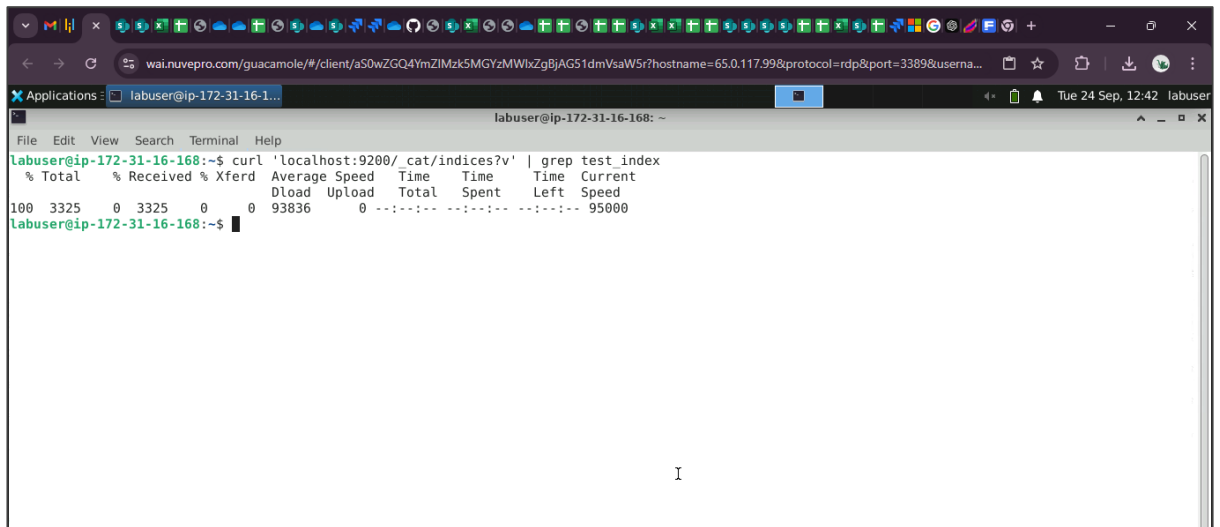
- 1.5 Run the command below in a new terminal tab to validate whether **test\_index** is created after the message is processed:

**curl 'localhost:9200/\_cat/indices?v' | grep test\_index**



A screenshot of a terminal window. The terminal title is 'labuser@ip-172-31-16-168: ~'. The prompt is 'labuser@ip-172-31-16-168:~\$'. The command entered is 'curl 'localhost:9200/\_cat/indices?v' | grep test\_index'. The command is highlighted with a red box.

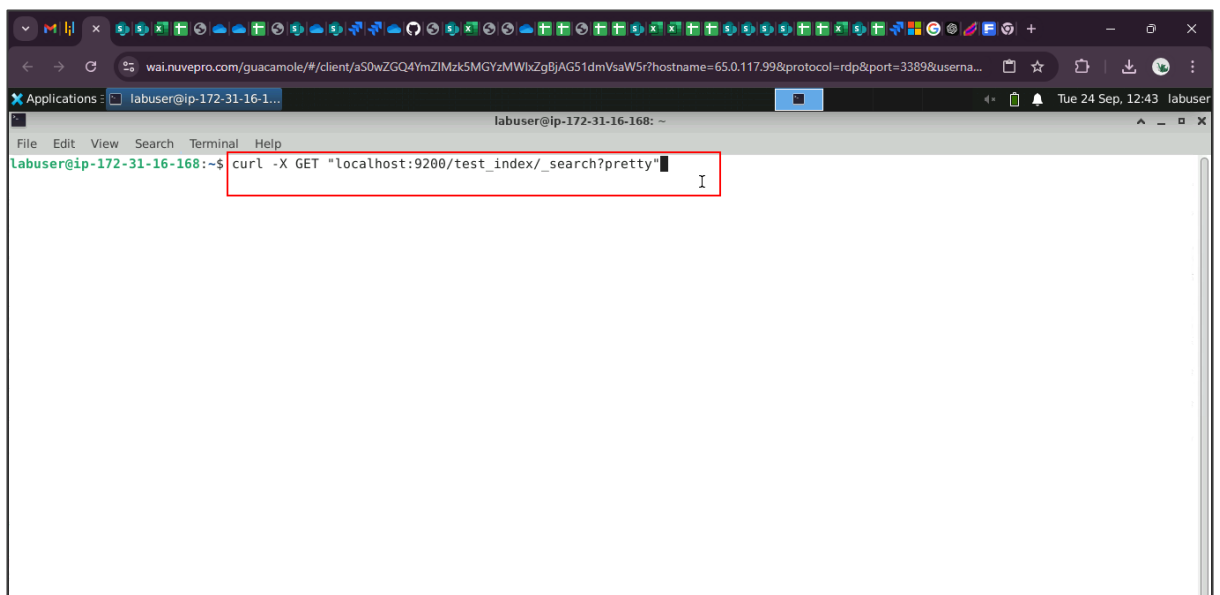
You will see the following output:



```
labuser@ip-172-31-16-168: ~  
labuser@ip-172-31-16-168:~$ curl 'localhost:9200/_cat/indices?v' | grep test_index  
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current  
           %         0         0             0          0          0     95000  
100  3325    0  3325    0     0    93836    0  --:--:--  --:--:--  --:--:--  95000  
labuser@ip-172-31-16-168:~$
```

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"**



```
labuser@ip-172-31-16-168: ~  
labuser@ip-172-31-16-168:~$ curl -X GET "localhost:9200/test_index/_search?pretty"
```

The output of the above command is as shown below:

```
"took" : 15,
"timed_out" : false,
"_shards" : {
  "total" : 1,
  "successful" : 1,
  "skipped" : 0,
  "failed" : 0
},
"hits" : {
  "total" : {
    "value" : 1,
    "relation" : "eq"
  },
  "max_score" : 1.0,
  "hits" : [
    {
      "_index" : "test index",
      "_id" : "ANPtISIBr2vX32x6GB1E",
      "_score" : 1.0,
      "_source" : {
        "event" : {
          "original" : "Hello, this is a test message."
        },
        "@version" : "1",
        "@timestamp" : "2024-09-24T12:06:33.358883924Z",
        "host" : {
          "hostname" : "ip-172-31-19-229"
        },
        "message" : "Hello, this is a test message."
      }
    }
  ]
}
```

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.