ELK stands for Elasticsearch, Logstash, and Kibana. It is a set of tools used for searching, analyzing, and visualizing large volumes of data, primarily logs.

Elasticsearch: It is a distributed, RESTful search and analytics engine. It is designed for horizontal scalability and real-time search and analysis of large amounts of data. It forms the core of the ELK stack, providing powerful search, aggregation, and data visualization capabilities.

Logstash: This is a server-side data processing pipeline that ingests data from various sources simultaneously, transforms it, and then sends it to your preferred “stash,” such as Elasticsearch. It handles data parsing, enriching, and formatting before it's stored.

Kibana: This tool provides visualization capabilities on top of the content indexed on an Elasticsearch cluster. It enables users to create and share dynamic dashboards that include charts, graphs, and other visualizations to analyze and gain insights from their data.

Installing ELK on a Ubuntu Server.

Here Ubuntu Server 24.04 is used. The configuration is 2 CPU, minimum 4 GB RAM, 40 GB HDD. Set a hostname for the server.

First update the system.

sudo apt update -y

sudo apt upgrade -y

Install Java. The ELK requires Java 11 or above. Here we install Java 11.

sudo apt install openjdk-11-jre -y

Install and Configure Elasticsearch.

Import the Elasticsearch GPG key to apt.

curl -fsSL https://artifacts.elastic.co/GPG-KEY-elasticsearch |sudo gpg --dearmor -o /usr/share/keyrings/elastic.gpg

Add Elastic source list to the sources.list.d directory. This is the directory where apt searches for the new sources.

echo "deb [signed-by=/usr/share/keyrings/elastic.gpg] https://artifacts.elastic.co/packages/7.x/apt stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-7.x.list

Update the package list

sudo apt update

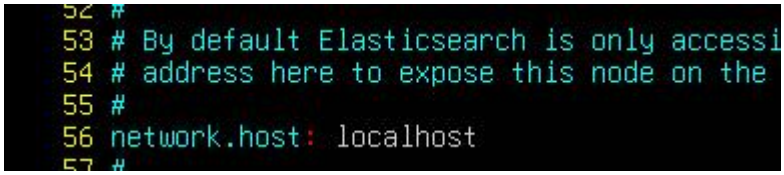
Now install Elasticsearch

sudo apt install elasticsearch

Configure the Elasticsearch. /etc/elasticsearch/elasticsearch.yml is the configuration file for Elasticsearch.

sudo nano /etc/elasticsearch/elasticsearch.yml

On line 56 following setting is there. Change the network.host setting to localhost as shown below.



Save the file.

Start the Elasticsearch service.

sudo systemctl start elasticsearch

Enable the Elasticsearch service for auto start.

sudo systemctl enable elasticsearch

Verify if Elasticsearch is running and receiving the http request.

curl -X GET "localhost:9200"

The output will be as shown below.

Install and Configure the Kibana

As per the official documentation, you should install Kibana only after the Elasticsearch is installed. This ensures the proper operation of each dependant component.

sudo apt install kibana

Now edit the Kibana configuration file.

sudo nano /etc/kibana/kibana.yml

Now make the following changes in the file.

```
# 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 host names are both valid values.
# The default is 'localhost', which usually means remote machines will not be able to connect.
# To allow connections from remote users, set this parameter to a non-loopback address.
server.host: "192.168.234.146"

```

Kibana dashboard by default works on port 5601. However to make sure you can enable this option. Kibana is normally works as a backend server. It is normally placed behind a reverse proxy like Nginx. However he we are running it on the Ubuntu server IP address. Thus in the server.host filed type the IP address of your Ubuntu server.

Save the file.

Enable and start the Kibana service for auto start.

sudo systemctl enable kibana
sudo systemctl start kibana

Now open the browser and type http://ubuntu-server-ip:5601

Wait for some time. The Kibana dashboard will be displayed. However currently it will not display any logs.

Install Logstash

Install Logstash

sudo apt install logstash

Configure logstash to receive input from beats. Also we will configure logstash to send the output to the Elasticsearch.

A Logstash pipeline has two required elements, input and output, and one optional element, filter. The input plugins consume data from a source, the filter plugins process the data, and the output plugins write the data to a destination.

Go to the /etc/logstash/conf.d directory.

cd /etc/logstash/conf.d

Create a pipeline file. You can specify any name for this file. Make sure the extension is .conf.

sudo nano demo-pipeline.conf

Type following into it.

```
input {
  beats {
    port => 5044
  }
}
#filter {
#}
output {
  if [@metadata][pipeline] {
            elasticsearch {
            hosts => ["localhost:9200"]
            manage_template => false
            index => "%{[@metadata][beat]}-%{[@metadata][version]}-%{+YYYY.MM.dd}"
            pipeline => "%{[@metadata][pipeline]}"
            }
  } else {
            elasticsearch {
            hosts => ["localhost:9200"]
            manage_template => false
            index => "%{[@metadata][beat]}-%{[@metadata][version]}-%{+YYYY.MM.dd}"
            }
  }
}
```

Save the File.

Tthis output configures Logstash to store the Beats data in Elasticsearch, which is running at localhost:9200, in an index named after the Beat used.

Test your Logstash configuration using following command.

sudo -u logstash /usr/share/logstash/bin/logstash --path.settings /etc/logstash -t

You should get *Config Validation Result: OK. Exiting Logstash* Message.

If evrything is OK. Start the Logstash service.

sudo systemctl start logstash

Enable the service.

sudo systemctl enable logstash

Beats collect data from various sources and transport them to Logstash or Elasticsearch. There are various Beats used by the Elasticsearch.

Below are the some of the examples of the Beats used.

Filebeat: collects and ships log files.
Metricbeat: collects metrics from your systems and services.
Packetbeat: collects and analyzes network data.
Winlogbeat: collects Windows event logs.
Auditbeat: collects Linux audit framework data and monitors file integrity.
Heartbeat: monitors services for their availability with active probing.

Install and configure Filebeat

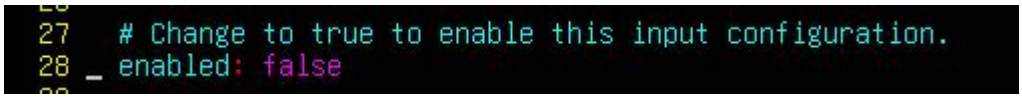
Install Filebeat. Here we are sending the local logs to the local Logstash server.

sudo apt install filebeat

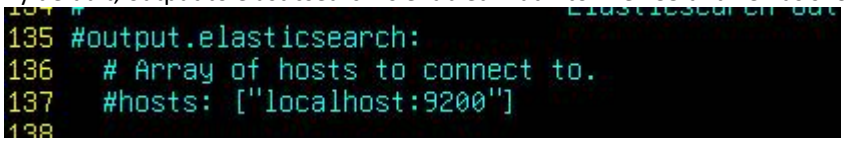
Configure Filebeat to send data to Logstash.

sudo nano /etc/filebeat/filebeat.yml

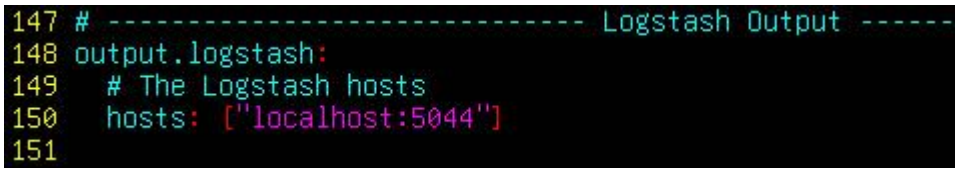
On line 28 change the enabled: false to true



By default, output to elasticsearch is enabled. Put # to line 135 and 137 as shown below.



Now enable output to logstash as shown below by removing hash from line 148 and 150.



Make sure the hosts: contains “localhost:5044”

Save the file.

Filebeat comes with lot of modules. Here we want to send system logs to the logstash. Thus we will enable the system module of the filebeat.

sudo filebeat modules enable system

To verify that system module is enable, give following command. Check in the enabled section of the output.

sudo filebeat modules list

Next step is to load the index template into Elasticsearch.

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

You should get the output as -- Index setup finished.

Filebeat comes with some sample Kibana dashboards. These dashboards allow you to visualize Filebeat data in Kibana. You need to create the index pattern and load the dashboards into the Kibana. As the Filebeat is configured to connect to Logstash, we will temporarily disable this and connect Filebeat to Elasticsearch.

sudo filebeat setup -E output.logstash.enabled=false -E output.elasticsearch.hosts=['localhost:9200'] -E setup.kibana.host=<Put-Server-IP>:5601

Start the Filebeat service.

sudo systemctl start filebeat

Enable the Filebeat service.

sudo systemctl enable filebeat

To verify that Filebeat data is received by the Elasticsearch, give following command.

curl -XGET 'http://localhost:9200/filebeat-*/_search?pretty'

Use Kibana Dashboard

Now open the browser and type http://ubuntu-server-ip:5601

Click the Discover link in the left-hand navigation bar. On the Discover page, select the predefined filebeat-* index pattern to see Filebeat data.

If you are able to see the log data of the local server, it means everything is configured properly.
Open TCP ports 5601 and 5044 in the firewall of this server.

Add external server for monitoring

Here we are using a Rocky Linux 9.5 server. Set a separate hostname for this server.

Use following link to install Filebeat on other Operating Systems.

www.elastic.co/docs/reference/beats/filebeat/filebeat-installation-configuration

Install Filebeat on this server.

curl -L -O https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-9.0.0-x86_64.rpm

sudo rpm -vi filebeat-9.0.0-x86_64.rpm

Edit the Filebeat configuration file.

sudo vi /etc/filebeat/filebeat.yml

Configure Filebeat to send logs to Logstash Server installed above.

```
27   # Change to true to enable this input configuration.
28   enabled: true
```

Change enabled: to true

```
31   paths:
32     - /var/log/messages
33     -/var/log/auth.log_
34     #- c:\programdata\elasticsearch\logs\*
35
```

Change the paths section and add two log file names. These logs will be sent to the ELK stack.

```
93 # ======================== Elasticsearch template setting ======
94
95 #setup.template.settings:
96   #index.number_of_shards: 1
97   #index.codec: best_compression
98   #_source.enabled: false
```

Disable lines 95 to 98 as shown above.

```
162 # ---------------------------- Elasticsearch Output ----------------------------
163 #output.elasticsearch:
164   # Array of hosts to connect to.
165   #hosts: ["localhost:9200"]
166
167   # Performance preset - one of "balanced", "throughput", "scale",
168   # "latency", or "custom".
169   preset: balanced
170
171   # Protocol - either `http` (default) or `https`.
172   #protocol: "https"
173
174   # Authentication credentials - either API key or username/password.
175   #api_key: "id:api_key"
176   #username: "elastic"
177   #password: "changeme"
178
179 # ------------------------------ Logstash Output -------------------------------
180 output.logstash:
181   # The Logstash hosts
182   hosts: ["192.168.234.146:5044"]
```

Here change the configuration so that Filebeat will send the logs to remote Logstash server rather that sending them to Elasticsearch. Please provide your Ubuntu server IP address in the hosts: fileld in the Logstash section.

Now save the file.

Enable Filebeat system module.

sudo filebeat modules enable system

Load Filebeat assets

sudo filebeat setup -e

Now verify if this Filebeat is able to communicate with the Logstash server.

Go to the ELK server (Ubuntu Server)

Stop the Filebeat service.

sudo systemctl stop filebeat.

Stop the Logstash service.

sudo systemctl stop logstash

Now start the logstash manually using following command.

cd /usr/share/logstash

sudo bin/logstash -f /etc/logstash/conf.d/demo-pipeline.conf --config.reload.automatic

Wait for some time. You should get a message Server started on port 5044.

Now go to the Rocky Linux server.

Give the following command, to start filebeat manually.

cd /etc/filebeat/modules.d

Edit the system.yml file

sudo vi system.yml

Now make changes so that the file looks as shown below.

```
# Module: system
# Docs: https://www.elastic.
ule-system.html

- module: system
  # Syslog
  syslog:
    enabled: true
```

Change syslog : enabled to true

```
    # Use journald to collect system logs
    #var.use_journald: false

  # Authorization logs
  auth:
    enabled: true
```

Also change auth: enabled to true.
Save the file.

sudo filebeat -e -d “publish”

Wait for some time. Check if the Filebeat is able to communicate with the Logstash server.

If everything is successful , then stop both the processes.

Start the Logstash service and Filebeat service on Ubuntu server using systemctl command.
Start the Filebeat service on the Rocky Linux server using systemctl command.

Now check in the Kibana dashboard. Check if logs from both the machines is received.