

Deploying and Automating ELK Stack on On-Premises Servers with Filebeat Integration

1. Introduction:-

The ELK Stack is a powerful set of open-source tools for collecting, processing, and visualizing logs and metrics. The core components are:

- **Elasticsearch:** The search engine for storing logs.
- **Logstash:** The data processing pipeline.
- **Kibana:** A web interface for visualizing and analyzing data.
- **Filebeat:** A lightweight shipper for forwarding and centralizing logs to Logstash or Elasticsearch.

2. Why We Need Log Capturing:-

2.1 Importance of Log Capturing

Logs are essential for maintaining the security, performance, and reliability of IT infrastructure. Capturing and analyzing these logs is crucial for several reasons:

1. **Security Monitoring:** Logs help detect unauthorized access, malware infections, and other security incidents in real-time.
2. **Incident Response and Troubleshooting:** Logs provide valuable insights when diagnosing issues.
3. **Compliance and Auditing:** Logs serve as an audit trail for regulatory compliance (GDPR, HIPAA, PCI-DSS).
4. **Operational Efficiency:** Analyze logs to optimize system performance.
5. **Forensic Investigations:** Logs help reconstruct events after a security breach or incident.

3. Why We Selected the ELK Stack

3.1 Advantages of the ELK Stack

The ELK Stack is chosen for its flexibility, scalability, and cost-effectiveness, with advantages such as:

1. **Open-Source:** Free and customizable without licensing costs.

2. **Real-Time Analysis:** Supports real-time data collection and analysis.
3. **Scalability:** Elasticsearch's distributed architecture enables scaling to handle large volumes of data.
4. **Search Capabilities:** Elasticsearch provides powerful search and filtering options.
5. **Data Visualization:** Kibana enables creating custom dashboards to monitor and visualize logs.
6. **Centralized Log Management:** Logs from multiple systems are managed in one place for easier analysis.

3.2 Why ELK Over Other Solutions?

Compared to Splunk: ELK is a cost-effective alternative, offering comparable functionality without licensing fees.

Compared to Graylog: ELK offers more flexibility and scalability, with better community support.

4. Installing and Configuring the ELK Stack on On-Premises Servers

Step 1: Installing Elasticsearch

Elasticsearch is the engine that stores logs and allows you to search, analyze, and visualize data.

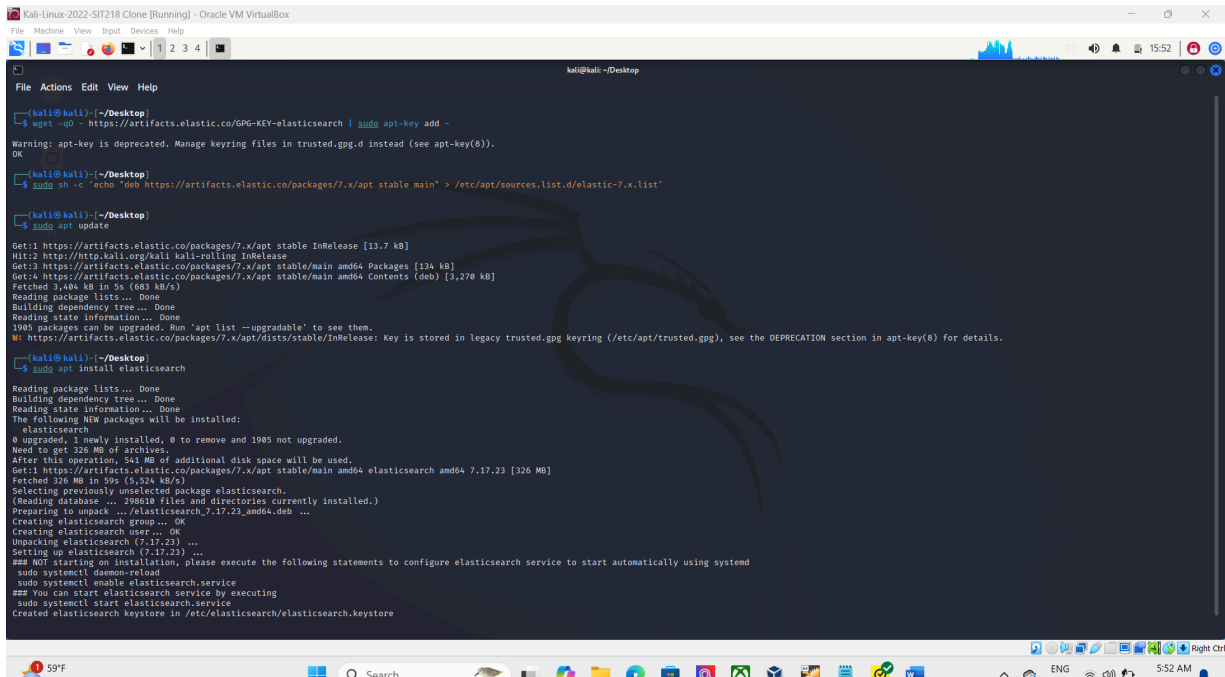
1. **Add the GPG Key and Repository:** This step adds the Elasticsearch repository to your system so that it can download the necessary packages. we run following commands to do it

```
wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo tee /usr/share/keyrings/elasticsearch-archive-keyring.gpg > /dev/null
echo "deb [signed-by=/usr/share/keyrings/elasticsearch-archive-keyring.gpg] https://artifacts.elastic.co/packages/7.x/apt stable main" | sudo tee /etc/apt/sources.list.d/elastic-7.x.list
```



2. Install and Start Elasticsearch: Here, you install Elasticsearch and enable it to start on boot

```
sudo apt update
sudo apt install elasticsearch
sudo systemctl start elasticsearch
sudo systemctl enable elasticsearch
```



```
kali@kali:~/Desktop
$ wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -
warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK

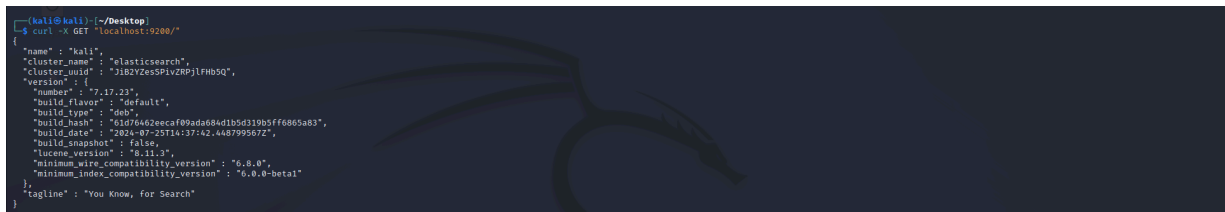
kali@kali:~/Desktop
$ sudo sh -c 'echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" > /etc/apt/sources.list.d/elasticsearch-7.x.list'

kali@kali:~/Desktop
$ sudo apt update
Get:1 https://artifacts.elastic.co/packages/7.x/apt stable InRelease [13.7 kB]
Hit:2 http://kali.org/kali kali-rolling InRelease
Get:3 https://artifacts.elastic.co/packages/7.x/apt stable/main amd64 Packages [134 kB]
Get:4 https://artifacts.elastic.co/packages/7.x/apt stable/main amd64 Contents (deb) [3,270 kB]
Fetched 3,418 kB in 5s (683 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
1985 packages can be upgraded. Run 'apt list --upgradable' to see them.
Warning: https://artifacts.elastic.co/packages/7.x/apt/dists/stable/InRelease: Key is stored in legacy trusted.gpg Keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.

kali@kali:~/Desktop
$ sudo apt install elasticsearch
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  elasticsearch
0 upgraded, 1 newly installed, 0 to remove and 1985 not upgraded.
Need to get 326 MB of archives.
After this operation, 541 MB of additional disk space will be used.
Get:1 https://artifacts.elastic.co/packages/7.x/apt stable/main amd64 elasticsearch amd64 7.17.23 [326 MB]
Fetched 326 MB in 50s (6,526 kB/s)
Selecting previously unselected package elasticsearch.
(Reading database ... 298610 files and directories currently installed.)
Preparing to unpack .../elasticsearch-7.17.23_amd64.deb ...
Unpacking elasticsearch (7.17.23) ...
Setting up elasticsearch (7.17.23) ...
## NOT starting on installation, please execute the following statements to configure elasticsearch service to start automatically using systemd
  sudo systemctl daemon-reload
  sudo systemctl enable elasticsearch.service
## You can start elasticsearch service by executing
  sudo systemctl start elasticsearch.service
Created elasticsearch keystore in /etc/elasticsearch/elasticsearch.keystore
```

3. Verify Elasticsearch Installation: After installation, test that Elasticsearch is running by making an HTTP request to **localhost:9200**. It should return information about the Elasticsearch node.

```
curl -X GET "localhost:9200/"
```



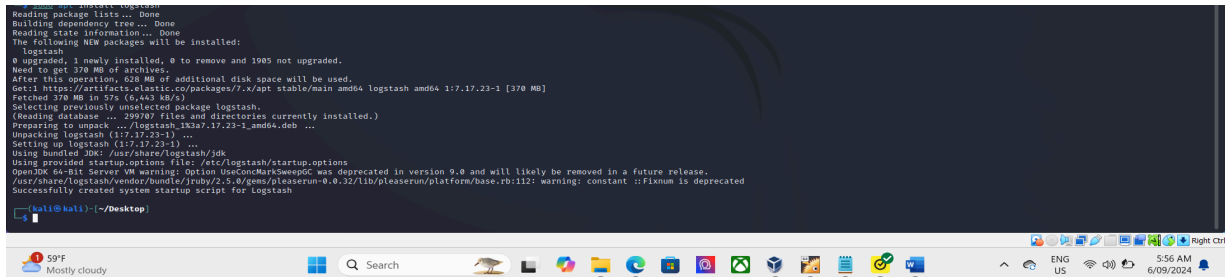
```
kali@kali:~/Desktop
$ curl -X GET "localhost:9200/"
{
  "name": "kali",
  "cluster_name": "elasticsearch",
  "cluster_uuid": "3182V2esSPivZNPjLHbDQ",
  "version": {
    "number": "7.17.23",
    "build_flavor": "default",
    "build_type": "deb",
    "build_hash": "61d7642eeca09ada60ad1b5d319b5ff6865a83",
    "build_date": "2024-07-22T14:37:42.440795672",
    "build_snapshot": false,
    "lucene_version": "8.11.3",
    "minimum_wire_compatibility_version": "6.8.0",
    "minimum_index_compatibility_version": "6.0.0-beta1"
  },
  "tagline": "You Know, for Search"
}
```

Step 2: Installing Logstash

Logstash processes logs sent by Filebeat and forwards them to Elasticsearch.

1.Install Logstash: This command installs Logstash on your system.

```
sudo apt install logstash
```



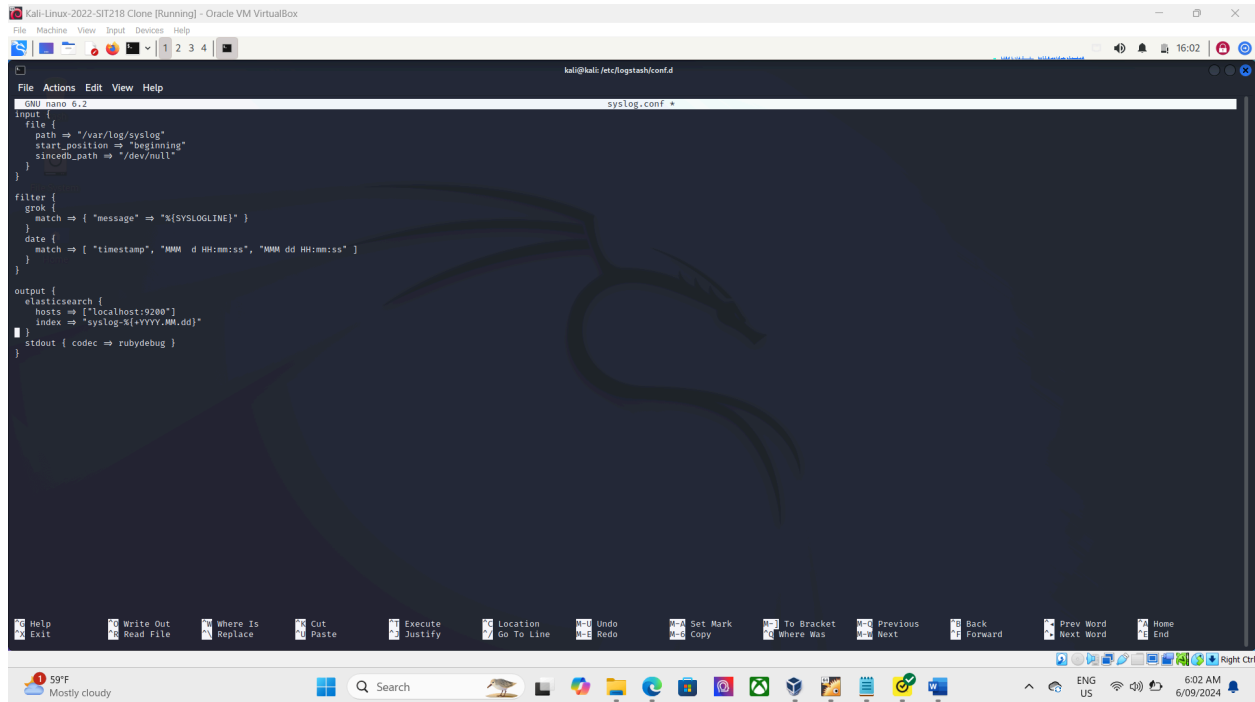
2.Configure Logstash: A configuration file (`02-beats-input.conf`) is created to specify that Logstash should listen for data on port 5044 from Filebeat. This file also defines Elasticsearch as the destination for the processed logs

```
sudo nano /etc/logstash/conf.d/02-beats-input.conf
```

Add the following configuration:

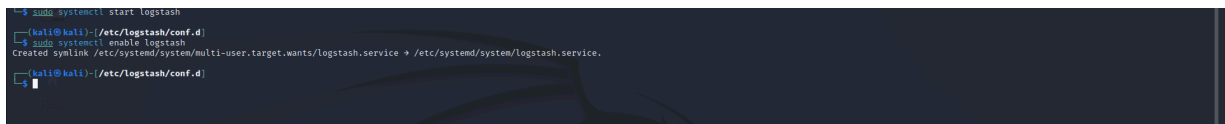
```
input {
  beats {
    port => 5044
  }
}

output {
  elasticsearch {
    hosts => ["localhost:9200"]
    index =>
"%{[@metadata][beat]}-%{[@metadata][version]}-%{+YYYY.MM.dd}"
  }
  stdout { codec => rubydebug }
}
```



3.Start and Enable Logstash: Start Logstash and configure it to start on boot

```
sudo systemctl start logstash
sudo systemctl enable logstash
```



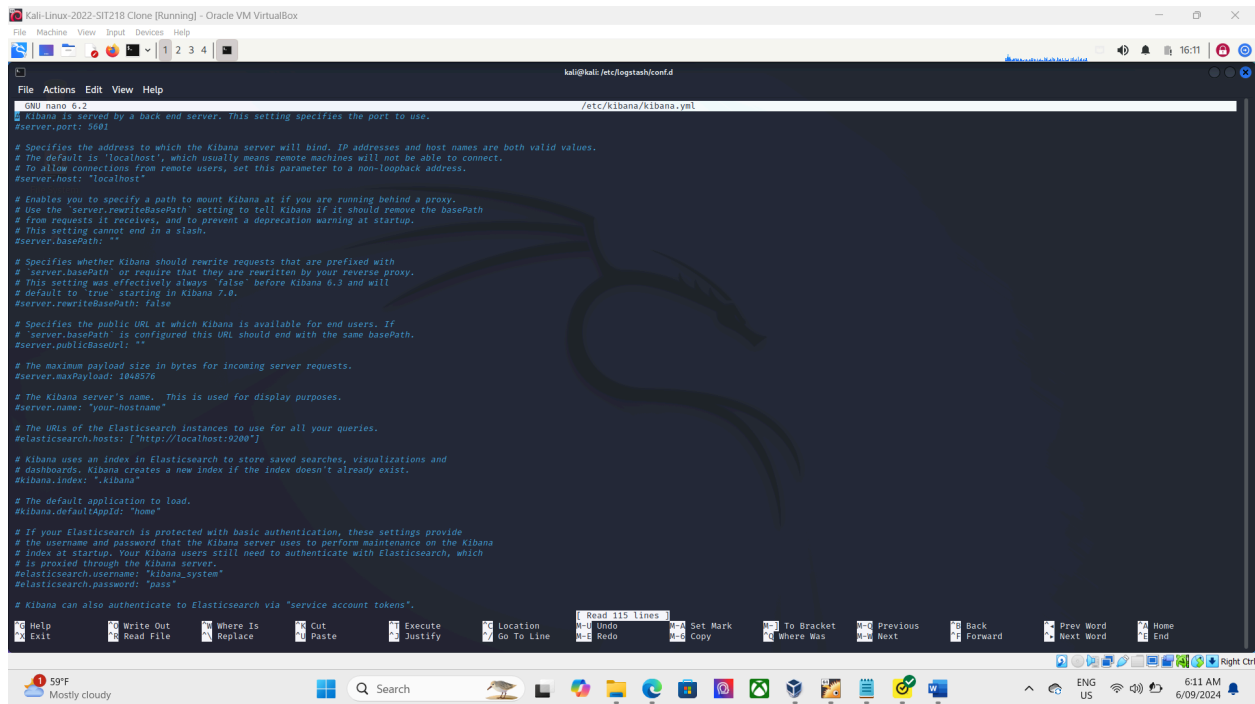
Step 3: Installing Kibana

Kibana provides the user interface to visualize and analyze data stored in Elasticsearch.

1. **Install Kibana:** Install Kibana using the package manager.

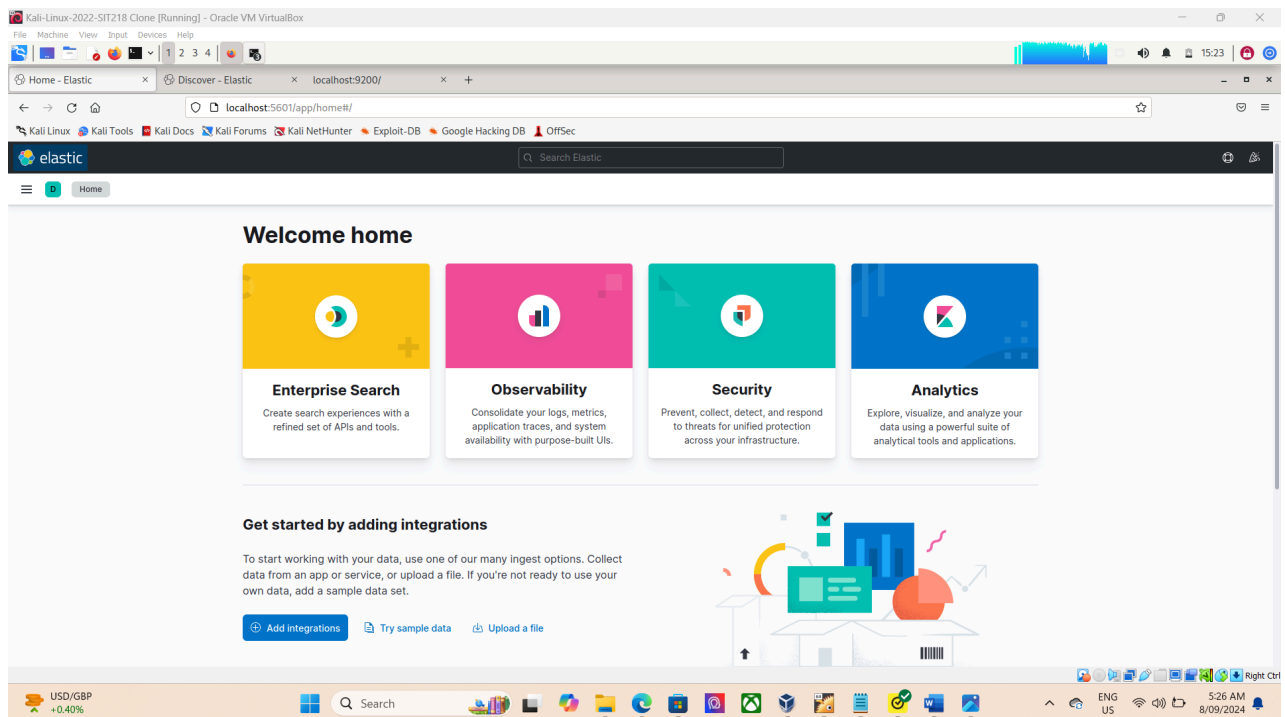
```
sudo apt install kibana
```

2. **Configure Kibana:** In Kibana's configuration file (/etc/kibana/kibana.yml), modify server.host to 0.0.0.0 to make Kibana accessible from other devices on the network.



```
sudo systemctl start kibana
sudo systemctl enable kibana
```

3. Access Kibana: You can access Kibana by opening a web browser and going to <http://localhost:5601>. This will bring up the Kibana dashboard.



5. Setting Up Filebeat for Log Collection

Filebeat collects and forwards logs to Logstash for processing.

1. **Install Filebeat:** Install Filebeat on your system to start shipping logs

```
sudo apt install filebeat
```

2. **Configure Filebeat:** Modify the Filebeat configuration file (/etc/filebeat/filebeat.yml) to send data to Logstash.

```
output.logstash:
```

```
  hosts: ["localhost:5044"]
```

Additionally, configure Filebeat to collect logs from `/var/log/syslog` or any other specific log file.

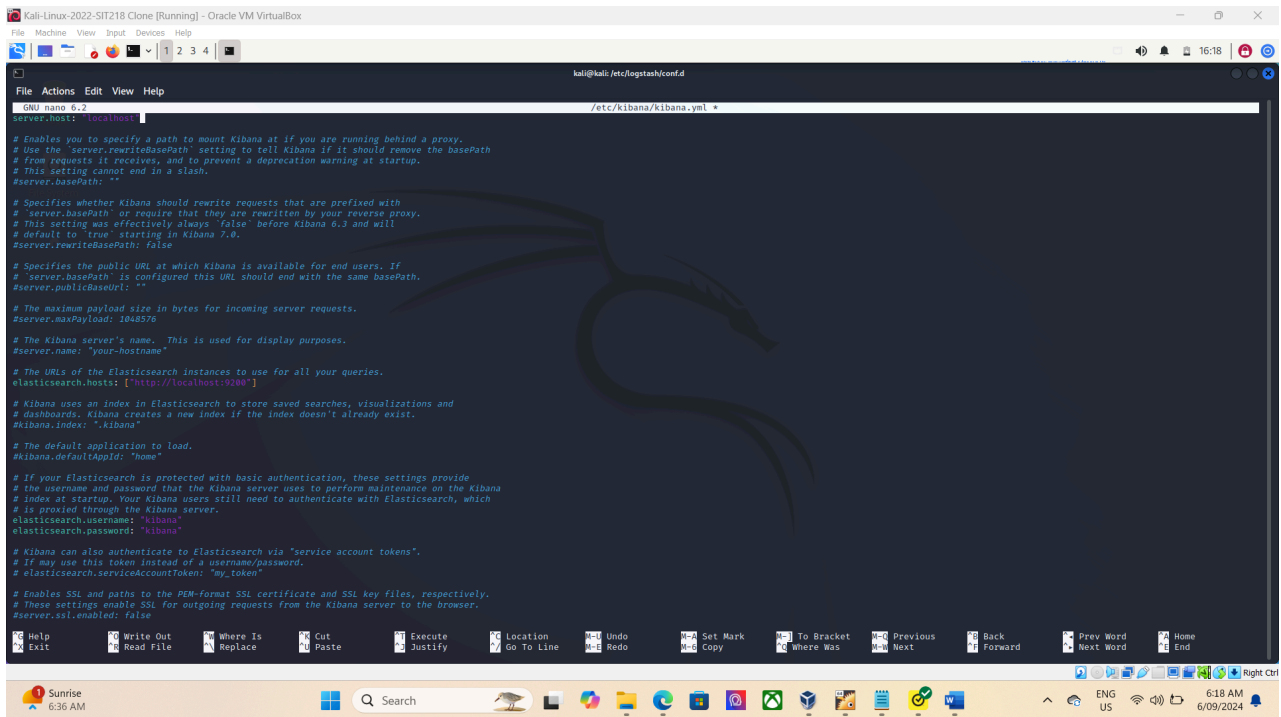
```
filebeat.inputs:
```

```
- type: log
```

```
  enabled: true
```

```
  paths:
```

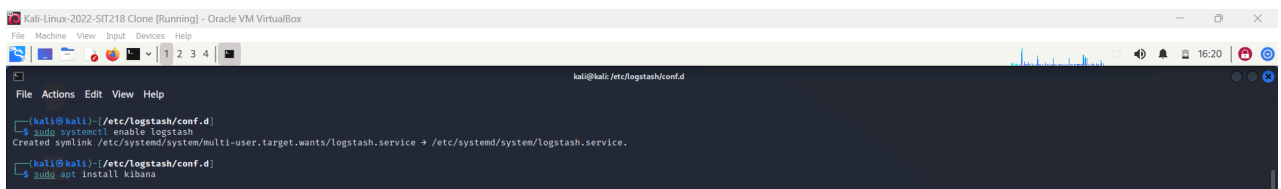
```
    - /var/log/syslog
```



3. Start Filebeat: Start Filebeat and enable it to start at boot.

```
sudo systemctl start filebeat
```

```
sudo systemctl enable filebeat
```



6. Verifying the ELK Stack Setup

6.1 Check Elasticsearch Indices

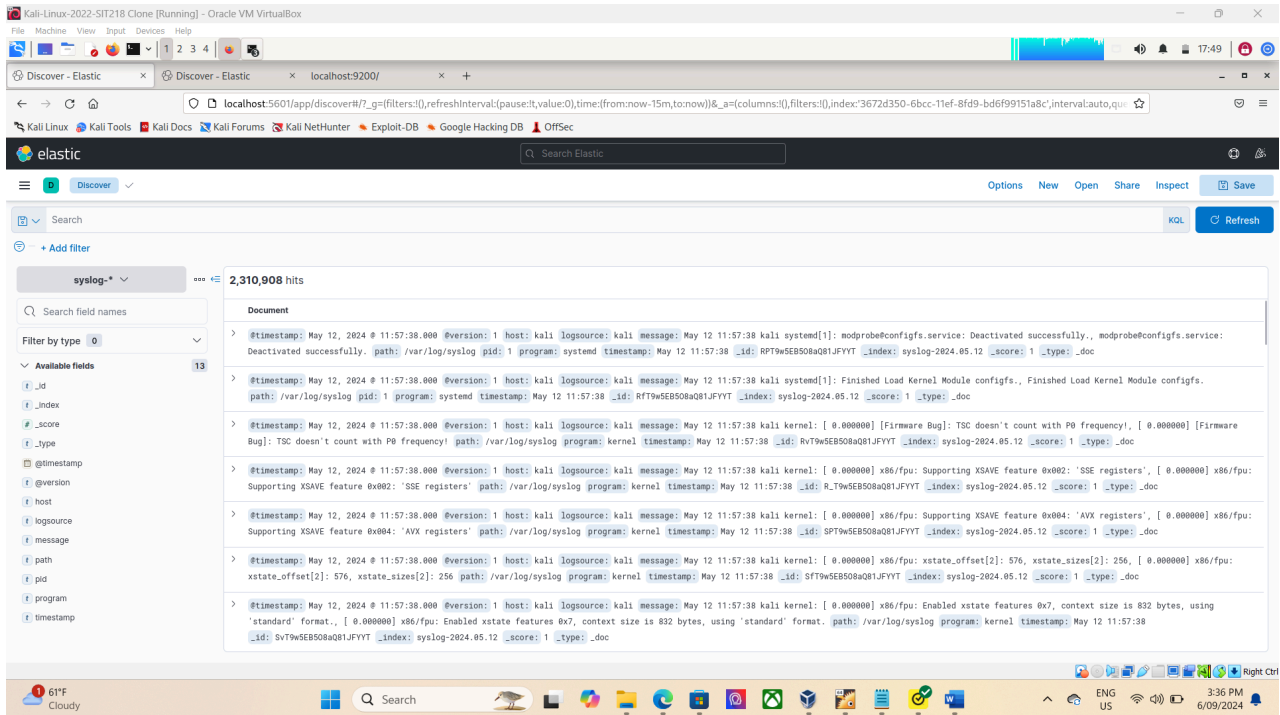
To verify logs are being received by Elasticsearch:

```
curl -X GET "localhost:9200/_cat/indices?v"
```

Look for indices with the prefix **filebeat-***.

6.2 Check Logs in Kibana

1. Go to **Kibana > Discover**.
2. Select the **filebeat-*** index pattern.
3. Logs should appear in Kibana if Filebeat is successfully sending logs.



7. Common Issues and Resolutions

Issue 1: Filebeat Service Fails to Start

Error: Filebeat service is inactive (dead)

Resolution:

Test the Filebeat configuration:

```
sudo filebeat test config
```

1. Test the Filebeat configuration:
2. Ensure only one output is configured (either Elasticsearch or Logstash).
3. Restart Filebeat:

```
sudo systemctl restart filebeat
```

Issue 2: Logstash Unable to Read /var/log/syslog

Error: Permission denied - /var/log/syslog

Resolution:

Add Logstash to the adm group:

```
sudo usermod -a -G adm logstash
```

```
sudo systemctl restart logstash
```

2. Alternatively, use Filebeat to collect logs and forward them to Logstash.

Issue 3: No Data in Kibana

Symptom: No logs in the Kibana Discover tab.

Resolution:

Ensure logs are indexed by Elasticsearch:

```
curl -X GET "localhost:9200/filebeat*/_search?pretty"
```

1. Confirm Kibana's index pattern:
 - Go to **Kibana > Stack Management > Index Patterns** and ensure filebeat-* is set up.
2. Restart services if necessary:

```
sudo systemctl restart elasticsearch logstash kibana
```

8. Automating the ELK Stack Deployment (Further Steps)

8.1 Automating ELK Deployment as a Future Enhancement

While manually installing and configuring the ELK Stack works well for initial setups, automation is a **further step** that significantly improves deployment efficiency and consistency across multiple environments.

Automation ensures repeatable processes, reducing manual errors and saving time in larger

deployments. Tools like **Ansible**, **Jenkins**, and **Docker** can be used to automate ELK Stack deployment.

Automating with Ansible

Ansible can be used to automate the ELK Stack deployment on multiple servers.