ELK Setup, Log Ingestion & Attack Detection

WE Innovate X Zero$ploit

Supervised by : Ahmed Halwagy – Zeyad Mazen – Adel Ahmed

Prepared by : Omar Hassan

Required Tasks

- Installing & configuring Elasticsearch
- Installing & configuring Kibana
- Connecting Elasticsearch with kibana
- Installing & configuring Fluentbit
- Installing & Configuring Winlogbeat
- Writing detection rules & simulating a suspicious activity

Requirements

- VMware / Virtual Box
- Windows 10/11 ISO – Ubuntu (20.0/22.0/24.0) ISO
- 16 GB RAM – 60 GB Disk Space
- 4 CPU Cores

Ubuntu Machine

Setting	Recommended
RAM	5-6 GB
Disk	20-30 GB
CPU	2-3 Cores
Network	NAT

Windows Machine

Setting	Recommended
RAM	2-3 GB
Disk	30-40 GB
CPU	1-2 Cores
Network	NAT

PHASE 1 : Installing & configuring Elasticsearch

Updating Ubuntu packages

```
$ sudo apt update
$ sudo apt upgrade -y
```

Installing required packages & dependencies

```
$ wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch |
sudo gpg --dearmor -o /usr/share/keyrings/elasticsearch-keyring.gpg

$ sudo apt-get install apt-transport-https

$ echo "deb [signed-by=/usr/share/keyrings/elasticsearch-
keyring.gpg] https://artifacts.elastic.co/packages/9.x/apt stable
main" | sudo tee /etc/apt/sources.list.d/elastic-9.x.list

$ sudo apt-get update && sudo apt-get install elasticsearch
```

Configuring elasticsearch.yml

```
$ sudo nano /etc/elasticsearch/elasticsearch.yml
```

Lines to be uncommented : Network.host & http.port

Lines to be added :

discovery.type: single-node

```
#
# By default Elasticsearch is only accessible on localhost. Set a different
# address here to expose this node on the network:
#
network.host: 0.0.0.0
#
# By default Elasticsearch listens for HTTP traffic on the first free port it
# finds starting at 9200. Set a specific HTTP port here:
#
http.port: 9200
#
# For more information, consult the network module documentation.
#
# ---------------------------------- Discovery ----------------------------------
#
# Pass an initial list of hosts to perform discovery when this node is started:
# The default list of hosts is ["127.0.0.1", "[::1]"]
#
#discovery.seed_hosts: ["host1", "host2"]
#
#bootstrap.memory_lock: false
discovery.type: single-node
```

Enabling & starting the Elasticsearch service

```
$ sudo systemctl enable elasticsearch
$ sudo systemctl start elasticsearch
$ sudo systemctl status elasticsearch
```

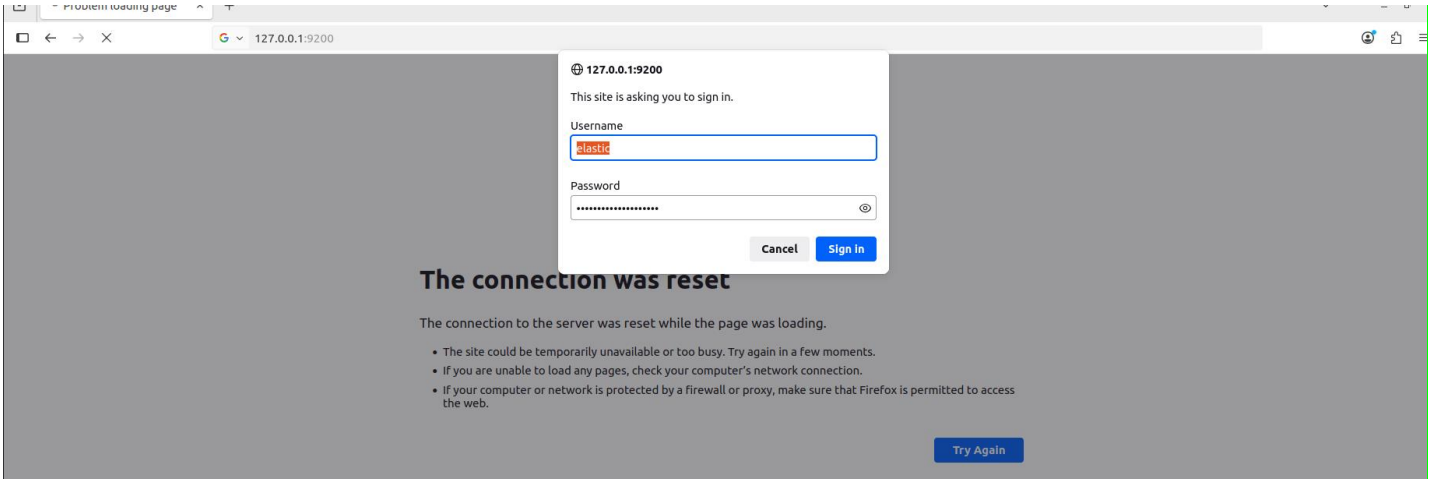
If you receive **active(running)** then everything is working

```
omar@omar-virtual-machine:~$ sudo systemctl enable elasticsearch
omar@omar-virtual-machine:~$ sudo systemctl start elasticsearch
omar@omar-virtual-machine:~$ sudo systemctl status elasticsearch
● elasticsearch.service - Elasticsearch
     Loaded: loaded (/lib/systemd/system/elasticsearch.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2025-08-14 20:57:52 EEST; 44min ago
       Docs: https://www.elastic.co
   Main PID: 951 (java)
      Tasks: 99 (limit: 6841)
     Memory: 1.5G
        CPU: 3min 58.995s
     CGroup: /system.slice/elasticsearch.service
             ├─ 951 /usr/share/elasticsearch/jdk/bin/java -Xms4m -Xmx64m -XX:+UseSerialGC -Dcli.name=server -Dcli.script=/usr/share/elasticsearch/bin/e
             ├─1548 /usr/share/elasticsearch/jdk/bin/java -Des.networkaddress.cache.ttl=60 -Des.networkaddress.cache.negative.ttl=10 -XX:+AlwaysPreTouc
             └─1991 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux-x86_64/bin/controller

20:54:26 14 أغس omar-virtual-machine systemd[1]: Starting Elasticsearch...
20:57:52 14 أغس omar-virtual-machine systemd[1]: Started Elasticsearch.
lines 1-15/15 (END)
```

Restarting the elasticsearch service

```
$ sudo systemctl restart elasticsearch
```

Check connectivity by going to **https://127.0.0.1:9200/** on your web browser on Ubuntu.



On the first go it should ask you about a username & password , the default username is **elastic , the password should be reset** using the following command.

```
$ sudo /usr/share/elasticsearch/bin/elasticsearch-reset-password -u
elastic
```

Then you can save your password in a txt file for later use , you can now login into elasticsearch using username elastic & the password displayed in the terminal when you reset it.

After logging in


```
JSON   Raw Data   Headers
Save  Copy  Collapse All  Expand All  Filter JSON
name:                                  "omar-hassan"
cluster_name:                          "elk-demo"
cluster_uuid:                          "CFmFqNn-TZ2iA0r7Foq-5g"
version:
  number:                              "9.1.2"
  build_flavor:                        "default"
  build_type:                          "deb"
  build_hash:                          "ca1a70216fbdefbef3c65b1dff04903ea5964ef5"
  build_date:                          "2025-08-11T15:04:41.449624592Z"
  build_snapshot:                      false
  lucene_version:                      "10.2.2"
  minimum_wire_compatibility_version:  "8.19.0"
  minimum_index_compatibility_version: "8.0.0"
tagline:                               "You Know, for Search"
```

PHASE 2 : Installing & configuring Kibana

```
$ sudo apt-get update && sudo apt-get install kibana
$ sudo nano /etc/kibana/kibana.yml
```

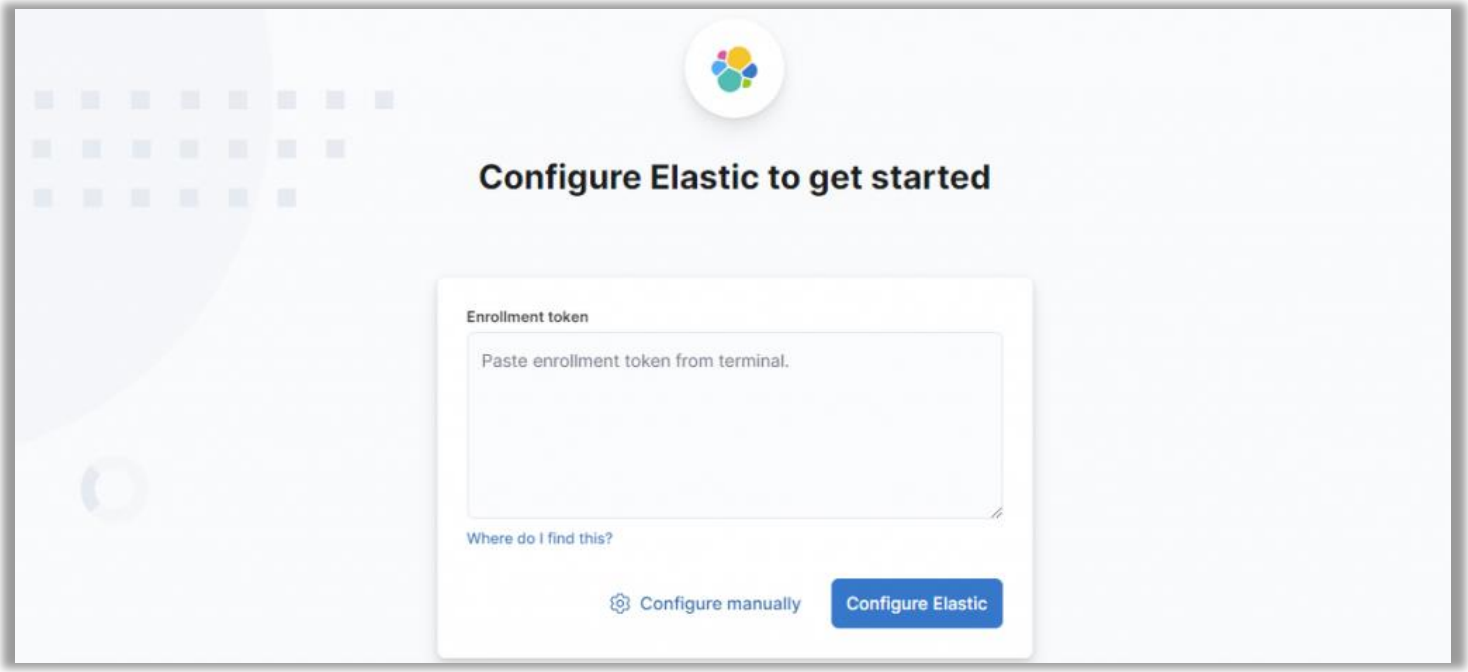
Only uncomment the server.port & server.host

```
# ================== System: Kibana Server ==================
# 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: "0.0.0.0"
```

Enabling & starting the kibana.service

```
$ sudo systemctl enable kibana.service
$ sudo systemctl start kibana.service
```

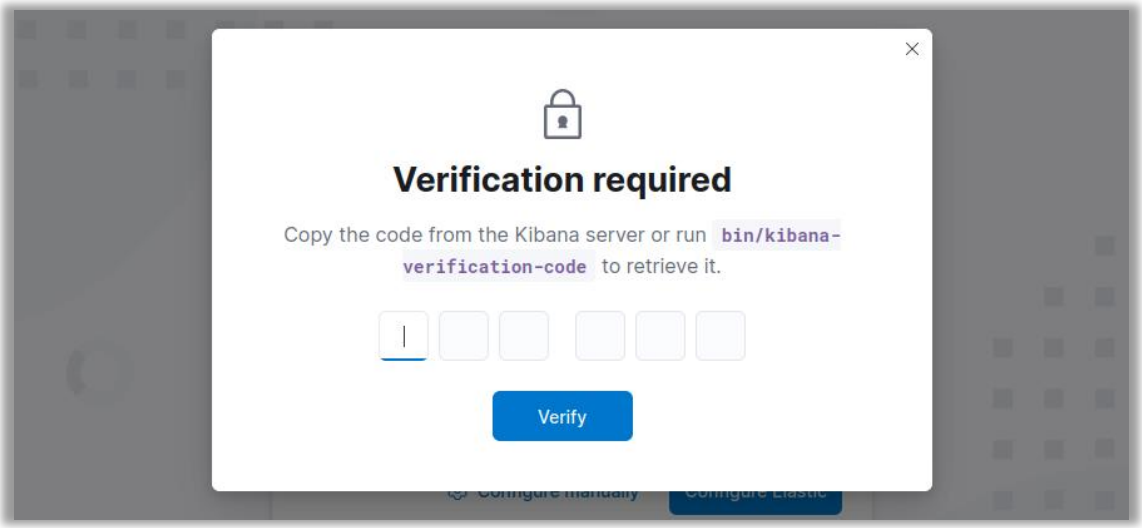
To check connectivity go to **http://127.0.0.1:5601 ,** and your will then be asked for an **enrollment token**.



PHASE 3 : Connecting elasticsearch with kibana

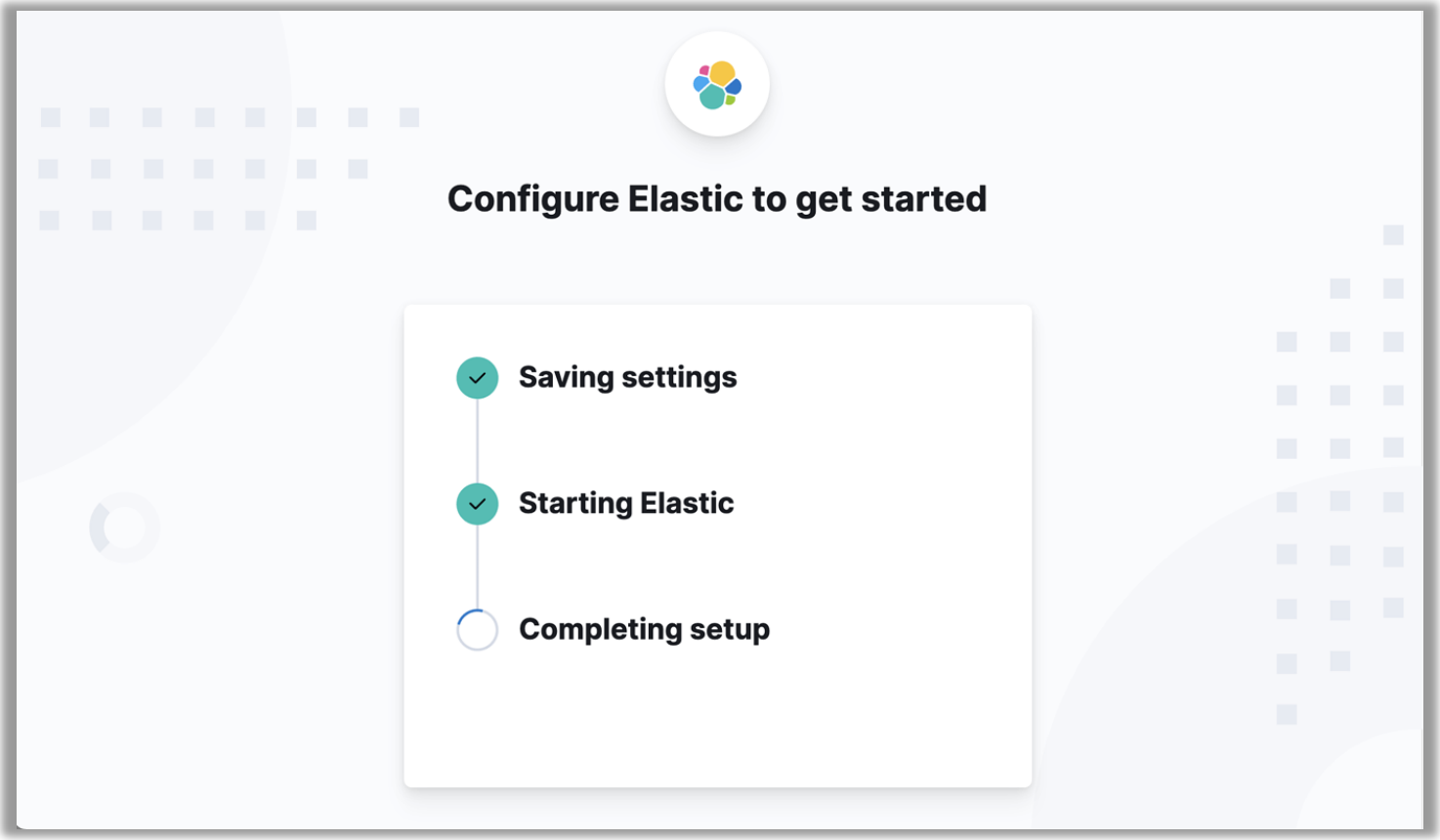
```
$ sudo /usr/share/elasticsearch/bin/elasticsearch-create-enrollment-token -s kibana
```

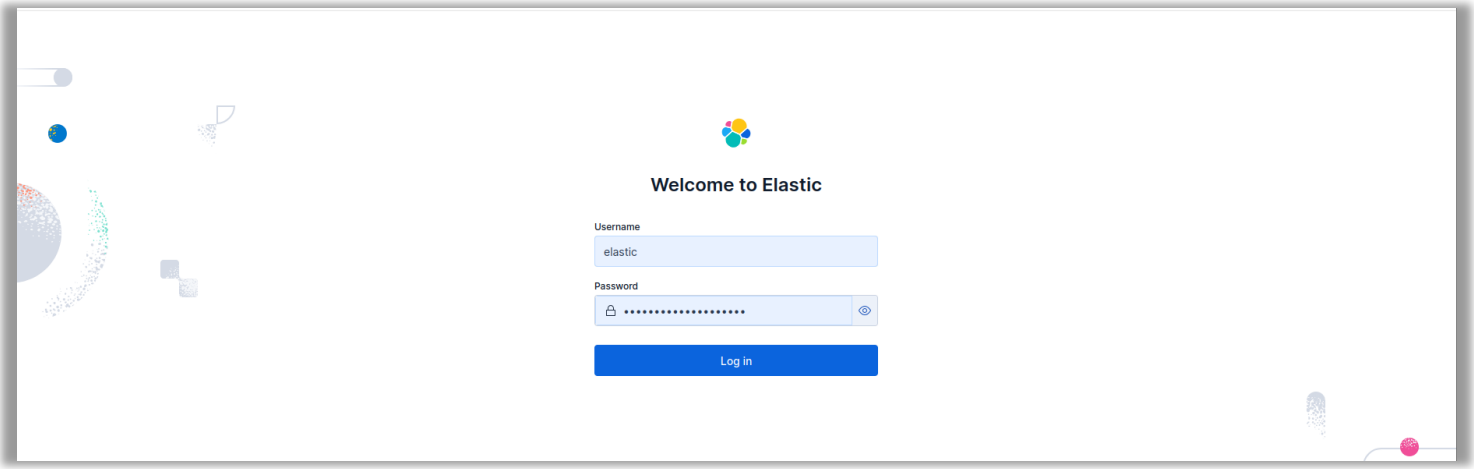
After getting your token and inserting it into Kibana, a verification code will be created

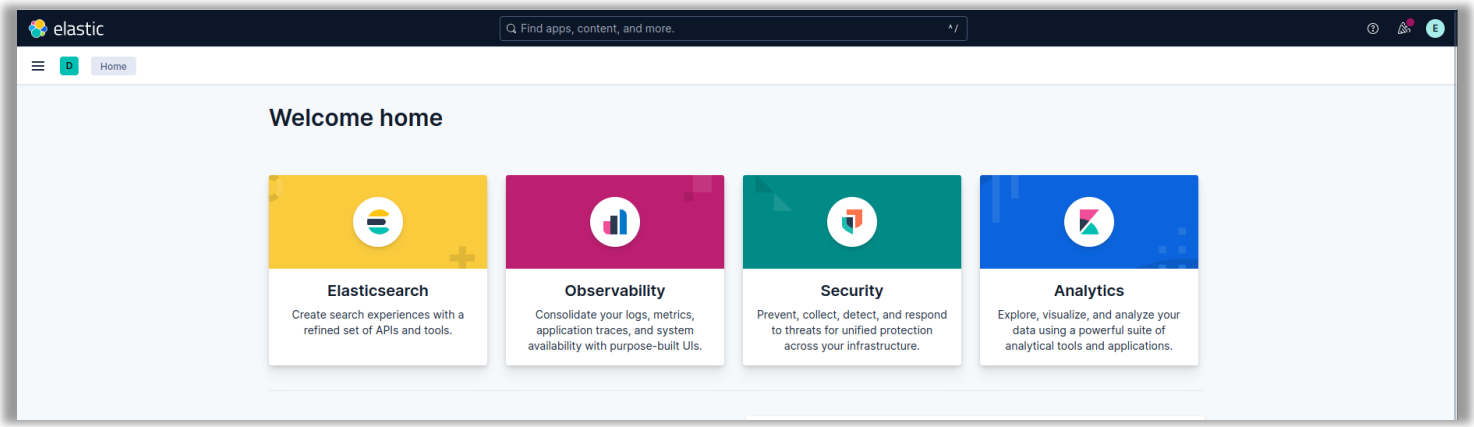


Getting verification code

```
$ sudo cd /usr/share/kibana
$ sudo ./kibana-verification-code
```





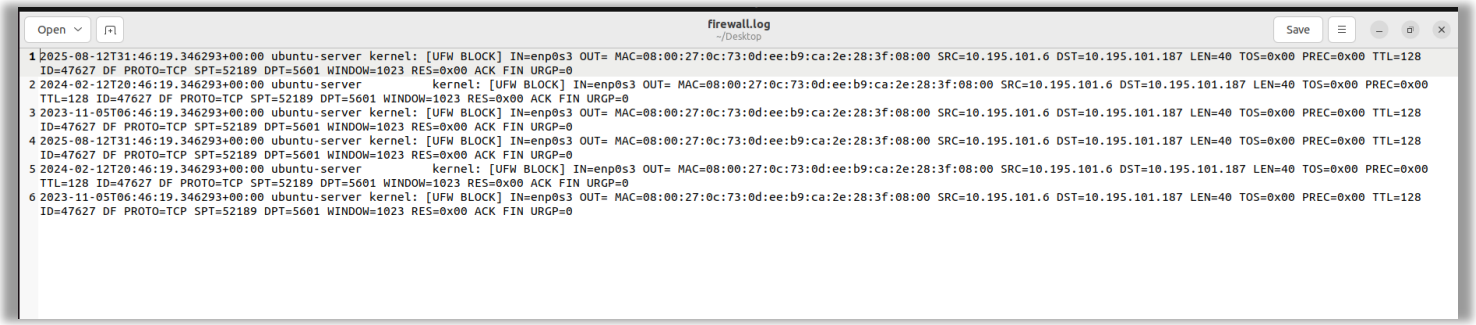


PHASE 4 : Installing & Configuring Fluentbit & sending logs

```
$ sudo apt-get update
$  sudo apt-get install fluent-bit
$  sudo systemctl enable fluent-bit
$  sudo systemctl start fluent-bit
```

For a simple simulation of logs

Create a .log file for example firewall.log and add a few logs to that file and then save it.



Configuring the fluent-bit.conf file

```
$ sudo nano /etc/fluent-bit/fluent-bit.conf
```

Allow these configurations but only change the path to the **path** where you stored the logs , change the **http_passwd** value to your current elastic password and modify the **index** as you wish.

```
[INPUT]
    name tail
    tag  ufw_logs
    path /home/omar/Desktop/firewall.log

    # Read interval (sec) Default: 1
    #Sinterval_sec 1

[OUTPUT]
    name                es
    match               *
    host                127.0.0.1
    port                9200
    index               my-ufw-logs
    http_user           elastic
    http_passwd         4XLiS0Omq6IpYMx349mW
    tls                 On
    tls.verify          Off
    trace_output        On
    suppress_type_name  On
```

MAKE SURE THE WORDS ARE ALIGNED TO AVOID ANY SYNTAX ERROR AS THIS IS A SENSITIVE FILE

Configuring the parsers.conf file

```
$ sudo nano /etc/fluent-bit/parsers.conf
```

Now we have to create a parser for our log using this syntax

Parser is different from one log to another so find your format

To create & test your own regex against your logs [click here](#).

```
  GNU nano 6.2                                        /etc/fluent-bit/parsers.conf
[PARSER]
    Name        ufw-firewall
    Format      regex
    Regex       ^(?<event_timestamp>\d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}\.\d+\+\d{2}:\d{2}).*SRC=(?<from_ip>\d+\.\d+\.\d+\.\d+)\s+DST=(?<to_ip>\d+\.\d+\
    Time_Key    time
    Time_Format %Y-%m-%dT%H:%M:%S.%L%z
    Time_Keep   On

[PARSER]
    Name   apache
    Format regex
    Regex  ^(?<host>[^ ]*) [^ ]* (?<user>[^ ]*) \[(?<time>[^\]]*)\] "(?<method>\S+)(?: +(?<path>[^\"]*?)(?: +\S*)?)?" (?<code>[^ ]*) (?<size>[^ ]*)(?: 
    Time_Key time
    Time_Format %d/%b/%Y:%H:%M:%S %z

[PARSER]
    Name   apache2
    Format regex
    Regex  ^(?<host>[^ ]*) [^ ]* (?<user>[^ ]*) \[(?<time>[^\]]*)\] "(?<method>\S+)(?: +(?<path>[^ ]*) +\S*)?" (?<code>[^ ]*) (?<size>[^ ]*)(?: "(?<ref
    Time_Key time
    Time_Format %d/%b/%Y:%H:%M:%S %z

[PARSER]
    Name   apache_error
    Format regex
    Regex  ^\[[^ ]* (?<time>[^\]]*)\] \[(?<level>[^\]]*)\](?: \[pid (?<pid>[^\]]*)\])?( \[client (?<client>[^\]]*)\])? (?<message>.*)$
                                                                          [ Read 147 lines ]
```

Check you configuration

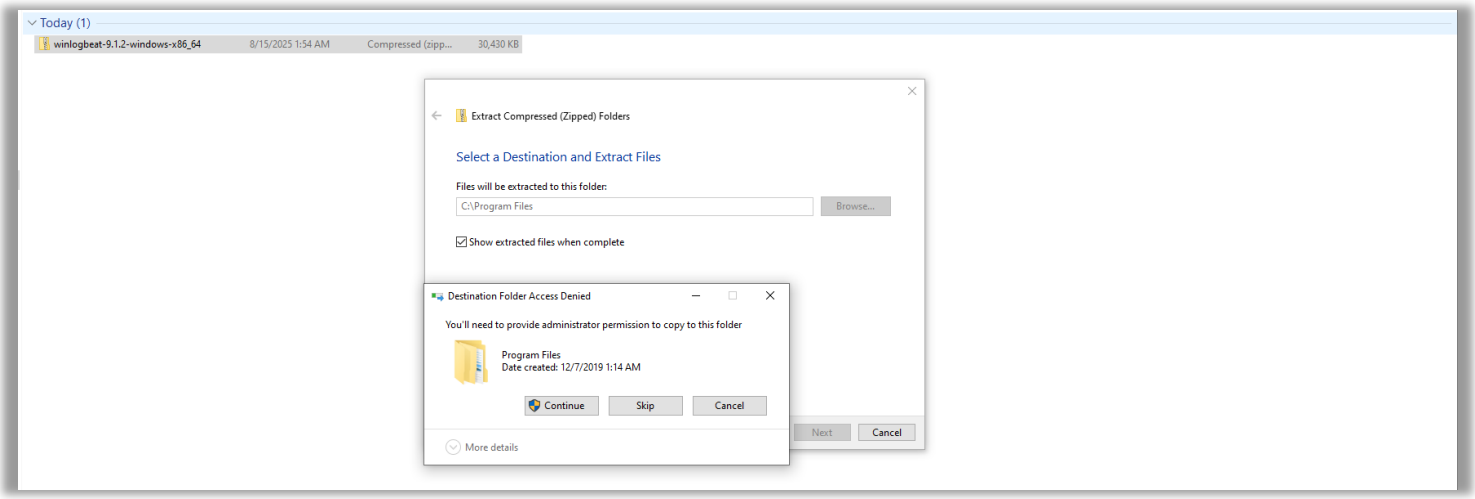
```
$ /opt/fluent-bit/bin/fluent-bit -c //etc/fluent-bit/fluent-bit.conf
```

```
omar@omar-virtual-machine:~$ /opt/fluent-bit/bin/fluent-bit -c //etc/fluent-bit/fluent-bit.conf
Fluent Bit v4.0.7
* Copyright (C) 2015-2025 The Fluent Bit Authors
* Fluent Bit is a CNCF sub-project under the umbrella of Fluentd
* https://fluentbit.io

_____ _                  _   _____ _ _                 ___  _____ |
|  ___| |                | |   | ___ (_) |               /   ||  _  |
| |__ | |_   _  ___ _ __ | |_  | |_/ /_| |_  __   __/ /| || |/' |
|  __| | | | | |/ _ \ '_ \| __| | ___ \ | __| \ \ / / /_| ||  /| |
| |   | | |_| |  __/ | | | |_  | |_/ / | |_   \ V /\___  |\ |_/ /
\_|   |_|\__,_|\___|_| |_|\__| \____/|_|\__|   \_/     |_(_)___/


[2025/08/14 23:12:04] [ info] [fluent bit] version=4.0.7, commit=, pid=7330
[2025/08/14 23:12:04] [ info] [storage] ver=1.5.3, type=memory, sync=normal, checksum=off, max_chunks_up=128
[2025/08/14 23:12:04] [ info] [simd    ] SSE2
[2025/08/14 23:12:04] [ info] [cmetrics] version=1.0.5
[2025/08/14 23:12:04] [ info] [ctraces ] version=0.6.6
[2025/08/14 23:12:04] [ info] [sp] stream processor started
[2025/08/14 23:12:04] [ info] [engine] Shutdown Grace Period=5, Shutdown Input Grace Period=2
```

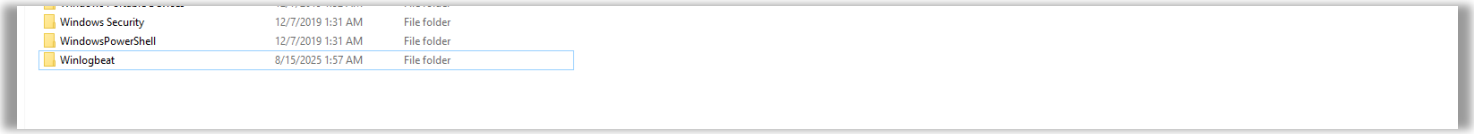
Everything works and you can't view the logs in Kibana ? try adding more logs or copy & pasting the same ones to simulate a real log update

PHASE 5 : Installing & Configuring Winlogbeat & sending logs

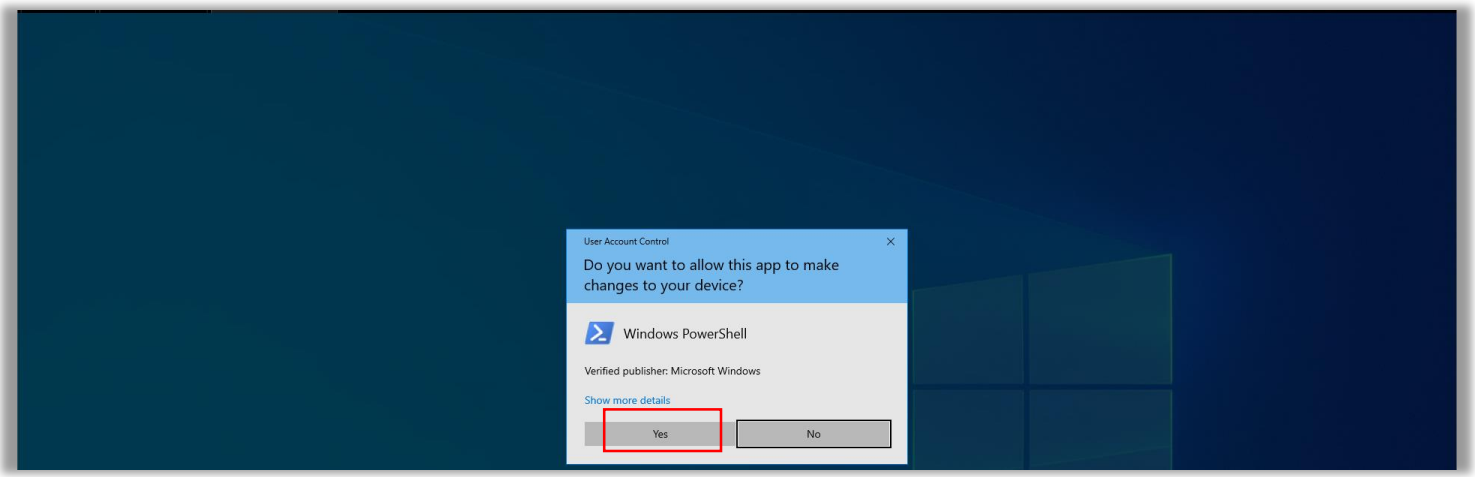
Download winlogbeat zip file from **here** & then extract to **"C:\Program files"**



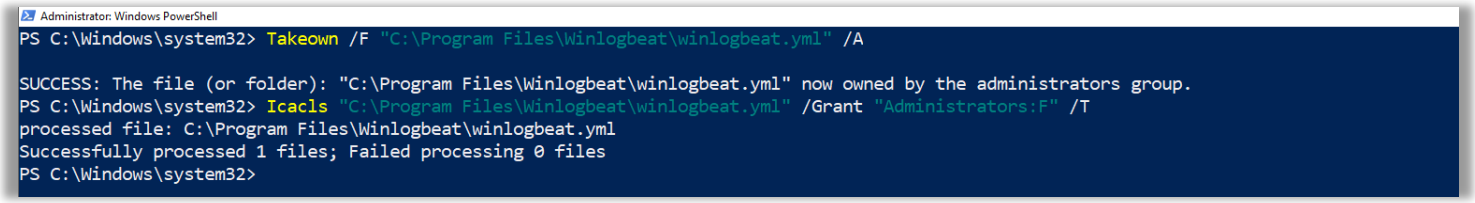
Preferably rename the folder to winlogbeat instead of winlogbeat.version



Run Powershell as Administrator and grant permissions to edit the winlogbeat.yml file

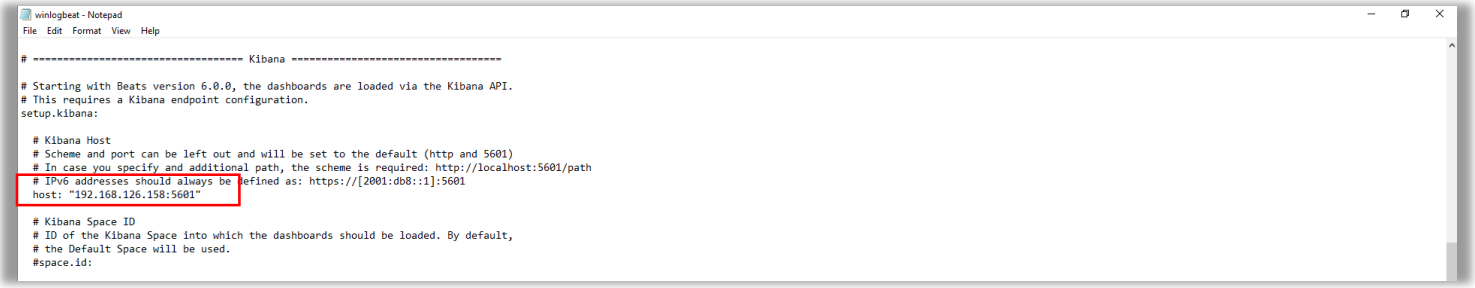



```
Takeown /F "C:\Program Files\Winlogbeat\winlogbeat.yml" /A

Icacls "C:\Program Files\Winlogbeat\winlogbeat.yml" /Grant "Administrators:F" /T
```



Now editing the winlogbeat.yml file inside the winlogbeat folder (using notepad)

Uncomment the host field under kibana and modify your IP address



Uncomment the host field under Elasticsearch Output and modify your IP address , uncomment the protocol (Only If you used https in Elasticsearch) , uncomment the username & password and adjust them , and finally add this part :

```
ssl:
    enabled: true
    certificate_authorities: ["C:/Program Files/Winlogbeat/http_ca.crt"]
```

If you don't have ssl enabled in elasticsearch then change " enabled : true " -> "enabled : false "

Adding the http_ca.crt to allow connection with elasticsearch (SKIP THIS PART IF SSL IS DISABLED)

To get the certificate go to Ubuntu , go to **`/etc/elasticseach/certs`** and there you will find http_ca.crt , you can copy the content inside that file then paste it into a txt file on your windows machine and change the extension to .crt to transfer it using a USB , after that add it to **`C:/Program Files/Winlogbeat`**

MAKE SURE ELASTICSEARCH & KIBANA ARE RUNNING

.\winlogbeat.exe test config -c .\winlogbeat.yml -e

```
PS C:\Windows\system32> cd "C:\Program Files\Winlogbeat"
PS C:\Program Files\Winlogbeat>  .\winlogbeat.exe test config -c .\winlogbeat.yml -e
{"log.level":"info","@timestamp":"2025-08-15T07:34:20.814-0700","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/cmd/instance.(*Beat).configure","file.name":"instance/beat.go","file.line":82
: [C:\\Program Files\\Winlogbeat] Data path: [C:\\Program Files\\Winlogbeat\\data] Logs path: [C:\\Program Files\\Winlogbeat\\logs]","service.name":"winlogbeat","ecs.version":"1.6.0"}
{"log.level":"info","@timestamp":"2025-08-15T07:34:20.857-0700","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/cmd/instance.(*Beat).configure","file.name":"instance/beat.go","file.line":83
e.name":"winlogbeat","ecs.version":"1.6.0"}
{"log.level":"info","@timestamp":"2025-08-15T07:34:20.887-0700","log.logger":"beat","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/cmd/instance.(*Beat).createBeater","file.name":"instance/
: 9.1.2 (FIPS-distribution: false)","service.name":"winlogbeat","ecs.version":"1.6.0"}
{"log.level":"info","@timestamp":"2025-08-15T07:34:20.889-0700","log.logger":"beat","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/cmd/instance.(*Beat).logSystemInfo","file.name":"instance
nlogbeat","system_info":{"beat":{"path":{"config":"C:\\Program Files\\Winlogbeat","data":"C:\\Program Files\\Winlogbeat\\data","home":"C:\\Program Files\\Winlogbeat","logs":"C:\\Program Files\\Winlogbeat
9ce74"},"ecs.version":"1.6.0"}}
{"log.level":"info","@timestamp":"2025-08-15T07:34:20.889-0700","log.logger":"beat","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/cmd/instance.(*Beat).logSystemInfo","file.name":"instance
inlogbeat","system_info":{"build":{"commit":"b036c1c565cf24c9b720605632234d20cb9dba60","libbeat":"9.1.2","time":"2025-08-11T13:56:43.000Z","version":"9.1.2"},"ecs.version":"1.6.0"}}
{"log.level":"info","@timestamp":"2025-08-15T07:34:20.889-0700","log.logger":"beat","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/cmd/instance.(*Beat).logSystemInfo","file.name":"instance
e":"winlogbeat","system_info":{"go":{"os":"windows","arch":"amd64","max_procs":1,"version":"go1.24.4"},"ecs.version":"1.6.0"}}
{"log.level":"info","@timestamp":"2025-08-15T07:34:20.902-0700","log.logger":"beat","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/cmd/instance.(*Beat).logSystemInfo","file.name":"instance
nlogbeat","system_info":{"host":{"architecture":"x86_64","native_architecture":"x86_64","boot_time":"2025-08-15T06:43:52-07:00","name":"DESKTOP-A2JN342","ip":["192.168.126.143","fe80::74c0:17c9:2a36:8dbc
4170 (WinBuild.160101.0800)","mac":["00:0c:29:f4:a5:29","0c:9a:3c:20:1a:38"],"os":{"type":"windows","family":"windows","platform":"windows","name":"Windows 10 Pro","version":"10.0","major":10,"minor":0,"
":-25200,"id":"09f0b707-fbe4-4c70-b4b6-f03c5f44afbe"},"ecs.version":"1.6.0"}}
{"log.level":"info","@timestamp":"2025-08-15T07:34:20.906-0700","log.logger":"beat","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/cmd/instance.(*Beat).logSystemInfo","file.name":"instance
"winlogbeat","system_info":{"process":{"cwd":"C:\\Program Files\\Winlogbeat","exe":"C:\\Program Files\\Winlogbeat\\winlogbeat.exe","name":"winlogbeat.exe","pid":2108,"ppid":5056,"start_time":"2025-08-15T
{"log.level":"info","@timestamp":"2025-08-15T07:34:20.990-0700","log.logger":"elasticsearch.esclientleg","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/esleg/eslegclient.NewConnection","fi
sticsearch url: https://192.168.126.10:9200","service.name":"winlogbeat","ecs.version":"1.6.0"}
{"log.level":"info","@timestamp":"2025-08-15T07:34:20.990-0700","log.logger":"publisher","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/publisher/pipeline.LoadWithSettings","file.name":"pi
N342","service.name":"winlogbeat","ecs.version":"1.6.0"}
{"log.level":"info","@timestamp":"2025-08-15T07:34:20.993-0700","log.logger":"winlogbeat","log.origin":{"function":"github.com/elastic/beats/v7/winlogbeat/beater.New","file.name":"beater/winlogbeat.go","
:\\Program Files\\Winlogbeat\\data\\.winlogbeat.yml","service.name":"winlogbeat","ecs.version":"1.6.0"}
Config OK
PS C:\Program Files\Winlogbeat>
```

.\winlogbeat.exe setup -e

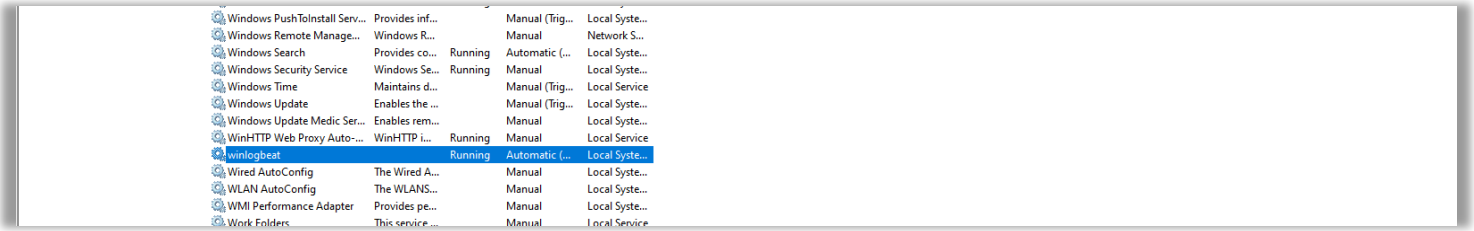
```
Select Administrator: Windows PowerShell
.","service.name":"winlogbeat","ecs.version":"1.6.0"}
{"log.level":"info","@timestamp":"2025-08-16T00:04:03.142-0700","log.logger":"template","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/template.(*templateBui
lder).buildBody","file.name":"template/load.go","file.line":262},"message":"Existing template will be overwritten, as overwrite is enabled.","service.name":"winlogbeat","ec
s.version":"1.6.0"}
{"log.level":"info","@timestamp":"2025-08-16T00:04:03.903-0700","log.logger":"template_loader","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/template.(*ESLo
ader).loadTemplate","file.name":"template/load.go","file.line":176},"message":"Try loading template winlogbeat-9.1.2 to Elasticsearch","service.name":"winlogbeat","ecs.vers
ion":"1.6.0"}
{"log.level":"info","@timestamp":"2025-08-16T00:04:04.078-0700","log.logger":"template_loader","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/template.(*ESLo
ader).Load","file.name":"template/load.go","file.line":133},"message":"Template with name \"winlogbeat-9.1.2\" loaded.","service.name":"winlogbeat","ecs.version":"1.6.0"}
{"log.level":"info","@timestamp":"2025-08-16T00:04:04.092-0700","log.logger":"template_loader","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/template.(*ESLo
ader).Load","file.name":"template/load.go","file.line":149},"message":"Data stream with name \"winlogbeat-9.1.2\" already exists.","service.name":"winlogbeat","ecs.version"
:"1.6.0"}
{"log.level":"info","@timestamp":"2025-08-16T00:04:04.108-0700","log.logger":"index-management","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/idxmgmt.(*inde
xManager).Setup","file.name":"idxmgmt/index_support.go","file.line":299},"message":"Loaded index template.","service.name":"winlogbeat","ecs.version":"1.6.0"}
Index setup finished.
Loading dashboards (Kibana must be running and reachable)
{"log.level":"info","@timestamp":"2025-08-16T00:04:04.149-0700","log.logger":"kibana","log.origin":{"function":"github.com/elastic/elastic-agent-libs/kibana.NewClientWithCo
nfigDefault","file.name":"kibana/client.go","file.line":181},"message":"Kibana url: http://192.168.126.158:5601","service.name":"winlogbeat","ecs.version":"1.6.0"}
{"log.level":"info","@timestamp":"2025-08-16T00:04:05.122-0700","log.logger":"kibana","log.origin":{"function":"github.com/elastic/elastic-agent-libs/kibana.NewClientWithCo
nfigDefault","file.name":"kibana/client.go","file.line":181},"message":"Kibana url: http://192.168.126.158:5601","service.name":"winlogbeat","ecs.version":"1.6.0"}
{"log.level":"info","@timestamp":"2025-08-16T00:04:04.925-0700","log.logger":"processors.add_cloud_metadata","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/p
rocessors/add_cloud_metadata.(*addCloudMetadata).init.func1","file.name":"add_cloud_metadata/add_cloud_metadata.go","file.line":100},"message":"add_cloud_metadata: hosting
provider type not detected.","service.name":"winlogbeat","ecs.version":"1.6.0"}
{"log.level":"info","@timestamp":"2025-08-16T00:04:40.318-0700","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/cmd/instance.(*Beat).loadDashboards","file.nam
e":"instance/beat.go","file.line":1070},"message":"Kibana dashboards successfully loaded.","service.name":"winlogbeat","ecs.version":"1.6.0"}
Loaded dashboards
{"log.level":"info","@timestamp":"2025-08-16T00:04:40.331-0700","log.logger":"esclientleg","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/esleg/eslegclient.N
ewConnection","file.name":"eslegclient/connection.go","file.line":132},"message":"elasticsearch url: https://192.168.126.158:9200","service.name":"winlogbeat","ecs.version"
:"1.6.0"}
{"log.level":"info","@timestamp":"2025-08-16T00:04:40.390-0700","log.logger":"esclientleg","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/esleg/eslegclient.(
*Connection).Ping","file.name":"eslegclient/connection.go","file.line":324},"message":"Attempting to connect to Elasticsearch version 9.1.2 (default)","service.name":"winlo
gbeat","ecs.version":"1.6.0"}
Loaded Ingest pipelines
PS C:\Program files\winlogbeat>
```

winlogbeat test output

```
Administrator: Windows PowerShell
PS C:\Program files\winlogbeat> .\winlogbeat.exe test output
elasticsearch: https://192.168.126.158:9200...
  parse url... OK
  connection...
    parse host... OK
    dns lookup... OK
    addresses: 192.168.126.158
    dial up... OK
  TLS...
    security: server's certificate chain verification is enabled
    handshake... OK
    TLS version: TLSv1.3
    dial up... OK
  talk to server... OK
  version: 9.1.2
PS C:\Program files\winlogbeat>
```

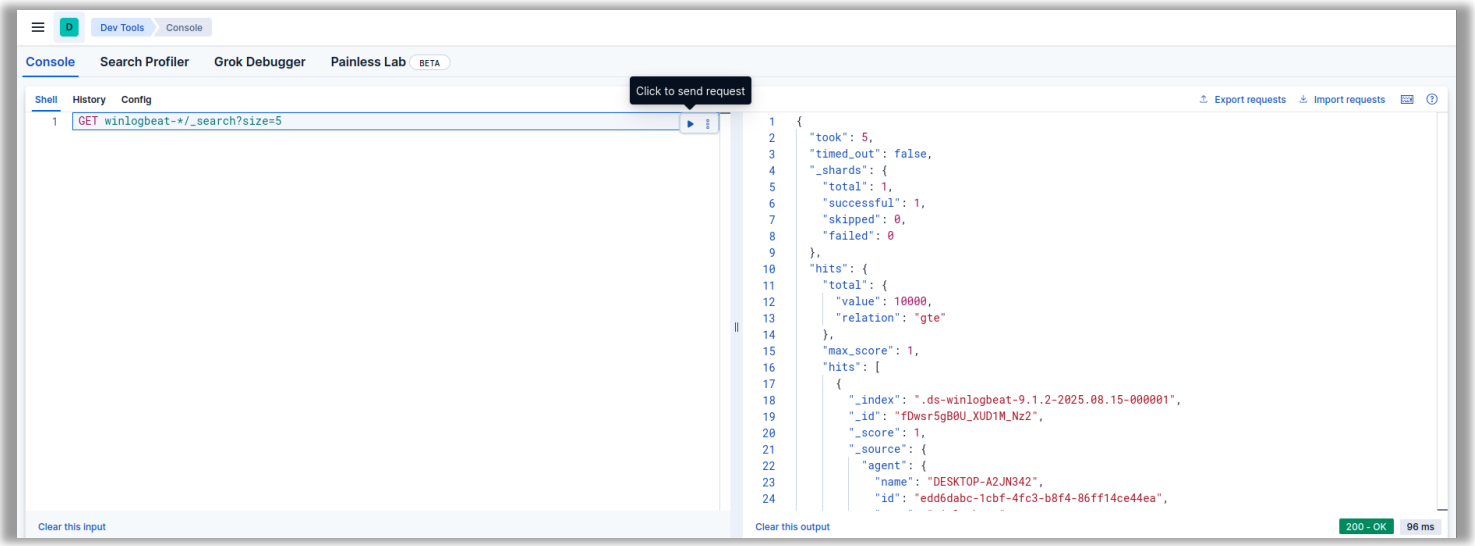
`Start-Service winlogbeat`

And then you can check if it's running using sevices.msc or **`Get-Service winlogbeat`**

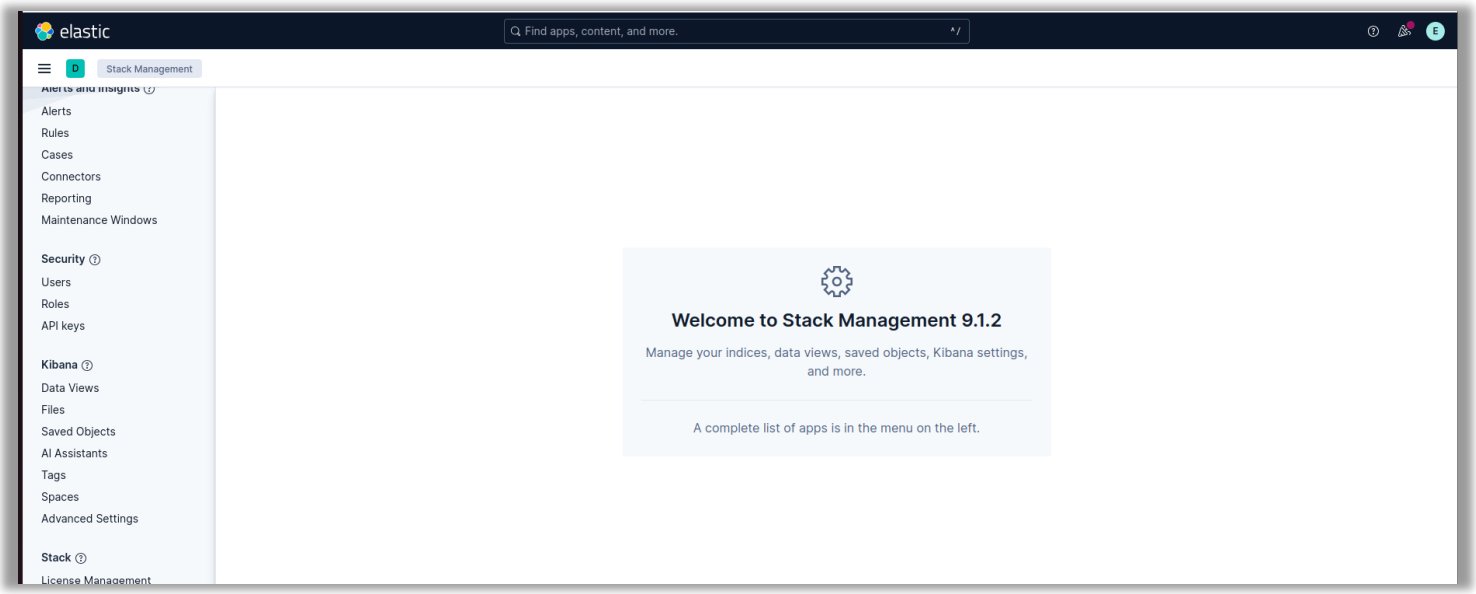


Viewing the logs using kibana dashboard

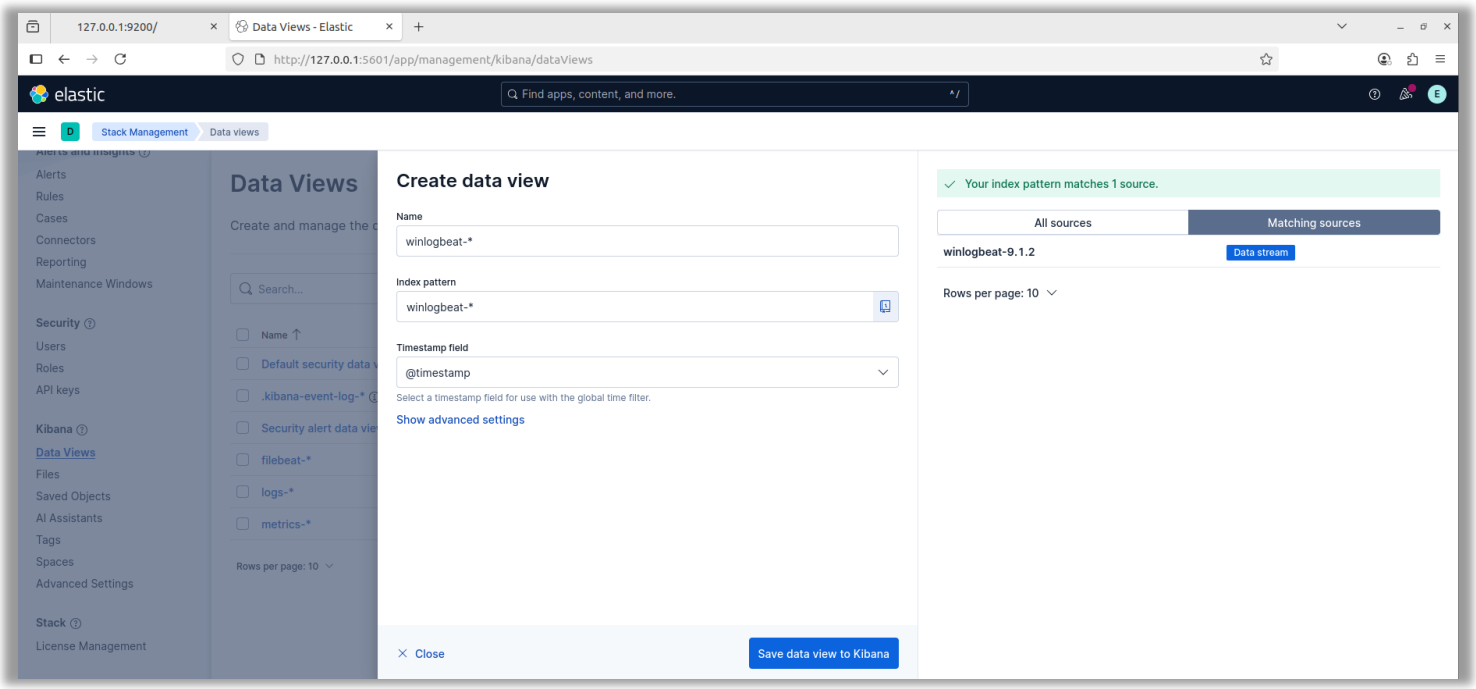
Testing first using Dev tools : **`GET winlogbeat-*/_search?size=5`**



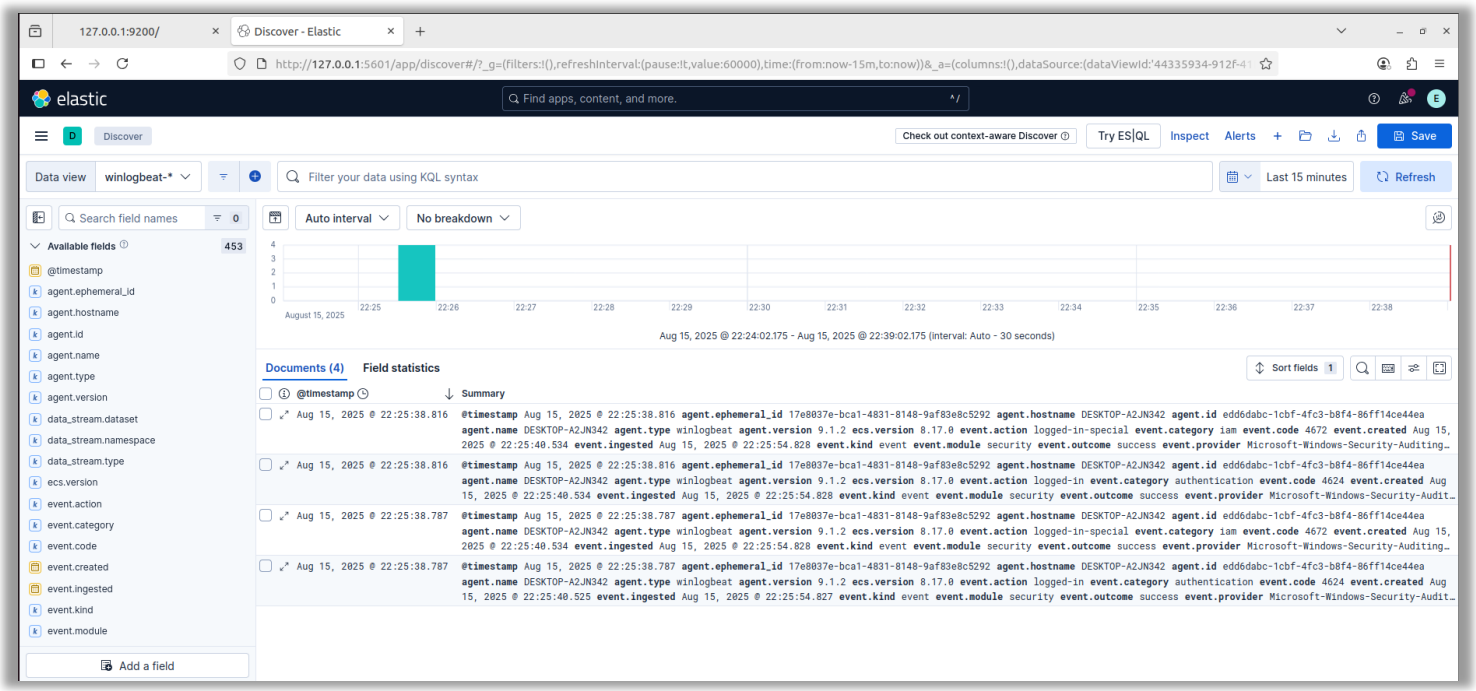
Stack Management > Data View



Index Pattern : winlogbeat-*
Select @timestamp as the time filter



Confirm logs and select the **winlogbeat-*** index

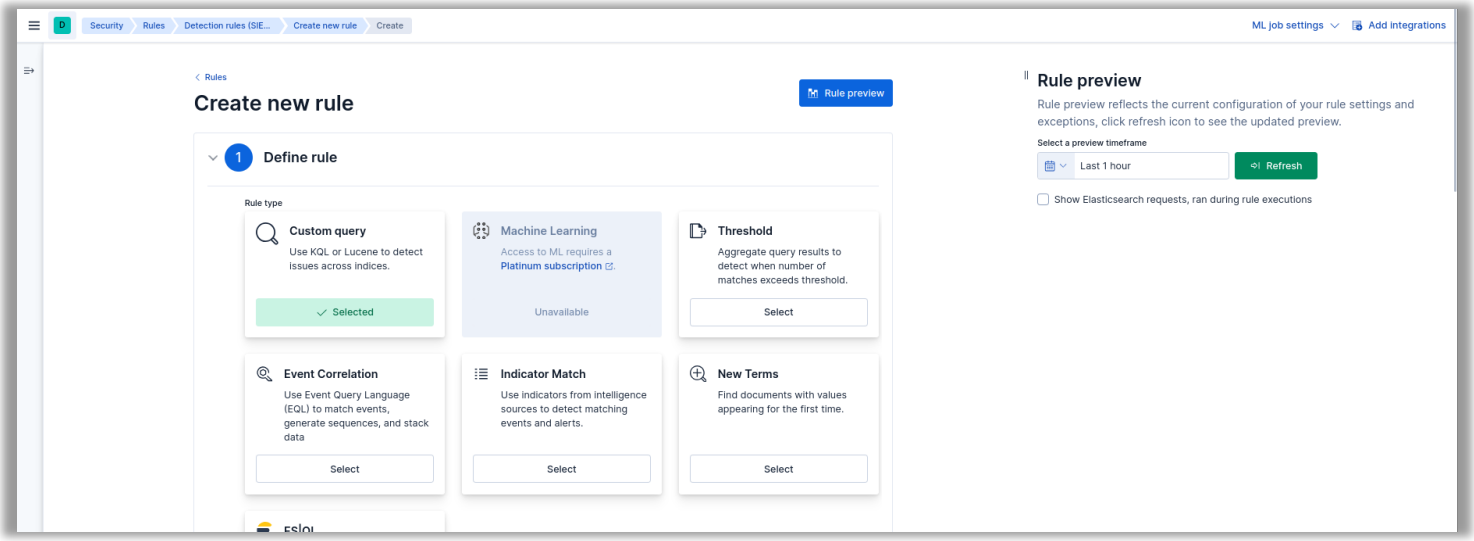


PHASE 6 : Threat detection rules & simulating suspicious activity

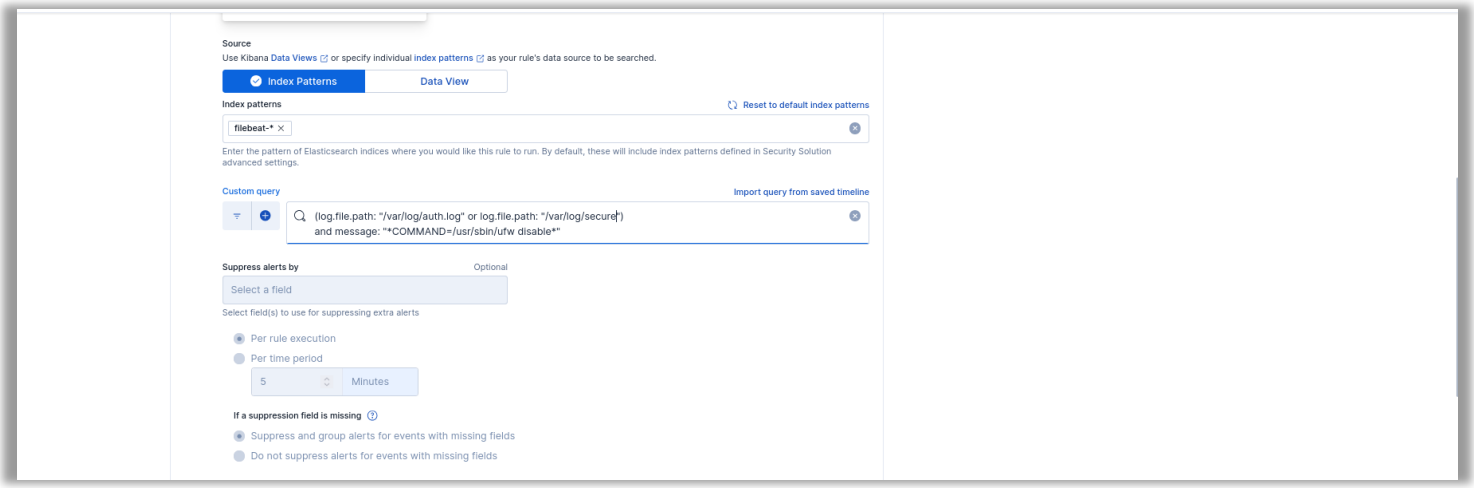
To demonstrate how security monitoring works, we'll simulate a basic but suspicious activity: an attacker disabling the Linux firewall (UFW).

In real-world scenarios, detection rules are far more sophisticated , incorporating behavioral analysis, anomaly detection, and threat intelligence . But this simplified example helps illustrate the core principles.

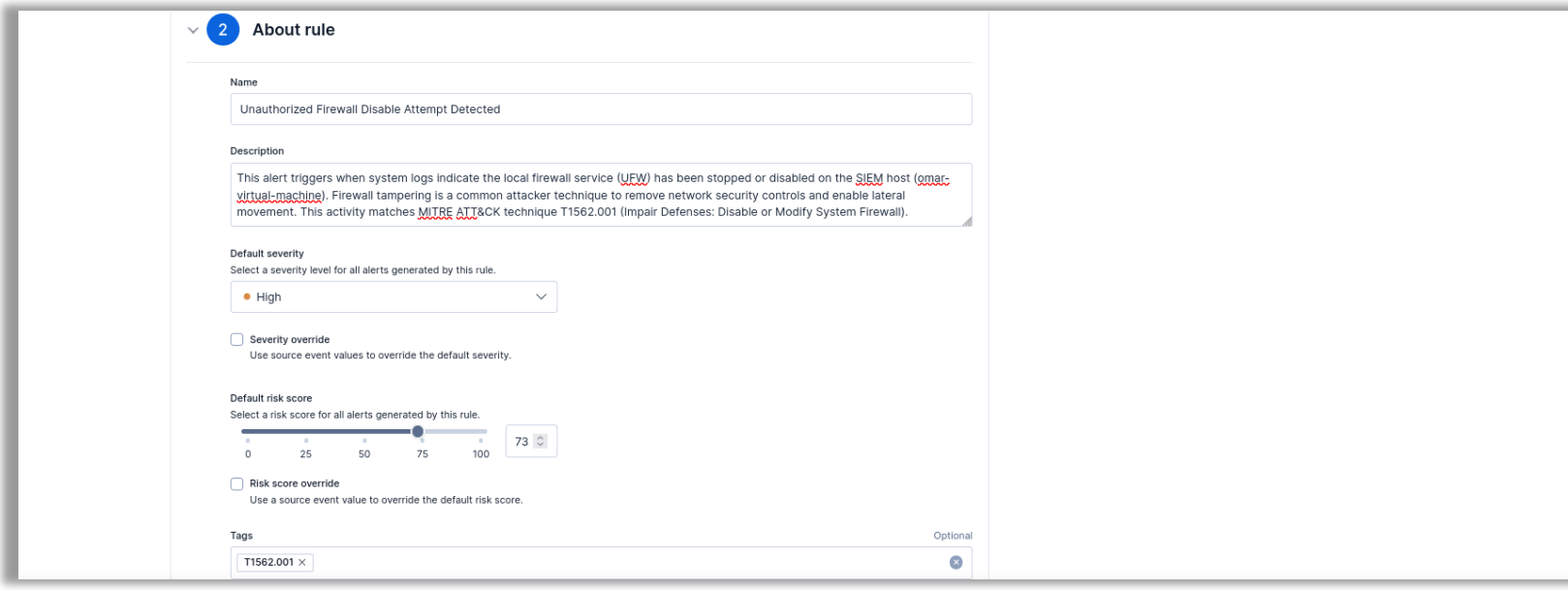
Security>rules>detection rules (SIEM) > create new rule



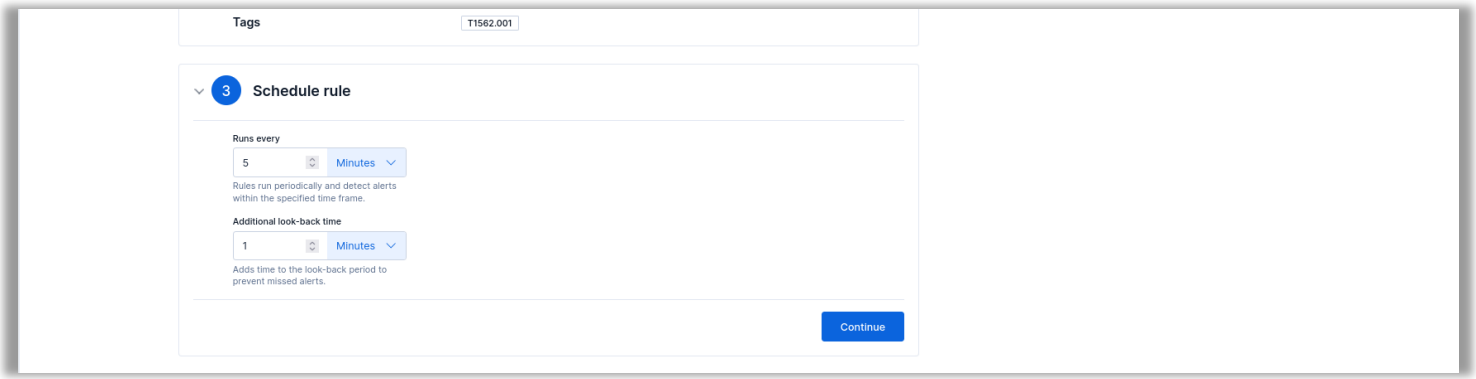
Choosing the wanted index pattern and writing our custom query (KQL) to detect the disabling of ufw firewall , make sure to test your query in discover first.



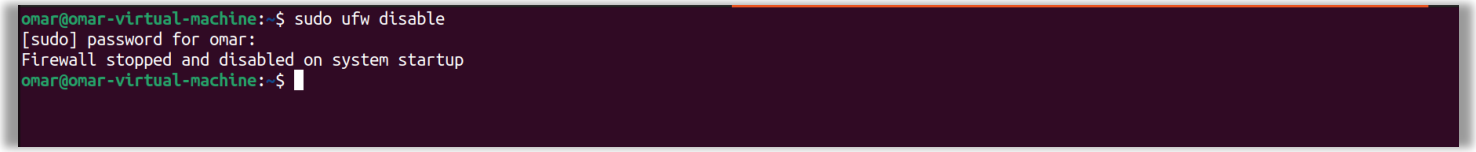
Specifying a name , description , selecting the severity and risk score



Leave on Default



Testing using the command : `sudo ufw disable`



View alerts through security>alerts