



TAREA 07

SIEM ELK

INCIDENTES DE CIBERSEGURIDAD

ALBA MOREJÓN GARCÍA

2024/2025

Ciberseguridad en Entornos de las Tecnologías de la Información

La Monitorización Multipunto en el SIEM

En la Unidad 7 hemos estudiado cómo instalar y configurar un SIEM ELK completo, situándolo en la misma red que los diferentes agentes IDS que detectarán las intrusiones y enviarán la información de registros de SNORT a través de Filebeat.

La estructura creada en la Tarea 6, es la presente en cualquier entorno productivo en la que suele haber una sonda Snort en cada una de las máquinas perimetrales, comprometidas, vulnerables, etc., cuya información de logging se ha de redirigir hacia una única máquina en la que estará instalado el SIEM (Elastic Stack).

Pues bien, continuando con el trabajo iniciado en la Tarea 6 asociada a la Unidad 6, en esta tarea abordaremos la lectura y tratamiento del log que centraliza la información de todas las sondas Snort, filtrando su contenido, almacenándolo en la base de datos y preparando un conjunto de visualizaciones que recogeremos en un Tablero de Monitorización Multipunto.

Apartado 1: Instalación y configuración de Elasticsearch.

a) Detallar la configuración a efectuar en Elasticsearch.

b) Prueba de funcionamiento de Elasticsearch.

Descargamos e instalamos Elasticsearch con los siguientes comandos:

```
wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-7.10.2-amd64.deb
```

```
sudo dpkg -i elasticsearch-7.10.2-amd64.deb
```

```
SOC-IC06 [Corriendo] - Oracle VirtualBox
alba.morejon@SOCIC06: ~
alba.morejon@SOCIC06: $ wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-7.10.2-amd64.deb
--2025-03-26 19:00:06-- https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-7.10.2-amd64.deb
Resolving artifacts.elastic.co (artifacts.elastic.co)... 34.120.127.130
Connecting to artifacts.elastic.co (artifacts.elastic.co)|34.120.127.130|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 318852724 (304M) [application/octet-stream]
Saving to: 'elasticsearch-7.10.2-amd64.deb'

elasticsearch-7.10. 100%[=====>] 304.08M  6.49MB/s    in 52s

2025-03-26 19:00:59 (5.81 MB/s) - 'elasticsearch-7.10.2-amd64.deb' saved [318852724/318852724]

alba.morejon@SOCIC06: $ sudo dpkg -i elasticsearch-7.10.2-amd64.deb
[sudo] password for alba.morejon:
Selecting previously unselected package elasticsearch.
(Reading database ... 149197 files and directories currently installed.)
Preparing to unpack elasticsearch-7.10.2-amd64.deb ...
Creating elasticsearch group... OK
Creating elasticsearch user... OK
Unpacking elasticsearch (7.10.2) ...
Setting up elasticsearch (7.10.2) ...
Created elasticsearch keystore in /etc/elasticsearch/elasticsearch.keystore
```

Iniciamos el servicio y comprobamos su estado.

Iniciamos el servicio y comprobamos su estado

Enlaces ▾

SOC-IC06 [Corriendo] - Oracle VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda Mar 26 19:05 es

alba.morejon@SOCIC06 ~

```
alba.morejon@SOCIC06: $ sudo systemctl start elasticsearch
alba.morejon@SOCIC06: $ sudo systemctl enable elasticsearch
Synchronizing state of elasticsearch.service with SysV service script with /usr/lib/
systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable elasticsearch
Created symlink /etc/systemd/system/multi-user.target.wants/elasticsearch.service →
/usr/lib/systemd/system/elasticsearch.service.
alba.morejon@SOCIC06: $ sudo systemctl status elasticsearch
● elasticsearch.service - Elasticsearch
   Loaded: loaded (/usr/lib/systemd/system/elasticsearch.service; enabled; preset:
Active: active (running) since Wed 2025-03-26 19:04:55 UTC; 48s ago
     Docs: https://www.elastic.co
Main PID: 4700 (java)
   Tasks: 61 (limit: 2873)
  Memory: 1.2G (peak: 1.2G)
    CPU: 26.727s
   CGroup: /system.slice/elasticsearch.service
           └─4700 /usr/share/elasticsearch/jdk/bin/java -Xshare:auto -Des.network=
               ├─4876 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux-x86_64

Mar 26 19:04:14 SOCIC06 systemd[1]: Starting elasticsearch.service - Elasticsearch...
Mar 26 19:04:55 SOCIC06 systemd[1]: Started elasticsearch.service - Elasticsearch.
```

Añadimos la siguiente configuración en el fichero `elasticsearch.yml` para que pueda iniciar y funcionar correctamente: asignamos un nombre al cluster y al nodo, le indicamos las interfaces de red disponibles, le indicamos la dirección y el nodo principal desde el que se actuará.

Enlaces ▾



```
GNU nano 7.2          /etc/elasticsearch/elasticsearch.yml
#
# ----- Cluster -----
#
# Use a descriptive name for your cluster:
#
cluster.name: siem-cluster
node.name: node-1
network.host: 0.0.0.0
discovery.seed_hosts: ["127.0.0.1"]
cluster.initial_master_nodes: ["node-1"]
```

Comprobamos que esté bien configurado:

Se realiza una solicitud al servidor para obtener información básica sobre el nodo (nombre del nodo y el cluster, versión elasticsearch y otras configuraciones)

```
alba.morejon@SOCIC06:~$ curl -X GET "localhost:9200/"
{
  "name" : "node-1",
  "cluster_name" : "siem-cluster",
  "cluster_uuid" : "noHte0JLQzGGuittEcR6dw",
  "version" : {
    "number" : "7.10.2",
    "build_flavor" : "default",
    "build_type" : "deb",
    "build_hash" : "747e1cc71def077253878a59143c1f785afa92b9",
    "build_date" : "2021-01-13T00:42:12.435326Z",
    "build_snapshot" : false,
    "lucene_version" : "8.7.0",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
```

Se realiza una solicitud para tener información sobre el estado del cluster, el número de nodos, número de shards...

```
alba.morejon@SOCIC06:~$ curl -X GET "localhost:9200/_cluster/health?pretty"
{
  "cluster_name" : "siem-cluster",
  "status" : "green",
  "timed_out" : false,
  "number_of_nodes" : 1,
  "number_of_data_nodes" : 1,
  "active_primary_shards" : 0,
  "active_shards" : 0,
  "relocating_shards" : 0,
  "initializing_shards" : 0,
  "unassigned_shards" : 0,
  "delayed_unassigned_shards" : 0,
  "number_of_pending_tasks" : 0,
  "number_of_in_flight_fetch" : 0,
  "task_max_waiting_in_queue_millis" : 0,
  "active_shards_percent_as_number" : 100.0
}
```

Creamos un índice (test.index) y vemos que se haya configurado correctamente. Listamos todos los índices existentes dentro del clúster, mostrando el estado, el nombre el UUID...

The screenshot shows a terminal window titled "SOC-IC06 [Corriendo] - Oracle VirtualBox". The terminal prompt is "alba.morejon@SOCIC06:~". The user runs two curl commands: "curl -X PUT "localhost:9200/test-index?pretty"" which creates the index, and "curl -X GET "localhost:9200/_cat/indices?v"" which lists the indices. The output shows the "test-index" has been created with 1 shard and 1 primary.

```
alba.morejon@SOCIC06:~$ curl -X PUT "localhost:9200/test-index?pretty"
{
  "acknowledged" : true,
  "shards_acknowledged" : true,
  "index" : "test-index"
}
alba.morejon@SOCIC06:~$ curl -X GET "localhost:9200/_cat/indices?v"
health status index      uuid                                     pri  rep docs.count docs.deleted store
.size pri.store.size
yellow open   test-index hAf_LQuARjaRTpnlxr0Dew    1   1          0          0
 208b     208b
alba.morejon@SOCIC06:~
```

Pruebas:

Creamos un documento en el índice creado anteriormente, test-index (con los datos del inicio del código)

The screenshot shows a terminal window titled "SOC-IC06 [Corriendo] - Oracle VirtualBox". The user runs "curl -X POST "localhost:9200/test-index/_doc/1?pretty" -H 'Content-Type: application/json' -d'" with a JSON payload. The response shows the document was created successfully with ID "1", version 1, and a primary term of 1.

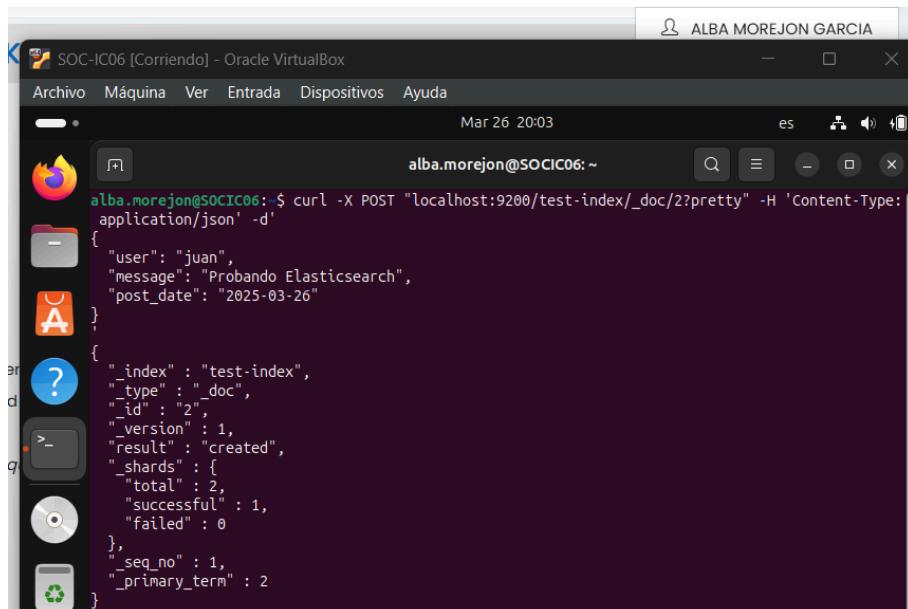
```
alba.morejon@SOCIC06:~$ curl -X POST "localhost:9200/test-index/_doc/1?pretty" -H 'Content-Type: application/json' -d'
{
  "user": "alba",
  "message": "Elasticsearch está funcionando correctamente",
  "post_date": "2025-03-26"
}
{
  "_index": "test-index",
  "_type": "_doc",
  "_id": "1",
  "_version": 1,
  "result": "created",
  "_shards": {
    "total": 2,
    "successful": 1,
    "failed": 0
  },
  "_seq_no": 0,
  "_primary_term": 1
}
```

Realizamos una búsqueda en el índice buscando el documento anterior

The screenshot shows a terminal window titled "SOC-IC06 [Corriendo] - Oracle VirtualBox". The user runs "curl -X GET "localhost:9200/test-index/_search?q=user:alba&pretty"" and receives a search response with one hit. The hit is the previously created document with ID "1".

```
alba.morejon@SOCIC06:~$ curl -X GET "localhost:9200/test-index/_search?q=user:alba&pretty"
{
  "took": 76,
  "timed_out": false,
  "_shards": {
    "total": 1,
    "successful": 1,
    "skipped": 0,
    "failed": 0
  },
  "hits": [
    {
      "total": {
        "value": 1,
        "relation": "eq"
      },
      "max_score": 0.2876821,
      "hits": [
        {
          "_index": "test-index",
          "_type": "_doc",
          "_id": "1",
          "_score": 0.2876821,
          "_source": [
            "user": "alba",
            "message": "Elasticsearch está funcionando correctamente",
            "post_date": "2025-03-26"
          ]
        }
      ]
    }
}
```

Agregamos otro documento en el mismo índice:

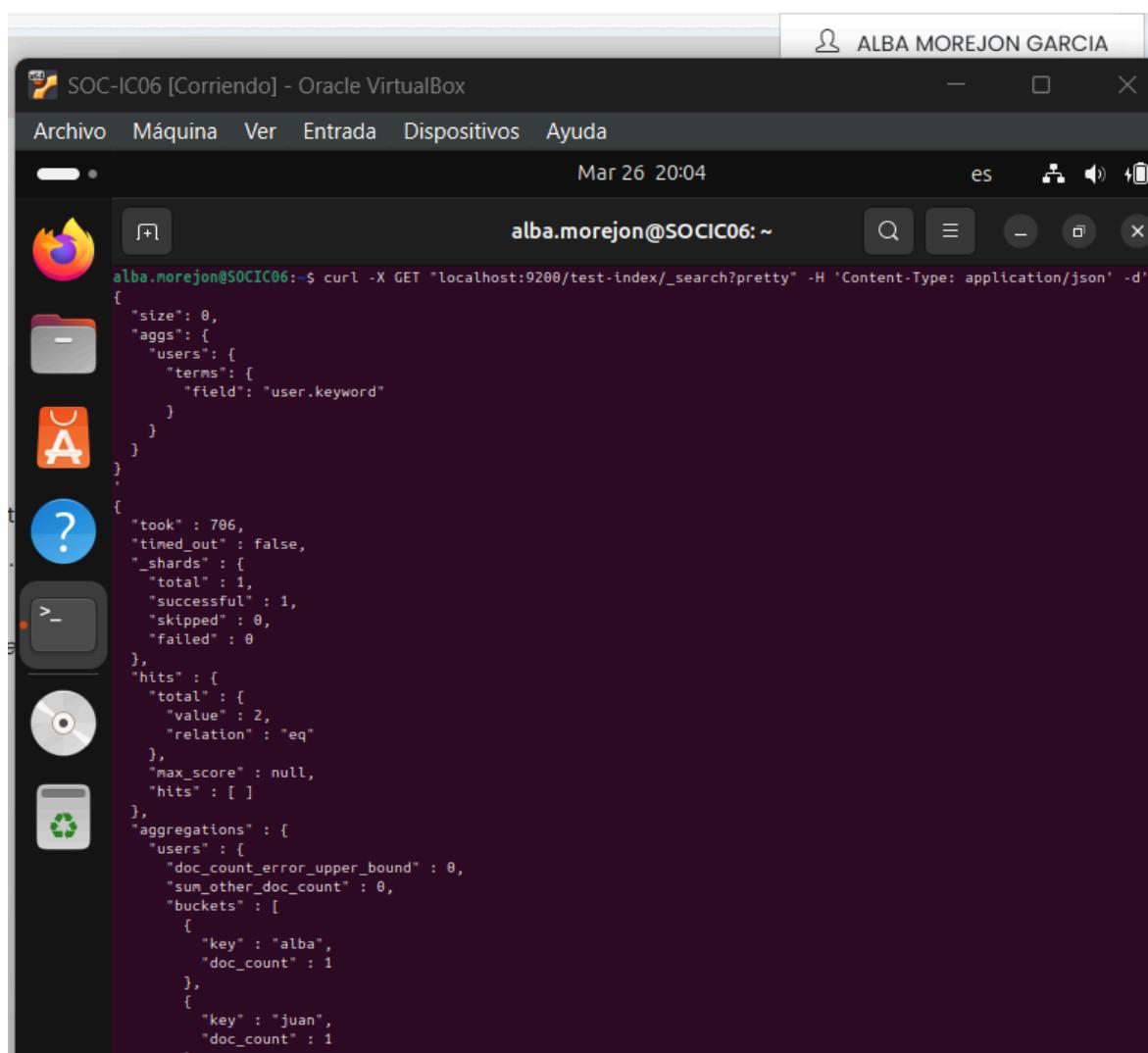


SOC-IC06 [Corriendo] - Oracle VirtualBox

```
alba.morejon@SOCIC06:~$ curl -X POST "localhost:9200/test-index/_doc/2?pretty" -H 'Content-Type: application/json' -d'
{
  "user": "juan",
  "message": "Probando Elasticsearch",
  "post_date": "2025-03-26"
}

{
  "_index": "test-index",
  "_type": "_doc",
  "_id": "2",
  "_version": 1,
  "result": "created",
  "_shards": {
    "total": 2,
    "successful": 1,
    "failed": 0
  },
  "_seq_no": 1,
  "_primary_term": 2
}'
```

Utilizamos el siguiente comando para contar cuantos documentos existen de cada usuario



SOC-IC06 [Corriendo] - Oracle VirtualBox

```
alba.morejon@SOCIC06:~$ curl -X GET "localhost:9200/test-index/_search?pretty" -H 'Content-Type: application/json' -d'
{
  "size": 0,
  "aggs": {
    "users": {
      "terms": {
        "field": "user.keyword"
      }
    }
  }
}

{
  "took": 706,
  "timed_out": false,
  "_shards": {
    "total": 1,
    "successful": 1,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 2,
      "relation": "eq"
    },
    "max_score": null,
    "hits": []
  },
  "aggregations": {
    "users": {
      "doc_count_error_upper_bound": 0,
      "sum_other_doc_count": 0,
      "buckets": [
        {
          "key": "alba",
          "doc_count": 1
        },
        {
          "key": "juan",
          "doc_count": 1
        }
      ]
    }
  }
}'
```

Obtenemos información y verificamos de nuevo que el clúster esté funcionando correctamente

```
alba.morejon@SOCIC06:~$ curl -X GET "localhost:9200/_cluster/stats?pretty"
{
  "_nodes": {
    "total": 1,
    "successful": 1,
    "failed": 0
  },
  "cluster_name": "siem-cluster",
  "cluster_uuid": "noHTe0JLQzGGuittEcR6dw",
  "timestamp": 174301950031,
  "status": "yellow",
  "indices": {
    "count": 1,
    "shards": {
      "total": 1,
      "primaries": 1,
      "replication": 0.0,
      "index": {
        "shards": {
          "min": 1,
          "max": 1,
          "avg": 1.0
        },
        "primaries": {
          "min": 1,
          "max": 1,
          "avg": 1.0
        },
        "replication": {}
      }
    }
  }
}
```

Apartado 2: Instalación y configuración de Kibana.

a) Detallar la configuración a efectuar en Kibana.

b) Prueba de acceso al menú principal de Kibana.

Descargamos e instalamos Kibana

```
alba.morejon@SOCIC06:~$ wget https://artifacts.elastic.co/downloads/kibana/kibana-7.10.2-amd64.deb
--2025-03-26 20:12:20-- https://artifacts.elastic.co/downloads/kibana/kibana-7.10.2-amd64.deb
Resolving artifacts.elastic.co (artifacts.elastic.co)... 34.120.127.130
Connecting to artifacts.elastic.co (artifacts.elastic.co)|34.120.127.130|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 253198756 (241M) [application/octet-stream]
Saving to: 'kibana-7.10.2-amd64.deb'

kibana-7.10.2-amd64.d 100%[=====] 241.47M 7.29MB/s   in 35s

2025-03-26 20:12:56 (6.90 MB/s) - 'kibana-7.10.2-amd64.deb' saved [253198756/253198756]

alba.morejon@SOCIC06:~$ sudo dpkg -i kibana-7.10.2-amd64.deb
[sudo] password for alba.morejon:
Selecting previously unselected package kibana.
(Reading database ... 150248 files and directories currently installed.)
Preparing to unpack kibana-7.10.2-amd64.deb ...
Unpacking kibana (7.10.2) ...
Setting up kibana (7.10.2) ...
```

Iniciamos el servicio y comprobamos el estado

```

SOC-IC06 [Corriendo] - Oracle VirtualBox
alba.morejon@SOCIC06: ~
Archivo Máquina Ver Entrada Dispositivos Ayuda
Mar 26 20:34 es
alba.morejon@SOCIC06: $ sudo systemctl start kibana
[sudo] password for alba.morejon:
alba.morejon@SOCIC06: $ sudo systemctl enable kibana
Synchronizing state of kibana.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable kibana
Created symlink /etc/systemd/system/multi-user.target.wants/kibana.service → /etc/systemd/system/kibana.service.
alba.morejon@SOCIC06: $ sudo systemctl status kibana
● kibana.service - Kibana
    Loaded: loaded (/etc/systemd/system/kibana.service; enabled; preset: enabled)
    Active: active (running) since Wed 2025-03-26 20:34:04 UTC; 15s ago
      Main PID: 6248 (node)
        Tasks: 11 (limit: 4609)
       Memory: 334.6M (peak: 334.8M)
         CPU: 7.735s
        CGroup: /system.slice/kibana.service
                └─6248 /usr/share/kibana/bin/../node/bin/node /usr/share/kibana/bin/..src

Mar 26 20:34:14 SOCIC06 kibana[6248]: {"type":"log","@timestamp":"2025-03-26T20:34:14Z"}>

```

Editamos el archivo de configuración `kibana.yml`, añadimos el puerto en el que la herramienta está escuchando, configuraremos todas las interfaces de red que escuchará, especificamos la dirección Elasticsearch al que se conectará y define el nombre del índice en Elasticsearch donde Kibana almacenará sus datos.

```

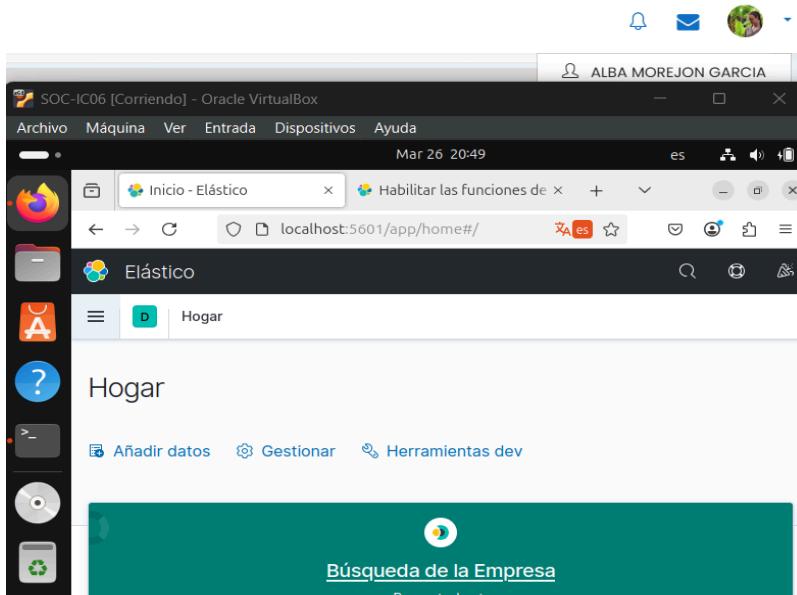
SOC-IC06 [Corriendo] - Oracle VirtualBox
alba.morejon@SOCIC06: ~
Archivo Máquina Ver Entrada Dispositivos Ayuda
Mar 26 20:36 es
alba.morejon@SOCIC06: ~
GNU nano 7.2 /etc/kibana/kibana.yml *
# Kibana is served by a back end server. This setting specifies the port to use.
server.port: 5601
server.host: "0.0.0.0"
elasticsearch.hosts: ["http://localhost:9200"]
kibana.index: ".kibana"

# specifies the address to which the Kibana servers will bind. IP addresses and host names

```

Comprobamos que funcione correctamente navegando en la web: <http://localhost:5601>

- Cargamos la página
- Creamos el patrón de índice para poder gestionarlo (elegimos el índice creado con `elasticsearch`)



- En la sección visualize, creamos una nueva visualización con los datos del índice text-index
- Creamos un Dashboard

Apartado 3: Instalación y configuración de Filebeat.

a) Detallar la configuración a efectuar en el agente IDS para enviar los registros de Snort a Logstash.

b) Mostrar una prueba de funcionamiento de Filebeat mostrando los registros por consola.

Descargamos e instalamos Filebeat

```

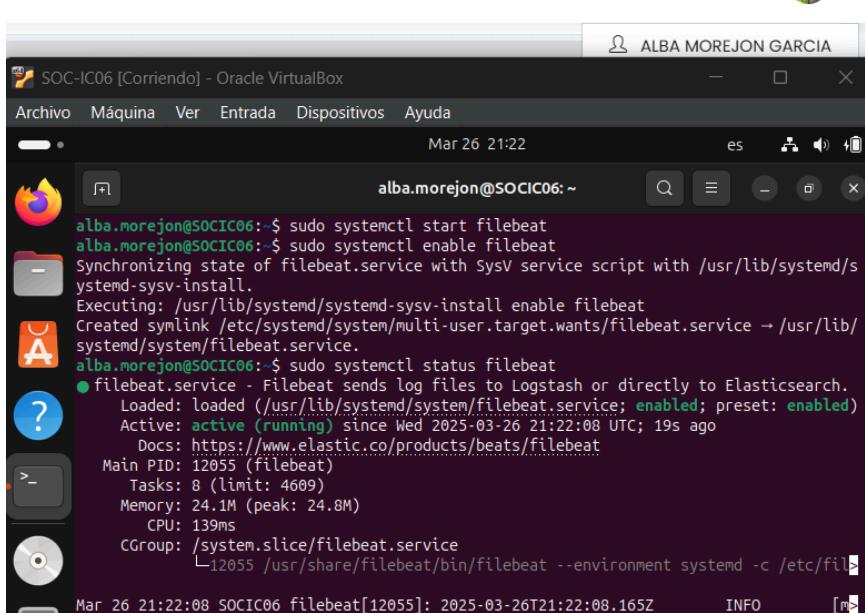
alba.morejon@SOCIC06:~$ wget https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-7.10.2-amd64.deb
--2025-03-26 21:20:00-- https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-7.10.2-amd64.deb
Resolving artifacts.elastic.co (artifacts.elastic.co)... 34.120.127.130
Connecting to artifacts.elastic.co (artifacts.elastic.co)|34.120.127.130|:443...
HTTP request sent, awaiting response... 200 OK
Length: 34274304 (33M) [application/octet-stream]
Saving to: 'filebeat-7.10.2-amd64.deb'

filebeat-7.10.2-amd64 100%[=====] 32.69M 8.09MB/s in 4.7s
2025-03-26 21:20:06 (6.99 MB/s) - 'filebeat-7.10.2-amd64.deb' saved [34274304/34274304]

alba.morejon@SOCIC06:~$ sudo dpkg -i filebeat-7.10.2-amd64.deb
[sudo] password for alba.morejon:
Selecting previously unselected package filebeat.
(Reading database ... 207348 files and directories currently installed.)
Preparing to unpack filebeat-7.10.2-amd64.deb ...
Unpacking filebeat (7.10.2) ...
Setting up filebeat (7.10.2) ...
alba.morejon@SOCIC06:~$ 

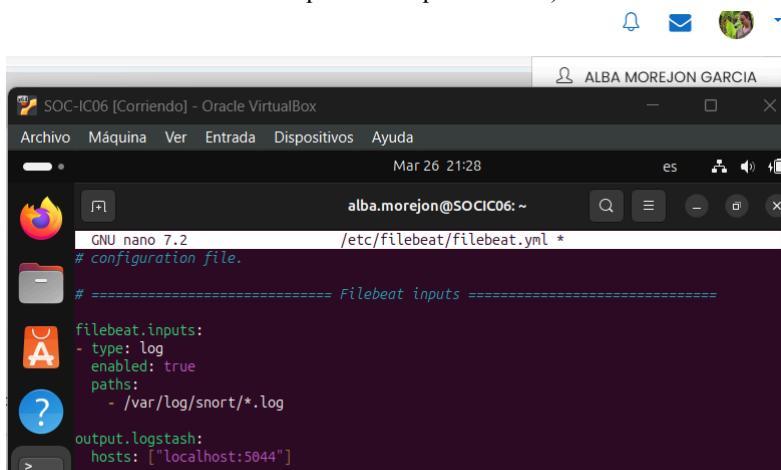
```

Iniciamos el servicio



```
alba.morejon@SOCIC06:~$ sudo systemctl start filebeat
alba.morejon@SOCIC06:~$ sudo systemctl enable filebeat
Synchronizing state of filebeat.service with sysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable filebeat
Created symlink /etc/systemd/system/multi-user.target.wants/filebeat.service → /usr/lib/systemd/filebeat.service.
alba.morejon@SOCIC06:~$ sudo systemctl status filebeat
● filebeat.service - Filebeat sends log files to Logstash or directly to Elasticsearch.
   Loaded: loaded (/usr/lib/systemd/system/filebeat.service; enabled; preset: enabled)
   Active: active (running) since Wed 2025-03-26 21:22:08 UTC; 19s ago
     Docs: https://www.elastic.co/products/beats/filebeat
      Main PID: 12055 (filebeat)
        Tasks: 8 (limit: 4609)
       Memory: 24.1M (peak: 24.8M)
          CPU: 139ms
         CGroup: /system.slice/filebeat.service
                  └─12055 /usr/share/filebeat/bin/filebeat --environment systemd -c /etc/fi
```

Configuraremos Filebeat para que pueda leer los registros de Snort (habilitamos la entrada de logs, con la ruta de archivos que debe leer) y enviarlos Logstash (especificando la dirección del servidor al que el servicio mandará los registros, utilizaremos esta misma máquina en el puerto 5044).



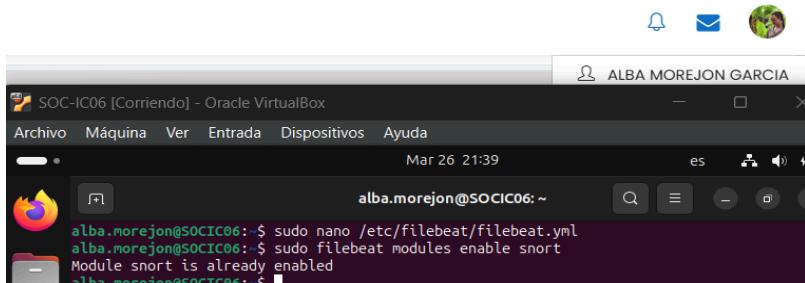
```
alba.morejon@SOCIC06:~$ nano /etc/filebeat/filebeat.yml
# configuration file.

# ===== Filebeat inputs =====

filebeat.inputs:
- type: log
  enabled: true
  paths:
    - /var/log/snort/*.log

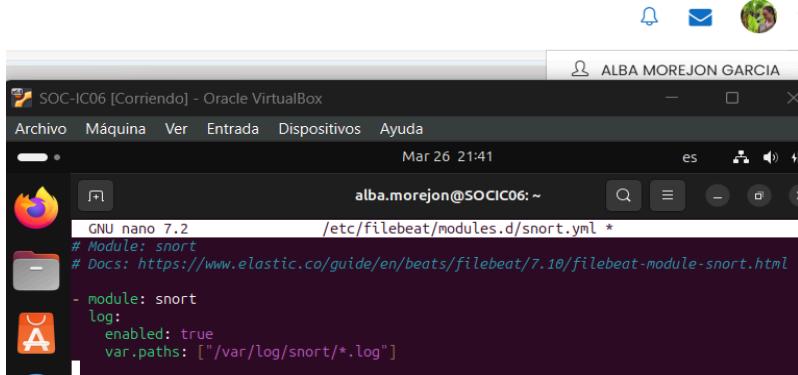
output.logstash:
  hosts: ["localhost:5044"]
```

Habilitamos el módulo de Filebeat para Snort



```
alba.morejon@SOCIC06:~$ sudo nano /etc/filebeat/filebeat.yml
alba.morejon@SOCIC06:~$ sudo filebeat modules enable snort
Module snort is already enabled
alba.morejon@SOCIC06:~$
```

Configuramos el módulo de Snort en el archivo snort.yml



```
alba.morejon@SOCIC06:~$ nano /etc/filebeat/modules.d/snort.yml
# Module: snort
# Docs: https://www.elastic.co/guide/en/beats/filebeat/7.10/filebeat-module-snort.html

- module: snort
  log:
    enabled: true
    var.paths: ["/var/log/snort/*.log"]
```

Verificamos que los puertos estén abiertos

```

alba.morejon@SOCIC06:~$ sudo netstat -tuln | grep 5044
tcp6       0      0 :::5044          :::*        LISTEN
alba.morejon@SOCIC06:~$ sudo ss -tuln | grep 5044
tcp    LISTEN  0      4096          *:5044          :::*        LISTEN
alba.morejon@SOCIC06:~$ sudo ufw allow 5044
Rule added
Rule added (v6)
alba.morejon@SOCIC06:~$ sudo ufw status
Status: active
To                         Action      From
--                         --          --
5601                       ALLOW      Anywhere
5044                       ALLOW      Anywhere
5601 (v6)                  ALLOW      Anywhere (v6)
5044 (v6)                  ALLOW      Anywhere (v6)

```

Hacemos una prueba de verificar la configuración y probamos la salida de Logstash

```

alba.morejon@SOCIC06:~$ sudo filebeat test config
Config OK
alba.morejon@SOCIC06:~$ sudo filebeat test output
logstash: localhost:5044...
connection...
  parse host... OK
  dns lookup... OK
  addresses: 127.0.0.1
  dial up... OK
TLS... WARN secure connection disabled
  talk to server... OK
alba.morejon@SOCIC06:~$ 

```

Ejecutamos Filebeat en modo depuración para ver los registros

```

alba.morejon@SOCIC06:~$ sudo filebeat -e -d "*"
2025-03-26T22:00:49.176Z     INFO  instance/beat.go:645  Home path: [/usr/share/filebeat] Config path: [/etc/filebeat] Data path: [/var/lib/filebeat] Logs path: [/var/log/filebeat]
2025-03-26T22:00:49.176Z     DEBUG  [beat]  instance/beat.go:697  Beat metadata path: /var/lib/filebeat/meta.json
2025-03-26T22:00:49.176Z     INFO  instance/beat.go:653  Beat ID: 5638821c-381d-4e24-85c1-2adee9fa74a94
2025-03-26T22:00:49.176Z     DEBUG  [conditions]  conditions/conditions.go:98  New condition contains: map[]
2025-03-26T22:00:49.193Z     DEBUG  [conditions]  conditions/conditions.go:98  New condition !contains: map[]
2025-03-26T22:00:49.193Z     DEBUG  [docker]  docker/client.go:48  Docker client will negotiate the API version on the first request.
2025-03-26T22:00:49.193Z     DEBUG  [add_docker_metadata]  add_docker_metadata/add_

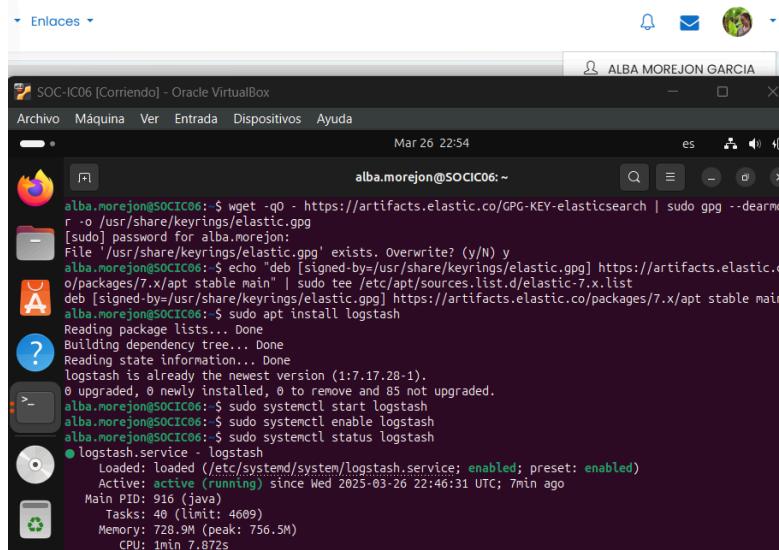
```

Apartado 4: Instalación y configuración de Logstash.

a) Detallar la configuración a efectuar en Logstash para recibir los logs de Filebeat y reenviarlos a Elasticsearch.

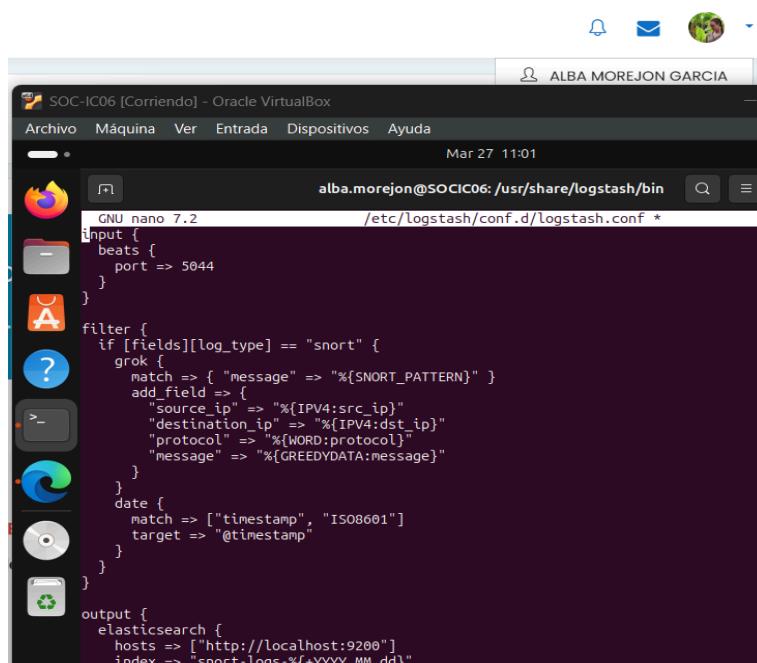
b) Mostrar una prueba de funcionamiento en la que se puede visualizar la información del índice de logstash en Kibana.

Agregamos la clave de Elasticshared y el repositorio de Logstash. Instalamos Logstash, iniciamos y habilitamos el servicio Logstash comprobando que su estado correcto



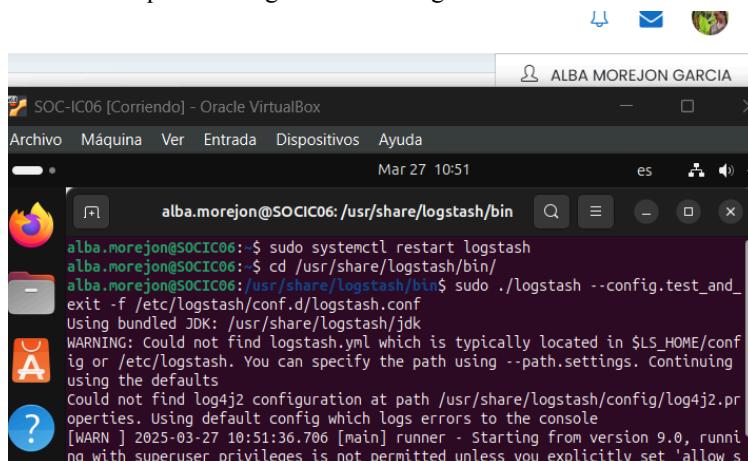
```
alba.morejon@SOCIC06:~$ wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo gpg --dearm  
[sudo] password for alba.morejon:  
File '/usr/share/keyrings/elastic.gpg' exists. Overwrite? (y/N) y  
alba.morejon@SOCIC06:~$ echo 'deb [signed-by=/usr/share/keyrings/elastic.gpg] https://artifacts.elastic.c  
o/packages/7.x/apt stable main' | sudo tee /etc/apt/sources.list.d/elasticsearch-7.x.list  
deb [signed-by=/usr/share/keyrings/elastic.gpg] https://artifacts.elastic.co/packages/7.x/apt stable main  
alba.morejon@SOCIC06:~$ sudo apt install logstash  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
logstash is already the newest version (1:7.17.28-1).  
0 upgraded, 0 newly installed, 0 to remove and 85 not upgraded.  
alba.morejon@SOCIC06:~$ sudo systemctl start logstash  
alba.morejon@SOCIC06:~$ sudo systemctl enable logstash  
alba.morejon@SOCIC06:~$ sudo systemctl status logstash  
● logstash.service - logstash  
   Loaded: loaded (/etc/systemd/system/logstash.service; enabled; preset: enabled)  
   Active: active (running) since Wed 2025-03-26 22:46:31 UTC; 7min ago  
     Main PID: 916 (java)  
       Tasks: 40 (limit: 4609)  
      Memory: 728.9M (peak: 756.5M)  
        CPU: 1min 7.872s
```

Añadimos la configuración para Filebeat



```
alba.morejon@SOCIC06:/usr/share/logstash/bin$ nano /etc/logstash/conf.d/logstash.conf  
input {  
  beats {  
    port => 5044  
  }  
}  
  
filter {  
  if [fields][log_type] == "snort" {  
    grok {  
      match => { "message" => "%{SNORT_PATTERN}" }  
      add_field => {  
        "source_ip" => "%{IPV4:src_ip}"  
        "destination_ip" => "%{IPV4:dst_ip}"  
        "protocol" => "%{WORD:protocol}"  
        "message" => "%{GREEDYDATA:message}"  
      }  
    }  
    date {  
      match => ["timestamp", "ISO8601"]  
      target => "@timestamp"  
    }  
  }  
}  
  
output {  
  elasticsearch {  
    hosts => ["http://localhost:9200"]  
    index => "snort-logs-%{+YYYY.MM.dd}"  
  }  
}
```

Verificamos que la configuración no tenga errores



```
alba.morejon@SOCIC06:~$ sudo systemctl restart logstash  
alba.morejon@SOCIC06:~$ cd /usr/share/logstash/bin/  
alba.morejon@SOCIC06:/usr/share/Logstash/bin$ sudo ./logstash --config.test_and_  
exit -f /etc/logstash/conf.d/logstash.conf  
Using bundled JDK: /usr/share/logstash/jdk  
WARNING: Could not find logstash.yml which is typically located in $LS_HOME/conf  
ig or /etc/logstash. You can specify the path using --path.settings. Continuing  
using the defaults  
Could not find log4j2 configuration at path /usr/share/logstash/config/log4j2.pr  
operties. Using default config which logs errors to the console  
[WARN] 2025-03-27 10:51:36.706 [main] runner - Starting from version 9.0, runni  
ng with superuser privileges is not permitted unless you explicitly set 'allow_s
```

Le añadimos la siguiente información a los ficheros logstash.yml

```

alba.morejon@SOCIC06:/etc/logstash
# ----- Node identity -----
# Use a descriptive name for the node:
#
# node.name: test
node.name: "logstash-node"
path.data: "/var/lib/logstash"
path.logs: "/var/log/logstash"
pipeline.workers: 2
pipeline.batch.size: 125
pipeline.batch.delay: 50

```

```

alba.morejon@SOCIC06:/usr/share/logstash/config
# ----- Node identity -----
# Use a descriptive name for the node:
#
# node.name: test
node.name: "logstash-node"
path.data: "/var/lib/logstash"
path.logs: "/var/log/logstash"
pipeline.workers: 2
pipeline.batch.size: 125
pipeline.batch.delay: 50

```

Creamos el fichero log4j2.properties y añadimos lo siguiente

```

alba.morejon@SOCIC06:/usr/share/logstash/config
status = error
name = LogstashPropertiesConfig
appender.console.type = Console
appender.console.name = console
appender.console.layout.type = PatternLayout
appender.console.layout.pattern = "%d{ISO8601} [%t] %-5p %c{1} %marker - %m%n"
rootLogger.level = error
rootLogger.appenderRefs = console
rootLogger.appenderRef.console.ref = console

```

Ejecutamos logs en modo depuración para ver los registros de consola

```

alba.morejon@SOCIC06: $ sudo /usr/share/logstash/bin/logstash --path.settings /usr/share/logstash/config -f /etc/logstash/conf.d/logstash.conf --log.level debug
Using bundled JDK: /usr/share/logstash/jdk
Sending Logstash logs to /var/log/logstash which is now configured via log4j2.properties
"2025-03-27T11:11:52,120 [main] WARN runner - Starting from version 9.0, running with superuser privileges is not permitted unless you explicitly set 'allow_superuser' to true, thereby acknowledging the possible security risks
""2025-03-27T11:11:52,129 [main] WARN runner - NOTICE: Running Logstash as a superuser is strongly discouraged as it poses a security risk. Set 'allow_superuser' to false for better security.
""2025-03-27T11:11:52,144 [main] INFO runner - Log4j configuration path used is: /usr/share/logstash/config/log4j2.properties
""2025-03-27T11:11:52,145 [main] WARN runner - 'pipeline.buffer.type' setting is not explicitly defined.Before moving to 9.x set it to 'heap' and tune heap size upward, or set it to 'direct' to maintain existing behavior.
""2025-03-27T11:11:52,148 [main] INFO runner - Starting Logstash {"logstash.version"=>"8.17.4", "jruby.version"=>"jruby 9.4.9.0 (3.1.4) 2024-11-04 547c6b150e OpenJDK 64-Bit Server VM 21.0.6+7-LTS on 21.0.6+7-LTS +indy +jit [x86_64-linux]"}
""2025-03-27T11:11:52,153 [main] INFO runner - JVM bootstrap flags: [-Xms1g, -Xmx1g, -Djava.awt.headless=true, -Dfile.encoding=UTF-8, -Djruby.compile.invokedynamic=true, -XX:+HeapDumpOnOutOfMemoryError, -Djava.security.egd=file:/dev/urandom, -Dlog4j2.isThreadContextMapInheritable=true, -Dlogstash.jackson.stream-read-constraints.max-string-length=200000000, -Dlogstash.jackson.stream-r

```

Editamos el archivo filebeat.yml y vemos los logs

```

GNU nano 7.2                               /etc/filebeat/filebeat.yml *
# For more available modules and options, please see the filebeat.reference.yml sample
# configuration file.

# ===== Filebeat inputs =====

filebeat.inputs:
- type: log
  enabled: true
  paths:
    - /var/log/snort/*.log
    fields:
      log_type: snort

output.logstash:
  hosts: ["localhost:5044"]

logging.level: debug
logging.to_files: true
logging.files:
  path: /var/log/filebeat
  name: filebeat
  keepfiles: 7
  permissions: 0644

```

```

root@SOCIC06:/var/log/filebeat# sudo tail -f /var/log/filebeat/filebeat
2025-03-27T11:42:34.462Z DEBUG [input] log/input.go:226      input
states cleaned up. Before: 0, After: 0, Pending: 0
2025-03-27T11:42:37.037Z DEBUG [input] input/input.go:139      Run in
put
2025-03-27T11:42:44.462Z DEBUG [input] input/input.go:139      Run in
put
2025-03-27T11:42:44.462Z DEBUG [input] log/input.go:205      Start
next scan
2025-03-27T11:42:44.462Z DEBUG [input] log/input.go:226      input
states cleaned up. Before: 0, After: 0, Pending: 0
2025-03-27T11:42:47.038Z DEBUG [input] input/input.go:139      Run in

```

Apartado 5: Creación de un filtro en Logstash.

a) Detallar la configuración a efectuar en Logstash para crear un Pipeline que filtre la información creando campos para los diferentes valores del mensaje de log creado en Snort.

b) Mostrar el índice que se crea con la información de sus diferentes campos.

Configuraremos el fichero logstash.conf

```

GNU nano 7.2                               /etc/logstash/conf.d/logstash.conf *
input {
  beats {
    port => 5044
  }
}

filter {
  if [fields][log_type] == "snort" {
    grok {
      match => { "message" => "%{SNORT_PATTERN}" }
      add_field => {
        "source_ip" => "%{IPV4:src_ip}"
        "destination_ip" => "%{IPV4:dst_ip}"
        "protocol" => "%{WORD:protocol}"
        "message" => "%{GREEDYDATA:message}"
      }
      date {
        match => ["timestamp", "ISO8601"]
        target => "@timestamp"
      }
    }
  }
}

output {
  elasticsearch {
    hosts => ["http://localhost:9200"]
    index => "snort-logs-%{+YYYY.MM.dd}"
  }
}

```

Creamos el índice snort-logs de forma manual

The top screenshot shows the "Create index pattern - Elasticsearch" page with the URL localhost:5601/app/dev_tools#/console. It displays a JSON configuration for the "snort-logs" index pattern:

```

1 PUT snort-logs-000001
2 {
3   "_index": "snort-logs",
4   "settings": {
5     "number_of_shards": 1,
6     "number_of_replicas": 1
7   },
8   "mappings": {
9     "properties": {
10       "@timestamp": {
11         "type": "date"
12     },
13     "source_ip": {
14       "type": "ip"
15     },
16     "destination_ip": {
17       "type": "ip"
18     },
19     "protocol": {
20       "type": "keyword"
21     },
22     "message": {
23       "type": "text"
24     }
25   }
26 }

```

The status bar indicates "200-OK" and "30 ms".

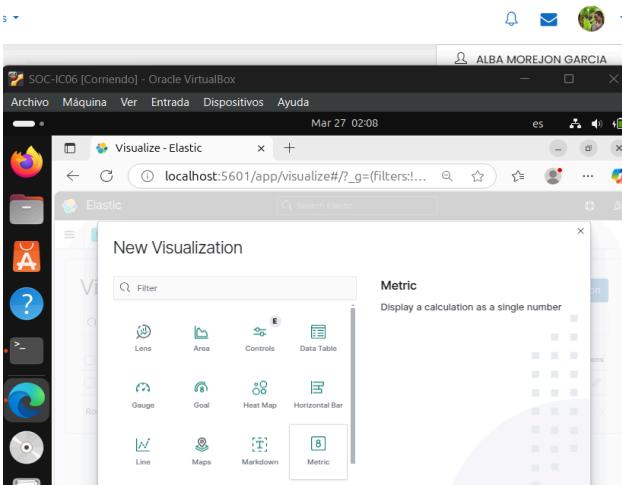
The bottom screenshot shows the "Index patterns" section of the Kibana management interface at the URL 127.0.0.1:5601/app/management/kibana/indexPatterns/snort-logs-*. It lists the fields defined in the index pattern:

Name	Type	Format	Searchable	Aggregatable	Excluded
_id	string		●	●	
_index	string		●	●	
_score	number				
_source	_source	_source			
_type	string		●	●	
destination_ip	ip	IP address	●	●	
message	string		●		
protocol	string		●	●	
source_ip	ip	IP address	●	●	

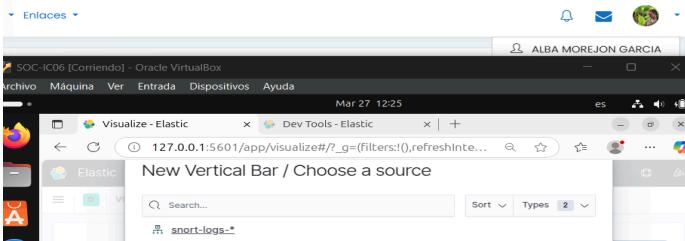
Apartado 6: Tablero de Monitorización Multipunto.

- Detallar la creación de dos contadores, uno para todos los PINGs desde la red interna y otro para los de la red externa.**
- Detallar la creación de dos histogramas, uno para los intentos de inicio de sesión por SSH y otro para los accesos a PHPMyadmin.**
- Detallar la creación de un nuevo tablero (dashboard) en Kibana que contenga los contadores y los histogramas creados anteriormente.**

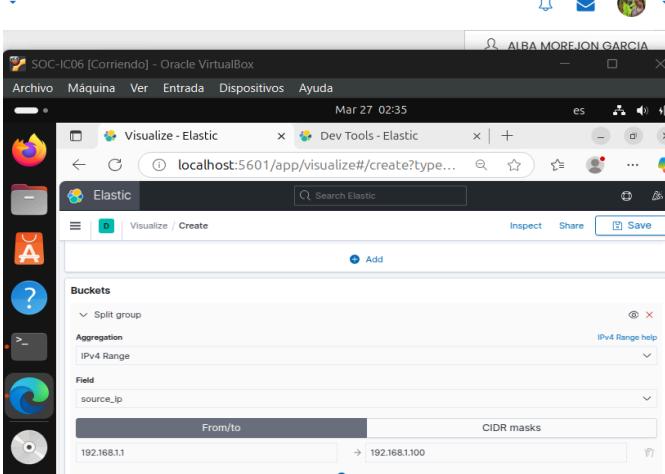
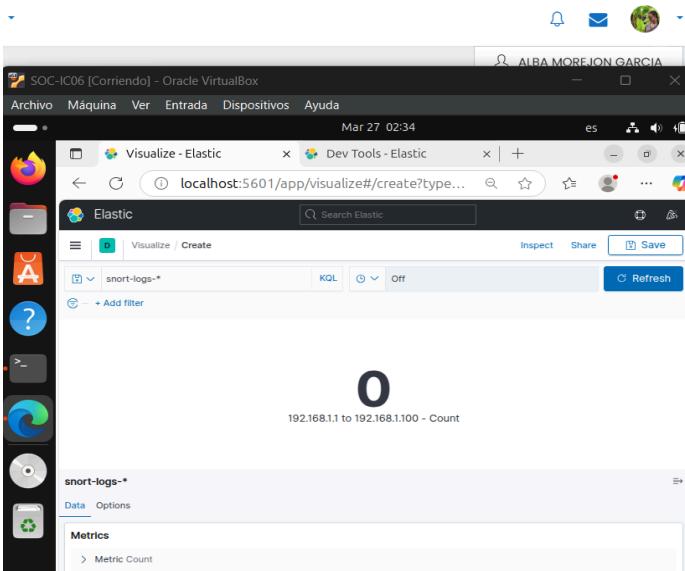
Seleccionando en el menú lateral el apartado Visualize (dentro de Kibana). y pulsamos en “Create visualization”. Nos da la opción de nos da la opción de seleccionar Metric y procedemos a crear dos visualizaciones para establecer red donde van a actuar.



Elegimos el índice que creamos

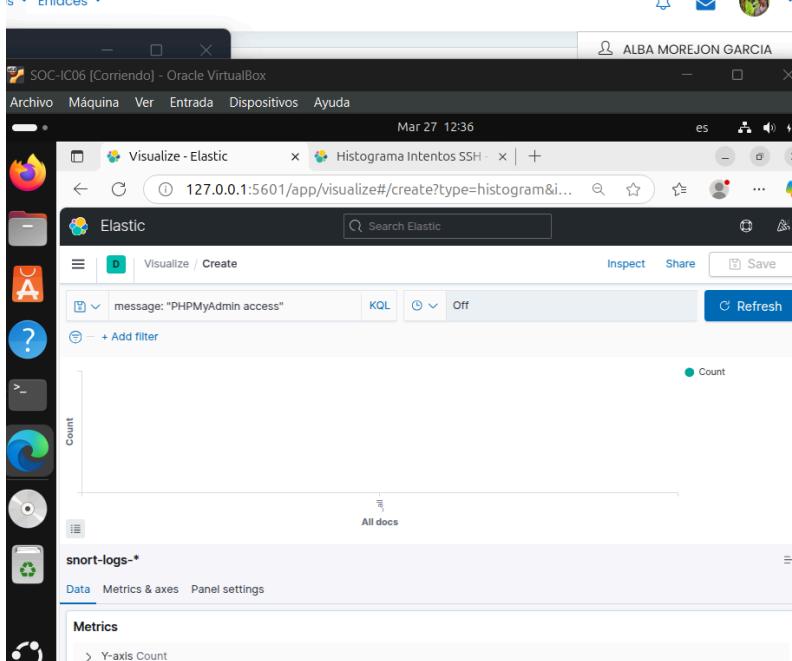
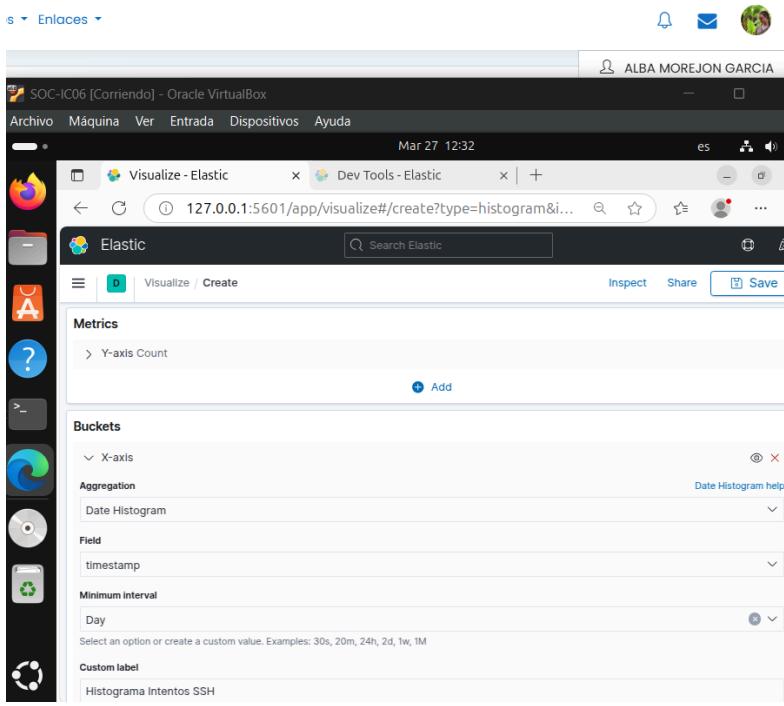
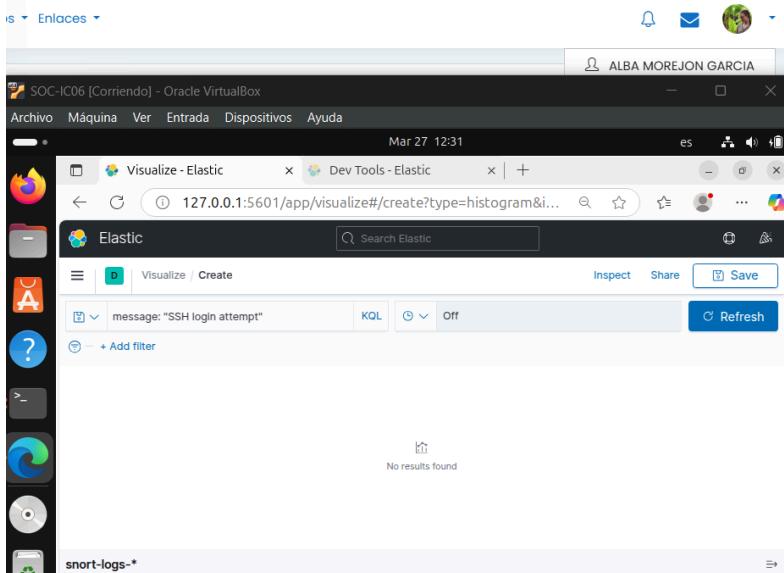


Establecemos los valores de Count y le damos el rango de IPs donde actuará



Haremos lo mismo pero cambiando el rango para la red externa

A continuación crearemos otras visualizaciones de tipo Histograma para los inicios de sesión con SSH y PHPMyadmin dando los siguientes valores



Creamos un nuevo Dashboard, seleccionando en el menú lateral “Dashboard” y haciendo click sobre “Create Dashboard” y añadimos las vistas creadas anteriormente

The image consists of two screenshots of a Linux desktop environment showing the Elastic Stack interface.

Screenshot 1 (Top): Creating a New Dashboard

This screenshot shows the "Dashboards - Elastic" section of the interface. A modal window titled "Add panels" is open, listing several objects available for addition:

- Contador PINGs Red Externa
- Contador PINGs Red Interna
- Histograma Intentos SSH
- PHPMyAdmin

Screenshot 2 (Bottom): Viewing an Existing Dashboard

This screenshot shows the "IC07 - Elastic" dashboard. It contains four panels:

- Contador PINGs Red Interna:** Shows a value of 0 for the range 192.168.1.1 to 192.168.1.100.
- Contador PINGs Red Externa:** Shows a value of 0 for the range 10.0.2.1 to 10.0.2.10.
- Histograma Intentos SSH:** Shows a message "No results found".
- PHPMyAdmin:** Shows a message "No results found".