

Para recopilar las diferentes métricas de las máquinas virtuales además de información sobre los nodos se han configurado los siguientes archivos y se han activado diferentes paquetes.

Metricbeat.yml

```
output.elasticsearch:
  # Array of hosts to connect to.
  hosts: ["https://192.199.1.59:9200", "https://192.199.1.60:9200", "https://192.199.1.61:9200"]

  # Performance preset - one of "balanced", "throughput", "scale",
  # "latency", or "custom".
  preset: balanced

  # Protocol - either `http` (default) or `https`.
  #protocol: "https"

  # Authentication credentials - either API key or username/password.
  api_key: "gFMZwJoB_ng_JLRryeaj:hh01Bw7gE8r7mPPsFPBZxA"
  #username: "elastic"
  #password: "elastic"

  ssl.certificate_authorities: ["/home/g2/ELK/elasticsearch-9.2.1/config/certs/http_ca.crt"]
```

1. **hosts:**
Define los nodos de Elasticsearch a los que Metricbeat se va a conectar. En este caso, hay tres nodos con HTTPS en los puertos 9200.
2. **preset:**
Es una configuración de rendimiento. "balanced" significa que Metricbeat intentará un equilibrio entre velocidad y consumo de recursos.
3. **api_key:**
Es la credencial que Metricbeat usa para autenticarse con Elasticsearch. En lugar de usuario y contraseña, se está usando un **API key**.
4. **ssl.certificate_authorities:**
Especifica la ruta al certificado de la autoridad certificadora (CA). Esto permite que Metricbeat verifique que la conexión HTTPS con Elasticsearch es segura y confiable.

Esta sección configura Metricbeat para enviar los datos de monitoreo a un cluster de Elasticsearch seguro (HTTPS), usando un API key y verificando los certificados SSL de los nodos.

```
setup.kibana:
  host: "http://192.199.1.59:5601"
  api_key: gFMZwJoB_ng_JLRryeaj:hh01Bw7gE8r7mPPsFPBZxA
```

1. **setup.kibana:**

Esta sección se usa para configurar cómo Metricbeat se comunica con **Kibana**. Metricbeat puede enviar dashboards, visualizaciones y plantillas de índices a Kibana cuando se inicializa.

2. **host: "http://192.199.1.59:5601"**

Aquí defines la URL donde está corriendo Kibana. En este caso:

- **192.199.1.59** es la IP del servidor donde está Kibana.
- **5601** es el puerto por defecto de Kibana.

3. Metricbeat usará esta dirección para conectarse y enviar configuraciones de dashboards o plantillas de visualización.

4. **api_key: gFMzWJoB_ng_JLRryeaj:hh01Bw7gE8r7mPPsFPBZxA**

Este es un **API Key** que Metricbeat usa para autenticarse con Kibana. Es más seguro que usar usuario y contraseña.

En resumen, esta parte del `metricbeat.yml` le dice a Metricbeat dónde está Kibana y cómo autenticarse para poder enviar dashboards y configuraciones automáticamente.

Paquete Elasticsearch-xpack

Este paquete sirve para monitorizar los nodos de elasticsearch, se ha instalado este pack en la maquina virtual 1 que es donde esta tambien kibana. El contenido de este módulo es el siguiente en el **elasticsearch-xpack**.

```
- module: elasticsearch
  xpack.enabled: true
  period: 10s
  hosts: ["https://192.199.1.59:9200", "https://192.199.1.60:9200", "https://192.199.1.61:9200"]
  ssl:
    certificate_authorities: ["/home/g2/ELK/elasticsearch-9.2.1/config/certs/http_ca.crt"]
```

Con esto podemos ver lo siguiente.

The screenshot shows the Elasticsearch Kibana dashboard with the following sections:

- Overview:** Health is ● Healthy. Version is 9.2.1. Uptime is 6 days. License is Basic.
- Nodes: 3** (with 1 alert icon):
 - Disk Available: 49.75% (70.3 GB / 141.2 GB)
 - JVM Heap: 44.36% (2.7 GB / 6.0 GB)
- Indices: 69**:
 - Documents: 3,498,647
 - Disk Usage: 2.7 GB
 - Primary Shards: 73
 - Replica Shards: 73
- Logs:** No log data found. A message suggests setting up Filebeat and configuring the Elasticsearch output.

Status	Alerts	Nodes	Indices	JVM Heap	Total shards	Unassigned shards	Documents	Data
Green	1	3	69	1.2 GB / 6.0 GB	146	0	3,499,520	2.7 GB

Filter Nodes...								
Name ↑	Alerts ↓	Status ↓	Roles ↓	Shards ↓	CPU Usage ↓	Load Average ↓	JVM Heap ↓	Disk Free Space ↓
<div>node-1</div> <div>192.199.1.59:9300</div>	Clear	Online	N/A	48	↓ 1%	↓ 0.04	↑ 63%	↓ 32.1 GB
<div>node-2</div> <div>192.199.1.60:9300</div>	Clear	Online	N/A	49	↓ 1%	↓ 0.14	↑ 19%	↓ 29.8 GB
<div>★ node-3</div> <div>192.199.1.61:9300</div>	1 alert	Online	N/A	49	↓ 1%	↑ 0.02	↑ 76%	↓ 8.4 GB

Activar Windows
Ve a Configuración para activar Windows.

Rows per page: 20

Paquete kibana-xpack

Este paquete sirve para monitorizar el nodo de kibana. Vemos la configuración del **kibana-xpack.yml**.

```
- module: kibana
  xpack.enabled: true
  period: 10s
  hosts: ["http://192.199.1.59:5601"]
  #basepath: ""
  ssl:
    certificate_authorities: ["/home/g2/ELK/elasticsearch-9.2.1/config/certs/http_ca.crt"]
```

Con esto podemos ver lo siguiente.

Status

Alerts

Instances

Memory

Requests

Connections

Max. Response Time

● Green

● 0

1

794.2 MB / 2.0 GB

6

7

190 ms

Filter Instances...

Name ↑	Alerts ↓	Last Reported Status ↓	Last Seen ↓	Load Average ↓	Memory Size ↓	Requests ↓	Response Times ↓
g2n1	● Clear	● Green	In a few seconds	0.00	794.2 MB	6	68 ms avg 190 ms max

Rows per page: 20

<

1

>

Paquete system

Con este paquete podemos ver métricas de las máquinas virtuales como el uso de cpu, ram, disco... Este paquete está instalado en cada metricbeat de cada una de las máquinas. Código del **system.yml**.

```
- module: system
period: 10s
metricsets:
  - cpu
  - memory
  - network
  - filesystem
  - diskio
  - load
  - process
  - process_summary
  #- entropy
  #- core
  #- diskio
  #- socket
  #- service
  #- users
process.include_top_n:
  by_cpu: 5      # include top 5 processes by CPU
  by_memory: 5   # include top 5 processes by memory
degrade_on_partial: false # mark metricset as degraded if partial metrics are emitted
# Configure the mount point of the host's filesystem for use in monitoring a host from within a container
# hostfs: "/hostfs"
```

```
process.include_top_n:
  by_cpu: 5      # include top 5 processes by CPU
  by_memory: 5   # include top 5 processes by memory
degrade_on_partial: false # mark metricset as degraded if partial metrics are emitted
# Configure the mount point of the host's filesystem for use in monitoring a host from within a container
# hostfs: "/hostfs"

- module: system
period: 1m
metricsets:
  - filesystem
  - fsstat
processors:
  - drop_event.when.regex:
    | system.filesystem.mount_point: '^/(sys|cgroup|proc|dev|etc|host|lib|snap)(|/)'

- module: system
period: 15m
metricsets:
  - uptime
```

Con esto podemos ver lo siguiente.



