

## Introduction

La stack ELK, composée d'Elasticsearch, Logstash et Kibana, est une puissante suite d'outils utilisée pour gérer les journaux (logs) et analyser les données en temps réel. Chaque composant de cette suite joue un rôle spécifique dans ce processus :

1. **Elasticsearch** : Il s'agit du moteur de recherche et de stockage distribué qui permet de stocker, indexer et rechercher efficacement les données, y compris les logs.
2. **Logstash** : Logstash est l'outil de collecte et de transformation des logs. Il prend en charge la collecte de données à partir de diverses sources, les transforme en un format cohérent et les envoie à Elasticsearch pour le stockage.
3. **Kibana** : Kibana est l'interface utilisateur qui permet de visualiser et d'analyser les données stockées dans Elasticsearch. C'est un outil essentiel pour créer des tableaux de bord, des graphiques et des visualisations pour comprendre et surveiller les données en temps réel.

En combinant ces trois composants, la stack ELK offre une solution complète pour gérer les logs et obtenir des informations précieuses à partir de ces données, facilitant ainsi la résolution de problèmes, la surveillance en temps réel et l'analyse des tendances.

# Installation

## Elasticsearch

**Étape 1 : Ajoutez la clé GPG pour télécharger le package Elasticsearch.**

```
wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -
```

**Étape 2 : Installez le support de transport HTTPS pour APT.**

```
sudo apt install apt-transport-https
```

**Étape 3 : Ajoutez le dépôt Elasticsearch à votre liste de sources.**

```
echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" | sudo tee -a  
/etc/apt/sources.list.d/elastic-7.x.list
```

**Étape 4 : Mettez à jour les dépôts et installez Elasticsearch.**

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

**Étape 5 : Configurez Elasticsearch (le fichier de configuration est `/etc/elasticsearch/elasticsearch.yml`). Pour le moment, laissez-le tel quel.**

**Étape 6 : Démarrez le service Elasticsearch.**

```
sudo service elasticsearch start
```

**Étape 7 : Pour que le service se lance automatiquement au démarrage, utilisez la commande suivante :**

```
sudo systemctl enable elasticsearch
```

Cela simplifie la documentation tout en maintenant les étapes essentielles pour l'installation et la configuration d'Elasticsearch. Assurez-vous de fournir des explications plus détaillées si nécessaire dans votre documentation complète.

1. Pour que elasticsearch soit accessible depuis le réseau lorsque nous sommes en lab pour vérifier que tout fonctionne il faut rajouter ces lignes dans le fichier de conf

```
#  
http.port: 9200  
cluster.name: my-cluster  
node.name: SRVISLOGPRD  
network.host: 0.0.0.0  
discovery.seed_hosts: ["172.16.5.37"]
```

#### **network.host: 0.0.0.0 :**

- Cette ligne indique à Elasticsearch d'écouter sur toutes les interfaces réseau disponibles sur la machine. L'adresse IP "0.0.0.0" signifie qu'Elasticsearch sera accessible via toutes les interfaces, y compris IPv4 et IPv6 si elles sont activées.

#### **discovery.seed\_hosts: ["172.16.5.37"] :**

- Cela configure les nœuds que ce nœud Elasticsearch utilisera comme points d'initialisation pour la découverte du cluster. Dans cet exemple, il spécifie que le nœud Elasticsearch doit essayer de se connecter à un nœud ayant l'adresse IP "172.16.5.37" pour découvrir d'autres nœuds dans le cluster. Les nœuds de découverte sont essentiels pour former un cluster Elasticsearch.

Lorsque via un navigateur j'accède à IP :9200 je vois que elasticsearch fonctionne correctement

```
1 {  
2   "name": "SRVISLOGPRD",  
3   "cluster_name": "my-cluster",  
4   "cluster_uuid": "a4P4RGCpSvu-uxO6nW6teQ",  
5   "version": {  
6     "number": "7.17.13",  
7     "build_flavor": "default",  
8     "build_type": "deb",  
9     "build_hash": "2b211dbb8bfdecaf7f5b44d356bdfe54b1050c13",  
10    "build_date": "2023-08-31T17:33:19.958690787Z",  
11    "build_snapshot": false,  
12    "lucene_version": "8.11.1",  
13    "minimum_wire_compatibility_version": "6.8.0",  
14    "minimum_index_compatibility_version": "6.0.0-beta1"  
15  },  
16   "tagline": "You Know, for Search"  
17 }
```

## Installation de kibana

sudo apt install kibana

Fichier de conf : /etc/kibana/kibana.yml

Dans le fichier de conf on veut vérifier que l'url d'elasticsearch est correct

```
# The URLs of the Elasticsearch instances to use for all your queries.  
#elasticsearch.hosts: ["http://localhost:9200"]
```

Elasticsearch est bien sur ma machine locale et il écoute sur le port 9200 en TCP

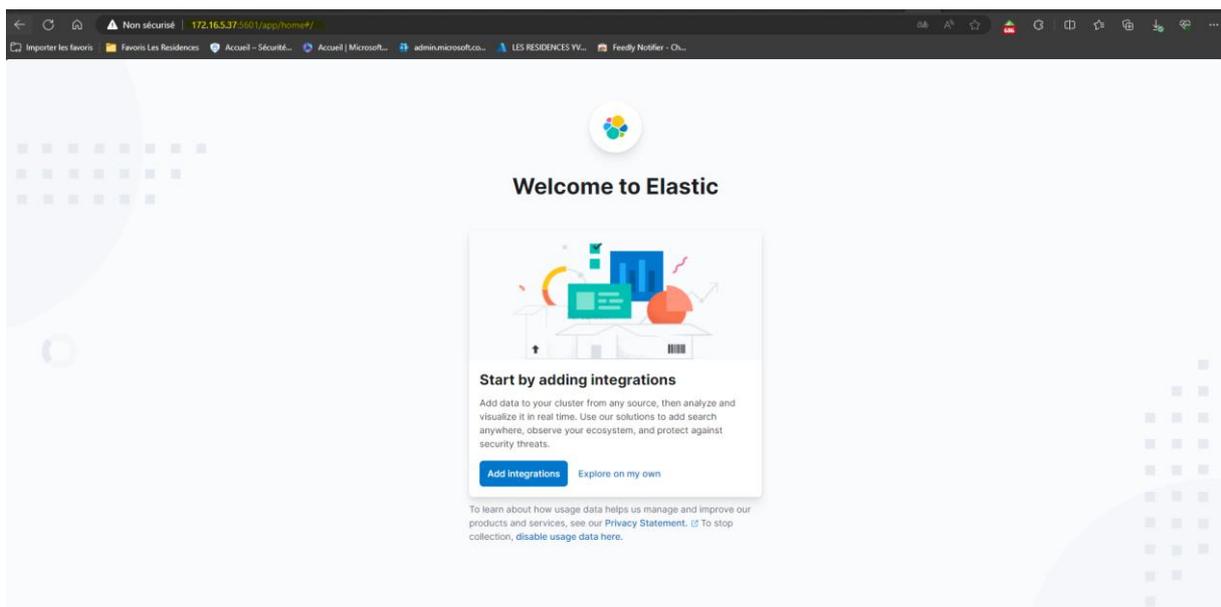
Par défaut il n'est accessible qu'en local je modifie cette directive pour qu'il écoute sur toute les interfaces

```
# 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
```

Pour que kibana se lance au démarrage du serveur tapez la commande :

sudo systemctl enable kibana

## Page web kibana



## Installation logstash

### Étape 1 : Installez Java.

```
sudo apt install default-jre
```

### Étape 2 : Installez Logstash.

```
sudo apt install logstash
```

### Étape 3 : Configurez Logstash (le fichier de configuration se trouve à /etc/logstash/logstash.yml).

### Étape 4 : Démarrez Logstash.

```
sudo systemctl start logstash
```

### Étape 5 : Pour que Logstash démarre automatiquement avec le système, utilisez la commande suivante :

```
sudo systemctl enable logstash
```

### Étape 6 : Dans le répertoire /etc/logstash/conf.d/, créez un fichier nommé syslog.conf pour configurer Logstash en fonction de vos besoins.

Cette version simplifiée résume les étapes essentielles pour l'installation et la configuration de Logstash. Vous pouvez fournir des détails plus spécifiques dans votre documentation complète si nécessaire.

```
[1/2] syslog.conf *
input {
  syslog {
    port => 5004
    type => syslog
  }
}
output {
  elasticsearch {
    hosts => ["localhost:9200"]
  }
}
```

## Étape 1 : Modifiez la configuration de rsyslog.

Ouvrez le fichier de configuration de rsyslog en éditant le fichier **/etc/rsyslog.conf** :

```
sudo nano /etc/rsyslog.conf
```

Ajoutez la ligne suivante pour rediriger les logs vers Logstash sur le port 5004 en localhost :

```
*.* @@localhost:5004
```

Sauvegardez et fermez le fichier.

## Étape 2 : Redémarrez le service rsyslog.

Après avoir modifié la configuration, redémarrez le service rsyslog pour appliquer les modifications :

```
sudo systemctl restart rsyslog
```

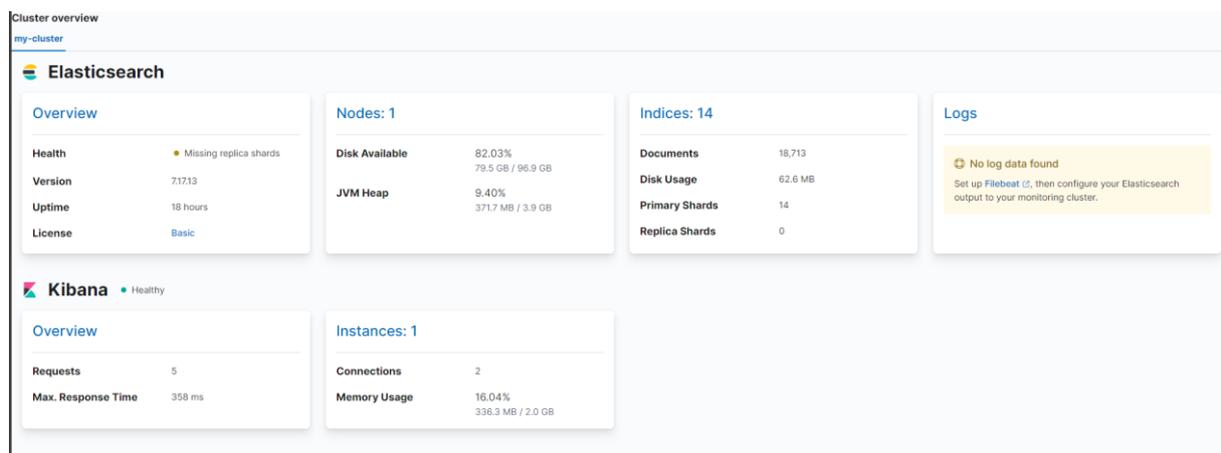
Cela redirigera tous les logs vers Logstash sur le port 5004 en localhost.

N'oubliez pas de configurer également Logstash pour écouter sur le port 5004 et traiter les logs entrants correctement. Assurez-vous que la configuration Logstash dans le fichier **/etc/logstash/conf.d/syslog.conf** (comme mentionné précédemment) correspond à ce que vous attendez en termes de traitement des logs reçus.

## Configuration Kibana

Repartir sur l'ip du serveur web port 5601

Je fais défiler le menu à gauche et je vais dans Management > stackmonitoring



The screenshot displays the Kibana Cluster Overview page for a cluster named 'my-cluster'. It is divided into two main sections: Elasticsearch and Kibana.

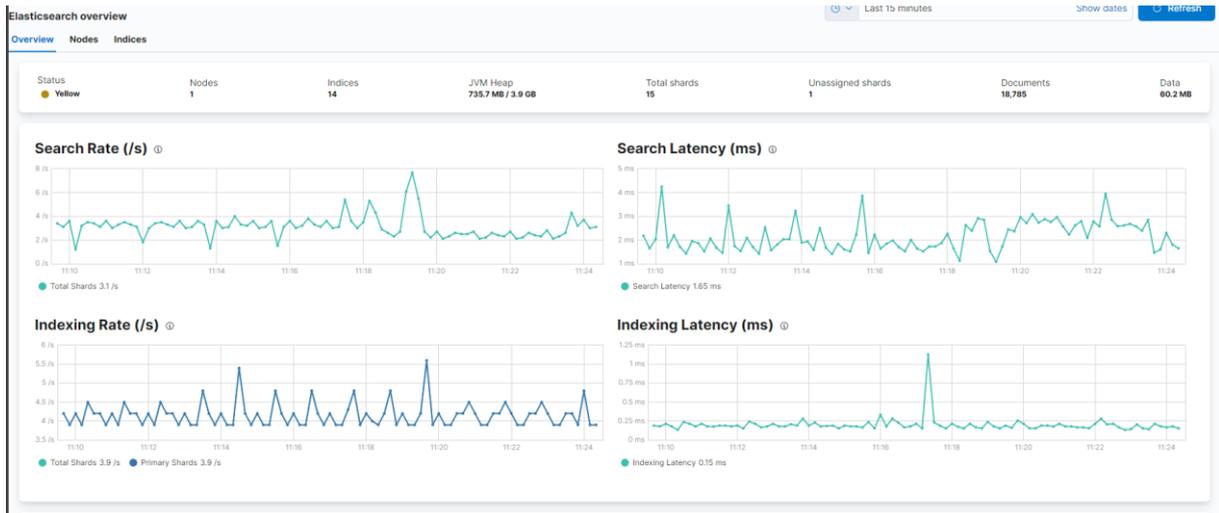
**Elasticsearch Overview:**

- Health:** Missing replica shards.
- Version:** 7.17.13
- Uptime:** 18 hours
- License:** Basic
- Nodes:** 1
- Disk Available:** 82.03% (79.5 GB / 96.9 GB)
- JVM Heap:** 9.40% (371.7 MB / 3.9 GB)
- Indices:** 14
- Documents:** 18,713
- Disk Usage:** 62.6 MB
- Primary Shards:** 14
- Replica Shards:** 0
- Logs:** No log data found. A message suggests setting up Filebeat to output logs to the monitoring cluster.

**Kibana Overview:**

- Health:** Healthy
- Requests:** 5
- Max. Response Time:** 358 ms
- Instances:** 1
- Connections:** 2
- Memory Usage:** 16.04% (336.3 MB / 2.0 GB)

Si j'appuie sur overview en dessous Elasticsearch cela me montre cela



Le nombre de log que je reçois si il y'a un pic d'activités ou non

J'ai fait un script qui redémarre le serveur 500 fois et je vais regarder si il y'a un pic à un moment

D'abord on peut dire à logstash de chercher les logs directement dans syslog comme ceci

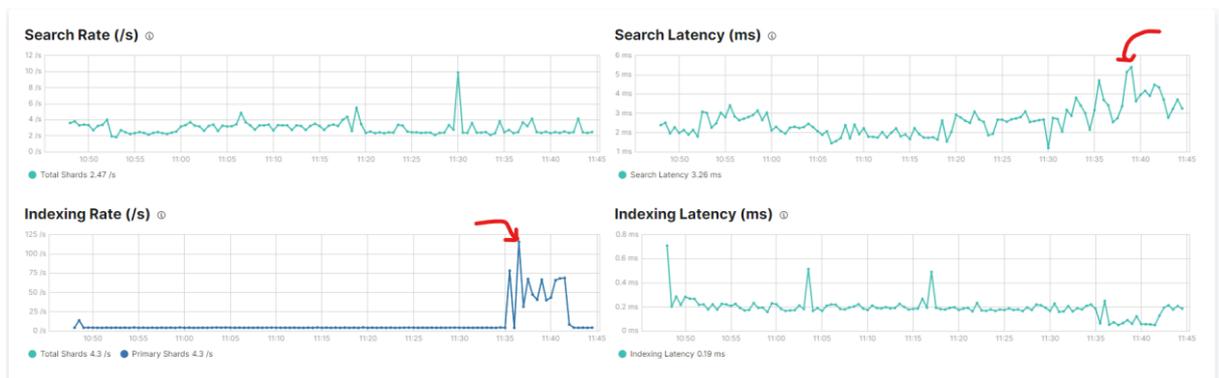
```

input {
  file {
    path => "/var/log/syslog"
    start_position => "beginning" # Vous pouvez également utiliser "end" pour lire le fichier à partir de la fin.
  }
}

output {
  elasticsearch {
    hosts => ["localhost:9200"]
  }
}

```

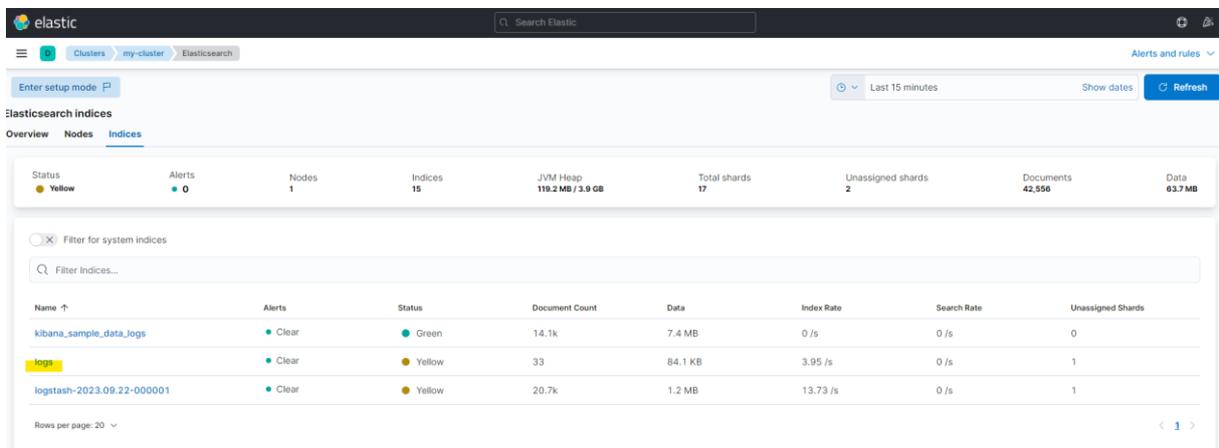
Le moment ou j'ai lancé le script vers 11h38



Je peux aussi séparer les logs dans logstash pour que ça soit beaucoup plus clair

```
input {
  file {
    path => "/var/log/syslog"
    start_position => "beginning" # Vous pouvez également utiliser "end" pour lire le fichier à partir de la fin.
  }
}

output {
  elasticsearch {
    hosts => ["localhost:9200"]
    index => "logs"
  }
}
```



The screenshot shows the Elastic Stack Management interface for the Elasticsearch Indices page. At the top, there's a search bar and navigation tabs for Clusters, my-cluster, and Elasticsearch. Below that, there's a filter for 'Last 15 minutes' and a 'Refresh' button. The main content area shows a summary of system indices with the following metrics:

Metric	Value
Status	Yellow
Alerts	0
Nodes	1
Indices	15
JVM Heap	119.2 MB / 3.9 GB
Total shards	17
Unassigned shards	2
Documents	42,556
Data	63.7 MB

Below the summary, there's a table of system indices with the following columns: Name, Alerts, Status, Document Count, Data, Index Rate, Search Rate, and Unassigned Shards.

Name	Alerts	Status	Document Count	Data	Index Rate	Search Rate	Unassigned Shards
kibana_sample_data_logs	Clear	Green	14.1k	7.4 MB	0 /s	0 /s	0
logs	Clear	Yellow	33	84.1 KB	3.95 /s	0 /s	1
logstash-2023.09.22-000001	Clear	Yellow	20.7k	1.2 MB	13.73 /s	0 /s	1

Maintenant pour visualiser ces logs dans kibana

Il faut aller dans stack management comme on a dit et aller dans Index Pattern

Ensuite sélectionner « Create Index Pattern »

Ensuite il faut taper le nom de l'index qu'on a définis dans logstash

## Create index pattern

Name

logs

An index pattern with this title already exists.

Use an asterisk (\*) to match multiple characters. Spaces and the characters /, ?, \*, <, >, | are not allowed.

Timestamp field

Select a timestamp field

Show advanced settings

✓ Your index pattern matches 1 source.

logs

Index

Rows per page: 10

Close

Create index pattern

Ensuite appuyer sur Create Index Pattern

Je me rends ensuite dans discover

En haut a gauche je selectionne la source dans notre cas « logs »

The screenshot shows the Elastic Discover interface. At the top, the 'elastic' logo and a search bar are visible. Below the search bar, there are navigation options like 'Discover' and 'Save'. The main area is divided into a left sidebar and a main content area. The sidebar shows the 'logs' index pattern selected. The main content area displays a histogram of hits over time, with a peak around 11:30:00. Below the histogram, a list of log entries is shown, each with a timestamp, host, and message. The messages include system logs for cron jobs and service restarts.

Time	Document
Sep 22, 2023 @ 12:01:02.329	<pre>@timestamp: Sep 22, 2023 @ 12:01:02.329 @version: 1 host: srvsyslogprd message: 2023-09-22T12:01:01+02:00 megnansi CROND[116639]: (root) CMD (run-parts /etc/cron.hourly) path: /var/log/syslog _id: yd0vI0ByJesng7EHh _index: logs _score: - _type: _doc</pre>
Sep 22, 2023 @ 12:01:02.329	<pre>@timestamp: Sep 22, 2023 @ 12:01:02.329 @version: 1 host: srvsyslogprd message: 2023-09-22T12:01:01+02:00 megnansi run-parts(/etc/cron.hourly)[1166] starting @anacron path: /var/log/syslog _id: yZ0vI0ByJesng7EHh _index: logs _score: - _type: _doc</pre>
Sep 22, 2023 @ 12:01:02.329	<pre>@timestamp: Sep 22, 2023 @ 12:01:02.329 @version: 1 host: srvsyslogprd message: 2023-09-22T12:01:01+02:00 megnansi run-parts(/etc/cron.hourly)[1166] finished @anacron path: /var/log/syslog _id: yZ0vI0ByJesng7EHh _index: logs _score: - _type: _doc</pre>
Sep 22, 2023 @ 12:01:02.329	<pre>@timestamp: Sep 22, 2023 @ 12:01:02.329 @version: 1 host: srvsyslogprd message: 2023-09-22T12:01:01+02:00 megnansi run-parts(/etc/cron.hourly)[1166] starting mceIog.cron path: /var/log/syslog _id: yS0vI0ByJesng7EHh _index: logs _score: - _type: _doc</pre>
Sep 22, 2023 @ 12:01:02.329	<pre>@timestamp: Sep 22, 2023 @ 12:01:02.329 @version: 1 host: srvsyslogprd message: 2023-09-22T12:01:01+02:00 megnansi run-parts(/etc/cron.hourly)[1166] finished mceIog.cron path: /var/log/syslog _id: zJ0vI0ByJesng7EHh _index: logs _score: - _type: _doc</pre>
Sep 22, 2023 @ 12:00:02.281	<pre>@timestamp: Sep 22, 2023 @ 12:00:02.281 @version: 1 host: srvsyslogprd message: 2023-09-22T12:00:01+02:00 megnansi CROND[116636]: (root) CMD (/usr/lib64/sa/sa1 1 1) path: /var/log/syslog _id: VZ0vI0ByJesngMAKFT _index: logs _score: - _type: _doc</pre>
Sep 22, 2023 @ 11:56:39.877	<pre>@timestamp: Sep 22, 2023 @ 11:56:39.877 @version: 1 host: srvsyslogprd message: 2023-09-22T11:56:39+02:00 megnansi rsyslogd: [origin software="rsyslogd" swVersion="5.8.10" x-pid="116623" x-info="http://www.rsyslog.com"] start path: /var/log/syslog _id: 1p0vI0ByJesngM0z-0 _index: logs _score: - _type: _doc</pre>

Résultat

