

# Atelier 1 : Installation et Configuration d'ElasticSearch & Kibana

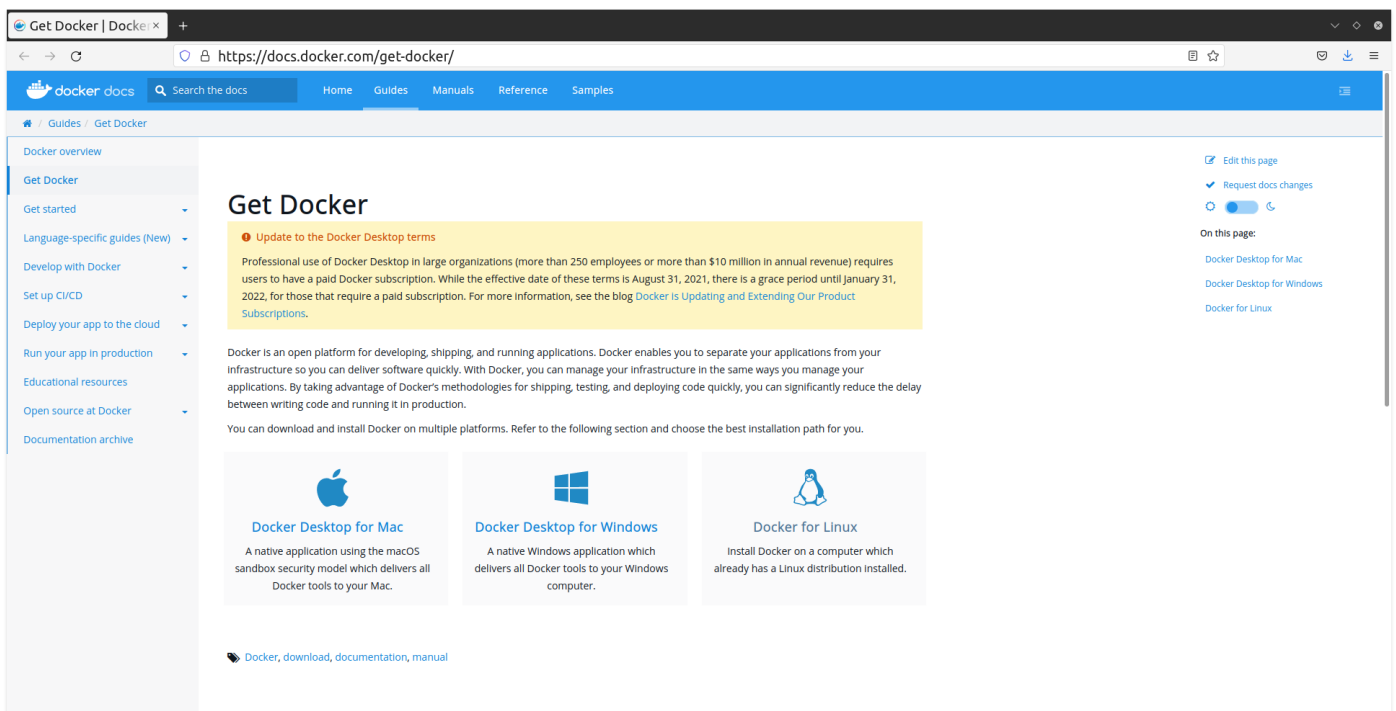
## Objectifs

Après avoir terminé cet atelier, vous serez en mesure de :

- Installer Docker
- Installer et tester ElasticSearch & Kibana sous Docker

## Installation de Docker

Docker est disponible pour Windows, Linux et MacOS : <https://docs.docker.com/get-docker/>



## Sous Debian (dont Ubuntu)

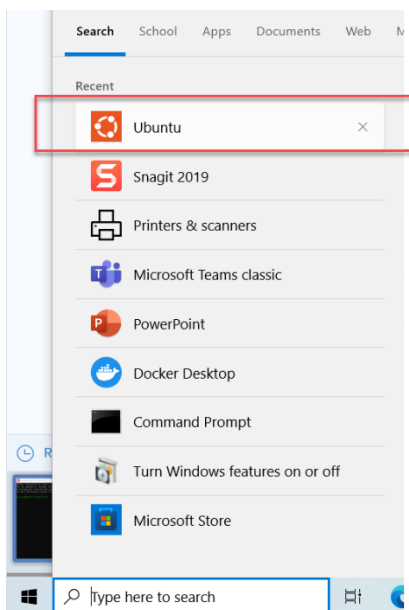
```

sudo apt-get update
sudo apt-get install ca-certificates curl gnupg lsb-release
curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o
/usr/share/keyrings/docker-archive-keyring.gpg
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-
keyring.gpg] https://download.docker.com/linux/debian \
  $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt-get update
sudo apt-get install docker-ce docker-ce-cli containerd.io
  
```

## Installation et Configuration Elasticsearch

### Lancer un terminal avec WSL ubuntu sous windows



### Téléchargement de l'image Elasticsearch

Nous spécifions la version 8.13.4 comme tag de l'image.

```
docker pull docker.elastic.co/elasticsearch/elasticsearch:8.13.4
```

### Créer et démarrer le container

Pour faciliter la communication entre les containers (elasticsearch, kibana, logstash, filebeat), nous allons utiliser un réseau Docker que nous appelons « elastic ».

```
docker network create elastic
```

C'est la commande docker run qui permettra de créer un container et l'exécuter.

```
docker run --name es01 -p 9200:9200 -p 9300:9300 \
-e "discovery.type=single-node" \
--net elastic docker.elastic.co/elasticsearch/elasticsearch:8.13.4
```

```
msellami@LAPTOP-5O60CAT1: ~
msellami@LAPTOP-5O60CAT1:~$ docker run --name es01 -p 9200:9200 -p 9300:9300 \
> -e "discovery.type=single-node" \
> --net elastic docker.elastic.co/elasticsearch/elasticsearch:8.13.4
```

Options détaillées :

- **--name es01 :**
  - Cette option permet de nommer le conteneur. Ici, le conteneur est nommé es01.
  - Exemple : Si vous avez plusieurs conteneurs Elasticsearch, leur donner des noms distincts vous aide à les identifier plus facilement.
- **--net elastic :**
  - Cette option connecte le conteneur à un réseau Docker personnalisé nommé elastic.
  - Les réseaux Docker permettent aux conteneurs de communiquer entre eux de manière isolée des autres réseaux.
- **-p 9200:9200 et -p 9300:9300 :**
  - Ces options publient des ports spécifiques du conteneur sur l'hôte.
  - **-p 9200:9200 :**
    - Mappe le port 9200 du conteneur au port 9200 de l'hôte.
    - Le port 9200 est utilisé pour les requêtes HTTP d'Elasticsearch.
  - **-p 9300:9300 :**
    - Mappe le port 9300 du conteneur au port 9300 de l'hôte.
    - Le port 9300 est utilisé pour la communication entre les nœuds du cluster Elasticsearch.
- **-e "discovery.type=single-node" :**
  - Cette option définit une variable d'environnement dans le conteneur.
  - **discovery.type=single-node :**
    - Configure Elasticsearch pour fonctionner en mode nœud unique.
    - Ce mode est idéal pour les environnements de développement et de test, car il désactive la découverte de nœuds supplémentaires, simplifiant ainsi la configuration.
- **-t :**
  - Cette option alloue un pseudo-terminal au conteneur, ce qui peut être utile pour l'interaction avec les applications qui nécessitent un terminal.
  - Cela permet également de voir les journaux du conteneur dans le terminal où la commande docker run a été exécutée.
- **docker.elastic.co/elasticsearch/elasticsearch:8.13.4 :**
  - Spécifie l'image Docker à utiliser pour créer le conteneur.
  - **docker.elastic.co/elasticsearch/elasticsearch :**
    - C'est le dépôt où l'image Elasticsearch est hébergée.
  - **:8.13.4 :**
    - Indique la version spécifique de l'image Elasticsearch à utiliser. Ici, la version 8.13.4.

Récupération des jetons d'inscriptions et du mot de passe

Copiez le mot de passe `elastic` généré et le jeton d'enrôlement (enrollment token) qui sont affichés dans votre terminal. Vous les utiliserez pour enregistrer Kibana avec votre cluster Elasticsearch et vous connecter. Ces identifiants ne sont affichés que lors du premier démarrage d'Elasticsearch.

```
{
  "timestamp": "2024-06-02T10:57:18.447Z",
  "log.level": "INFO",
  "current.health": "GREEN",
  "message": "Cluster health status changed from [YELLOW] to [GREEN] (reason: [shards started [[.security-7][0]]]).",
  "previous.health": "YELLOW",
  "reason": "shards started [[.security-7][0]]",
  "ecs.version": "1.2.0",
  "service.name": "ES_ECS",
  "event.dataset": "elasticsearch.server",
  "process.thread.name": "elasticsearch[50294cbe0db4][masterServiceUpdateTask][T#1]",
  "log.logger": "org.elasticsearch.cluster.routing.allocation.AllocationService",
  "elasticsearch.cluster.uid": "cn3mMEloTHec9iQNKgFamA",
  "elasticsearch.node.id": "MPYY5wxrRKKccVWw4BPqQ",
  "elasticsearch.node.name": "50294cbe0db4",
  "elasticsearch.cluster.name": "docker-cluster"
}
```

```
Authentication is enabled and cluster connections are encrypted.
Elasticsearch security features have been automatically configured!

Password for the elastic user (reset with `bin/elasticsearch-reset-password -u elastic`):
ch*IqL292zftZVizrzPR

HTTP CA certificate SHA-256 fingerprint:
b6fbc93044a9dbe1e56cd8d3739bb86c9bb8de516410a087efd0b1b4a83a6a38

Configure Kibana to use this cluster:
• Run Kibana and click the configuration link in the terminal when Kibana starts.
• Copy the following enrollment token and paste it into Kibana in your browser (valid for the next 30 minutes):
eyJ2ZXI0i0iI4LjEzLjQlCjZHI0IsMTcyLjE4LjAuMj05MjAwI0sImZnciI6ImI2ZmJjOTMwNDRhOWRiZTFINTZjZDhkMzczOWJiODZjOWJiOGRlbnRlbnQ9MTg4M19nIn0=

Configure other nodes to join this cluster:
• Copy the following enrollment token and start new Elasticsearch nodes with `bin/elasticsearch --enrollment-token <token>` (valid for the next 30 minutes):
eyJ2ZXI0i0iI4LjEzLjQlCjZHI0IsMTcyLjE4LjAuMj05MjAwI0sImZnciI6ImI2ZmJjOTMwNDRhOWRiZTFINTZjZDhkMzczOWJiODZjOWJiOGRlbnRlbnQ9MTg4M19nIn0=

If you're running in Docker, copy the enrollment token and run:
`docker run -e "ENROLLMENT_TOKEN=<token>" docker.elastic.co/elasticsearch/elasticsearch:8.13.4`
```

Lancer un autre Shell Ubuntu et créer un fichier `keys.text` et copier les paramètres de sécurité d'Elasticsearch

```
mkdir elastic
cd elastic nano && keys.txt
```

Taper CTRL+X puis Y et Entrer pour enregistrer le fichiers

```
msellami@LAPTOP-5060CAT1: ~/elastic
GNU nano 6.2 keys.txt

Authentication is enabled and cluster connections are encrypted.

Password for the elastic user (reset with `bin/elasticsearch-reset-password -u elastic`):
ch*IqL292zftZVizrzPR

HTTP CA certificate SHA-256 fingerprint:
b6fbc93044a9dbe1e56cd8d3739bb86c9bb8de516410a087efd0b1b4a83a6a38

Configure Kibana to use this cluster:
• Run Kibana and click the configuration link in the terminal when Kibana starts.
• Copy the following enrollment token and paste it into Kibana in your browser (valid for the next 30 minutes):
eyJ2ZXI0i0iI4LjEzLjQlCjZHI0IsMTcyLjE4LjAuMj05MjAwI0sImZnciI6ImI2ZmJjOTMwNDRhOWRiZTFINTZjZDhkMzczOWJiODZjOWJiOGRlbnRlbnQ9MTg4M19nIn0=

Configure other nodes to join this cluster:
• Copy the following enrollment token and start new Elasticsearch nodes with `bin/elasticsearch --enrollment-token <token>` (valid for the next 30 minutes):
eyJ2ZXI0i0iI4LjEzLjQlCjZHI0IsMTcyLjE4LjAuMj05MjAwI0sImZnciI6ImI2ZmJjOTMwNDRhOWRiZTFINTZjZDhkMzczOWJiODZjOWJiOGRlbnRlbnQ9MTg4M19nIn0=

If you're running in Docker, copy the enrollment token and run:
`docker run -e "ENROLLMENT_TOKEN=<token>" docker.elastic.co/elasticsearch/elasticsearch:8.13.4`

[ Read 24 lines ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo     M-A Set Mark
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line M-E Redo     M-6 Copy
```



Nous vous recommandons de stocker le mot de passe `elastic` en tant que variable d'environnement dans votre shell.

```

Elasticsearch security features have been automatically configured!
Authentication is enabled and cluster connections are encrypted.
Password for the elastic user (reset with `bin/elasticsearch-reset-password -u elastic`):
ch*IqL292zftZVizrzPR
HTTP CA certificate SHA-256 fingerprint:

```

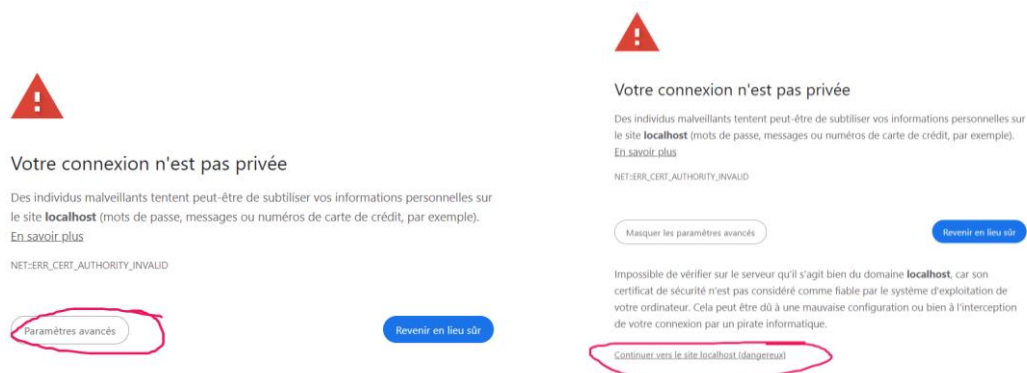
```
export ELASTIC_PASSWORD="ch*IqL292zftZVizrzPR"
```

```

msellami@LAPTOP-5060CAT1: ~
msellami@LAPTOP-5060CAT1:~$ export ELASTIC_PASSWORD="ch*IqL292zftZVizrzPR"

```

À partir d'un navigateur Web (Firefox, Chrome...), accédez à l'URL suivante : <https://localhost:9200>. Et cliquer sur Paramètres avancés et



Taper les paramètres de connexions :

**Se connecter**

<https://localhost:9200>

Nom d'utilisateur

Mot de passe

Mot de passe récupéré du  
Shell

Le résultat obtenu doit être semblable à cet aperçu :

```

  ← → ↻ × Non sécurisé https://localhost:9200
  Impression élégante ☒
  {
    "name": "50294cbe0db4",
    "cluster_name": "docker-cluster",
    "cluster_uuid": "cn3mMEloTHec9iQNKgfAmA",
    "version": {
      "number": "8.13.4",
      "build_flavor": "default",
      "build_type": "docker",
      "build_hash": "da95df118650b55a500dcc181889ac35c6d8da7c",
      "build_date": "2024-05-06T22:04:45.107454559Z",
      "build_snapshot": false,
      "lucene_version": "9.10.0",
      "minimum_wire_compatibility_version": "7.17.0",
      "minimum_index_compatibility_version": "7.0.0"
    },
    "tagline": "You Know, for Search"
  }

```

```

{
  "name": "50294cbe0db4",
  "cluster_name": "docker-cluster",
  "cluster_uuid": "cn3mMEloTHec9iQNKgfAmA",
  "version": {
    "number": "8.13.4",
    "build_flavor": "default",
    "build_type": "docker",
    "build_hash": "da95df118650b55a500dcc181889ac35c6d8da7c",
    "build_date": "2024-05-06T22:04:45.107454559Z",
    "build_snapshot": false,
    "lucene_version": "9.10.0",
    "minimum_wire_compatibility_version": "7.17.0",
    "minimum_index_compatibility_version": "7.0.0"
  },
  "tagline": "You Know, for Search"
}

```

Copier le certificat SSL `http_ca.crt` du conteneur vers votre machine locale

```
docker cp es01:/usr/share/elasticsearch/config/certs/http_ca.crt .
```

```
msellami@LAPTOP-5060CAT1:~/elastic$ docker cp es01:/usr/share/elasticsearch/config/certs/http_ca.crt .
Successfully copied 3.58kB to /home/msellami/elastic/.
msellami@LAPTOP-5060CAT1:~/elastic$
```

Faire un appel à l'API REST d'Elasticsearch pour s'assurer que le conteneur Elasticsearch est en cours d'exécution

```
curl --cacert http_ca.crt -u elastic:$ELASTIC_PASSWORD https://localhost:9200
```

```
msellami@LAPTOP-5060CAT1:~/elastic$ docker cp es01:/usr/share/elasticsearch/config/certs/http_ca.crt .
Successfully copied 3.58kB to /home/msellami/elastic/.
msellami@LAPTOP-5060CAT1:~/elastic$ curl --cacert http_ca.crt -u elastic:$ELASTIC_PASSWORD https://localhost:9200
{
  "name" : "50294cbe0db4",
  "cluster_name" : "docker-cluster",
  "cluster_uuid" : "cn3mMEloTHec9iQNKgfAmA",
  "version" : {
    "number" : "8.13.4",
    "build_flavor" : "default",
    "build_type" : "docker",
    "build_hash" : "da95df118650b55a500dcc181889ac35c6d8da7c",
    "build_date" : "2024-05-06T22:04:45.107454559Z",
    "build_snapshot" : false,
    "lucene_version" : "9.10.0",
    "minimum_wire_compatibility_version" : "7.17.0",
    "minimum_index_compatibility_version" : "7.0.0"
  },
  "tagline" : "You Know, for Search"
}
```

## Démarrage/arrêt container Elasticsearch

Une fois le container est créé, il n'est plus nécessaire de taper la commande « docker run » pour l'exécuter.

Pour arrêter Elasticsearch :

```
docker stop es01
```

Pour le démarrer de nouveau

```
docker start es02
```

## Installation et Configuration de Kibana

### Téléchargement de l'images Kibana

Nous spécifions la version 7.15.2 comme tag de l'image.

```
docker pull docker.elastic.co/kibana/kibana:8.13.4
```

### Créer et démarrer le container

Nous utilisons le même réseau Docker créé dans l'étape précédente.

```
docker run -p 5601:5601 -e "ELASTICSEARCH_HOSTS=http://es:9200" --net elastic \
--name kib docker.elastic.co/kibana/kibana:8.13.4
```

- L'option -e permet d'ajouter les variables d'environnement pour indiquer l'URL du serveur Elasticsearch créé précédemment.



```
msellami@LAPTOP-5O60CAT1: ~/elastic
{"minimum_index_compatibility_version" : "7.0.0"
},
"tagline" : "You Know, for Search"
}
msellami@LAPTOP-5O60CAT1:~/elastic$ docker run --name kibana --net elastic -p 5601:5601 docker.elastic.co/kibana/kibana:8.13.4

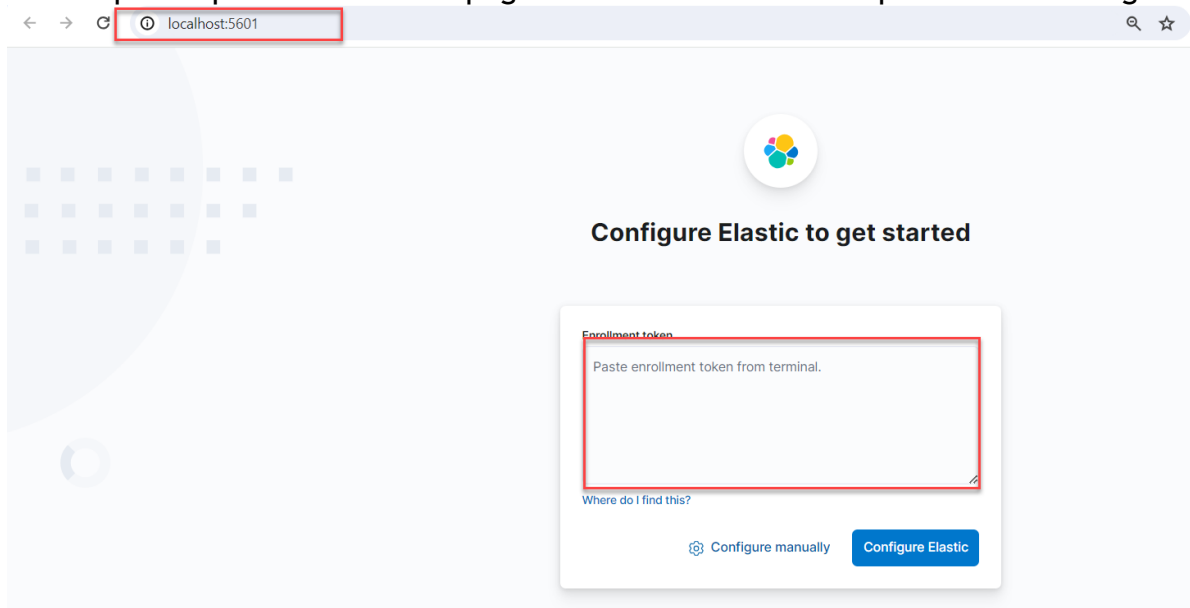
Kibana is currently running with legacy OpenSSL providers enabled! For details and instructions on how to disable see https://www.elastic.co/guide/en/kibana/8.13/production.html#openssl-legacy-provider
{"log.level":"info","@timestamp":"2024-06-02T11:26:42.413Z","log.logger":"elastic-apm-node","ecs.version":"8.10.0","agentVersion":"4.4.0","env":{"pid":7,"proctitle":"/usr/share/kibana/bin/./node/bin/node","os":"linux 5.10.102.1-microsoft-standard-WSL2","arch":"x64","host":"9f261264fe98","timezone":"UTC+00","runtime":"Node.js v20.12.2"},"config":{"active":{"source":"start","value":true},"breakdownMetrics":{"source":"start","value":false},"captureBody":{"source":"start","value":"off"},"commonName":"capture_body","captureHeaders":{"source":"start","value":false},"centralConfig":{"source":"start","value":false},"contextPropagationOnly":{"source":"start","value":true},"environment":{"source":"start","value":"production"},"globalLabels":{"source":"start","value":["git_rev","f5dc24d1969f80e4aa3ced7cc375dd00554f8c0c"],"sourceValue":{"git_rev":"f5dc24d1969f80e4aa3ced7cc375dd00554f8c0c"},"logLevel":{"source":"default","value":"info","commonName":"log_level"},"metricsInterval":{"source":"start","value":120,"sourceValue":"120s"},"serverUrl":{"source":"start","value":"https://kibana-cloud-apm.apm.us-east-1.aws.found.io/","commonName":"server_url"},"transactionSampleRate":{"source":"start","value":0.1,"commonName":"transaction_sample_rate"},"captureSpanStackTraces":{"source":"start","sourceValue":false},"secretToken":{"source":"start","value":"[REDACTED]","commonName":"secret_token"},"serviceName":{"source":"start","value":"kibana","commonName":"service_name"},"serviceVersion":{"source":"start","value":"8.13.4","commonName":"service_version"},"activationMethod":"require","message":"Elastic APM Node.js Agent v4.4.0"}
Native global console methods have been overridden in production environment.
[2024-06-02T11:26:43.739+00:00][INFO ][root] Kibana is starting
[2024-06-02T11:26:43.826+00:00][INFO ][node] Kibana process configured with roles: [background_tasks, ui]
```

Lorsque vous démarrez Kibana, une URL unique est affichée dans votre terminal. Pour accéder à Kibana :

1. Ouvrez l'URL générée dans votre navigateur.

L'accès à Kibana est par l'URL : <http://localhost:5601>

La capture qui suit montre la page d'accueil de Kibana au premier démarrage :





2. Collez le jeton d'enrôlement que vous avez copié plus tôt pour connecter votre instance Kibana à Elasticsearch.

```
msellami@LAPTOP-5060CAT1: ~/elastic
GNU nano 6.2 keys.txt

Authentication is enabled and cluster connections are encrypted.

Password for the elastic user (reset with `bin/elasticsearch-reset-password -u elastic`):
ch*IqL292zftZVizrzPR

HTTP CA certificate SHA-256 fingerprint:
b6fbc93044a9dbe1e56cd8d3739bb86c9bb8de516410a087efd0b1b4a83a6a38

Configure Kibana to use this cluster:
• Run Kibana and click the configuration link in the terminal when Kibana starts.
• Copy the following enrollment token and paste it into Kibana in your browser (valid for the next 30 minutes):
eyJ2ZXIiOiI4LjEzLjQlLCJhZHII0lsMTcyLjE4LjAuMj05MjAwIl0sImZnciI6ImI2ZmJjOTMwNDRhOWRiZTF1NTZjZDhkMzczOWJiODZjOWJiOGRIN
Configure other nodes to join this cluster:
• Copy the following enrollment token and start new Elasticsearch nodes with `bin/elasticsearch --enrollment-token <token>
eyJ2ZXIiOiI4LjEzLjQlLCJhZHII0lsMTcyLjE4LjAuMj05MjAwIl0sImZnciI6ImI2ZmJjOTMwNDRhOWRiZTF1NTZjZDhkMzczOWJiODZjOWJiOGRIN

If you're running in Docker, copy the enrollment token and run:
`docker run -e "ENROLLMENT_TOKEN=<token>" docker.elastic.co/elasticsearch/elasticsearch:8.13.4`

[ Read 24 lines ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line M-E Redo      M-6 Copy
```


Copier ce jeton et cliquer « Configure Elastic »

## Configure Elastic to get started

Enrollment token

```
eyJ2ZXliOiI4LjEzLjQiLCJhZHliOiI0MTcyLjE4LjAuMjo5MjAwIl0sImZncil6ImI2ZmJjOTMwNDRhOWRiZTFINTjZjZDhkMzczOWJiODZjOWJiOGRINTE2NDEwYTA4N2VmZDBiMWI0YTgzYTZhMzgiLCJrZXkiOiJGZVdZMkk4QmoxUGZkd2s0dG5xQzptOU1NbEV0LVJQT3NVUE5wWWw0eHZRIn0=
```

Connect to <https://172.18.0.2:9200>

 [Configure manually](#)

[Configure Elastic](#)

## Configure Elastic to get started

### Enrollment token

```
eyJ2ZXIiOiI4LjEzLjQlLCJhZHIIOiI0SiMTcyLjE4LjAuMjo5MjAwIiwsc2VudCI6ImI2ZmJjOTMwNDRhOWRiZTFINTzZDhkMzczOWJiODZjOWJiOGRINTE2NDEwYTA4N2VmZDBiMWI0YTgzYTZhMzgiLCJrZXkiOiJGZWdZMkk4QmoxUGZkd2s0dG5xQzptOU1NbEV0LVJQT3NVUE5wWWw0eHZRIn0=
```

Connect to <https://172.18.0.2:9200>

 Configure manually

**Configure Elastic**

Il demande un code qui sera généré dans le Shell associé au lancement du Kibana

## Configure Elastic to get started



### Verification required

Copy the code from the Kibana server or run `bin\kibana-verification-code.bat` to retrieve it.

**Verify**

## Copier le code

```
msellami@LAPTOP-5O60CAT1: ~/elastic
etToken":{"source":"start","value":"[REDACTED]","commonName":"secret_token"},"serviceName":{"source":"start","value":"ki
bana","commonName":"service_name"},"serviceVersion":{"source":"start","value":"8.13.4","commonName":"service_version"}},
"activationMethod":"require","message":"Elastic APM Node.js Agent v4.4.0"}
Native global console methods have been overridden in production environment.
[2024-06-02T11:26:43.739+00:00][INFO ][root] Kibana is starting
[2024-06-02T11:26:43.826+00:00][INFO ][node] Kibana process configured with roles: [background_tasks, ui]
[2024-06-02T11:26:51.203+00:00][INFO ][plugins-service] The following plugins are disabled: "cloudChat,cloudExperiments,
cloudFullStory,profilingDataAccess,profiling,securitySolutionServerless,serverless,serverlessObservability,serverlessSea
rch".
[2024-06-02T11:26:51.281+00:00][INFO ][http.server.Preboot] http server running at http://0.0.0.0:5601
[2024-06-02T11:26:51.398+00:00][INFO ][plugins-system.preboot] Setting up [1] plugins: [interactiveSetup]
[2024-06-02T11:26:51.410+00:00][INFO ][preboot] "interactiveSetup" plugin is holding setup: Validating Elasticsearch con
nection configuration...
[2024-06-02T11:26:51.437+00:00][INFO ][root] Holding setup until preboot stage is completed.


i Kibana has not been configured.

Go to http://0.0.0.0:5601/?code=124738 to get started.

Your verification code is: 124 738
```

## Copier et cliquer Vérifier

## Configure Elastic to get started



### Verification required

Copy the code from the Kibana server or run `bin\kibana-verification-code.bat` to retrieve it.

1

2

4

7

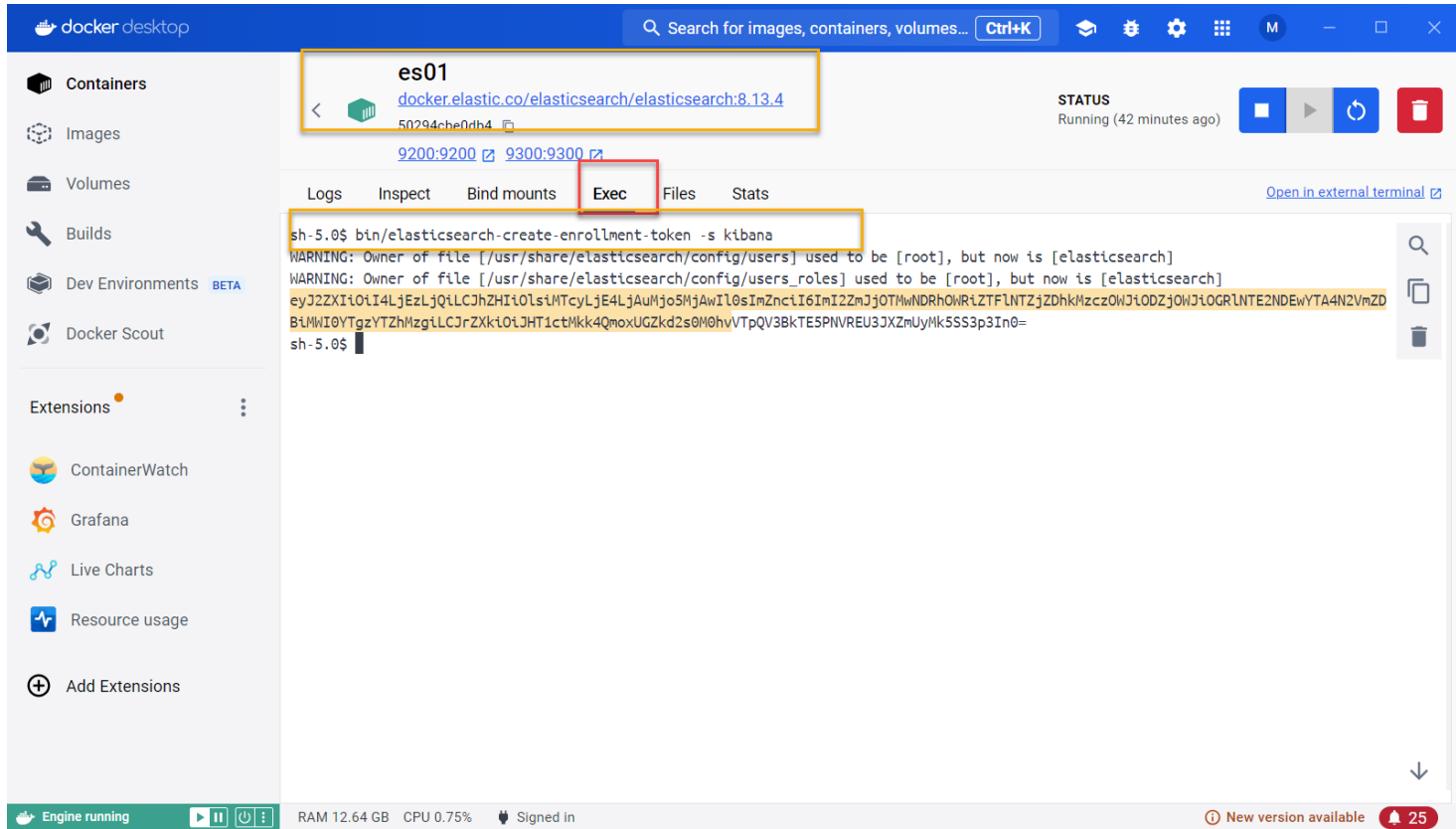
3

8

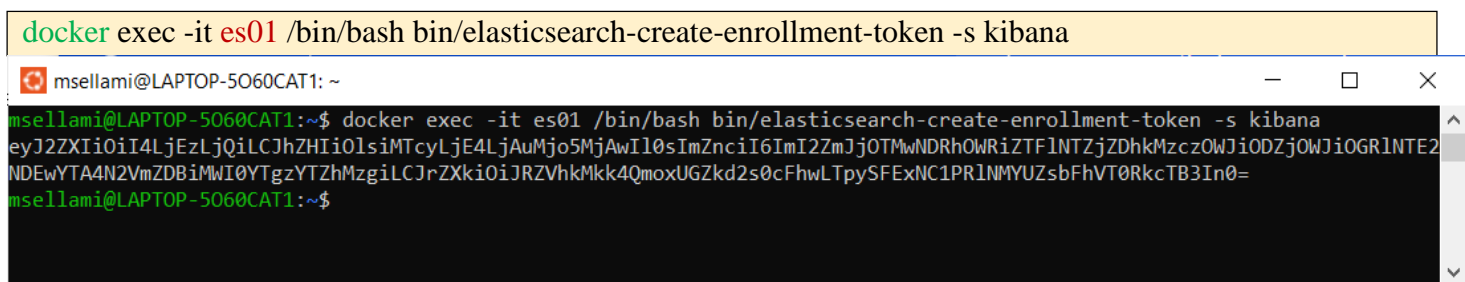
Verify

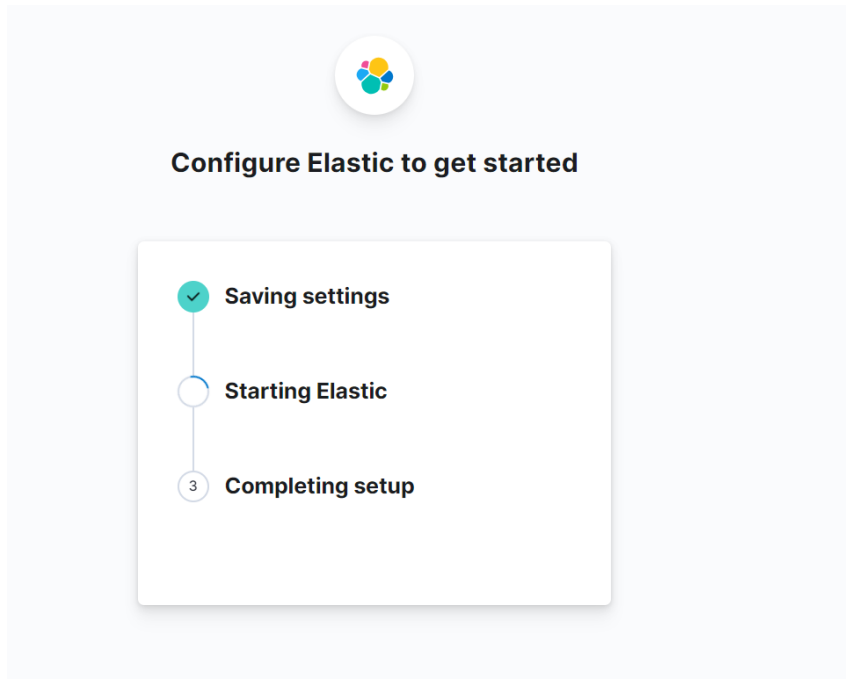
En cas d'échec ou dépassement de temps 30mn, il faut générer un nouveau token ;

Lancer un Shell via « exec » avec Docker desktop ou via un Shell Ubuntu

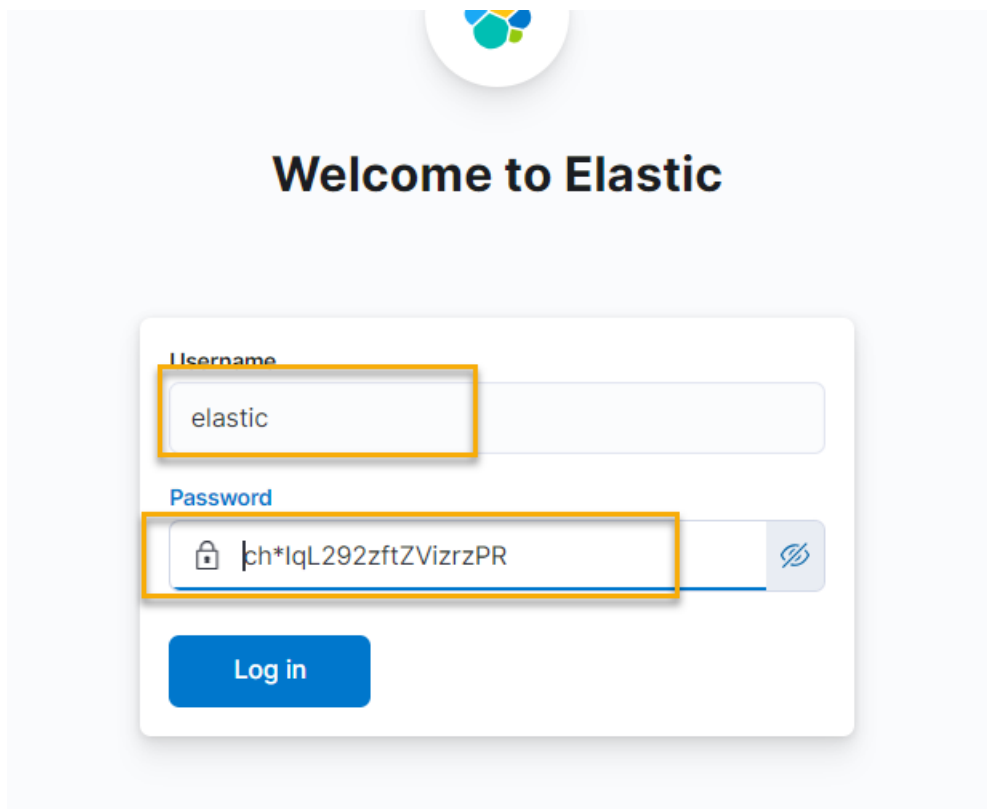


Ou via la commande suivante via Ubuntu Shell





3. Connectez-vous à Kibana en tant qu'utilisateur `elastic` avec le mot de passe généré lors du démarrage d'Elasticsearch.





Formateur : SELLAMI MOKHTAR  
[mokhtar.sellami@gmail.com](mailto:mokhtar.sellami@gmail.com)



localhost:5601/app/home#/

elastic Find apps, content, and more.

Home

Help us improve the Elastic Stack

Usage collection is enabled. This allows us to learn what our users are most interested in, so we can improve our products and services. Refer to our [Privacy Statement](#). Disable usage collection.

Dismiss

## Welcome home

### Search

Create search experiences with a refined set of APIs and tools.

### Observability

Consolidate your logs, metrics, application traces, and system availability with purpose-built UIs.

### Security

Prevent, collect, detect, and respond to threats for unified protection across your infrastructure.

### Analytics

Explore, visualize, and analyze data using a powerful suite of analytical tools and applications.

## Get started by adding integrations

To start working with your data, use one of our many ingest options. Collect data from an app or service, or upload a file. If you're not ready to use your own data, play with a sample data set.

[Add integrations](#)
[Try sample data](#)
[Upload a file](#)

### Try managed Elastic

Deploy, scale, and upgrade your stack faster with Elastic Cloud help you quickly move your data.

[Move to Elastic Cloud](#)

## Démarrage/arrêt container Kibana

Pour arrêter Kibana :

**docker stop kibana**

docker desktop Search for images, containers, volumes... Ctrl+K

Containers Give feedback

Container CPU usage 113.89% / 1600% (16 CPUs available)

Container memory usage 7.72GB / 11.94GB

Show charts

Search Only show running containers Delete Play Pause Stop

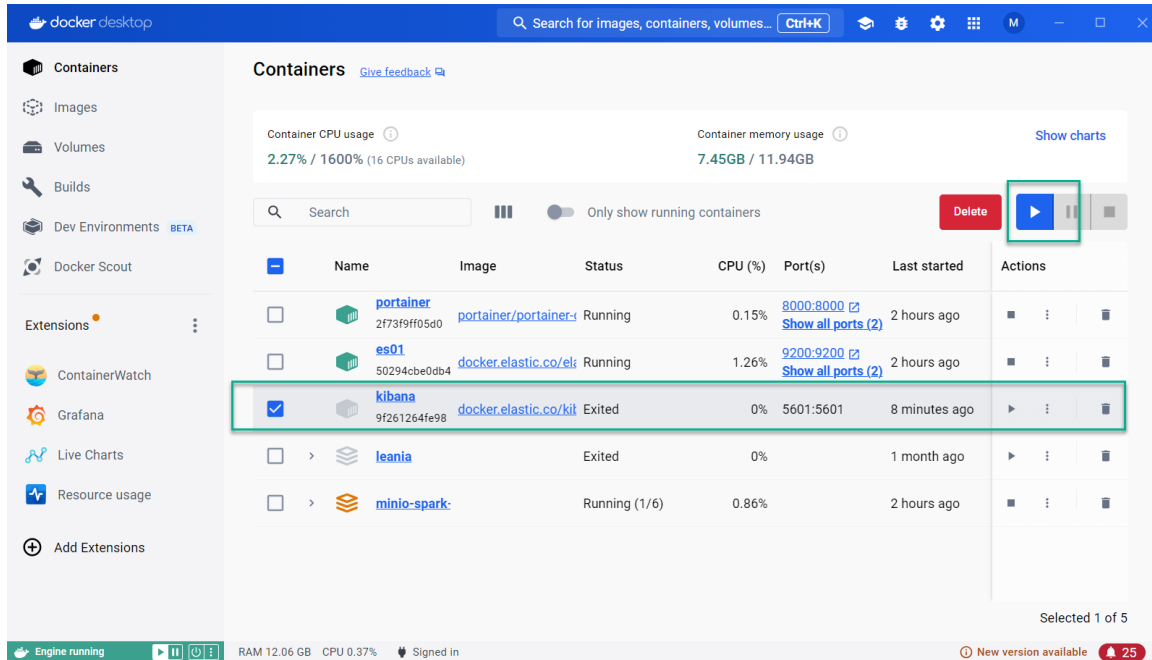
	Name	Image	Status	CPU (%)	Port(s)	Last started	Actions
<input type="checkbox"/>	portainer	portainer/portainer	Running	0.04%	8000:8000	2 hours ago	⋮
<input type="checkbox"/>	es01	docker.elastic.co/elasticsearch/elasticsearch	Running	1.2%	9200:9200	2 hours ago	⋮
<input checked="" type="checkbox"/>	kibana	docker.elastic.co/kibana/kibana	Running	112.16%	5601:5601	8 seconds ago	⋮
<input type="checkbox"/>	leania		Exited	0%		1 month ago	⋮
<input type="checkbox"/>	minio-spark		Running (1/6)	0.49%		2 hours ago	⋮

Selected 1 of 5

Engine running RAM 12.30 GB CPU 9.36% Signed in New version available 25

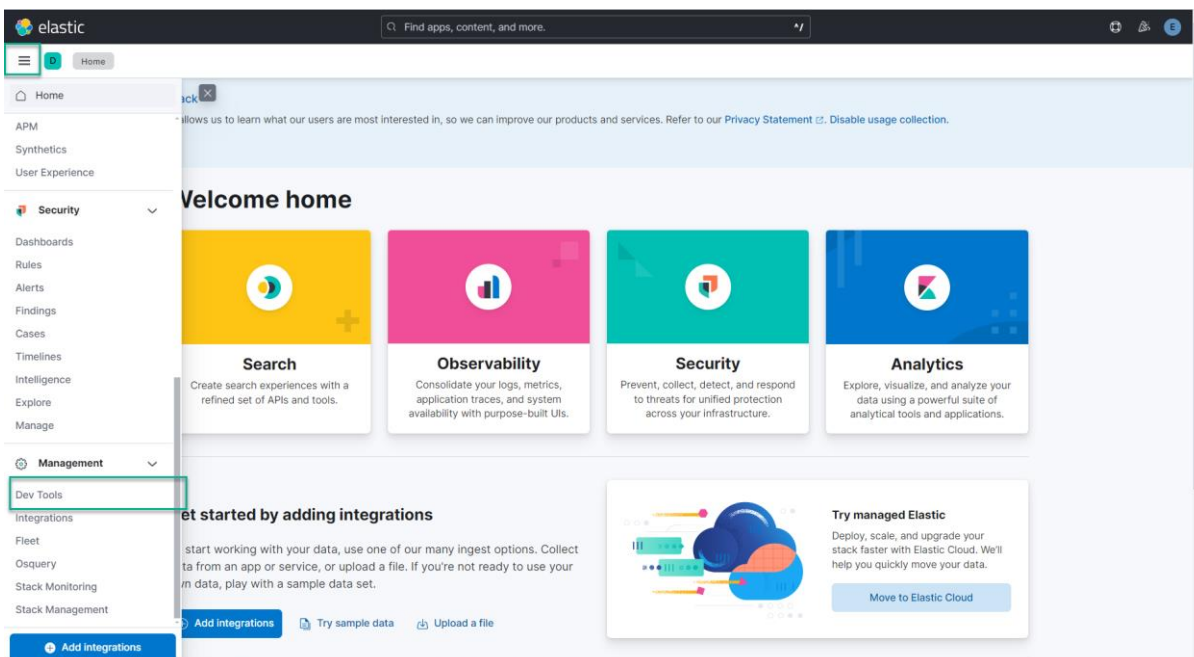
Pour le démarrer de nouveau

**docker start kibana**



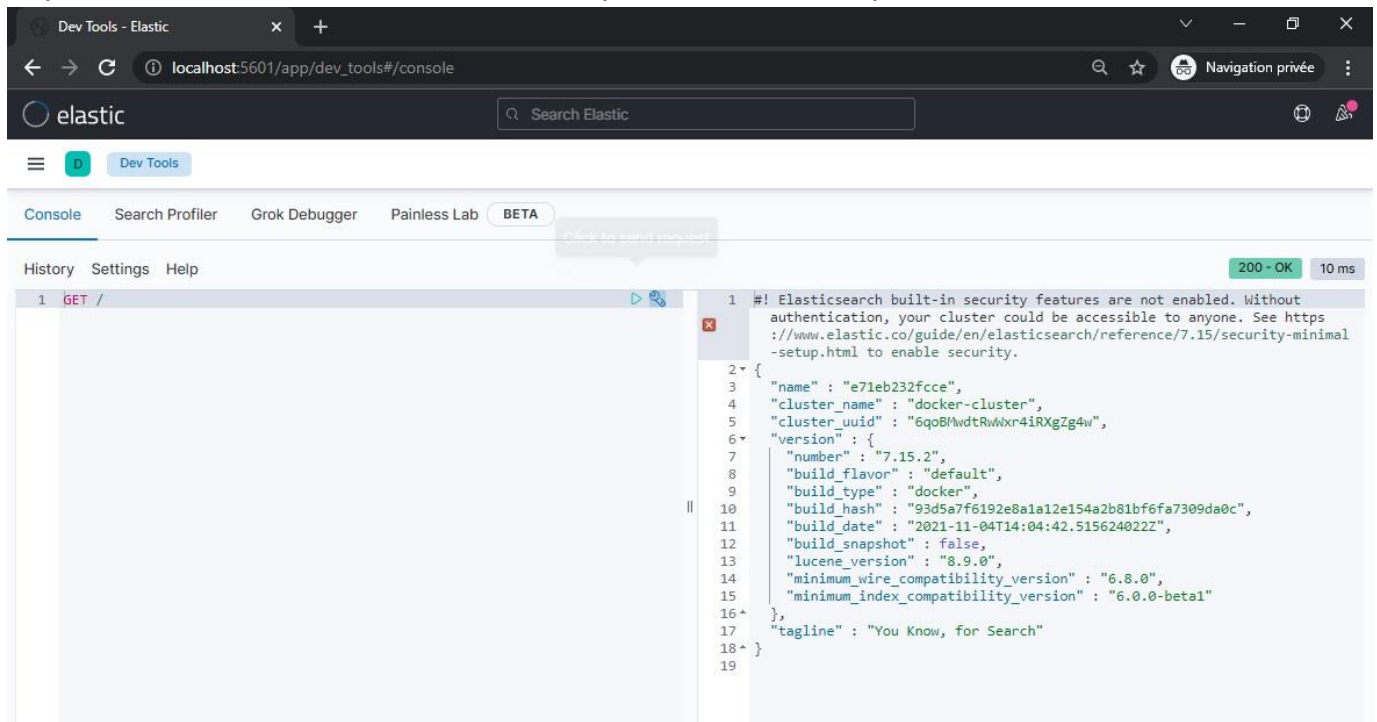
## Kibana Dev Tools

Kibana dispose d'un outil permettant de saisir également des requêtes à l'API REST. Pour y accéder, cliquez sur « **Explore on my own** », puis, à partir du menu à icône en hamburger, allez à « **Dev Tools** » sous la rubrique « **Management** ». C'est l'outil que nous recommandons pour la suite des ateliers.





À partir de la console, saisissez la requête : « **GET /** » pour avoir le statut Elasticsearch :



## Utilisation de cURL ou Postman

Le principal moyen d'interagir avec Elasticsearch consiste à utiliser l'API REST sur HTTP. Si Kibana ou Sense n'est pas une option pour vous, vous pouvez utiliser n'importe lequel des clients HTTP populaires, tels que **cURL** ou **Postman**.

Installation Postman : <https://www.postman.com/downloads/>



## Download Postman

Download the app to get started using the Postman API Platform today. Or, if you prefer a browser experience, you can try the web version of Postman.

### The Postman app

Download the app to get started with the Postman API Platform.

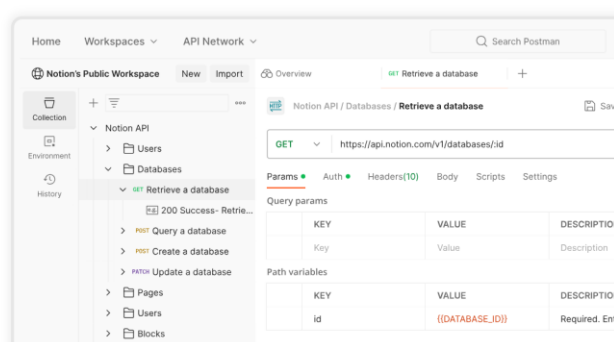
**Windows 64-bit**

By downloading and using Postman, I agree to the [Privacy Policy](#) and [Terms](#).

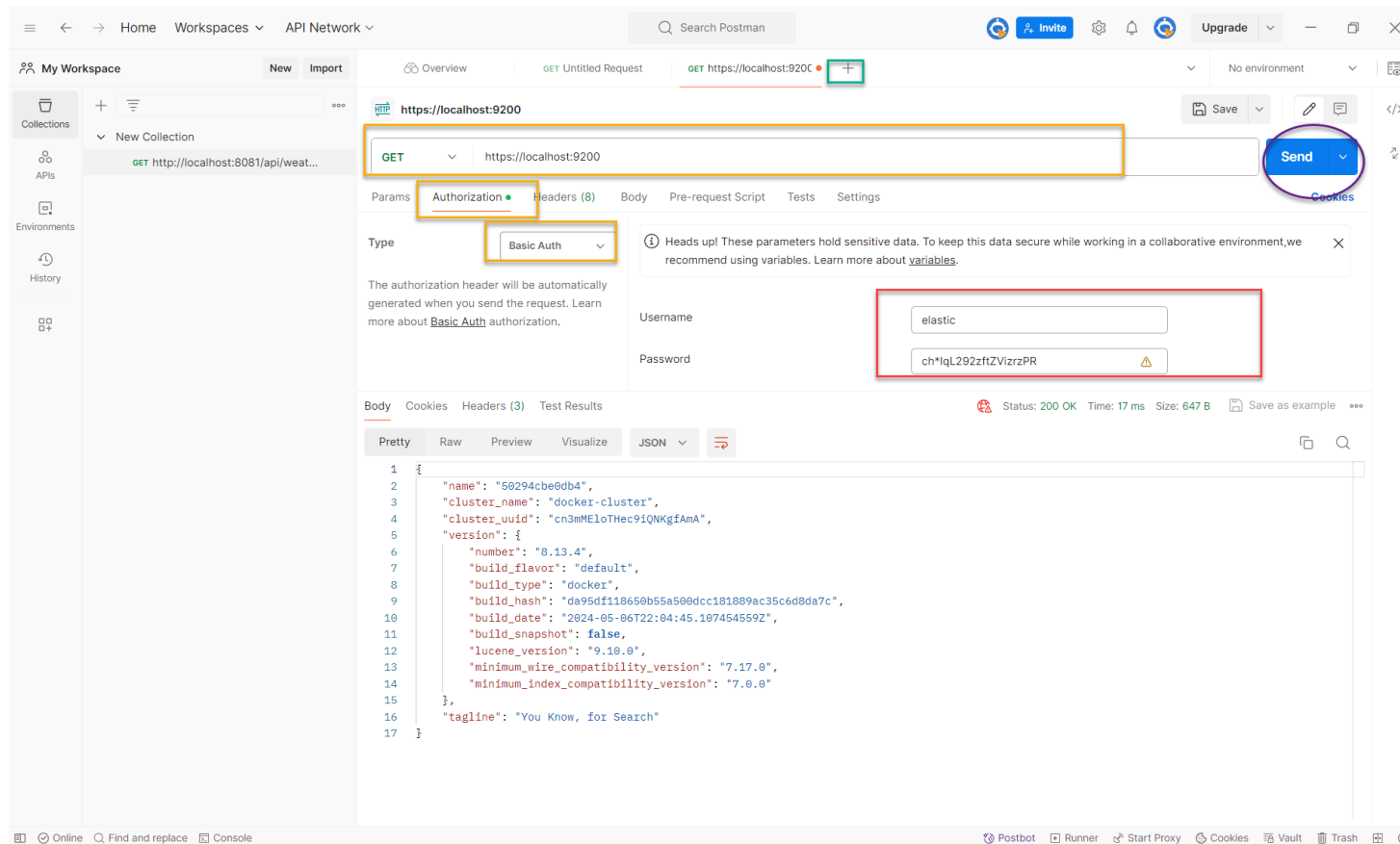
[Release Notes](#)

Not your OS? Download for Mac (Intel Chip, Apple Chip) or Linux (x64, arm64)

Postman on the web



## Tester Elastic API Rest avec Postman



Curl est un client basé sur la ligne de commande disponible sur la plupart des systèmes d'exploitation. Postman est un client HTTP basé sur l'interface utilisateur disponible pour les principaux systèmes d'exploitation.

```

export ELASTIC_PASSWORD="votre_mot_de_passe"
#export ELASTIC_PASSWORD="ch*IqL292zftZVizrzPR"
docker cp es01:/usr/share/elasticsearch/config/certs/http_ca.crt .

curl --cacert http_ca.crt -u elastic:$ELASTIC_PASSWORD https://localhost:9200
  
```

```

msellami@LAPTOP-5060CAT1:~/elastic$ curl --cacert http_ca.crt -u elastic:$ELASTIC_PASSWORD https://localhost:9200
{
  "name": "50294cbe0db4",
  "cluster_name": "docker-cluster",
  "cluster_uuid": "cn3mMEloThec9iQNKgfAmA",
  "version": {
    "number": "8.13.4",
    "build_flavor": "default",
    "build_type": "docker",
    "build_hash": "da95df118650b55a500dcc181889ac35c6d8da7c",
    "build_date": "2024-05-06T22:04:45.107454559Z",
    "build_snapshot": false,
    "lucene_version": "9.10.0",
    "minimum_wire_compatibility_version": "7.17.0",
    "minimum_index_compatibility_version": "7.0.0"
  },
  "tagline": "You Know, for Search"
}
msellami@LAPTOP-5060CAT1:~/elastic$
  
```

## CURL Aide-Mémoire :

### Curl cheatsheet

#### Options

```

-o <file> # --output: write to file
-u user:pass # --user: Authentication

-v # --verbose
-vv # Even more verbose
-s # --silent

-i # --include: Include the HTTP-header in the output
-I # --head: headers only

```

#### SSL

```

--cacert <file>
--capath <dir>

-E, --cert <cert> # --cert: Client cert file
--cert-type # der/pem/eng
-k, --insecure # for self-signed certs

```

#### Request

```

-X POST # --request
-L # follow link if page redirects

```

#### Data

```

-d 'data' # --data: HTTP post data, URL encoded (eg, status="Hello")
-d @file # --data via file
-G # --get: send -d data via get

```

#### Headers

```

-A <str> # --user-agent
-b name=val # --cookie
-b FILE # --cookie
-H "X-Foo: y" # --header
--compressed # use deflate/gzip

```

## Examples

```

# Post data:
curl -d password=x http://x.com/y

# Auth/data:
curl -u user:pass -d status="Hello" http://twitter.com/statuses/update.xml

# multipart file upload
curl -v -include --form key1=value1 --form upload=@localfilename URL

# Use Curl to Check if a remote resource is available
# details: https://matthewsetter.com/check-if-file-is-available-with-curl/
curl -o /dev/null --silent -Iw "%{http_code}" https://example.com/my.remote.tarball.gz

```