

# SAE 5.Cyber.03 - R5.Cyber.11

## Installation d'Elastic

---

Flavien Marchand

---

## Sommaire

<b>Sommaire</b>	<b>1</b>
<b>Travail à faire</b>	<b>2</b>
<b>Installation d'Elastic</b>	<b>3</b>
<b>Installation de Kibana</b>	<b>5</b>
Beats	6
1. Supervision d'une machine Ubuntu	9
2. Supervision d'un service web (Apache, Nginx, ...)	9
3. Supervision d'équipements Cisco	9
4. Supervision d'un autre service/équipement	9
Installation des Beats	10
AuditBeat	10
Filebeat	12
Metricbeat :	14
Winlogbeat	16

### Commandes pour démarrer elastic, kibana et les beats :

```
cd elasticsearch-8.10.4/  
./bin/elasticsearch
```

```
cd kibana-8.10.4/  
./bin/kibana
```

```
cd auditbeat-8.10.4-linux-x86_64/  
sudo ./auditbeat -e
```

```
cd filebeat-8.10.4-linux-x86_64/  
sudo ./filebeat -e
```

```
cd heartbeat-8.10.4-linux-x86_64/  
sudo ./heartbeat -e
```

<http://localhost:5601>

---

## Travail à faire

- 1/ Un compte rendu d'installation de votre suite elastic sur Ubuntu. **(SAE5.Cyber.03)**
  - 2/ Un compte rendu de supervision avec votre suite elastic d'une machine Ubuntu **(SAE5.Cyber.03)**
  - 3/ Un compte rendu de supervision avec votre suite elastic d'un service web (Apache, nginx, ...) **(SAE5.Cyber.03)**
  - 4/ Un compte rendu de supervision avec votre suite elastic d'équipement(s) Cisco **(SAE5.Cyber.03)**
  - 5/ Un compte rendu de supervision d'un autre service/équipement **(SAE5.Cyber.03)**
  - 6/ Comptes rendus (dans un seul fichier) de vos analyses de logs des TP0 à TP5 **(R5.Cyber.11)**
-

# Installation d'Elastic

wget

[https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-8.10.4-linux-x86\\_64.tar.gz](https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-8.10.4-linux-x86_64.tar.gz)

wget

[https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-8.10.4-linux-x86\\_64.tar.gz.sha512](https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-8.10.4-linux-x86_64.tar.gz.sha512)

shasum -a 512 -c elasticsearch-8.10.4-linux-x86\_64.tar.gz.sha512

tar -xzf elasticsearch-8.10.4-linux-x86\_64.tar.gz

**Mettre tous les dossier de elastic en 777 :**

chmod 777 -R elasticsearch-8.10.4

cd elasticsearch-8.10.4/

**Modifier ces lignes dans elasticsearch-8.10.4/config/elasticsearch.yml :**

```
# ----- Network -----  
#  
# 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.  
#az
```

**Ajouter cette ligne dans elasticsearch-8.10.4/config/elasticsearch.yml :**

**action.auto\_create\_index:**  
**.monitoring\*,.watches,.triggered\_watches,watcher-history\*,.ml\***

**Lancer ElasticSearch dans le répertoire elasticsearch-8.10.4/ avec un user **non root** :**

./bin/elasticsearch

---

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

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

**password**

**i** HTTP CA certificate SHA-256 fingerprint:

**88e1e06a7695b7ffdded7c49fe72c88f859b257644e4f8710d86a2e542e81031**

**i** 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):

**AAEAAWVsYXN0aWMva2liYW5hL2RlZmF1bHQ6MkowQkZsRVNSUWVwbnlOSExidVF4dw**

**i** 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):

**eyJ2ZXIiOiI4LjEwLjQiLCJhZHliOiIsImTcyLjE3LjAuMTU5MjAwI0sImZnciI6Ijg4ZTFIMDZhNzY5NWl3ZmZkZGVkN2M0OWZlbnZJjODhmODU5YjI1NzY0NGU0Zjg3MTBkODZhMmU1NDJlODEwMzEiLCJrZXkiOiJwZjRzMUpNQnpNUEFHdExxZ0VkdDpmWTk0eW1RMlItQ1RvWEV2TUFUYW1RIn0=**

- On this node:
    - Create an enrollment token with ``bin/elasticsearch-create-enrollment-token -s node``.
    - Uncomment the `transport.host` setting at the end of `config/elasticsearch.yml`.
    - Restart Elasticsearch.
  - On other nodes:
    - Start Elasticsearch with ``bin/elasticsearch --enrollment-token <token>``, using the enrollment token that you generated.
- 

**Si besoin pour recréer un nouveau token pour kibana:**

`bin/elasticsearch-create-enrollment-token -scope kibana`

**Modifier cette ligne dans config/elasticsearch.yml pour accepter les connections depuis n'importe quel hôte :**

# Allow HTTP API connections from anywhere

# Connections are encrypted and require user authentication

**http.host: 0.0.0.0**

## Installation de Kibana

wget

[https://artifacts.elastic.co/downloads/kibana/kibana-8.10.4-linux-x86\\_64.tar.gz](https://artifacts.elastic.co/downloads/kibana/kibana-8.10.4-linux-x86_64.tar.gz)

wget

[https://artifacts.elastic.co/downloads/kibana/kibana-8.10.4-linux-x86\\_64.tar.gz.sha512](https://artifacts.elastic.co/downloads/kibana/kibana-8.10.4-linux-x86_64.tar.gz.sha512) | shasum -a 512 -c kibana-8.10.4-linux-x86\_64.tar.gz.sha512

tar -xzf kibana-8.10.4-linux-x86\_64.tar.gz

**Mettre tous les dossier de kibana en 777 :**

chmod 777 -R kibana-8.10.4

cd kibana-8.10.4/

**Modifier ces lignes dans kibana-8.10.4/config/kibana.yml :**

# ===== 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**

# ===== System: Elasticsearch =====

# The URLs of the Elasticsearch instances to use for all your queries.

**elasticsearch.hosts: ["https://127.0.0.1:9200"]**

**# If your Elasticsearch is protected with basic authentication, these settings provide**

**# the username and password that the Kibana server uses to perform maintenance on the Kibana**

**# index at startup. Your Kibana users still need to authenticate with Elasticsearch, which**

**# is proxied through the Kibana server.**

**#elasticsearch.username: "elastic"**

**elasticsearch.password: "password"**

**# Kibana can also authenticate to Elasticsearch via "service account tokens".**

**# Service account tokens are Bearer style tokens that replace the traditional username/password based configuration.**

**# Use this token instead of a username/password.**

**elasticsearch.serviceAccountToken:**

**"AAEAAWVsYXN0aWMva2liYW5hL2RlZmF1bHQ6MkowQkZsRVNSUWVwbnlOSExi  
dVF4dw  
"**

**Start kibana dans le répertoire kibana-8.10.4/ en **non root** :**

**./bin/kibana**

**se rendre sur et entrer le token pour kibana que l'on a reçu lors du démarrage de elastic:**

**<http://localhost:5601/?code=451133>**

**Ensuite se connecter avec le user **elastic** et son mdp **bx4qA8+DgzjMWRDx2Q-N****

## **Beats**

### **Auditbeat :**

Auditbeat est un expéditeur léger que vous pouvez installer sur vos serveurs pour auditer les activités des utilisateurs et des processus sur vos systèmes. Par exemple, vous pouvez utiliser Auditbeat pour collecter et centraliser les événements d'audit de Linux Audit Framework. Vous pouvez également utiliser Auditbeat pour détecter les modifications apportées aux fichiers critiques, tels que les fichiers binaires et les fichiers de configuration, et identifier les violations potentielles des politiques de sécurité.

### **Filebeat :**

Filebeat est un expéditeur léger pour transférer et centraliser les données de journaux. Installé en tant qu'agent sur vos serveurs, Filebeat surveille les fichiers journaux ou les emplacements que vous spécifiez, collecte les événements de journal et les transmet à Elasticsearch ou Logstash pour indexation.

### **Functionbeat :**

Functionbeat est un Elastic Beat que vous déployez en tant que fonction dans votre environnement sans serveur pour collecter des données à partir de services cloud et les envoyer à la Suite Elastic.

### **Heartbeat :**

Heartbeat est un démon léger que vous installez sur un serveur distant pour vérifier périodiquement l'état de vos services et déterminer s'ils sont disponibles. Contrairement à Metricbeat, qui vous indique uniquement si vos serveurs sont opérationnels ou indisponibles, Heartbeat vous indique si vos services sont accessibles.

Heartbeat est utile lorsque vous devez vérifier que vous respectez vos accords de niveau de service concernant la disponibilité du service. Il est également utile dans d'autres scénarios, tels que les cas d'utilisation de la sécurité, lorsque vous devez vérifier que personne de l'extérieur ne peut accéder aux services de votre serveur d'entreprise privé.

Vous pouvez configurer Heartbeat pour envoyer une requête ping à toutes les adresses IP résolubles par DNS pour un nom d'hôte spécifié. De cette façon, vous pouvez vérifier tous les services dont la charge est équilibrée pour voir s'ils sont disponibles.

Lorsque vous configurez Heartbeat, vous spécifiez des moniteurs qui identifient les noms d'hôte que vous souhaitez vérifier. Chaque moniteur s'exécute en fonction de la planification que vous spécifiez. Par exemple, vous pouvez configurer un moniteur pour qu'il s'exécute toutes les 10 minutes et un autre moniteur pour qu'il s'exécute entre 9h00 et 17h00.

Heartbeat prend actuellement en charge les moniteurs pour vérifier les hôtes via :

- Requêtes d'écho ICMP (v4 et v6). Utilisez le moniteur ICMP lorsque vous souhaitez simplement vérifier si un service est disponible. Ce moniteur nécessite un accès root.
- TCP. Utilisez le moniteur TCP pour vous connecter via TCP. Vous pouvez éventuellement configurer ce moniteur pour vérifier le point de terminaison en envoyant et/ou en recevant une charge utile personnalisée.
- HTTP. Utilisez le moniteur http pour vous connecter via HTTP. Vous pouvez éventuellement configurer ce moniteur pour vérifier que le service renvoie la réponse attendue, telle qu'un code d'état spécifique, un en-tête de réponse ou un contenu.

**Metricbeat :**

Metricbeat est un expéditeur léger que vous pouvez installer sur vos serveurs pour collecter périodiquement des métriques du système d'exploitation et des services exécutés sur le serveur. Metricbeat prend les métriques et statistiques qu'il collecte et les envoie à la sortie que vous spécifiez, telle qu'Elasticsearch ou Logstash.

**Packetbeat :**

Packetbeat est un analyseur de paquets réseau en temps réel que vous pouvez utiliser avec Elasticsearch pour fournir un système de surveillance des applications et d'analyse des performances. Packetbeat complète la plateforme Beats en offrant une visibilité entre les serveurs de votre réseau.

Packetbeat fonctionne en capturant le trafic réseau entre vos serveurs d'applications, en décodant les protocoles de la couche application (HTTP, MySQL, Redis, etc.), en corrélant les requêtes avec les réponses et en enregistrant les champs intéressants pour chaque transaction.

Packetbeat peut vous aider à détecter facilement les problèmes de votre application back-end, tels que des bugs ou des problèmes de performances, et permet de les dépanner - et donc de les réparer - beaucoup plus rapidement.

**Winlogbeat :**

Winlogbeat envoie les journaux d'événements Windows à Elasticsearch ou Logstash. Vous pouvez l'installer en tant que service Windows.

Winlogbeat lit un ou plusieurs journaux d'événements à l'aide des API Windows, filtre les événements en fonction de critères configurés par l'utilisateur, puis envoie les données d'événement aux sorties configurées (Elasticsearch ou Logstash). Winlogbeat surveille les journaux d'événements afin que les nouvelles données d'événements soient envoyées en temps opportun. La position de lecture de chaque journal d'événements est conservée sur le disque pour permettre à Winlogbeat de reprendre après le redémarrage.

Winlogbeat peut capturer les données d'événements de tous les journaux d'événements exécutés sur votre système. Par exemple, vous pouvez capturer des événements tels que :

- événements d'application
- événements matériels
- événements de sécurité
- événements système



## Voici les différents Beats que je vais installer pour chaque Supervision que je dois effectuer :

### 1. Supervision d'une machine Ubuntu

- **Metricbeat** : C'est l'outil idéal pour surveiller la performance du système Ubuntu (CPU, mémoire, disque, réseau, etc.).
- **Auditbeat** : Pour la surveillance des activités au niveau du système d'exploitation, comme les connexions utilisateurs, les modifications de fichiers et la surveillance des processus.

### 2. Supervision d'un service web (Apache, Nginx, ...)

- **Filebeat** : Pour collecter les logs des serveurs web comme Apache ou Nginx. Filebeat peut être configuré pour lire les fichiers de logs spécifiques à ces services.

### 3. Supervision d'équipements Cisco

- **Filebeat** : Filebeat avec le module Cisco peut être configuré pour capturer et analyser les logs générés par les équipements Cisco.

**Enfinement Logstash à été utilisé car Filebeat ne renvoyait pas les dashboards à Elastic ou ne possédait pas les bons modules (voir [CR Supervision Switch Cisco](#))**

### 4. Supervision d'un autre service/équipement

- **Filebeat** : Pour capturer et analyser les logs du service ou de l'équipement spécifique.
- **Metricbeat** : Si l'équipement ou le service peut fournir des métriques, Metricbeat peut aider à les collecter et les analyser.
- **Heartbeat** : Si on veut surveiller la disponibilité et la latence d'un service, Heartbeat est idéal pour cela.

En conclusion il faut installer **Auditbeat**, **Filebeat**, et potentiellement **Heartbeat** si je veux superviser la disponibilité des services et **Winlogbeat** si le service que je dois tester est sous windows.

**Pour l'installation des beats, il faut les installer sur les clients que l'on veut superviser.**

## **Installation des Beats**

### **AuditBeat**

wget

[https://artifacts.elastic.co/downloads/beats/auditbeat/auditbeat-8.10.4-linux-x86\\_64.tar.gz](https://artifacts.elastic.co/downloads/beats/auditbeat/auditbeat-8.10.4-linux-x86_64.tar.gz)

tar xzvf auditbeat-8.10.4-linux-x86\_64.tar.gz

cd auditbeat-8.10.4-linux-x86\_64

**Modifier dans auditbeat.yml :**

#### **Elastic :**

# ----- Elasticsearch Output -----

output.elasticsearch:

# Array of hosts to connect to.

**hosts: ["adresse\_du\_serveur:9200"]**

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

**protocol: "https"**

# Authentication credentials - either API key or username/password.

**#api\_key: "id:api\_key"**

**username: "elastic"**

**password: "password"**

**ssl.verification\_mode: "none"**

#### **Kibana :**

# ===== Kibana

=====

# Starting with Beats version 6.0.0, the dashboards are loaded via the Kibana API.

# This requires a Kibana endpoint configuration.

setup.kibana:

# Kibana Host  
 # Scheme and port can be left out and will be set to the default (http and 5601)  
 # In case you specify an additional path, the scheme is required:  
 http://localhost:5601/path  
 # IPv6 addresses should always be defined as: https://[2001:db8::1]:5601  
**host: "http://adresse\_du\_serveur:5601"**

# Kibana Space ID  
 # ID of the Kibana Space into which the dashboards should be loaded. By default,  
 # the Default Space will be used.  
 #space.id:

### Installer les assets :

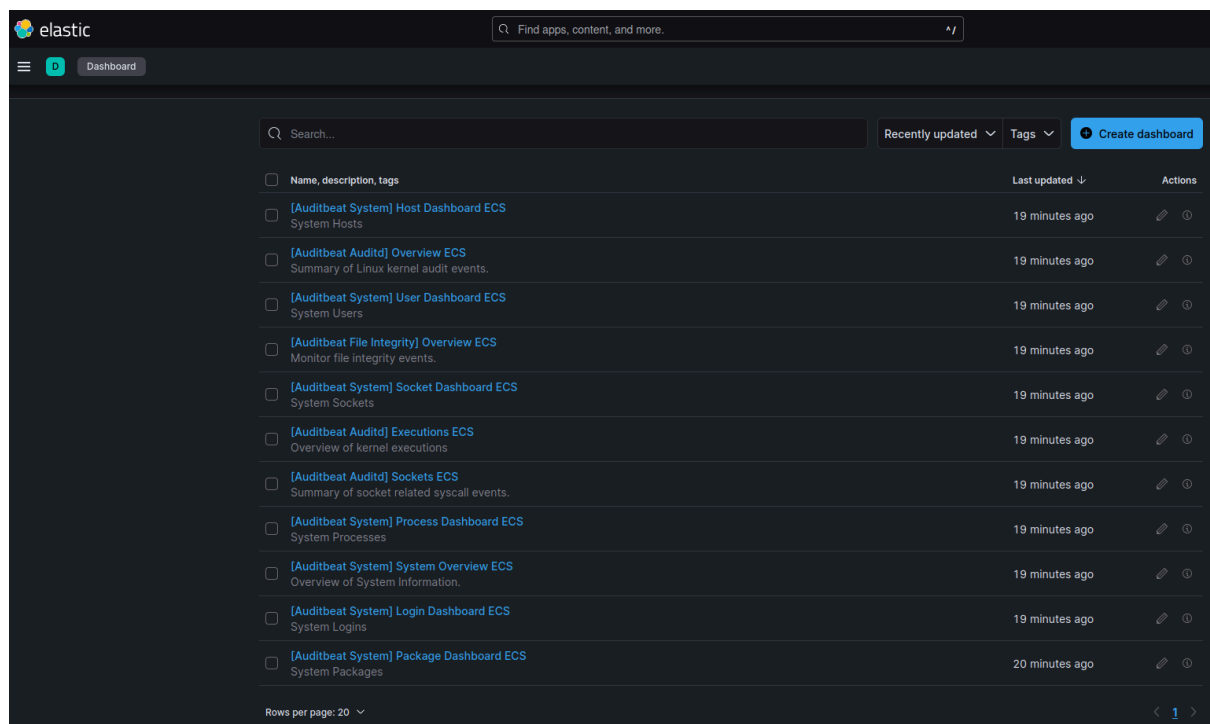
./auditbeat setup -e

### Modifiez les informations d'identification de l'utilisateur dans auditbeat.yml et spécifiez un utilisateur autorisé à publier des événements :

sudo chown root auditbeat.yml

### Lancer auditbeat en **root** :

En retournant sur l'interface graphique d'Elastic on peut observer dans le dashboard que l'on a bien plusieurs dashboards qui se sont ajoutés grâce au module AuditBeat :



## Filebeat

wget

[https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-8.10.4-linux-x86\\_64.tar.gz](https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-8.10.4-linux-x86_64.tar.gz)

```
tar xzvf filebeat-8.10.4-linux-x86_64.tar.gz
```

```
cd filebeat-8.10.4-linux-x86_64
```

**Modifier dans filebeat.yml :**

### Kibana :

```
# ===== Kibana
=====
```

```
# Starting with Beats version 6.0.0, the dashboards are loaded via the Kibana API.
# This requires a Kibana endpoint configuration.
setup.kibana:
```

```
# Kibana Host
```

```
# Scheme and port can be left out and will be set to the default (http and 5601)
```

```
# In case you specify an additional path, the scheme is required: http://localhost:5601/path
```

```
# IPv6 addresses should always be defined as: https://[2001:db8::1]:5601
```

```
host: "http://adresse_du_serveur:5601"
```

```
# Kibana Space ID
```

```
# ID of the Kibana Space into which the dashboards should be loaded. By default, the
# the Default Space will be used.
```

```
#space.id:
```

### Elastic :

```
# ----- Elasticsearch Output -----
```

```
output.elasticsearch:
```

```
# Array of hosts to connect to.
```

```
hosts: ["adresse_du_serveur:9200"]
```

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

```
protocol: "https"
```

```
# Authentication credentials - either API key or username/password.
```

```
#api_key: "id:api_key"
```

```
username: "elastic"  
password: "password"  
ssl.verification_mode: "none"
```

**Activer les modules de collection de données :**

```
./filebeat modules list
```

**Ensuite on peut activer un module comme ceci (nginx étant un exemple) :**

```
./filebeat modules enable nginx
```

**Dans le répertoire [modules.d](#) on peut accéder aux fichiers de conf des modules, on peut modifier le fichier [nginx.yml](#) après l'avoir enable :**

**il faut l'enable et spécifier le chemin d'accès aux logs si elles ne sont pas au chemin par défaut :**

```
- module: nginx  
  # Access logs  
  access:  
    enabled: true  
    var.paths: ["/var/log/nginx/access.log*"]  
  
  # Error logs  
  error:  
    enabled: true  
    var.paths: ["/var/log/nginx/error.log*"]
```

**Installer les assets :**

```
./filebeat setup -e
```

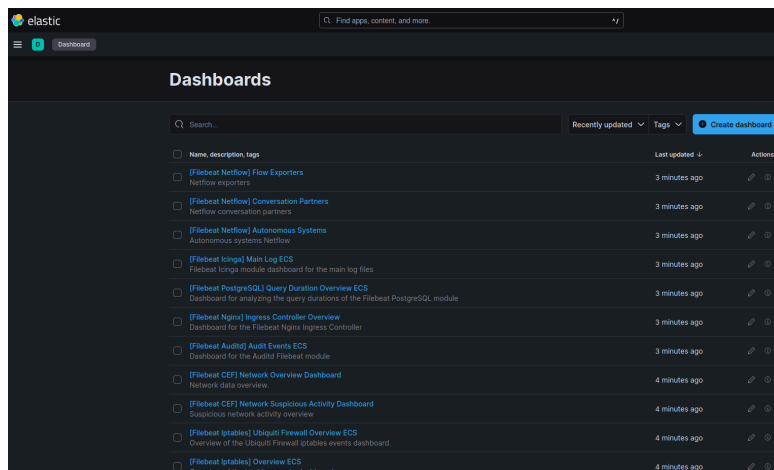
**Modifiez les informations d'identification de l'utilisateur dans [auditbeat.yml](#) et spécifiez un utilisateur autorisé à publier des événements :**

```
sudo chown root filebeat.yml
```

**Lancer filebeat en [root](#) :**

```
./filebeat -e
```

**En retournant sur l'interface graphique d'Elastic on peut observer dans le dashboard que l'on a bien plusieurs dashboards qui se sont ajoutés grâce au module Filebeat :**



**Bien sûr pour obtenir des informations dans les dashboards il faut activer les modules nécessaires, je n'ai activé que nginx pour l'instant.**

**Metricbeat :**

wget

[https://artifacts.elastic.co/downloads/beats/metricbeat/metricbeat-8.10.4-linux-x86\\_64.tar.gz](https://artifacts.elastic.co/downloads/beats/metricbeat/metricbeat-8.10.4-linux-x86_64.tar.gz)

tar -xzf metricbeat-8.10.4-linux-x86\_64.tar.gz

**Modifier dans metricbeat.yml :**

**Kibana :**

```
# ===== Kibana
=====
```

# Starting with Beats version 6.0.0, the dashboards are loaded via the Kibana API  
# This requires a Kibana endpoint configuration.  
setup.kibana:

# Kibana Host

# Scheme and port can be left out and will be set to the default (http and 5601)

# In case you specify an additional path, the scheme is required: http://localhost/

# IPv6 addresses should always be defined as: https://[2001:db8::1]:5601

**host: "http://adresse\_du\_serveur:5601"**

# Kibana Space ID

# ID of the Kibana Space into which the dashboards should be loaded. By default

# the Default Space will be used.

#space.id:

### **Elastic :**

# ----- Elasticsearch Output -----

output.elasticsearch:

# Array of hosts to connect to.

**hosts: ["adresse\_du\_serveur:9200"]**

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

**protocol: "https"**

# Authentication credentials - either API key or username/password.

**#api\_key: "id:api\_key"**

**username: "elastic"**

**password: "password"**

**ssl.verification\_mode: "none"**

### **Activer le module docker :**

sudo ./metricbeat modules enable docker

### **Ajouter cette conf dans le docker.yml :**

nano modules.d/docker.yml

```
- module: docker
  metricsets:
    - container
    - cpu
    - diskio
    - healthcheck
    - info
    - memory
    - network
  hosts: ["unix:///var/run/docker.sock"]
  period: 10s
```

### **Activer le module mysql :**

sudo ./metricbeat modules enable mysql

### **Ajouter cette conf dans le mysql.yml :**

nano modules.d/mysql.yml

```
- module: mysql
  metricsets:
    - status
    - galera_status
    - performance
```

```

    - query
period: 10s

namespace: "my_namespace"

hosts: ["user:password@tcp(Docker_ip:3306)/database_name"]

# Queries for the `query` metricset
queries:
  - name: forums_table
    query: "SELECT * from forums;"
    response_format: table
    query_namespace: "forums_table"
  - name: logins_table
    query: "SELECT * from logins;"
    response_format: table
    query_namespace: "logins_table"

```

**Ma bdd étant dans un docker l'ip de la bdd est donc celle de mon docker (Docker\_ip).**

**Installer les assets :**

```
./metricbeat setup -e
```

**Modifiez les informations d'identification de l'utilisateur dans heartbeat.yml et spécifiez un utilisateur autorisé à publier des événements :**

```
sudo chown root metricbeat.yml
```

**Lancer auditbeat en root :**

```
./metricbeat -e
```

## Winlogbeat

[https://artifacts.elastic.co/downloads/beats/winlogbeat/winlogbeat-8.10.4-windows-x86\\_64.zip](https://artifacts.elastic.co/downloads/beats/winlogbeat/winlogbeat-8.10.4-windows-x86_64.zip)

**Modifier dans winlogbeat-8.10.4-windows-x86\_64/winlogbeat.yml :**

**Elastic :**

```
# ----- Elasticsearch Output -----
```

```
output.elasticsearch:
```

```
# Array of hosts to connect to.
```

```
hosts: ["adresse_du_serveur:9200"]
```

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



**protocol: "https"**

# Authentication credentials - either API key or username/password.

#api\_key: "id:api\_key"

**username: "elastic"**

**password: "password"**

**ssl.verification\_mode: "none"**

## **Kibana :**

# ===== Kibana

=====

# Starting with Beats version 6.0.0, the dashboards are loaded via the Kibana API.

# This requires a Kibana endpoint configuration.

setup.kibana:

# Kibana Host

# Scheme and port can be left out and will be set to the default (http and 5601)

# In case you specify an additional path, the scheme is required:

http://localhost:5601/path

# IPv6 addresses should always be defined as: https://[2001:db8::1]:5601

**host: "http://adresse\_du\_serveur:5601"**

# Kibana Space ID

# ID of the Kibana Space into which the dashboards should be loaded. By default,

# the Default Space will be used.

#space.id:

## **Installer les assets :**

winlogbeat.exe setup -e

## **Lancer Winlogbeat**

winlogbeat.exe -e