COMPRENDRE ET UTILISER FILEBEAT DANS LA STACK ELK

Introduction

Tout comme Logstash, Filebeat peut être utilisé pour envoyer des logs à partir d'une source de données basée sur des fichiers vers une destination de sortie prise en charge. Mais la comparaison s'arrête là. Car dans la plupart des cas, nous utiliserons les deux en tandem lors de la création de notre pipeline de journalisation avec la pile ELK, puisque les deux ont une fonction différente.

La limite de Logstash

Logstash a été développé à l'origine par Jordan Sissel pour gérer le streaming d'une grande quantité de données de logs provenant de plusieurs sources, et après que Sissel a rejoint l'équipe Elastic, Logstash est passé d'un outil autonome à une partie intégrante de la pile ELK (Elasticsearch, Logstash, Kibana).

Pour pouvoir déployer un système de journalisation centralisé efficace, un outil capable à la fois d'extraire des données de plusieurs sources de données et de leur donner un sens est nécessaire. Tel est le rôle joué par Logstash, il gère les tâches de réception des données provenant de plusieurs systèmes, les transformants en un ensemble significatif de champs et éventuellement en continu la sortie vers une destination définie pour le stockage.

Eh bien, il y avait, et il y a toujours, un problème en suspens avec Logstash, et c'est **la performance**. Logstash nécessite la JVM (Java Virtual Machine) pour s'exécuter, et cette dépendance couplée à l'implémentation dans Ruby est devenue la cause

première d'une consommation de mémoire importante, en particulier lorsque plusieurs pipelines et qu'un filtrage avancé sont impliqués.

La naissance de Beats

Lumberjack a été initialement développé comme une expérience d'externalisation des tâches d'extraction de données et était destiné à être utilisé comme un **expéditeur léger pour collecter les logs** avant de les envoyer pour traitement sur une autre plate-forme (telle que Logstash ou directement sur Elasticsearch). Écrit en Go, le concept derrière Lumberjack était de développer un protocole réseau qui serait plus efficace pour gérer de gros volumes de données, aurait une **faible empreinte mémoire** et prendrait en charge le chiffrement.

Le projet a été renommé ensuite ensuite en "Logstash-Forwarder", constituant et incluant désormais le protocole réseau et le programme de journalisation. Ensuite, une deuxième version du protocole Lumberjack a été développée, dépréciant ainsi Logstash-Forwarder. Ce nouveau protocole connu sous le nom de "Beats" est utilisé par une nouvelle famille d'expéditeurs (ex d'agent Beats : Filebeat pour les fichiers logs, Packetbeat pour les métriques réseaux Metricbeat, etc ...).

Dans ce chapitre nous allons **apprendre à utiliser Filebeat**.

Dois-je utiliser Filebeat ou Logstash ?

Pour collecter des logs sur des machines distantes, Filebeat est recommandé car il nécessite moins de ressources qu'une instance de Logstash. Cependant, vous utiliseriez Logstash si vous souhaitez manipuler vos logs pour ajouter ou supprimer des champs ou enrichir vos données à l'aide de filtres d'entrée/sortie pour les envoyer ensuite à Elasticsearch. Quant à Filebeat, c'est un choix parfait pour

récupérer une donnée déjà traitée et la transmettre directement à Elasticsearch.

Pour information, **Filebeat et Logstash peuvent être utilisés conjointement**. Par exemple si vous devez collecter des logs à partir de machines distantes, vous pouvez utiliser Filebeat pour les récupérer et ensuite les envoyer à Logstash si vous voulez faire des transformations sur vos données avant de les envoyer à Elasticsearch.

Utilisation de Filebeat

Installation de Filebeat

Avant de commencer l'installation, assurez-vous d'avoir installé Elasticsearch pour stocker et rechercher nos données, et d'avoir installé Kibana pour les visualiser et les gérer (mon tuto d'installation est disponible ici).

Il existe plusieurs façons d'installer Filebeat. Dans notre cas nous allons **installer Filebeat depuis le gestionnaire de paquets par défaut** de notre distribution. Pour ce faire, vous devrez d'abord mettre à niveau votre système et vos paquets:

```
sudo apt update -y && sudo apt upgrade -y
```

Pour les machines appartenant à la famille debian, vous devrez peut-être installer le paquet `apt-transport-https` avant de continuer:

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

Téléchargez et installez ensuite la clé de signature publique:

Sous la famille debian:

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

Sous la famille redhat:

```
sudo rpm --import https://packages.elastic.co/GPG-KEY-elasticsearch
```

L'étape suivante consiste à ajouter le dépôt Elastic sur votre système :

Sous la famille debian:

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

Sous la famille redhat, créez un fichier et nommez le par exemple **elastic.repo** dans le répertoire **/etc/yum.repos.d/** , contenant:

```
[elastic-7.x]
name=Elastic repository for 7.x packages
baseurl=https://artifacts.elastic.co/packages/7.x/yum
gpgcheck=1
gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch
enabled=1
autorefresh=1
type=rpm-md
```

Il ne vous reste plus qu'à mettre à jour vos référentiels et installer Filebeat:

Sous la famille debian:

```
sudo apt-get update && sudo apt-get install filebeat
```

Sous la famille redhat:

```
sudo yum install filebeat
```

Pour exécuter Filebeat, utilisez la commande suivante:

```
sudo systemctl start filebeat
```

Si jamais vous rencontrez des problèmes d'initialisation, veuillez vérifier les logs du service Filebeat à l'aide de la commande suivante :

```
sudo journalctl -f -u filebeat
```

Configuration de Filebeat

Le fichier de configuration de Filebeat se retrouve dans **/etc/filebeat/filebeat.yml** . Dans ce fichier, assurez-vous bien que la configuration Filebeat possède les bonnes informations pour communiquer avec Kibana et Elasticsearch, si besoin décommentez les lignes suivantes :

```
output.elasticsearch:
  hosts: ["localhost:9200"]
  username: "" #(si pas de login/mot de passe ne rien mettre)
  password: "" #(si pas de login/mot de passe ne rien mettre)

...

setup.kibana:
  host: "localhost:5601"

...

# Optionnelle
filebeat.config.modules:
  path: ${path.config}/modules.d/*.yml
  reload.enabled: true
  reload.period: 15s
```

Ci-dessous une explication détaillée sur **les options de configuration Filebeat** utilisées:

- `filebeat.config.modules` : où nous retrouverons l'option `path` qui est l'emplacement des fichiers de configuration des modules Filebeat situés par défaut dans le dossier `/etc/filebeat/modules.d/`, et pour les cibler par défaut filebeat utilise la regex `*.yml`. Par ailleurs, nous avons également la possibilité

d'activer ou non le rechargement automatique de la configuration qui est par défaut à `false` dans l'option `reload.enabled`. Si vous passez la valeur `true` comme moi, alors Filebeat surveillera périodiquement (ici toutes les 15 secondes) vos fichiers de configuration et si des changements sont détectés, il rechargera l'ensemble de la configuration.

- `setup.kibana` : pour que les tableaux de bord fonctionnent, nous devons spécifier le point de terminaison Kibana. Vous devrez entrer l'URL de votre hôte Kibana et vos informations d'identification (nom d'utilisateur/mot de passe) si nécessaire.
- `output.elasticsearch` : spécifie la sortie à laquelle nous envoyons les métriques Filebeat. Nous utilisons Elasticsearch, vous devrez donc fournir l'hôte, le protocole et les informations d'identification Elasticsearch si nécessaire.

Pour initialiser le service à chaque démarrage de la machine, lancez la commande suivante:

```
sudo systemctl enable filebeat
```

Les modules Filebeat (Apache2)

Les modules Filebeat vous permettent de commencer rapidement à traiter les formats de logs courants. Ils contiennent des configurations par défaut, des définitions de pipeline pour Elasticsearch et des tableaux de bord préfabriqués Kibana pour vous aider à mettre en œuvre et à déployer une solution de surveillance des logs rapidement. La liste des modules est disponible ici.

Pour cet article, nous allons utiliser le module apache2 afin d'**analyser les logs d'accès et d'erreurs créés par un serveur Apache** . Lorsque vous exécutez le module, il effectue quelques tâches automatiquement pour vous :

- Définit les chemins des fichiers logs par défaut (vous pouvez remplacer les valeurs par défaut)
- S'assure que chaque événement de logs multiligne est envoyé en tant qu'événement unique
- Analyse et traite les lignes de log Apache automatiquement, et façonne les données dans une structure adaptée à la visualisation dans Kibana
- Déploie des tableaux de bord automatique dans Kibana pour visualiser les données de vos logs

Avant de suivre ces étapes, vérifiez qu'Elasticsearch et Kibana sont en cours d'exécution et qu'Elasticsearch est prêt à recevoir des données de Filebeat:

```
sudo systemctl status elasticsearch kibana
```

Résultat :

```
? elasticsearch.service - Elasticsearch
  Loaded: loaded (/lib/systemd/system/elasticsearch.service; disabled; vendor preset
  Active: active (running) since Mon 2021-07-12 14:47:17 CEST; 3h 21min ago

? kibana.service - Kibana
  Loaded: loaded (/etc/systemd/system/kibana.service; disabled; vendor preset: enable
  Active: active (running) since Mon 2021-07-12 14:46:47 CEST; 3h 21min ag
```

Pour **remplacer les chemins par défaut pour l'accès au serveur HTTP Apache et les logs d'erreurs dans Logstash**. Rendez-vous dans le fichier de configuration du module apache **/etc/filebeat/modules.d/apache.yml** :

```
- module: apache
  access:
    enabled: true
    var.paths: ["/nouveau-path/log/apache/access.log*"]
  error:
    enabled: true
    var.paths: ["/nouveau-path/log/apache/error.log*"]
```

Pour configurer et exécuter le module Apache, lancez la commande suivante:

```
sudo filebeat modules enable apache
```

Pour voir la liste des modules activés et désactivés, exécutez :

```
sudo filebeat modules list
```

Résultat :

```
Enabled:
apache

Disabled:
activemq
...
```

Ensuite lancez la sous-commande `setup` pour charger les tableaux de bord dans Kibana (la commande peut prendre un peu de temps pour se terminer):

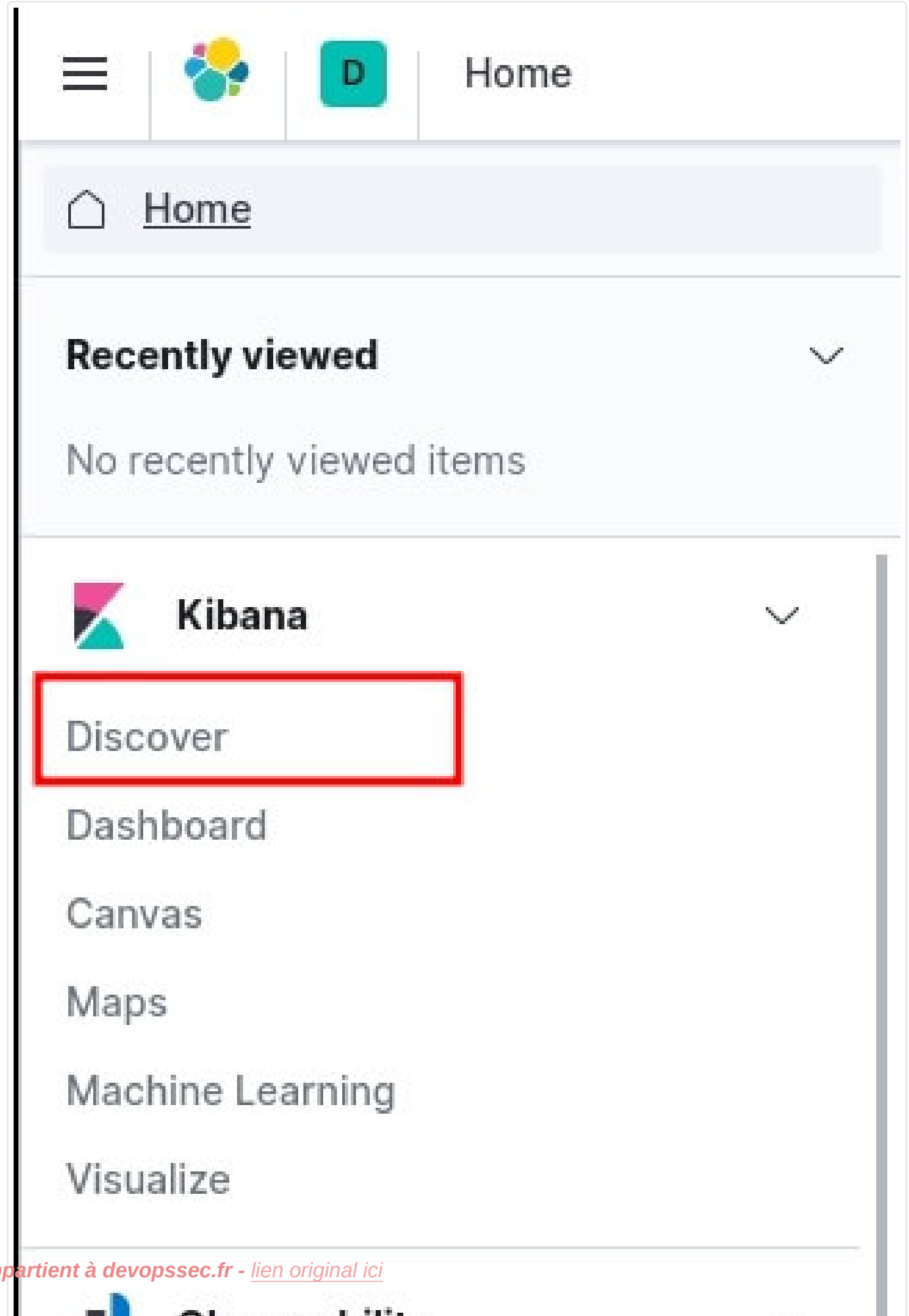
```
sudo filebeat setup
```

Résultat :

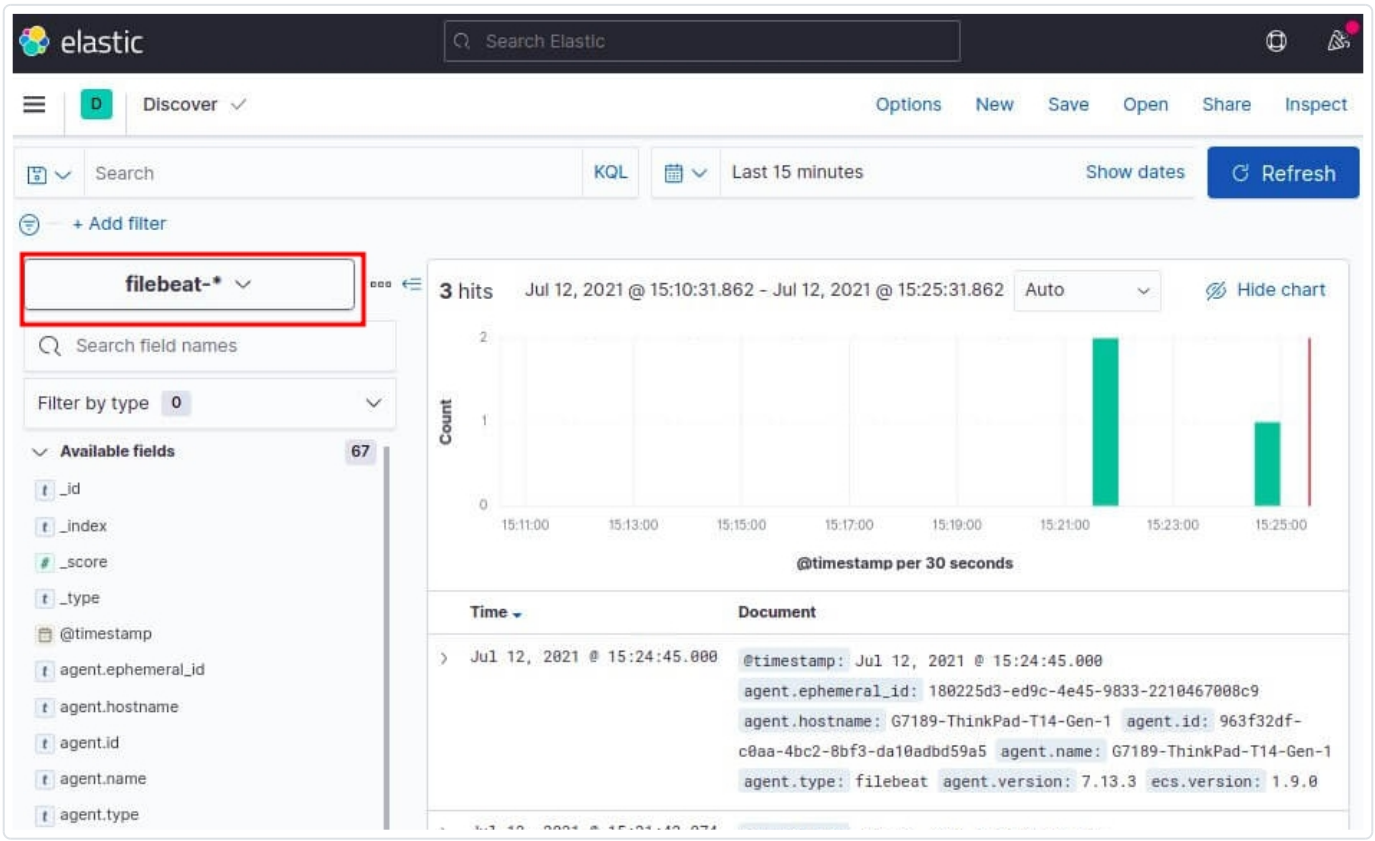
```
Overwriting ILM policy is disabled. Set `setup.ilm.overwrite: true` for enabling.

Index setup finished.
Loading dashboards (Kibana must be running and reachable)
Loaded dashboards
Setting up ML using setup --machine-learning is going to be removed in 8.0.0. Please
See more: https://www.elastic.co/guide/en/machine-learning/current/index.html
Loaded machine learning job configurations
Loaded Ingest pipelines
```

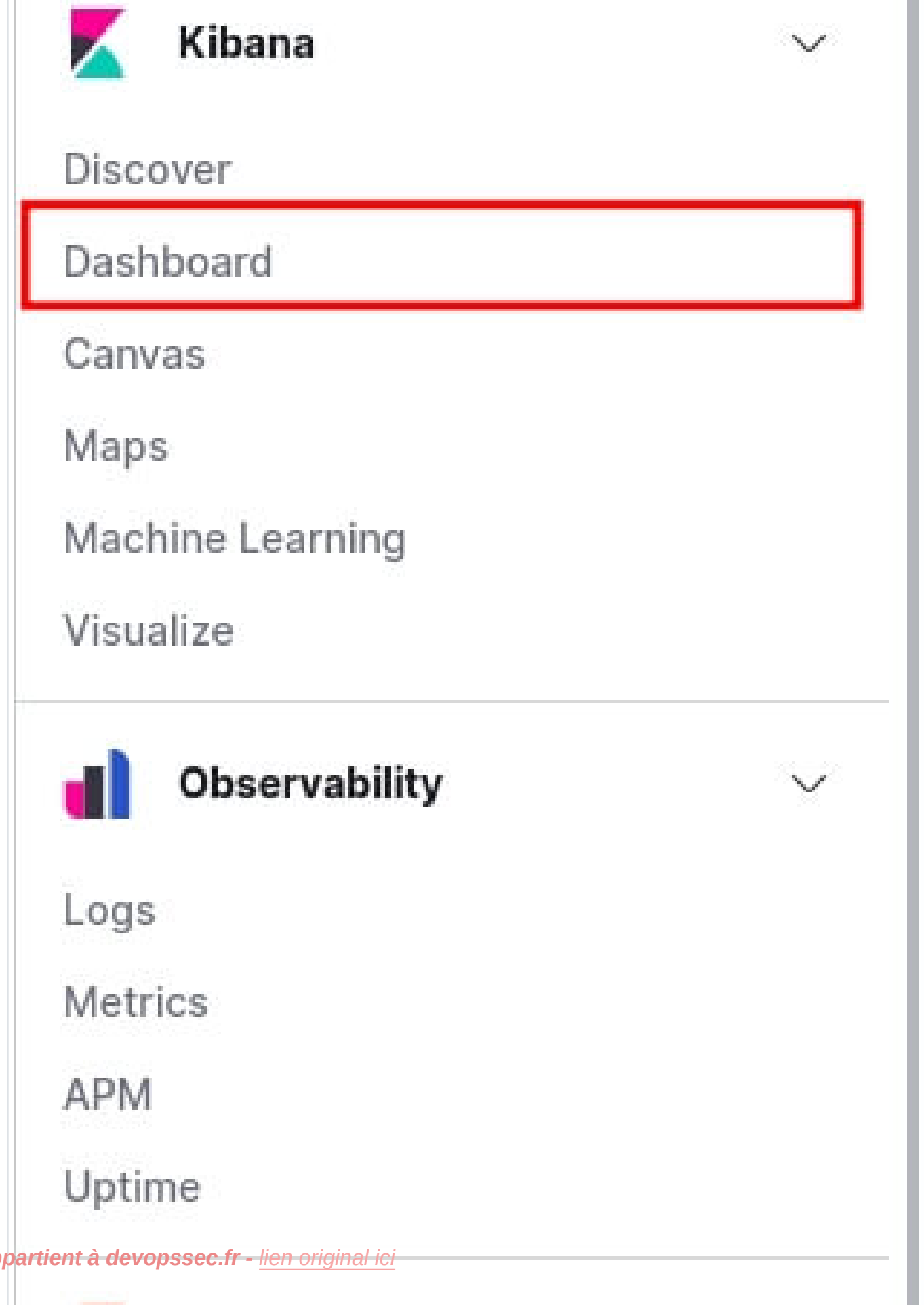
Pour découvrir vos logs, rendez-vous dans http://localhost:5601/ et sur le menu à gauche cliquez sur Discover:



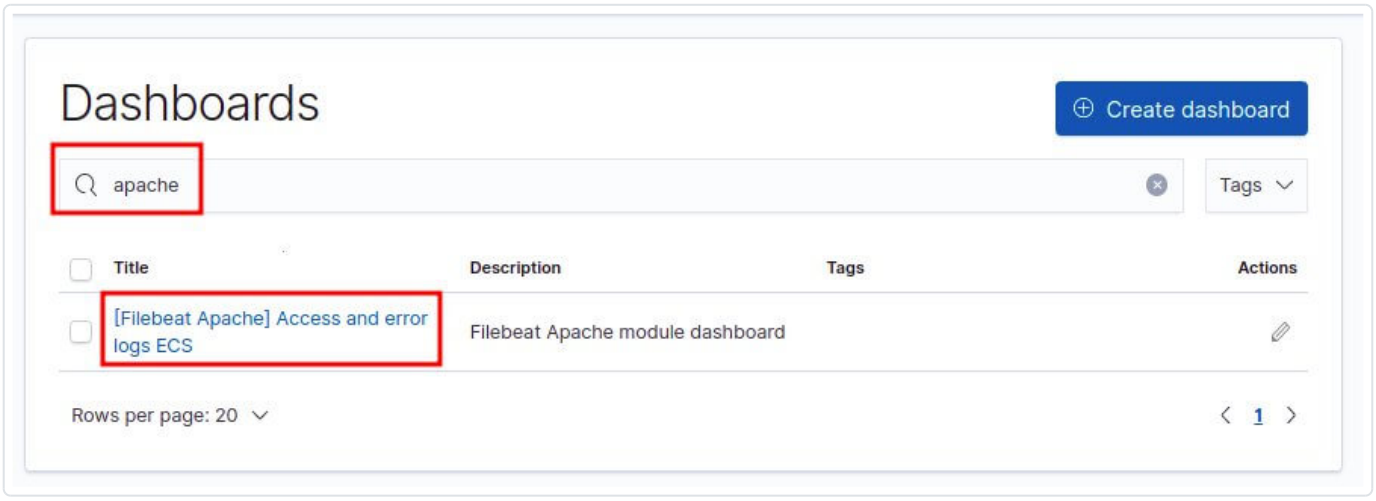
Choisissez ensuite le pattern index `filebeat-*` pour visualiser les logs du module Apache Filebeat:



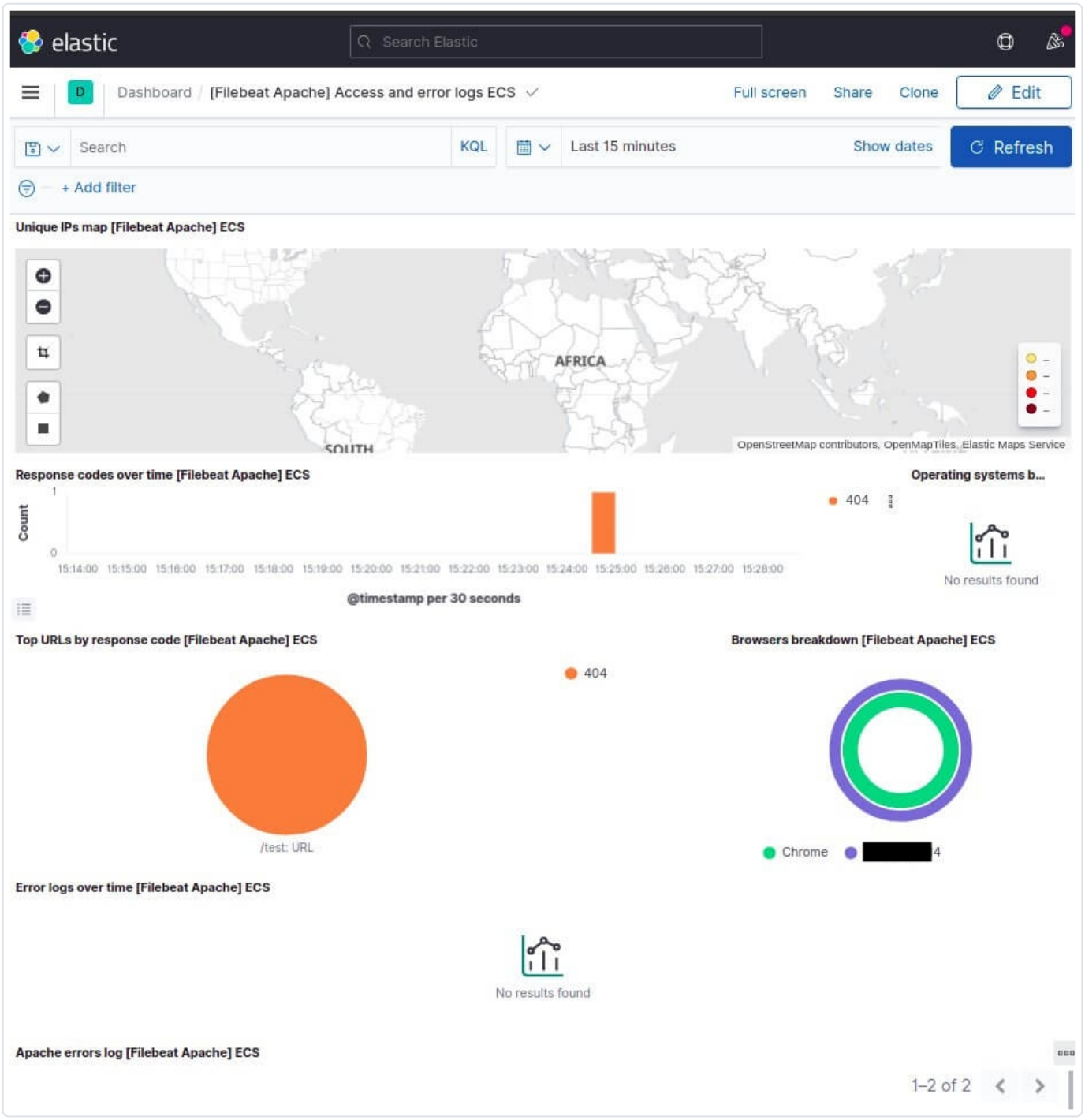
L'étape suivante est de visualiser notre tableau de bord afin de visualiser la collection de visualisations en temps réel issue par notre module Apache Filebeat. Pour ce faire, sur le menu à gauche cliquez sur Dashboard:



Recherchez et cliquez sur les Dashboards Apache:



Vous obtenez ainsi le dashboard suivant:



Dans notre beau dashboard, nous observons:

- Une carte monde qui géolocalise la provenance des requêtes selon les adresses IP sources
- Un graphe du taux de code de réponse HTTP dans le temps

- Des graphiques circulaire avec des statistiques sur les principales URL par code de réponse HTTP
- Un autre sur la répartition des navigateurs
- Un graphe sur les logs d'erreurs au fil du temps
- Puis en dessous comme pour le tableau de bord des logs système, on a les trames des logs que l'on peut dérouler pour avoir plus d'informations, triables par date, par machine, user-agent, ip ou n'importe quels champs.

Les tableaux de bord restent personnalisables, on peut ajouter, supprimer et réorganiser les fonctions des dashboards à notre guise.

Combiner Filebeat et Logstash

Configuration de Filebeat pour envoyer les logs à Logstash

Si vous souhaitez utiliser Logstash pour effectuer un traitement supplémentaire sur les données collectées par Filebeat, vous devez **configurer Filebeat pour utiliser Logstash**.

Avant de créer le pipeline Logstash, vous configurerez Filebeat pour envoyer des lignes de logs à Logstash. Pour cela, modifiez le fichier **/etc/filebeat/filebeat.yml**:

```
- type: log
  # Mettre à true pour activer cette l'entrée Filebeat dans Logstash
  enabled: true
  # Chemins qui doivent être explorés et récupérés
  paths:
    - /var/log/*.log

# On ne souhaite plus envoyer directement nos données à elasticsearch (à commenter co
#output.elasticsearch:
```

```
  #hosts: ["localhost:9200"]

# On souhaite à la place envoyer directement nos données à Logstash (à décommenter co
output.logstash:
  hosts: ["localhost:5044"]
```

Dans ce fichier de configuration, Filebeat enverra tous les logs à l'intérieur du dossier **/var/log/** vers Logstash. Pour prendre en considération notre configuration, on va redémarrer le service Filebeat:

```
sudo systemctl restart filebeat
```

Ensuite créons un fichier **apache.conf** dans le dossier de configuration Logstash situé dans le dossier **/etc/logstash/conf.d/** et rajoutons-y les trois sections de configuration principales:

```
input {
    beats {
        port => 5044
    }
}

filter {
    grok {
        match => { "message" => "%{COMBINEDAPACHELOG}" }
    }

    date {
        match => [ "timestamp" , "dd/MMM/yyyy:HH:mm:ss Z" ]
    }

    mutate {
        convert => {
            "response" => "integer"
            "bytes" => "integer"
        }
    }
}

output {
    stdout { codec => rubydebug }
}
```

En plugin d'entrée nous utilisons Beats qui écoute par défaut sur le port 5044. Pour le plugin de filtre et de sortie nous utilisons ceux déjà expliqués dans le chapitre précédent.

Vérifions déjà si notre configuration Logstash est correcte. Pour cela il faut commencer par stopper le service Logstash:

```
sudo systemctl stop logstash
```

Lançons ensuite notre analyseur de configuration Logstash:

```
sudo /usr/share/logstash/bin/logstash --path.settings /etc/logstash -f /etc/logstash/
```

Résultat :

```
...
Configuration OK
[2021-07-12T19:58:12,834][INFO ][logstash.runner] ... Config Validation Result: OK. E
```

Une fois la syntaxe validée, relançons la commande sans l'option `-t` mais avec l'option `--debug`:

```
sudo /usr/share/logstash/bin/logstash --debug --path.settings /etc/logstash -f /etc/
```

Visitons ensuite depuis notre navigateur la page d'accueil d'Apache http://localhost/ et revenons sur la sortie standard de notre Logstash pour observer le résultat suivant:

```
{
  "timestamp" => "12/Jul/2021:20:03:10 +0200",
 "@timestamp" => 2021-07-20T15:03:10.000Z,
       "tags" => [
    [0] "beats_input_codec_plain_applied"
],
....
   "@version" => "1",
"httpversion" => "1.1",
...
```

```
},
      "event" => {
    "dataset" => "apache.access",
     "module" => "apache"
},
        "ecs" => {
    "version" => "1.9.0"
},
    "fileset" => {
    "name" => "access"
},
   "clientip" => "::1",
      "ident" => "-",
    "service" => {
    "type" => "apache"
},
    "request" => "/",
   "response" => 200,
      "input" => {
    "type" => "log"
},
       "verb" => "GET",
       "auth" => "-",
   "referrer" => "\"-\""
}
```

Le résultat nous dévoile que nous récupérons les bonnes données. Maintenant, modifions notre fichier **apache.conf** pour communiquer désormais avec elasticsearch:

```
input {
  beats {
      port => 5044
  }
}

filter {
  grok {
      match => { "message" => "%{COMBINEDAPACHELOG}" }
  }

  date {
      match => [ "timestamp" , "dd/MMM/yyyy:HH:mm:ss Z" ]
  }

  mutate {
      convert => {
          "response" => "integer"
          "bytes" => "integer"
```

```
      }
  }
}

output {
  elasticsearch {
       hosts => "localhost:9200"
       index => "apache-%{+YYYY.MM.dd}"
  }
}
```

On démarre par la suite le service Logstash avec la commande suivante:

```
sudo systemctl start logstash
```

Après démarrage de notre service, Logstash va lancer automatiquement notre pipeline. Ensuite nous pouvons vérifier si l'index s'est correctement créé, pour cela appelons l'API REST Elasticsearch fournie par elasticsearch, comme suit :

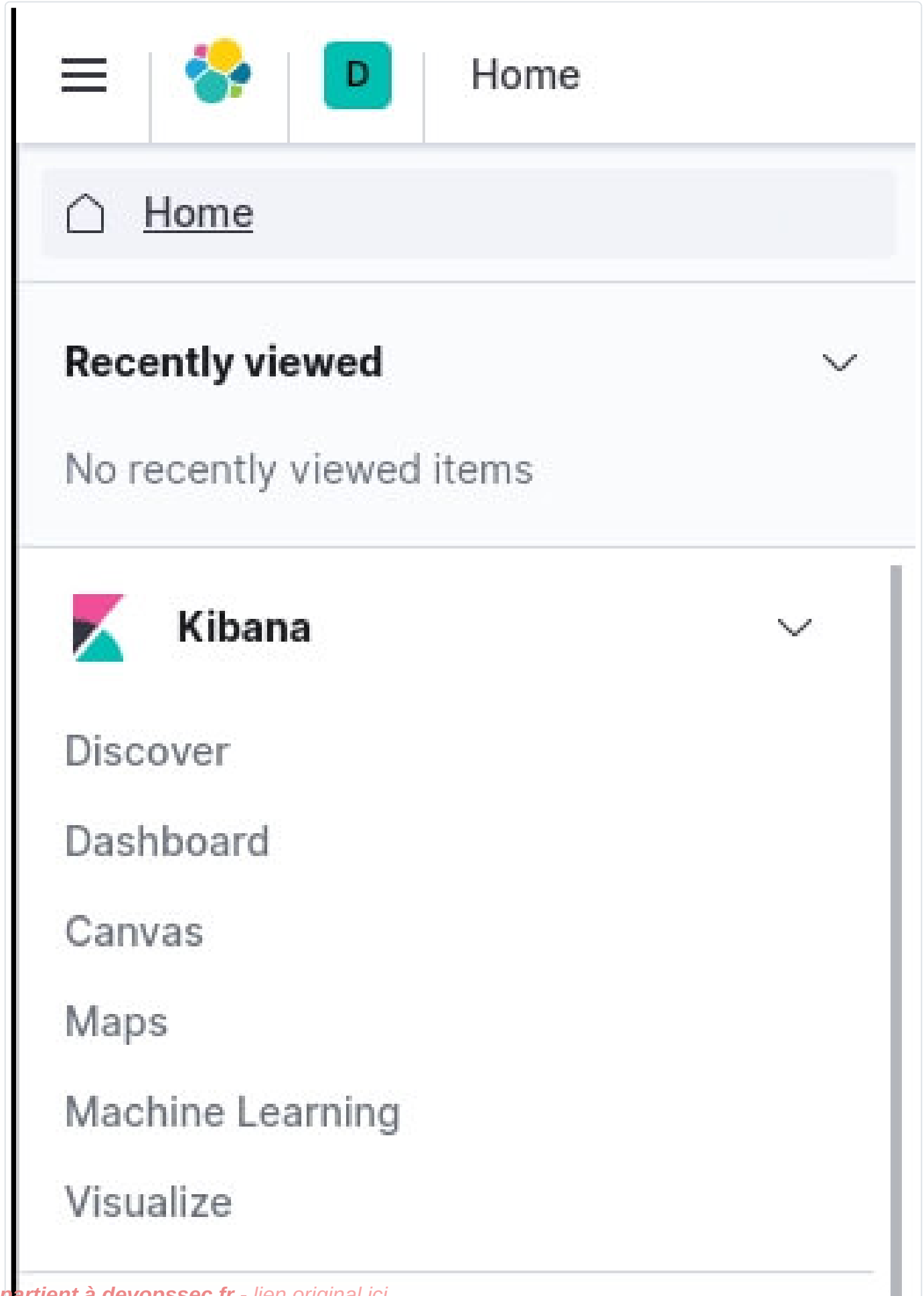
```
curl "localhost:9200/_cat/indices?v"
```

Résultat :

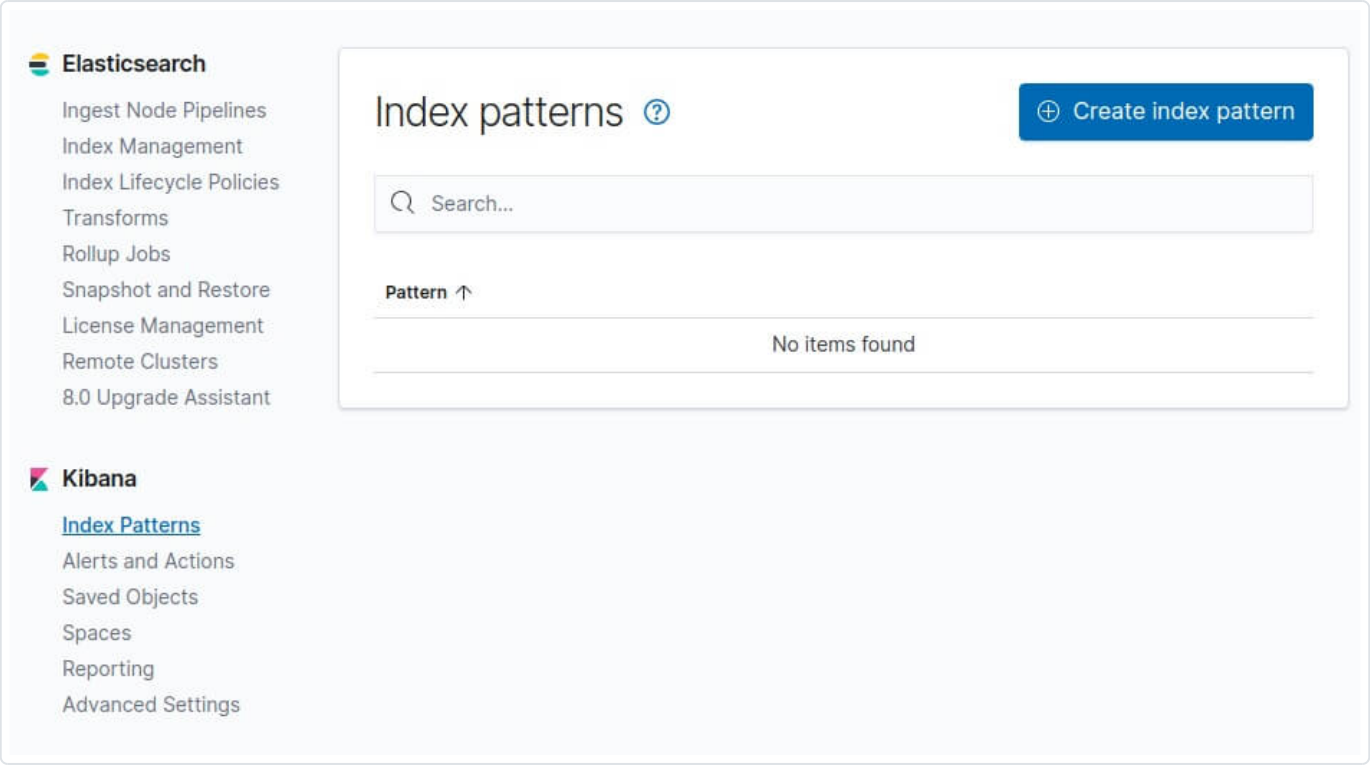
```
yellow open   apache-2021.07.12
```

visualisation dans Kibana

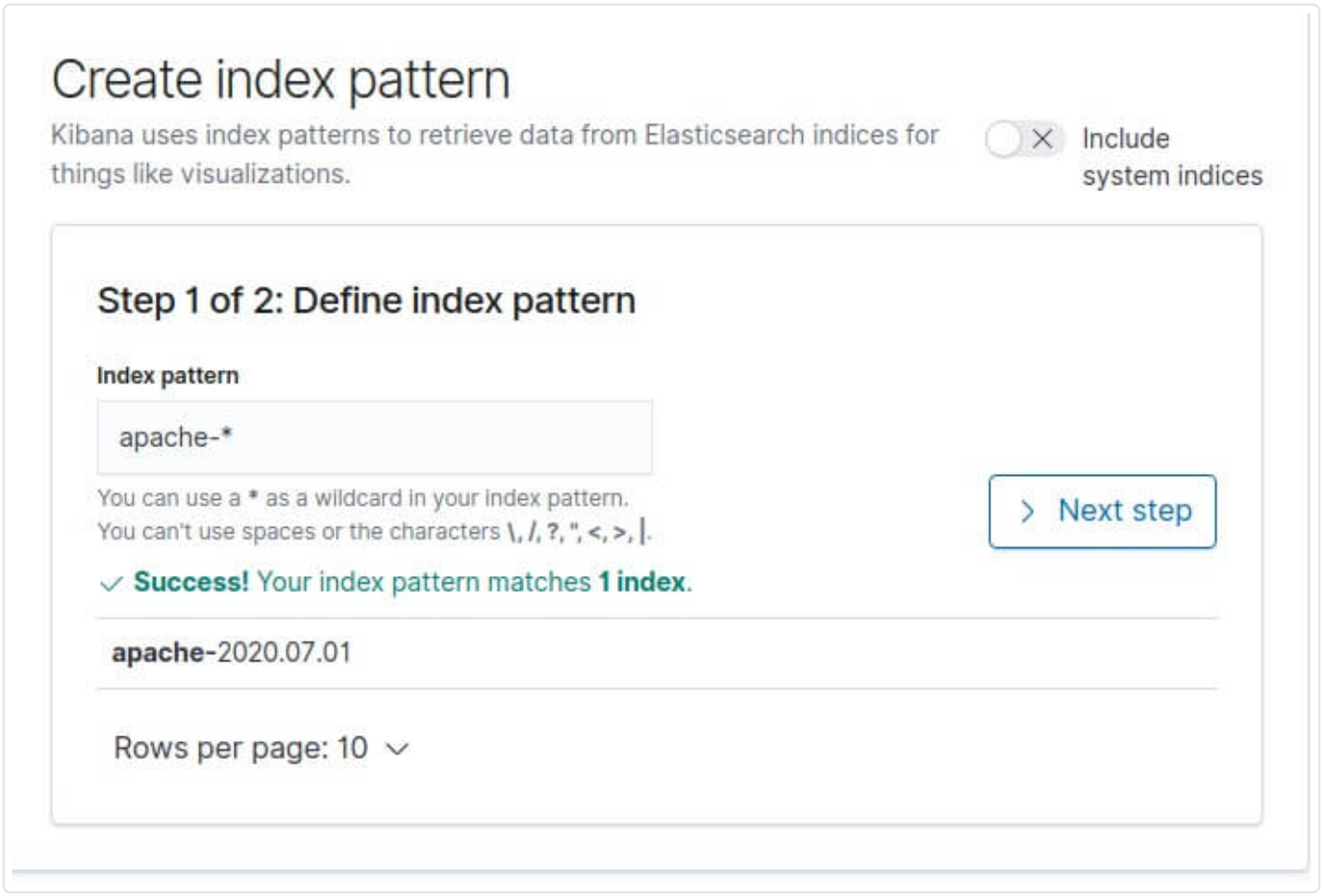
Rendez-vous ensuite sur Kibana depuis l'url http://localhost:5601/ et allons sur le menu à gauche et cliquons sur "Stack Management" pour visualiser notre index:



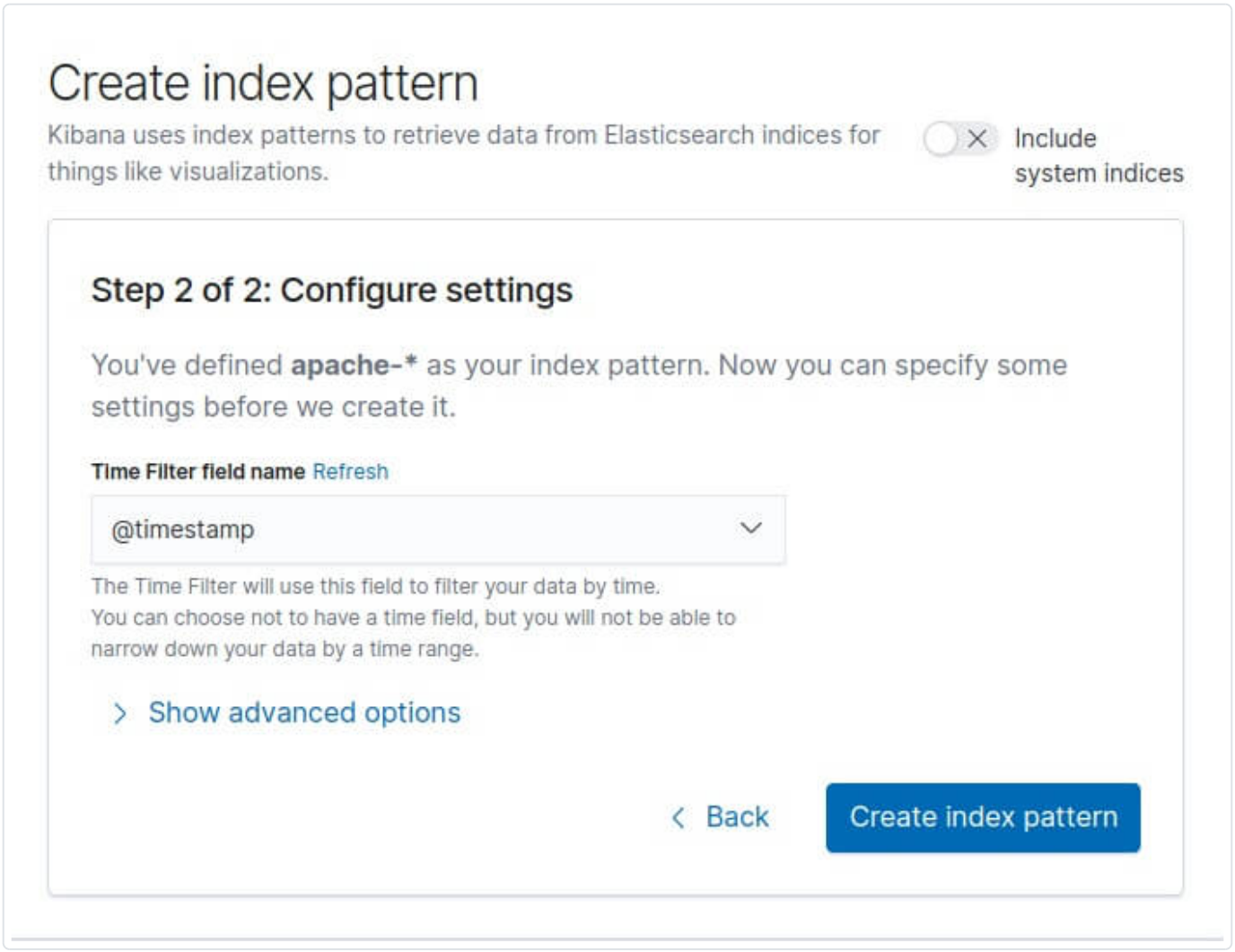
Comme pour le chapitre précédent, nous allons ajouter un nouveau pattern index dans kibana afin de prendre en considération nos indexs quotidiens Apache récupérés par Elasticsearch. Pour ce faire, cliquons sur "Index Patterns" sur le volet à gauche et cliquons sur le bouton "Create index pattern" :



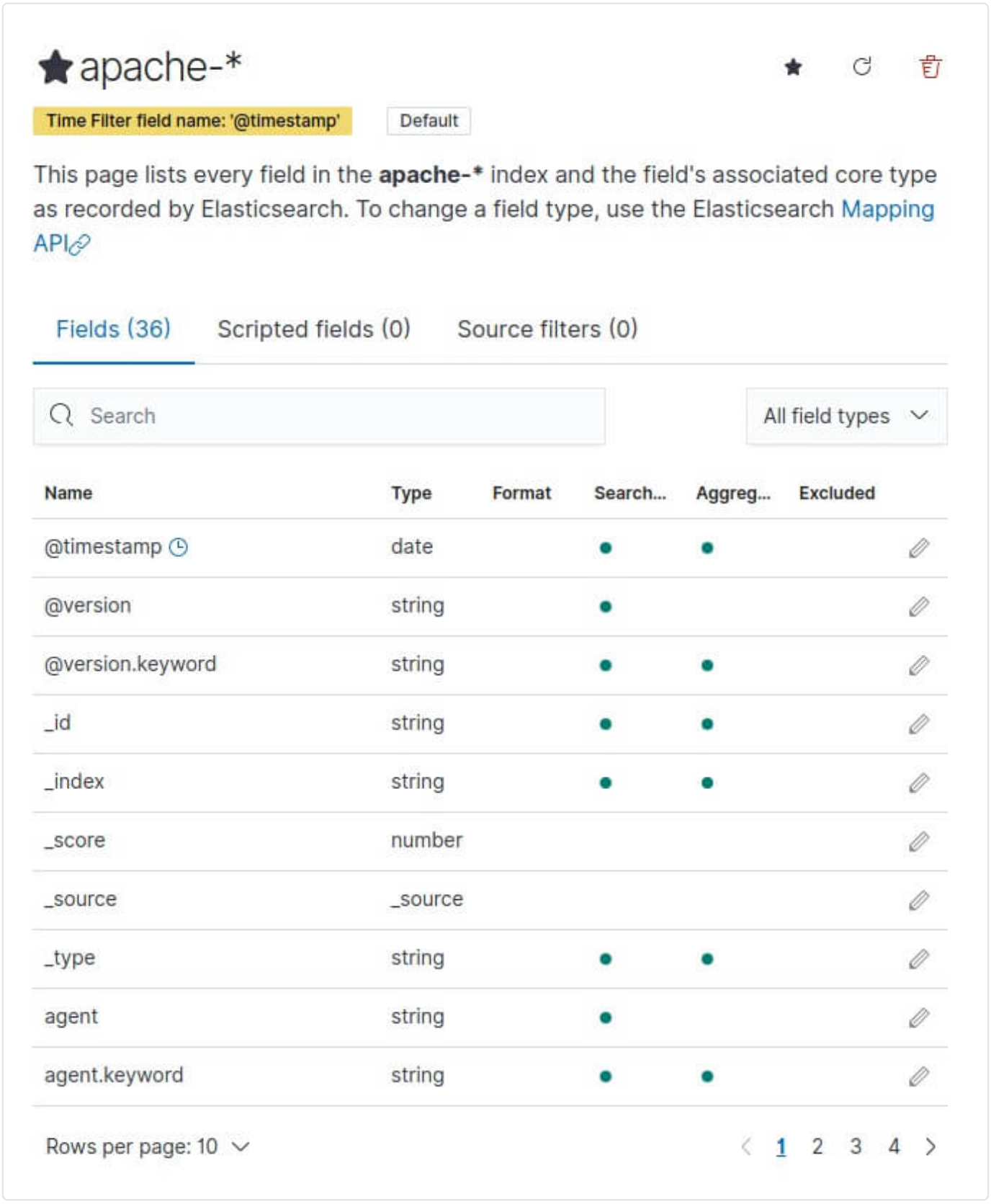
Dans notre cas le préfix de nos indexs apache est "apache-", donc le pattern index à créer dans kibana sera `apache-*` :



Cliquons ensuite sur "Next Step". Ensuite il nous demande par quel moyen il doit gérer le filtre temporel (timestamp) afin d'affiner nos données par plage horaire. Nous sélectionnerons le champ `@timestamp` :

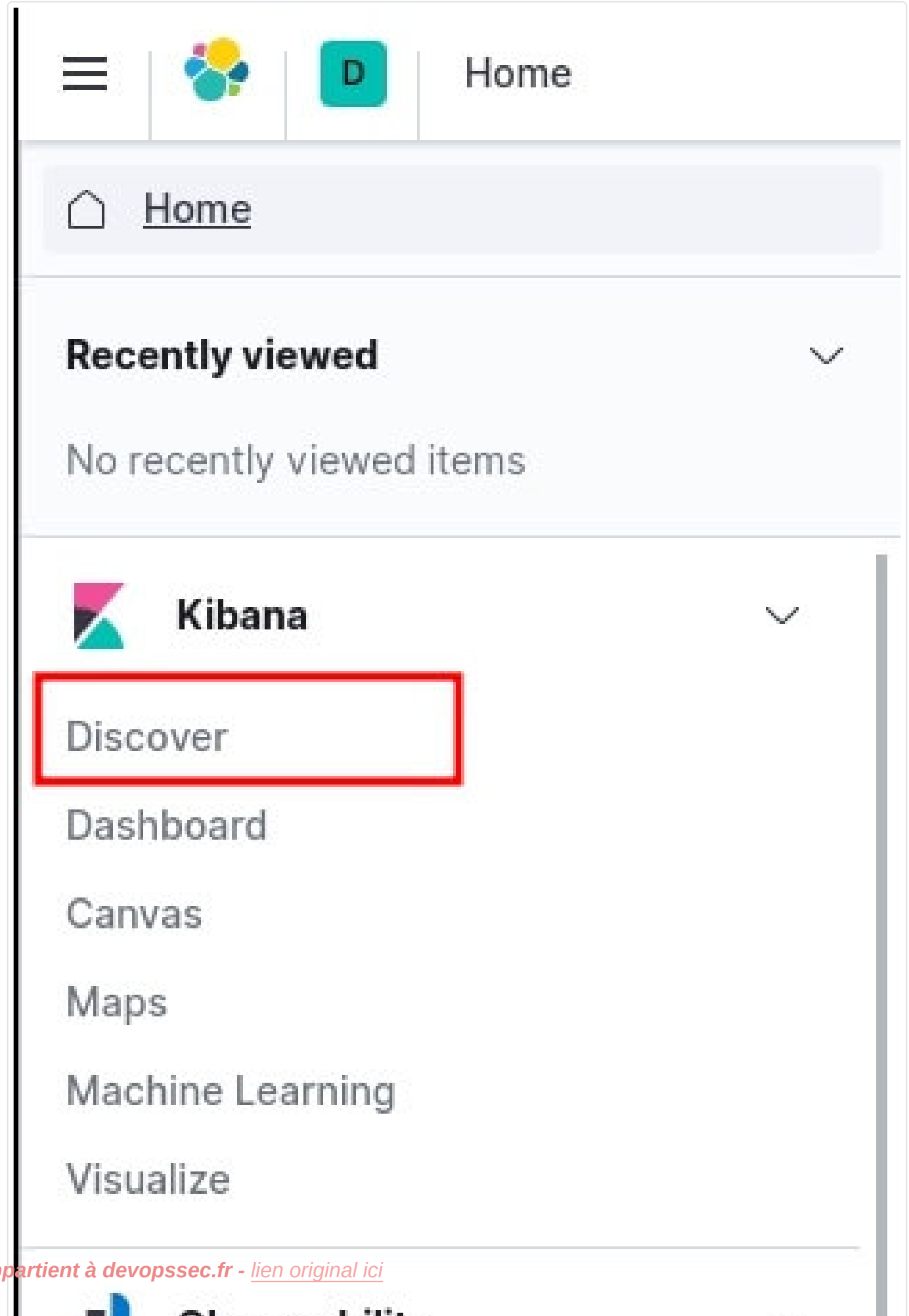


Enfin, cliquons sur le bouton "Create index pattern" et nous verrons apparaître tous nos champs:

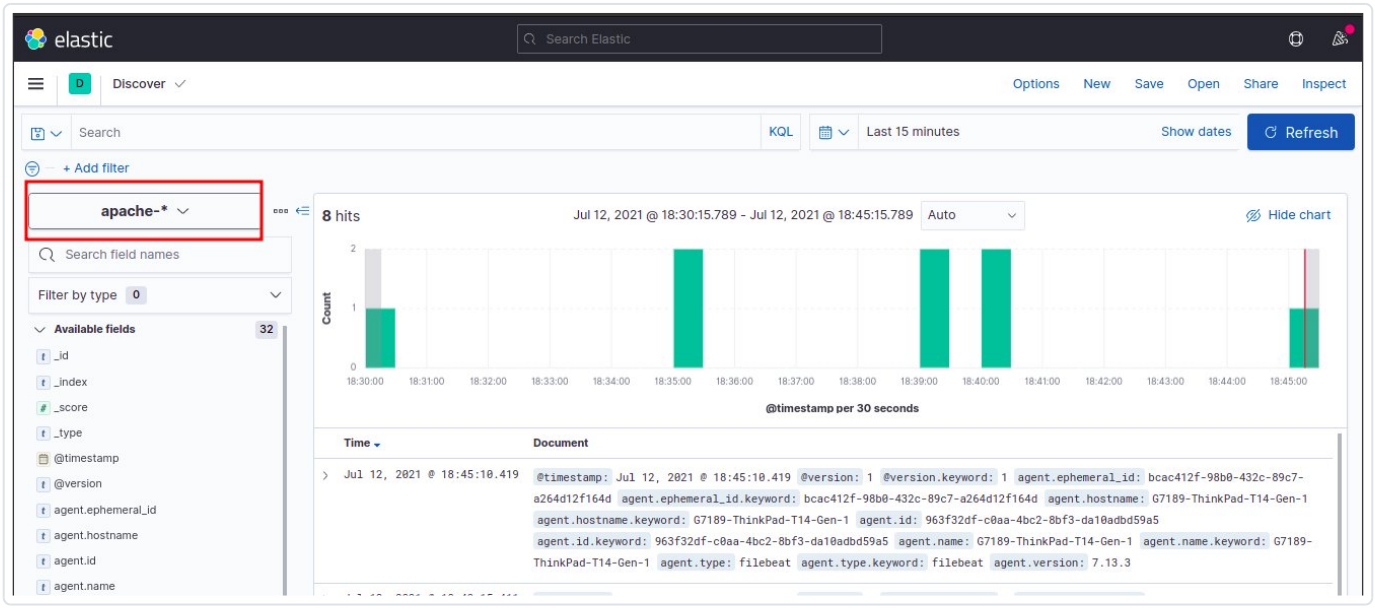


>

Pour découvrir nos logs, il suffit d'aller sur le menu à gauche et de cliquer sur "Discover":



Ensuite, faisons quelques visites depuis notre navigateur sur la page d'accueil d'Apache http://localhost/ et revenons sur la page de Discover en choisissant le bon pattern index `apache-*` :



Conclusion

Une consommation de ressources a fait de Logstash un maillon faible de la pile ELK. Pourtant, malgré ces défauts, Logstash reste un composant crucial de la pile, et ce principalement pour manipuler les données avant de les envoyer à Elasticsearch.

Elastic a fait de grands pas en essayant d'atténuer ces douleurs en introduisant Beats, ce qui a permis aux utilisateurs de créer et de configurer plusieurs pipelines de journalisation résilientes et finalement de **faire la journalisation avec ELK beaucoup plus fiable**.

Dans le prochain chapitre nous aborderons l'expéditeur métrique Metricbeat