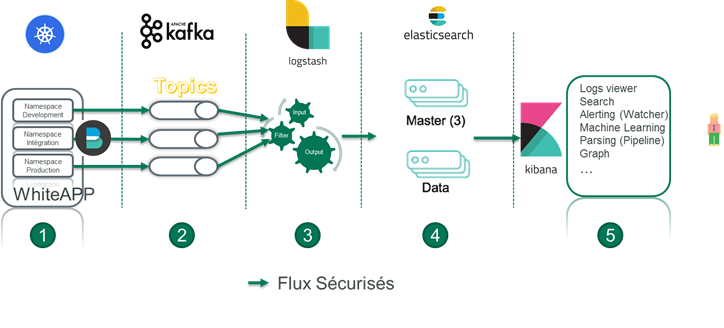
**Kit de Démarrage ELK/CAGIP**

**L'objet de ce Wiki est de vous orienter sur l'utilisation de la Stack Elastic et d'appliquer les bonnes pratiques d'utilisation du Cluster ECE mis à disposition par CAGIP dans le cadre de l'usage par les Squads CATS.**

[](http://elzudeggscas.yres.ytech/file/data/oxxnhtafeijrdwu7bmk7/PHID-FILE-t6dedvgkzmicwacxwzsu/image.png)

**Ce Kit de démarrage doit vous permettre d'implémenter les différentes phases suivantes :**

* Mise en oeuvre de la collecte de vos logs
* Production de DashBoard à partir de Visualization Kibana

**En complément, n'hésitez-pas à vous référer à la documentation en ligne d'Elasticsearch.**  
[Elasticstack and Product Documentation](https://www.elastic.co/guide/index.html)

**Et visionnez les Webinars organisés par Cagip et Elastic!**

[Introduction à la Suite Elastic](http://groove.groove.co/url/X2Dq2VKCbP_IkFVAQPvAGtG8eWU/aHR0cHM6Ly9ldmVudHMuZWxhc3RpYy5jby8yMDE5LTA5LTI0LXdlYmluYXItY3JlZGl0LWFncmljb2xl)

[Webinar dédié aux méthodes d’ingestion des données dans Elasticsearch](http://groove.groove.co/url/X2Dq2VKCbP_IkFVAQPvAGtG8eWU/aHR0cHM6Ly9ldmVudHMuZWxhc3RpYy5jby8yMDE5LTEwLTA4LWVsYXN0aWMtZGF0YS1pbmdlc3Rpb24td2ViaW5hci1jcmVkaXQtYWdyaWNvbGU=)

[***Cas d'usages Elisa***](http://elzudeggscas.yres.ytech/w/kit_demarrage_ece/q_-_cas_d_usages_elk/)  
[***Architecture Elisa CAGIP***](http://elzudeggscas.yres.ytech/w/kit_demarrage_ece/architecture_cagip_elk/)  
[**Organisation Des Différents Objets Elk**](http://elzudeggscas.yres.ytech/w/kit_demarrage_ece/organisation_des_diff%C3%A9rents_objets_elk/)  
[***Elk Les Différentes étapes***](http://elzudeggscas.yres.ytech/w/kit_demarrage_ece/elk_les_diff%C3%A9rentes_%C3%A9tapes/)[***Mise En Oeuvre De Logstash***](http://elzudeggscas.yres.ytech/w/kit_demarrage_ece/mise_en_oeuvre_de_logstash/)[***Création Du Template Elasticsearch***](http://elzudeggscas.yres.ytech/w/kit_demarrage_ece/cr%C3%A9ation_du_template_elasticsearch/)[***Mise En Oeuvre de Filebeat***](http://elzudeggscas.yres.ytech/w/kit_demarrage_ece/mise_en_oeuvre_de_filebeat/)[***Démarrage de l'ingestion Logstash***](http://elzudeggscas.yres.ytech/w/kit_demarrage_ece/d%C3%A9marrage_de_l_ingestion/)[***Démarrage de Filebeat***](http://elzudeggscas.yres.ytech/w/kit_demarrage_ece/d%C3%A9marrage_de_filebeat/)[***Vérification de l'Indexation Elasticsearch***](http://elzudeggscas.yres.ytech/w/kit_demarrage_ece/verification_de_indexation_elasticsearch/)[***Création de l'Index Pattern Kibana***](http://elzudeggscas.yres.ytech/w/kit_demarrage_ece/creation_index_pattern_kibana/)[***Le Discover Kibana***](http://elzudeggscas.yres.ytech/w/kit_demarrage_ece/le_discover_kibana/)[***Création d'une Visualization***](http://elzudeggscas.yres.ytech/w/kit_demarrage_ece/cr%C3%A9ation_d_une_visualization/)[***Création d'un DashBoard***](http://elzudeggscas.yres.ytech/w/kit_demarrage_ece/cr%C3%A9ation_d_un_dashboard/)[***Exemple De Requête Rest Elasticsearch***](http://elzudeggscas.yres.ytech/w/kit_demarrage_ece/exemple_de_requ%C3%AAte_rest_elasticsearch/)[***Procédure d'installation manuelle Logstash+EL+K***](http://elzudeggscas.yres.ytech/w/kit_demarrage_ece/procedure_installation_manuelle/)[***Mise en configuration des composants***](http://elzudeggscas.yres.ytech/w/kit_demarrage_ece/o_-_mise_en_configuration_rtc_elk/)[***Déploiement des composants***](http://elzudeggscas.yres.ytech/w/kit_demarrage_ece/p_-_mise_en_deploiement_elk/)[***Limites d'ELK et contournements possibles***](http://elzudeggscas.yres.ytech/w/kit_demarrage_ece/limites_delk_et_contournements_possibles/)

**Cas d'usages Elk**

### Pour Qui?

Les PP métiers

### Pour Quoi?

Effectuer de la centralisation et de l'analyse de logs.  
Ce besoin est aujourd'hui renforcé avec les architectures réparties et les approches micro-services.

ELISA est la solution proposée pour répondre aux usages liés à l'analyse de logs

* Analyse d'incident
  + Effectuer une recherche sur mes logs quelques soit le composant de mon application
* Suivi technique des services/applications déployées => KPI produit
  + Temps de réponse en fonction de critères techniques
  + Suivre les causes d'erreurs de mon application
* Analyse de qualité
  + Nombre de consultation de fonctionnalités
  + Mesure de fonctionnement
  + Analyse de tendances

[](http://elzudeggscas.yres.ytech/file/data/xrckf6zxqmfjv6cwow5m/PHID-FILE-3js6cturoac246e5hpg3/image.png)

### Ce que ne permet pas ELISA?

Ce n'est pas un outil d'analytique et d'exploration métier => Il est centré sur un usage IT  
N'est pas destiné à un stockage des logs pour une longue durée => 1 an maximum

### Couverture fonctionnelle

**Exposition**

* Kibana pour la visualisation des données

**Ingestion**

* Fichier de logs => via Kafka

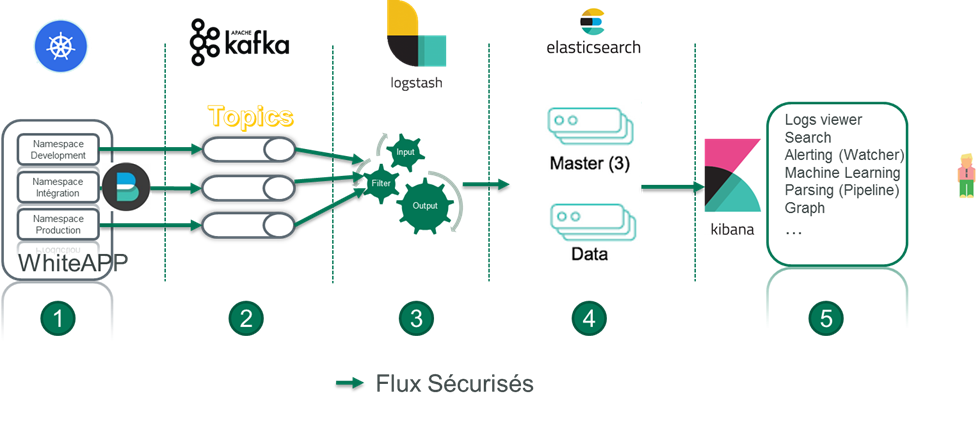
**Transformation de données**

* Découpage des logs en évènement
* Filtrage
* Split => 1 vers n évènements
* Calculs mathématiques simples

**A - Architecture ELK / CAGIP**

**L'architecture ECE / ELK CAGIP**

CAGIP met à disposition un Cluster ECE 2.2.2 comprenant la version 7.4 d'ElasticSearch.  
L'architecture est hautement disponible, les noeuds étant répartis sur les 2 sites du PSI.  
La scalabilité est assurée par ajout de noeuds, dockérisés logstash, elasticsearch et kibana.  
Kafka permet également la scalabilité par ajout de noeuds au cluster Kafka.

[](http://elzudeggscas.yres.ytech/file/data/mhtxzkn7suscuirmq4q4/PHID-FILE-decluodgfy34qg3t7ivo/image.png)

* le cluster ELK/CAGIP -

Le principe est le suivant :  
Le logshipper ou agent de collecte des fichiers de logs produit les lignes de logs dans un topic kafka dédié  
Logstash est l'ETL permettant de consommer les logs depuis ce topic kafka, de traiter chaque ligne de log en appliquant des règles de parsing et d'indexer le résultat dans un index elasticsearch  
Elasticsearch est le support de persistance des logs dans les collections que l'on appelle index.  
Enfin Kibana permet d'explorer le contenu de ces index elasticsearch, de créer des DashBoards composés de Visualizations à partir d'index-pattern regroupant plusieurs index elasticsearch de même nature (même format).

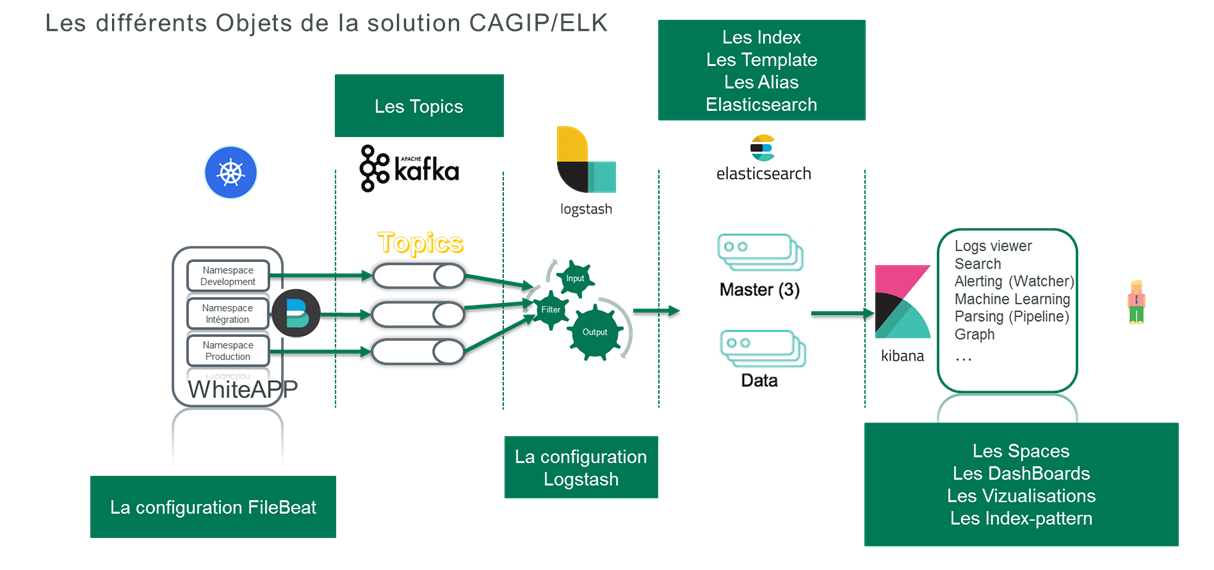
Elasticsearch est la solution d'analyse de logs la plus utilisée du marché, de nombreux sites, notamment celui de l'éditeur, vous permettrons d'appronfondir le sujet.

L'objet de ce Wiki est de vous orienter sur l'utilisation de la Stack Elastic et d'appliquer les bonnes pratiques d'utilisation dans le cadre de CATS.  
En particulier, veillez à bien respecter les normes de nommage des différents objets et les appliquer les pattern qui sont imposés dans le cadre de l'utilisation de cette plateforme.  
Pour donner un simple exemple, il est important de pouvoir identifier clairement le contenu d'un index à partir de la norme de nommage :  
ix\_[adabo]\_[produit]\_[typelog]\_[appl]-${yyyy-MM-dd} est un index contenant des logs du yyyy-MM-dd, de l'environnement adabo, émise par une application APPL composante du produit dont le format est caractérisé par un typelog.

Le WIki proposant la démarche permettant d'aboutir à la mise en oeuvre dans ELK de votre cas d'usage, ces différents points seront détaillés par suite.  
**La démarche de mise en oeuvre de l'ingestion des logs dans ELK**

**Organisation Des Différents Objets Elk**

**Voici un détail des différents objets manipulés dans la Stack ELK**

[](http://elzudeggscas.yres.ytech/file/data/laaaj3yb6khfoecf2pza/PHID-FILE-hnrmmri3yqpsmhgy4ldq/image.png)

|  |
| --- |
| **Type Objet** |
| File Beat Configuration | fichier de configuration au format yml permettant de collecter les logs à la source |
| Topic Kafka | Permet la bufferisation des logs émises par le logshipper, consommés par logstash |
| Pipeline Logstash | Définit les Entrées, les sorties et les règles de filtrage |
| Index ElasticSearch | C’est comme une collection SOLR (lucene), ça contient les logs |
| Template ElasticSearch | Définit les règles à appliquer sur un ou plusieurs pattern de nom d’inde, Replica, Sharding, Mapping (type de champs, docvalue, keyword…) |
| Alias ElasticSearch | Un Alias, c’est comme une vue. Il permet de faire de l’union entre index, du filtering… |
| Space Kibana | Espace de développement et d’exposition des Objets Kibana |
| Index-Pattern Kibana | Définit les pattern d’index ou d’alias elasticsearch que l’on souhaite explorer, de type timeseries (time filter) ou non |
| Discover Kibana | Permet de visualiser, de requêter, de filtrer les logs d’un index-Pattern Kibana. Exporter au format CSV… |
| Visualization Kibana | Permet la représentation graphique de données agrégées ou non à partir de modèle |
| DashBoard Kibana | Composition d’une ou plusieurs Visualization |
|  |  |

**Voici la normalisation proposée des différents Objets ELK**  
La normalisation prend en compte les notions suivantes :

Type Objet / Code Environnement / Code Produit / Famille de Format de Log / Code Application

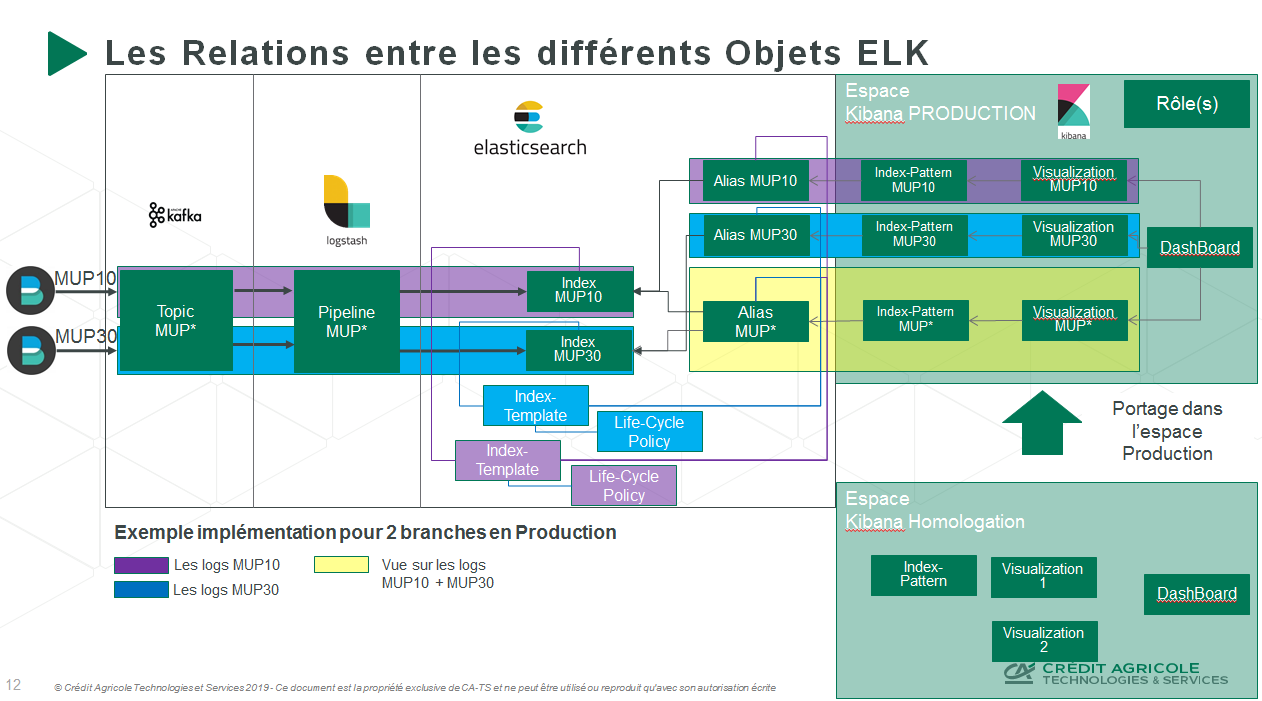
Type Objet : Préfixe du nom de l’Objet : FB : fileBeat / LS : Logstash / TE : template Elasticsearch / IX : Index Elastic / AL : Alias Elastic / IX ou AL : Index Pattern Kibana /

|  |  |
| --- | --- |
| [ENV] | {dev, zur, mup} Le code environnement permettra l’étanchéité des données entre le Dev, l’Homologation et la Production, le code environnement n’est pas le code ADABO |
| [ADABO] | Le code Adabo de l’instance de l’application |
| [PRODUIT] | Le code Produit est celui délivré par MAPS, le produit appartenant à une Tribu/Squad |
| [APPL] | Le code Application, étant une composante du produit, il permet d’identifier la source productrice de la log. Exemple MDC9 qui est le code application de l’API MesDocuments |
| [TYPELOG] | La Famille de format de Log a la même signification que le code APP-Name déjà utilisé dans le cadre de la Plateforme Synapse, il permet de catégoriser par format de log, par exemple les logs http accesslog, les logs squid,…. |
|  |  |

**Ce qui donne :**

|  |  |
| --- | --- |
|  | **Attention bien respecter la case (ex : les index sont impérativement en minuscules)** |
| File Beat Configuration | fb\_[adabo]\_[produit]\_[appl]\_[typelog].yml |
| Topic Kafka | cats.elisa.[produit].[env].[produit].[appl].[typelog] |
| Pipeline Logstash | pl\_[env]\_[produit]\_[appl]\_[typelog] |
| Index ElasticSearch | ix\_[adabo]\_[produit]\_[appl]\_[typelog]-${yyyy-MM-dd} |
| Template ElasticSearch | te\_[adabo]\_[produit]\_[appl]\_[typelog] |
| Alias ElasticSearch | al\_[adabo]\_[produit]\_[appl]\_[typelog] (alias sur les index d’une instance adabo du produit) al\_[env]\_[produit]\_[appl]\_[typelog] (alias sur les index d’un environnement du produit) al\_[adabo]\_[libelle] (alias libre permettant le croisement de logs d’une instance adabo) al\_[env]\_[libelle] (alias libre permettant le croisement de logs d’une instance env) |
| Index-Pattern Kibana | Même Norme que les alias ElasticSearch |
| (id) Space Kibana | sp\_[env]\_[produit]\_[libelle] |
| Visualization Kibana | vk\_[adabo]\_[libelle] vk\_[env]\_[libelle] |
| DashBoard Kibana | dh\_[adabo]\_[libelle] dh\_[env]\_[libelle] |
| Rôle utilisateur | à définir |
| Groupe Utilisateur AD | GG\_[ENTITE]\_A\_[PROPRIETAIRE]\_D5?\_ELK\_UTI à définir GG\_[ENTITE]\_A\_[PROPRIETAIRE]\_D5?\_ELK\_ADM GG\_[ENTITE]\_A\_[PROPRIETAIRE]\_D5?\_ELK\_GUEST à définir |

Cette normalisation permet de bien identifier les différents types d'objets manipulés, en particulier de connaître par cette nomenclature le contenu des index elasticsearch

[](http://elzudeggscas.yres.ytech/file/data/rc27kv7e34zcvvk774qv/PHID-FILE-vobjwsmjh7f3zkp4j7r2/image.png)

{anchor #organisation\_des\_différents\_objets\_elk/}

**C - Elk Les Différentes étapes**

**La première étape : identifier le format de log et mettre en oeuvre le parsing de ces logs**

[](http://elzudeggscas.yres.ytech/file/data/6llwfiammm7lhbwjdf5r/PHID-FILE-qecfgvb33rlzxdprqhf2/image.png)

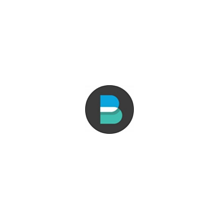
Pour mettre en oeuvre une ingestion de logs, il faut au préalable se pencher sur le format de la log produites dans les fichiers par l'Application.  
L'AEG (Architecture Entreprise Groupe) a publié un document de recommandation sur le format des logs permettant d'en faciliter le parsing.  
Cependant il existe des formats de log bien différents, issus des formats middleware, de progiciel ou d'application CATS qui peuvent être bien loin du format préconisé.  
Logstash va permettre d'appliquer des règles de parsing propre au format des logs que vous souhaitez ingérés dans un index elasticsearch.

**La deuxième étape : définir la structure de l'index elasticsearch et créer l'index template**

[](http://elzudeggscas.yres.ytech/file/data/x4ix2vayxq7hvef7jgsd/PHID-FILE-uafahqhk5anczzwmieaq/image.png)

Il est nécessaire de définir les règles d'indexation, nous verrons qu'elles seront utilisées par logstash, lorsque celui-ci indexera dans un indice elasticsearch.  
C'est l'index Template d'elasticsearch qui permet de définir ces règles. En particulier il permet de poser des alias sur les index, à la manière d'une vue pour une base de données. Ce point est important car nous préconisons ce découplage entre l'index et l'exposition que l'on en fait dans les Visualization Kibana. Vous en retrouverez le détail plus dans l'implémentation.

**La Troisième étape : collecter les logs mise en oeuvre du LogShipper**

[](http://elzudeggscas.yres.ytech/file/data/uw4d6nziftpjnuxzdk6o/PHID-FILE-bihha4as76rvk5b4sxtf/image.png)

Le log Shipper est l'agent de collecte à déployer sur les sources produisants les logs que l'on veut indexer.  
Avant de le déployer sur l'infra applicative de l'environnement de votre application, il est nécessaire de produire le fichier de paramétrage.  
La source de collecte, c'est à dire le path sur lequel les fichiers de logs sont persistés, les tags (attributs) obligatoires à renseigner et la cible d'envoi, c'est à dire le Topic Kafka, point d'entrée de Logstash.

**L'ultime étape avant de pouvoir visualiser les logs dans Kibana : création de l'index Pattern**

[](http://elzudeggscas.yres.ytech/file/data/luounprcz3twybllbcrq/PHID-FILE-enlbkc3v34xfvi5e4ojy/image.png)

Afin de pouvoir visualiser vos logs dans Kibana, il est nécessaire de créer L'index Pattern Kibana.  
L'objectif de l'Index Pattern est d'indiquer à Kibana quels index elasticsearch vous voulez explorer. Il peut regrouper un ou plusieurs index en se basant sur une regex de reconnaissance des index ou des alias elasticsearch.  
Nous verrons encore une fois que nous préconisons (voire nous imposons) la création d'index pattern kibana sur la base d'alias elasticsearch.

**Mise En Oeuvre De Logstash**

**Logstash : Mise en oeuvre du paramétrage d'un Pipeline**  
Le paramétrage d'un pipeline consiste à renseigner les trois sections suivantes :

* input { }
* filter { }
* output { }

**La section input**  
La section input permet de paramétrer la source sur laquelle logstash va appliquer les règles de filtrage.  
Dans la solution Cagip/ELK, c'est Kafka, ainsi il nous faut renseigner les paramétres entre double accolade, dans la section suivante :

Les valeurs seront données par CAGIP :

* SERVEUR\_BOOTSTRAP\_KAFKA : dns:port de tous les serveurs Kafka composants le cluster CAGIP séparés par une virgule.
* KAFKA\_CONS\_GROUPID : le groupe id du consummer Kafka
* Puis les valeurs ada (3 premières lettres de l'adabo), codeproduit, typelog, codeappl permettant de constituer le nom du topic kafka

input {

kafka {

bootstrap\_servers => "{{SERVEUR\_BOOTSTRAP\_KAFKA}}"

decorate\_events => true

group\_id => "{{KAFKA\_CONS\_GROUPID}}"

topics\_pattern => ["cats.elisa.{{ada}}.{{codeproduit}}.{{typelog}}.{{codeappl}}"]

codec => "json"

security\_protocol => "SASL\_PLAINTEXT"

sasl\_mechanism => "PLAIN"

jaas\_path => "/usr/share/logstash/config/jaas-kafka-client.conf"

enable\_auto\_commit => "false"

auto\_offset\_reset => "earliest"

}

}

A noter que le nom du topic doit évidemment être le même dans la configuration filebeat que l'on fera par la suite.  
Le typelog caractérise la famille de log, par exemple dans Synapse, nous avons caractérisé les logs de performance de l'Apimanager par une code famille APIM-PERF, il s'agit ici de la même notion.  
Par exemple, pour les fichiers accesslog produit par le serveur http, typelog = access.  
Pour des logs issus d'une infra technique de type squid, on mettra....typelog=squid  
La nomenclature reste à définir.

**La section filter**

* filter { }

La partie filter permet le découpage de la log consommée depuis Kafka en appliquant des règles de parsing et d'enrichissement de chacun des messages de logs.  
Pour cela il s'appuie sur un langage d'analyse syntaxique basé sur deux filtres principaux :

* le dissect qui se base sur des séparateurs
* le grok qui lui se base sur des patterns d'expressions régulières.

grok est largement répandu et utilisé et propose un ensemble de pattern d'expressions régulières déjà prêtes à l'emploi.  
Voici un site qui peut vous aider à utiliser les différents patterns grok disponibles :  
[Grok Patterns - Lucidworks Documentation](https://doc.lucidworks.com/fusion-server/4.2/reference-guides/parser-stages/grok-patterns.html)

Voici un autre site qui peut vous permettre de développer et de tester vos pattern Grok :  
[grokconstructor](https://grokconstructor.appspot.com/do/construction)

L'exemple ci dessous permet de mettre en oeuvre le parsing d'un log produite par l'API DSP2

[2019/02/21 17:00:47,099] [PERF] [CR:88200] [idPart:00000021521027] [88200e2ed4545-4afb-4c66-b1e8-f203956793aa33333630-3936-3332-3030-343935303730] [fr.ca.cat.npc.synthese.servlets.SyntheseLoadKPIServlet] [Socle AEM:NPC:1.0] [verbeHttp:POST] [consommateur:NPC:1.0] [consommateurOrigine:NPC:60205900062] [acteur:00000021521027] [distribCanalID:] [userTypeCASA:0] [bankNetwork:] - [MOSTCODE=IDENT\_SYNTHESE\_NPC] Temps Load Synthese||4677 (ms)

filter {

grok {

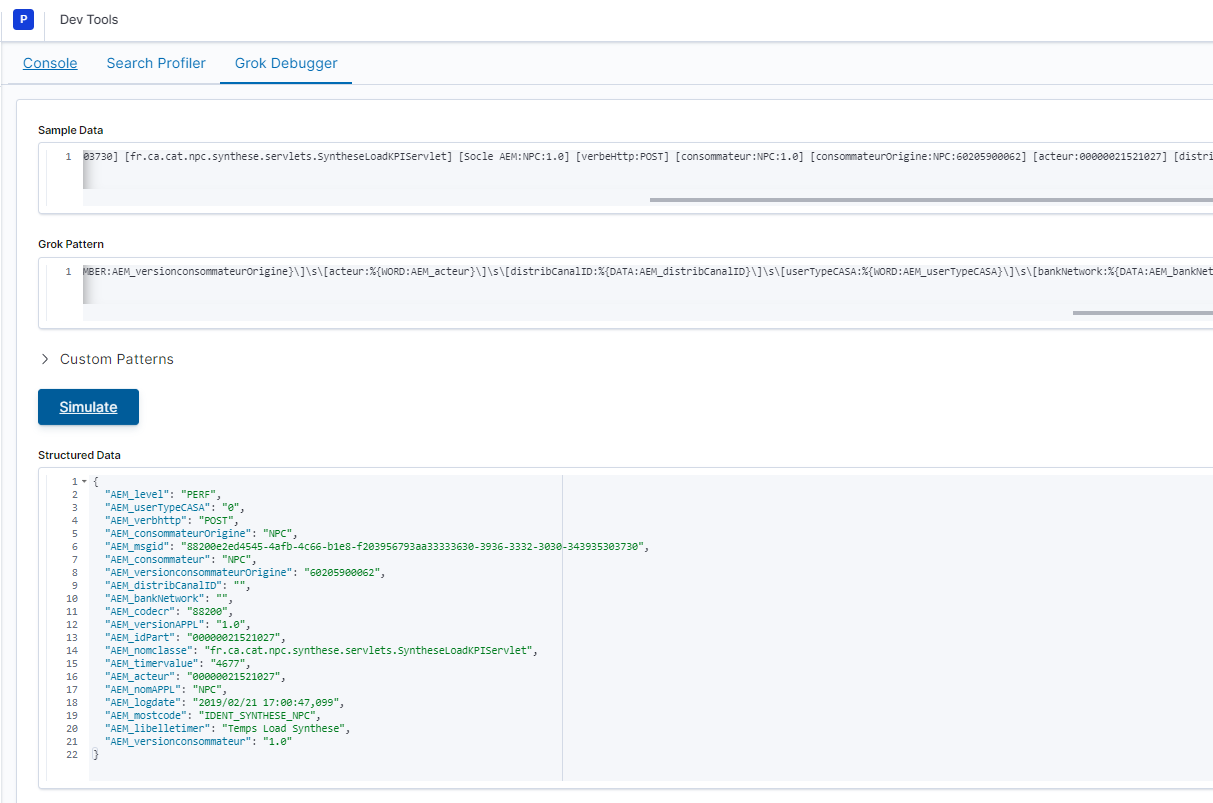
match => {"message" => "%{TIMESTAMP\_ISO8601:dsp2\_logdate}]\s+%{LOGLEVEL:dsp2\_level} \-%{SPACE}%{WORD:dsp2\_direction}:%{SPACE}%{NOTSPACE:dsp2\_url}\|( WSAction:%{DATA:dsp2\_wasaction}%{SPACE}\| SOAPAction:%{DATA:dsp2\_soapaction}\| | %{SPACE})MessageID: urn:uuid:%{UUID:dsp2\_msgid}%{SPACE}\| Direction: %{WORD:dsp2\_typeEchange}%{GREEDYDATA:dsp2\_rest}"}

}

}

Ainsi la règle de filtrage permet de découper l'attribut "message" en autant d'attributs précisés dans le parsing dans les différents patterns grok

**Comment tester un parsing grok dans Kibana ?**  
Il existe de nombreux parser grok online, mais le plus efficace est celui proposé dans l'UI Kibana dans la section DevTool

[](http://elzudeggscas.yres.ytech/file/data/zukuk4koqvi2tyck6w75/PHID-FILE-itvgzth3isvu5we53sj2/image.png)

La mise au point de la séquence grok de parsing de la log DSP2 se fait aisément à partir d'une log sample.  
Le bouton simulate permet d'obtenir le résultat du parsing!

**ajout d'attributs obligatoire dans la section filter**  
Certains attributs nécessaires sont à ajouter afin de définir le nom de l'index elasticsearch dans la section output, nous le verrons plus tard :  
L'année AAAA  
Le mois MM  
Le jour JJ  
Car souvenez-vous, nous créons un index elasticsearch par jour pour chaque type de log, produit, adabo, appl.  
La section fingerprint calcule un identifiant unique (key value) à partir du contenu du message de log. L'identifiant est utilisé plus loin dans la section output pour garantir l'unicité du document dans l'index elasticsearch.

filter {

grok {

match => {"message" => "%{TIMESTAMP\_ISO8601:timestamp}]\s+%{LOGLEVEL:level} \-%{SPACE}%{WORD:dsp2\_direction}:%{SPACE}%{NOTSPACE:dsp2\_url}\|( WSAction:%{DATA:dsp2\_wasaction}%{SPACE}\| SOAPAction:%{DATA:dsp2\_soapaction}\| | %{SPACE})MessageID: urn:uuid:%{UUID:dsp2\_msgid}%{SPACE}\| Direction: %{WORD:dsp2\_typeEchange}%{GREEDYDATA:dsp2\_rest}"}

}

#création de la date de traitement par logstash

mutate { add\_field => { "tk\_create"=> "%{@timestamp}" } }

# génération identifiant unique de la log pour l'unicité de la log si rejeu

fingerprint {

source => "message"

concatenate\_all\_fields => "true"

target => "[@metadata][fingerprint]"

method => "SHA256"

}

# Récupération du mois sur deux chiffres

mutate {

add\_field => {"[JJ]" => "%{+dd}"}

add\_field => {"[MM]" => "%{+MM}"}

add\_field => {"[AAAA]" => "%{+yyyy}"}

}

}

L'attribut adabo est issu de la configuration filebeat, l'agent de collecte, par précaution, nous le forçons en lowercase :

filter {

grok {

match => {"message" => "%{TIMESTAMP\_ISO8601:timestamp}]\s+%{LOGLEVEL:level} \-%{SPACE}%{WORD:dsp2\_direction}:%{SPACE}%{NOTSPACE:dsp2\_url}\|( WSAction:%{DATA:dsp2\_wasaction}%{SPACE}\| SOAPAction:%{DATA:dsp2\_soapaction}\| | %{SPACE})MessageID: urn:uuid:%{UUID:dsp2\_msgid}%{SPACE}\| Direction: %{WORD:dsp2\_typeEchange}%{GREEDYDATA:dsp2\_rest}"}

}

#création de la date de traitement par logstash

mutate { add\_field => { "tk\_create"=> "%{@timestamp}" } }

# génération identifiant unique de la log pour l'unicité de la log si rejeu

fingerprint {

source => "message"

concatenate\_all\_fields => "true"

target => "[@metadata][fingerprint]"

method => "SHA256"

}

# Récupération du mois sur deux chiffres

mutate {

add\_field => {"[JJ]" => "%{+dd}"}

add\_field => {"[MM]" => "%{+MM}"}

add\_field => {"[AAAA]" => "%{+yyyy}"}

}

# Passage en minuscule du code ADABO

mutate {

lowercase => [ "adabo"]

}

}

Et enfin, la date de production de la log doit être au format utc, pour cela nous allons appliquer la règle permettant de la convertir. Cette conversion permet de s'affranchir des changements d'heure été/hiver et de l'heure locale de production de la log (cas des DomTom par exemple). C'est à la restitution que le format de date doit être converti à l'heure locale du consommateur.  
Rassurez-vous, Kibana le fait automatiquement.

filter {

grok {

match => {"message" => "%{TIMESTAMP\_ISO8601:timestamp}]\s+%{LOGLEVEL:level} \-%{SPACE}%{WORD:dsp2\_direction}:%{SPACE}%{NOTSPACE:dsp2\_url}\|( WSAction:%{DATA:dsp2\_wasaction}%{SPACE}\| SOAPAction:%{DATA:dsp2\_soapaction}\| | %{SPACE})MessageID: urn:uuid:%{UUID:dsp2\_msgid}%{SPACE}\| Direction: %{WORD:dsp2\_typeEchange}%{GREEDYDATA:dsp2\_rest}"}

}

#création de la date de traitement par logstash

mutate { add\_field => { "tk\_create"=> "%{@timestamp}" } }

# génération identifiant unique de la log pour l'unicité de la log si rejeu

fingerprint {

source => "message"

concatenate\_all\_fields => "true"

target => "[@metadata][fingerprint]"

method => "SHA256"

}

# Récupération du mois sur deux chiffres

mutate {

add\_field => {"[JJ]" => "%{+dd}"}

add\_field => {"[MM]" => "%{+MM}"}

add\_field => {"[AAAA]" => "%{+yyyy}"}

}

# Passage en minuscule du code ADABO

mutate {

lowercase => [ "adabo"]

}

date {

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

timezone => "Europe/Paris"

target => "@timestamp"

}

}

**En recommandation :**

* utilisez Grok
* faîtes des règles de parsing simples, ne multipliez pas les conditions (if...else) qui peuvent être coûteuses en performance
* ne mélanger pas les formats de logs au sein d'un même parsing, gardez à l'esprit qu'un pipeline logstash c'est pour un format de log d'une application et d'un produit
* n'oubliez-pas d'ajouter les règles obligatoires

**La section output**

* output{ }

La section output permet l'indexation des messages de logs captés par la section input { } et parsés et enrichis par la section filter { }  
Dans cette section, nous souhaitons persister les messages de logs dans un index elasticsearch à partir d'un index template.  
Le template ci-dessous doit être utilisé, en spécialisant les variables précisées entre double accolade.  
*NB : dans cette version, on remarque que le user et surtout le password sont en clair. Une évolution côté CAGIP permettra de préciser un fichier coffre-fort de type jceks.*

output {

elasticsearch {

hosts => "{{URL\_ELASTICSEARCH}}"

user => "{{USER\_ELASTICSEARCH}}"

password => "{{PWD\_ELASTICSEARCH}}"

ssl => true

ssl\_certificate\_verification => false

cacert => "/usr/share/logstash/config/ssl/ca.crt"

index => "ix\_%{adabo}\_{{produit}}\_{{codeappl}}\_{{typelog}}-%{AAAA}-%{MM}-%{JJ}"

template\_name => "te\_%{adabo}\_{{produit}}\_{{codeappl}}\_{{typelog}}"

document\_id => "%{[@metadata][fingerprint]}"

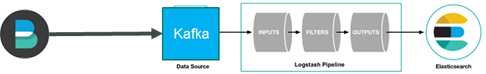
}

}

**E - Création Du Template Elasticsearch**

### Le template elasticsearch vous permet de caractériser les propriétés des index qui sont créés à partir de ce template.

Comme nous l'avons vu, c'est logstash qui a le rôle d'indexeur dans elasticsearch

[](http://elzudeggscas.yres.ytech/file/data/ec6p7l6lc2avqo3x2mv4/PHID-FILE-hlqqq4vs3azlot7u62vu/image.png)

Nous avons renseigné un nom d'index et un nom de template associé dans la configuration du pipeline logstash :

output {

elasticsearch {

hosts => "{{URL\_ELASTICSEARCH}}"

user => "{{USER\_ELASTICSEARCH}}"

password => "{{PWD\_ELASTICSEARCH}}"

ssl => true

ssl\_certificate\_verification => false

cacert => "/usr/share/logstash/config/ssl/ca.crt"

index => "ix\_%{adabo}\_{{produit}}\_{{typelog}}\_{{codeappl}}-%{AAAA}-%{MM}-%{JJ}"

template\_name => "te\_%{adabo}\_{{produit}}\_{{typelog}}\_{{codeappl}}"

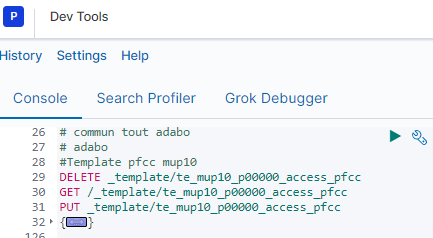
}

}

**NOTE:** Il faut savoir que si on ne précise pas de template index, elasticsearch va appliquer des règles d'indexation par défaut et interpréter le typage des différents attributs que nous avons parsé au préalable dans la section filter. En général, l'ensemble des champs seront de type "text".

**IMPORTANT:** Il est donc important de préciser les règles d'indexation à partir d'un template index : policy sur la durée de rétention, règle de sharding, mapping et enfin alias associés aux pattern d'index

##### Pour déclarer un index template, nous utiliserons l'API elasticsearch et pour jouer les requêtes la section devtools/console de l'UI ECE.

[](http://elzudeggscas.yres.ytech/file/data/kl5snezdoy73m35fzj2w/PHID-FILE-cua3twksl3cyho6mhu7a/image.png)

**IMPORTANT:** norme de nommage d'un template te\_{{adabo}}\_{{codeproduit}}\_{{codeappl}}\_{{typelog}}

### Détails des différentes sections à renseigner

Il y a 4 sections à renseigner dans la request rest de création de l'index template, vous permettant de spécifier votre index template

* index-patterns
* Sharding, Replica et policy de rétention
* mappings
* aliases

## Création de la Policy de rétention des index elasticsearch de vos logs

**NOTE:** Remplacer {{nbreJours}} par le nombre de jours de rétention de vos logs dans elasticsearch

# Création indice policy

PUT /\_ilm/policy/lp\_{{adabo}}\_{{codeproduit}}\_{{codeappl}}\_{{typelog}}

{

"policy" : {

"phases" : {

"delete" : {

"min\_age" : "{{nbreJours}}d",

"actions" : {

"delete" : { }

}

}

}

}

}

## Index elasticsearch

C'est le pattern regex permettant de regrouper tous les index qui seront concernés par ces spécifications.  
Souvenez-vous, il y a une norme sur les index, ce qui permet d'appliquer correctement ce pattern regex aux index que nous souhaitons cibler.  
Et comme nous avons défini une granularité adabo, codeproduit, codeappl, typelog,le nom du template suit lui aussi cette norme et donc cette granularité.

PUT \_template/te\_{{adabo}}\_{{codeproduit}}\_{{codeappl}}\_{{typelog}}

{

"index\_patterns": [

"ix\_{{adabo}}\_{{codeproduit}}\_{{codeappl}}\_{{typelog}}-\*"

],

"settings": {...},

"mappings": {...},

"aliases": {...}

}

## Sharding, Replica et policy de rétention

Le sharding et le replica caractérise la manière de stocker dans un index.  
La Policy de rétention définit les règles de purge des index, la requête plus haut a permis de la créer, il faut maintenant l'associer à l'index template  
Le replica doit systématiquement être à 1, c'est à dire une réplication à chaque indexation, ce qui assure la sauvegarde de chaque donnée, sachant que l'implémentation du stockage elasticsearch est en bi-site pour le PSI.  
Le sharding sera systématiquement positionné à 1.

PUT \_template/te\_{{adabo}}\_{{codeproduit}}\_{{codeappl}}\_{{typelog}}

{

"index\_patterns": ["ix\_{{adabo}}\_{{codeproduit}}\_{{codeappl}}\_{{typelog}}-\*"],

"settings": {

"lifecycle" : {

"name" : "lp\_{{adabo}}\_{{codeproduit}}\_{{codeappl}}\_{{typelog}}"

},

"number\_of\_shards": "1",

"number\_of\_replicas": "1"

},

"mappings": {...},

"aliases": {...}

}

## Mapping des attributs

C'est dans cette section que vous spécifiez

* le typage de certains champs si nécessaire,
* les champs sur lesquels je souhaite spécifiquement faire de l'agrégation,
* les champs sur lesquels je souhaite optimiser les recherches (docvalue, keyword)

**IMPORTANT:** Ajoutez dans cette section le codeadabo en tant que doc\_values, souvenez-vous que le codeadabo est valorisé par le logshipper.

Voici un exemple à adapter en fonction du mapping souhaité pour le format des logs que vous traitez :

PUT \_template/te\_{{adabo}}\_{{codeproduit}}\_{{codeappl}}\_{{typelog}}

{

"index\_patterns": [....],

"settings": {...},

"mappings": {

"properties" : {

"@timestamp" : {

"type" : "date"

},

"taille" : {

"index" : "false",

"type" : "integer"

},

"asicRessource" : {

"type" : "text",

"fields" : {

"keyword" : {

"ignore\_above" : 256,

"type" : "keyword"

}

}

},

"duree" : {

"index" : "false",

"type" : "integer"

},

"ipAppelante" : {

"type" : "text"

},

"adabo" : {

"type" : "keyword",

"doc\_values" : true

}

}

},

"aliases": {...}

}

## Alias

La section alias permet de définir des alias associés à ce template et donc aux index qui répondent aux règles de l'index pattern défini.  
L'alias devra être utilisé dans kibana lorsque vous allez déclarer un index pattern dans kibana.  
Cela permet d'assurer un découplage entre l'index elastic et son exposition dans kibana.  
Vous devez utiliser des index pattern reposant sur des alias dans les visualizations Kibana, c'est la bonne pratique.

Il faut voir l'alias comme une sorte de vue sur une table, avec les possibilités suivantes :

* par défaut exposer les index répondant au(x) pattern(s) défini(s) dans la section indexes
* définir un alias commun à plusieurs templates organisés par adabo pour avoir une vue globale de toutes les branches de cet adabo
* filtrer le contenu des index définis dans le template.

L'exemple ci-dessous est le template standard à utiliser, il permet de créer 2 alias distincts

* 1 alias al\_{{adabo}}\_{{codeproduit}}\_{{typelog}}\_{{codeappl}} permettant l'exposition des logs de l'instance adabo, par exemple mup10
* 1 alias al\_{{ada}}\_{{codeproduit}}\_{{typelog}}\_{{codeappl}} permettant l'exposition des logs des instance ada, par exemple mup10, mup20, mup30 et mup40

PUT \_template/te\_{{adabo}}\_{{codeproduit}}\_{{codeappl}}\_{{typelog}}

{

"index\_patterns": [

"ix\_{{adabo}}\_{{codeproduit}}\_{{codeappl}}\_{{typelog}}-\*"

],

"settings": {...},

"mappings": {...},

"aliases": {

"al\_{{adabo}}\_{{codeproduit}}\_{{codeappl}}\_{{typelog}}" : {},

"al\_{{ada}}\_{{codeproduit}}\_{{codeappl}}\_{{typelog}}" : {}

}

}

exemple :

Les API sont instanciées sur 2 branches en production mup10 et mup30, une branche passive et une branche active à un instant t.  
Dans ce cas, nous devons créer 2 index template, chacun contenant 2 alias

* 1 template pour mup10 associant les index mup10 avec une exposition mup10 et mup en déclarant 2 alias

PUT \_template/te\_mup10\_{{codeproduit}}\_{{typelog}}\_{{codeappl}}

{

"index\_patterns": [

"ix\_mup10\_{{codeproduit}}\_{{typelog}}\_{{codeappl}}-\*"

],

"settings": {...},

"mappings": {...},

"aliases": {

"al\_mup10\_{{codeproduit}}\_{{typelog}}\_{{codeappl}}" : {},

"al\_mup\_{{codeproduit}}\_{{typelog}}\_{{codeappl}}" : {}

}

}

* 1 template pour mup30 associant les index mup30 avec une exposition mup30 et mup en déclarant 2 alias

PUT \_template/te\_mup30\_{{codeproduit}}\_{{typelog}}\_{{codeappl}}

{

"index\_patterns": [

"ix\_mup30\_{{codeproduit}}\_{{typelog}}\_{{codeappl}}-\*"

],

"settings": {...},

"mappings": {...},

"aliases": {

"al\_mup30\_{{codeproduit}}\_{{typelog}}\_{{codeappl}}" : {},

"al\_mup\_{{codeproduit}}\_{{typelog}}\_{{codeappl}}" : {}

}

}

Ainsi, nous pouvons exposer les logs des APIM par branche adabo mup10 et mup30 à travers les alias al\_mup10\_... et al\_mup30\_...  
Dans chacun des 2 templates nous avons déclaré l'alias al\_mup...., celui va fédérer le contenu des index elasticsearch ix\_mup10 et ix\_mup30, il permettra une exposition de logs de l'environnement de production.

***Pourquoi les alias sont-ils déclarés dans l'index template ?***Effectivement, nous aurions pu créer les alias indépendamment des index et donc hors des index template, mais après de nombreux tests, nous nous sommes aperçu qu'au fil des données indexées, le contenu des alias n'est pas mis à jour et qu'il est dans ce cas nécessaire de les rafraîchir manuellement., soit à partir de l'UI ECE, soit par l'API Rest.

##### Voici un exemple permettant de créer le template elasticsearch des access log de PFCC Comcli :

# Création indice policies pfcc avec une durée de rétention de 30 jours

PUT /\_ilm/policy/lp\_zuda0\_p0000\_pfcc\_access\_sample

{

"policy" : {

"phases" : {

"delete" : {

"min\_age" : "30d",

"actions" : {

"delete" : { }

}

}

}

}

}

**Requête de création**

PUT \_template/te\_zuda0\_p0000\_pfcc\_access\_sample

{

"index\_patterns": [

"ix\_zuda0\_p00000\_access\_pfcc-\*"

],

"settings": {

"lifecycle" : {

"name" : "lp\_zuda0\_p0000\_pfcc\_access\_sample"

},

"index": {

"number\_of\_shards": "1",

"number\_of\_replicas": "1"

}

},

"mappings": {

"properties": {

"@timestamp": {

"type": "date"

},

"geoip": {

"dynamic": true,

"properties": {

"ip": {

"type": "ip"

},

"latitude": {

"type": "half\_float"

},

"location": {

"type": "geo\_point"

},

"longitude": {

"type": "half\_float"

}

}

},

"httpVersion": {

"type": "text",

"fields": {

"keyword": {

"ignore\_above": 256,

"type": "keyword"

}

}

},

"taille": {

"index": "false",

"type": "integer"

},

"@version": {

"type": "keyword"

},

"httpCodeRetour": {

"type": "text",

"fields": {

"keyword": {

"ignore\_above": 256,

"type": "keyword"

}

}

},

"asicFonction": {

"type": "text",

"fields": {

"keyword": {

"ignore\_above": 256,

"type": "keyword"

}

}

},

"asicRessource": {

"type": "text",

"fields": {

"keyword": {

"ignore\_above": 256,

"type": "keyword"

}

}

},

"duree": {

"index": "false",

"type": "integer"

},

"ipAppelante": {

"type": "text"

},

"adabo": {

"type": "keyword",

"doc\_values": true

}

}

},

"aliases": {

"al\_zuda0\_p0000\_pfcc\_access\_sample": {},

"al\_dev\_p0000\_pfcc\_access\_sample": {}

}

}

**Requête de lecture d'un template elasticsearch**

GET /\_template/te\_zuda0\_p0000\_pfcc\_access\_sample

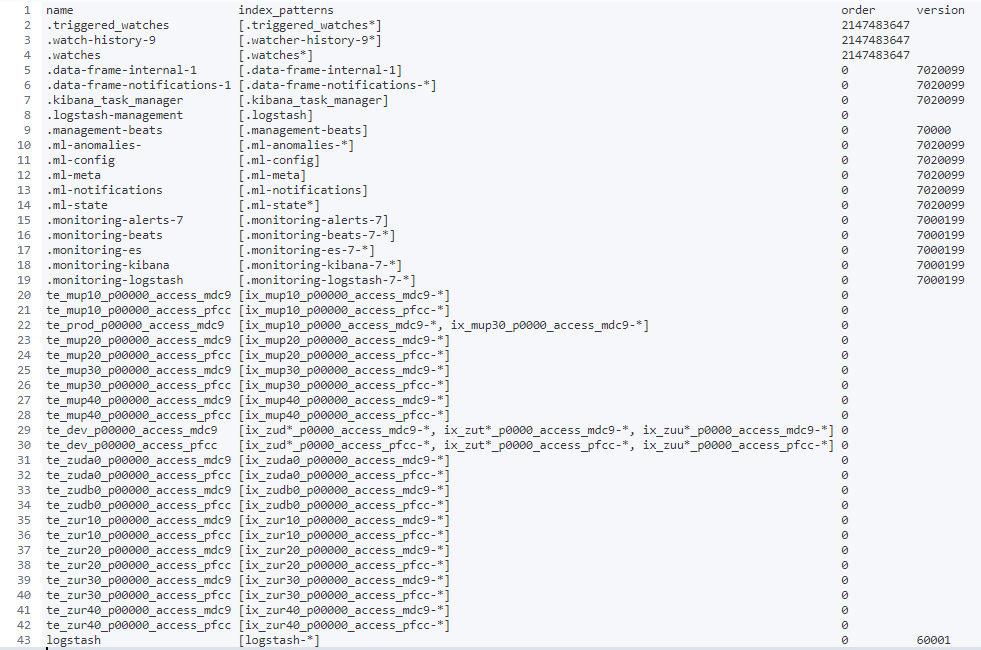
**Requête de suppression d'un template elasticsearch**

DELETE \_template/te\_zuda0\_p0000\_pfcc\_access\_sample

**Requête permettant de voir l'association template et index pattern elasticsearch**

GET /\_cat/templates?v&s=order:desc,index\_patterns

exemple de résultat obtenu :

[](http://elzudeggscas.yres.ytech/file/data/vhmvujtmret4i5grqwnn/PHID-FILE-rt7p3vwsqh37inu3btna/image.png)

* [Edit Document](http://elzudeggscas.yres.ytech/phriction/edit/82/)
* [View History](http://elzudeggscas.yres.ytech/phriction/history/kit_demarrage_ece/cr%C3%A9ation_du_template_elasticsearch/)
* [Publish Draft](http://elzudeggscas.yres.ytech/phriction/publish/82/650/)
* [Move Document](http://elzudeggscas.yres.ytech/phriction/move/82/)
* [Delete Document](http://elzudeggscas.yres.ytech/phriction/delete/82/)
* [Printable Page](http://elzudeggscas.yres.ytech/w/kit_demarrage_ece/cr%C3%A9ation_du_template_elasticsearch/?__print__=1)

Haut du formulaire

Subscribe

Bas du formulaire

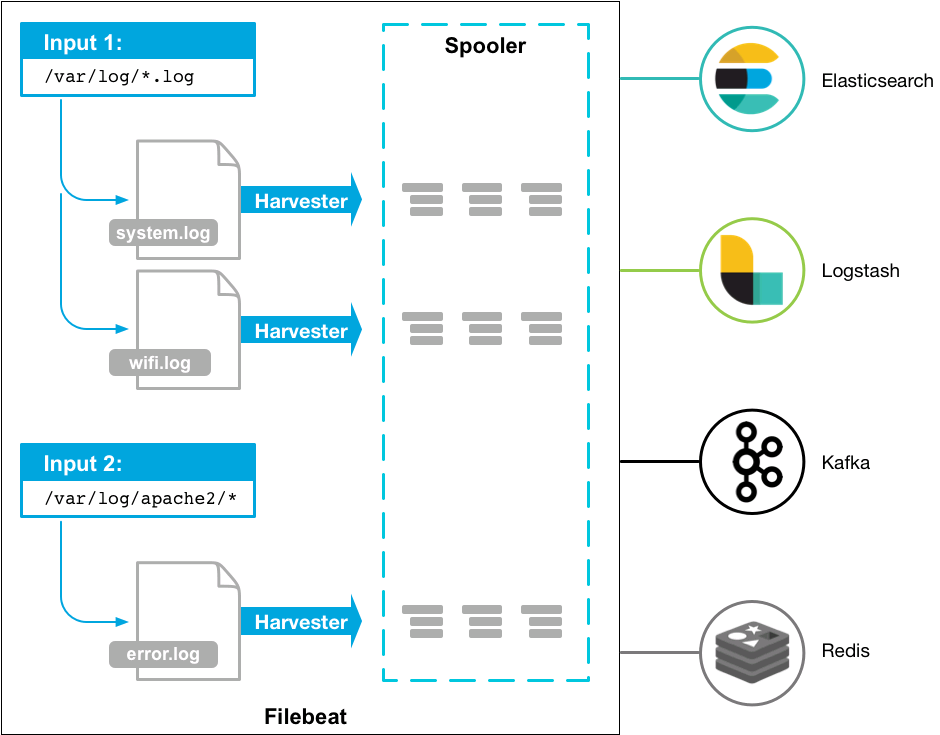
* [Mute Notifications](http://elzudeggscas.yres.ytech/subscriptions/mute/PHID-WIKI-sovmooon2g6woxphgigi/)
* [Award Token](http://elzudeggscas.yres.ytech/token/give/PHID-WIKI-sovmooon2g6woxphgigi/)
* [Flag For Later](http://elzudeggscas.yres.ytech/flag/edit/PHID-WIKI-sovmooon2g6woxphgigi/)
* [Advanced/Developer...](http://elzudeggscas.yres.ytech/w/kit_demarrage_ece/cr%C3%A9ation_du_template_elasticsearch/)

**Subscribers**

**Mise En Oeuvre de Filebeat**

**FileBeat : le logshipper de la plateforme ECE**

FileBeat est l'agent de collecte des logs, il écoute une source, en général un pattern représentant un nom d'un ou plusieurs fichiers dans un path, puis le distribue vers un point de collecte : Kafka, Elasticsearch, Logstash, Redis, ...ce sont les plus couramment utilisés.  
Il doit être installé au plus près de la source à collecter et repose sur une configuration de paramétrage permettant d'appliquer le routage de ces logs vers le point de collecte.

[](http://elzudeggscas.yres.ytech/file/data/o2mucedm3ulbmn5iu6it/PHID-FILE-fjmrwzygu7j5kgzwlw7a/image.png)

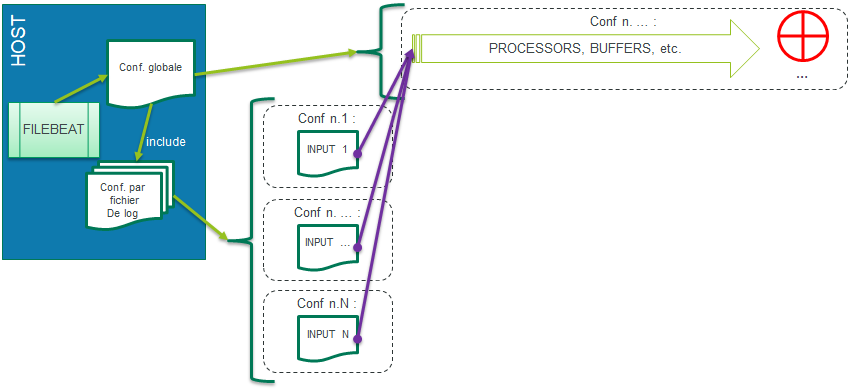
**Les modules de base**  
Des modules sont disponibles pour collecter les formats de logs les plus courants sans fichier de configuration particulière, mais en activant ces différents modules : log apache http, log kafka, log squid,...  
Après installation, la commande suivante vous permet de lister les différents modules enable/disable

\*\*filebeat modules list\*\*

Résultat :

[](http://elzudeggscas.yres.ytech/file/data/7l2ryomw6oar4u2zcaw7/PHID-FILE-guclthye27ub2jye7hr4/image.png)

**Filebeat fonctionne sur la base d'une configuration de fichier en yaml**

[](http://elzudeggscas.yres.ytech/file/data/yn5kf6gp2jfrqne6f2dj/PHID-FILE-bgsnfbpazhzuxhf37vl5/image.png)

##### L'objet de ce Tuto n'est pas d'utiliser les modules de base de FileBeat, mais plutôt de se focaliser sur une configuration nous permettant de respecter le pattern CAGIP.

C'est à dire collecter des fichiers à la source et les produire dans un topic kafka dédié à l'application, partie d'un produit pour un format de log et pour un adabo.

Supposons que FileBeat soit déjà installé sur l'infra source pour laquelle nous collectons les logs.  
Il est maintenant nécessaire de définir une configuration spécifique à cette collecte, dans laquelle nous préciserons :

* la source à collecter : un pattern de fichier de logs
* la cible de collecte : le topic Kafka de l'infra ECE/GACIP

La configuration est au format YAML

NB : Vous avez déjà défini la configuration logstash qui se raccroche à ce même topic et permet d'indexer dans elasticsearch.

**filebeat.inputs**

Voici une configuration sample permettant de collecter les logs de comcli.  
Dans l'exemple ci-dessous :  
-FileBeat doit écouter les fichiers correspondant au pattern \*\_localhost\_access\_log\*.\*.txt, produit par l'application dans le path /var/log/mup10/comcli/.

* un tag "fb\_p00000\_pfcc\_access" est ajouté, permettant d'identifier la configuration du logshipper
* l'attribut adabo est ajouté et nous indique l'environnement source de la collecte. Et c'est bien cet attribut que nous avons ajouté dans la configuration mapping de l'index template exlasticsearch
* l'identification du topic dans lequel chaque log sera produite., nous verrons dans la configuration output qu'il sera exploité en tant qu'attribut. Noté également qu'il porte qu'il est mutualisé ada pour l'ensemble des adabo. Ce qui permet de factorisé l'effort de parsing dans le pipeline logstash.

#=========================== Filebeat inputs =============================

filebeat.inputs:

# Each - is an input. Most options can be set at the input level, so

# you can use different inputs for various configurations.

# Below are the input specific configurations.

- type: log

# Change to true to enable this input configuration.

enabled: true

# Path des access log pour PFCC

paths:

-/var/log/mup10/comcli/\*\_localhost\_access\_log.\*.\*.txt

tags: ["fb\_p00000\_pfcc\_access"]

fields:

adabo: "mup10"

kafka\_topic: "cats.elisa.pfcc.mup.p00000.pfcc.access"

fields\_under\_root: true

**pattern générique filebeat.inputs**  
Voici le pattern générique de la configuration filebeat.input, il vous suffit de spécialiser les attributs entre double acolade :

#=========================== Filebeat inputs =============================

filebeat.inputs:

# Each - is an input. Most options can be set at the input level, so

# you can use different inputs for various configurations.

# Below are the input specific configurations.

- type: log

# Change to true to enable this input configuration.

enabled: true

# Path des access log pour {{codeproduit}} {{codeappl}} {{adabo}} {{typelog}}

paths:

- {{pathlog}}/{{patternlog}}

tags: ["fb\_{{codeproduit}}{{codeappl}}{{typelog}}"]

fields:

adabo: "{{adabo}}"

kafka\_topic: "cats.elisa.{{ada}}.{{codeproduit}}.{{codeappl}}.{{typelog}}"

fields\_under\_root: true

**La configuration Output de Filebeat**

La cible FileBeat c'est obligatoirement Kafka dans l'Offre ECE/CAGIP.  
Tout comme dans la configuration logstash, nous devons préciser le BootStrap Kafka, c'est à dire l'ensemble des noeuds composant le cluster.  
Les flux sont sécurisés, cela nécessite un paramétrage ssl et l'utilisation d'un certificat d'autorité Carioca. Ce certificat sera installé avec FileBeat, il faut simplement préciser son path dans la configuration fileBeat.  
Le user et password permettant de produire les messages dans le Topic Kafka. A noter que ce n'est pas le paramétrage cible, à terme ils seront sécurisés dans un fichier jceks de type coffre-fort et donc non visibles dans la configuration.

Exemple : écriture des logs pour COMCLI pfcc

#----------------------------- KAFKA output --------------------------------

output.kafka:

# The Kafka hosts

hosts: ["broker-101.streaming.iaas.cagip.group.gca:9093", "broker-102.streaming.iaas.cagip.group.gca:9093", "broker-103.streaming.iaas.cagip.group.gca:9093", "broker-104.streaming.iaas.cagip.group.gca:9093", "broker-105.streaming.iaas.cagip.group.gca:9093","broker-106.streaming.iaas.cagip.group.gca:9093"]

ssl.enabled: true

ssl.verification\_mode: full

ssl.supported\_protocols: [TLSv1.0, TLSv1.1, TLSv1.2]

ssl\_certificate\_authorities: D:\certificatsELK\kafkaELKCagip.pem

version: '2.0.0'

username: 'cats.elisa.pfcc.producer'

password: 'quauGhei9iep'

# message topic selection + partitioning

topic: '%{[kafka\_topic]}'

# partition.round\_robin:

# reachable\_only: true

required\_acks: 1

client\_id: fb\_p00000\_pfcc\_access

worker: 1

compression: none

max\_message\_bytes: 1000000

**Pattern générique filebeat output.kafka**

Ci-dessous le pattern de configuration output que vous devez spécialiser :

* SERVEUR\_BOOTSTRAP\_KAFKA : dns:port de tous les serveurs Kafka composants le cluster CAGIP séparés par une virgule.
* PATH\_FILE\_CERTIFICATE\_CARIOCA : path du fichier certificat carioca
* USER\_PRODUCER\_KAFKA : username du producer kafka
* PWD\_PRODUCER\_KAFKA : password du producer kafka

NB : Les user / pwd kafka doivent à terme être sécurisé dans un fichier jceks de type coffre

* {{codeproduit}}\_{{codeappl}}\_{{typelog}}

#----------------------------- KAFKA output --------------------------------

output.kafka:

# The Kafka hosts

hosts: ["{{SERVEUR\_BOOTSTRAP\_KAFKA}}"]

ssl.enabled: true

ssl.verification\_mode: full

ssl.supported\_protocols: [TLSv1.0, TLSv1.1, TLSv1.2]

ssl\_certificate\_authorities: {{PATH\_FILE\_CERTIFICATE\_CARIOCA}}

version: '2.0.0'

username: '{{USER\_PRODUCER\_KAFKA}}'

password: '{{PWD\_PRODUCER\_KAFKA}}'

# message topic selection + partitioning

topic: '%{[kafka\_topic]}'

# partition.round\_robin:

# reachable\_only: true

required\_acks: 1

client\_id: fb\_{{codeproduit}}\_{{codeappl}}\_{{typelog}}

worker: 1

compression: none

max\_message\_bytes: 1000000

* [Edit Document](http://elzudeggscas.yres.ytech/phriction/edit/83/)
* [View History](http://elzudeggscas.yres.ytech/phriction/history/kit_demarrage_ece/mise_en_oeuvre_de_filebeat/)
* [Publish Draft](http://elzudeggscas.yres.ytech/phriction/publish/83/673/)
* [Move Document](http://elzudeggscas.yres.ytech/phriction/move/83/)
* [Delete Document](http://elzudeggscas.yres.ytech/phriction/delete/83/)
* [Printable Page](http://elzudeggscas.yres.ytech/w/kit_demarrage_ece/mise_en_oeuvre_de_filebeat/?__print__=1)

Haut du formulaire

Subscribe

Bas du formulaire

* [Mute Notifications](http://elzudeggscas.yres.ytech/subscriptions/mute/PHID-WIKI-eo2tlxwrgcbez3eedisr/)
* [Award Token](http://elzudeggscas.yres.ytech/token/give/PHID-WIKI-eo2tlxwrgcbez3eedisr/)
* [Flag For Later](http://elzudeggscas.yres.ytech/flag/edit/PHID-WIKI-eo2tlxwrgcbez3eedisr/)
* [Advanced/Developer...](http://elzudeggscas.yres.ytech/w/kit_demarrage_ece/mise_en_oeuvre_de_filebeat/)

**Subscribers**

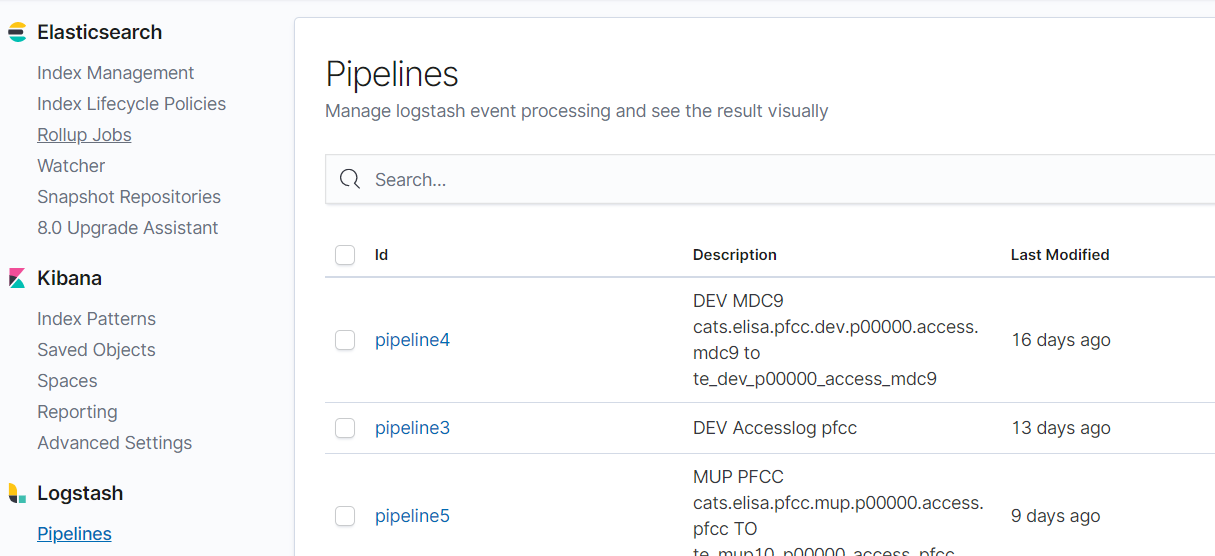
**Démarrage de l'ingestion Logstash**

**A cette étape, vous avez :**

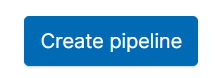
* un index template elasticsearch vous permettant d'appliquer les règles d'indexation et de mapping, de déclarer un ou plusieurs alias associés à ces index.
* un fichier logstash définissant les sections input{ } filter { } et output { } permettant de consommer les logs disponibles sur le topic kafka, d'appliquer les règles de parsing et enfin de persister ces logs dans un index elasticsearch défini à partir de l'index template
* un fichier de configuration filebeat, permettant de collecter des logs écrites par l'application source dans des fichiers et de les produire dans un topic kafka.
* des logs à disposition

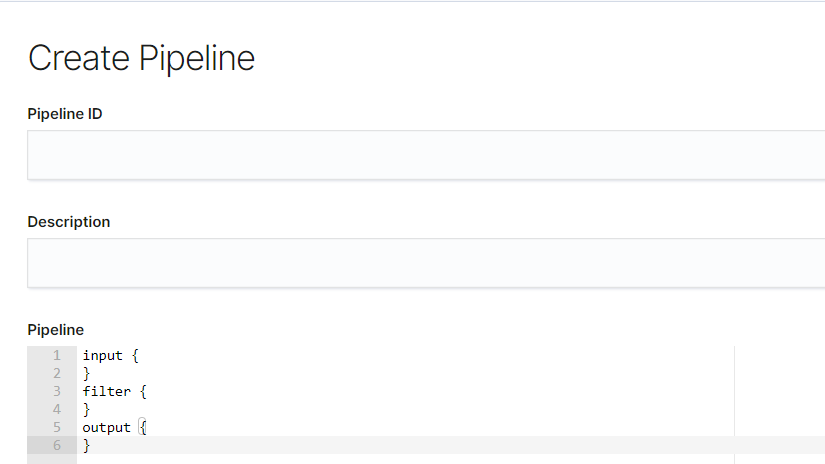
**Démarrage du pipeline logstash**

* Aller dans la console ECE dans Management et cliquer sur Logstash\Pipelines

[](http://elzudeggscas.yres.ytech/file/data/s4i2ufwvciilx3bfnfql/PHID-FILE-xymorc7ybog4sjl3bm2r/image.png)

* Cliquer sur "Create Pipeline"

[](http://elzudeggscas.yres.ytech/file/data/zvnmdqtyqnxdq3gk5xos/PHID-FILE-cotjhbw4ripps7j7dlfg/image.png)

[](http://elzudeggscas.yres.ytech/file/data/l3guuxanulvwxuh66j3o/PHID-FILE-qsckaaomngy4xhdftogu/image.png)

* Utiliser un id pipeline réservé par l'équipe CAGIP, typiquement pipeline1, pipeline2,...
* Donner une description claire de ce qu'effectue le pipeline sous la forme

{{ADA}} {{CODEAPPL}} {{nomTopicSource}} to {{nomTemplateIndex}}  
Par exemple :  
DEV MDC9 cats.elisa.pfcc.dev.p00000.access.mdc9 to te\_dev\_p00000\_access\_mdc9

Pour identifier le pipeline de DEV, donc qui traite les logs de ZUDA0/ZUDB0, du code application MDC9 de la source input cats.elisa.pfcc.dev.p00000.access.mdc9, donc le topic, vers la cible ouput te\_dev\_p00000\_access\_mdc9, c'est à dire tous les index elasticsearch indexés par le template index correspondant à ces logs.

* Faire un Copier/Coller de votre configuration logstash que nous avons défini précédemment

**Performance et Gestion du back-pressure**  
Avant de démarrer votre pipeline, vous devez renseigner le paramétrage permettant d'optimiser la performance et de gérer le back-pressure

* Performance :

Renseigner le nbre de Workers en fonction de la volumétrie attendue : nbre de worker, pipeline batch size, pipeline batch delay  
Vous pouvez vous référer au site suivant si vous rencontrez des pb de performance :  
[Tuning dans Profiling Logstash](https://www.elastic.co/guide/en/logstash/current/tuning-logstash.html)

Le paramètre pipeline batch size, permet de traiter l'indexation des messages par paquets et le paramétre batch delay, le temps max d'attente avant indexation des messages en attente lorsqu'il y en a moins que le nbre paramétré dans batch size.

Comme starting point, nous préconisons 1 worker, pipeline batch size = 1000, pipeline batch delay = 50

* Persistance et Back-pressure :

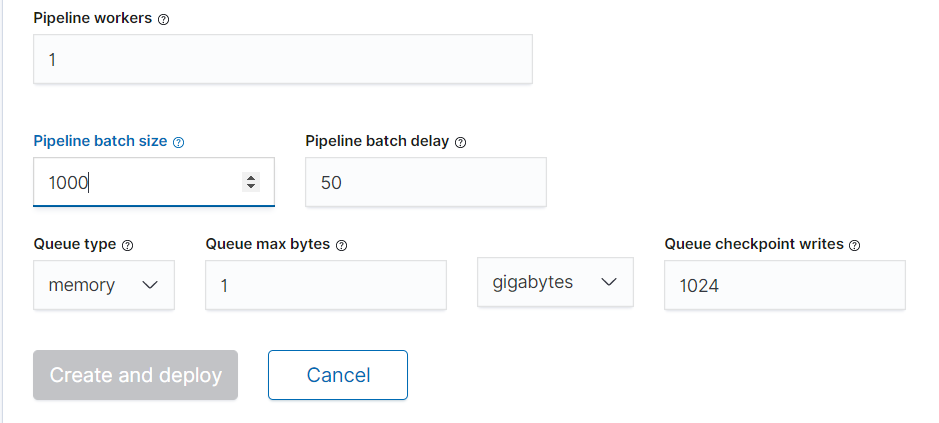
Le paramétrage de la queue est essentielle pour un bon fonctionnement nominal de logstash et aussi lorsque ça se passe moins bien.  
Le back-pressure permet d'assurer une persistance temporaire des records avant traitement.  
Cela permet d'éviter la perte de données, jusqu'à une certaine limite.  
Au delà de la capacité de rétention allouée, logstash n'acceptera plus de nouveaux messages de log.  
Par exemple sur une indisponibilité du cluster elasticsearch ou bien un manque de performance dans la section filter { } de votre pipeline logstash ou encore un manque de capacité logstash compte-tenu de la volumétrie à traiter.

Vous pouvez vous référer au site suivant pour de plus amples détails sur ce mécanisme :  
[Persistent Queues](https://www.elastic.co/guide/en/logstash/current/persistent-queues.html)

Dans les faits, souvenez-vous que kafka est en frontal de Filebeat et donc en source de logstash, le topic fait donc lui-même office de buffer et donc permet la gestion du back-pressure.

**Comme starting point,pour le paramétrage de queue des messages input logstash, mettre les paramètres par défaut :**

* queue type : memory
* queue max : 1 GigaBytes
* queue checkpoint writes : 1024

[](http://elzudeggscas.yres.ytech/file/data/sxojmatc43bl4iuf3ccj/PHID-FILE-x6nfljut3ejqm2mmicqy/image.png)

Votre pipeline est maintenant paramétrer, démarrer-le en cliquant sur

**H - Démarrage de Filebeat**

**Lors de la phase de développement, vous n'avez pas encore déployé FileBeat sur vos infra.**

TODO : à préciser  
Vous pouvez procéder de la manière suivante :

* Installer FileBeat sur une VM
* Installer FileBeat sur votre poste de travail

2 - Copier le certificat pem dans le path suivant  
3 - Copier votre fichier de configuration dans le path suivant  
4- Démarrer filebeat avec la commande suivante

filebeat -e -c filebeat.yml

Filebeat démarre, voici ci-dessous un exemple de trace de FileBeat.

2019-09-26T09:09:55.660+0200 INFO instance/beat.go:606 Home path: [C:\LOGICIELS\filebeat-7.2.1-windows-x86\_64] Config path: [C:\LOGICIELS\filebeat-7.2.1-windows-x86\_64] Data path: [C:\LOGICIELS\

2019-09-26T09:09:55.709+0200 INFO instance/beat.go:614 Beat ID: ffdaca1c-c1c2-46a5-9ea3-f13157c68e2b

2019-09-26T09:09:55.833+0200 INFO [beat] instance/beat.go:902 Beat info {"system\_info": {"beat": {"path": {"config": "C:\\LOGICIELS\\filebeat-7.2.1-windows-x86\_64", "data": "C:\\LOGICIELS

S\\filebeat-7.2.1-windows-x86\_64\\logs"}, "type": "filebeat", "uuid": "ffdaca1c-c1c2-46a5-9ea3-f13157c68e2b"}}}

2019-09-26T09:09:55.835+0200 INFO [beat] instance/beat.go:911 Build info {"system\_info": {"build": {"commit": "dd3f47f0fb299aa5de9c5c1468faacc1b9b3c27f", "libbeat": "7.2.1", "time": "2019-

2019-09-26T09:09:55.841+0200 INFO [beat] instance/beat.go:914 Go runtime info {"system\_info": {"go": {"os":"windows","arch":"amd64","max\_procs":4,"version":"go1.12.4"}}}

2019-09-26T09:09:55.980+0200 INFO [beat] instance/beat.go:918 Host info {"system\_info": {"host": {"architecture":"x86\_64","boot\_time":"2019-09-25T08:38:06.47+02:00","name":"NSTEPNTE0309",

91a:d0b7:8fed/64","192.168.56.1/24","fe80::2834:7039:5f55:723f/64","192.168.99.1/24","fe80::c1eb:3da:b4c1:acbd/64","192.168.220.1/24","fe80::ccef:8f71:7bb9:b8f9/64","192.168.102.1/24","::1/128","127.0.0.

0::ffff:ffff:fffe/64","fe80::5efe:a9c:1528/128"],"kernel\_version":"6.1.7601.24475 (win7sp1\_ldr.190516-0600)","mac":["e4:a7:a0:9e:b7:a4","98:e7:f4:f2:db:76","0a:00:27:00:00:10","0a:00:27:00:00:11","00:50:

:00:00:00:00:00:e0","00:00:00:00:00:00:00:e0"],"os":{"family":"windows","platform":"windows","name":"Windows 7 Enterprise","version":"6.1","major":1,"minor":0,"patch":0,"build":"7601.0"},"timezone":"CEST

2019-09-26T09:09:55.985+0200 INFO [beat] instance/beat.go:947 Process info {"system\_info": {"process": {"cwd": "C:\\LOGICIELS\\filebeat-7.2.1-windows-x86\_64", "exe": "C:\\LOGICIELS\\filebeat

09:09:55.230+0200"}}}

2019-09-26T09:09:55.989+0200 INFO instance/beat.go:292 Setup Beat: filebeat; Version: 7.2.1

2019-09-26T09:09:55.998+0200 INFO [publisher] pipeline/module.go:97 Beat name: NSTEPNTE0309

2019-09-26T09:09:56.004+0200 WARN beater/filebeat.go:152 Filebeat is unable to load the Ingest Node pipelines for the configured modules because the Elasticsearch output is not configured/enabled.

2019-09-26T09:09:56.005+0200 INFO instance/beat.go:421 filebeat start running.

2019-09-26T09:09:56.005+0200 INFO [monitoring] log/log.go:118 Starting metrics logging every 30s

2019-09-26T09:09:56.018+0200 INFO registrar/registrar.go:145 Loading registrar data from C:\LOGICIELS\filebeat-7.2.1-windows-x86\_64\data\registry\filebeat\data.json

2019-09-26T09:09:56.020+0200 INFO registrar/registrar.go:152 States Loaded from registrar: 45

2019-09-26T09:09:56.023+0200 WARN beater/filebeat.go:368 Filebeat is unable to load the Ingest Node pipelines for the configured modules because the Elasticsearch output is not configured/enabled.

2019-09-26T09:09:56.027+0200 INFO crawler/crawler.go:72 Loading Inputs: 1

2019-09-26T09:09:56.053+0200 INFO log/input.go:148 Configured paths: [D:\Partage\_VMWARE\comcli\logs\mup10\\*\_localhost\_access\_log.\*.\*.txt]

2019-09-26T09:09:56.054+0200 INFO input/input.go:114 Starting input of type: log; ID: 12647202936984183044

2019-09-26T09:09:56.058+0200 INFO crawler/crawler.go:106 Loading and starting Inputs completed. Enabled inputs: 1

2019-09-26T09:09:56.059+0200 INFO cfgfile/reload.go:172 Config reloader started

2019-09-26T09:09:58.837+0200 INFO add\_cloud\_metadata/add\_cloud\_metadata.go:347 add\_cloud\_metadata: hosting provider type not detected.

Maintenant, vous pouvez copier un ou plusieurs fichiers de log répondant au pattern que vous avez défini dans la configuration filebeat.input\path dans votre fichier yml.

A présent, vous devez voir filebeat produire une log de ce type  
Nous voyons particulièrement qu'un harvester filebeat démarre lorsque vous avez copier le fichier de logs, celui-ci initie la connexion au broker kafka et publie les messages en roundrobin sur les 6 partitions du topic kafka dédié à votre type de log, ici cats.elisa.pfcc.mup.p00000.access.pfcc.

2019-09-26T09:19:56.509+0200 INFO log/harvester.go:253 Harvester started for file: D:\Partage\_VMWARE\comcli\logs\mup10\ELMUPEGHDI9C\_localhost\_access\_log.2019-08-01.jm - Copie -

2019-09-26T09:19:56.993+0200 INFO pipeline/output.go:95 Connecting to kafka(broker-101.streaming.iaas.cagip.group.gca:9093,broker-102.streaming.iaas.cagip.group.gca:9093,broker-

roker-106.streaming.iaas.cagip.group.gca:9093)

2019-09-26T09:19:57.001+0200 INFO kafka/log.go:53 kafka message: Initializing new client

2019-09-26T09:19:57.009+0200 INFO kafka/log.go:53 client/metadata fetching metadata for all topics from broker broker-106.streaming.iaas.cagip.group.gca:9093

2019-09-26T09:20:18.180+0200 INFO kafka/log.go:53 Failed to connect to broker broker-106.streaming.iaas.cagip.group.gca:9093: dial tcp 1.1.1.5:9093: connectex: A connection attemp

d host has failed to respond.

2019-09-26T09:20:18.180+0200 INFO kafka/log.go:53 client/metadata got error from broker -1 while fetching metadata: dial tcp 1.1.1.5:9093: connectex: A connection attempt failed b

failed to respond.

2019-09-26T09:20:18.180+0200 INFO kafka/log.go:53 client/metadata fetching metadata for all topics from broker broker-102.streaming.iaas.cagip.group.gca:9093

2019-09-26T09:20:26.153+0200 INFO [monitoring] log/log.go:145 Non-zero metrics in the last 30s {"monitoring": {"metrics": {"beat":{"cpu":{"system":{"ticks":920,"time":{

{"ephemeral\_id":"4ea55c69-d062-42ac-a045-0cca06f211fe","uptime":{"ms":630382}},"memstats":{"gc\_next":31028400,"memory\_alloc":19348864,"memory\_total":47476696,"rss":19685376},"runtime":{

beat":{"config":{"module":{"running":0},"reloads":1},"pipeline":{"clients":8,"events":{"active":4117,"filtered":1,"published":4116,"total":4118}}},"registrar":{"states":{"current":46,"u

2019-09-26T09:20:28.250+0200 INFO kafka/log.go:53 Failed to connect to broker broker-102.streaming.iaas.cagip.group.gca:9093: dial tcp 1.1.1.5:9093: connectex: An established conn

2019-09-26T09:20:28.319+0200 INFO kafka/log.go:53 client/metadata got error from broker -1 while fetching metadata: dial tcp 1.1.1.5:9093: connectex: An established connection was

2019-09-26T09:20:28.454+0200 INFO kafka/log.go:53 client/metadata fetching metadata for all topics from broker broker-103.streaming.iaas.cagip.group.gca:9093

2019-09-26T09:20:28.519+0200 INFO kafka/log.go:53 Failed to connect to broker broker-103.streaming.iaas.cagip.group.gca:9093: dial tcp: lookup broker-103.streaming.iaas.cagip.grou

2019-09-26T09:20:28.651+0200 INFO kafka/log.go:53 client/metadata got error from broker -1 while fetching metadata: dial tcp: lookup broker-103.streaming.iaas.cagip.group.gca: no

2019-09-26T09:20:28.700+0200 INFO kafka/log.go:53 client/metadata fetching metadata for all topics from broker broker-105.streaming.iaas.cagip.group.gca:9093

2019-09-26T09:20:30.122+0200 INFO kafka/log.go:53 kafka message: Successful SASL handshake

2019-09-26T09:20:30.143+0200 INFO kafka/log.go:53 SASL authentication successful with broker broker-105.streaming.iaas.cagip.group.gca:9093:4 - [0 0 0 0]

2019-09-26T09:20:30.148+0200 INFO kafka/log.go:53 Connected to broker at broker-105.streaming.iaas.cagip.group.gca:9093 (unregistered)

2019-09-26T09:20:30.180+0200 INFO kafka/log.go:53 client/brokers registered new broker #5 at broker-105.streaming.iaas.cagip.group.gca:9093

2019-09-26T09:20:30.183+0200 INFO kafka/log.go:53 client/brokers registered new broker #1 at broker-101.streaming.iaas.cagip.group.gca:9093

2019-09-26T09:20:30.196+0200 INFO kafka/log.go:53 client/brokers registered new broker #4 at broker-104.streaming.iaas.cagip.group.gca:9093

2019-09-26T09:20:30.201+0200 INFO kafka/log.go:53 client/brokers registered new broker #6 at broker-106.streaming.iaas.cagip.group.gca:9093

2019-09-26T09:20:30.205+0200 INFO kafka/log.go:53 client/brokers registered new broker #2 at broker-102.streaming.iaas.cagip.group.gca:9093

2019-09-26T09:20:30.213+0200 INFO kafka/log.go:53 client/brokers registered new broker #3 at broker-103.streaming.iaas.cagip.group.gca:9093

2019-09-26T09:20:30.225+0200 INFO kafka/log.go:53 kafka message: Successfully initialized new client

2019-09-26T09:20:30.228+0200 INFO pipeline/output.go:105 Connection to kafka(broker-101.streaming.iaas.cagip.group.gca:9093,broker-102.streaming.iaas.cagip.group.gca:9093,broker-

roker-106.streaming.iaas.cagip.group.gca:9093) established

2019-09-26T09:20:30.253+0200 INFO kafka/log.go:53 producer/broker/5 starting up

2019-09-26T09:20:30.262+0200 INFO kafka/log.go:53 producer/broker/5 state change to [open] on cats.elisa.pfcc.mup.p00000.access.pfcc/0

2019-09-26T09:20:30.253+0200 INFO kafka/log.go:53 producer/broker/2 starting up

2019-09-26T09:20:30.294+0200 INFO kafka/log.go:53 producer/broker/2 state change to [open] on cats.elisa.pfcc.mup.p00000.access.pfcc/1

2019-09-26T09:20:30.278+0200 INFO kafka/log.go:53 producer/broker/1 starting up

2019-09-26T09:20:30.306+0200 INFO kafka/log.go:53 producer/broker/1 state change to [open] on cats.elisa.pfcc.mup.p00000.access.pfcc/2

2019-09-26T09:20:30.301+0200 INFO kafka/log.go:53 producer/broker/4 starting up

2019-09-26T09:20:30.278+0200 INFO kafka/log.go:53 producer/broker/3 starting up

2019-09-26T09:20:30.318+0200 INFO kafka/log.go:53 producer/broker/1 state change to [open] on cats.elisa.pfcc.mup.p00000.access.pfcc/3

2019-09-26T09:20:30.331+0200 INFO kafka/log.go:53 producer/broker/4 state change to [open] on cats.elisa.pfcc.mup.p00000.access.pfcc/4

2019-09-26T09:20:30.343+0200 INFO kafka/log.go:53 producer/broker/3 state change to [open] on cats.elisa.pfcc.mup.p00000.access.pfcc/5

2019-09-26T09:20:30.456+0200 INFO kafka/log.go:53 kafka message: Successful SASL handshake

2019-09-26T09:20:30.465+0200 INFO kafka/log.go:53 kafka message: Successful SASL handshake

2019-09-26T09:20:30.478+0200 INFO kafka/log.go:53 SASL authentication successful with broker broker-105.streaming.iaas.cagip.group.gca:9093:4 - [0 0 0 0]

2019-09-26T09:20:30.483+0200 INFO kafka/log.go:53 Connected to broker at broker-105.streaming.iaas.cagip.group.gca:9093 (registered as #5)

2019-09-26T09:20:30.493+0200 INFO kafka/log.go:53 SASL authentication successful with broker broker-102.streaming.iaas.cagip.group.gca:9093:4 - [0 0 0 0]

2019-09-26T09:20:30.498+0200 INFO kafka/log.go:53 Connected to broker at broker-102.streaming.iaas.cagip.group.gca:9093 (registered as #2)

2019-09-26T09:20:33.688+0200 INFO kafka/log.go:53 kafka message: Successful SASL handshake

2019-09-26T09:20:33.714+0200 INFO kafka/log.go:53 SASL authentication successful with broker broker-103.streaming.iaas.cagip.group.gca:9093:4 - [0 0 0 0]

2019-09-26T09:20:33.718+0200 INFO kafka/log.go:53 Connected to broker at broker-103.streaming.iaas.cagip.group.gca:9093 (registered as #3)

2019-09-26T09:20:34.296+0200 INFO kafka/log.go:53 kafka message: Successful SASL handshake

2019-09-26T09:20:34.378+0200 INFO kafka/log.go:53 SASL authentication successful with broker broker-104.streaming.iaas.cagip.group.gca:9093:4 - [0 0 0 0]

2019-09-26T09:20:34.381+0200 INFO kafka/log.go:53 Connected to broker at broker-104.streaming.iaas.cagip.group.gca:9093 (registered as #4)

2019-09-26T09:20:35.194+0200 INFO kafka/log.go:53 kafka message: Successful SASL handshake

2019-09-26T09:20:35.236+0200 INFO kafka/log.go:53 SASL authentication successful with broker broker-101.streaming.iaas.cagip.group.gca:9093:4 - [0 0 0 0]

2019-09-26T09:20:35.239+0200 INFO kafka/log.go:53 Connected to broker at broker-101.streaming.iaas.cagip.group.gca:9093 (registered as #1)

**Vérification de l'Indexation Elasticsearch**

A ce stade, vous avez

* vous avez démarré le pipeline logstash d'ingestion des logs produites dans votre topic Kafka vers l'index elasticsearch que l'on a défini
* vous avez démarré filebeat,

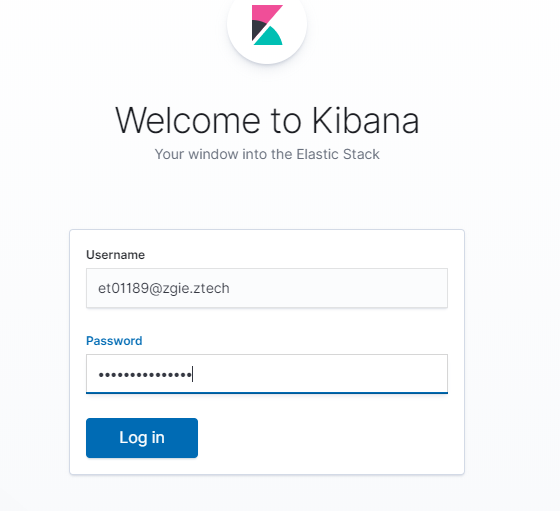
**J - Création de l'Index Pattern KibanaLes index Pattern Kibana**  
A ce stade les logs ont bien été indexés dans votre index elasticsearch par logstash à partir d'un index template dans lequel vous avez déclaré des alias.  
Pour pouvoir visualiser vos logs à partir du Discover Kibana, ou construire des dashboards à partir de visualizations, vous devez déclarer un index pattern dans Kibana.  
Il faut savoir que tous les objets kibana appartiennent à un et un seul Espace Kibana dit Space.

NB : A ce jour, les rôles n'ont pas encore été défini, cependant nous considérons qu'un rôle Administrateur vous aura créer l'espace Kibana correspondant à votre besoin à la granularité Produit, Squad ou Tribu, cela reste à définir.

* Lancer l'UI ECE de dev

<https://cats-logs-elk0-zuy.elisa.saas.cagip.group.gca/login>

* connectez-vous avec votre user ZGIE et votre password

[](http://elzudeggscas.yres.ytech/file/data/d4rudi3ygpixzmpswxn7/PHID-FILE-v2c37r4vt3dvnuveadmr/image.png)

* Sélectionner le space Kibana qui vous a été réservé

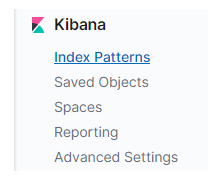
*exemple*

[](http://elzudeggscas.yres.ytech/file/data/rpv4rlcpbeh65zbwj6bo/PHID-FILE-c45j2jcda4gzpbfqc5ly/image.png)

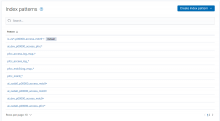
* Cliquer sur Management

[](http://elzudeggscas.yres.ytech/file/data/bcsyrhpc6bggmnmu2k7m/PHID-FILE-z4lil6r5ukxam7zn7o4m/image.png)

* Cliquer sur Index pattern

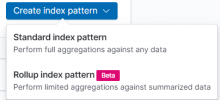
[](http://elzudeggscas.yres.ytech/file/data/bhhpfo5vauerzv4pt6r7/PHID-FILE-4j3xma6nyxu5zwddxl2v/image.png)

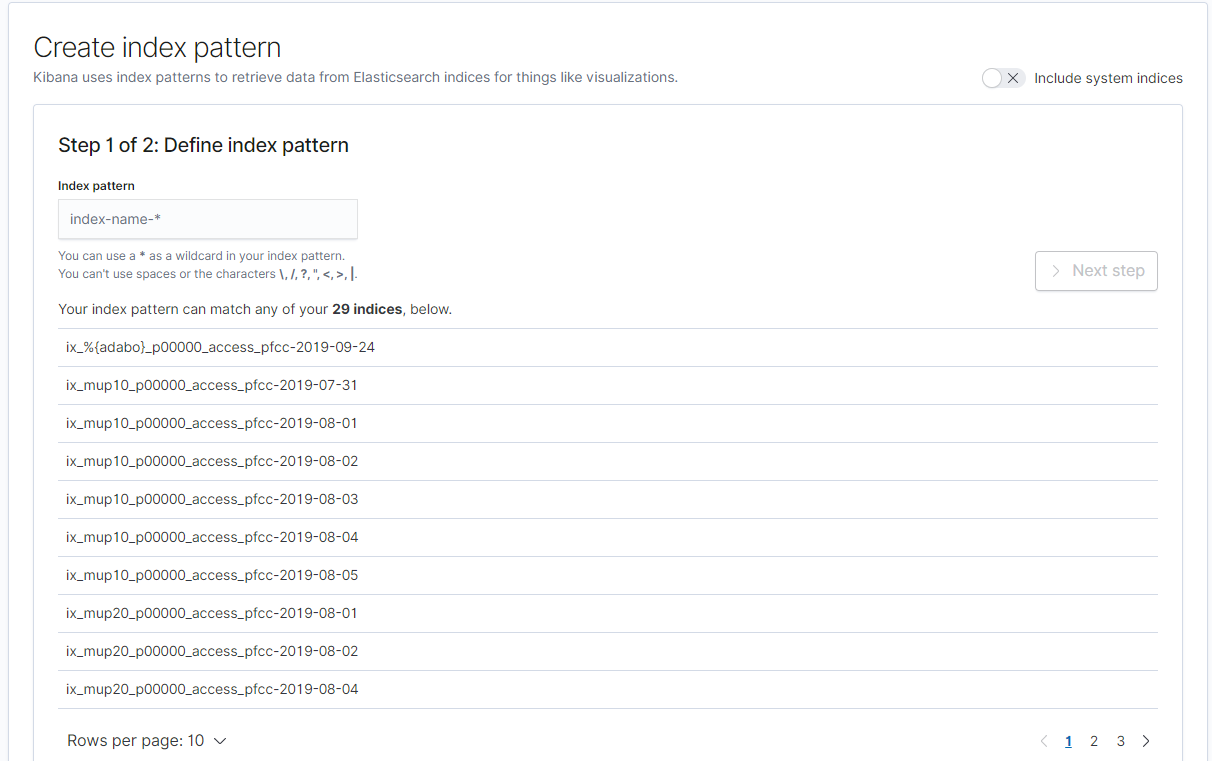
La liste des index elasticsearch s'affiche, logiquement elle devrait être vide si vous n'avez pas encore défini d'index.  
Sinon, vous avez la liste de vos index pattern déjà définis

[](http://elzudeggscas.yres.ytech/file/data/6z7egolcxfdrjaxctikf/PHID-FILE-uaha7lcdx44kwprpt32d/image.png)

**Définir l'index pattern permettant la visualisation de vos logs**

* Cliquer sur Create Index pour dérouler les choix possibles et sélectionner standard index pattern

[](http://elzudeggscas.yres.ytech/file/data/gskyml5zlfzacg3xvs7f/PHID-FILE-jejjfjyol6txp2tu6znt/image.png)

[](http://elzudeggscas.yres.ytech/file/data/incmonb7mlp6gq6zt5kh/PHID-FILE-432fhitwuwthfvdcsfad/image.png)

* Step 1 : renseigner le pattern correspondant à votre alias elasticsearch défini dans votre index template elasticsearch

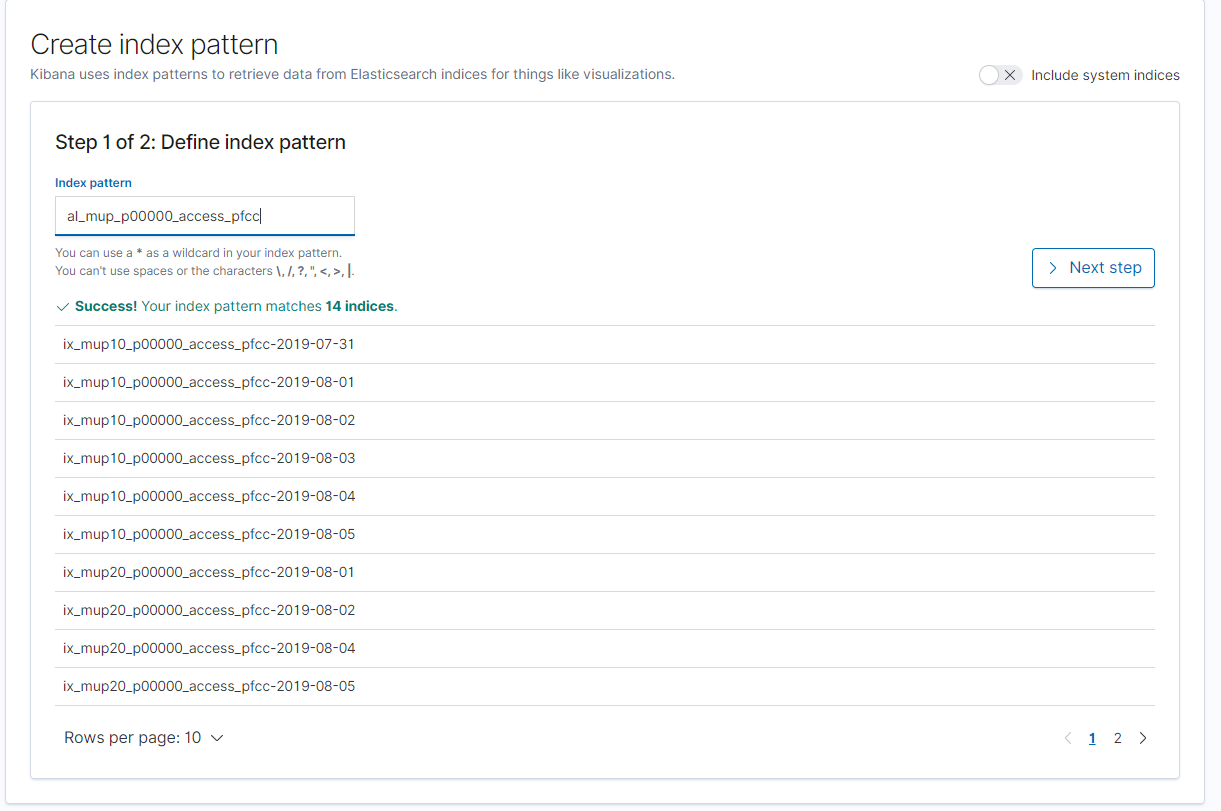
Celui est normalisé de la façon suivante :  
soit al\_[adabo]\_[produit]\_[appl]\_[typelog] (alias sur les index d’une instance adabo du produit)  
soit al\_[env]\_[produit]\_[appl]\_[typelog] (alias sur les index d’un environnement du produit)

par exemple pour les log de type access du code application pfcc composante du produit p0000 en production :  
Vous avez 5 alias

|  |  |
| --- | --- |
| **nomAlias** | **Description** |
| al\_mup10\_p0000\_pfcc\_access | alias répresentant les logs de mup10 |
| al\_mup20\_p0000\_pfcc\_access | alias répresentant les logs de mup10 |
| al\_mup30\_p0000\_pfcc\_access | alias répresentant les logs de mup10 |
| al\_mup40\_p0000\_pfcc\_access | alias répresentant les logs de mup10 |
| al\_mup\_p0000\_pfcc\_access | alias permettant de faire un UNION sur toutes les branches de l'environnement : mup10, mup20, mup30, mup40 |
|  |  |

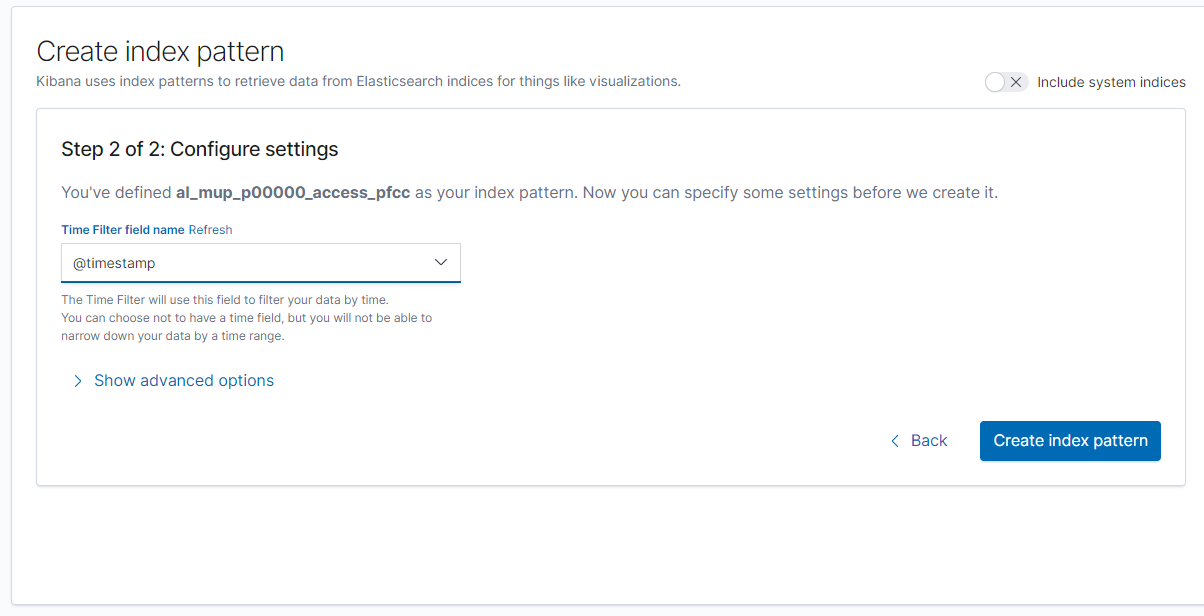
* Taper le nom de l'alias que vous souhaitez déclarer dans le champ index pattern

Dans l'exemple suivant, nous remarquons que Kibana détecte automatiquement les index elasticsearch rattachés à cet alias

[](http://elzudeggscas.yres.ytech/file/data/23li34d2uwtoynsemeut/PHID-FILE-sguh36czolnqfl5a3tjb/image.png)

* Cliquer sur next

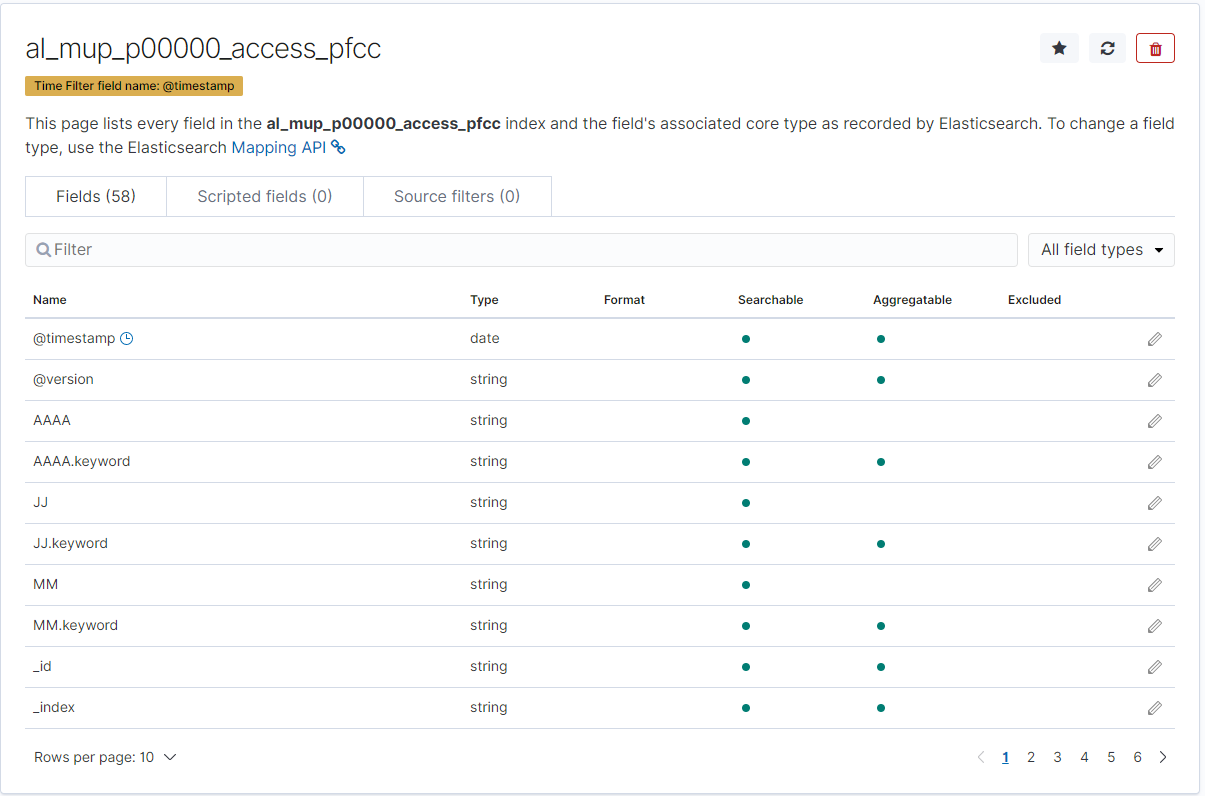
Sélectionner @timestamp dans la liste déroulante time filter field name

[](http://elzudeggscas.yres.ytech/file/data/lqkbh4b6kcwsjo76lsds/PHID-FILE-rtdmo3x5noig2vnf2hff/image.png)

\*Enfin cliquer sur Create Index pattern

[http://elzudeggscas.yres.ytech/file/data/fhqyd3mhj266r645grco/PHID-FILE-5i3ydk4ga7yivyem4dq7/preview-image.png](http://elzudeggscas.yres.ytech/file/data/2ppulj7legulyfr5zykk/PHID-FILE-nigthr4g7f37e5p26s5a/image.png)

Félicitations! vous avez créer votre premier index pattern Kibana dans votre espace Kibana!  
Vous pouvez maintenant instrospecter les logs indéxés dans la partie discover de Kibana

[](http://elzudeggscas.yres.ytech/file/data/hz2yu3fqpsci5wqs5qrz/PHID-FILE-5qt7k65sfhgl6xpsmc46/image.png)

**K - Le Discover Kibana**

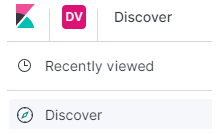
# **Updated 244 Days Ago**[**Public**](http://elzudeggscas.yres.ytech/policy/explain/PHID-WIKI-ohthzp4e6qxtxxn4nkvu/view/)

**Première visualisation de données indexées**  
A ce stade

* vous avez indexé des données dans elasticsearch
* vous avez créer l'index pattern Kibana en utilisant l'alias regroupant les index à visualiser

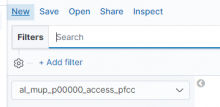
**Le Discover Kibana vous permet maintenant de visualiser le contenu de l'index elasticsearch à partir de l'index pattern Kibana.**

* Cliquer sur Discover dans la console ECE

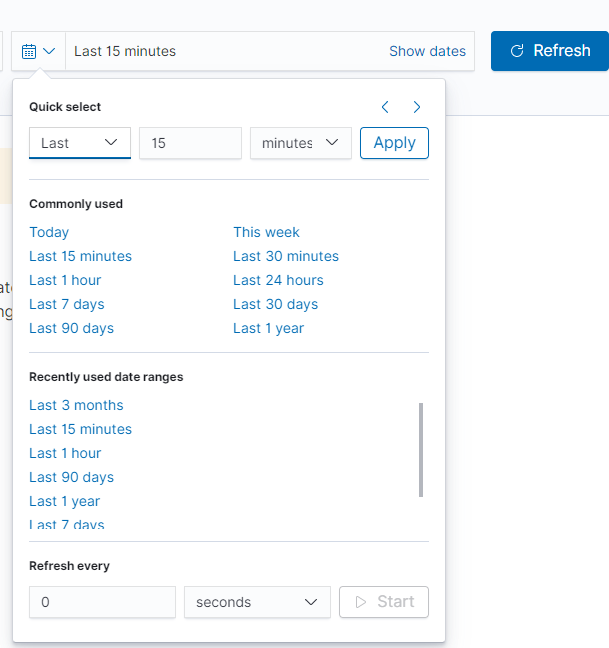
[](http://elzudeggscas.yres.ytech/file/data/h7awpkaysjmnojk34yll/PHID-FILE-f2hwijhqinl2xctpepce/image.png)

* Sélectionner l'index pattern que vous avez créer, dans la liste déroulante des index pattern disponible.

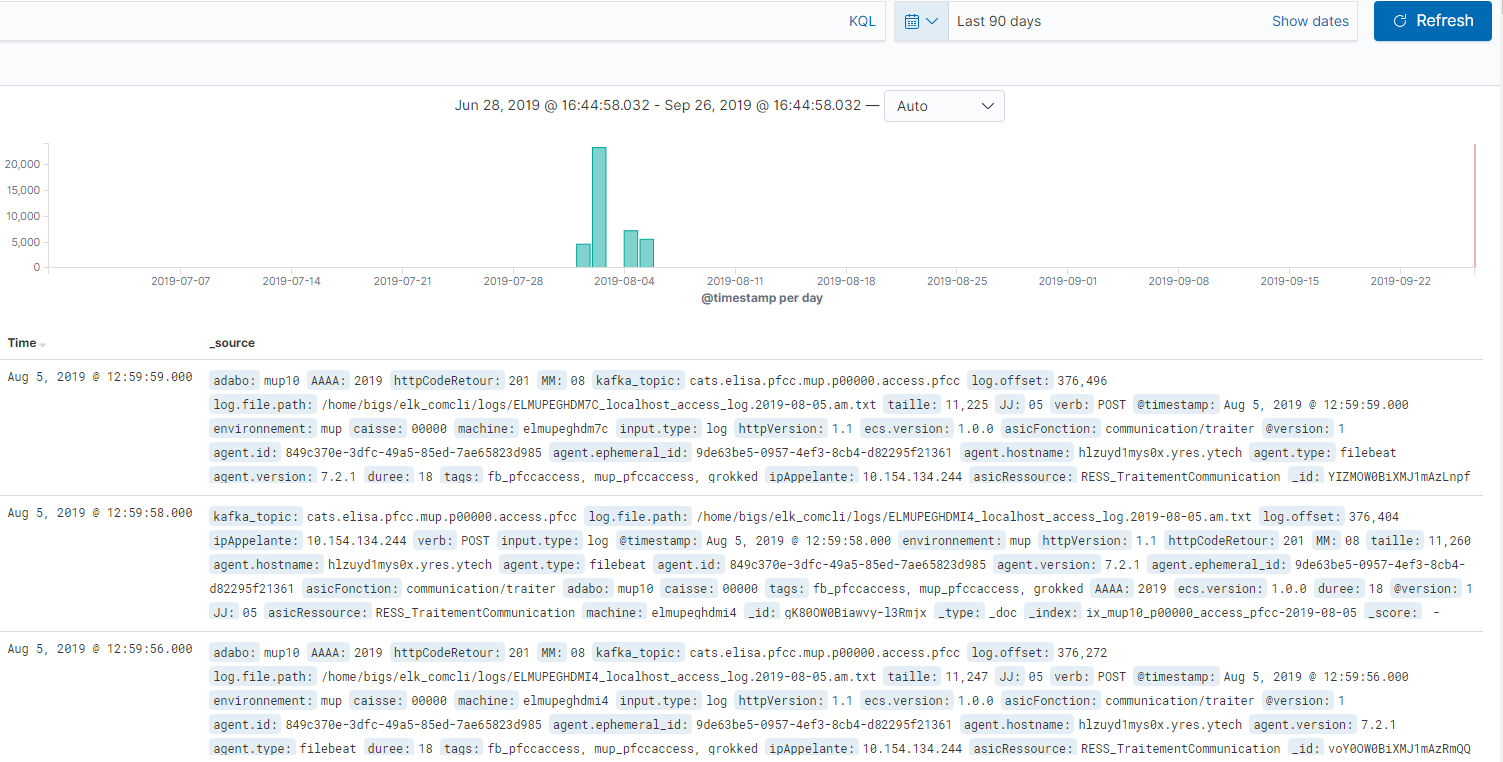
Noter que seuls les index pattern créés dans votre Espace kibana sont visibles dans cette liste

[](http://elzudeggscas.yres.ytech/file/data/7auokn6a7nofipzdxjzz/PHID-FILE-ercauauowlejtgrnnv6a/image.png)

* Sélectionner le ou les critère date de sélection des index, en cohérence avec les dates des logs produites dans l'index

[](http://elzudeggscas.yres.ytech/file/data/xz7itjlrntidyeyueaeb/PHID-FILE-yppl7y6wtms2o7nvu5u7/image.png)

* Voici un exemple de ce que vous pouvez obtenir

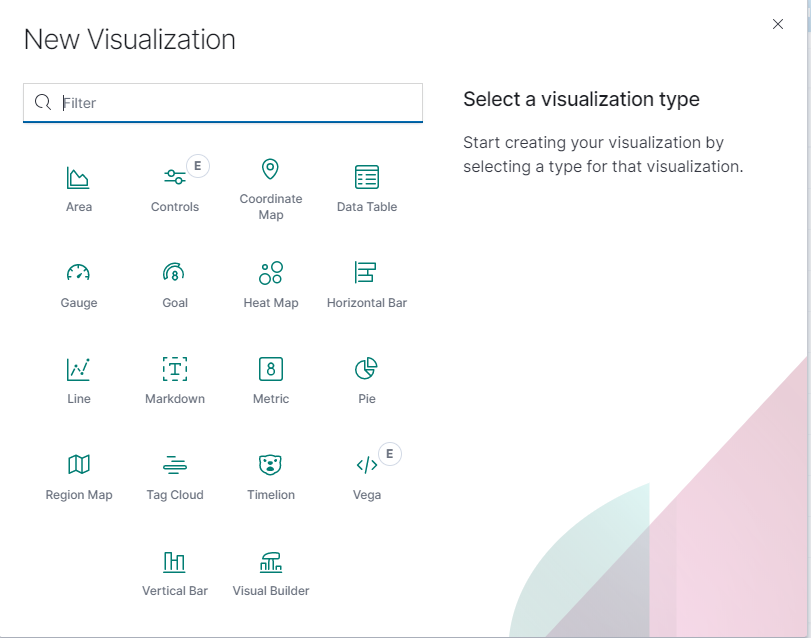
[](http://elzudeggscas.yres.ytech/file/data/thza576axqhg5l2esnoi/PHID-FILE-aemibejnkiziokb3os2r/image.png)

**Vous pouvez maintenant visualiser les logs, les analyser, faire des filtres, exporter les données, sauvegarde vos critères de recherche pour une utilisation ultérieure,...  
Je vous laisse découvrir l'ensemble des fonctionnalités disponibles du discover à travers le lien ci-dessous :**

**Création d'une Visualization**

**Une visualization permet de créer des objets graphiques constitués de requêtes jouées sur l'index-pattern que vous aurez sélectionné**  
C'est à partir de Visualizations que vous pourrez constituer un DashBoard.

Vous pouvez en créer de nature différente

[](http://elzudeggscas.yres.ytech/file/data/nxw3yyd7azurx52gpmr6/PHID-FILE-mn5bnic5ssgjyz7mdp6j/image.png)

**En voici un exemple fait sur la base d'un Vertical bar à partir des données de l'index pattern al\_mup\_p0000\_access\_pfcc**  
La Visualization montre la répartition du nombre de logs par intervalle de 10 m, il s'agit d'une agrégation basique axée sur le @timestamp des logs.  
Vous pouvez en imaginer d'autres....

**Respect de la Normalisation des objets**  
Lorsque vous sauvegarderez votre visualization, respecter la norme de nommage suivante.  
N'oubliez-pas que la visualization porte sur un index pattern, lui-même lié à un alias  
La partie libellé est libre soyez explicite, car si dans un même espace vous êtes plusieurs à développer des visualizations, il va falloir vous y retrouver par la suite.  
Vous devez aussi penser à ceux qui vont utiliser vos visualizations dans les Dashboards...  
*C'est une recommandation qui nous a été donnée par la Squad COMCLI lors du POC ELK*

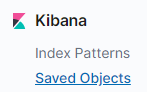
|  |  |
| --- | --- |
| **norme Visualization** | **Description** |
| vk\_[adabo]\_[libelle] | si votre visualization porte sur un adabo, par exemple mup10 |
| vk\_[env]\_[libelle] | si votre visualization porte sur l'ensemble des adabo, par exemple mup, mup10, mup20, mup30, mup40 |
|  |  |

N'oubliez-pas que la visualization porte sur un index pattern, lui-même lié à un alias  
exemple possible :  
vk\_mup\_p00000\_pfcc\_access\_verticalBar\_Repartition\_NbreHit\_10m

[](http://elzudeggscas.yres.ytech/file/data/b6mxawwrqsb65oxvqybd/PHID-FILE-72xum3hsklymdbwpj5uh/image.png)

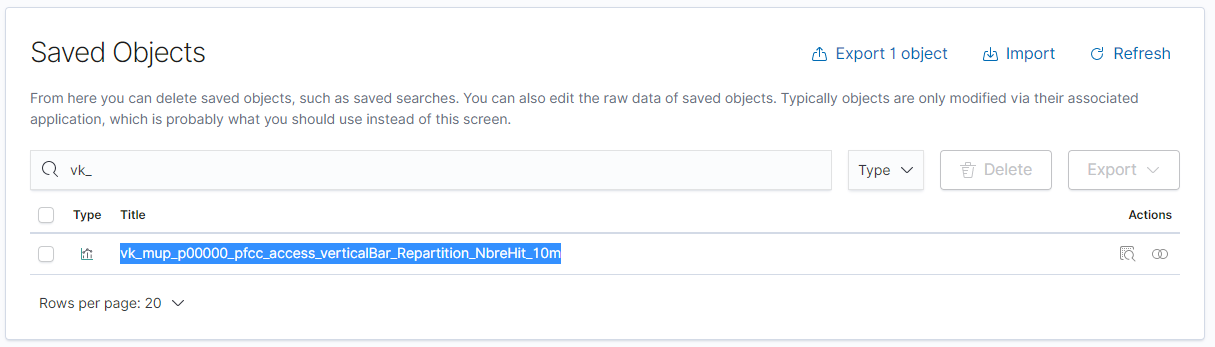
**Le management des objets kibana sera hautement simplifié par la normalisation**

* Cliquer sur Saved objects

[](http://elzudeggscas.yres.ytech/file/data/3q3bhp2qx2wmkfjicowu/PHID-FILE-fbxi5fnlupd3okeyo2lu/image.png)

* Rechercher les visualizations par le préfixe vk ou bien par le nom complet ou partiel de l'objet

Un nommage correct de l'objet permet sa bonne identification

[](http://elzudeggscas.yres.ytech/file/data/7eqcutve22d5hvn2cbqz/PHID-FILE-v7cliuzlklv5nwpuqz6l/image.png)

Pour de plus amples informations référez-vous au site suivant :

**M - Création d'un Dashboard**

**Création d'un DashBoards à partir de vos Visualizations**

Les DashBoards sont composés de Visualizations  
Si vous vous sentez l'âme d'un Picasso, vous pourrez aussi construire des Canvas personnalisés de toutes pièces, avec l'image de l'entrprise, le designe que vous souhaitez....

* Pour construire un DashBoard :
* Cliquer sur DashBoard

[http://elzudeggscas.yres.ytech/file/data/i6gpixys2mvcf535hmut/PHID-FILE-fbk6wq5d2c5ry6yynqai/preview-image.png](http://elzudeggscas.yres.ytech/file/data/gxvpdnfvkcb4xcoonerp/PHID-FILE-tkkhfxyqghbtepu7rept/image.png)

* Cliquer sur create new DashBoard

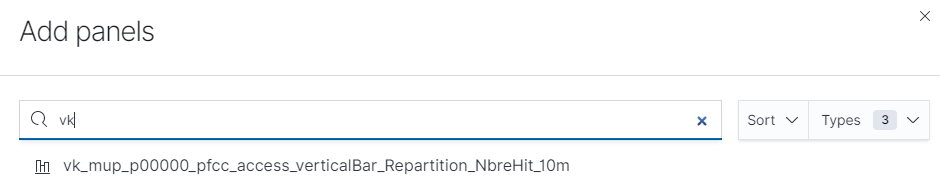
[http://elzudeggscas.yres.ytech/file/data/bucgsdnnvzm63gsv22r5/PHID-FILE-bv73jeohqctvsutxcnxx/preview-image.png](http://elzudeggscas.yres.ytech/file/data/fanhmox5ruglf5bwhjg7/PHID-FILE-pimnr7lygqoerryaemr3/image.png)

* Vous pouvez à présent importer des Visualizations en cliquant sur Add

[http://elzudeggscas.yres.ytech/file/data/yqy4sapki5ltjaoqcnw2/PHID-FILE-nh342z32w2tkvhbmvm47/preview-image.png](http://elzudeggscas.yres.ytech/file/data/q3ly7hwtr2os4a7iiwpx/PHID-FILE-nxksw4bz46irq4tt3q37/image.png)

* Sélectionner les visualizations que vous souhaitez ajouter à votre dashBoards

Exemple :

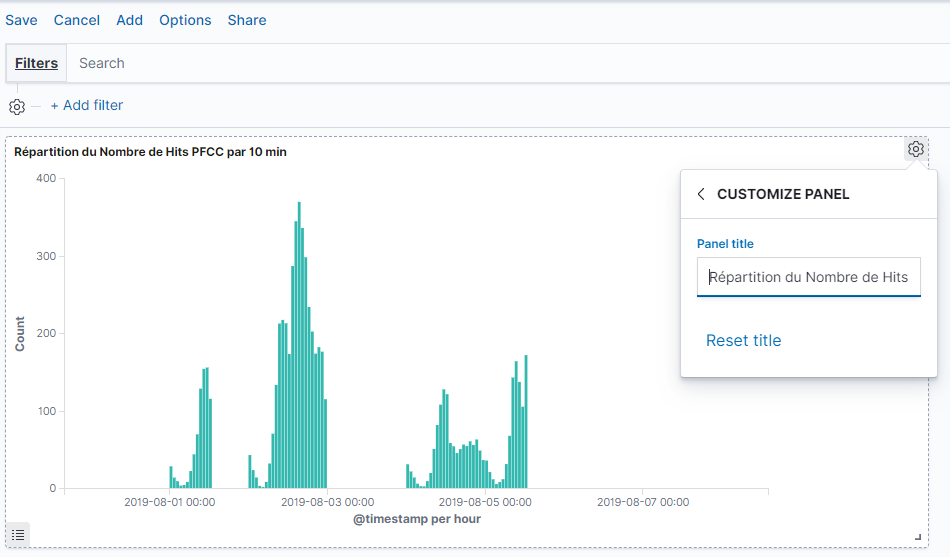
[](http://elzudeggscas.yres.ytech/file/data/qou55ju6dtvfnbuyqt6z/PHID-FILE-43lbck6on7wpl7guvs6j/image.png)

}

[](http://elzudeggscas.yres.ytech/file/data/epvsy2j77qpoalz6a6of/PHID-FILE-7lvd227ot7rewzll5qqm/image.png)

}

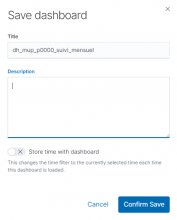
Le Titre de la Visualization n'est pas très friendly car par défaut c'est le nom de l'objet, rassurez-vous vous pouvez le modifier

[](http://elzudeggscas.yres.ytech/file/data/ol4hrx44f5stgzcnyc3d/PHID-FILE-rg4cufsk2apslemby7em/image.png)

Lorsque vous avez finalisé votre Dashboard, vous devez le sauvegarder en respectant la norme ci-dessous, l'objectif étant de pouvoir les retrouver facilement par la suite

|  |  |
| --- | --- |
| **Norme proposée** | **Description** |
| dh\_[adabo]\_[libelle] | Commence par dh pour dashboard, mettre adabo permet d'identifier la branche d'exécution |
| dh\_[env]\_[libelle] | dh pour dashboard, mettre env, mup par exemple si le dashboard concerne toutes les branches d'un environnement |
|  |  |

*NB : Nous essayons d'appliquer cette normalisation, dans les faits il faut voir si elle s'avère pratique à l'usage, car les dashboards peuvent être diffusés plus largement hors de la squad. Un nom parlant peut être plus pratique pour la communication*

[](http://elzudeggscas.yres.ytech/file/data/7cks6u7bc24g5qeevo5f/PHID-FILE-5qiteuks6yqgu77b45sb/image.png)

Pour de plus amples informations référez-vous au site suivant :

**N- Exemple De Requête Rest Elasticsearch**

**Exemple de requêtes Rest Elasticsearch**

Elasticsearch expose des Api Rest. Comme nous l'avons vu précédemment par exemple nous avons appelé cette API pour créer le template Index elasticsearch.

Les exemples suivants portent

* sur des requêtes d'administration

par exemple lister les index associés à un alias, lister l'ensemble des index ou bien obtenir la description d'un template

* sur des requêtes de search

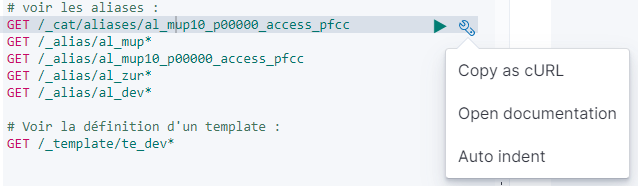
**Vous pourrez librement faire un copier/coller de ces requêtes dans la partie DevTool de Kibana afin de les utiliser après les avoir adaptées à vos besoins**

[http://elzudeggscas.yres.ytech/file/data/nwq3fkh3odq7cfefn5a4/PHID-FILE-mxpi5fnq2j4r24kjf4vv/preview-image.png](http://elzudeggscas.yres.ytech/file/data/cr7xx4k5ermh7alcv5gj/PHID-FILE-6iwtlsfobjyzytnr2h4s/image.png)

**Vous pourrez les exécuter directement dans l'UI ECE en cliquant sur la flèche (click to send request)**

[](http://elzudeggscas.yres.ytech/file/data/xtqsmeopmt5wtcwme3kh/PHID-FILE-edj45lz7fcu2ogrkfyor/image.png)

**Vous pouvez également bien récuperer la requête dans un format cUrl en cliquant sur copy as cURL**  
**Vous pouvez accéder à la documentation de chaque ressource d'API en cliquant sur "Open Documentation"**

[](http://elzudeggscas.yres.ytech/file/data/ypl2szakqfqvy66rahjm/PHID-FILE-xewl7wqwcnwnz3jmsqjl/image.png)

*Exemples de requêtes d'administration*

# Voir tous les index :

GET /\_cat/indices?v

# Voir la définition d'un template :

GET /\_template/te\_dev\*

# Voir les templates au format tabulaire trié par nom :

GET /\_cat/templates?v&s=order:desc,index\_patterns

# Voir les templates commençant par te\*

GET /\_template/te\*

# Supprimer le template [montemplate]

DELETE \_template/[montemplate]

# Voir les aliases commençant par mup :

GET /\_alias/al\_mup\*

# Voir les aliases commençant par mup :

GET /\_cat/aliases/al\_mup10\_p00000\_access\_pfcc

GET /\_alias/al\_mup10\_p00000\_access\_pfcc

# Voir la description des attributs des index associés à un Alias au format Tabulaire

POST \_sql?format=txt

{

"query":"describe al\_mup10\_p00000\_access\_pfcc"

}

*Exemple de requêtes de search*

POST \_sql?format=txt

{

"query":"select \"@timestamp\" from al\_mup10\_p00000\_access\_pfcc order by \"@timestamp\" asc"

}

POST \_sql/translate

{

"query":"select \"@timestamp\" from al\_mup10\_p00000\_access\_pfcc order by \"@timestamp\" asc"

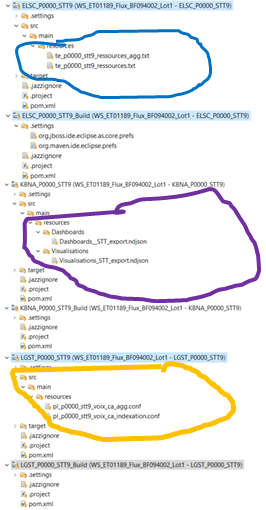
}

**O - Mise En Configuration RTC Elk**

Une fois les développements terminés sur le Cluster de DEV ELK, vous devez mettre la matière en configuration dans RTC.

3 types de composants permettent de mettre la matière ELK en gestion de Conf :

* Matière Pipeline Logstash
* Matière lifecycle policy et template ElasticSearch
* Matière Kibana, Visualisations, DashBoards et index pattern

[](http://elzudeggscas.yres.ytech/file/data/2sqmyo727wsroap4g4pu/PHID-FILE-yp5za6unm3wb5lfdv7rc/image.png)

Chacun des types de composant doit être normalisé de la façon suivante :

|  |  |
| --- | --- |
| Type Matière | Préfixe retenu |
| Elasticsearch | ELSC\_[Produit]\_[InstanceDeDéploiement ou CodeAPPL] |
| Logstash | LGST\_[Produit]\_[InstanceDeDéploiement ou CodeAPPL] |
| Kibana | KBNA\_[Produit]\_[InstanceDeDéploiement ou CodeAPPL] |
|  |  |

**P - Mise En Deploiement Elk**

#### Dans gitv

Pour ces étapes, contacter votre Leader Intégration.

##### Créer un dépot

[http://elzudeggscas.yres.ytech/file/data/pls7xl72mupkz63cme2c/PHID-FILE-r5glyaotjlpsmhxpn3pp/preview-image.png](http://elzudeggscas.yres.ytech/file/data/547atiliri5673y66y2d/PHID-FILE-tj74nkasdtq6zami4gen/image.png)

Le nom du depot doit respect pyyyy\_sxxxx\_elk

##### Mise en place des artefacts personnalisant les variables par environnement.

<http://ttp10-gitvihmrefvar.caas-cicd.ca-technologies.fr/variables>

[](http://elzudeggscas.yres.ytech/file/data/6sxkuz2eove7l5kqy3ji/PHID-FILE-ontfupiawlzzyzuafftb/image.png)

Section de description du produit  
produit: "pyyyy"  
solution : "sxxxx"  
ada: "env\_adabo"  
appl: "code\_appli"

Variables liées au Space Kibana du produit  
kbna\_produit: "**pyyyy**"  
kbna\_libelleSpaceKibana: "**Libellé de l'espace kibana devant contenir le code produit ({{produit | upper}})**"  
kbna\_descriptionSpaceKibana: "**Description de l'espace kibana**"  
kbna\_colorSpaceKibana: "**Code Couleur de l'espace kibana**"  
kbna\_initiales2CSpaceKibana: "**Initiales sur 2 caractères de l'espace**"

pipelines:

\*\*Nom du fichier de pipeline du module gitlab logstash\*\*:

description\_pipeline: "\*\*{{produit | upper}} - {{solution | upper}} : Description du pipeline {{appl | upper}}\*\*"

id: "\*\*Numéro de pipeline à caler avec les acteurs CAGIP\*\*"

Facultatives car surchargées par des valeurs par défaut

pipeline\_batch\_delay: "51"

pipeline\_batch\_size: "1000"

pipeline\_workers: "1"

queue\_checkpoint\_writes: "1024"

queue.max\_bytes: "1gb"

queue\_type: "memory"

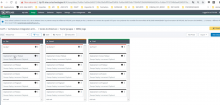
# Fin Facultatives car surchargées par des valeurs par défaut

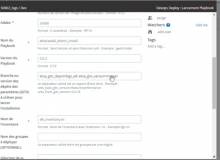
Liste des composants à livrer  
elasticsearch\_archive\_url: "**URL de l'archive Nexus elasticsearch**"  
kibana\_archive\_url: "**URL de l'archive Nexus des livrables kibana**"  
logstash\_archive\_url: "**URL de l'archive Nexus de logstash**"

#### Dans XL release

Prendre contact avec son Leader Intégration pour mettre en place les pipeline de livraison.

**TODO à compléte**r => Pas d'accés aux flow pour mettre en place une démarche pas à pas

[](http://elzudeggscas.yres.ytech/file/data/wgmze6o4ahtgzltsdhpq/PHID-FILE-2dsfcdipdkrpr6ezmn4o/image.png)

[](http://elzudeggscas.yres.ytech/file/data/jeudjig3bcoja5frv442/PHID-FILE-gukhf7bdno2hw5sxipjg/image.png)

**Limites d'ELK Et Contournements Possibles**

#### Les limites d'utilisation d'ELK

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| MiddleWare | Point Faible | Interdit | A éviter | préconisé | Autorisé |
| FileBeat | Gestion du Back Pressure | La contrainte de respect de l'ordre d'émission de logs | Collecter des métriques de bas niveaux( voir les chantiers de métrologie : APM, Prometheus, Graphite,é) |  |  |
|  | Gestion centralisée des CONF non implémenté |  |  |  |  |
| Kafka | Pas de Gestion fine des partitions |  |  |  |  |
| Logstash | Mise au point du traitement de transformation | Des pipelines traitant des formats de logs différents. |  | Factoriser des formats de logs identiques dans un méme Pipeline | Dispatch des logs dans des index différents. |
|  | (pas d'outillage) | La dénormalisation |  |  | Calcul mathématique é partir d'une log autoporteuse des informations nécessaires (exemple durée calculée entre 2 attributs de la log) |
|  | Affectation statique des pipelines | Le complexe evt processing (lié des evts entre eux) |  |  |  |
|  | Password en Clair |  |  |  |  |
|  |  |  |  |  |  |
| Elasticsearch | Les jointures |  | L'indexation de format hétérogéne au sein d'un méme index | Un index par produit/appl/format | Un index par format? |
| Index |  |  |  |  |  |
| Elasticsearch | Pas de possibilité de définir un champ calculé (é vérifier idem processor SOLR) |  |  |  |  |
| template | Pas de reprise é chaud de la modification d'un type de champs dans le schéma (contrairement é SOLR). |  |  |  |  |
| Elasticsearch | Pas de jointure |  |  | Union |  |
| Alias | Les alias des templates ne rafraichissent pas automatiquement |  |  | Filter |  |
|  | Pas d'Alias basé sur des alias |  |  |  |  |
| Kibana | Pas de possibilité d'ajouter des champs calculés (scriptedfield) éa ne marche pas. | Les scripted fields car peut faire planter Kibana |  |  |  |
| Index-pattern |  |  |  |  |  |
| Kibana | Vision corrélée de l'ensemble des logs d'une chaéne applicative dont les formats sont différents ou issus de produits différents |  |  |  |  |
| Discover |  |  |  |  |  |
| Kibana | Les requétes de type search ne fonctionnent pas alors qu'elles sont ok par l'API |  |  | Utilisation d'un ALIAS | Utilisation de filtres |
| Visualisation | Impossible de constituer un Dataset é partir d'une requéte complexe en input d'une visualisation, seulement un index, un alias ou un search statique |  |  |  |  |
|  | Pas de possibilité d'ajouter un champ calculé |  |  |  |  |
|  | Légende tronquée |  |  |  |  |
| Kibana | Variabilisation de la matiére difficile (exemple le code ADABO) |  |  |  |  |
| Kibana Space | Un seul niveau de Space, pas d'approche en Drill-Down. |  |  |  |  |
|  |  |  |  |  |  |