



# INFRASTRUCTURE DE PRODUCTION

---

## TEST ET DIAGNOSTIQUE

00.00.2017

Photo by Diogène Molton on [Unsplash](#)

# TABLE DES MATIÈRES

---

- 
- 1 | CONSTAT
  - 2 | ELASTICSEARCH
  - 3 | LOGSTASH
  - 3 | KIBANA

Photo by Martha Dominguez de Gouveia on [Unsplash](#)



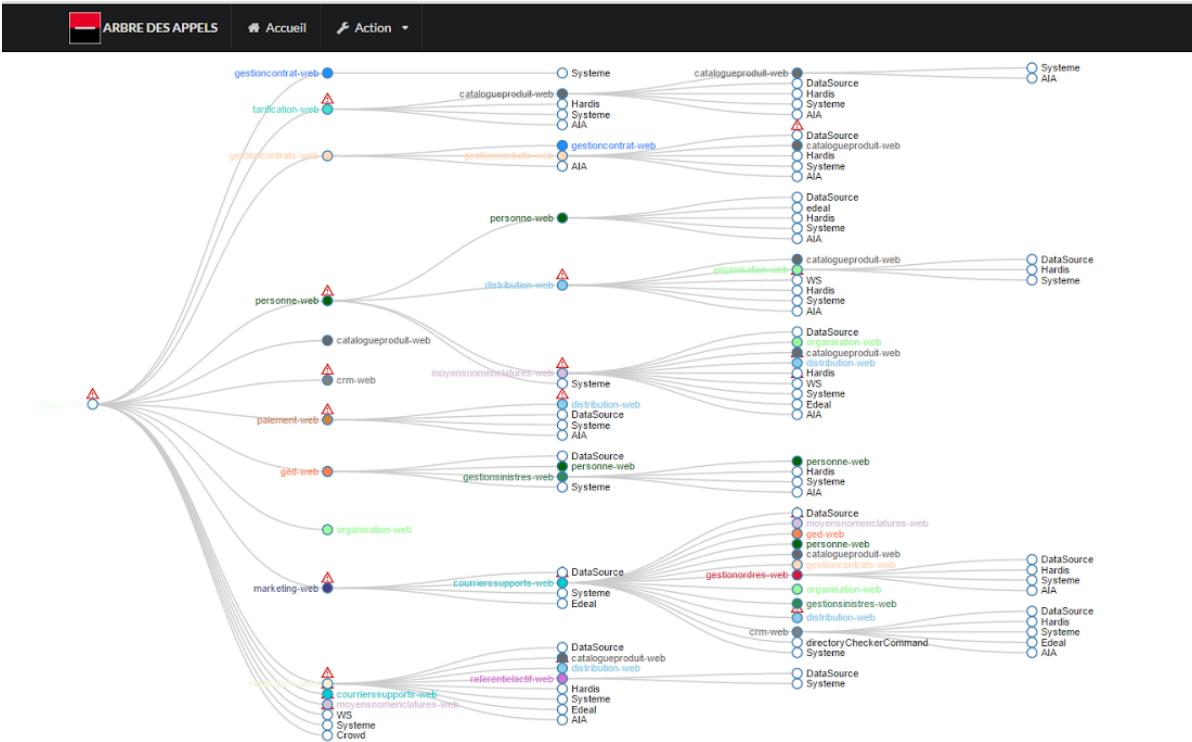
A photograph of a multi-story red brick building with white trim and fire escapes. A yellow traffic light is visible in the bottom left corner. The building's facade is textured and shows signs of age.

1

---

CONSTAT

Photo by Diogène Mouton on [Unsplash](#)



# LES CAS D'USAGES



# LES CAS D'USAGES

---



## DIAGNOSTIQUER

Traiter les incidents sur tous les serveurs d'une application

Identifier précisément les l'origine d'une erreur



## ANALYSER

rechercher preuves indices, ... dans le cas d'une alerte sécurité

Analyse d'un incident fonctionnel (exemple avec un problème sur les versements)



## DÉTECTER

déetecter les ruptures de service sur les écarts de performance

Détecteur des tentatives de connexion



## ALERTER

Absence de réponse d'un serveur

Codes seuil de code erreur 500 dépassée sur une période

timeout dépassée sur une période  
seuil de souscription non conforme à un seuil min ou max par rapport à la semaine



## PREVENIR

historique à 13 mois pour du machine learning



## RESTITUER

Restitution d'indicateur de risque et de performance

Indicateur de performances des services

Indicateur des erreurs sur les appels

Indicateur du nombre d'appels

Indicateur sur les timeout



## SUIVRE

Contrôle permanent

# NIVEAU DE MATURITÉ



- Ajouter les données relatives aux erreurs
- Travailler les erreurs et les formater pour une meilleure classification et un meilleur comptage
- Compléter les logs avec des données métier
- Self-healing

## Alerter sur divergance

## Détection & réaction

## Prévention & pro-action

## Niveau 04

- Ajouter dans le système des logs la performance du système
- CPU
- RAM
- Disk
- ...
- Self-healing

## Prévoir le futur

# LES PILIERS DE LA SUPERVISION



## Latence

- Impacte directement l'expérience utilisateur
- Indicateur fort d'incidents imminents
- Indicateur de demande croissante en capacité



## Trafic

- Indicateur de l'activité
- Historique de l'activité utilisé pour les prévisions
- Décisions de capacité basées sur le *cycle de vie de la demande*
- Indique le *coût de l'infrastructure cloud*



## Capacité

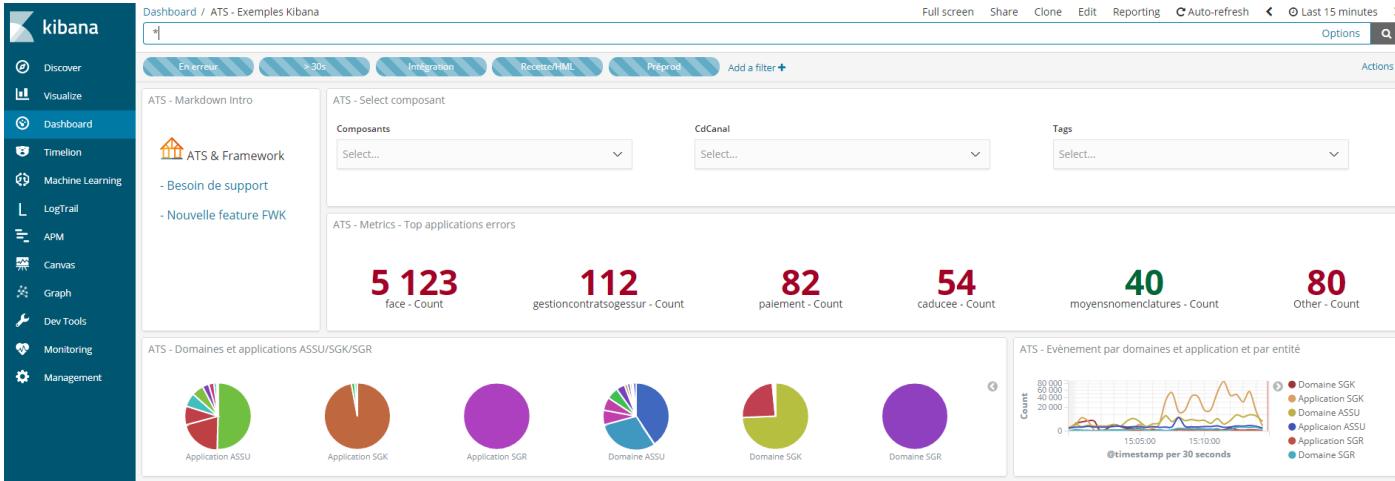
- Indicateur d'atteinte de **capacité maximum**
- L'analyse permet de prendre des **mesures d'autoscaling**
- Prédit la **dégradation des performances** quand la demande excède la capacité



## Erreurs

- Les erreurs des applications montrent un **disfonctionnement**
- Les erreurs au niveau de l'**infra** indiquent des pbs. de **configuration** ou des limites de capacités
- Des erreurs au niveau des **Services** peuvent être sources de pb. de **permissions/accès**

# LES PILIERS DE LA SUPERVISION

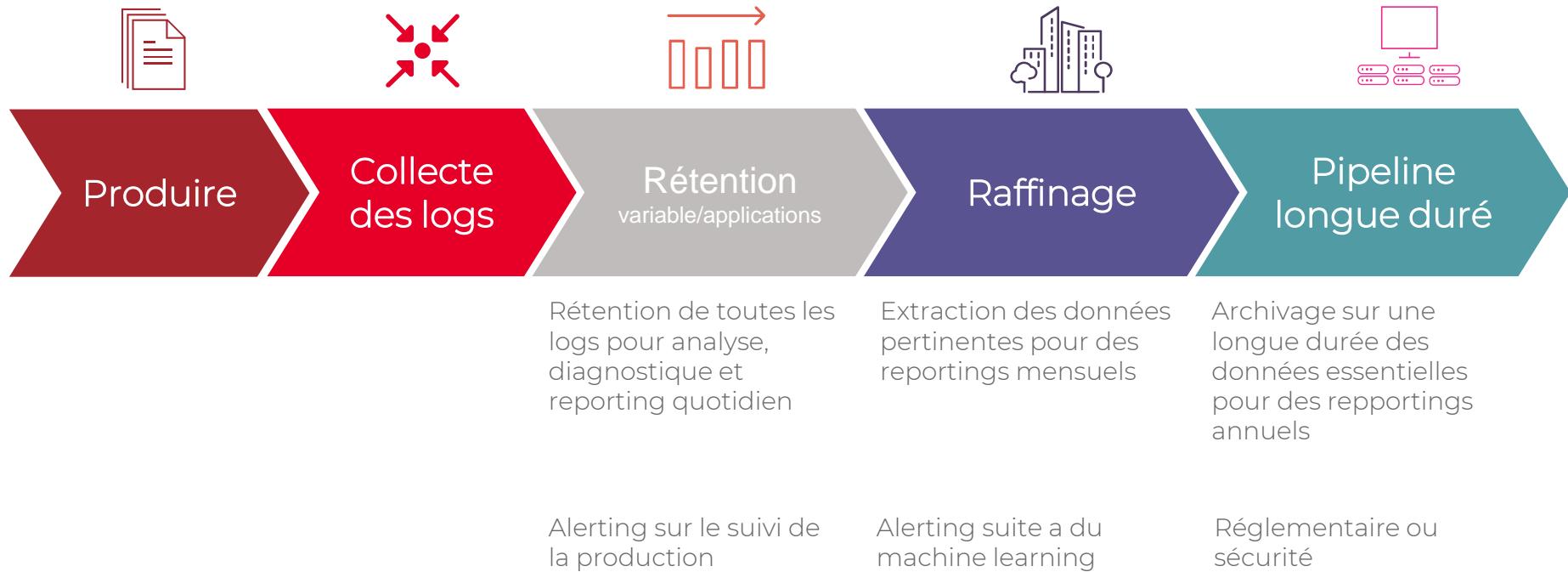


Il est conçu pour présenter les indicateurs les plus importants pour ses superviseurs.

Le tableau de bord des mesures est une application qui fournit une vue récapitulative des métriques de base d'un service.

Un Allié clé: Le tableau de bord de mesures de supervision

# LES ÉTAPES DE LA SUPERVISION AVEC ELASTIC



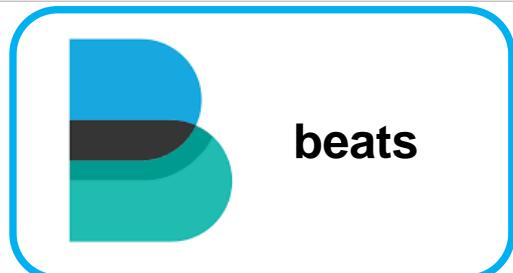


2

---

# ELASTICSEARCH

Photo by Fancycrave on [Unsplash](#)



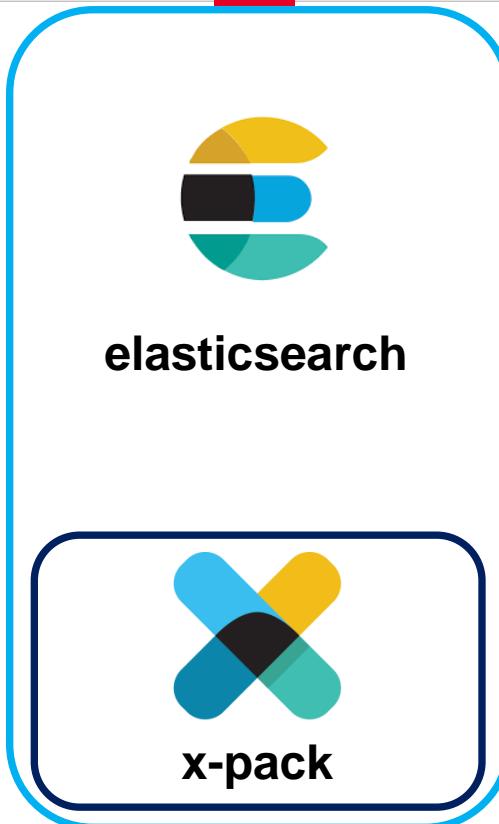
beats



logstash



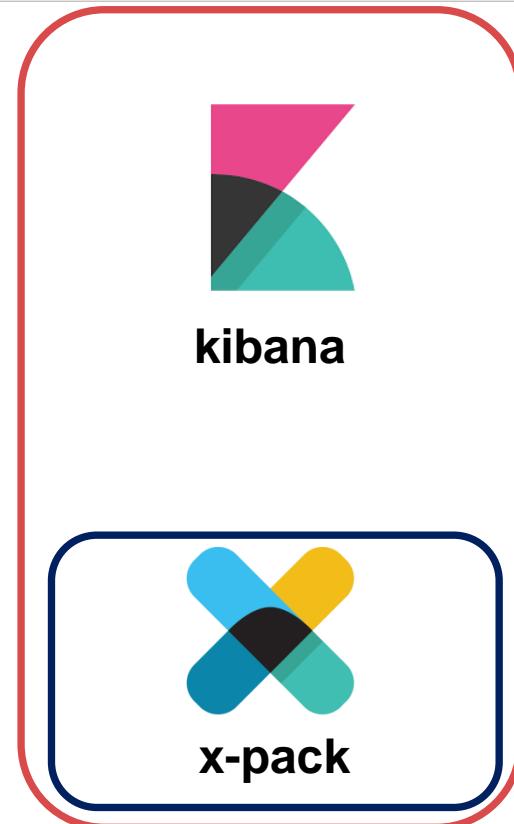
es-hadoop



elasticsearch



x-pack



kibana



x-pack



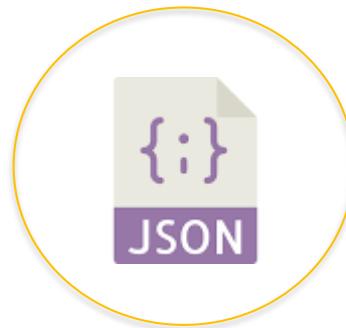
# Elasticsearch

## Performance



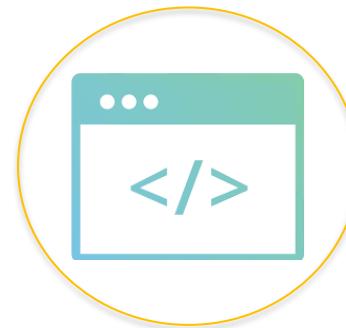
« Near real-time »  
Haute disponibilité  
Scalabilité horizontale

## Fonctionnement



Stockage JSON  
Moteur d'indexation  
Moteur de recherche  
Moteur d'agrégation

## Accessibilité



API REST  
Clients TCP  
Multi-langages



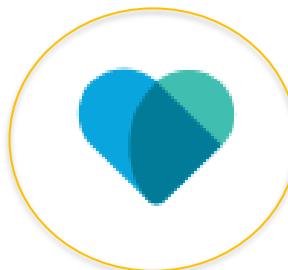
Sécurité



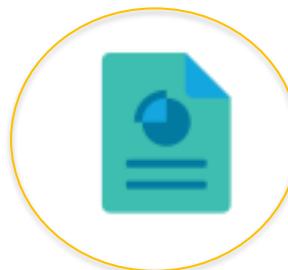
Alerting



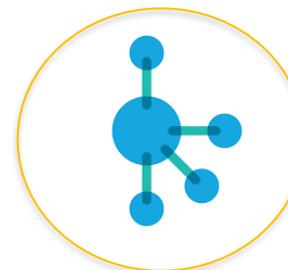
Supervision



Reporting



Graph



ML



# LES RECHERCHES

The screenshot shows a search results page for the query "elasticsearch".

- Search:** The search bar contains "elasticsearch".
- Recherche structurée (Structured Search):** A sidebar on the left lists repository facets:
  - Stories: 10,610
    - Issues: 7,132
    - Wikis: 8,664
    - Users: 92
  - Languages:
    - Java: 1,718
    - JavaScript: 1,325
    - Shell: 1,127
    - Python: 1,073
    - Ruby: 992
    - PHP: 225
    - Go: 182
    - C#: 182
    - Scala: 225
    - HTML: 182
- Suggestion:** Below the search bar, it says "We've found 10,610 repository results".
- Trie (Sorting):** A dropdown menu shows "Sort: Best match".
- Enrichissement (Enrichment):** Below the first search result, it says "ElasticSearch Dockerfile for trusted automated Docker builds." with stats: ★ 365, 294, Updated 24 days ago.
- Agrégation (Aggregation):** Below the second search result, it says "Simple PHP client for Elasticsearch" with stats: PHP, ★ 279, 84, Updated 11 Aug.
- Pagination:** At the bottom, there is a navigation bar with buttons for "Previous", page numbers (1, 2, 3, 4, 5, ..., 99, 100), and "Next".

## Données en temps réel

Les données arrivent en temps réel dans le système



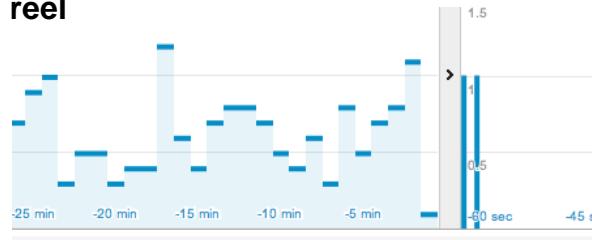
### Haute disponibilité

Elasticsearch permet la construction d'un cluster résilient. Les nœuds cassés sont automatiquement retirés et les données réparties



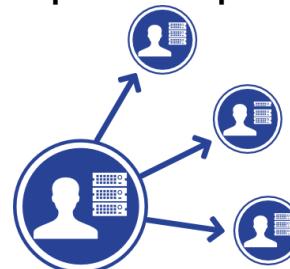
## Analyse en temps réel

Les données peuvent être agrégées et analysées en temps réel



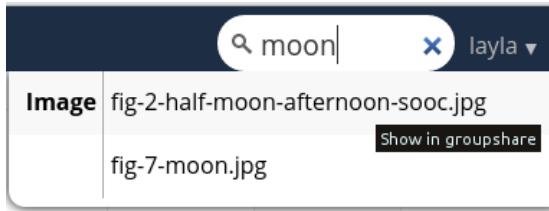
### Multi-Tenant

Un cluster peut contenir de multiple indices qui peuvent être requêtés indépendamment



## Recherche plein texte

Utilisation de lucene pour permettre une recherche plein texte performante



## Gestion du conflit

Chaque version de chaque document test gérée pour permettre de ne pas perdre des données en raison de conflit



## Orienté document

Les données sont stockées sous forme de json dans leurs intégralités

```
{title: 'Babylon 5',  
  seasons: [  
    {season_number: '1',  
     episodes: [  
       {ordinal_within_season: '1',  
        title: 'Midnight on the Firing Line'  
        reviews: [...],  
        cast_members: [...]  
      }  
    ]  
  ]  
}
```

## Sans schéma

Toute sortes de données peuvent être stockées.  
ES détecte automatiquement la structure des données et adapte les index en conséquence,



# UN SEUL FORMAT

---

- Format JSON ultra flexible
- Structuré en documents
- Avec des données typées
- Similaire à une ligne en base de données

```
{  
  "prenom" : "Diogène",  
  "nom" : "MOULRON",  
  "age" : 39,  
  "tags" : ["spring", "elastic" ,  
            "java"],  
  "email" : "diogene.moulron@gmail.com",  
  "github" : "moulron" ,  
  "geek" : true  
}
```

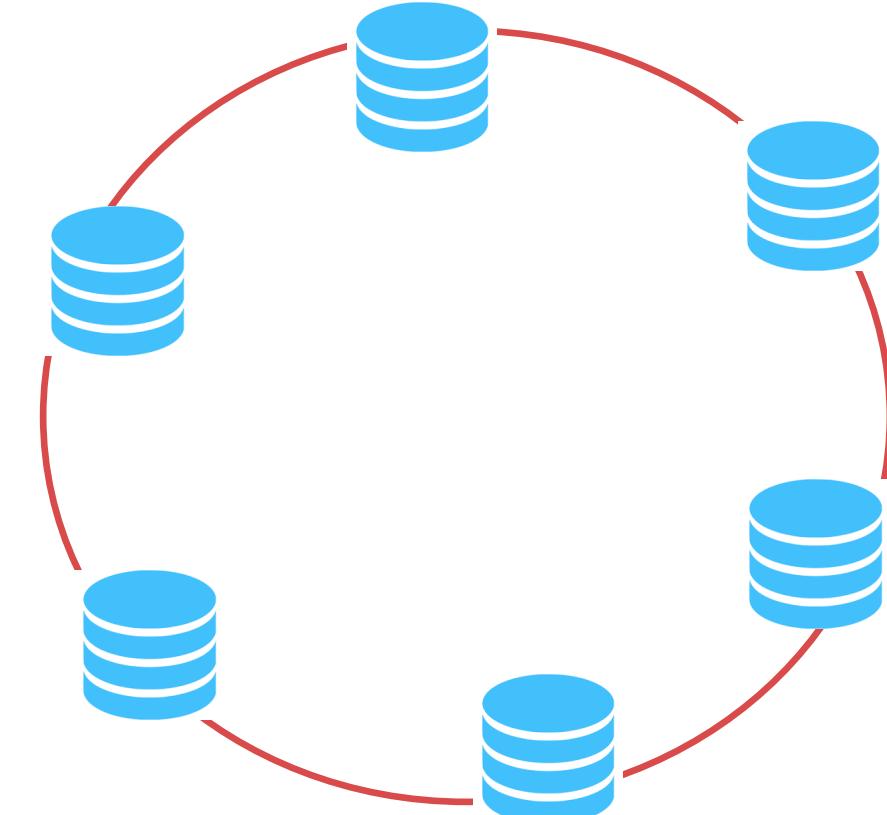
# L'INFRASTRUCTURE D'ELASTICSEARCH

## ■ Cluster

- Elasticsearch fonctionne en cluster
- Un cluster est formé de plusieurs nœuds
- Généralement 1 nœud = 1 serveur
- Un cluster est identifié par un nom unique

## Nœuds

- Master : consistance et distribution des données
- Data : stocke une partie des données & exécute les requêtes clients
- Client : équilibrage de charge des requêtes clients
- Ingest : intégration et transformation de données
- ML : dédié à l'exécution des jobs de Machine Learning



## UN PEU DE VOCABULAIRE

---

### indice



Ensemble de données consistantes  
Lié à un type de données spécifiques ou à une date

### Shard



Partition d'un indice  
Il peut être primaire ou réplica

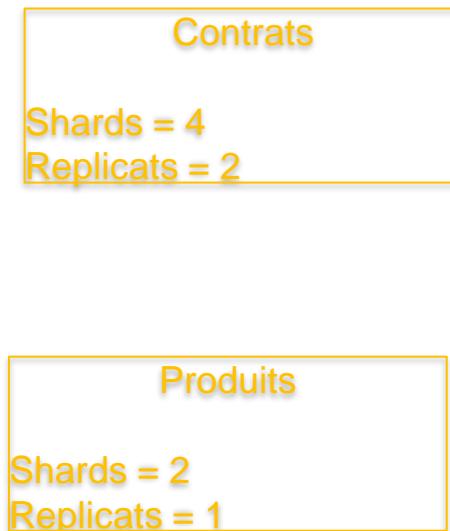
### Segment



Partie de données d'un shard  
Il correspond au fichier présent sur disque

# DISTRIBUTION ET SCALABILITÉ

---



## ■ Replication

- Chaque donnée est répliquée

## ■ Sharding

- Positionne les données sur plusieurs machines et dans plusieurs zones

## Noeud 1

### Contrats

- 1
- 2
- 3
- 4

### Produits

- 1

## Noeud 2

### Contrats

- 1
- 2
- 3
- 4

### Produits

- 2

## Noeud 1

### Contrats

1

2

4

### Produits

1

## Noeud 2

### Contrats

2

3

### Produits

2

## Noeud 3

### Contrats

1

3

4

### Produits



3

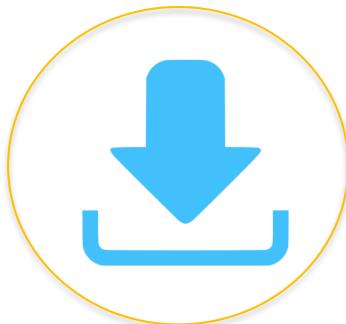
—  
LOGSTASH

Photo by Imgix on [Unsplash](#)



# Logstash

Collecte de données



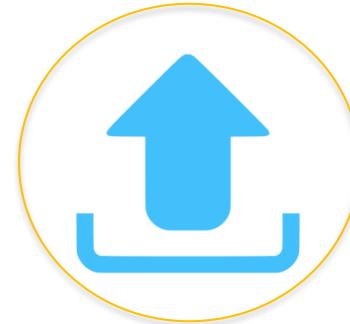
Input { }

Analyse et transformation



Filter { }

Transport de données

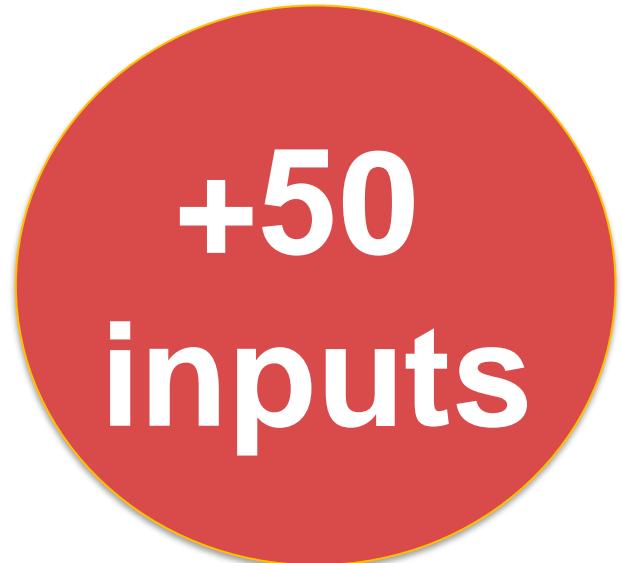


Output { }

## INPUT {

---

- beats { ... }
- elasticsearch { ... }
- exec { ... }
- file { ... }
- jdbc { ... }
- github { ... }
- http { ... }
- kafka { ... }
- log4j { ... }
- meetup { ... }



## FILTER {

---

- aggregate { ... }
- anonymize { ... }
- csv { ... }
- date { ... }
- dns { ... }
- fingerprint { ... }
- geoip { ... }
- grok { ... }
- mutate { ... }
- xml { ... }



## OUTPUT {

---

- csv { ... }
- elasticsearch { ... }
- email { ... }
- exec { ... }
- file { ... }
- google\_cloud\_storage { ... }
- http { ... }
- kafka { ... }
- mongodb { ... }
- s3 { ... }



## 3 PARAMÉTRAGES DISPONIBLES

---

### logstash.yml

node.name  
pipeline.workers  
config.string  
path.config  
path.data  
path.queue  
path.logs

### Command line

--path.config  
--config.string  
--pipeline.workers  
--path.data  
--path.plugins  
--path.logs  
--config.debug  
--config.test\_and\_exit  
--path.settings

### jvm.options

-Xmx  
-Xms  
-Duser.language  
-Duser.country  
-Dfile.encoding  
-XX:+HeapDumpOnOutOfMemoryError

# ELÉMENTS DE LANGAGE

---

Plusieurs éléments de langages sont disponibles :

- **[fieldname1]** ou **fieldname1** : accéder à un champs
- **[fieldname1][fieldname2]** : accéder à un champs imbriqué
- **%{fieldname1}** : contenu du champs fieldname1
- **theValue** : accès à une variable d'environnement
- **if EXPRESSION {...} else if EXPRESSION {...} else {...}**
- **Comparaison** : ==, !=, <, >, <=, >=, =~, !~, and, or, xor, nand
- **Négation** : !
- **Inclusion** : in, not in
- **@metadata** : table de hashage contenant les métadonnées du document. Utile pour les données temporaires.
- **Commentaire** : #

```
input {
    # si le contact est défini
    if [donnees][contact] {
        mutate {
            add_field => {
                "contact" =>
                "%{[donnees][contact]}"
            }
        }
    }
}
```

# LIRE LES DONNÉE

---

- Pour éviter de télécharger systématiquement un fichier opendata, on préfèrera traiter un fichier local, mis à jour quand nécessaire. On utilisera le plugin d'input file avec les options principales suivantes :
  - path : chemin d'accès au fichier
  - sinceDb : chemin d'accès à la base de données d'historique
  - codec : interprétation des données
  - start\_position : point de prise en compte du fichier, généralement depuis le début
- Attention, les chemins d'accès doivent être en format Unix (/ au lieu de \)

```
input {  
    file {  
        path => "C:/elastic/ bureaux-de-votes.csv"  
        sinceDb_path => "C:/elastic/ bureaux-de-votes.sdb"  
        codec => plain{"charset" => "UTF-8"}  
        start_position => "beginning"  
    }  
}
```

# ÉCRIRE LES DONNÉES

---

- Pour valider les données avant de les écrire, on pourra utiliser le plugin d'output stdout : `stdout { codec => rubydebug{metadata => true } }`
- Une fois validées, on pourra écrire dans Elasticsearch avec le plugin elasticsearch prévu à cet effet. Les options essentielles sont :
  - hosts : urls du cluster
  - index : nom de l'index cible
  - document\_type : type du document
  - document\_id : id du document à créer
- Les documents seront chargés en utilisant la configuration par défaut détectée par le cluster.

```
output {  
    # stdout { codec => rubydebug{metadata => true } }  
    elasticsearch {  
        hosts => ["http://localhost:9200"]  
        index => "bureaux-de-votes"  
        document_type => "bureaux-de-votes"  
        document_id => "%{[@metadata][_id]}"  
    }  
}
```

# ÉCRIRE LES DONNÉES

- Pour améliorer le typage et l'analyse des données dans Elasticsearch, on peut appliquer un template aux indices. Ce template a une structure similaire à la définition d'un indice :

- order : ordre de priorité du template
- template : pattern de reconnaissance de l'indice
- settings : paramètres à appliquer à l'indice
- mappings : mapping à appliquer aux données
- aliases : alias à définir sur l'indice

- Ce template sera pris en compte en ajoutant les paramètres suivant au plugin output elasticsearch :

- manage\_template => true
- template => "chemin\_du\_fichier\_tmpl"
- template\_name => "nom\_du\_template"
- template\_overwrite => true

```
output {  
    # stdout { codec => rubydebug{metadata =>  
true } }  
    elasticsearch {  
        hosts => ["http://localhost:9200"]  
        index => "bureaux-de-votes"  
        document_type => "bureaux-de-votes"  
        document_id => "%{[@metadata][_id]}"  
        manage_template => true  
        template =>  
        "C:/elastic/bureaux_de_votes.tmpl"  
        template_name => "bureaux-de-votes"  
        template_overwrite => true  
    }  
}
```

# TRANSFORMER LES DONNÉES

---

Les données opendata récupérées nécessitent très souvent d'être retravaillées avec des plugins filter.

Les plugins filter principaux pour réaliser ces opérations sont les suivants :

- csv : interprétation d'un fichier csv
- mutate : réaliser les principales opérations de transformation d'une donnée
- date : type une chaîne de caractères en date
- fingerprint : réaliser un hash des données
- elasticsearch : requêter Elasticsearch pour enrichir les données

```
filter {  
  csv { ... }  
  mutate { ... }  
  date { ... }  
  fingerprint { ... }  
  elasticsearch { ... }  
  ...  
}
```

# MUTER VOS DONNÉES

Le plugin mutate est certainement le plus utile dans le traitement des données.

Il permet de réaliser les actions courantes sur les données :

- **split** : découper une chaîne suivant un séparateur
- **join** : joindre des champs avec un séparateur
- **replace** : remplacer la valeur d'un champs
- **gsub** : remplacement dans un champs à partir d'une regex
- **convert** : convertir un champs dans un type cible
- **add\_field** : ajouter un champs
- **remove\_field** : enlever un champs
- Les 3 derniers sont disponibles sur un grand nombre de plugins.
- Il est cependant plus lisible de les incorporer à mutate lorsque
- c'est possible.

```
filter {  
  mutate {  
    split =>{ "coordonnees"=> ", " }  
    gsub => [ "message",  
              "\r\n",  
              "" ]  
    add_field =>  
    { "[location][lat]"=>"%{[coordonnees][0]}"}  
    add_field =>  
    { "[location][lon]"=>"%{[coordonnees][1]}"}  
    remove_field => [ "coordonnees" ]  
  }  
}
```

# EXTRAIRE DES DONNÉES D'UN MESSAGE

---

- Le plugin grok est très utile dans le traitement des données de log.
- Regex lisible facilement, Pattern prédefini pour les patterns les plus commun et Definisable dans des properties
  - USERNAME [a-zA-Z0-9.\_-]+
  - USER %{USERNAME}
  - INT(?:[+-]?(?:[0-9]+))
  - WORD \b\w+\b
  - NOTSPACE \S+
  - DATA .\*?
  - GREEDYDATA .\*
  - HTTPDATE %{MONTHDAY}/%{MONTH}/%{YEAR}:%{TIME} %{INT}
  - COMBINEDAPACHELOG %{IPORHOST:clientip}

```
filter {
  grok {
    match => [ "message", "%{LOG_DATE:log_date} \[%{NOTSPACE:thread}\] %{LOGLEVEL:log_level} %{NOTSPACE:classname} - %{GREEDYDATA:msg}" ]
  }
}
```

4

---

**KIBANA**



# Kibana

## Discover



Données brutes

## Visualize



Tableaux  
Graphiques  
Informations

## Dashboard



Vue complexe

## Dev Tools



Client REST  
Auto-complétions

# ELASTIC | DISCOVER, VISUALIZE ET DASHBOARD

The screenshot shows the Kibana interface with three main sections:

- Discover:** Shows a histogram of logstash-\* events over time, with a count of 1,381 hits and a duration range from 14:29:22.715 to 16:44:22.715. It includes a sidebar with available fields like @version, \_id, \_index, \_score, \_type, \_idPath, \_scorePath, \_idPath, and host.
- Visualize:** Shows a histogram of logstash-\* events over time, with a count of 10 hits and a duration range from 14:29:22.715 to 16:44:22.715. It includes a sidebar with available fields like app, @Canal, @Correlation, @other, @message, @tag\_duration\_n, and tags.
- Dashboard:** Shows a donut chart titled "Catalogue d'productions" with the following distribution:
  - lifnet (49.49%)
  - risquent (12.79%)
  - av (11.78%)
  - courrielsupport (4.04%)
  - distribution (2.69%)

Identique à la version previous

- *Discover pour les logs*
- *Visualize pour les distributions*
- *Dashboard pour les applications*

The screenshot shows a Kibana dashboard titled "ATS - Exemples Kibana" with the following components:

- ATS - Markdown Intro:** A section with a title and some text.
- ATS & Framework:** A section with a title and some text.
- ATS - Metrics - Top applications errors:** A section with four large red numbers: 5 046, 225, 61, and 60, representing face-count, lifenet-count, personne-count, and distribution-count respectively.
- ATS - Domaines et applications ASSU/SKG/SGR:** A section with five pie charts showing the distribution of Application ASSU, Application SKG, Application SGR, Domaine ASSU, Domaine SKG, and Domaine SGR.
- ATS - Cloud - Applications:** A section with a title and some text.
- ATS - PieChart - LogLevel / Application:** A section with a title and some text.



- Identifier des anomalies par :

- Isolation des écarts par rapport à un historique

- Cas concrets d'utilisations

- Ecart de performance (temps de réponse)
- Ecart d'utilisation/visite (nombre d'appel)
- Ecart sur le nombre de dossiers traités, saisis...

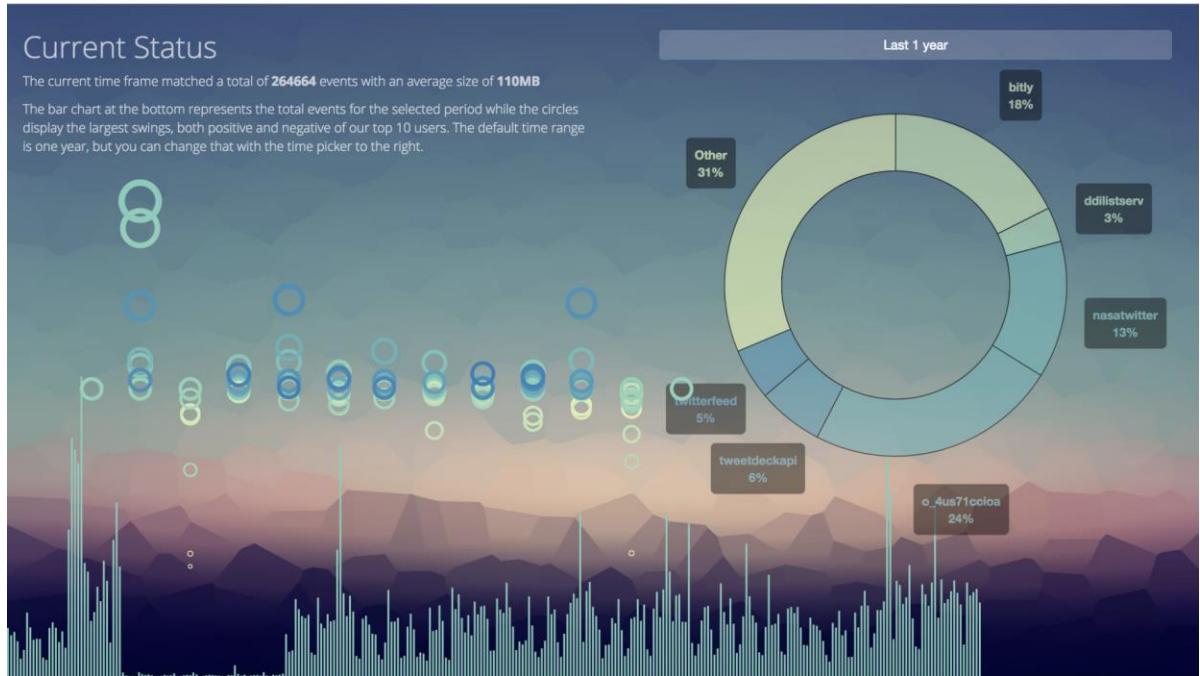


Identifier des anomalies par :  
Isolation des écarts par rapport à un historique

# ELASTIC | CANVAS



- Le mode canvas permet de réaliser des tableaux de bord de bord de type powerpoint.
- Utilisation des données de elastic

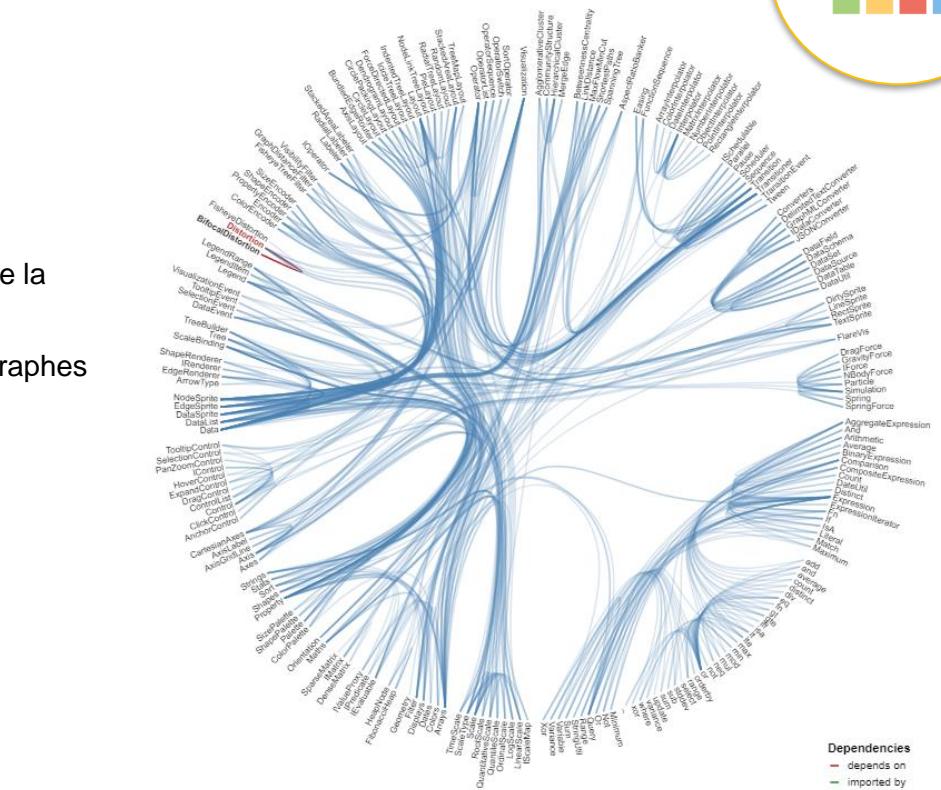


Le mode canvas permet de réaliser des tableaux de bord de type powerpoint.  
Utilisation des données de elastic

ELASTIC | VEGA



- Moteur de visualisation générique
  - Utilisation de la puissance de JSON pour la description de la visualisation
  - Utilisation de la puissance de D3js pour la création des graphes





## Création d'alertes

- Création de « watcher » basé sur les données
- Déclenchement automatique de notification
- Possibilité de chaînage des entrées

## Notification & intégration

- Email
- Elastic
- ....



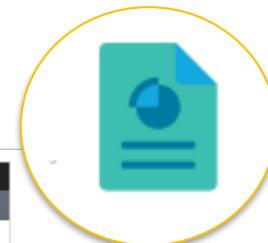
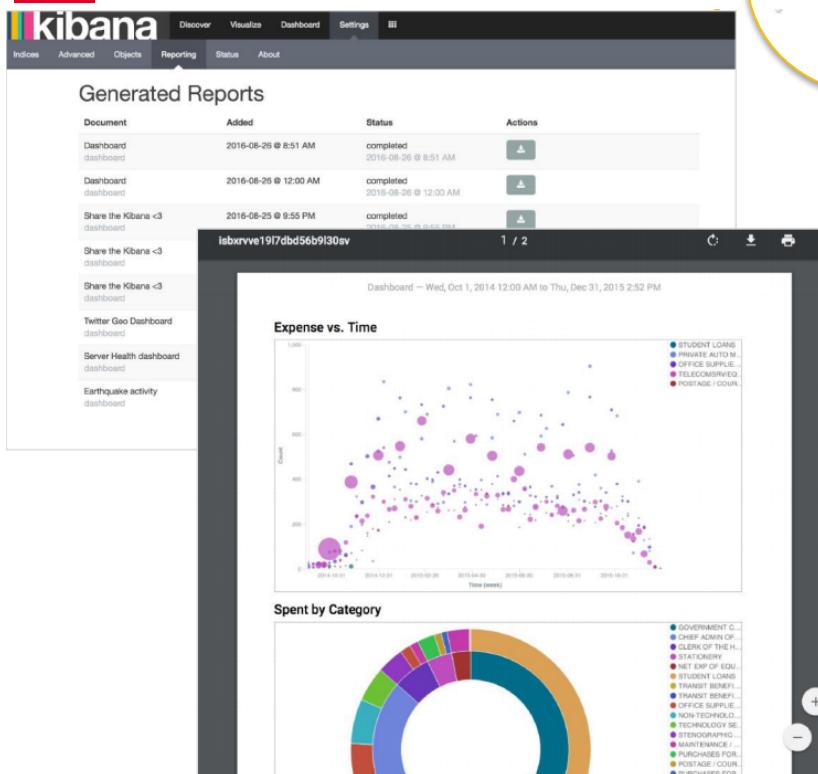
# ELASTIC | REPORTING

- Génération et partage de rapport

- Export de PDF issues des tableaux de bord et des visualisations

- Utilisation du module de alerting pour envoyer automatiquement les rapports :

- À intervalle régulier
- Sur la présence d'un événement



# ELASTIC | SPACE

---



A white circle containing a stylized 'K' shape composed of three colored segments: pink, black, and teal.

## Select your space

You can change your space at anytime.

**D**

**Default**

This is your default space!

**E**

**Engineering**

This is where the magic happens

**SO**

**Security Operations**

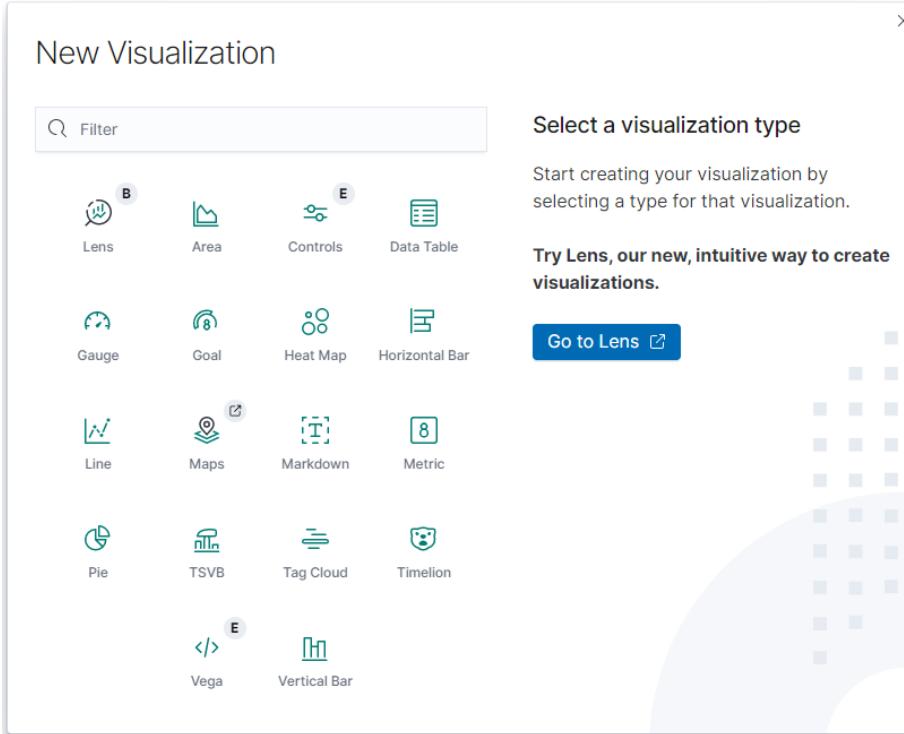
Keeping you safe

**M**

**Marketing**

Marketing campaigns, metrics, etc.

# LES GRAPHIQUES BASIQUES



TP