Ateliers
« ElasticStack / Kibana»

Pré-requis :

- Bonne Connexion Internet
- OS : Linux, MacOs, Windows 10
- YAML Editor : (VSCode, Atom, ...)
- Docker, Git

Mise en place
git clone + FileTransfer

Table des matières

Atelier 1 : API ElasticStack et Dev Console

1.1 Démarrage cluster et kibana

Ouvrir une ligne de commande dans le répertoire :

```
$TP_DATA/1_API_Cluster
```

Exécuter la commande qui initialise les utilisateurs systèmes de l'installation et démarrer un cluster 3 nœuds ElasticSearch ainsi que Kibana.

```
docker compose up -d
```

Accéder à ElasticSearch http(s)://localhost:9200

Utiliser les crédentiels de l'utilisateur elastic : ***elastic/secret***

Accéder ensuite à Kibana à http://localhost:5601

Utiliser l'utilisateur : ***elastic/secret***

Retrouver la *Dev Console*.

Exécuter les requêtes suivantes :

```
GET /_cluster/health
GET /_cat/nodes
GET /_cat/indexes
GET /_search
```

1.2 Concepts de cluster

Dans la DevConsole, créer un index nommé ***blogs***

```
PUT /blogs {
 "settings" : {
 "number_of_shards" : 3,
 "number_of_replicas" : 1
 }
}
```

Exécuter :

```
GET _cluster/health?pretty
```

Couleur du statut de santé ? Combien de shards disponibles, actifs ?

Arrêter le second nœud du cluster :

```
cd $TP_DATA/1_API_Cluster
docker stop es02
```

Re-executer

```
GET  _cluster/health?pretty
```

Couleur du statut de santé ? Combien de shards disponibles, actifs ?

Augmenter le nombre de répliques :

```
PUT /blogs/_settings
 { "number_of_replicas" : 2 }
```

Couleur du statut de santé ? Combien de shards disponibles, actifs ?

Arrêter le troisième nœud

```
cd $TP_DATA/1_API_Cluster
docker stop es03
```

Couleur du statut de santé ? Combien de shards disponibles, actifs ?

Atelier 2 : Alimentation des index et SearchLite

2.1 Document API

- Indexer les 3 documents fournis via la commande *curl* suivante ou via Kibana
  `curl -XPOST /blogs/_doc/ --data-binary "@entry1.json"`
- Noter les ids et récupérer les document via un ID
- Mettre à jour 1 document en ajoutant de nouveau champs (Remarquer l'incrémentation de la version) :
  - *tags* : Array
  - *views* : Initialisé à 0
- Supprimer l'index et effectuer toutes les opérations suivantes en une seule commande Bulk API

2.2 Pipelines d'ingestion

Arrêter votre cluster :

```
cd $TP_DATA/1_API_Cluster
docker compose down
```

Démarrer un autre cluster qui a désactivé la sécurité :

```
cd $TP_DATA/2.2_IngestionJava
docker compose up -d
```

Accèder à Kibana et définir une pipeline ayant comme seul processeur *attachment.*
Dans la Dev Console :

```
PUT /_ingest/pipeline/attachment
{
  "description" : "Extract attachment information",
  "processors" : [
    {  "attachment" : { "field" : "data" } },
    {  "remove" : { "field" : "data" } }
  ]
}
```

Vérifier la création via

```
GET /_ingest/pipeline
```

Utiliser le programme Java fourni (fichier jar dans répertoire *ingest*) pour indexer les documents bureautiques fournis

```
cd $TP_DATA/2.2_IngesrionJava/projects/ingest-8
java -jar ingest.8.x.jar ../../doc bureautique attachment
```

Vérifier le nombre de documents indexés
GET /bureautique/_count
GET /bureautique/_search

2.3 Pipelines logstash

Récupérer les archives ***apache2.zip*** et ***server.tar.gz*** et les dézipper dans :

```
$TP_DATA/2.3_Logstash/logstash/source_files
```

Arrêter votre cluster :

```
cd $TP_DATA/2.2_IngestionJava
docker compose down
```

Positionner vous dans le répertoire ***$TP_DATA/2.3_Logstash***

Démarrer un nouveau cluster qui intègre également logstsash avec 2 pipelines configurées

```
cd $TP_DATA/2.3_Logstash
docker compose up -d
```

Les fichiers de configuration de logstash sont visibles dans le répertoire ***logstash***.

Le fichier ***pipelines.yml*** définit 2 pipeline d'ingestion :

- ***jboss*** : Ingère les fichiers de trace d'une application Java (source_files/server)
- ***apache*** : Ingère les fichiers de traces des accès à un serveur Web (source_files/apache2)

Au démarrage de la stack, l'ingestion de données doit commencer.

Vous pouvez le vérifier par plusieurs moyens :

Accès aux logs de logstash :

```
docker logs -f logstash
```

Dans la Kibana Console, vous pouvez effectuer les commandes suivantes :

```
GET /cat/_indices # Pour voir les index créés
GET /jboss*/_count # Nombre docs des index commençant par jboss
GET /logstash*/_count # Nbre docs des index commençant par jboss
```

2.4 Search Lite et Lucene

Sur l'index bureautique, effectuer les recherches suivantes :

- Documents répondant à « Java »
- Documents ne répondant pas à « Java »
- Limiter les documents retournés de la première requête
- Documents dont le contenu répond à « Java »
- Documents PDF dont le contenu répond à « Java »
- Documents dont le contenu répond à « Elastic Search »
- Documents dont le champ titre contient administration
- Document créés après une date particulière
- Document créés après une date particulière et dont le contenu répondant « Java Elastic Search » mais pas « Administration »

Ateliers 3 : Mapping et analyzers

3.1 Mapping et analyseur

Visualiser le mapping de l'index contenant les documents bureautiques. Quels sont les champs de texte intégral et quels analyseurs sont utilisés ?

Tester les différences entre l'analyseur standard et l'analyseur français sur des textes français avec des requêtes REST. Quels sont les mots vides utilisés ? Comment se comportent les mots avec des accents, avec des apostrophes, des mots composés ?

```
GET /_analyze?analyzer=standard
La cérémonie des JO 2024 s’est déroulée le 26 Juillet 2024

GET /_analyze?analyzer=french
La cérémonie des JO 2024 s’est déroulée le 26 Juillet 2024

GET /_analyze?analyzer=english
La cérémonie des JO 2024 s’est déroulée le 26 Juillet 2024
```

```
GET /_analyze?analyzer=english
The 2024 Olympic Games ceremony took place on July 26, 2024
```

3.2 Gabarit d’index

Regarder les gabarits d’index existants :

```
GET /_template
```

Regarder le mapping d’un des index créés par les pipelines logstash précédentes

Dans les index ***logstash-apache****, nous avons un petit défaut qui nous empêchera d’effectuer des requêtes de géo-localisation : les champs ***client.geo.location*** n’est pas reconnus en tant que *geo_point* mais comme un objet embarquant 2 float.

Pour corriger ce problème nous créons un gabarit d’index comme suit :

```
PUT _template/logstash_apache_template
{
  "index_patterns": [
    "logstash-apache-*"
  ],
  "settings": {
    "number_of_shards": 2
  },
  "mappings": {
    "properties": {
      "client.geo.location": {
        "type": "geo_point"
      }
    }
```

```
  }
}
```

I

l faut ensuite relancer l’alimentation depuis le début

Arrêter logstash

```
cd $TP_DATA/2.3_Logstash/
docker compose down logstash
```

Il faut désormais supprimer les index créés :

Pour autoriser l’utilisation de wildcard dans les commandes de suppression d’index :

```
PUT /_cluster/settings
{
  "persistent": {
    "action.destructive_requires_name": false
  }
}
```

Supprimer ensuite les indexs préalablement créés

```
DELETE /jboss*
DELETE /logstash-apache-*
```

Relancer logstash

```
docker compose up logstash -d
```

Vérifier le mapping des index

Atelier 4: Recherche via DSL

4.1 Syntaxe DSL

Effectuez des requêtes DSL basées sur l'index ***bureautique*** :

• Documents PDF triés par date

• Documents dont le champ de contenu répond à "administration"

• Documents dont le champ contenu ou titre répond à "administration"

• Les documents dont le champ contenu ou titre répond à "administration" et dont la date de création se situe dans une fourchette

• Documents PDF dont le champ de contenu ou de titre répond à "administration" et dont la date de création se situe dans une fourchette

• Documents dont le champ de contenu répond à "Administration" ou "Oracle"

• Documents dont le champ de contenu répond à "Administration" et éventuellement "Oracle"

4.2 Recherches avancées

Effectuer les requêtes suivantes :

- Document dont le titre commence par la lettre a
- Récupérer les documents contenant l'expression "cadre java", permettre une distance de 5 mots
- Récupérer les documents dont le titre commence par "administration j"
- Effectuez des recherches floues sur le champ auteur avec des fautes de frappe

4.3 Agrégations

Sur l'index Apache

- Effectuer des agrégations de comptage
  - par type de code retour
  - par verbe
  - par les 2

- Effectuer des regroupements par intervalles de taille
- Trouver la taille moyenne d'une requête
- Trouver la taille moyenne des requêtes groupées par code retour
- Trouver la somme des octets transférés par trimestre
- Trouver le client ayant un taux anormal de code réponse 400

4.4 Géo-localisation

Sur l'index Apache

- Effectuer des agrégation de type *geo_hash_grid* sur les localisation des clients en augmentant progressivement la précision

Ateliers 5: Kibana et tableaux de bord

5.1 Démarrer avec Kibana

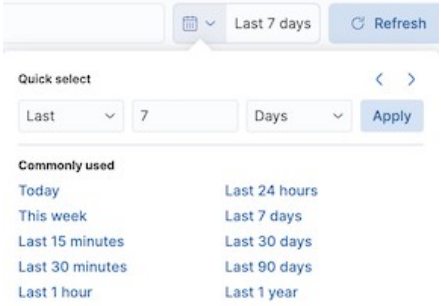
Ajouter les données

1. Sur la page d'accueil, cliquez sur ***Try sample Data***
2. Cliquez sur ***Other sample data sets***.
3. Choisir ***Sample eCommerce orders***

Explorer les données

Aller sur ***Discover***

Sélectionner la fenêtre temporel au 7 derniers jours :



Filtrer les données afin de ne voir : « *les commandes de vêtements pour femmes d'une valeur de 60 $ ou plus* »

Dans le champ de recherche saisir :

```
products.taxless_price >= 60 and category : Women's Clothing
```

Ajouter le champ ***category***

Visualiser les données

Aller dans ***Dashboard*** et visualiser le tableau de bord : [eCommerce] Revenue Dashboard.

Dans la barre d'outils, cliquez sur ***Edit***.

Sur le tableau de bord, cliquez sur ***Create Visualization***

Dans l'éditeur de visualisation ouvrez la liste déroulante ***Visualization Type*** puis sélectionnez ***Treemap***.

Dans la liste des champs disponibles, faites glisser les champs suivants vers l'espace de travail :

- ***geoip.city_name***
- ***manufacturer.keyword***

Sauvegarder

Interactivité

Utiliser en suite le panel [eCommerce] Controls pour sélectionner un fabricant.

Filtre

1. Cliquer sur ***Add Filter***
2. Dans la liste déroulante ***Field***, sélectionnez ***day_of_week***
3. Dans la liste déroulante Opérateur, sélectionnez ***is***.
4. Dans la liste déroulante Valeur, sélectionnez ***Wednesday***.

5.2 Fonctionnalités de Discover

Toujours sur la dataview ***eCommerce***

Ajouter le champ ***manufacturer*** dans le tableau de données, visualiser les 10 valeurs les plus courantes

Trouver les champs ***customer_first_name*** et ***customer_last_name*** et les ajouter

Ajouter un champ customer et indiquer le script Painless suivant :

```
String str = doc['customer_first_name.keyword'].value;
char ch1 = str.charAt(0);
emit(doc['customer_last_name.keyword'].value + ", " + ch1);
```

Supprimer ***customer_first_name*** et ***customer_last_name***

Rechercher des documents avec la requête KQL suivante :
geoip.country_iso_code : US and products.taxless_price >= 75

Ajouter un filtre afin d'exclure les documents dont le champ ***day_of_week*** n'est pas égal à ***Wednesday***

Accéder au détail d'un document

Visualiser les boutons associés à un champ

Essayer les 2 liens actions

5.3 La syntaxe KQL

Commencer par créer une dataview ***logstash-apache*** sur les index ***logstash-apache****, la positionner comme dataview par défaut.

Visualiser les informations de Mapping dans Kibana

Accéder ensuite à la page Discover

- Quel est le pays le plus présent ?
- Faire une recherche sur le mot google dans quel champ est présent ce terme
- Rechercher dans le champ url.original le mot clé microsoft
- Recherche dans le champ message la valeur exacte /formation/microsoft/
- Faire une recherche isolant les requêtes POST PUT DELETE
- Toutes les requêtes ont elle un champ client.ip ?

Effectuer une requête KQL qui isole les requêtes en erreur qui ne provienne pas de machine ayant une IP commençant par 192

Sauvegarder la requêtes sous le noms « Requêtes en erreur »

5.4 La syntaxe ES|QL

Écrire une requête ES|QL qui affiche le total d'octets transférés par pays, limiter au 10 premier pays les plus importants

Editer la visualisation pour la convertir en camembert

Ajouter le camembert à un nouveau tableau de bord

A l'intérieur du tableau de bord, revenir sur la requête pour afficher 20 pays

Ateliers 6. Tableaux de bord et visualisation

6.1 Création du premier tableau de bord

6.1.1 Création du tableau de bord

Sur l'index représentant les logs d'accès Apache.

- Ouvrez le menu principal, puis cliquez sur ***Dashboard***
- Cliquez sur ***Create a Dashboard***.
- Définissez le filtre temporel sur l'année 2015.

6.1.2 Visualisations

Nous voulons visualiser le nombre de visiteurs uniques en utilisant client.ip.

Utilisez ainsi la visualisation ***Metrics*** pour afficher le champ sous forme de nombre.

La seule fonction numérique que vous pouvez utiliser avec ***client.ip*** est la cardinalité, qui se rapproche du nombre de valeurs uniques.

- Ouvrez la liste déroulante ***Visualization type***, puis sélectionnez ***Metrics***.
- Dans la liste des champs disponibles, faites glisser ***client.ip*** vers l'espace de travail.
- Cliquer ensuite sur ***Unique count of client.ip***. Et saisir le texte « Visiteurs unique »

Sauvegarder

Nous voulons visualiser l'évolution du champ byte en fonction du temps.

Sur le tableau de bord, cliquez sur ***Create a Visualization***

Dans la liste Champs disponibles, faites glisser ***http.response.body.bytes*** vers l'espace de travail Par défaut, Kibana affiche un Histogramme temporel affichant la valeur médiane.

- Changer le type de visualisation en ***Line***

Pour augmenter l'intervalle de temps minimum,

- Cliquez sur ***timestamp***.
- Modifiez l'intervalle minimum à ***1d***, puis cliquez sur ***Fermer***.

Pour gagner de la place sur le tableau de bord, masquez les libellés des axes.

- Ouvrez le menu de l'axe de gauche, puis sélectionnez ***None*** dans la liste déroulante du titre de l'axe.
- Faites de même pour l'axe du bas.

Save and Return

Ajouter le titre « Médiane des octets transférés »

Créez ensuite une visualisation qui affiche les valeurs les plus fréquentes de ***url.original.keyword*** sur votre site Web, classées par visiteurs uniques.

Pour créer la visualisation, utilisez les valeurs principales de ***url.original.keyword*** classées par nombre unique de clients, au lieu d'être classées par nombre d'enregistrements.

1. Dans le tableau de bord, cliquez sur ***Create a Visualization***
2. Dans la liste Champs disponibles, faites glisser ***client.ip*** vers le champ Axe vertical
3. Faites glisser ***url.original.keyword*** vers l'espace de travail.

Les libellés du graphique ne peuvent pas s'afficher car le champ ***url.original.keyword*** contient des champs de texte longs.

La meilleure façon d'afficher les champs de texte longs est d'utiliser la visualisation Tableau.

1. Ouvrez la liste déroulante Type de visualisation, puis sélectionnez Tableau.

Cliquez sur les 5 premières valeurs de ***url.original.keyword***.

1. Dans le champ Nombre de valeurs, saisissez 10.
2. Dans le champ Nom, saisissez l'URL de la page.
3. Cliquez sur Fermer.

Save and Return

Pas besoin de titre

Créez une visualisation qui aide à déterminer si les utilisateurs transfèrent plus d'octets à partir de documents de moins de 10 Ko par rapport à des documents de plus de 10 Ko.

1. Dans le tableau de bord, cliquez sur ***Create a Visualization***
2. Dans la liste Champs disponibles, faites glisser le champ ***http.response.body.bytes*** vers le champ Axe vertical dans le volet des couches.
3. Cliquez sur Médiane des bytes.
4. Cliquez sur la fonction rapide ***Somme***, puis sur Fermer.
5. Dans la liste Champs disponibles, faites glisser les ***http.response.body.bytes*** vers le champ ***Break down by***

Spécifiez les plages de taille de fichier :

- Dans l'onglet layer, cliquez sur ***http.response.body.bytes***.
- Cliquez sur *Create custom ranges*, saisissez les informations suivantes dans le champ Range, puis appuyez sur Retour :

  Plages : 0 → 10 240

  Libellé : Moins de 10 Ko

Cliquez sur Ajouter une plage, saisissez les informations suivantes, puis appuyez sur Retour :

Plages : 10 240 → +∞

Libellé : Plus de 10 Ko

Dans la liste déroulante ***Value format***, sélectionnez Bytes (1024), puis cliquez sur Fermer.

Choisir le type de visualisation Pie

Save and return

Éditer le titre : *« Somme des octets provenant de requêtes volumineuses »*

Analysez le trafic du site Web par heure pour trouver le meilleur moment pour la maintenance

1. Dans le tableau de bord, cliquez sur Créer une visualisation.
2. Dans la liste Champs disponibles, faites glisser ***http.response.body.bytes*** vers le champ Axe vertical dans l'onglet layer
3. Cliquez sur Médiane des octets.
4. Cliquez sur la fonction rapide Somme.
5. Dans le champ Nom, saisissez *Octets transférés.*
6. Dans la liste déroulante Value Format, sélectionnez Octets (1024), puis cliquez sur Fermer.
7. Dans la liste Champs disponibles, faites glisser ***hour.keyword*** vers le champ Axe horizontal.
8. Cliquez sur hour.keyword, puis faites glisser le curseur de granularité Intervalles jusqu'à ce que l'axe horizontal affiche des intervalles horaires.

Save and Return

Editer le titre : *« Traffic Web »*

Les visualisations Table et Proportions prennent en charge plusieurs fonctions. Par exemple, pour créer des visualisations qui décomposent les données par sources de trafic du site Web et par géographie de l'utilisateur, appliquez les fonctions Filter et Top Values.

1. Dans le tableau de bord, cliquez sur Créer une visualisation.
2. Ouvrez la liste déroulante Type de visualisation, puis sélectionnez ***Treemap***.
3. Dans la liste Champs disponibles, faites glisser ***Records*** vers le champ ***Size by***
4. Dans l'onglet Layer, cliquez sur ***Add or drag-and-drop a field*** pour ***Group by***

Créez un filtre pour chaque source de trafic du site Web :

1. Cliquez sur Filtres.
2. Cliquez sur Tous les enregistrements, saisissez les informations suivantes dans la barre de requête, puis appuyez sur Retour :

KQL — http.request.referrer : *bing*

Libellé — Bing

3. Cliquez sur Ajouter un filtre, saisissez les informations suivantes dans la barre de requête, puis appuyez sur Retour :

   KQL — referer : *google*

   Libellé — Google

4. Cliquez sur Ajouter un filtre, saisissez les informations suivantes dans la barre de requête, puis appuyez sur Retour :

   KQL — NOT referer : *bing* OR NOT referer : *google*

   Libellé — Autre

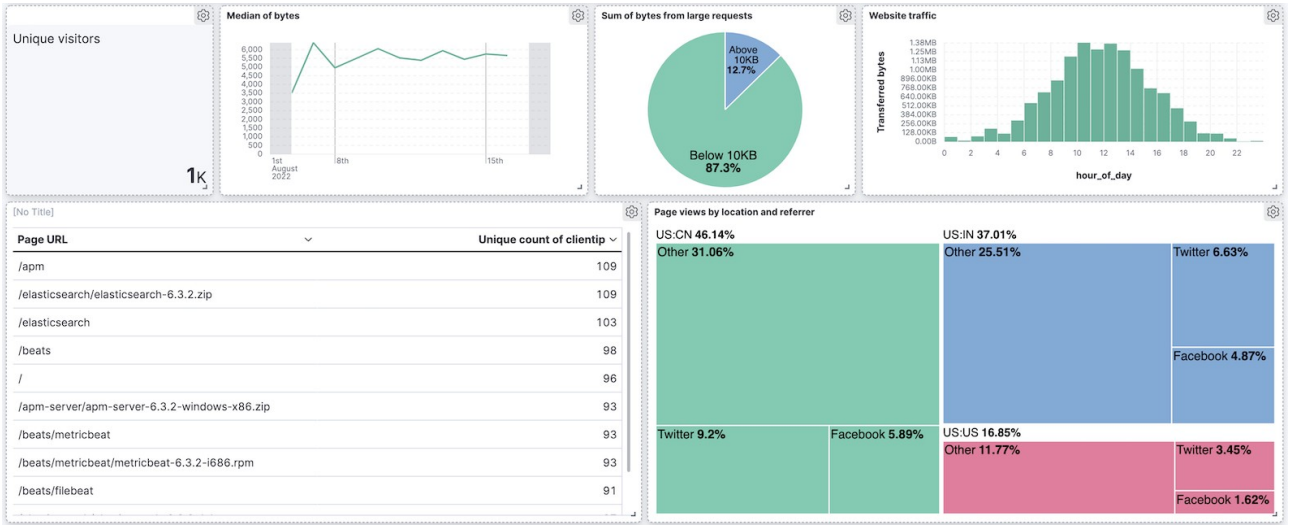
Ajoutez le groupement géographique des utilisateurs :

1. Dans la liste Champs disponibles, faites glisser ***client.geo.continent_code*** vers l'espace de travail.
2. Pour modifier l'ordre de regroupement, faites glisser les 3 premières valeurs de ***client.geo.continent_code*** dans l'onglet layer pour qu'elles apparaissent en premier.

Save and Return

Editer le titre : « Pages vues par emplacement et référent »

Réarranger les panels pour obtenir le résultat suivant :



Sauvegarder le tableau de bord

6.2 Lens

Le but de cet telier est de construire un nouveau tableau de bord sur les accès Apache.

Ce tableau de bord affichera plusieurs visualisations :

- Les KPI
  - Total de hits

    - Nombre d'Ips différentes
    - Taille moyenne d'une requête
- Le profil type des connexions sur une journée montrant le volume de requêtes heure par heure en distinguant les code retour des requêtes (2xx, 4xx, 5xx…) et en filtrant les requêtes 3xx
- Un camembert montrant la répartition par pays des requêtes en erreur

Les visualisations seront effectués après avoir filtrer les ip clients correspondant à des adresses internes

6.3 TSVB

Moyenne requête bytes

Max requête bytes

Math pour percentage

Ajout d'un autre métrique

6.4 Vega

https://www.elastic.co/guide/en/kibana/current/vega.html

6.5 Agrégations

Ouvrir l'éditeur d'agrégations et choisir un intervalle de temps de 7 jours

Créer un graphique barre

Ajouter un Buckets Agregations à Date Histogramme

Ajouter un sous-agrégations de type ***terms*** en sélectionnant le champ ville

Ouvrir la visualisation dans Lens

6.6 Map

Ouvrir l'éditeur Map et choisir un intervalle de 7 jours

La première couche est une couche choroplèthe pour colorer les pays du monde en fonction du trafic. Les nuances plus foncées symboliseront les pays avec plus de trafic, et les nuances plus claires symboliseront les pays avec moins de trafic.

Cliquez sur ***Add Layer***, puis ***Choropleth***.

Dans le menu déroulant ***EMS boundaries***, sélectionnez World countries.

Dans la source des statistiques, définissez :

- Affichage des données sur logstash-apache
- Joindre le champ sur client.geo.country.iso_code
- Cliquez sur Ajouter et continuer.

Dans les paramètres de la couche, définissez :

- Nom : Nombre total de requêtes par destination
- Opacité : 50 %

Ajouter un champ d'info-bulle :

- Le code ISO 3166-1 alpha-2 est ajouté par défaut.
- Cliquez sur + Ajouter pour ouvrir la sélection de champ.
- Sélectionnez ***name*** et cliquez sur Ajouter.

Dans le style de couche :

- Définissez ***Fill color*** > ***As number*** sur la palette de couleurs grises.
- Définissez ***Border color*** à blanc.
- Sous ***Label***, modifiez par ***Fixed value***

Cliquez sur ***Keep modifications***.

Le résultat devrait ressembler à ceci

La deuxième couche affichera des données agrégées lorsque les utilisateurs effectueront un zoom arrière sur la carte.

Cliquez sur ***Add layer***, puis sur ***Documents***.

Définissez la vue Données sur *logstash-apache*.

Cliquez sur ***Add and continue***.

Dans les paramètres de calque, définissez :

- Nom : Requêtes
- Visibilité : la plage [9, 24]
- Opacité à 100 %

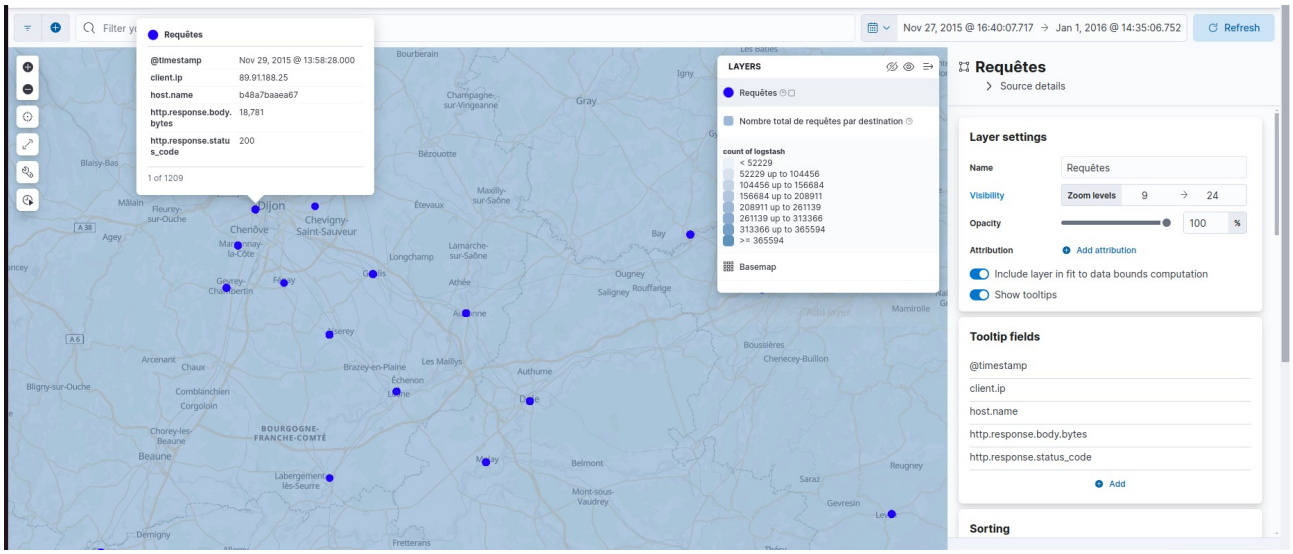
Ajoutez un champ d'info-bulle et sélectionnez ***user_agent.original***, ***http.response.body.bytes***, ***client.ip***, ***host.name***, ***url.original***, ***http.response.status_code*** et ***@timestamp***.

Dans ***Scaling***, activez ***Limit results*** à 10 000.

Dans ***Layer style***, définissez ***Fill color*** sur ***#2200FF.***

Cliquez sur ***Keep changes***

Avec un zoom de niveau 9, votre carte peut ressembler à ceci :



Enfin, on créée une couche visible uniquement lorsque la carte est dézoomée.
Les couleurs plus foncées symboliseront les grilles avec plus de trafic, et les couleurs plus claires symboliseront les grilles avec moins de trafic.
Les cercles plus grands symboliseront les grilles avec plus d'octets transférés au total, et les cercles plus petits symboliseront les grilles avec moins d'octets transférés.

Cliquez sur ***Add layer*** et sélectionnez ***Clusters***.

- Définissez la vue Données sur ***logstah-apache***
- Cliquez sur ***Add and continue***.

Dans ***Layer settings***, définissez :

- Nom : Nombre total de requêtes et d'octets
- Visibilité sur la plage [0, 9]
- Opacité sur 100 %

Dans ***Metrics*** :

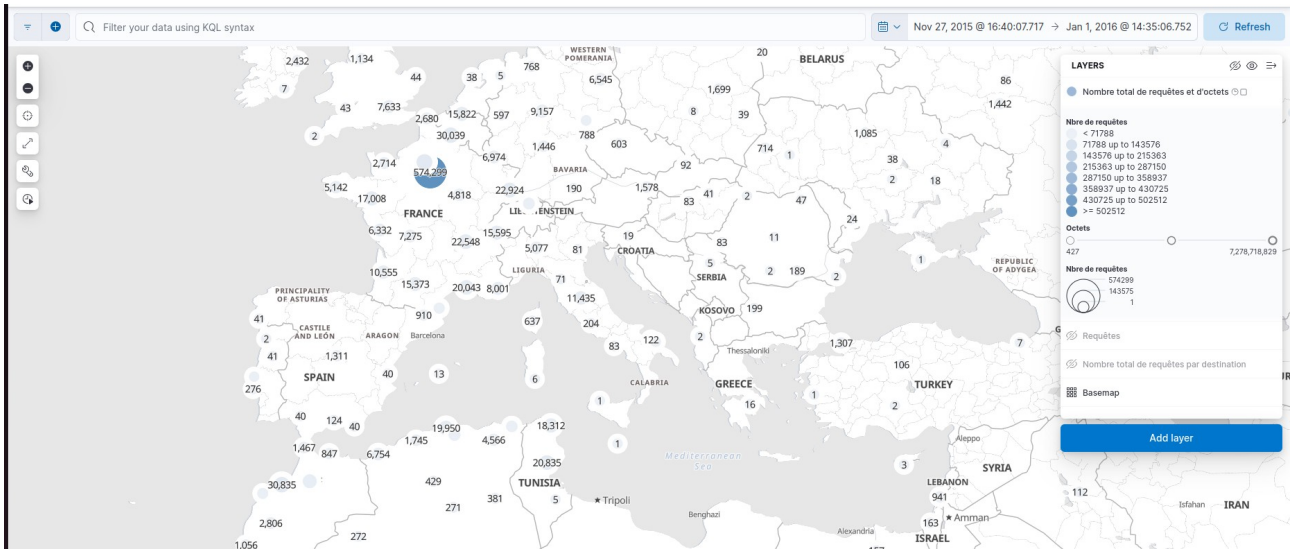
- Définissez ***Aggregation*** sur ***Count***.
- Cliquez sur ***Add Metric***
- Définissez ***Aggregation*** à ***Sum*** avec le champ ***http.response.body.bytes***

Dans ***Layer style***, modifiez ***Symbol size*** :

- Définissez ***By Value*** sur somme d'octets.
- Définissez la taille minimale sur 7 et la taille maximale sur 25 px.

Cliquez sur ***Keep changes***

A la fin de l'atelier votre carte doit ressembler à ceci :



Sauvegarder et ajouter dans le tableau de bord

6.7 Timelion

Tutoriel Elastic
https://www.elastic.co/guide/en/kibana/current/timelion.html

Le résultat final consiste à afficher 3 graphiques :

- L'usage CPU en temps réel comparée à celle de la dernière heure

- Le traffic réseau en entrée et en sortie
- La consommation mémoire avec l’affichage d’alerte si elle dépasse un certain seuil

Atelier Interactions

Drilldown dashboard

https://www.elastic.co/guide/en/kibana/current/create-drilldowns.html

Canvas

https://www.elastic.co/guide/en/kibana/current/canvas-tutorial.html