TD : Lancement d'Elasticsearch et Kibana avec Docker

Objectif : déployer rapidement un environnement Elasticsearch + Kibana local à l'aide de Docker.

Partie 1 : Démarrage

1. **Décompressez** le fichier `elastic1.zip`.
   Vous obtenez un dossier nommé `elastic`.
2. **Ouvrez le projet dans VS Code :**
   - Lancez **Visual Studio Code**
   - Menu **Fichier** → **Ouvrir un dossier**
   - Sélectionnez le dossier `elastic` précédemment décompressé.
3. **Ouvrez un terminal** dans VS Code.
   Exécutez successivement les commandes suivantes :

   ```
   docker compose up setup
   ```

   Attendez que le processus se termine normalement, puis lancez :

   ```
   docker compose up -d
   ```

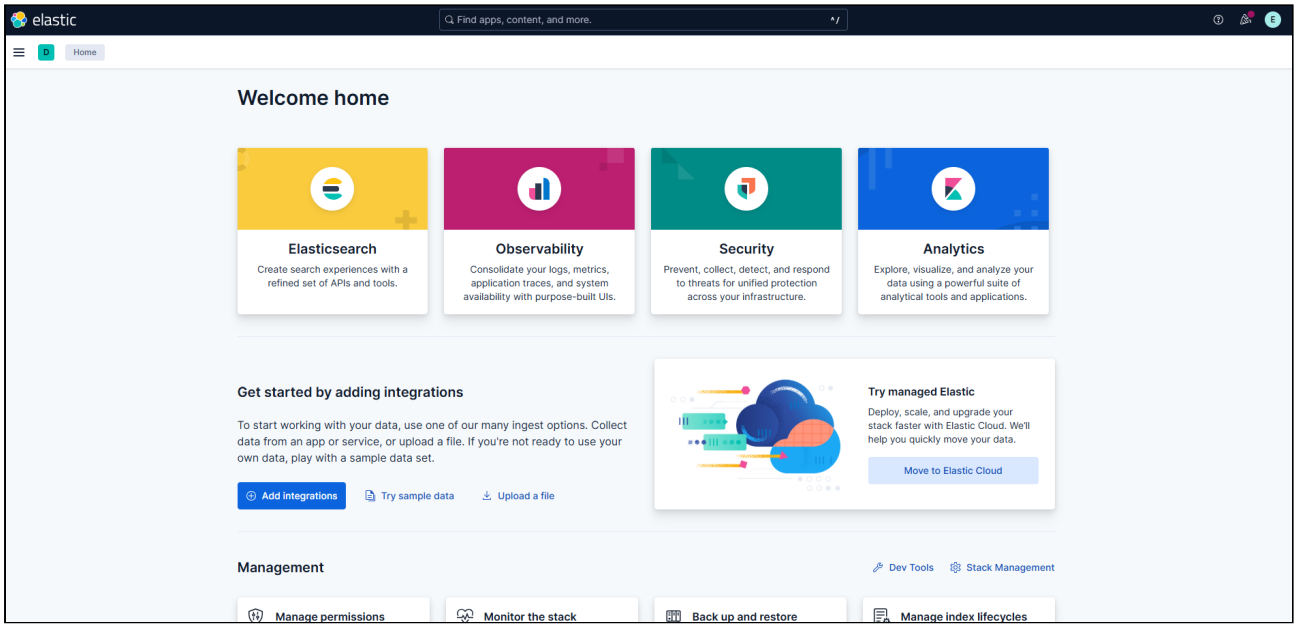
4. **Patientez 2 à 5 minutes** le temps que les services démarrent.
5. **Accédez à Kibana** depuis votre navigateur :
   http://localhost:5601

   Identifiants :
   - **Login :** `elastic`
   - **Mot de passe :** `changeme`

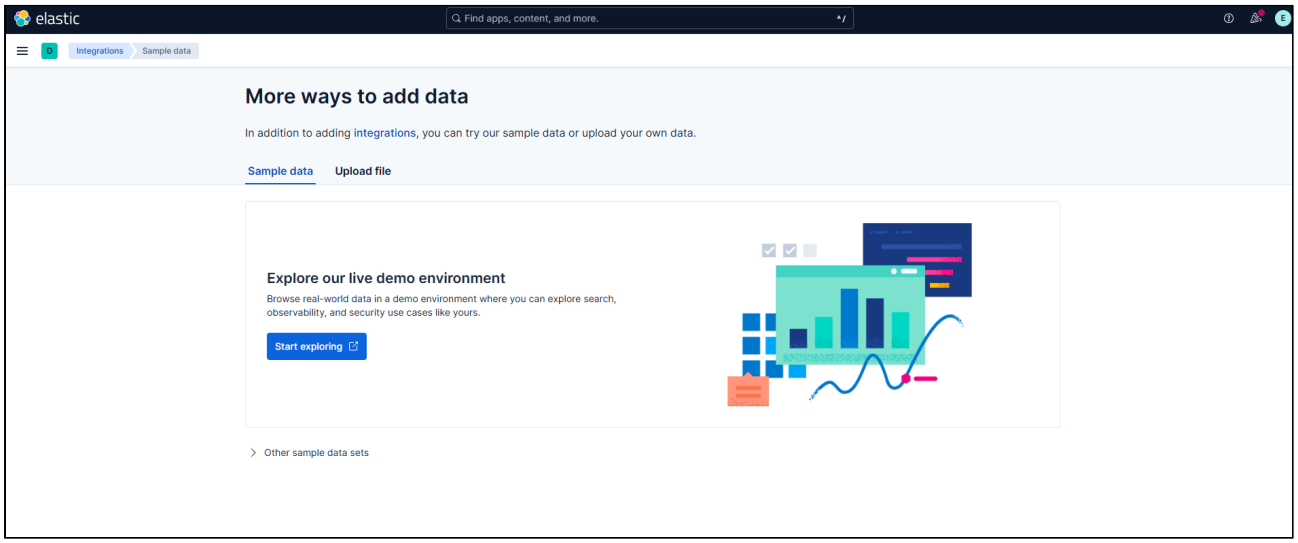
Charger les données

Sur l’écran d’accueil de Kibana :

Cliquez sur “Try sample data”.

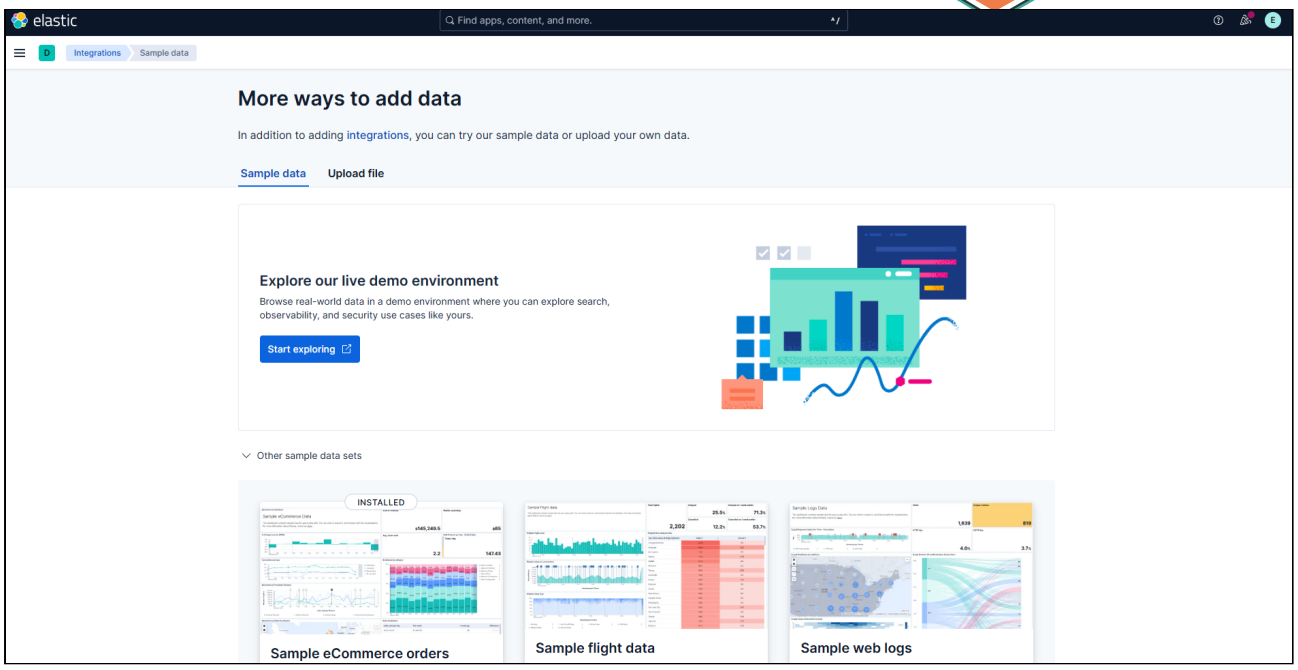


Faites défiler jusqu’à la section “Other sample data sets”.



Chargez les trois jeux de données suivants :

- **Sample eCommerce orders** (données de commandes e-commerce)
- **Sample flight data** (données de vols)
- **Sample web logs** (logs web)



1.2 : Découverte de l'interface Discover

1. Dans le menu principal, cliquez sur **Discover**
2. Sélectionnez la vue de données `kibana_sample_data_ecommerce`
3. Observez l'interface :
   - La barre de requête en haut
   - Le sélecteur de période (Time picker)
   - L'histogramme de distribution temporelle
   - La liste des documents

Ouvrez la page : http://localhost:5601/app/discover

Partie 2 : Premières requêtes ES|QL

Exercice 2.1 : Syntaxe de base

ES|QL utilise une syntaxe en pipeline avec l'opérateur `|`.

Requête 1 : Afficher les 10 premiers documents

```
FROM kibana_sample_data_ecommerce
| LIMIT 10
```

À faire :

- Copiez cette requête dans la barre de recherche de Discover

- Changez le mode de requête en **ES|QL** (en haut à droite de la barre de recherche)
- Exécutez la requête

Question : Quels champs voyez-vous dans les résultats ?

Exercice 2.2 : Sélection de colonnes

Requête 2 : Afficher uniquement certains champs

```
FROM kibana_sample_data_ecommerce
| KEEP customer_first_name, customer_last_name, email, total_quantity,
taxful_total_price
| LIMIT 20
```

À faire :

- Exécutez cette requête
- Observez que seules les colonnes spécifiées sont affichées

Question : Quelle est la différence entre `KEEP` et `DROP` ?

Exercice : Réécrivez la requête en utilisant `DROP` pour exclure tous les champs sauf ceux listés ci-dessus.

Exercice 2.3 : Filtrage avec WHERE

Requête 3 : Filtrer les commandes de plus de 100€

```
FROM kibana_sample_data_ecommerce
| WHERE taxful_total_price > 100
| KEEP customer_first_name, customer_last_name, taxful_total_price, order_date
| SORT taxful_total_price DESC
| LIMIT 10
```

À faire :

- Exécutez cette requête
- Analysez les résultats

Exercices supplémentaires :

1. Trouvez toutes les commandes de moins de 50€
2. Trouvez les commandes entre 75€ et 125€ (utilisez `AND`)
3. Trouvez les commandes du client "Eddie" (utilisez `customer_first_name`)

Exercice 2.4 : Tri des résultats

Requête 4 : Trier par plusieurs colonnes

```
FROM kibana_sample_data_ecommerce
| KEEP customer_last_name, customer_first_name, taxful_total_price
| SORT customer_last_name ASC, taxful_total_price DESC
| LIMIT 20
```

À faire :

- Trouvez les 5 commandes les plus récentes
- Trouvez les 10 clients qui ont commandé le plus d'articles (`total_quantity`)

Partie 3 : Manipulation de données

Exercice 3.1 : Création de nouveaux champs avec EVAL

Requête 5 : Calculer une remise

```
FROM kibana_sample_data_ecommerce
| EVAL discount_amount = taxful_total_price * 0.1
| EVAL price_after_discount = taxful_total_price - discount_amount
| KEEP customer_first_name, taxful_total_price, discount_amount, price_after_discount
| LIMIT 10
```

Exercices :

1. Calculez la TVA (20%) sur chaque commande
2. Calculez le prix moyen par article (`taxful_total_price / total_quantity`)
3. Créez un champ `full_name` en concaténant prénom et nom (utilisez `CONCAT`)

Aide pour la concaténation :

```
| EVAL full_name = CONCAT(customer_first_name, " ", customer_last_name)
```

Exercice 3.2 : Fonctions de texte

Requête 6 : Manipuler les chaînes de caractères

```
FROM kibana_sample_data_ecommerce
| EVAL name_upper = TO_UPPER(customer_first_name)
| EVAL name_lower = TO_LOWER(customer_last_name)
```

```
| EVAL email_length = LENGTH(email)
| KEEP customer_first_name, name_upper, customer_last_name, name_lower, email,
email_length
| LIMIT 10
```

Exercices :

1. Affichez les 3 premiers caractères du prénom (utilisez `SUBSTRING`)
2. Trouvez tous les clients dont le nom de famille contient "son"

Exercice 3.3 : Fonctions de date

Requête 7 : Extraire des composants de date

```
FROM kibana_sample_data_ecommerce
| EVAL day_of_week = DATE_EXTRACT(day_of_week, order_date)
| STATS count = COUNT(*) BY day_of_week
| SORT day_of_week ASC
```

Exercices :

1. Trouvez toutes les commandes du mois de décembre
2. Trouvez toutes les commandes passées un lundi (day_of_week = 1)
3. Calculez l'âge en jours de chaque commande depuis aujourd'hui

---

Partie 4 : Agrégations et statistiques

Exercice 4.1 : Agrégations simples avec STATS

Requête 8 : Statistiques globales

```
FROM kibana_sample_data_ecommerce
| STATS
    total_orders = COUNT(*),
    total_revenue = SUM(taxful_total_price),
    avg_order_value = AVG(taxful_total_price),
    max_order = MAX(taxful_total_price),
    min_order = MIN(taxful_total_price)
```

Questions :

- Quel est le montant total des ventes ?

- Quelle est la valeur moyenne d'une commande ?

Exercices :

1. Calculez le nombre total d'articles vendus (`total_quantity`)
2. Trouvez la médiane du prix des commandes (utilisez `PERCENTILE`)

Exercice 4.2 : Groupement avec BY

Requête 9 : Statistiques par catégorie

```
FROM kibana_sample_data_ecommerce
| STATS
    order_count = COUNT(*),
    total_sales = SUM(taxful_total_price),
    avg_sale = AVG(taxful_total_price)
  BY customer_gender
| SORT total_sales DESC
```

Exercices :

1. Calculez les ventes totales par jour de la semaine
2. Trouvez le top 10 des clients qui ont dépensé le plus (grouper par `customer_id`)
3. Calculez le nombre de produits différents commandés par catégorie

Requête bonus : Top 5 des clients avec leurs informations

```
FROM kibana_sample_data_ecommerce
| STATS
    total_spent = SUM(taxful_total_price),
    order_count = COUNT(*),
    avg_order = AVG(taxful_total_price)
  BY customer_id, customer_first_name, customer_last_name
| SORT total_spent DESC
| LIMIT 5
```