Elastic
ElasticSearch

Nirah-Technology

Présentation

- **Définition** : Moteur de recherche et d'analyse distribué basé sur Apache Lucene.
- **But principal** : Indexer et rechercher de grandes quantités de données de manière rapide et flexible.
- **Nature** : NoSQL, conçu pour une recherche rapide sur des données textuelles, structurées et semi-structurées

Historique

- **Création** : Lancé en 2010 par Shay Banon.
- **Base** : Basé sur Apache Lucene, un moteur de recherche puissant mais complexe.
- **Évolution** : De simple moteur de recherche à une plateforme complète d'analyse de données (logs, métriques, données géospatiales, etc.).

Comparatif

Caractéristique	Elasticsearch	Solr	MongoDB
Basé sur	Apache Lucene	Apache Lucene	BSON (NoSQL)
Distribué	Oui	Oui	Non
Scalabilité	Très haute	Haute	Moyenne
Indexation	JSON	XML, JSON	BSON
Utilisation principale	Recherche texte	Recherche texte et analyse	Stockage JSON

Cas d'utilisation

- **Logs et monitoring** : Analyse de logs serveur (ex. : ELK stack : Elasticsearch, Logstash, Kibana).
- **Recherche dans des applications web** : Recherche rapide dans des contenus web ou bases de données.
- **Analyse de données en temps réel** : Mesure et analyse de données d’événements ou de transactions

Architecture

- **Nœud** : Instance d'Elasticsearch, il peut avoir différentes fonctions (master, data, client)
- **Cluster** : Groupe de nœuds Elasticsearch qui travaillent ensemble.
- **Indice** : Conteneur logique pour les documents.
- **Document** : Unité de base dans Elasticsearch (équivalent à une ligne dans une base de données).
- **Shards et Réplicas** : Division des indices pour la scalabilité et la résilience.

Nœuds, Clusters et Indices

- **Nœuds** :
  - **Master Node** : Gère l'état du cluster.
  - **Data Node** : Stocke les données et exécute les requêtes.
  - **Client Node** : Gère les requêtes client.
- **Cluster** : Ensemble de nœuds connectés.
- **Indice** : Collection de documents, subdivisée en shards.

Shards et Réplicas

- **Shards** : Partition des données pour la scalabilité horizontale.
- **Réplicas** : Copies de shards pour assurer la disponibilité et la tolérance aux pannes.
- **Routage** : Détermination des shards sur lesquels les données sont stockées

TP - Installation & Configuration

Concepts d'Indexation et Mapping

- **Indexation** : Processus d'ajout de documents dans un indice.
- **Mapping** : Définition des types de champs et de la manière dont les documents sont indexés.
- **Types de champs** : Text, Keyword, Integer, Date, etc.
- **Analyseurs** : Processus de transformation des données avant indexation (ex. : suppression des stop words, découpage des mots).

Structure des Documents JSON

Un document est une unité de données indexée.

Exemple :


```
{
        "user": "John Doe",
        "message": "Elasticsearch is awesome!",
        "post_date": "2025-03-10"
}
```

Les Requêtes de Recherche

- **Match** : Recherche sur un champ textuel avec analyse.
- **Term** : Recherche exacte sur un champ non analysé.
- **Range** : Recherche sur une plage de valeurs (ex. : dates, chiffres).

- curl -X GET "localhost:9200/mon_index/_search
- curl -X PUT "localhost:9200/mon_index/_search ...

Concepts clés - Mapping et Types de données

Le **mapping** définit la **structure des documents**.

```
curl -X PUT "localhost:9200/mon_index/_mapping" -H "Content-Type: application/json"
-d '{
  "properties": {
    "nom": {"type": "text"},
    "age": {"type": "integer"}
  }
}'
```

Concepts clés - Analyse et Tokenisation

Elasticsearch découpe les mots pour optimiser la recherche.

```
curl -X GET "localhost:9200/_analyze" -H "Content-Type: application/json" -d '{
  "analyzer": "standard",
  "text": "Bonjour tout le monde"
}'
```

Concepts clés - Aggrégation

Les **agrégations** permettent de **calculer des statistiques**.

- Agrégation **GroupBy :** Regrouper les résultats (par exemple, par auteur).
- Agrégation **Avg :** Calcul de la moyenne d'un champ numérique.

```
{
  "aggs": {
    "moyenne_age": {
      "avg": { "field": "age" }
    }
  }
}
```

Optimisation et Bonnes-Pratiques

Optimisation des requêtes :

- Utiliser des filtres plutôt que des recherches full-text
- Privilégier les champs keyword pour des valeurs exactes
- Limiter le nombre de résultats (size dans les requêtes)

Gestion des indices :

- Utiliser l’alias d’index pour faciliter les mises à jour
- Archiver les vieux indices pour libérer de l’espace
- Recycler les shards pour éviter la fragmentation