

Système de Surveillance des Données de Pression Artérielle patient avec Kafka, Elasticsearch et Kibana :

Introduction

Ce projet vise à concevoir un système de surveillance des données de pression artérielle basé sur une architecture distribuée utilisant **Kafka**, **Elasticsearch**, et **Kibana**. notre système gère les lectures de pression artérielle de manière autonome, en classant les valeurs en fonction de seuils médicaux prédéfinis et en les traitant différemment selon leur catégorie (normale ou anormale).

Objectif principal

Le système vise à répondre à plusieurs besoins critiques dans le domaine de la surveillance médicale. Il doit être capable de générer et transmettre des lectures de pression artérielle en temps réel, d'identifier et classer ces lectures en fonction de seuils médicaux prédéfinis, et de sauvegarder les données normales localement tout en indexant les données anormales dans **Elasticsearch** pour une visualisation avancée via **Kibana**.

Objectifs spécifiques

Pour atteindre cet objectif principal, plusieurs objectifs spécifiques ont été définis. Tout d'abord, il est nécessaire d'implémenter un **producteur Kafka** capable de générer des données de pression artérielle conformes au standard FHIR (Fast Healthcare Interoperability Resources). Ensuite, un **consommateur Kafka** doit être développé pour analyser les lectures et les classer selon les seuils médicaux. Les données normales seront sauvegardées localement, tandis que les données anormales seront indexées dans **Elasticsearch**. Enfin, **Kibana** sera utilisé pour créer des tableaux de bord interactifs permettant de visualiser les anomalies de pression artérielle.

Architecture Technique

Le système suit une architecture basée sur des microservices comprenant les composants suivants :

1. Producer Service

- Génère des lectures synthétiques de pression artérielle en FHIR (JSON).
- Publie ces messages dans un topic Kafka.

2. Consumer Service

- Consomme les messages du topic Kafka.
- Classifie les lectures selon les seuils médicaux prédéfinis.
- Sauvegarde les lectures normales localement.
- Indexe les lectures anormales dans Elasticsearch.

3. Kafka

- Sert de plateforme de streaming pour transmettre les données entre le producteur et le consommateur.

4. Elasticsearch

- Stocke et indexe les lectures anormales pour permettre des requêtes avancées.

5. Kibana

- Fournit une interface de visualisation pour analyser les anomalies de pression artérielle.

Fonctionnement

1. Génération des Données :

- Le Producer génère des messages FHIR contenant des données de pression artérielle (systolique et diastolique) pour différents patients.
- Ces messages sont publiés sur le topic Kafka blood_pressure_topic.

2. Transmission des Données :

- Kafka transmet les messages du Producer au Consumer en temps réel.

3. Analyse des Données :

- Le Consumer analyse les données pour détecter les anomalies en fonction des seuils médicaux prédéfinis.
- Les données normales sont sauvegardées localement dans des fichiers JSON.
- Les données anormales sont indexées dans Elasticsearch.

4. Visualisation des Données :

- Kibana est utilisé pour visualiser les anomalies de pression artérielle via des tableaux de bord interactifs.

Configuration de Kafka :

Kafka est configuré pour gérer les échanges de données entre le Producer et le Consumer. Voici les configurations clés :

- **Producer Configuration :**

- **bootstrap_servers** : "kafka:9092" (adresse du broker Kafka).
- **value_serializer** : Les messages sont sérialisés en JSON avant d'être envoyés.
- **Topic** : "blood_pressure_topic" (le topic sur lequel les messages sont publiés).

- **Consumer Configuration :**

- **bootstrap_servers** : "kafka:9092".

- **value_deserializer** : Les messages sont désérialisés en JSON après réception.
- **auto_offset_reset** : "earliest" (le Consumer commence à lire depuis le début du topic).
- **enable_auto_commit** : True (le Consumer valide automatiquement les messages lus).

Modèle d'index Elasticsearch :

Les anomalies de pression artérielle sont stockées dans un index Elasticsearch nommé `blood_pressure_anomalies`. Chaque document contient les champs suivants :

- **patient_id** : Identifiant unique du patient.
- **timestamp** : Date et heure de la mesure.
- **systolic_pressure** : Valeur de la pression artérielle systolique.
- **diastolic_pressure** : Valeur de la pression artérielle diastolique.
- **category** : Catégorie de l'anomalie (ex. : hypertension, hypotension).
- **location** : Coordonnées géographiques de la mesure (latitude, longitude).
-

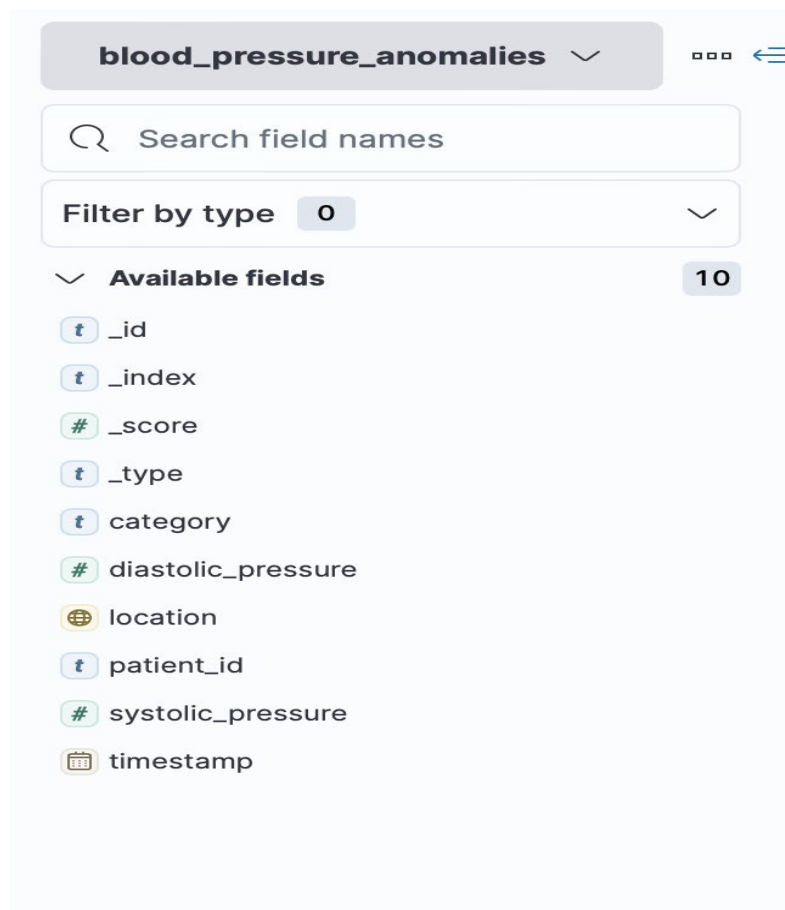
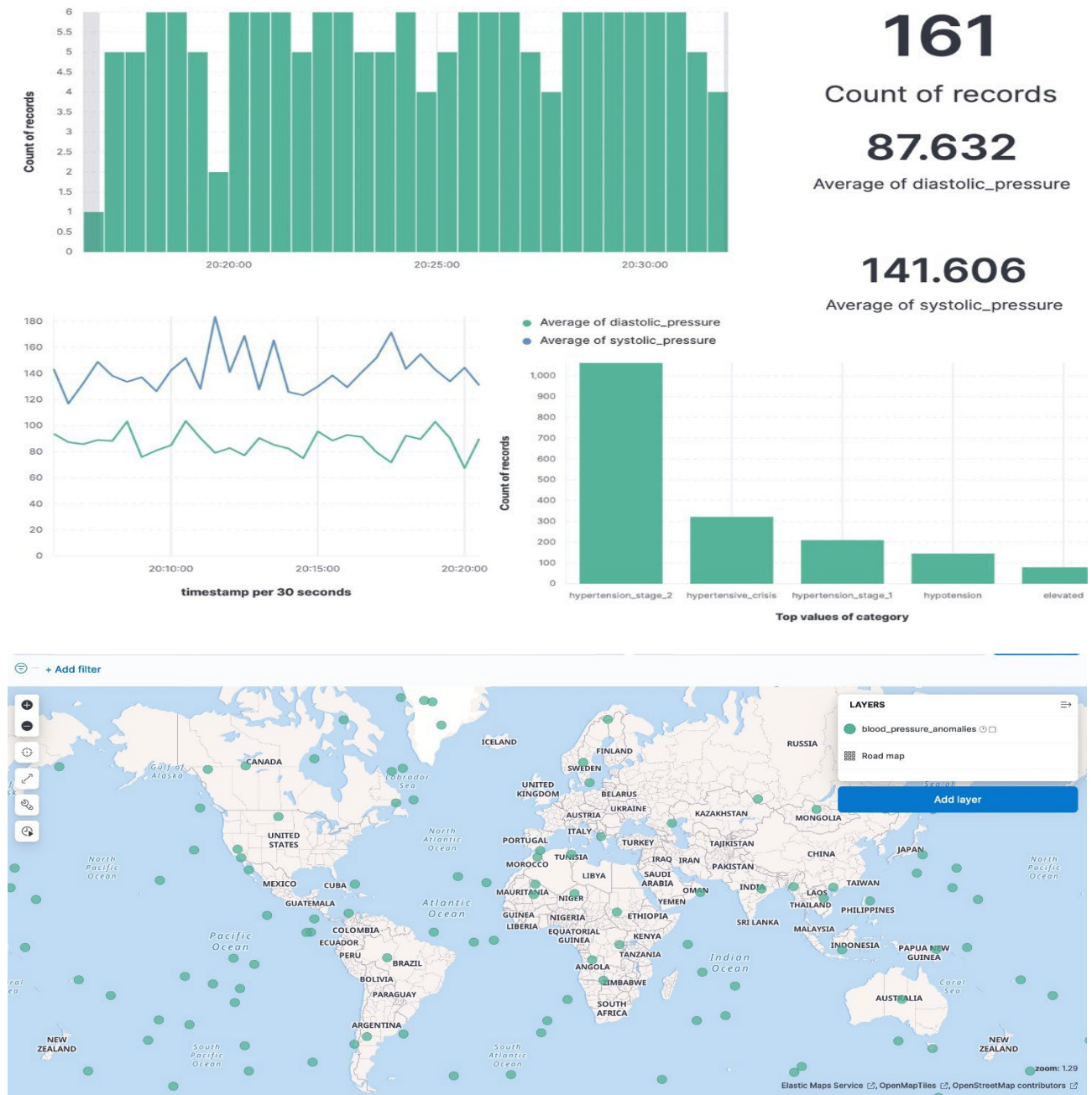


Tableau de bord Kibana

Le tableau de bord Kibana a été conçu pour visualiser les anomalies de pression artérielle de manière interactive. Pour les 161 premiers enregistrements, les résultats montrent une répartition claire des anomalies selon les catégories prédéfinies. Les graphiques et les visualisations permettent de comprendre rapidement les tendances et les patterns dans les données, ce qui est essentiel pour une surveillance médicale efficace.



Conclusion :

Ce système de surveillance des données de pression artérielle démontre une intégration efficace de Kafka, Elasticsearch et Kibana dans un contexte distribué. Il offre une solution robuste pour analyser et visualiser les données médicales en temps réel, avec des perspectives d'évolution pour des cas d'usage pratiques ou industriels .