BENAICHA Khaoula
AIT KARROUM Khawla
TANANI Kaoutar
4A ICY SI

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES HAUTS-DE-FRANCE

Compte Rendu TP Kafka
Sécurisation SASL
Détails du code

Afin d'implémenter la sécurisation SASL dans le projet Kafka fourni, il est important de modifier et d'ajouter certaines configurations dans des fichiers bien précis.

Les différentes modifications sont portées sur les fichiers suivants:

- Dossier GenKP :
  - **« genkp.properties »**
- Dossier KC-Étudiants :
  - **« kc-etudiants.properties »**
- Broker :
  - **« server.properties »**
  - **« kafka-server- start.bat »**
- Zookeeper :
  - **« Zookeeper.properties »**
  - **« Zookeeper-server- start.bat »**

De plus, deux autres fichiers sont ajoutés dans le dossier config :
- **« Kafka_server_jaas.conf »**
- **« Zookeeper_server_jaas.conf »**

Examinons de plus près les différents changements effectués :

➢ **« genkp.properties » :**

Genkp représente le producer, ainsi, afin d'activer la sécurité SASL et spécifier le mécanisme PLAIN, on a ajouté les propriétés suivantes :

```
spring.kafka.properties.security.protocol=SASL_PLAINTE
XT
spring.kafka.properties.sasl.mechanism=PLAIN;
```

Et on a également configuré les informations d'authentification, comme suit :

```
spring.kafka.properties.sasl.jaas.config=org.apache.ka
fka.common.security.plain.PlainLoginModule required \
username="admin" \ password="admin-sec";
```

De plus, on a activé une propriété spécifique à spring Kafka dans le but de prendre en charge la configuration JASS spécifiée et l'utiliser pour authentifier et autoriser les clients Kafka :

```
spring.kafka.jaas.enable = true
```

➢ **« kc-etudiants.properties » :**

KC-Étudiant représente le consumer, ainsi, afin de s'authentifier mutuellement, on l'a configuré exactement de la même manière que le **genkp** en ajoutant les mêmes propriétés.

➢ **« server.properties » :**

On a décommenté la ligne suivante afin de spécifier le protocole de sécurité SASL :

```
listeners=SASL_PLAINTEXT://:9092
```

On a également spécifier l'adresse et le protocole auxquels les clients doivent se connecter dans le but de communiquer avec le broker Kafka d'une façon sécurisée :

```
advertised.listeners=SASL_PLAINTEXT://localhost:9092
```

De plus, afin d'activer le protocole de sécurité SASL et le mécanisme d'authentification :

```
security.inter.broker.protocol=SASL_PLAINTEXT
sasl.mechanism.inter.broker.protocol=PLAIN
sasl.enabled.mechanisms=PLAIN
```

➢ **« kafka-server- start.bat » :**

Dans ce fichier, on a ajouté la ligne suivante :

```
set
KAFKA_OPTS=-Djava.security.auth.login.config=C:/kafka/
config/kafka_server_jaas.conf
```

Cette ligne a pour but de configurer les options de sécurité lorsqu'on démarre le serveur Kafka tout en indiquant le fichier de configuration JAAS utilisé pour l'authentification JASS sécurisée.

➢ **« Zookeeper.properties » :**

On a ajouté :

```
authProvider.1=org.apache.zookeeper.server.auth.SASLA
uthenticationProvider
```

Cette ligne représente le fournisseur d'authentification SASL à utiliser.

```
jaasRealm=ZooKeeper
jaasServerName=zookeeper
```

- **JaasRealm** va définir le royaume JAAS utilisé par Zookeeper pour l'authentification SASL.
- **JaasServerName** spécifie le nom du serveur jaas utilisé par le Zookeeper.

```
requireClientAuthScheme=sasl
```

Indique le schéma d'authentification requis pour les clients qui se connectent à ZooKeeper. Dans notre cas, il indique que l'authentification SASL est requise.

- **« Zookeeper-server- start.bat »**

```
set
ZOOKEEPER_OPTS=-Djava.security.auth.login.config=C:/ka
fka/config/Zookeeper_server_jaas.conf
```

Cette ligne a pour but de configurer les options de sécurité lorsqu'on démarre le serveur Kafka tout en indiquant le fichier de configuration JAAS utilisé pour l'authentification JASS sécurisée.

Ces deux fichiers ajoutés sont des fichiers de configuration JAAS qui ont pour but de vérifier les informations d'identification fourni par le client lors de sa connexion:

- **« Kafka_server_jaas.conf » :**
  Ce fichier définit un module de connexion ***PlainLoginModule*** pour l'authentification du serveur Kafka. Ce module utilise une authentification qui se base sur les informations d'authentification. Dans notre cas, un utilisateur nommé **"myuser"** est configuré avec le mot de passe **"azerty"**.
  Aussi l'utilisateur **"user_admin"** est configuré avec le même mot de passe **"azerty"**.
  Autrement dit, un client qui veut se connecter à Kafka doit fournir les informations valides correspondants à l'un de ces utilisateurs.

```
KafkaServer {

org.apache.kafka.common.security.plain.PlainLoginModu
le required
   username="myuser"
   password="azerty"
```

```
   user_admin="azerty";
};
```

➢ **« Zookeeper_server_jaas.conf » :**

Ce fichier définit un module de connexion ***DigestLoginModule*** pour l'authentification du serveur Zookeeper. Ce module utilise une authentification qui se base sur les informations d'authentification en utilisant un mécanisme de hachage digest. Dans notre cas, un utilisateur nommé **"user-admin"** est configuré avec le mot de passe **"azerty"**.

Autrement dit, un client qui veut se connecter à Kafka doit fournir les informations valides correspondants à cet utilisateur.

```
Server{
     org.apache.zookeeper.server.auth.DigestLoginModu
le required
   user_admin="azerty";
};
```