

# KYLIN-5442 Ticket auto-renewal is not supported when kerberos is enabled for the real-time feature, resulting in failure of the build job

## Design

This article only introduces Kafka Kerberos use keytab authentication scenarios.

The official Kafka doc: <https://kafka.apache.org/documentation/#consumerconfigs>

### 3. Configuring Kafka Clients

To configure SASL authentication on the clients:

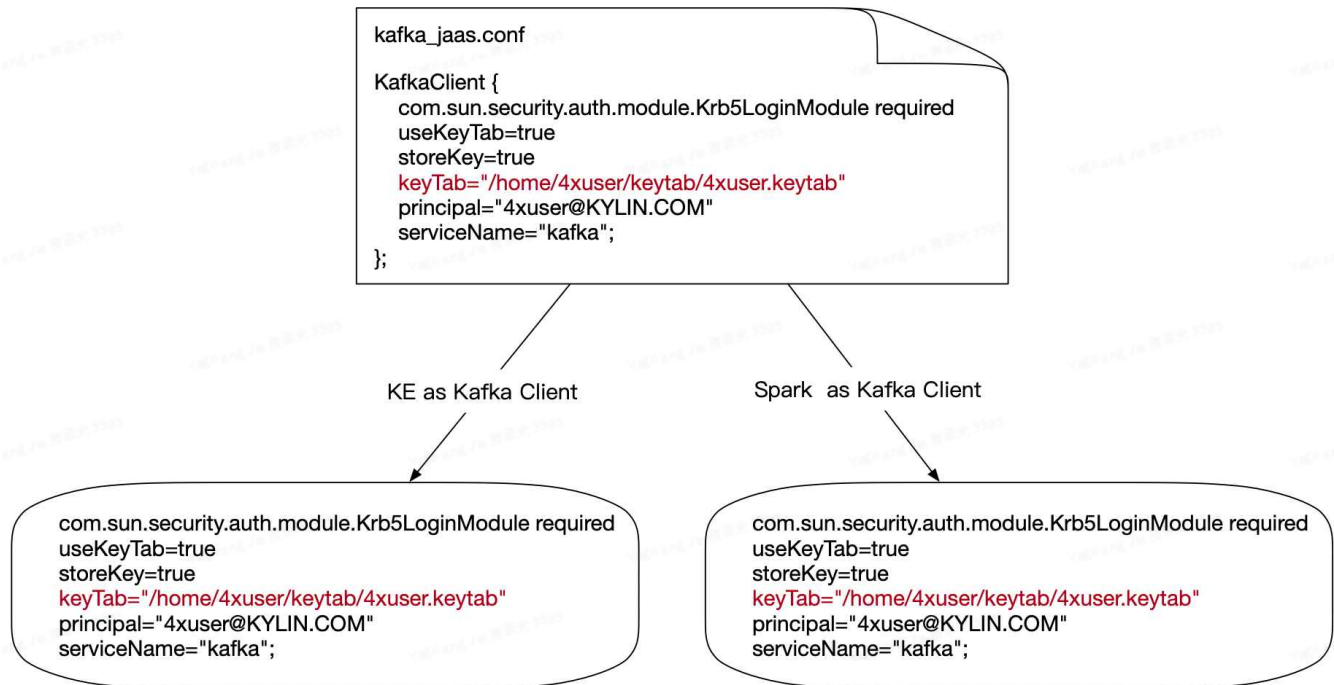
1. Clients (producers, consumers, connect workers, etc) will authenticate to the cluster with their own principal (usually with the same name as the user running the client), so obtain or create these principals as needed. Then configure the JAAS configuration property for each client. Different clients within a JVM may run as different users by specifying different principals. The property `sasl.jaas.config` in `producer.properties` or `consumer.properties` describes how clients like producer and consumer can connect to the Kafka Broker. The following is an example configuration for a client using a keytab (recommended for long-running processes):

```
1 sasl.jaas.config=com.sun.security.auth.module.Krb5L
2 useKeyTab=true \
3 storeKey=true \
4 keyTab="/etc/security/keytabs/kafka_client.keyt
5 principal="kafka-client-1@EXAMPLE.COM";
```

You can see that the officially recommended Long-Running application uses keytab to log in Kerberos.

## Original Design

The first version of Kafka adaptation Kerberos done before, and the part Design about Kafka Kerberos certification is like this.



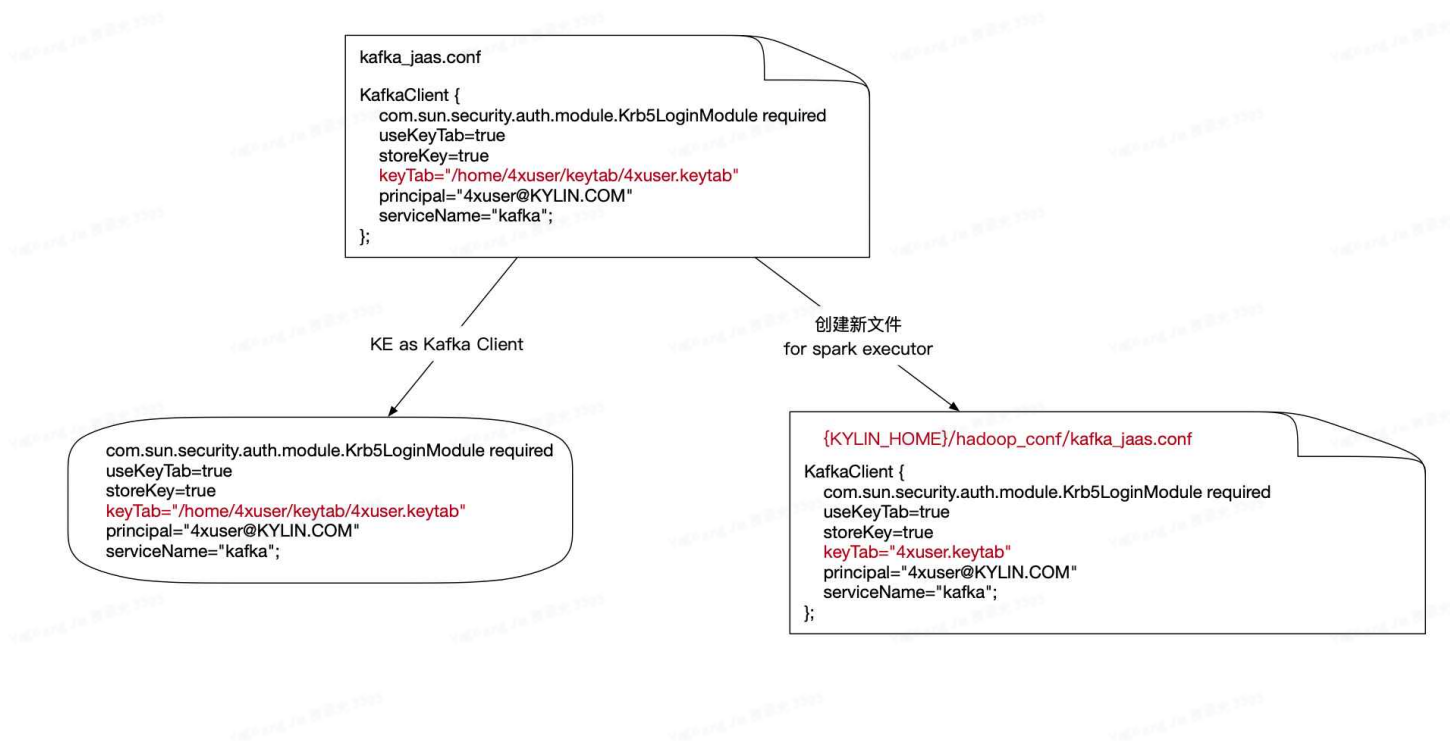
会有如下问题

There will be the following questions

- Require all node machines on Yarn to put the **keyTab** file in the same directory, otherwise Spark Executor will fail to find the **keyTab** file when connecting to Kafka.

This is also the point that this ISSUE aims to address.

## Current Design



When starting a live build task

1. Spark Driver uses local `{KYLIN_HOME}/conf/kafka_jaas.conf` files.
2. Create `{KYLIN_HOME}/hadoop_conf/kafka_jaas.conf`
  - a. The keyTab in its content is a **relative path**
  - b. Upload this **keytab** to the spark temporary directory for the executor to read
  - c. `{KYLIN_HOME}/hadoop_conf/kafka_jaas.conf`, will be uploaded to the spark temporary directory for the executor to read
  - d. Overwrite every time a live build task starts `{KYLIN_HOME}/hadoop_conf/`
3. When Spark Executor connects to Kafka, the keytab file used comes from the temporary directory

At this point, Spark tasks do not require all machines to have a keytab file with the same path.

After the task starts, observe the env configuration of spark, you can see the following two keywords

```
1 spark.driver.extraJavaOptions=-Djava.security.auth.login.config=
  {KYLIN_HOME}/conf/kafka_jaas.conf
2
3
4 spark.executor.extraJavaOptions=-
  Djava.security.auth.login.config=./__spark_conf__/_hadoop_conf__/_kafka_jaas.co
```

## Design

本篇只介绍Kafka Kerberos 使用keytab认证的场景。

Kafka官方文档：<https://kafka.apache.org/documentation/#consumerconfigs>

### 3. Configuring Kafka Clients

To configure SASL authentication on the clients:

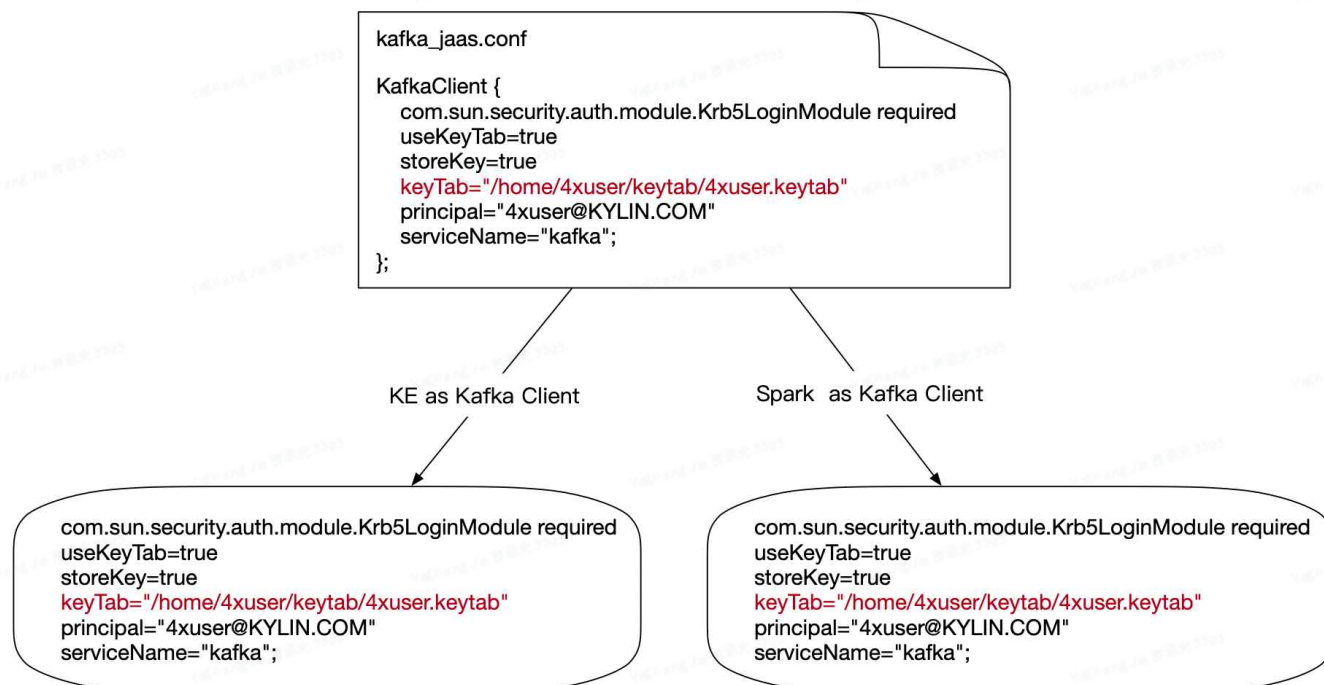
1. Clients (producers, consumers, connect workers, etc) will authenticate to the cluster with their own principal (usually with the same name as the user running the client), so obtain or create these principals as needed. Then configure the JAAS configuration property for each client. Different clients within a JVM may run as different users by specifying different principals. The property `sasl.jaas.config` in `producer.properties` or `consumer.properties` describes how clients like producer and consumer can connect to the Kafka Broker. The following is an example configuration for a client using a keytab (recommended for long-running processes):

```
1 | sasl.jaas.config=com.sun.security.auth.module.Krb5l  
2 |   useKeyTab=true \  
3 |   storeKey=true  \  
4 |   keyTab="/etc/security/keytabs/kafka_client.keyt  
5 |   principal="kafka-client-1@EXAMPLE.COM";
```

可以看到官方推荐 Long-Running 的应用使用keytab登录Kerberos。

## 原先的Design

之前做的第一版Kafka 适配 Kerberos ，其中关于Kafka Kerberos认证的部分Design是这样的。

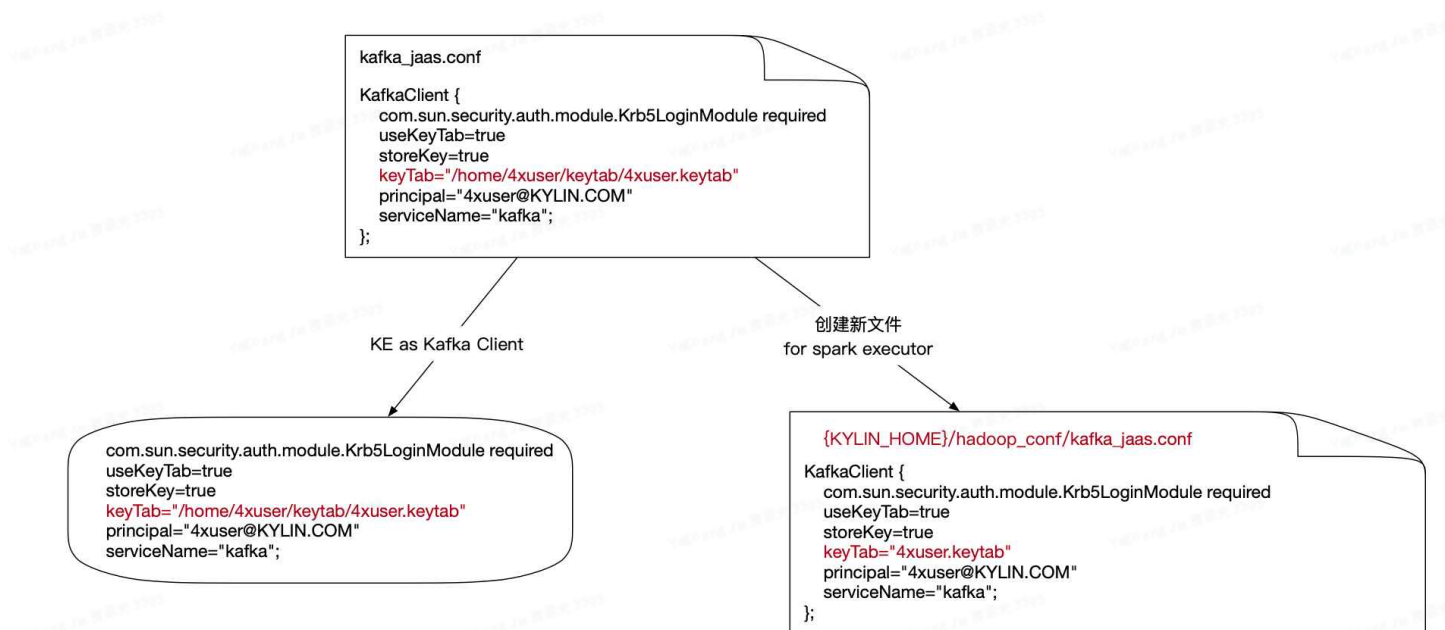


会有如下问题

- 要求Yarn上所有的节点机器在同一目录下放入 `keyTab` 文件，否则Spark Executor 在连接Kafka时会因为找不到 `keyTab` 文件而失败。

这也是本ISSUE要解决的点。

## 现在的Design



## 启动实时构建任务时

1. Spark Driver 使用本地 `{KYLIN_HOME}/conf/kafka_jaas.conf` 文件。
  2. 创建 `{KYLIN_HOME}/hadoop_conf/kafka_jaas.conf` 文件
    - a. 其内容中的keyTab为 **相对路径**
    - b. 将此 **keytab** upload到spark临时目录中，供executor读取
    - c. `{KYLIN_HOME}/hadoop_conf/kafka_jaas.conf`，会一起upload到spark临时目录中，供executor读取
    - d. 每有一个实时构建任务启动，就会overwrite一次 `{KYLIN_HOME}/hadoop_conf/kafka_jaas.conf`
  3. Spark Executor 连接Kafka时，使用的keytab文件就来自于临时目录
- 至此，Spark任务就不需要所有的机器都有个相同路径的keytab文件。

任务启动后观察spark的env配置，可以看到下面两个关键字

```
1 spark.driver.extraJavaOptions=-Djava.security.auth.login.config=
  {KYLIN_HOME}/conf/kafka_jaas.conf
2
3
4 spark.executor.extraJavaOptions=-
  Djava.security.auth.login.config=./__spark_conf__/__hadoop_conf__/kafka_jaas.co
  nf
```