KAFKA

 

<https://github.com/dilipsundarraj1/kafka-for-developers-using-spring-boot/blob/master/SetUpKafka3.md>

* Make sure you are inside the **bin/windows** directory.

**Start Zookeeper and Kafka Broker**

* Start up the Zookeeper.

zookeeper-server-start.bat ..\..\config\zookeeper.properties

* Start up the Kafka Broker.

kafka-server-start.bat ..\..\config\server.properties

## How to create a topic?

kafka-topics.bat --create --topic test-topic --bootstrap-server localhost:9092 --replication-factor 1 --partitions 4

## How to instantiate a Console Producer?

### Without Key

kafka-console-producer.bat --broker-list localhost:9092 --topic test-topic

### With Key

kafka-console-producer.bat --broker-list localhost:9092 --topic test-topic --property "key.separator=-" --property "parse.key=true"

## How to instantiate a Console Consumer?

### Without Key

kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic test-topic --from-beginning

### With Key

kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic test-topic --from-beginning -property "key.separator= - " --property "print.key=true"

### With Consumer Group

kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic test-topic --group <group-name>

## List the topics in a cluster

kafka-topics.bat --bootstrap-server localhost:9092 –list

## How to view consumer groups

kafka-consumer-groups.bat --bootstrap-server localhost:9092 --list

# Enabling SSL in Kafka: -

* Follow the below steps for enabling SSL in your local environment

## Generating the KeyStore

* The below command is to generate the **keyStore**.
* KeyStore in general has information about the server and the organization

keytool -keystore server.keystore.jks -alias localhost -validity 365 -genkey -keyalg RSA

**Example**

* After entering all the details the final value will look like below.

CN=localhost, OU=localhost, O=localhost, L=Chennai, ST=TN, C=IN

## Generating CA

* The below command will generate the ca cert(SSL cert) and private key. This is normally needed if we are self signing the request.

openssl req -new -x509 -keyout ca-key -out ca-cert -days 365 -subj "/CN=local-security-CA"

## Certificate Signing Request(CSR)

* The below command will create a **cert-file** as a result of executing the command.

keytool -keystore server.keystore.jks -alias localhost -certreq -file cert-file

## Signing the certificate

* The below command takes care of signing the CSR and then it spits out a file **cert-signed**

openssl x509 -req -CA ca-cert -CAkey ca-key -in cert-file -out cert-signed -days 365 -CAcreateserial -passin pass:password

* To view the content inside the file **cert-signed**, run the below command.

keytool -printcert -v -file cert-signed

## Adding the Signed Cert in to the KeyStore file

keytool -keystore server.keystore.jks -alias CARoot -import -file ca-cert

keytool -keystore server.keystore.jks -alias localhost -import -file cert-signed

## Generate the TrustStore

* The below command takes care of generating the truststore for us and adds the **CA-Cert** in to it.
* This is to make sure the client is going to trust all the certs issued by CA.

keytool -keystore client.truststore.jks -alias CARoot -import -file ca-cert

## Broker SSL Settings

## Add below property in server.properties

ssl.keystore.location=<location>/server.keystore.jks

ssl.keystore.password=password

ssl.key.password= password

ssl.endpoint.identification.algorithm=

also

listeners=PLAINTEXT://:9092,,SSL://localhost:9095

# Accessing SSL Enabled Topics using Console Producers/Consumers

* Create a topic

kafka-topics.bat --create --topic test-topic --bootstrap-server localhost:9092 --replication-factor 1 --partitions 4

* Create a file named **client-ssl.properties** and have the below properties configured in there.

security.protocol=SSL

ssl.truststore.location=<location>/client.truststore.jks

ssl.truststore.password=password

ssl.truststore.type=JKS

## Producing Messages to Secured Topic

* Command to Produce Messages to the secured topic

kafka-console-producer.bat --broker-list localhost:9095 --topic test-topic --producer.config ..\..\client-ssl.properties

## Consuming Messages from a Secured Topic

* Command to Produce Messages to the secured topic

kafka-console-consumer.bat --bootstrap-server localhost:9095 --topic test-topic --consumer.config ..\..\client-ssl.properties

## Producing Messages to Non-Secured Topic

kafka-console-producer.bat --broker-list localhost:9092 --topic test-topic

## Consuming Messages from a Non-Secured Topic

kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic test-topic

## 2 Way Authentication

* This config is to enable the client authentication at the cluster end.

keytool -keystore server.truststore.jks -alias CARoot -import -file ca-cert

* Add the **ssl.client.auth** property in the **server.properties**.

ssl.truststore.location=<location>/server.truststore.jks

ssl.truststore.password=password

ssl.client.auth=required

* Kafka Client should have the following the config in the **client-ssl.properties** file

ssl.keystore.type=JKS

ssl.keystore.location=<location>/client.keystore.jks

ssl.keystore.password=password

ssl.key.password=password