TLS IMPLEMENT FOR KAFKA IN DOCKER-COMPOSE

1. CREATE CERTIFICATES OF EACH BROKER IN LOCALLY :

A. Using openssl command create ca certificate of certification authority

- openssl req -new -x509 -keyout ca-key -out ca-cert -days 365 -subj "/C=US/ST=Texas/L=Keller/O=Linux Academy”

B. Create each broker’s trust store and keystore JDK files and import the certificates of ca authority

- keytool -keystore client.truststore.jks -alias CARoot -import -file ca-cert
- keytool -keystore server.truststore.jks -alias CARoot -import -file ca-cert
- keytool -keystore kafka-1.keystore.jks -alias localhost -validity 365 -genkey -keyalg RSA -dname "CN=wboyd1c.mylabserver.com, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown" -ext san=dns:kafka-1,dns:localhost,ip:127.0.0.1,ip:172.31.100.110
- keytool -keystore kafka-2.keystore.jks -alias localhost -validity 365 -genkey -keyalg RSA -dname "CN=wboyd1c.mylabserver.com, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown" -ext san=dns:kafka-2,dns:localhost,ip:127.0.0.1,ip:172.31.100.110
- keytool -keystore kafka-3.keystore.jks -alias localhost -validity 365 -genkey -keyalg RSA -dname "CN=wboyd1c.mylabserver.com, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown"

-ext
san=dns:kafka-3,dns:localhost,ip:127.0.0.1,ip:172.31.100.1
10

C. Create certificates of each brokers using keytool command and creating certification request file

- keytool -keystore kafka-1.keystore.jks -alias localhost -certreq -file kafka-1-cert-file
- keytool -keystore kafka-2.keystore.jks -alias localhost -certreq -file kafka-2-cert-file
- keytool -keystore kafka-3.keystore.jks -alias localhost -certreq -file kafka-3-cert-file
- echo subjectAltName = DNS:kafka-1,DNS:localhost,IP:127.0.0.1,IP:172.31.100.11
0 >> kafka-1.cnf

D. Request to CA to sign the certificates of brokers

- openssl x509 -req -CA ca-cert -CAkey ca-key -in kafka-1-cert-file -out kafka-1-cert-signed -days 365 -CAcreateserial
- openssl x509 -req -CA ca-cert -CAkey ca-key -in kafka-2-cert-file -out kafka-2-cert-signed -days 365 -CAcreateserial
- openssl x509 -req -CA ca-cert -CAkey ca-key -in kafka-3-cert-file -out kafka-3-cert-signed -days 365 -CAcreateserial

E. Importing the signed certificates of each broker into the respected broker’s keystore jdk files

- keytool -keystore kafka1.keystore.jks -alias CARoot -import -file ca-cert
- keytool -keystore kafka1.keystore.jks -alias localhost -import -file kafka-1-cert-signed

- keytool -keystore kafka2.keystore.jks -alias CARoot -import -file ca-cert
- keytool -keystore kafka2.keystore.jks -alias localhost -import -file kafka-2-cert-signed
- keytool -keystore kafka3.keystore.jks -alias CARoot -import -file ca-cert
- keytool -keystore kafka3.keystore.jks -alias localhost -import -file kafka-3-cert-signed

2. BUILDING DOCKER-COMPOSE.YML FILE PLAINTEXT PROTOCOL :

DOCKER-COMPOSE.YML :

```
version: '3.7'
services:
  zookeeper:
    image: 'confluentinc/cp-zookeeper'
    container_name: zookeeper
    ports:
      - '2182:2181'
    environment:
      ZOOKEEPER_CLIENT_PORT: 2181
    restart: always

  kafka-1:
    image: 'confluentinc/cp-kafka:latest'
    container_name: kafka-1
    depends_on:
      - zookeeper
    ports:

        - '9094:9094'

    environment:
      KAFKA_BROKER_ID: 1
      KAFKA_ZOOKEEPER_CONNECT: zookeeper:2181
      KAFKA_ADVERTISED_LISTENERS:
PLAINTEXT://localhost:9094
      KAFKA_INTER_BROKER_LISTENER_NAME: INTERNAL
      KAFKA_LISTENER_SECURITY_PROTOCOL_MAP:
INTERNAL:PLAINTEXT,PLAINTEXT:PLAINTEXT
      KAFKA_OFFSETS_TOPIC_REPLICATION_FACTOR: 2
    restart: always

  kafka-2:
    image: 'confluentinc/cp-kafka:latest'
    container_name: kafka-2
    depends_on:
      - zookeeper
      - kafka-1
    ports:
      - '9095:9095'
       environment:
      KAFKA_BROKER_ID: 2
      KAFKA_ZOOKEEPER_CONNECT: zookeeper:2181
      KAFKA_ADVERTISED_LISTENERS:
PLAINTEXT://localhost:9095
      KAFKA_INTER_BROKER_LISTENER_NAME: INTERNAL
      KAFKA_LISTENER_SECURITY_PROTOCOL_MAP:
PLAINTEXT:PLAINTEXT
      KAFKA_OFFSETS_TOPIC_REPLICATION_FACTOR: 2
    restart: always

kafka-3:
    image: 'confluentinc/cp-kafka:latest'
    container_name: kafka-3
    depends_on:
      - zookeeper
      - kafka-1
      - kafka-2
    ports:
      - '9096:9096'
    environment:
      KAFKA_BROKER_ID: 3
      KAFKA_ZOOKEEPER_CONNECT: zookeeper:2181
      KAFKA_ADVERTISED_LISTENERS:
PLAINTEXT://localhost:9096,INTERNAL://kafka-3:9092
      KAFKA_INTER_BROKER_LISTENER_NAME: INTERNAL
      KAFKA_LISTENER_SECURITY_PROTOCOL_MAP:
PLAINTEXT:PLAINTEXT
      KAFKA_OFFSETS_TOPIC_REPLICATION_FACTOR: 2
    restart: always
```

3. CONFIGURING THE EACH CONTAINERS FOR SSL PROTOCOL :

A. ZOOKEEPER :

```
version: '3.7'
services:
  zookeeper:
    image: 'confluentinc/cp-zookeeper'
    container_name: zookeeper
    ports:
      - '2181:2181'
    environment:
      ZOOKEEPER_CLIENT_PORT: 2181
      ZOOKEEPER_TICK_TIME: 2000
      ZOOKEEPER_SASL_ENABLED: 'false'
```

B. KAFKA BROKER 1 :

```
kafka-1:
    image: 'confluentinc/cp-kafka:latest'
    container_name: kafka-1
    depends_on:
      - zookeeper
    ports:
```

```
  - '9094:9094'
 environment:
   KAFKA_BROKER_ID: 1
   KAFKA_ZOOKEEPER_CONNECT: zookeeper:2181

   KAFKA_ADVERTISED_LISTENERS:
PLAINTEXT://localhost:9094,SSL://kafka-1:9097
   KAFKA_LISTENER_SECURITY_PROTOCOL_MAP:
PLAINTEXT:PLAINTEXT,SSL:SSL
   KAFKA_OFFSETS_TOPIC_REPLICATION_FACTOR: 2
   KAFKA_SSL_KEYSTORE_FILENAME: kafk1.keystore.jks
   KAFKA_SSL_KEYSTORE_CREDENTIALS: ssl.creds
   KAFKA_SSL_KEY_CREDENTIALS: ssl.creds
   KAFKA_SSL_TRUSTSTORE_FILENAME:
server.truststore.jks
   KAFKA_SSL_TRUSTSTORE_CREDENTIALS: ssl.creds
   KAFKA_SSL_CLIENT_AUTH: 'required'
   KAFKA_SECURITY_PROTOCOL: SSL
```

Files :

- Server.truststore.jks >>> Contains the CA certificate
- Kafka1.keystore.jks >>> Contains certificate signed by the CA (kafka-1-cert-signed file and ca cert)
- Ssl.creds >>> paswds of keystore.jks files and truststore.jks files

C. KAFKA BROKER 2 :

```
kafka-2:
 image: 'confluentinc/cp-kafka:latest'
 container_name: kafka-2
 depends_on:
  - zookeeper
```

```
    - kafka-1
   ports:
    - '9095:9095'
   environment:
      KAFKA_BROKER_ID: 2
      KAFKA_ZOOKEEPER_CONNECT: zookeeper:2181
      KAFKA_ADVERTISED_LISTENERS:
PLAINTEXT://localhost:9095,SSL://kafka-1:9097
      KAFKA_LISTENER_SECURITY_PROTOCOL_MAP:
PLAINTEXT:PLAINTEXT,SSL:SSL
      KAFKA_OFFSETS_TOPIC_REPLICATION_FACTOR: 2
      KAFKA_SSL_KEYSTORE_FILENAME: kafka2.keystore.jks
      KAFKA_SSL_KEYSTORE_CREDENTIALS: ssl.creds
      KAFKA_SSL_KEY_CREDENTIALS: ssl.creds
      KAFKA_SSL_TRUSTSTORE_FILENAME:
server.truststore.jks
      KAFKA_SSL_TRUSTSTORE_CREDENTIALS: ssl.creds
      KAFKA_SSL_CLIENT_AUTH: 'required'
      KAFKA_SECURITY_PROTOCOL: SSL
```

Files :

- Server.truststore.jks >>> Contains the CA certificate
- Kafka2.keystore.jks >>> Contains certificate signed by the CA (kafka-2-cert-signed file and ca cert)
- Ssl.creds >>> paswds of keystore.jks files and truststore.jks files

D. KAFKA BROKER 3:

```
 kafka-3:
   image: 'confluentinc/cp-kafka:latest'
   container_name: kafka-3
   depends_on:
    - zookeeper
```

```
      - kafka-1
      - kafka-2
    ports:
      - '9096:9096’
    environment:
        KAFKA_BROKER_ID: 3
        KAFKA_ZOOKEEPER_CONNECT: zookeeper:2181
        KAFKA_ADVERTISED_LISTENERS:
PLAINTEXT://localhost:9096,SSL://kafka-1:9097
        KAFKA_LISTENER_SECURITY_PROTOCOL_MAP:
PLAINTEXT:PLAINTEXT,SSL:SSL
        KAFKA_OFFSETS_TOPIC_REPLICATION_FACTOR: 2
        KAFKA_SSL_KEYSTORE_FILENAME: kafka3.keystore.jks
        KAFKA_SSL_KEYSTORE_CREDENTIALS: ssl.creds
        KAFKA_SSL_KEY_CREDENTIALS: ssl.creds
        KAFKA_SSL_TRUSTSTORE_FILENAME:
server.truststore.jks
        KAFKA_SSL_TRUSTSTORE_CREDENTIALS: ssl.creds
        KAFKA_SSL_CLIENT_AUTH: 'required'
        KAFKA_SECURITY_PROTOCOL: SSL
```

Files :

- Server.truststore.jks >>> Contains the CA certificate
- Kafka3.keystore.jks >>> Contains certificate signed by the CA (kafka-3-cert-signed file and ca cert)
- Ssl.creds >>> paswds of keystore.jks files and truststore.jks files

E. Adding volume :

```
 volumes:
        - /home/charan/stores:/etc/kafka/secrets
```

Here we are mounting the volumes on each broker to access the files for each container brokers. If any changes we will make in local files that we have made then Files in containers will also

change.
The /home/charan/stores directory contains the >>

1). server.truststore.jks
2). kafka1.keystore.jks
3). Kafka2.keystore.jks
4). Kafka3.keystore.jks
5). Client.truststore.jks
6). ssl.creds