Scenario 8

Problem-statement :

The kafka brokers are down after an upgrade to a newer version. Several properties and files were changed during the upgrade and the customer does not have an audit of the changes.
Review the logs and come up with a solution to start the kafka brokers.

Observation :

When I run docker-compose up -d then I observed that kafka1, kafka2, kafka3 were down . I have checked the logs and observed the ssl engine issue

Errors :

org.apache.kafka.common.config.ConfigException: Invalid value
javax.net.ssl.SSLHandshakeException: Empty client certificate chain for configuration A client SSLEngine created with the provided settings can't connect to a server SSLEngine created with those settings.
at org.apache.kafka.common.security.ssl.SslFactory.configure(SslFactory.java:103)
at
org.apache.kafka.common.network.SslChannelBuilder.configure(SslChannelBuilder.java:84)
at org.apache.kafka.common.network.ChannelBuilders.create(ChannelBuilders.java:265)
at
org.apache.kafka.common.network.ChannelBuilders.serverChannelBuilder(ChannelBuilders.java:166)
at kafka.network.Processor.<init>(SocketServer.scala:1177)
at kafka.network.Acceptor.newProcessor(SocketServer.scala:1050)
at kafka.network.Acceptor.$anonfun$addProcessors$1(SocketServer.scala:1010)
at scala.collection.immutable.Range.foreachmVcsp(Range.scala:190)
at kafka.network.Acceptor.addProcessors(SocketServer.scala:1009)
at kafka.network.DataPlaneAcceptor.configure(SocketServer.scala:670)
at
kafka.network.SocketServer.createDataPlaneAcceptorAndProcessors(SocketServer.scala:278)
at kafka.network.SocketServer.$anonfun$new$51(SocketServer.scala:224)
at kafka.network.SocketServer.$anonfun$new51adapted(SocketServer.scala:224)
at scala.collection.IterableOnceOps.foreach(IterableOnce.scala:575)
at scala.collection.IterableOnceOps.foreach$(IterableOnce.scala:573)
at scala.collection.AbstractIterable.foreach(Iterable.scala:933)
at kafka.network.SocketServer.<init>(SocketServer.scala:224)
at kafka.server.KafkaServer.startup(KafkaServer.scala:528)
at kafka.Kafka$.main(Kafka.scala:114)

Aproach / Method :

TLS AND MTLS (renewal of certificates) &&
valid configuration superuser names for kafkabrokers and broker.users in kafka brokers
Importing the correct certificates to the truststore

Detailed Solution :

1. **Renewal of certificates** :
   While I up the docker-compose first I observed the kafka-brokers are down and then I logged the each containers then I have got the ssl certificate errors .then I have checked the certificate validities and passwords and correct paths and matched . I pointed out that error was due to certificate validity they have expired .
   Then I have renewed the certificates of all brokers by using commands
   Commands :

   keytool -keystore kafka.server.keystore.jks -alias localhost -certreq -file kafka-broker-new.csr

   openssl x509 -req -CA ~/tasks/cp-sandbox/certs/ca-cert -CAkey ~/tasks/cp-sandbox/certs/ca-key -in kafka-broker-new.csr -out kafka-broker-new-signed-cert.pem -days 365 -CAcreateserial

   keytool -keystore kafka.server.keystore.jks -alias localhost -import -file kafka-broker-new-signed-cert.pem

   Again then I have checked the certificates and then watched the validate the expiry date its good and updated for 365 days.

2. **Importing the ca-cert to the trust store :**

   Command :

   keytool -import -trustcacerts -file ~/tasks/cp-sandbox/certs/ca-cert -alias caroot -keystore kafka.server.truststore.jks

   After adding the (importing) the ca-cert then again up the broker or restart the broker then we can see they are up and running (healthy)

Conclusion :

- [ ] If we get error like Empty client certificate chain for configuration A client we have to check the truststore and keystore certificates and

check the validities and next look for the correct chains like ca-cert s and broker certs