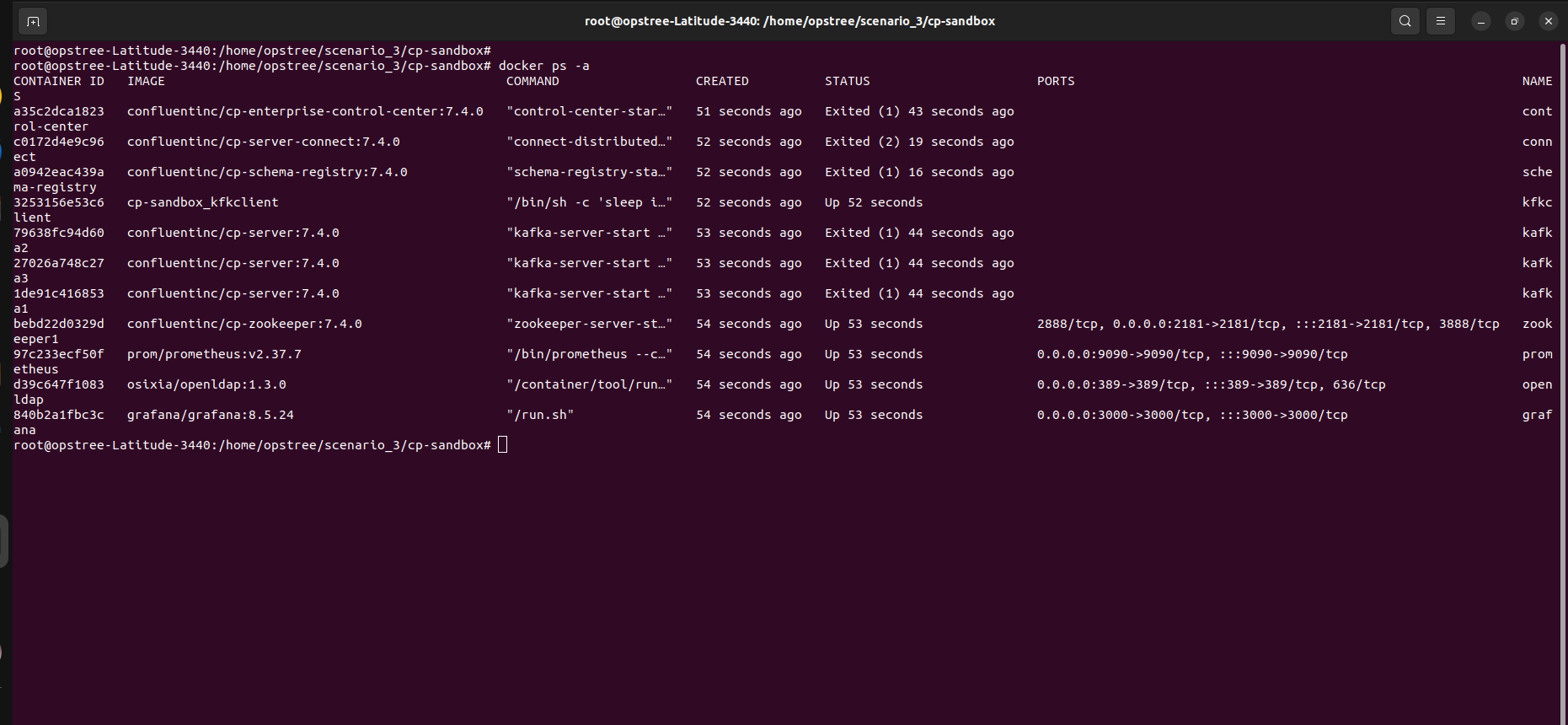
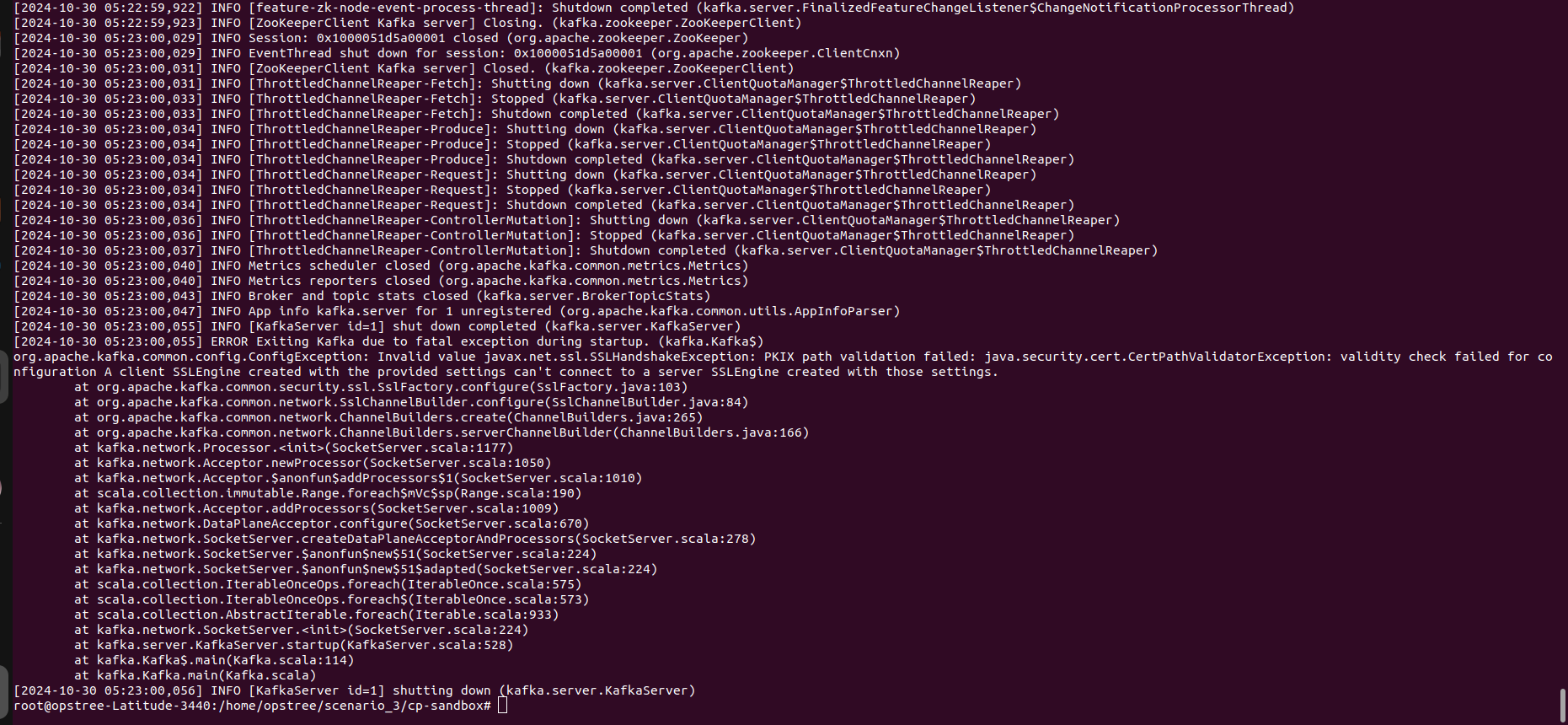
**Scenario 3**



SSL Handshake error:



Generate certificates since the existing have expired.

### **Steps to Set Up Kafka with New Certificates**

1. **Clean Up Old Files**

#### **Clean Up Old Files in ~/scenario\_3/cp-sandbox/certs:**

cd cp-sandbox/certs  
rm -f ca-key ca-cert cert-signed-1 cert-signed-2 cert-signed-3 cert-file-1 cert-file-2 cert-file-3 ca-cert.srl

#### **Clean Up Old Files in Each Broker Directory:**

Assuming that the Kafka broker directories (kafka1, kafka2, kafka3) are inside ~/scenario\_3/cp-sandbox/, execute the following:

# For kafka1  
cd ../kafka1  
rm -f kafka.server.truststore.jks kafka.server.keystore.jks  
  
# For kafka2  
cd ../kafka2  
rm -f kafka.server.truststore.jks kafka.server.keystore.jks  
  
# For kafka3  
cd ../kafka3  
rm -f kafka.server.truststore.jks kafka.server.keystore.jks

1. **Create New Certificates for the CA**

Run the following commands in the ~/scenario\_3/cp-sandbox/certs directory:

cd ../certs  
  
# Step 1: Create a new Root CA  
openssl req -new -x509 -days 365 -keyout ca-key -out ca-cert -subj "/C=DE/ST=NRW/L=MS/O=juplo/OU=kafka/CN=Root-CA" -passout pass:kafka-broker

1. **Create Truststore for Each Kafka Broker**

Now, import the CA certificate into the truststore for each broker:

# For kafka1  
keytool -keystore ../kafka1/kafka.server.truststore.jks -storepass kafka-broker -import -alias ca-root -file ca-cert -noprompt  
  
# For kafka2  
keytool -keystore ../kafka2/kafka.server.truststore.jks -storepass kafka-broker -import -alias ca-root -file ca-cert -noprompt  
  
# For kafka3  
keytool -keystore ../kafka3/kafka.server.truststore.jks -storepass kafka-broker -import -alias ca-root -file ca-cert -noprompt

1. **Generate Keystores and Certificates for Each Broker**

#### **For kafka1:**

# Step 3: Generate a Key Pair for kafka1  
keytool -keystore ../kafka1/kafka.server.keystore.jks -storepass kafka-broker -alias kafka1 -validity 365 -keyalg RSA -genkeypair -keypass kafka-broker -dname "CN=kafka1,OU=kafka,O=juplo,L=MS,ST=NRW,C=DE"  
  
# Step 4: Create a CSR for kafka1  
keytool -alias kafka1 -keystore ../kafka1/kafka.server.keystore.jks -certreq -file cert-file-1 -storepass kafka-broker -keypass kafka-broker  
  
# Step 5: Sign the CSR with the CA  
openssl x509 -req -CA ca-cert -CAkey ca-key -in cert-file-1 -out cert-signed-1 -days 365 -CAcreateserial -passin pass:kafka-broker -extensions SAN -extfile <(printf "\n[SAN]\nsubjectAltName=DNS:kafka1,DNS:localhost")  
  
# Step 6: Import the CA Certificate into the Keystore  
keytool -importcert -keystore ../kafka1/kafka.server.keystore.jks -alias ca-root -file ca-cert -storepass kafka-broker -keypass kafka-broker -noprompt  
  
# Step 7: Import the Signed Certificate into the Keystore  
keytool -keystore ../kafka1/kafka.server.keystore.jks -alias kafka1 -import -file cert-signed-1 -storepass kafka-broker -keypass kafka-broker -noprompt

#### **For kafka2:**

# Step 3: Generate a Key Pair for kafka2  
keytool -keystore ../kafka2/kafka.server.keystore.jks -storepass kafka-broker -alias kafka2 -validity 365 -keyalg RSA -genkeypair -keypass kafka-broker -dname "CN=kafka2,OU=kafka,O=juplo,L=MS,ST=NRW,C=DE"  
  
# Step 4: Create a CSR for kafka2  
keytool -alias kafka2 -keystore ../kafka2/kafka.server.keystore.jks -certreq -file cert-file-2 -storepass kafka-broker -keypass kafka-broker  
  
# Step 5: Sign the CSR with the CA  
openssl x509 -req -CA ca-cert -CAkey ca-key -in cert-file-2 -out cert-signed-2 -days 365 -CAcreateserial -passin pass:kafka-broker -extensions SAN -extfile <(printf "\n[SAN]\nsubjectAltName=DNS:kafka2,DNS:localhost")  
  
# Step 6: Import the CA Certificate into the Keystore  
keytool -importcert -keystore ../kafka2/kafka.server.keystore.jks -alias ca-root -file ca-cert -storepass kafka-broker -keypass kafka-broker -noprompt  
  
# Step 7: Import the Signed Certificate into the Keystore  
keytool -keystore ../kafka2/kafka.server.keystore.jks -alias kafka2 -import -file cert-signed-2 -storepass kafka-broker -keypass kafka-broker -noprompt

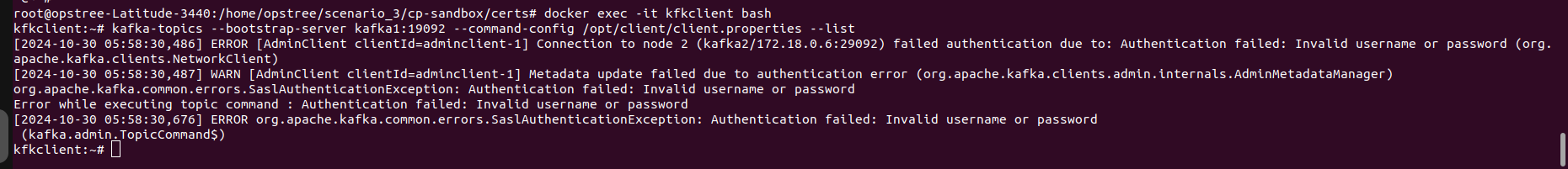
#### **For kafka3:**

# Step 3: Generate a Key Pair for kafka3  
keytool -keystore ../kafka3/kafka.server.keystore.jks -storepass kafka-broker -alias kafka3 -validity 365 -keyalg RSA -genkeypair -keypass kafka-broker -dname "CN=kafka3,OU=kafka,O=juplo,L=MS,ST=NRW,C=DE"  
  
# Step 4: Create a CSR for kafka3  
keytool -alias kafka3 -keystore ../kafka3/kafka.server.keystore.jks -certreq -file cert-file-3 -storepass kafka-broker -keypass kafka-broker  
  
# Step 5: Sign the CSR with the CA  
openssl x509 -req -CA ca-cert -CAkey ca-key -in cert-file-3 -out cert-signed-3 -days 365 -CAcreateserial -passin pass:kafka-broker -extensions SAN -extfile <(printf "\n[SAN]\nsubjectAltName=DNS:kafka3,DNS:localhost")  
  
# Step 6: Import the CA Certificate into the Keystore  
keytool -importcert -keystore ../kafka3/kafka.server.keystore.jks -alias ca-root -file ca-cert -storepass kafka-broker -keypass kafka-broker -noprompt  
  
# Step 7: Import the Signed Certificate into the Keystore  
keytool -keystore ../kafka3/kafka.server.keystore.jks -alias kafka3 -import -file cert-signed-3 -storepass kafka-broker -keypass kafka-broker -noprompt

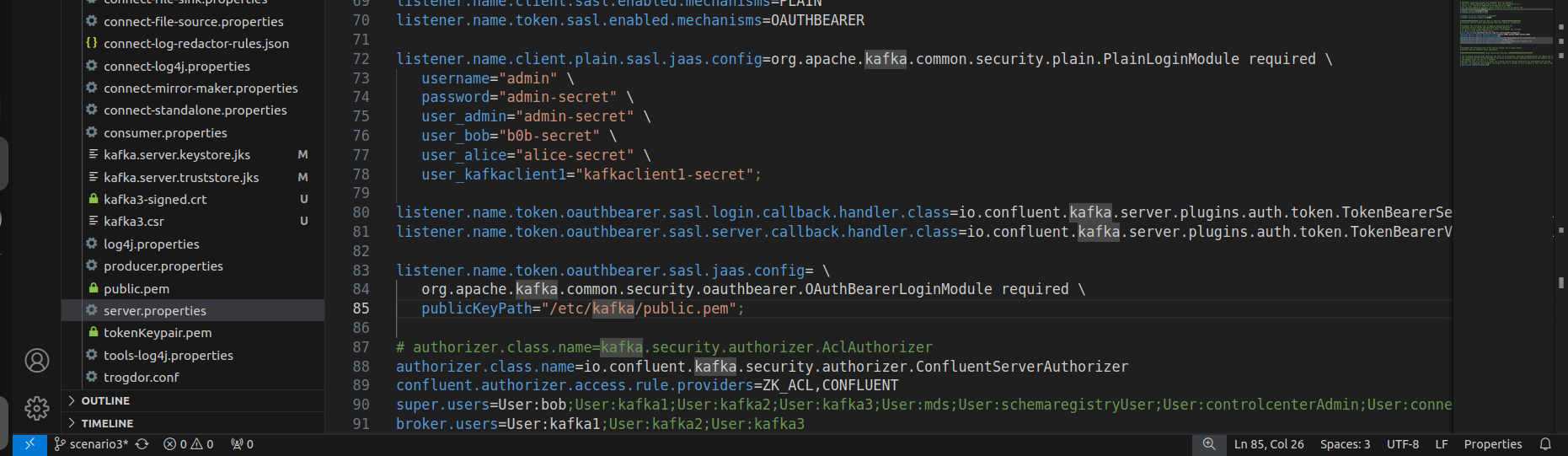
Now,



Error in testing connection with kafka broker



Incorrect password in kafka2 and kafka3 server.properties:



Password corrected to bob-secret

Incorrect (old) line:  
  
 user\_bob="b0b-secret"  
  
Correct (new) line:  
  
 user\_bob="bob-secret"

Restart docker service:

Check again:

Client:  
  
 To use a kafka client, exec into the kfkclient container which contains the Kafka CLI and other tools necessary for troubleshooting Kafka. THe kfkclient container also contains a properties file mounted to /opt/client, which can be used to define the client properties for communicating with Kafka.  
  
 docker exec -it kfkclient bash  
  
 Test Connection with Kafka Brokers: Use the following commands to list topics for each Kafka broker:  
  
 Kafka1 (Port: 19092):  
  
 kafka-topics --bootstrap-server kafka1:19092 --command-config /opt/client/client.properties --list  
  
 Kafka2 (Port: 29092):  
  
 kafka-topics --bootstrap-server kafka2:29092 --command-config /opt/client/client.properties --list  
  
 Kafka3 (Port: 39092):  
  
 kafka-topics --bootstrap-server kafka3:39092 --command-config /opt/client/client.properties --list

