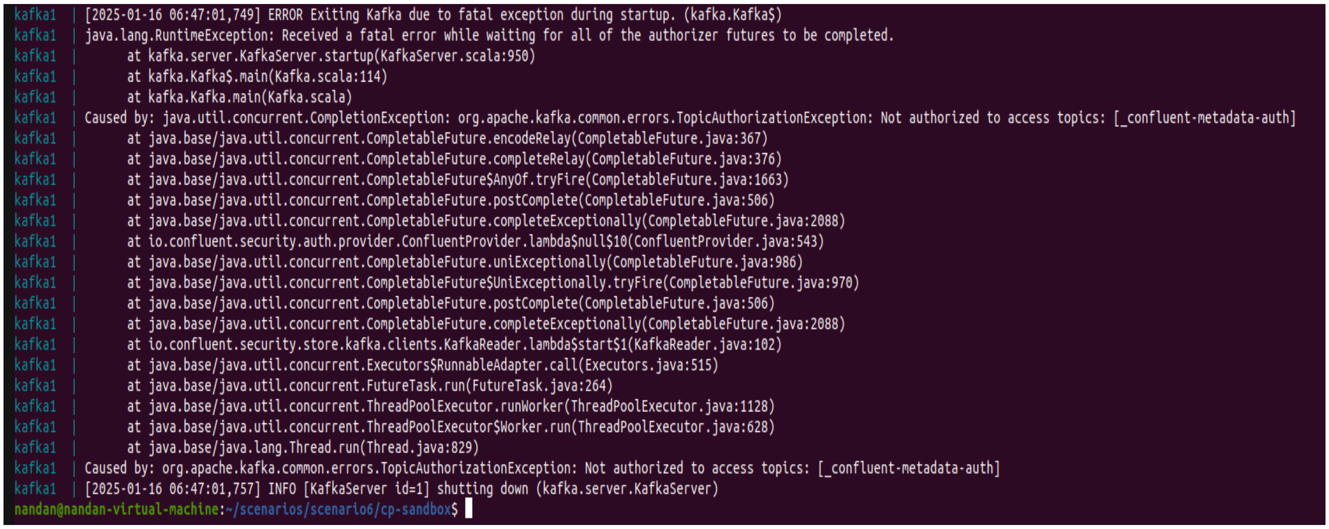
Scenario-6

Problem Statement

The client just performed a lift and shift on their entire platform to different machines. The brokers and several other components are down.

1. Check logs of kafka1, kafka2 and kafka3 containers

→ docker compose logs kafka1



```
kafka1  | [2025-01-16 06:47:01,749] ERROR Exiting Kafka due to fatal exception during startup. (kafka.Kafka$)
kafka1  | java.lang.RuntimeException: Received a fatal error while waiting for all of the authorizer futures to be completed.
kafka1  |       at kafka.server.KafkaServer.startup(KafkaServer.scala:950)
kafka1  |       at kafka.Kafka$.main(Kafka.scala:114)
kafka1  |       at kafka.Kafka.main(Kafka.scala)
kafka1  | Caused by: java.util.concurrent.CompletionException: org.apache.kafka.common.errors.TopicAuthorizationException: Not authorized to access topics: [_confluent-metadata-auth]
kafka1  |       at java.base/java.util.concurrent.CompletableFuture.encodeRelay(CompletableFuture.java:367)
kafka1  |       at java.base/java.util.concurrent.CompletableFuture.completeRelay(CompletableFuture.java:376)
kafka1  |       at java.base/java.util.concurrent.CompletableFuture$AnyOf.tryFire(CompletableFuture.java:1663)
kafka1  |       at java.base/java.util.concurrent.CompletableFuture.postComplete(CompletableFuture.java:506)
kafka1  |       at java.base/java.util.concurrent.CompletableFuture.completeExceptionally(CompletableFuture.java:2088)
kafka1  |       at io.confluent.security.auth.provider.ConfluentProvider.lambda$null$10(ConfluentProvider.java:543)
kafka1  |       at java.base/java.util.concurrent.CompletableFuture.uniExceptionally(CompletableFuture.java:986)
kafka1  |       at java.base/java.util.concurrent.CompletableFuture$UniExceptionally.tryFire(CompletableFuture.java:970)
kafka1  |       at java.base/java.util.concurrent.CompletableFuture.postComplete(CompletableFuture.java:506)
kafka1  |       at java.base/java.util.concurrent.CompletableFuture.completeExceptionally(CompletableFuture.java:2088)
kafka1  |       at io.confluent.security.store.kafka.clients.KafkaReader.lambda$start$1(KafkaReader.java:102)
kafka1  |       at java.base/java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:515)
kafka1  |       at java.base/java.util.concurrent.FutureTask.run(FutureTask.java:264)
kafka1  |       at java.base/java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1128)
kafka1  |       at java.base/java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:628)
kafka1  |       at java.base/java.lang.Thread.run(Thread.java:829)
kafka1  | Caused by: org.apache.kafka.common.errors.TopicAuthorizationException: Not authorized to access topics: [_confluent-metadata-auth]
kafka1  | [2025-01-16 06:47:01,757] INFO [KafkaServer id=1] shutting down (kafka.server.KafkaServer)
nandan@nandan-virtual-machine:~/scenarios/scenario6/cp-sandbox$
```

Observation

It was noticed that kafka1, 2 and 3 all have the same error logs, and indicated that it does not have access to the topic “confluent-metadata-auth”.

Root cause

The brokers do not have access to the topic “confluent-metadata-auth”. This topic is an **internal topic** created and managed by **Confluent's Metadata Service (MDS)**. It is used to store **access control metadata** for managing authentication and authorization in a Confluent Platform deployment.

Purpose

- It stores **authorization policies** (ACLs) for resources in your Confluent deployment.
- Used by the MDS to manage and replicate ACL configurations for components like Kafka, Schema Registry, Connect, and Control Center.

Content

- The topic holds serialized data about:
  - Role-based access control (RBAC) mappings.
  - Permissions granted to users, service accounts, or other principals.
  - Metadata on access policies applied to topics, consumer groups, and other Kafka resources.

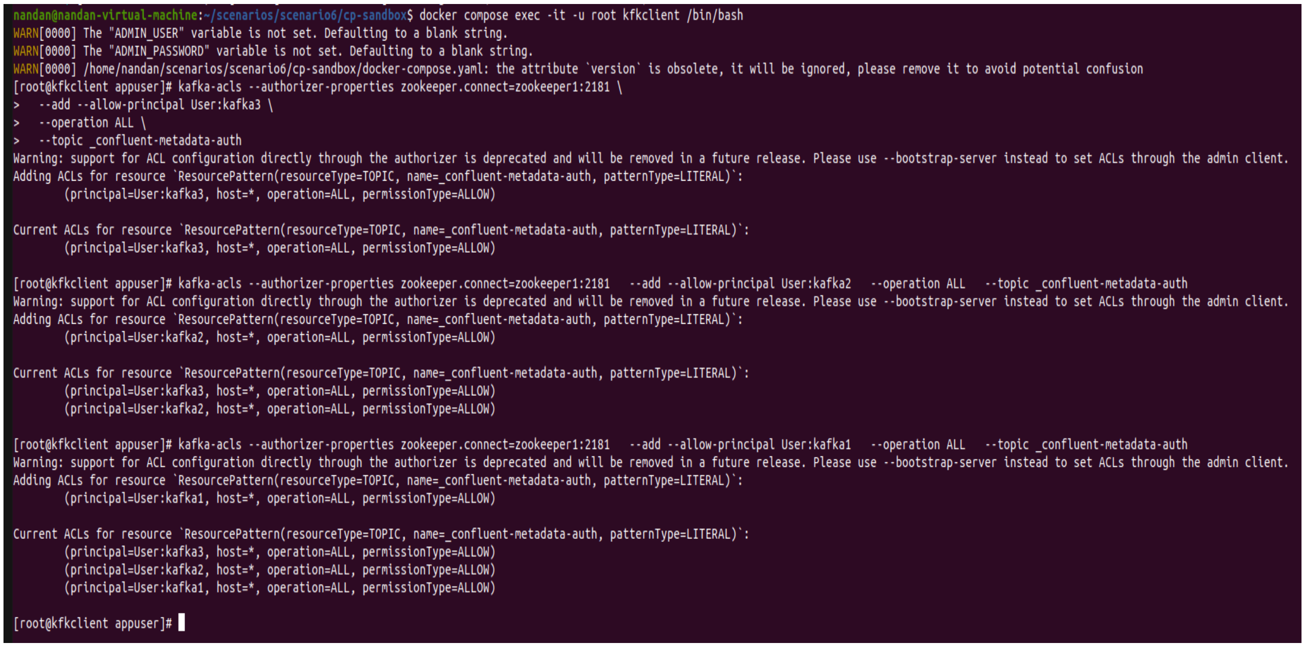
Solution to debug the issue:

1. Manually create an ACL for the clients to access the topic "confluent-metadata-auth".

→kafka-acls --authorizer-properties zookeeper.connect=zookeeper1:2181 --add --allow-principal User:kafka1   --operation ALL --topic _confluent-metadata-auth

→kafka-acls --authorizer-properties zookeeper.connect=zookeeper1:2181 --add --allow-principal User:kafka2   --operation ALL --topic _confluent-metadata-auth

→kafka-acls --authorizer-properties zookeeper.connect=zookeeper1:2181 --add --allow-principal User:kafka3   --operation ALL --topic _confluent-metadata-auth



```
nandan@nandan-virtual-machine:~/scenarios/scenario6/cp-sandbox$ docker compose exec -it -u root kfkclient /bin/bash
WARN[0000] The "ADMIN_USER" variable is not set. Defaulting to a blank string.
WARN[0000] The "ADMIN_PASSWORD" variable is not set. Defaulting to a blank string.
WARN[0000] /home/nandan/scenarios/scenario6/cp-sandbox/docker-compose.yaml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion
[root@kfkclient appuser]# kafka-acls --authorizer-properties zookeeper.connect=zookeeper1:2181 \
>   --add --allow-principal User:kafka3 \
>   --operation ALL \
>   --topic _confluent-metadata-auth
Warning: support for ACL configuration directly through the authorizer is deprecated and will be removed in a future release. Please use --bootstrap-server instead to set ACLs through the admin client.
Adding ACLs for resource `ResourcePattern(resourceType=TOPIC, name=_confluent-metadata-auth, patternType=LITERAL)`:
        (principal=User:kafka3, host=*, operation=ALL, permissionType=ALLOW)

Current ACLs for resource `ResourcePattern(resourceType=TOPIC, name=_confluent-metadata-auth, patternType=LITERAL)`:
        (principal=User:kafka3, host=*, operation=ALL, permissionType=ALLOW)

[root@kfkclient appuser]# kafka-acls --authorizer-properties zookeeper.connect=zookeeper1:2181   --add --allow-principal User:kafka2   --operation ALL   --topic _confluent-metadata-auth
Warning: support for ACL configuration directly through the authorizer is deprecated and will be removed in a future release. Please use --bootstrap-server instead to set ACLs through the admin client.
Adding ACLs for resource `ResourcePattern(resourceType=TOPIC, name=_confluent-metadata-auth, patternType=LITERAL)`:
        (principal=User:kafka2, host=*, operation=ALL, permissionType=ALLOW)

Current ACLs for resource `ResourcePattern(resourceType=TOPIC, name=_confluent-metadata-auth, patternType=LITERAL)`:
        (principal=User:kafka3, host=*, operation=ALL, permissionType=ALLOW)
        (principal=User:kafka2, host=*, operation=ALL, permissionType=ALLOW)

[root@kfkclient appuser]# kafka-acls --authorizer-properties zookeeper.connect=zookeeper1:2181   --add --allow-principal User:kafka1   --operation ALL   --topic _confluent-metadata-auth
Warning: support for ACL configuration directly through the authorizer is deprecated and will be removed in a future release. Please use --bootstrap-server instead to set ACLs through the admin client.
Adding ACLs for resource `ResourcePattern(resourceType=TOPIC, name=_confluent-metadata-auth, patternType=LITERAL)`:
        (principal=User:kafka1, host=*, operation=ALL, permissionType=ALLOW)

Current ACLs for resource `ResourcePattern(resourceType=TOPIC, name=_confluent-metadata-auth, patternType=LITERAL)`:
        (principal=User:kafka3, host=*, operation=ALL, permissionType=ALLOW)
        (principal=User:kafka2, host=*, operation=ALL, permissionType=ALLOW)
        (principal=User:kafka1, host=*, operation=ALL, permissionType=ALLOW)

[root@kfkclient appuser]# 
```

Observation

Successfully created the ACL for all the brokers.

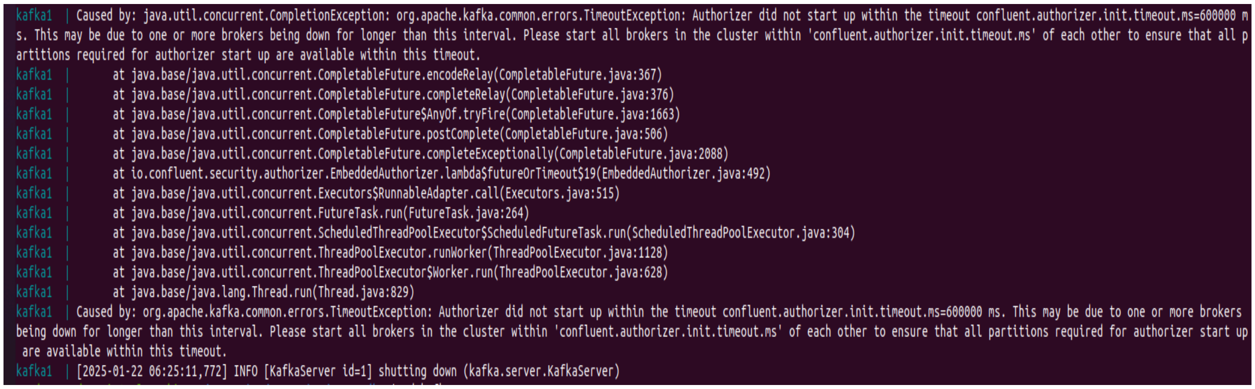
2. Now restart the containers

→ docker compose restart kafka1

→ docker compose restart kafka2

→ docker compose restart kafka3

After restarting the kafka brokers, they are failing with the following errors.

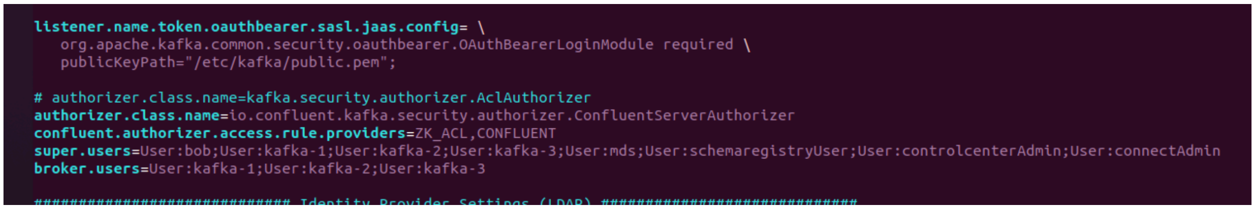


```
kafka1  | Caused by: java.util.concurrent.CompletionException: org.apache.kafka.common.errors.TimeoutException: Authorizer did not start up within the timeout confluent.authorizer.init.timeout.ms=600000 m
s. This may be due to one or more brokers being down for longer than this interval. Please start all brokers in the cluster within 'confluent.authorizer.init.timeout.ms' of each other to ensure that all p
artitions required for authorizer start up are available within this timeout.
kafka1  |       at java.base/java.util.concurrent.CompletableFuture.encodeRelay(CompletableFuture.java:367)
kafka1  |       at java.base/java.util.concurrent.CompletableFuture.completeRelay(CompletableFuture.java:376)
kafka1  |       at java.base/java.util.concurrent.CompletableFuture$AnyOf.tryFire(CompletableFuture.java:1663)
kafka1  |       at java.base/java.util.concurrent.CompletableFuture.postComplete(CompletableFuture.java:506)
kafka1  |       at java.base/java.util.concurrent.CompletableFuture.completeExceptionally(CompletableFuture.java:2088)
kafka1  |       at io.confluent.security.authorizer.EmbeddedAuthorizer.lambda$futureOrTimeout$19(EmbeddedAuthorizer.java:492)
kafka1  |       at java.base/java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:515)
kafka1  |       at java.base/java.util.concurrent.FutureTask.run(FutureTask.java:264)
kafka1  |       at java.base/java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.run(ScheduledThreadPoolExecutor.java:304)
kafka1  |       at java.base/java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1128)
kafka1  |       at java.base/java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:628)
kafka1  |       at java.base/java.lang.Thread.run(Thread.java:829)
kafka1  | Caused by: org.apache.kafka.common.errors.TimeoutException: Authorizer did not start up within the timeout confluent.authorizer.init.timeout.ms=600000 ms. This may be due to one or more brokers
being down for longer than this interval. Please start all brokers in the cluster within 'confluent.authorizer.init.timeout.ms' of each other to ensure that all partitions required for authorizer start up
 are available within this timeout.
kafka1  | [2025-01-22 06:25:11,772] INFO [KafkaServer id=1] shutting down (kafka.server.KafkaServer)
```

Observation

Kafka brokers fails after waiting for Authorizer to start up within the timeout specified in confluent.authorizer.init.timeout.ms=600000

3. Check the properties file of the brokers for any error while setting them up.

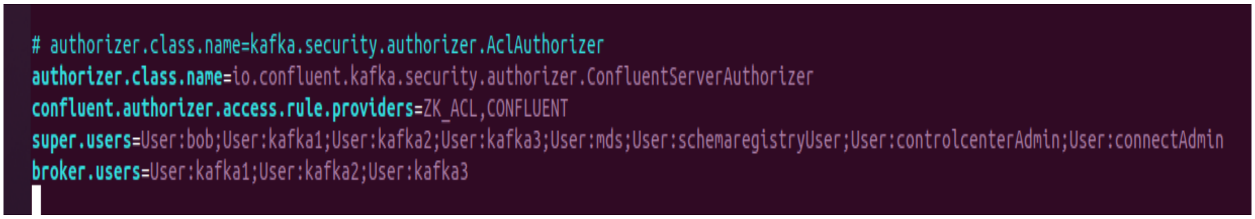


```
listener.name.token.oauthbearer.sasl.jaas.config= \
   org.apache.kafka.common.security.oauthbearer.OAuthBearerLoginModule required \
   publicKeyPath="/etc/kafka/public.pem";

# authorizer.class.name=kafka.security.authorizer.AclAuthorizer
authorizer.class.name=io.confluent.kafka.security.authorizer.ConfluentServerAuthorizer
confluent.authorizer.access.rule.providers=ZK_ACL,CONFLUENT
super.users=User:bob;User:kafka-1;User:kafka-2;User:kafka-3;User:mds;User:schemaregistryUser;User:controlcenterAdmin;User:connectAdmin
broker.users=User:kafka-1;User:kafka-2;User:kafka-3

############################# Identity Provider Settings (LDAP) #############################
```

Observation

It is observed that in the broker users and super users list, the names of the users are set up with kafka-1, kafka-2 and kafka-3 for broker 1, broker 2 and broker 3 respectively which were failing to get authenticated with the CN names which were given in the certificates in the time of creation.

4. Change the names of the users to match the CN names on the certificates.



```
# authorizer.class.name=kafka.security.authorizer.AclAuthorizer
authorizer.class.name=io.confluent.kafka.security.authorizer.ConfluentServerAuthorizer
confluent.authorizer.access.rule.providers=ZK_ACL,CONFLUENT
super.users=User:bob;User:kafka1;User:kafka2;User:kafka3;User:mds;User:schemaregistryUser;User:controlcenterAdmin;User:connectAdmin
broker.users=User:kafka1;User:kafka2;User:kafka3
```

Observation

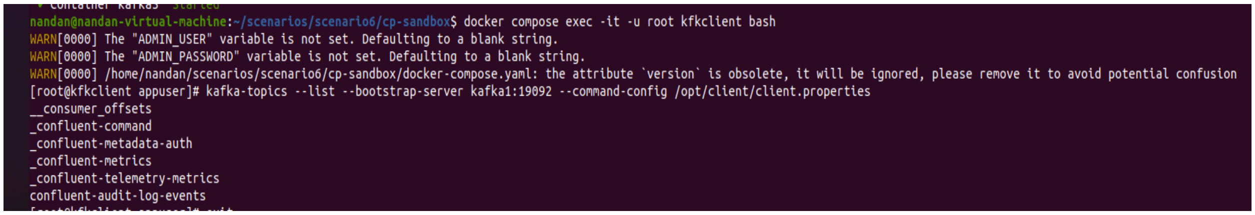
Successfully changed the names of the users to match the CN names in the certificate.

5. Now restart the containers

→ docker compose restart kafka1

→ docker compose restart kafka2

→ docker compose restart kafka3



```
 ✔ Container kafka3  Started
nandan@nandan-virtual-machine:~/scenarios/scenario6/cp-sandbox$ docker compose exec -it -u root kfkclient bash
WARN[0000] The "ADMIN_USER" variable is not set. Defaulting to a blank string.
WARN[0000] The "ADMIN_PASSWORD" variable is not set. Defaulting to a blank string.
WARN[0000] /home/nandan/scenarios/scenario6/cp-sandbox/docker-compose.yaml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion
[root@kfkclient appuser]# kafka-topics --list --bootstrap-server kafka1:19092 --command-config /opt/client/client.properties
__consumer_offsets
_confluent-command
_confluent-metadata-auth
_confluent-metrics
_confluent-telemetry-metrics
confluent-audit-log-events
[root@kfkclient appuser]# exit
```

Conclusion

Since the brokers were not authorized to access the topic "confluent-metadata-auth", the error was thrown, once the ACL for the brokers on that particular topic was added it was observed that the kafka brokers were failing after waiting for authoriser to start, upon further inspection of the setup of the kafka brokers, the usernames were not aligned with the CN names of the certificates, changing them to match the CN names on the certificate successfully resolved the issue.