Steps to perform for Zookeeper to Kraft migration

1. Remove below server properties from kafka broker and add the below annotation. Kafka brokers will take a RESTART. This annotation will help kraft to maintain the same IBP version throughout the migration process. Upon RESTART the IBP of kafka broker will degrade to 2.6

```
#REMOVE

  configOverrides:
    server:
    - inter.broker.protocol.version=3.7
        - authorizer.class.name=kafka.security.authorizer.AclAuthorizer
    - super.users=User:kafka.confluent.svc.cluster.local
        - kafka.rest.client.security.protocol=SSL
    - kafka.rest.client.ssl.key.password=${file:/mnt/sslcerts/jksPassword.txt:jksPassword}
    - kafka.rest.client.ssl.keystore.password=${file:/mnt/sslcerts/jksPassword.txt:jksPassword}
    - kafka.rest.client.ssl.keystore.location=/mnt/sslcerts/keystore.p12
    - kafka.rest.client.ssl.truststore.location=/mnt/sslcerts/truststore.p12
    - kafka.rest.client.ssl.truststore.password=${file:/mnt/sslcerts/jksPassword.txt:jksPassword}

#ADD

annotations:
  platform.confluent.io/kraft-migration-ibp-version: "3.7"
spec:
  authorization:
    superUsers:
    - User:kafka.confluent.svc.cluster.local
    type: simple
  services:
    kafkaRest:
      listeners:
        internal:
          tls:
            enabled: false
      tls:
        enabled: false
```

2. Deploy kraft controller with below annotation. Which will keep kraft in hold stage.

```
annotations:
    platform.confluent.io/kraft-migration-hold-krc-creation: "true"
spec:
  configOverrides:
    jvm:
    - -Djava.security.auth.login.config=/mnt/secrets/digest-jaas.conf
    server:
    - zookeeper.sasl.client=true
    - zookeeper.sasl.clientconfig=Client
    - confluent.balancer.enable=true
    - confluent.balancer.heal.uneven.load.trigger=ANY_UNEVEN_LOAD
    - inter.broker.listener.name=REPLICATION
    - listener.security.protocol.map=CONTROLLER:SSL,EXTERNAL:SSL,INTERNAL:SSL,REPLICATION:SSL
    - listener.name.replication.ssl.client.auth=required
    - listener.name.replication.ssl.key.password=${file:/mnt/sslcerts/jksPassword.txt:jksPassword}
    - listener.name.replication.ssl.keystore.location=/mnt/sslcerts/keystore.p12
    - listener.name.replication.ssl.keystore.password=${file:/mnt/sslcerts/jksPassword.txt:jksPassword}
    - listener.name.replication.ssl.principal.mapping.rules=RULE:.*CN\s?=([a-zA-Z0-9.\-]*)?.*/$1/
    - listener.name.replication.ssl.truststore.location=/mnt/sslcerts/truststore.p12
    - listener.name.replication.ssl.truststore.password=${file:/mnt/sslcerts/jksPassword.txt:jksPassword}
  authorization:
    superUsers:
    - User:kafka.confluent.svc.cluster.local
    type: simple
```

3. Deploy kraft migration job. Kafka will take a RESTART twice, one for updating IBP version (IBP will get upgraded to the version mentioned in the annotation) and another for updating kraft configuration.
4. Monitor the migration job. Wait for the migration job to reach the DUAL WRITE phase. Check using below command.

   kubectl describe kmj <job-name> -n <namespace>

5. Apply below annotation to move to kraft only mode.

   kubectl annotate kraftmigrationjob kraft-migration platform.confluent.io/kraft-migration-trigger-finalize-to-kraft=true

6. Kafka will take a RESTART.
7. Kraft will take a RESTART.
8. Run below command to check for the migration job to reach the COMPLETE stage.

   kubectl describe kmj <job-name> -n <namespace>

9. Apply below annotation

   kubectl annotate kmj kraft-migration -n kraft platform.confluent.io/kraft-migration-release-cr-lock=true --overwrite

10. Download updated kafka and kraft CR.
11. Remove Zookeeper manually.

    kubectl delete zookeeper <zookeeper-cr-name> -n <namespace>

Steps to perform to rollback to Zookeeper only mode

Note: These steps can only be performed while in DUAL WRITE PHASE. Once moved to kraft only mode, then there is no going back.

1. Run the below command to check if the kraft migration job has reached the DUAL WRITE PHASE.

   kubectl describe kmj <job-name> -n <namespace>
2. Run the below command to annotate the migration job to move to ZK mode. Wait until kafka completes a **RESTART**.

   **kubectl annotate kraftmigrationjob <job-name> **
   **platform.confluent.io/kraft-migration-trigger-rollback-to-zk=true --overwrite **
   --namespace <namespace>
3. Run the below command to check if the kraft migration job has reached the **SubPhaseRollbackToZkWaitForManualNodeRemovalFromZk** SUBPHASE.

   kubectl describe kmj <job-name> -n <namespace>

4. Run the below commands to move inside one of the zookeeper POD and delete the nodes manually.

   a) kubectl exec -it zookeeper-0 bash

   b) cat > /tmp/adm.properties << EOF
   zookeeper.connect=zookeeper.confluent.svc.cluster.local:2182
   zookeeper.ssl.client.enable=true
   zookeeper.clientCnxnSocket=org.apache.zookeeper.ClientCnxnSocketNetty
   zookeeper.ssl.truststore.location=/mnt/sslcerts/truststore.p12
   zookeeper.ssl.truststore.password=<password>
   zookeeper.ssl.keystore.location=/mnt/sslcerts/keystore.p12
   zookeeper.ssl.keystore.password=<password>
   zookeeper.set.acl=true
   zookeeper.ssl.keystore.type=PKCS12
   zookeeper.ssl.truststore.type=PKCS12
   EOF

   c) cat > zk-client-jaas.conf << EOF
   Client {
     org.apache.zookeeper.server.auth.DigestLoginModule required
     username="kafka"
     password="<password>";
   };
   EOF

   d) JAVA_TOOL_OPTIONS="-Djava.security.auth.login.config=/tmp/zk-client-jaas.conf" zookeeper-shell zookeeper.<namespace>.svc.cluster.local:2182 -zk-tls-config-file adm.properties deleteall /kafka-<namespace>/migration

   e) JAVA_TOOL_OPTIONS="-Djava.security.auth.login.config=/tmp/zk-client-jaas.conf" zookeeper-shell zookeeper.<namespace>.svc.cluster.local:2182 -zk-tls-config-file adm.properties deleteall /kafka-<namespace>/controller

   f) JAVA_TOOL_OPTIONS="-Djava.security.auth.login.config=/tmp/zk-client-jaas.conf" zookeeper-shell zookeeper.<namespace>.svc.cluster.local:2182 -zk-tls-config-file adm.properties deleteall /kafka-<namespace>/controller_epoch

5. Come out of the zookeeper POD to run the below command manually

   **kubectl annotate kraftmigrationjob kraft-migration **
   **platform.confluent.io/continue-kraft-migration-post-zk-node-removal=true **
   **--overwrite **
   --namespace <namespace>

6. Kafka will RESTART thrice, until the kraft migration job reaches the **SubPhaseRollbackToZkComplete** SUBPHASE. Which indicates that the rollback to ZK has been completed. To check this run the below command

   kubectl describe kmj <job-name> -n <namespace>

7. Remove the kraft controller manually running the below command

   kubectl delete kraft <kraft-cr-name> -n <namespace>

CONFLUENT REFERENCE LINKS

https://docs.confluent.io/platform/current/security/authorization/acls/overview.html#zk-based-ak-clusters
https://docs.confluent.io/platform/current/security/authorization/acls/overview.html#kraft-based-ak-clusters
https://docs.confluent.io/operator/current/co-migrate-kraft.html#issue-kraft-cannot-authenticate-to-zk-during-migration