

# Migrating Kafka from ZooKeeper to KRaft: adventures in operations

Kate Stanley  
Paolo Patierno



Paolo Patierno – [github.com/ppatierno](https://github.com/ppatierno)

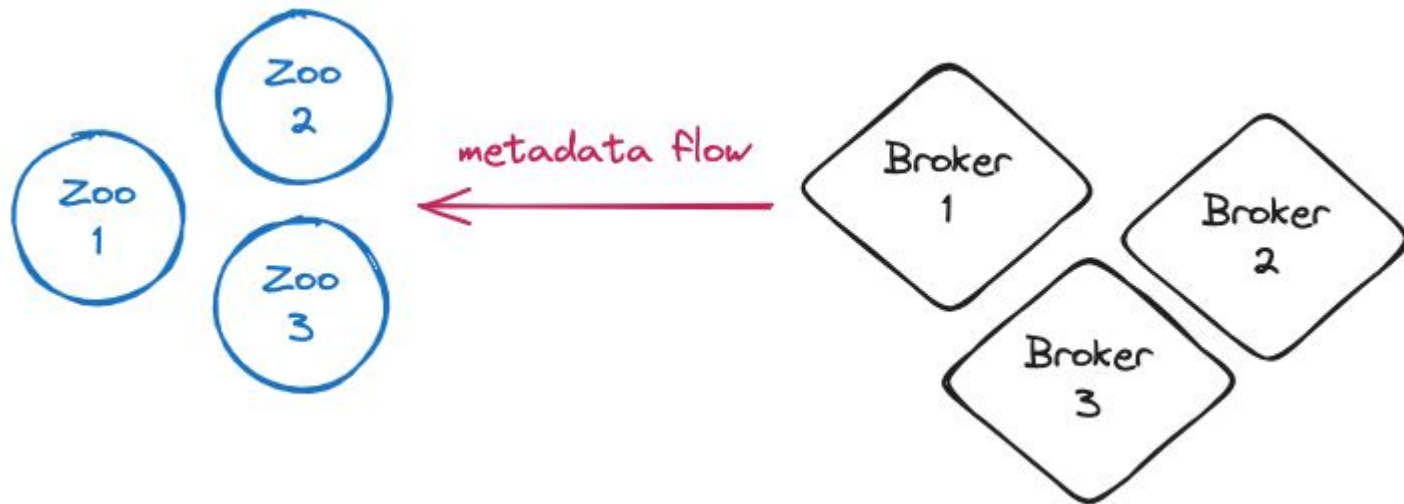
- Senior Principal Software Engineer at Red Hat
- Strimzi maintainer
- CNCF Ambassador
- Formula 1 & MotoGP addicted

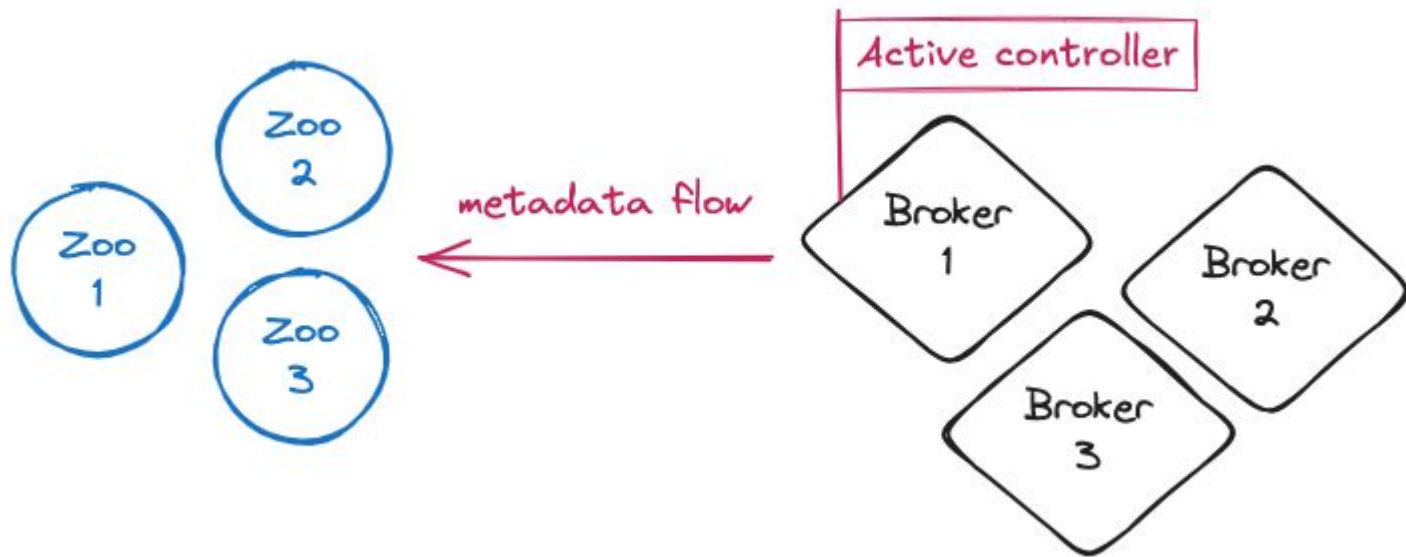


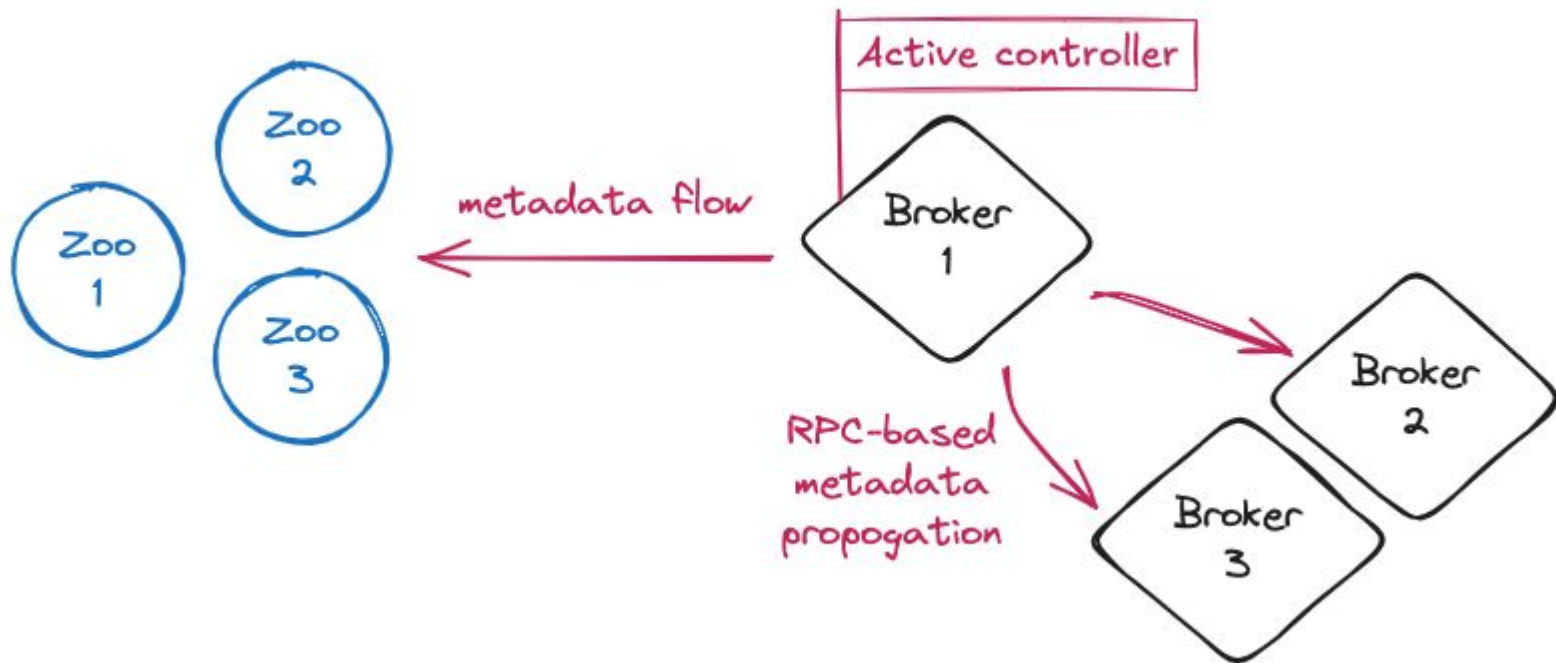
Kate Stanley – [github.com/katheris](https://github.com/katheris)

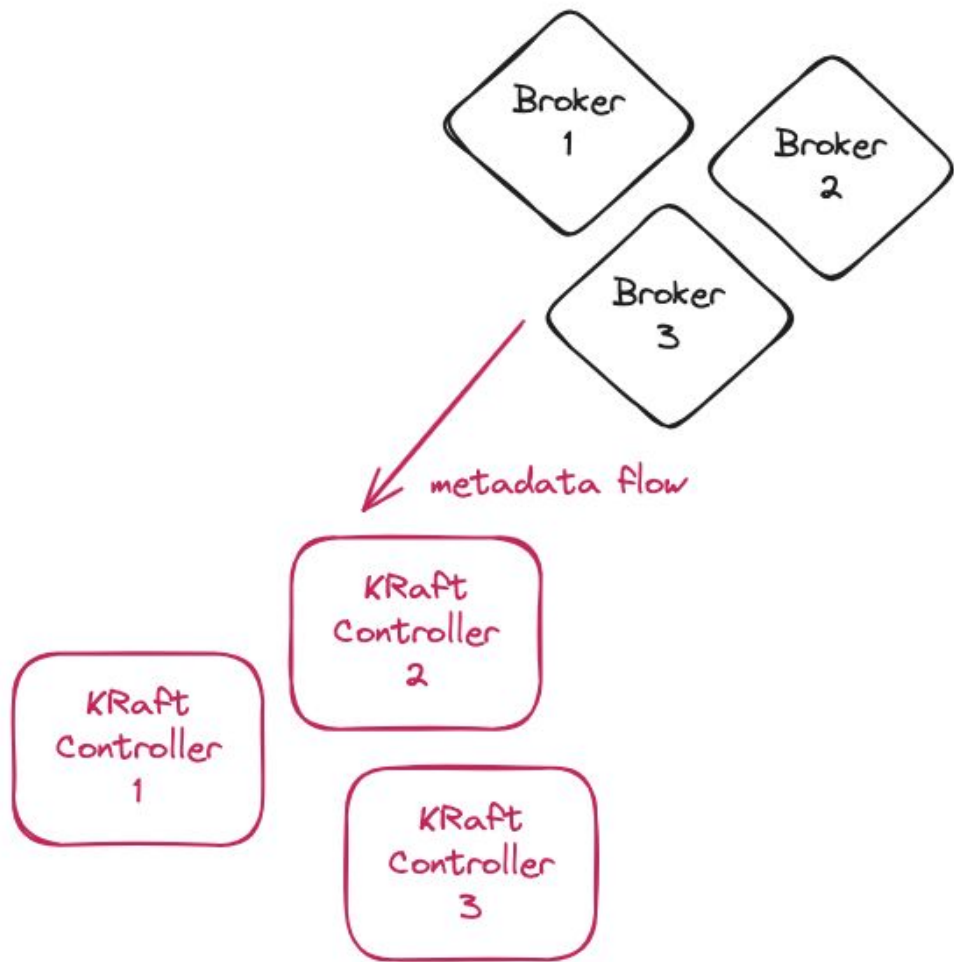
- Principal Software Engineer at Red Hat
- Strimzi maintainer
- LinkedIn Learning Presenter
- Author

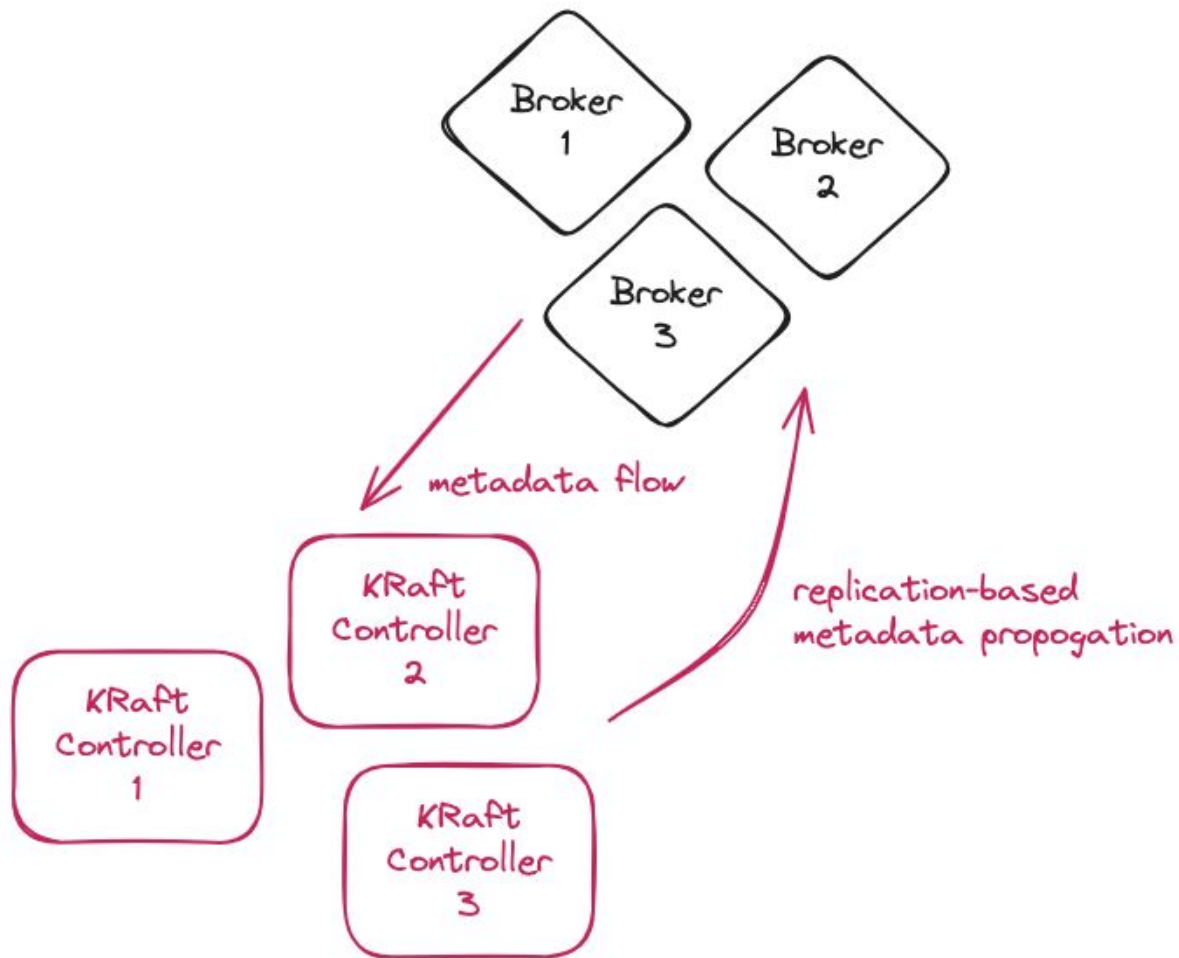
# KRaft Mode











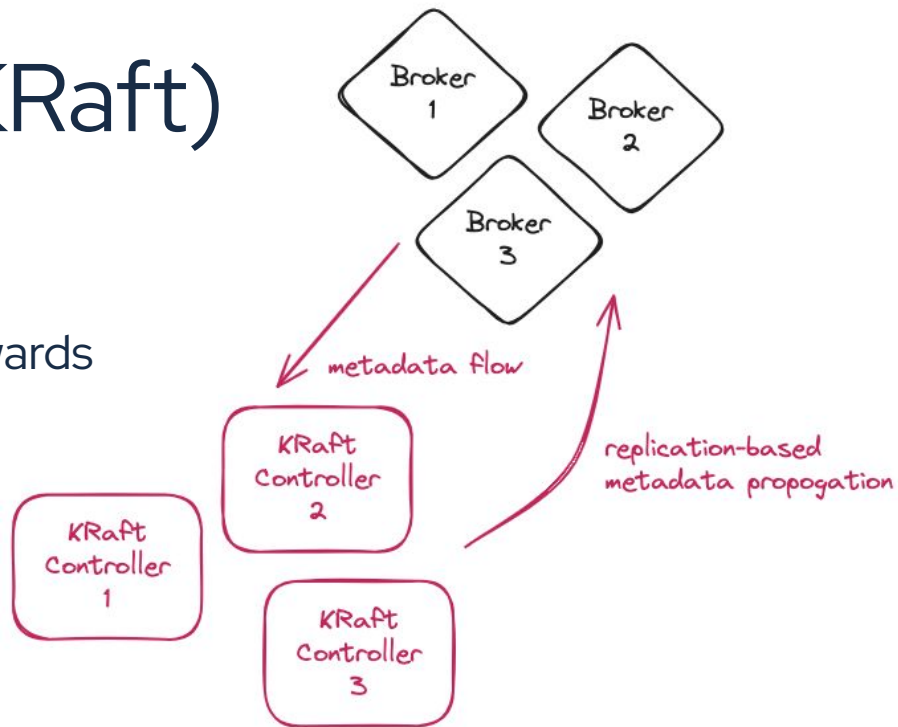


# Apache Kafka Raft (KRaft)

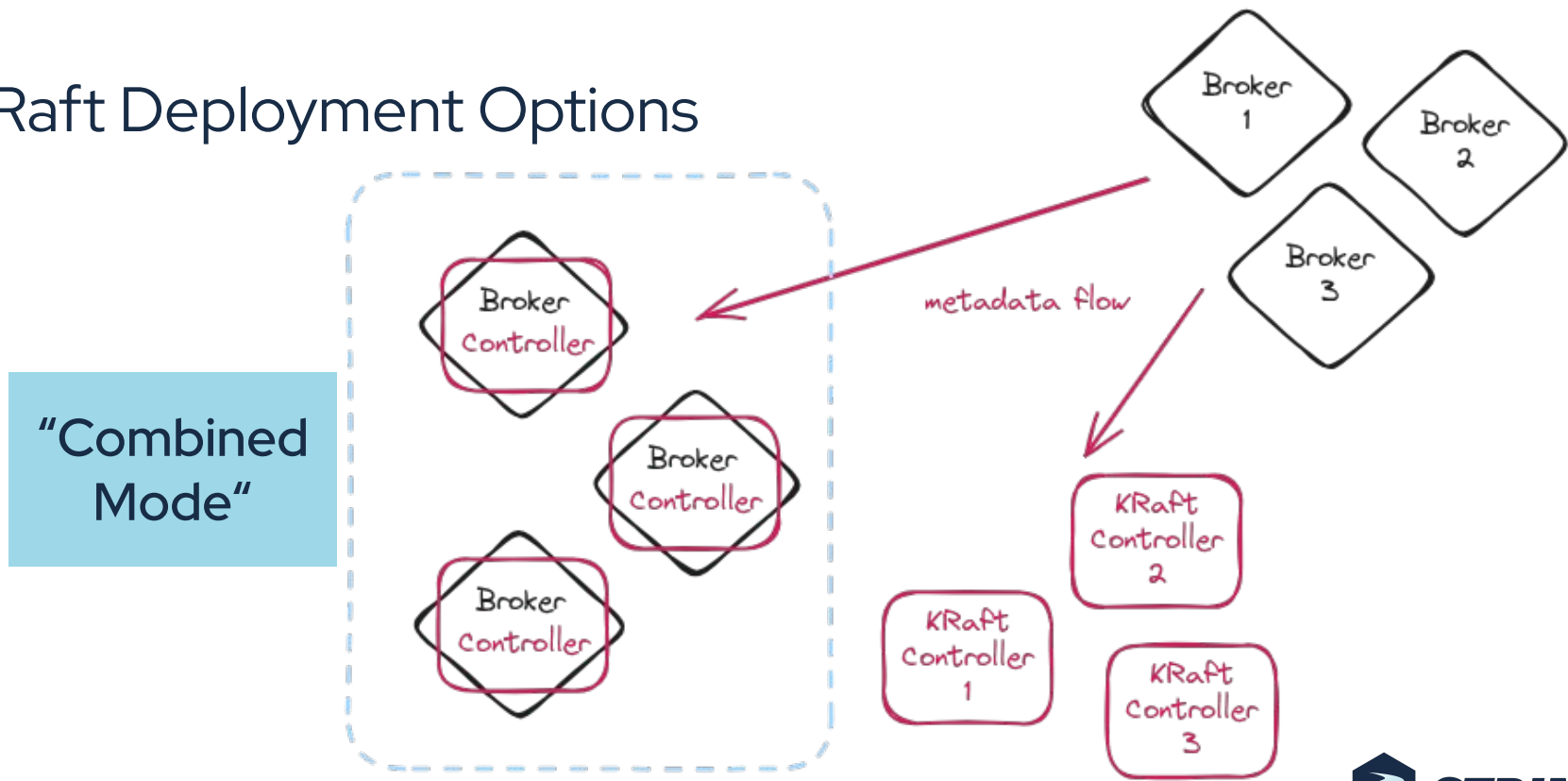
Only supported option from Kafka 4.0 onwards

ZooKeeper nodes no longer needed

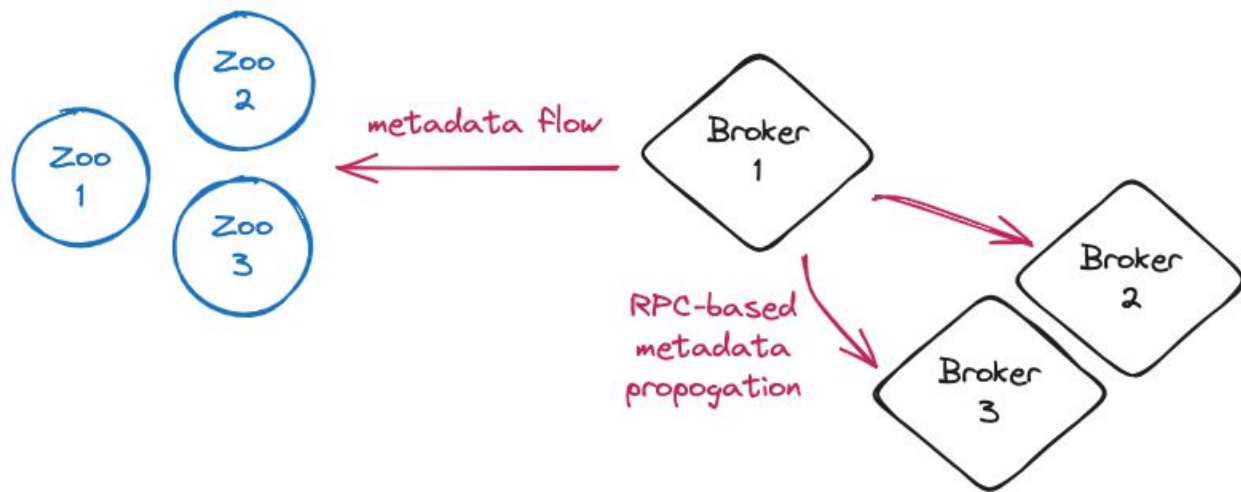
Metadata now stored in controller nodes



## KRaft Deployment Options

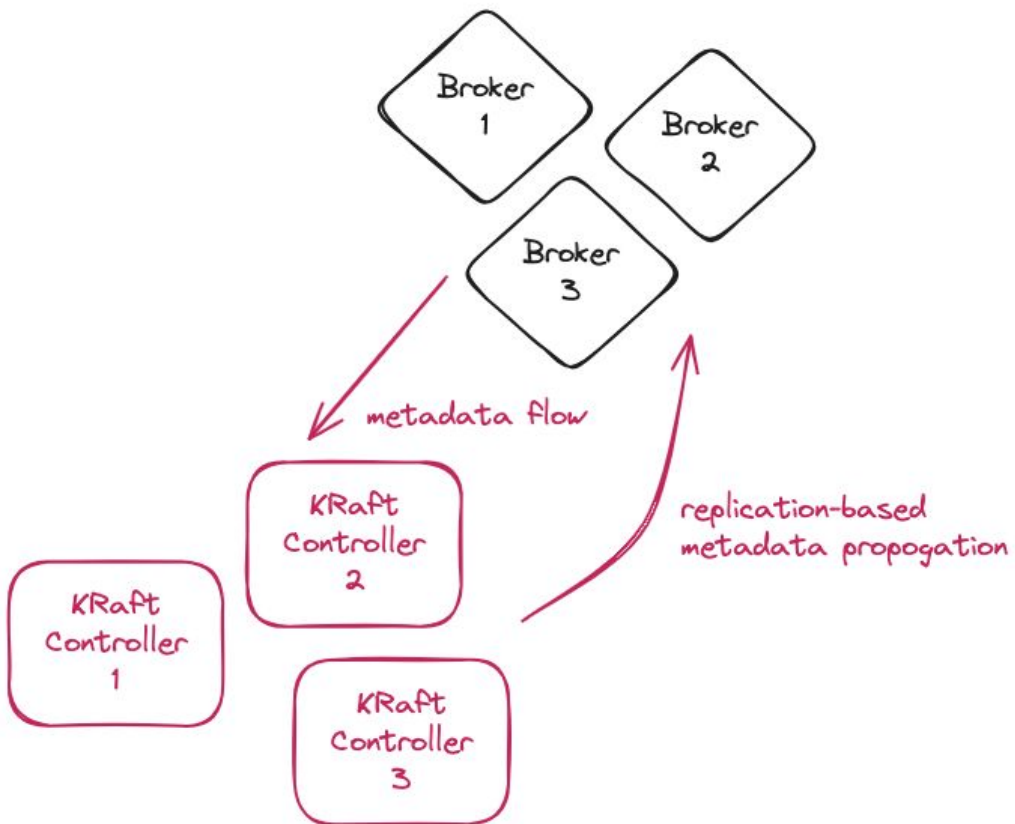


# Migration overview

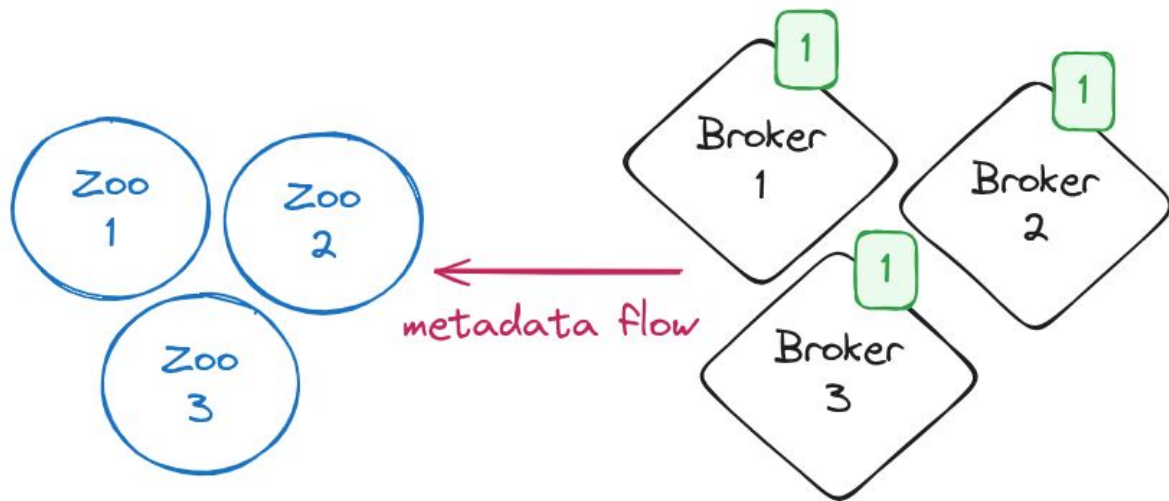


## Migration phases:

- 1 **Initial phase** - all brokers in ZK mode, and ZK-based controller
- 2 **Initial metadata load**
- 3 **Hybrid phase**
- 4 **Dual-write phase**
- 5 **Finalized** - no longer writing metadata to ZooKeeper



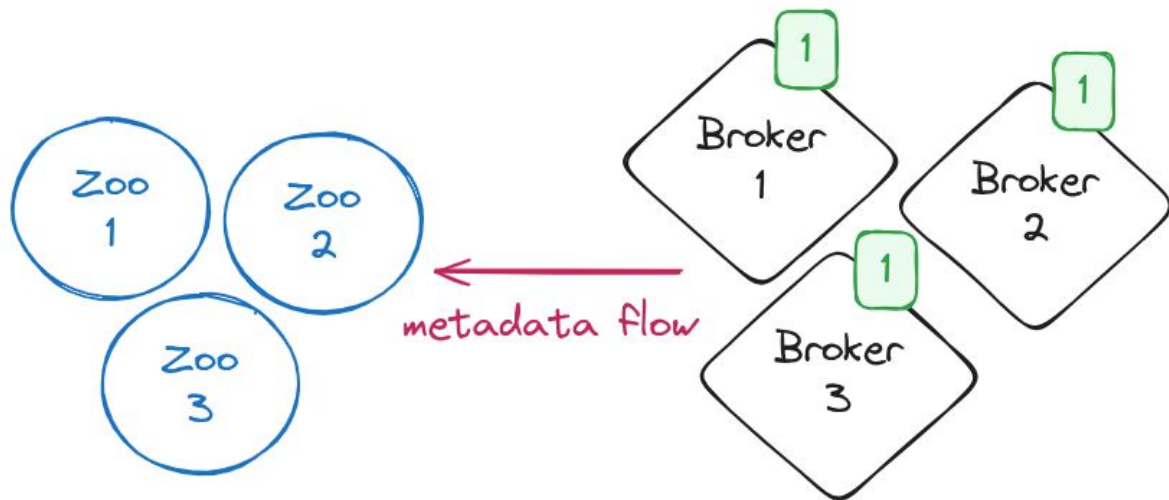
# Migration Steps: In detail



1 "generation" of the nodes to track rolling restarts

## Initial phase



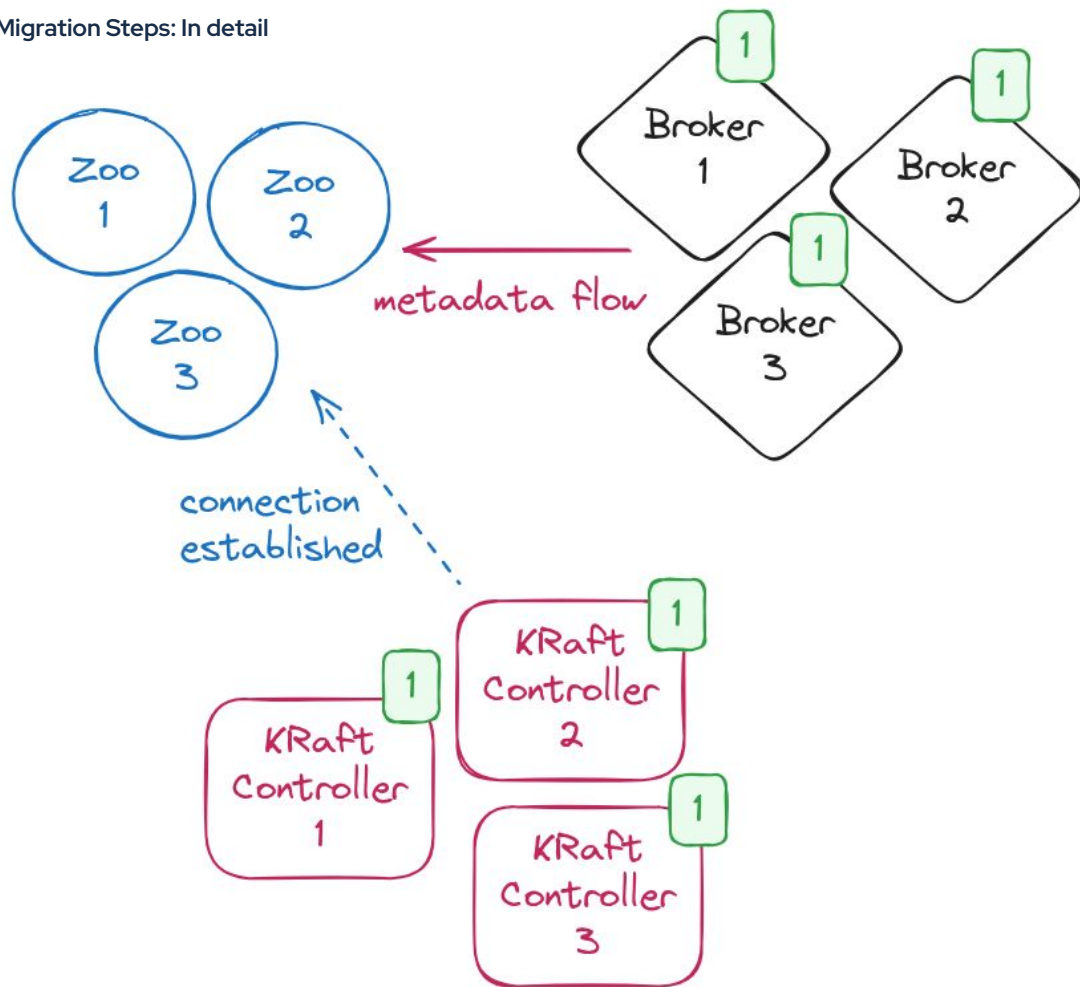


## Initial phase

### Actions:

- Create controller nodes

```
zookeeper.metadata.migration.enable=true  
zookeeper.connect=localhost:2181
```



## Initial phase

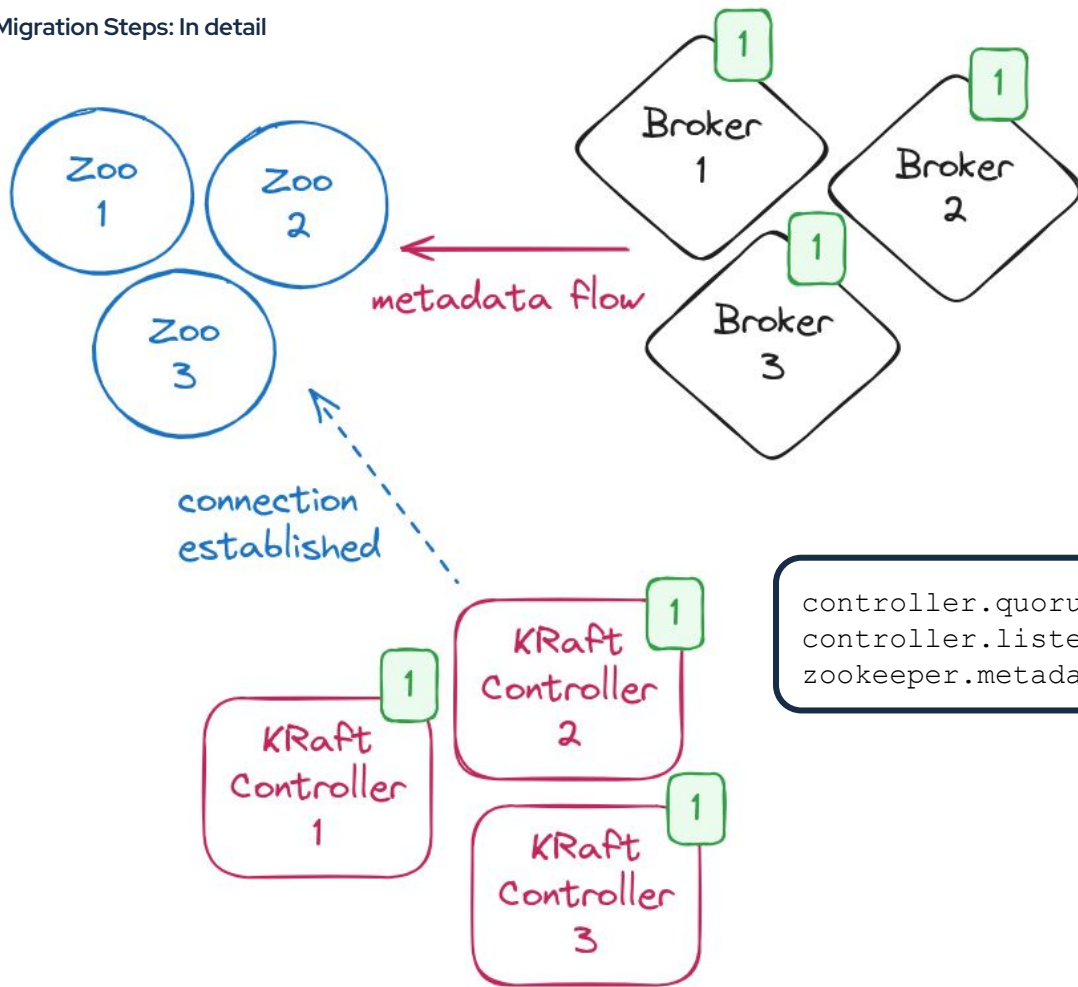
### Actions:

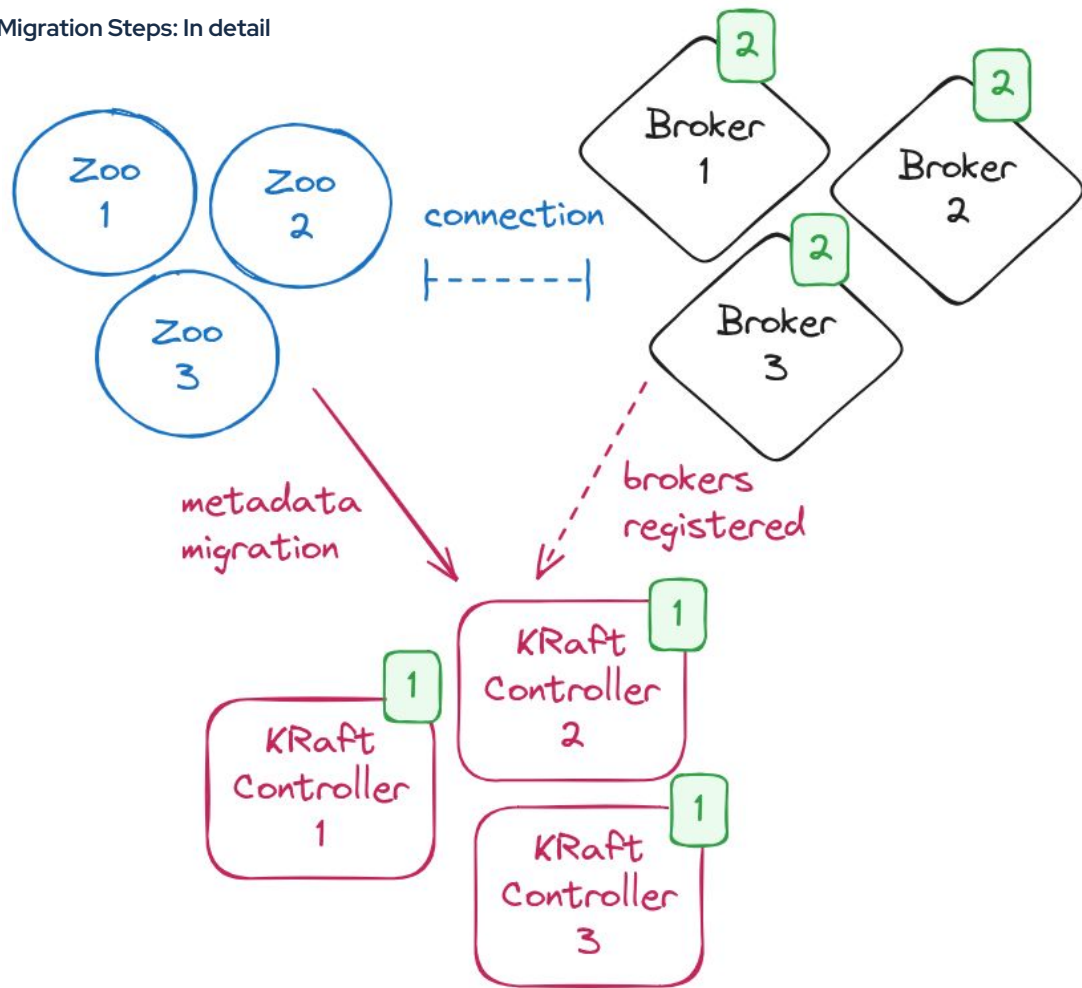
- Create controller nodes

## Initial phase

### Actions:

- Update broker configuration
- Roll Kafka pods

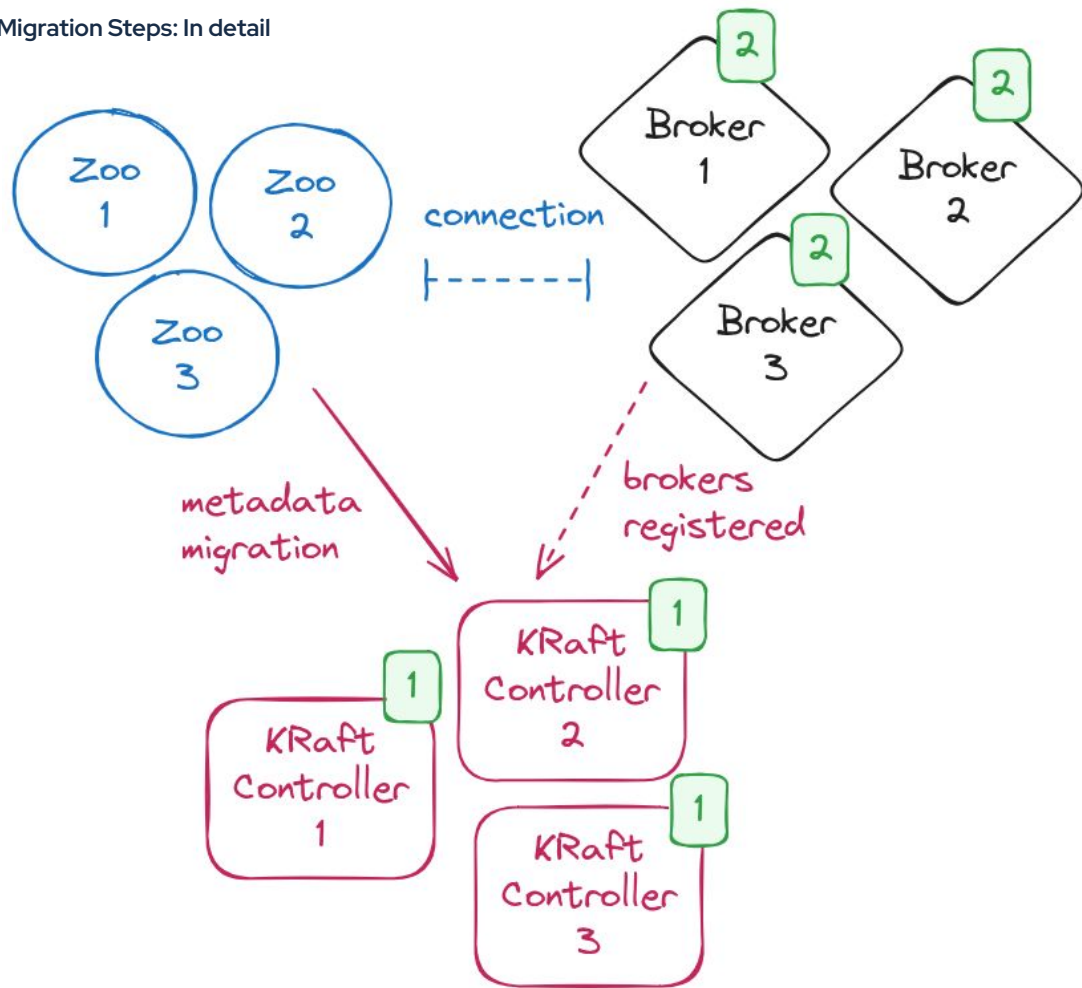




## Initial metadata load

### Actions:

- Update broker configuration
- Roll Kafka pods

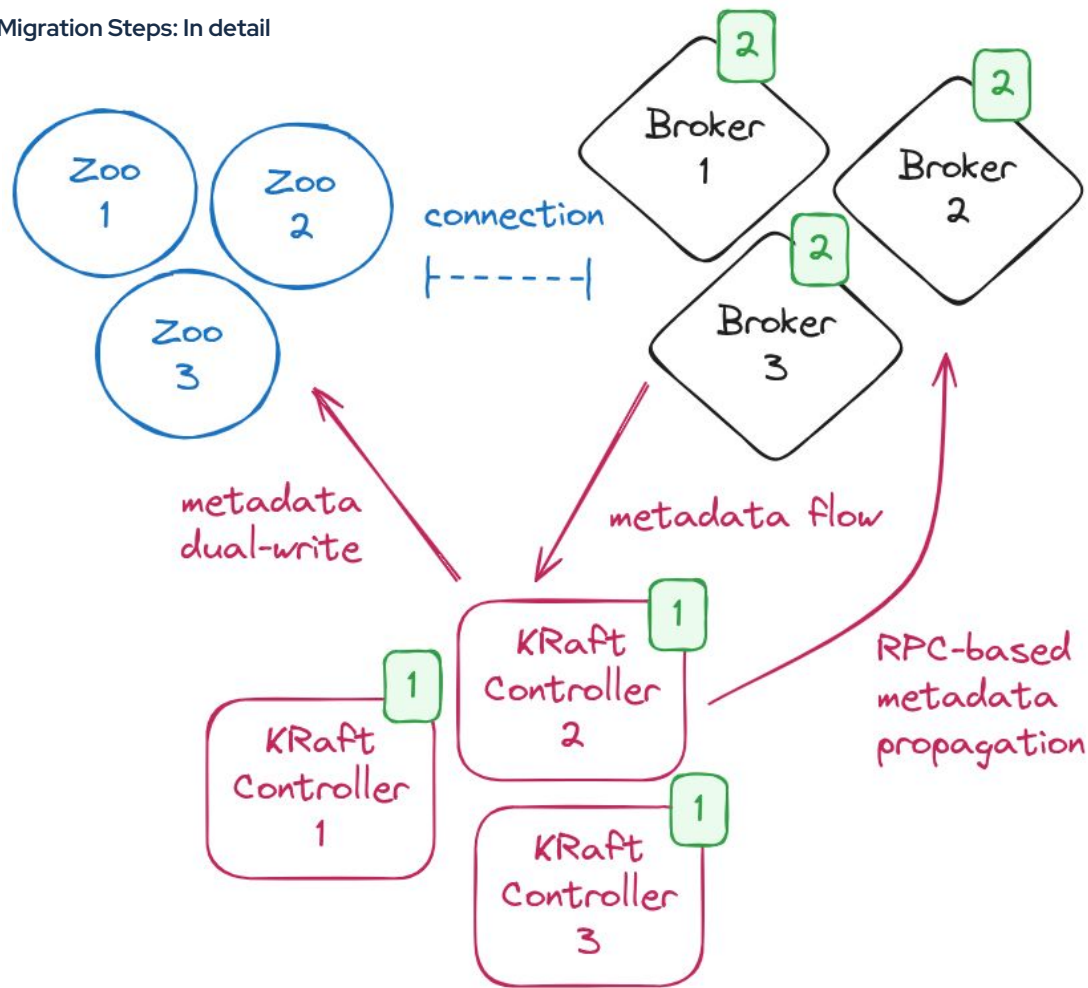


## Initial metadata load

### Actions:

- Wait for migration

Completed migration of metadata from Zookeeper to KRaft



## Hybrid mode Dual-write mode

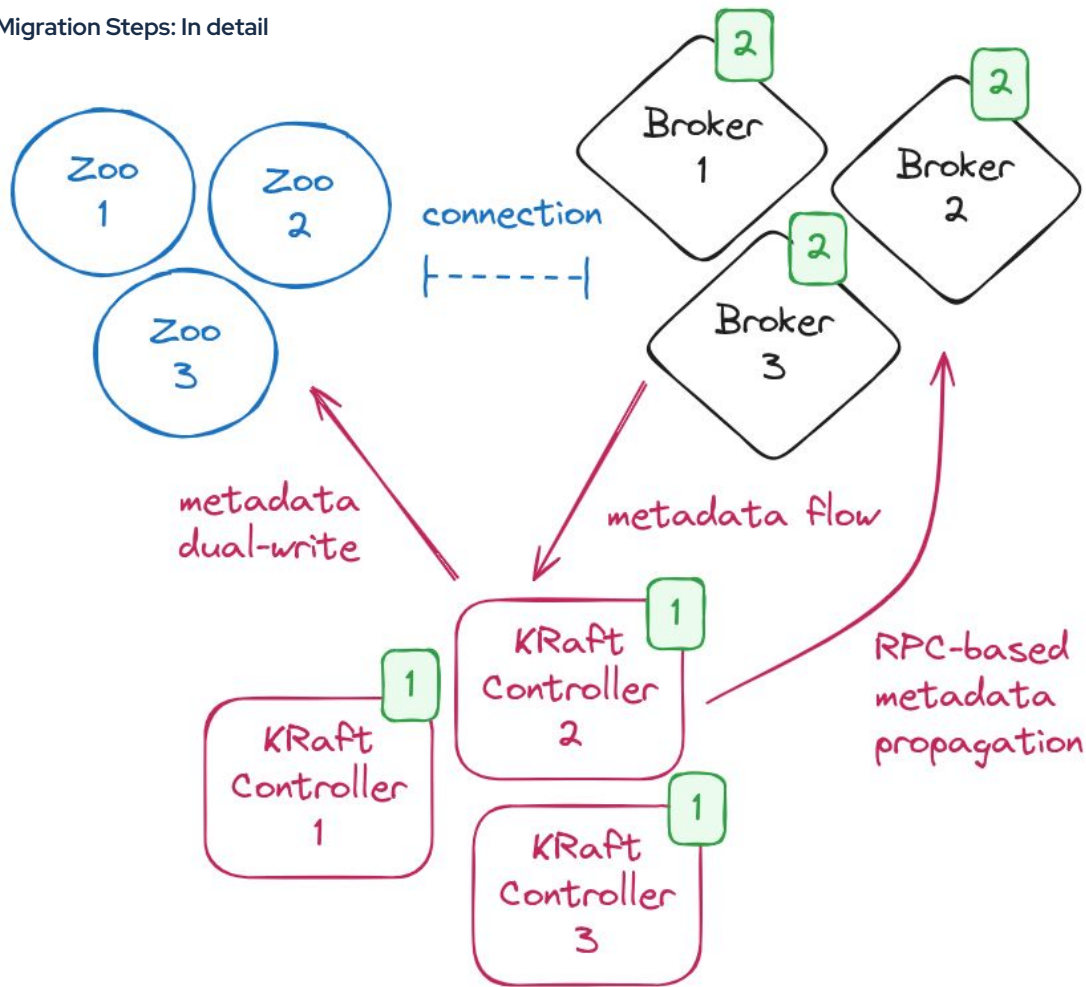
### Actions:

- Wait for migration

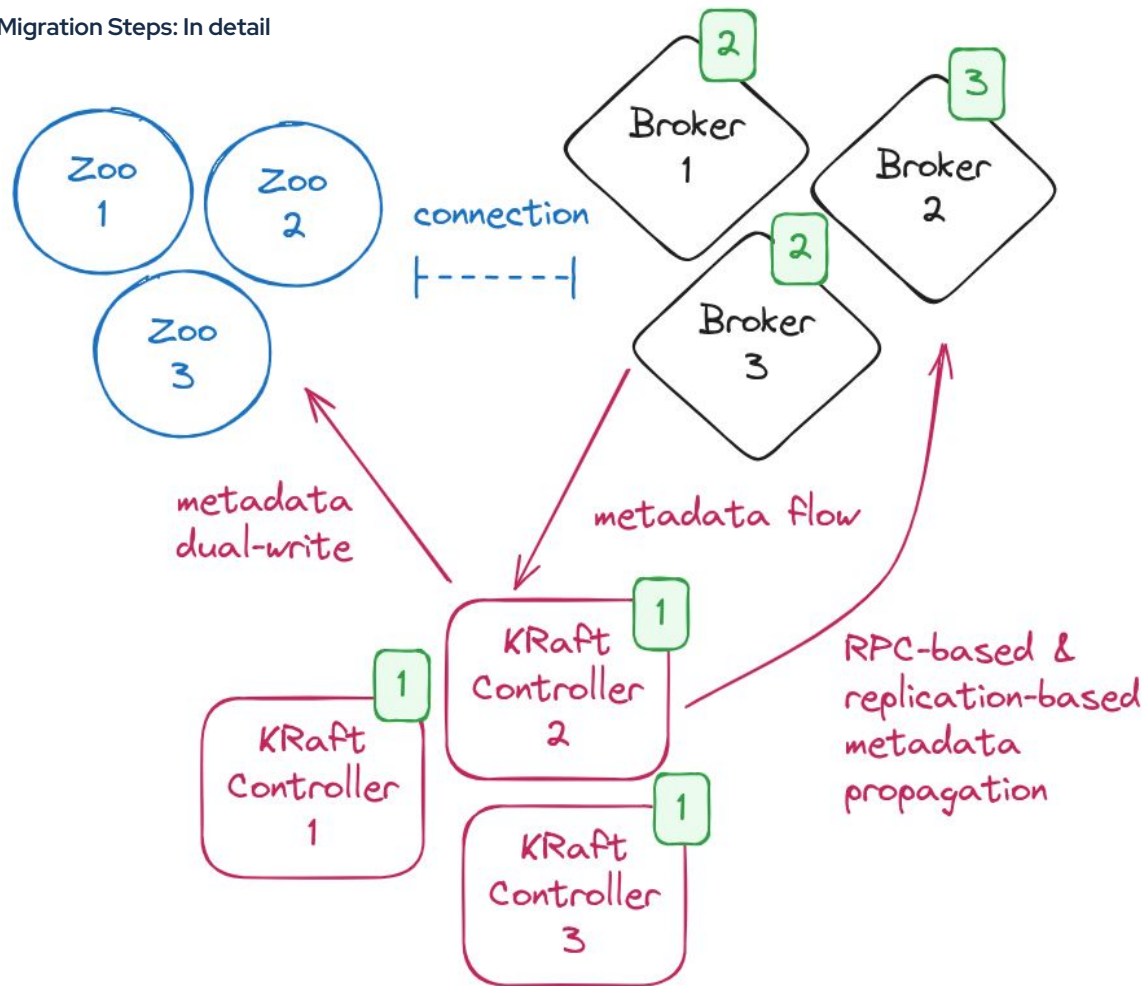
## Hybrid mode Dual-write mode

### Actions:

- Update brokers to remove ZooKeeper connection
- Disable migration flag





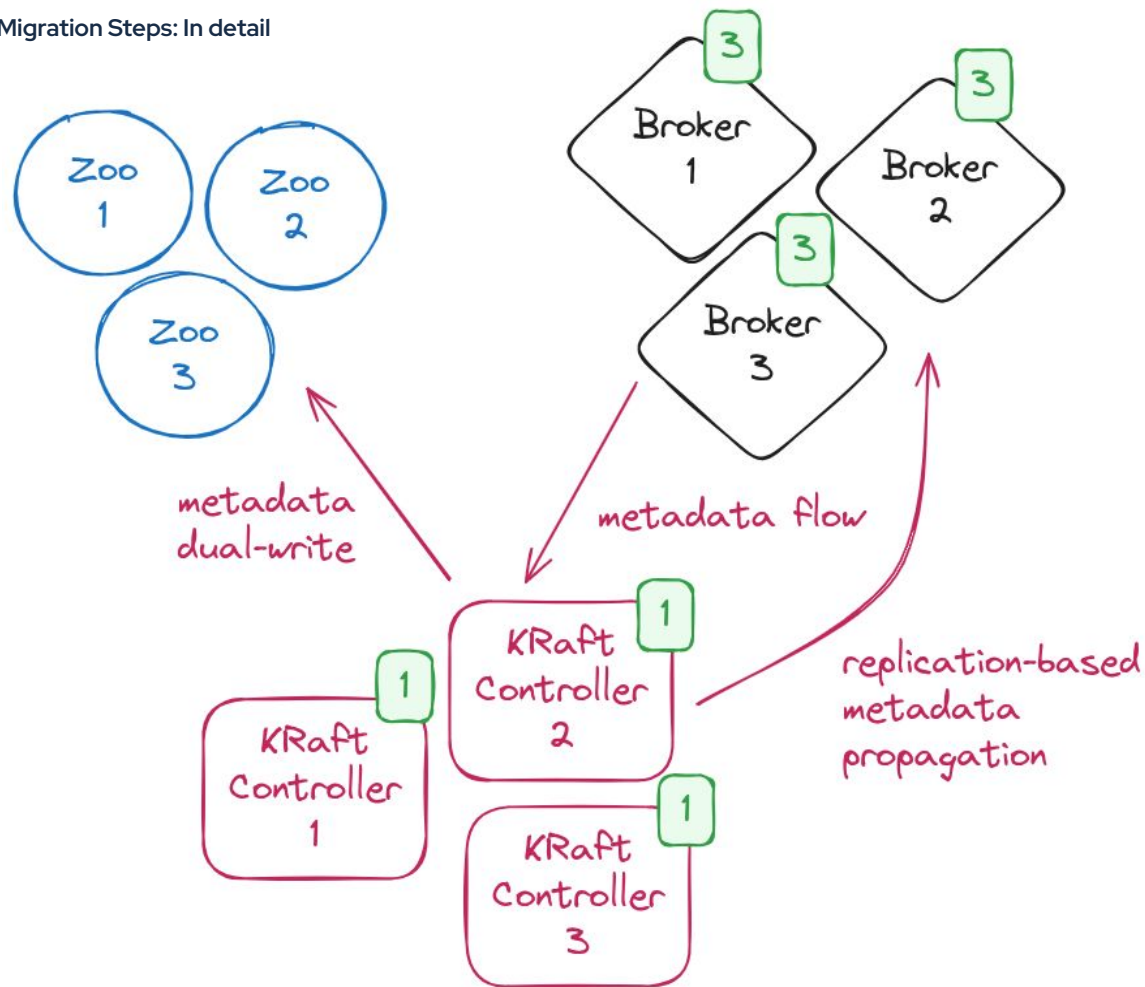


## Hybrid mode Dual-write mode

### Actions:

- Update brokers to remove ZooKeeper connection
- Disable migration flag





## Dual-write mode

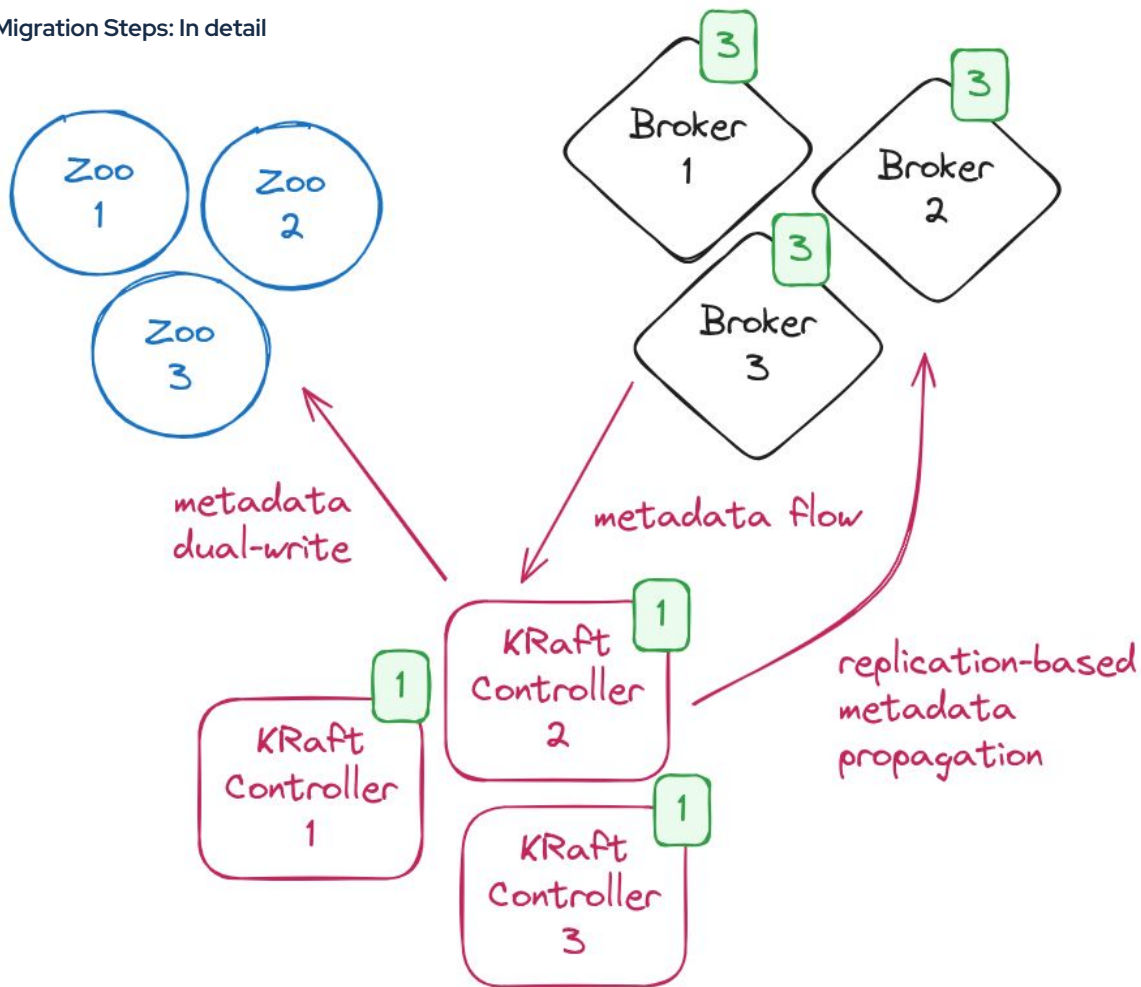
### Actions:

- Update brokers to remove ZooKeeper connection
- Disable migration flag

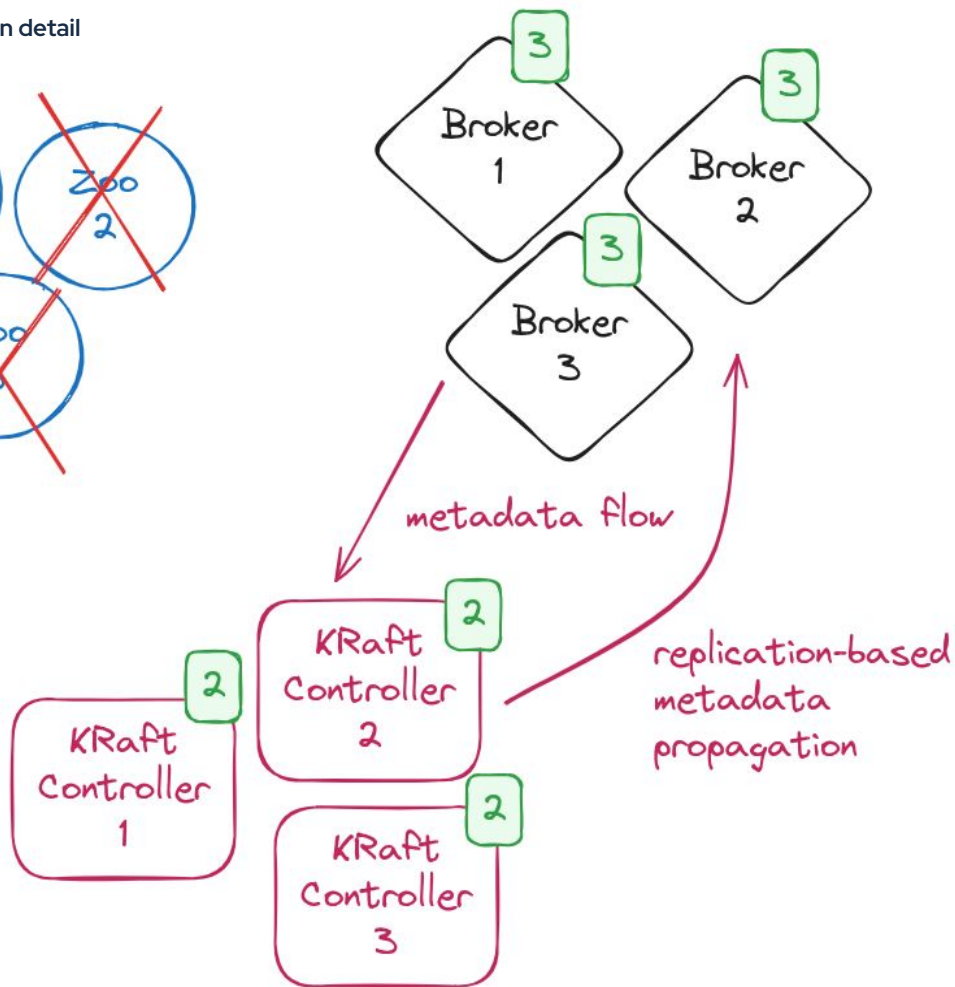
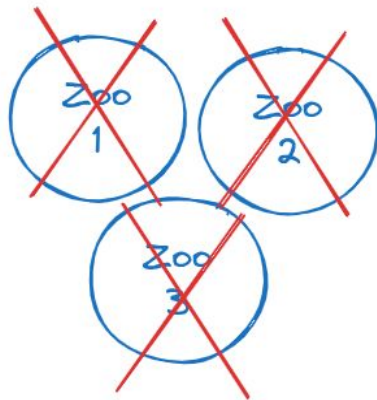
## Dual-write mode

### Actions:

- Update controllers to remove ZooKeeper connection
- Disable migration flag



## Migration Steps: In detail



## Finalized

### Actions:

- Update controllers to remove ZooKeeper connection
- Disable migration flag
- Remove ZooKeeper

## Migration phases:

- 1 **Initial phase** - all brokers in ZK mode, and ZK-based controller
- 2 **Initial metadata load** - KRaft quorum loads metadata from ZooKeeper
- 3 **Hybrid phase** - some brokers in ZK mode, but there is a KRaft controller
- 4 **Dual-write phase** - all brokers are KRaft, but KRaft controller still to writing to ZK
- 5 **Finalized** - no longer writing metadata to ZooKeeper

## What have we learnt?

Migration is not straight forward, lots of steps to do...

## What have we learnt?

Migration is not straight forward, lots of steps to do...

- Updating configuration
- Rolling nodes
- Waiting for migration to complete

## What have we learnt?

Migration is not straight forward, lots of steps to do...

- Updating configuration
- Rolling nodes
- Waiting for migration to complete

...but it should be automatable!

# Strimzi



# Strimzi

Open Source project (Apache License 2.0)

Focuses on Apache Kafka on Kubernetes

CNCF Incubating Project

“...considered stable and used  
successfully in production environments”



[strimzi.io](https://strimzi.io)

Strimzi is used by



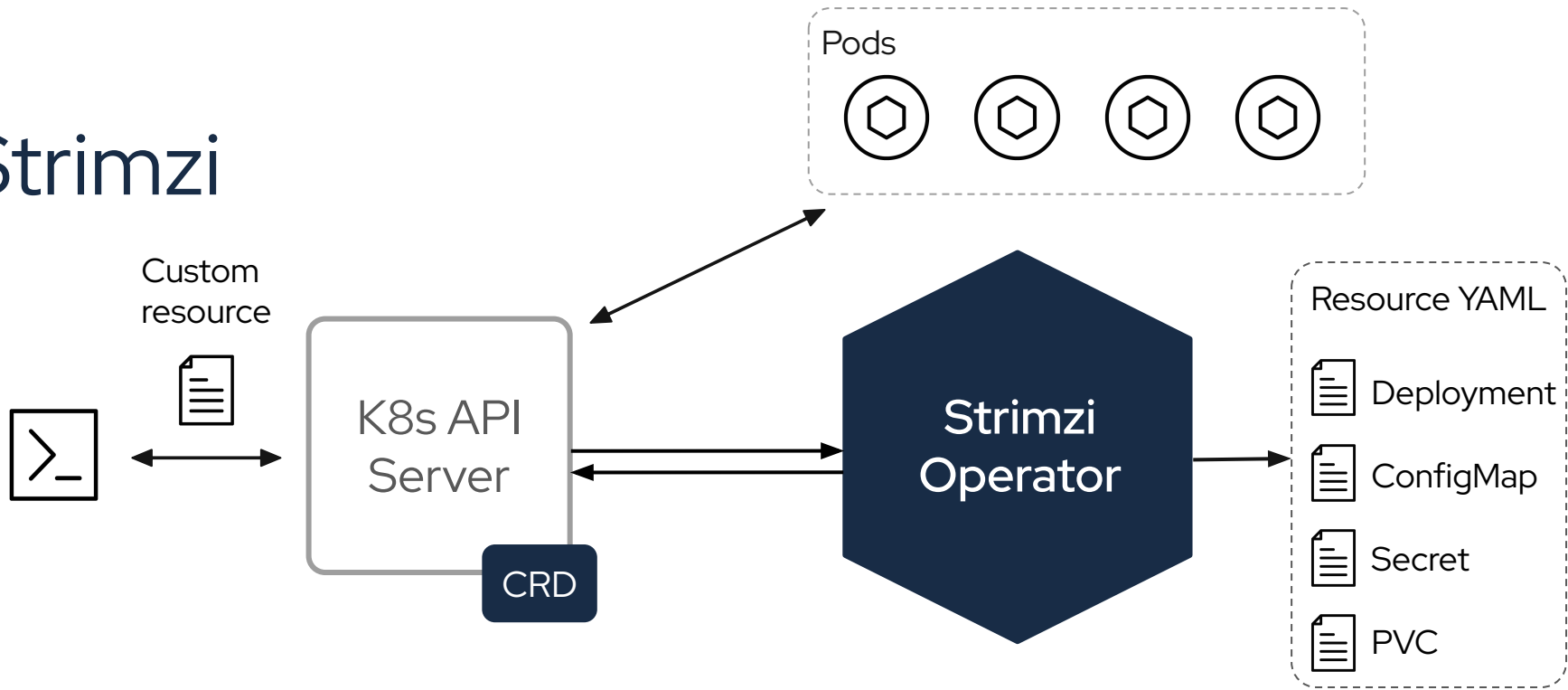
GRUPOMASMOVIL



COZYETACK



# Strimzi



```
apiVersion: kafka.strimzi.io/v1beta2
kind: Kafka
metadata:
  name: my-cluster
  annotations:
    strimzi.io/node-pools: enabled
    strimzi.io/kraft: enabled
spec:
  kafka:
    version: 4.0.0
    metadataVersion: 4.0-IV3
    listeners:
      - name: plain
        port: 9092
        type: internal
        tls: false
      - name: tls
        port: 9093
        type: internal
        tls: true
    config:
      offsets.topic.replication.factor: 3
      default.replication.factor: 3
      min.insync.replicas: 2
      # ...
```

```
apiVersion: kafka.strimzi.io/v1beta2
kind: KafkaNodePool
metadata:
  name: nodes
  labels:
    strimzi.io/cluster: my-cluster
spec:
  replicas: 3
  roles:
    - controller
    - broker
  storage:
    type: jbod
  volumes:
    - id: 0
      type: persistent-claim
      size: 100Gi
      kraftMetadata: shared
      deleteClaim: false
```

# Automating Migration

## The Plan

User takes as few actions as possible:

- Trigger migration
- Rollback (if needed)
- Finalize

## To-Do List

- ☐ Updating configuration on brokers and controllers
- ☐ Rolling nodes and waiting for readiness
- ☐ Waiting for migration to complete

## To-Do List

- ☐ Updating configuration on brokers and controllers
- ☐ Rolling nodes and waiting for readiness
- ☐ Waiting for migration to complete
- ☐ User triggers
- ☐ Tracking state



## To-Do List

- ☒ Updating configuration on brokers and controllers
  - ☒ Rolling nodes and waiting for readiness
  - ☐ Waiting for migration to complete
  - ☐ User triggers
  - ☐ Tracking state
- } Strimzi already handles

## Waiting for Migration

```
kafka.controller:type=KafkaController,name=ZkMigrationState
```

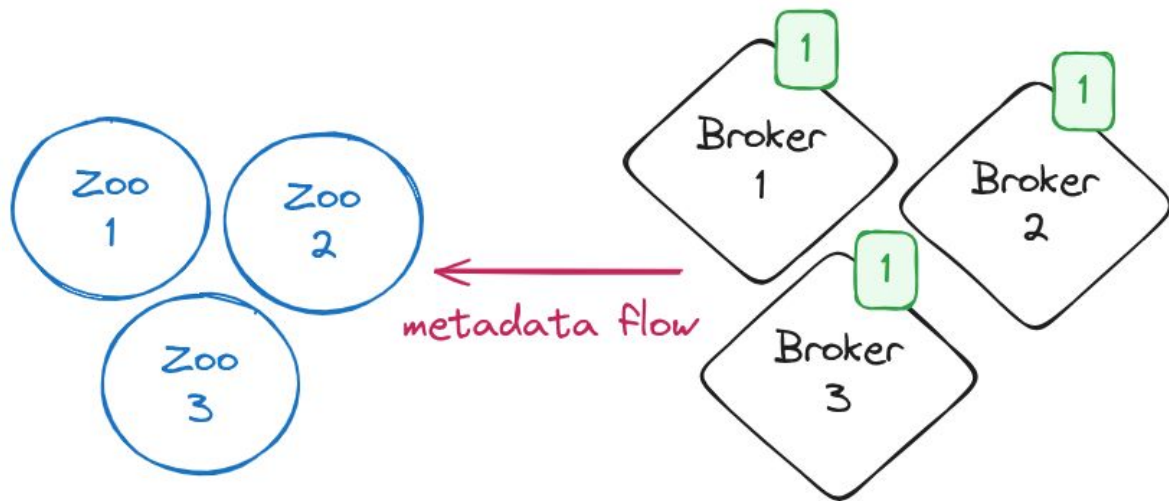
Look for state `MIGRATION`

## User Triggers

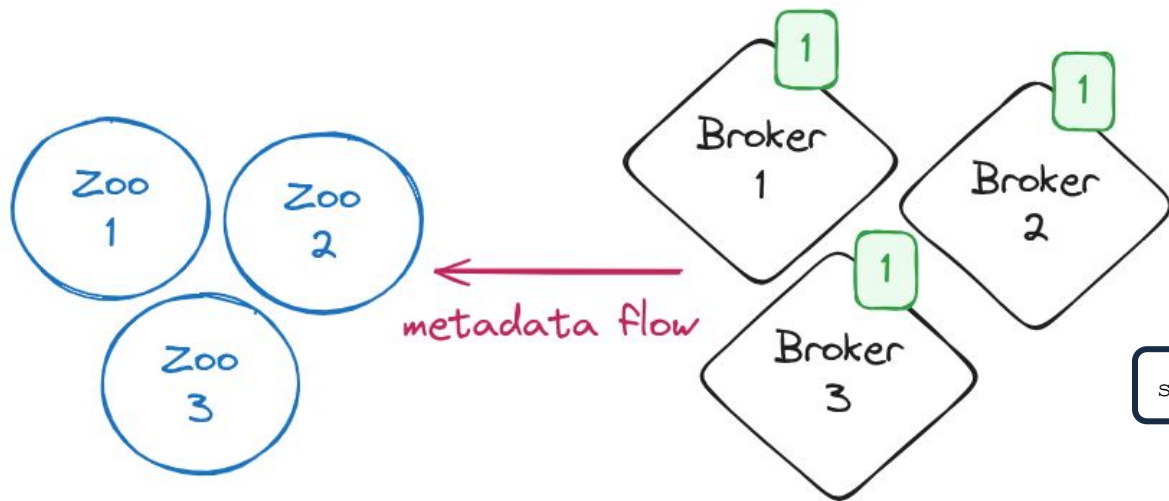
```
apiVersion: kafka.strimzi.io/v1beta2
kind: Kafka
metadata:
  name: my-cluster
  annotations:
    strimzi.io/kraft: migration
spec:
  kafka:
    # ...
```

## Storing Migration State

```
apiVersion: kafka.strimzi.io/v1beta2
kind: Kafka
metadata:
  name: my-cluster
spec:
  kafka:
    # ...
status:
  # ...
  kafkaMetadataState: KraftMigration
  kafkaMetadataVersion: 3.9-IV0
  # ...
```



1 "generation" of the nodes to track rolling restarts



## Initial phase ZooKeeper

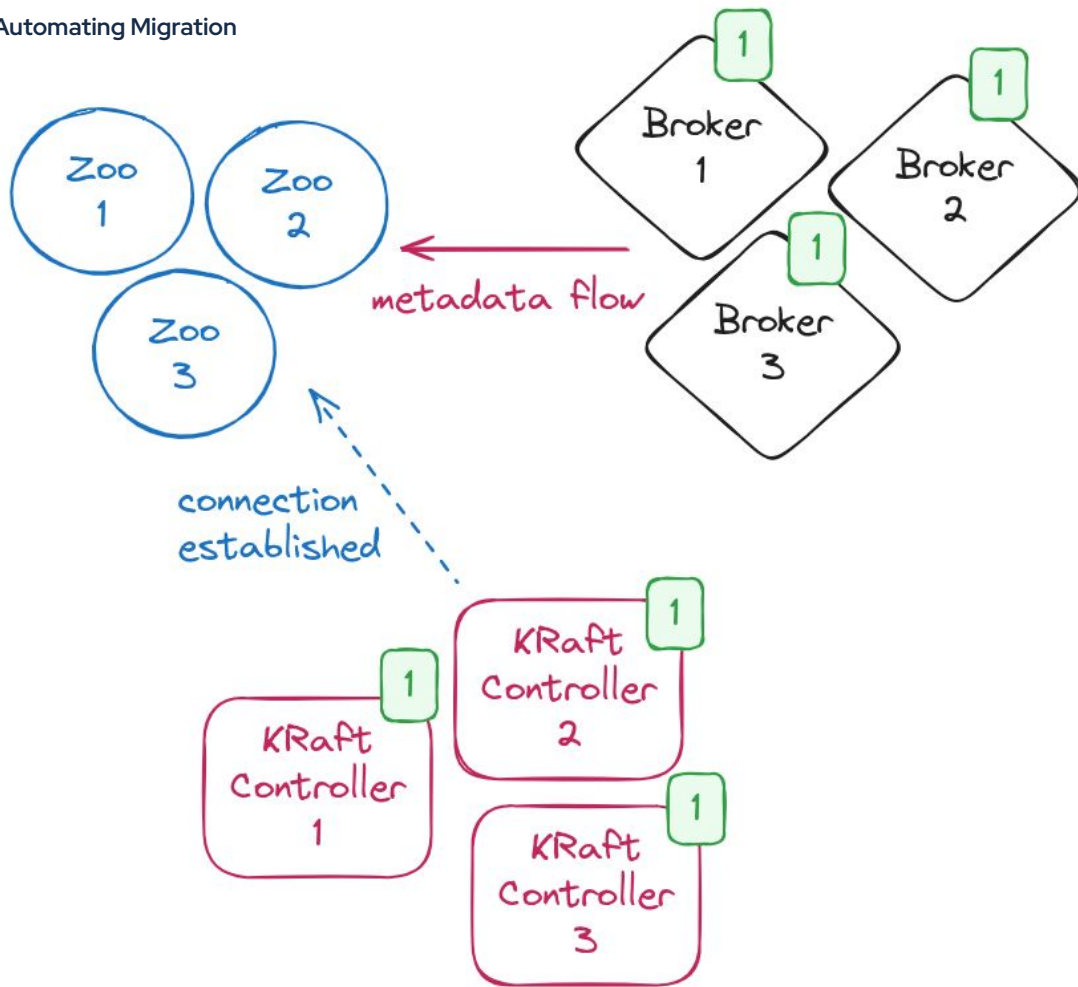
### User Actions:

- Create controller CRs
- Trigger migration

```
strimzi.io/kraft: migration
```

### Strimzi Operator Actions:

- Start controller Pods
- Update state



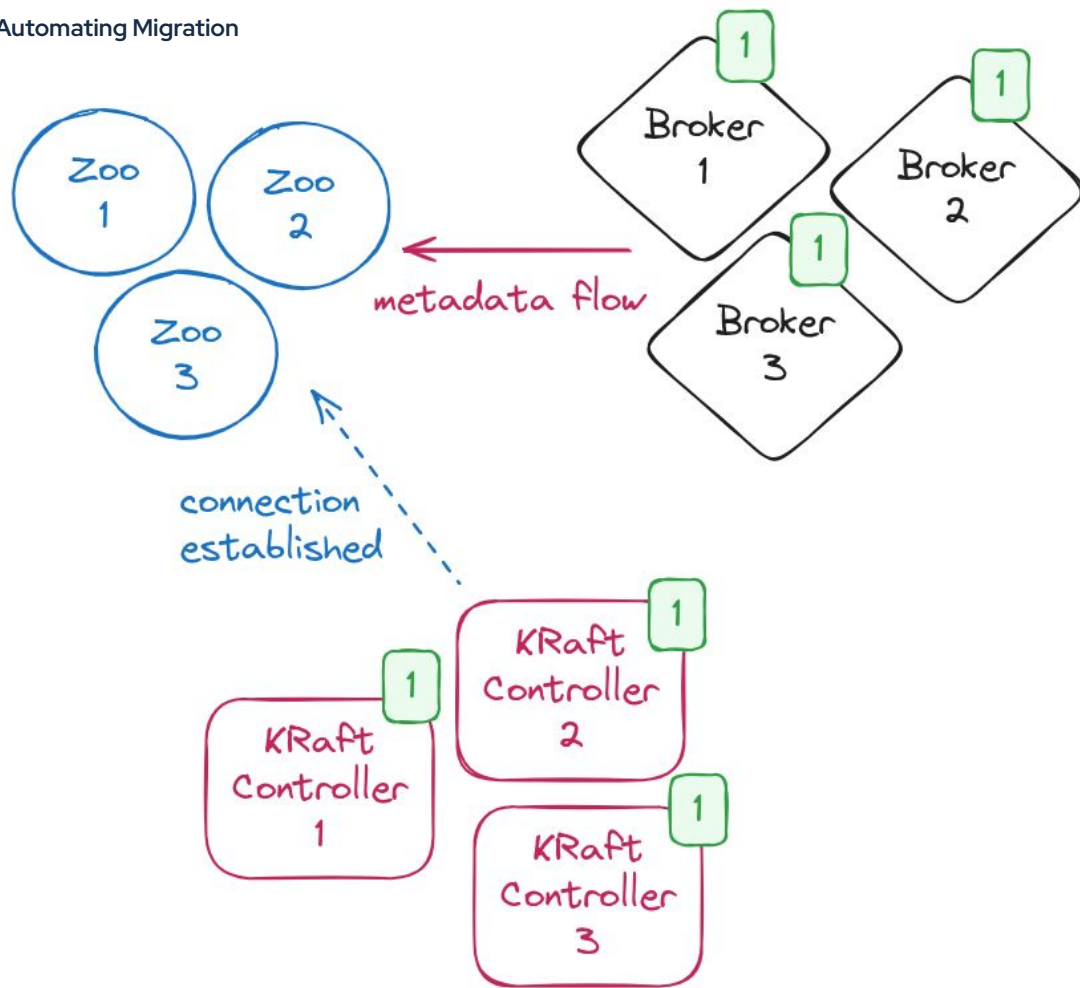
## Initial phase ZooKeeper

### User Actions:

- Create controller CRs
- Trigger migration

### Strimzi Operator Actions:

- Start controller Pods
- Update state



## Initial phase ZooKeeper

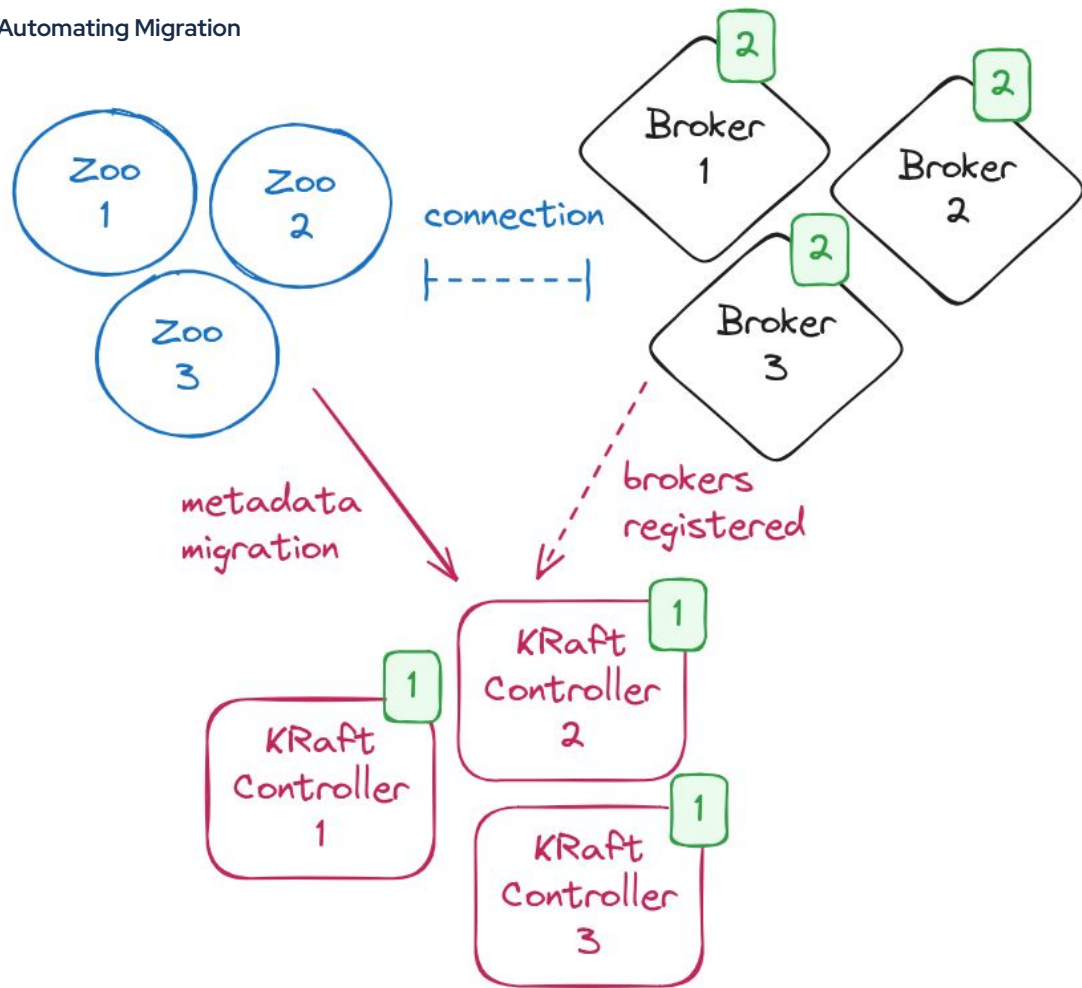
### User Actions:

- <no action>

### Strimzi Operator Actions:

- Update brokers configuration
- Roll broker Pods
- Wait for migration
- Update state





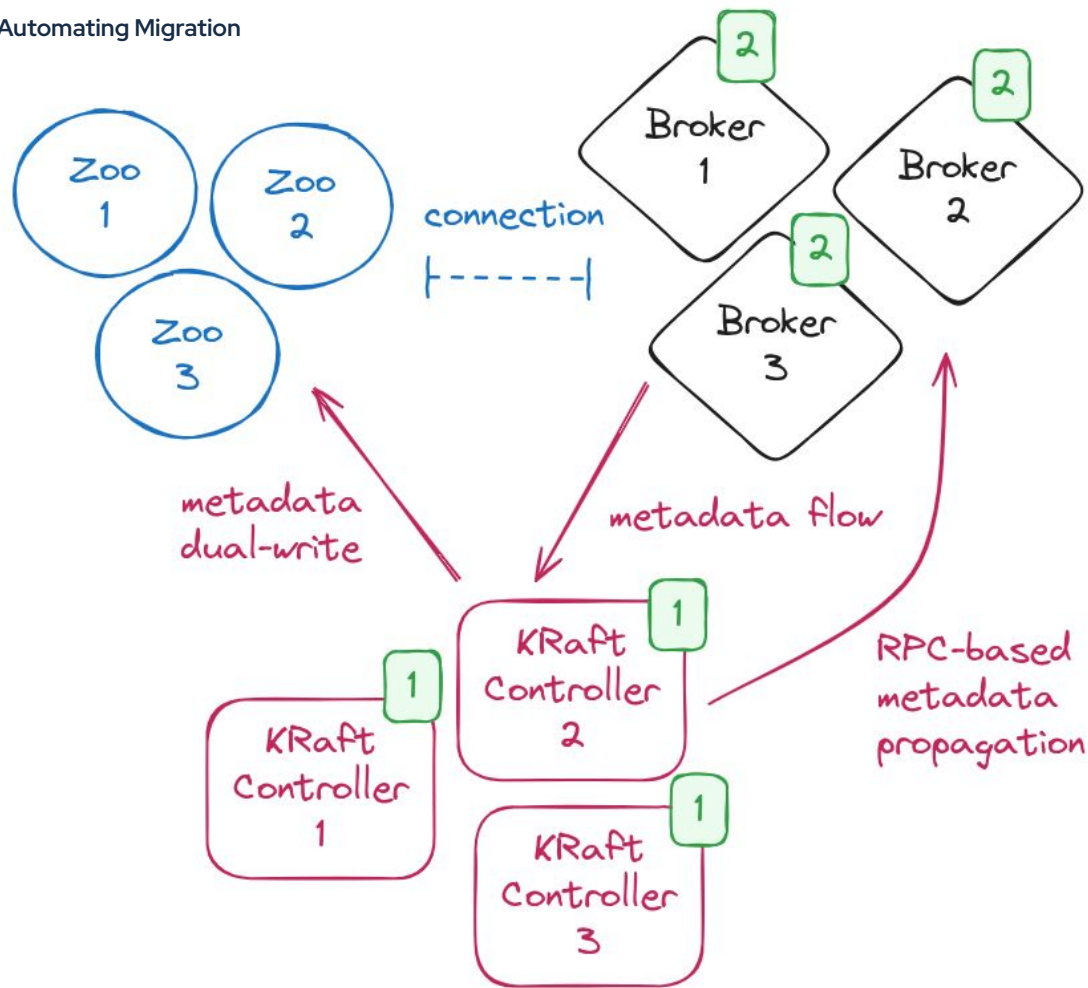
## Initial metadata load KRaftMigration

### User Actions:

- <no action>

### Strimzi Operator Actions:

- Update brokers configuration
- Roll broker Pods
- Wait for migration
- Update state



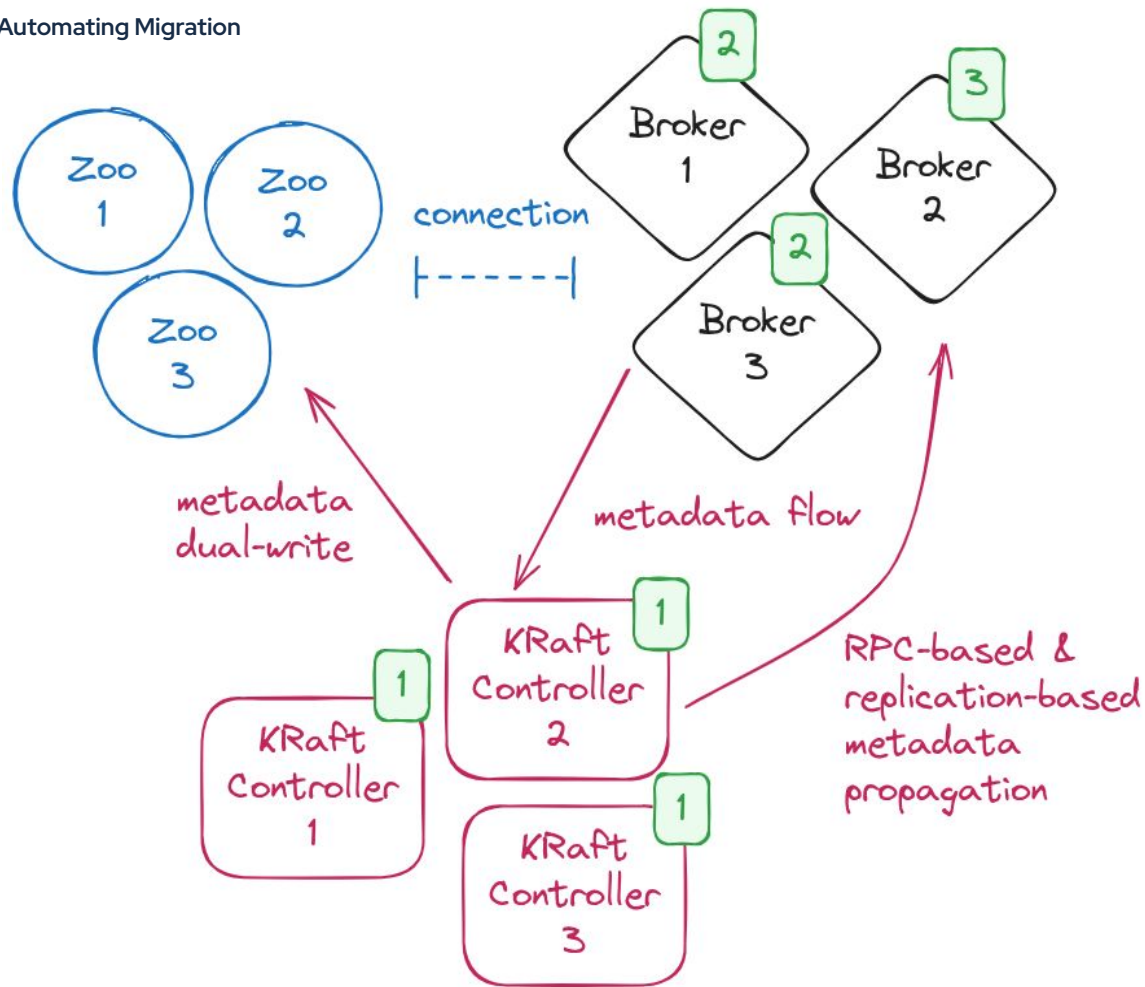
## Hybrid/Dual-write mode KRaftDualWriting

### User Actions:

- <no action>

### Strimzi Operator Actions:

- Update brokers configuration
- Roll broker Pods
- Update state



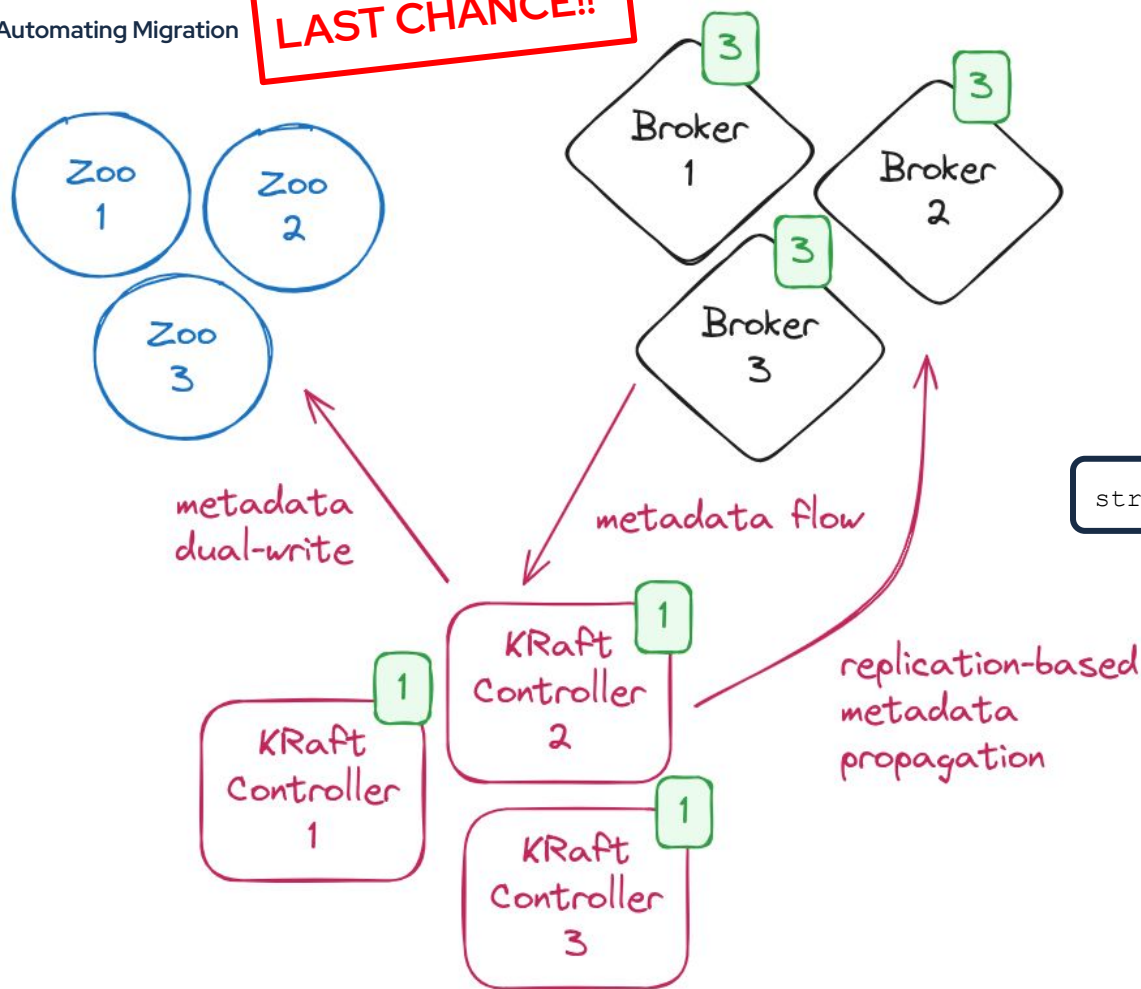
## Hybrid/Dual-write mode KRaftDualWriting

### User Actions:

- <no action>

### Strimzi Operator Actions:

- Update brokers configuration
- Roll broker Pods
- Update state

**LAST CHANCE!!**

## Dual-write mode KRaftPostMigration

### User Actions:

- Finalize
- OR
- Rollback

```
strimzi.io/kraft: enabled|rollback
```

### Strimzi Operator Actions:

- <no action>

## Dual-write mode KRaftPostMigration

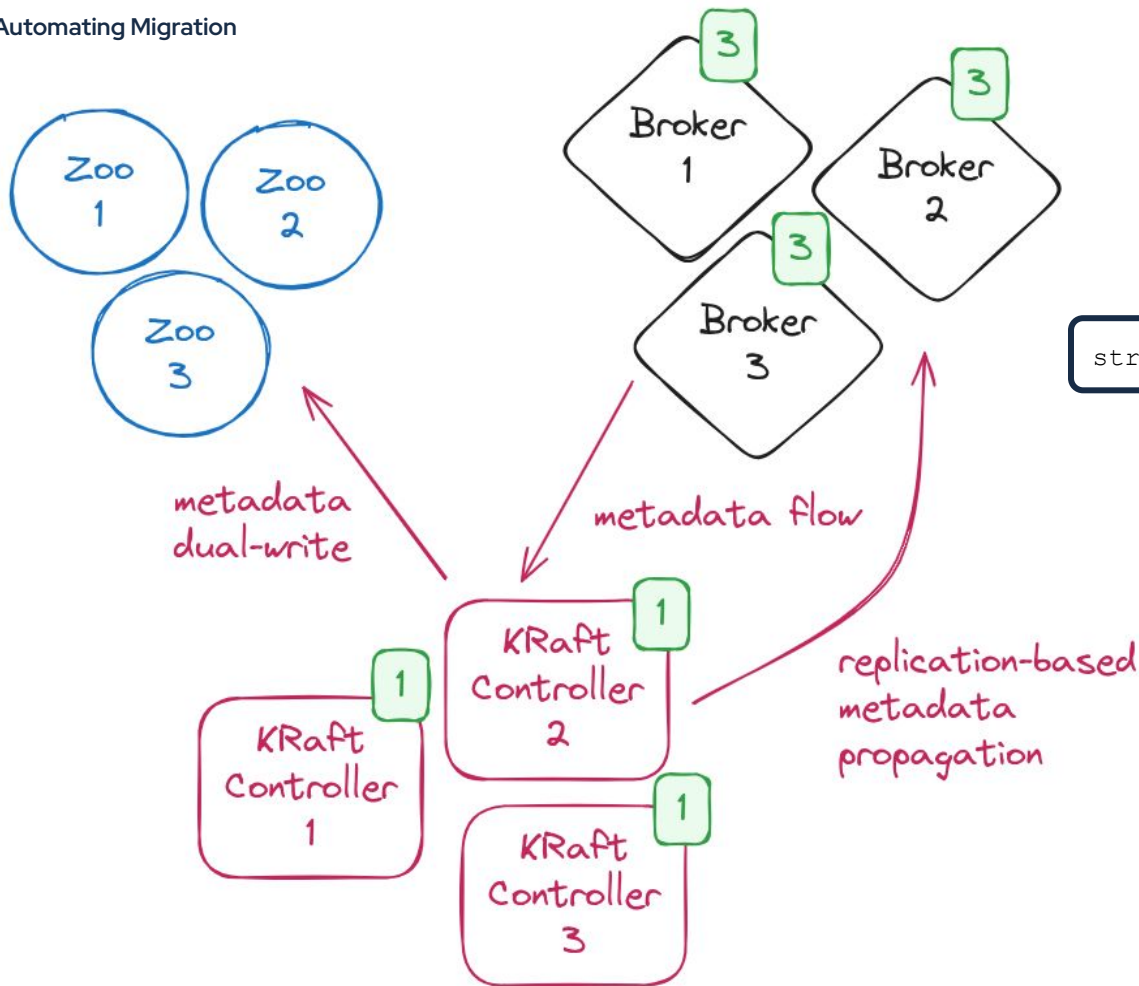
### User Actions:

- Finalize

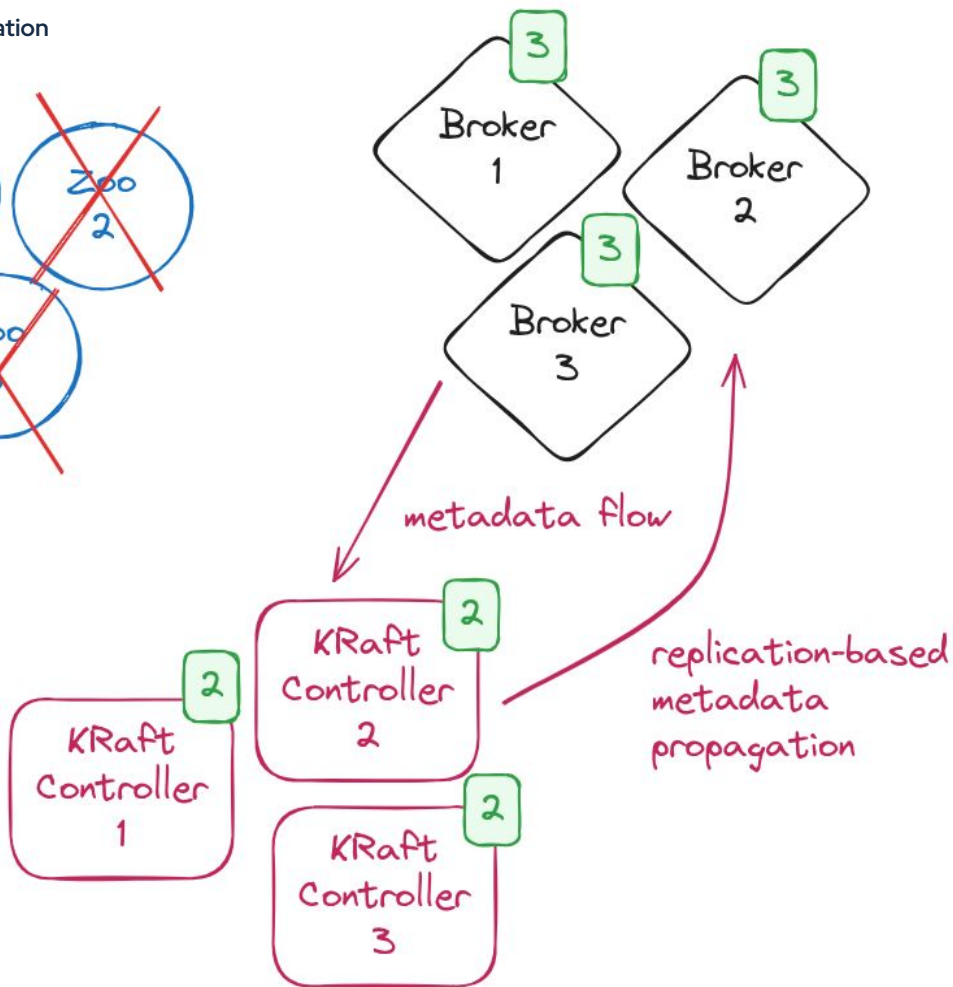
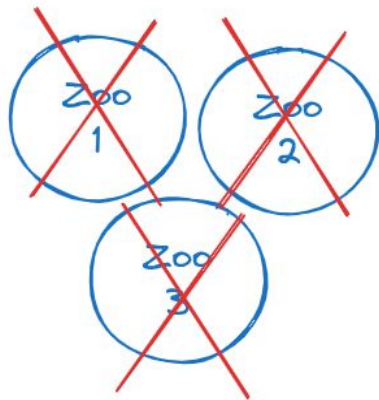
```
strimzi.io/kraft: enabled
```

### Strimzi Operator Actions:

- Update controllers configuration
- Roll controller pods
- Remove ZooKeeper
- Update state



## Automating Migration



## Finalized PreKRaft -> KRaft

### User Actions:

- Finalize

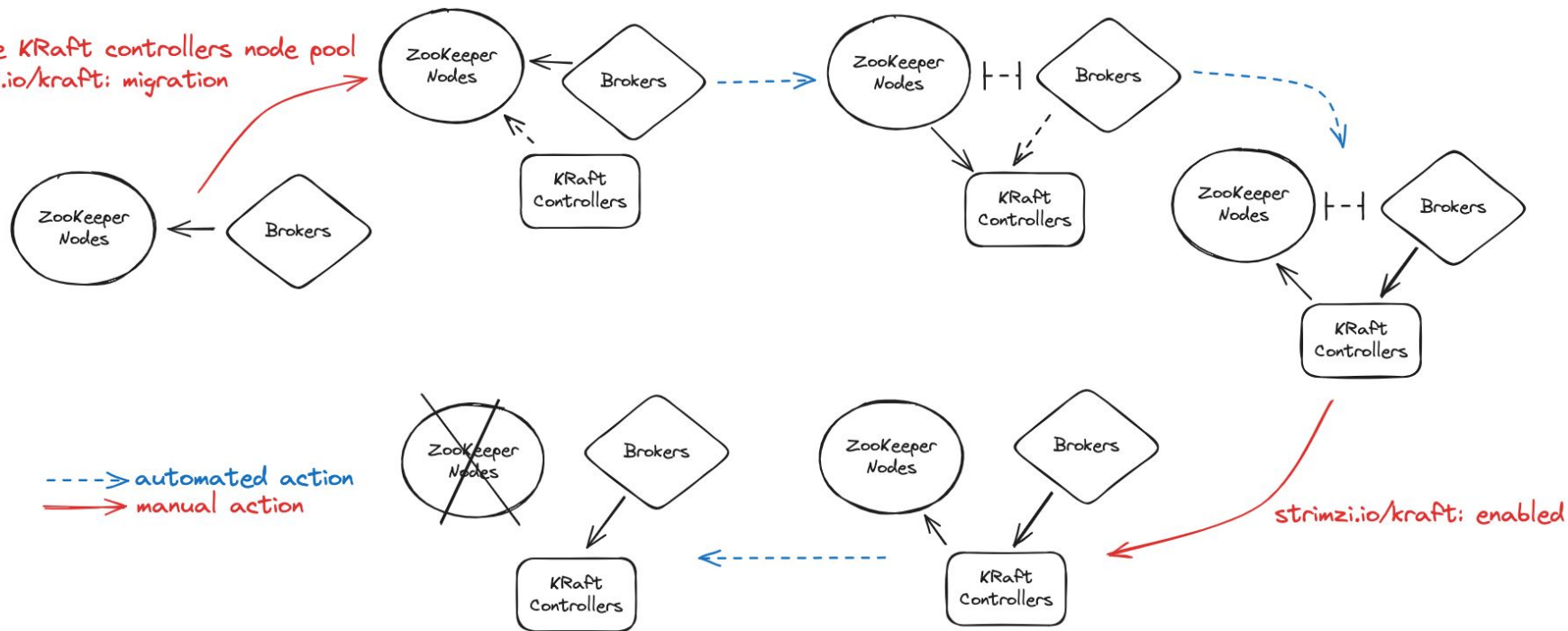
### Strimzi Operator Actions:

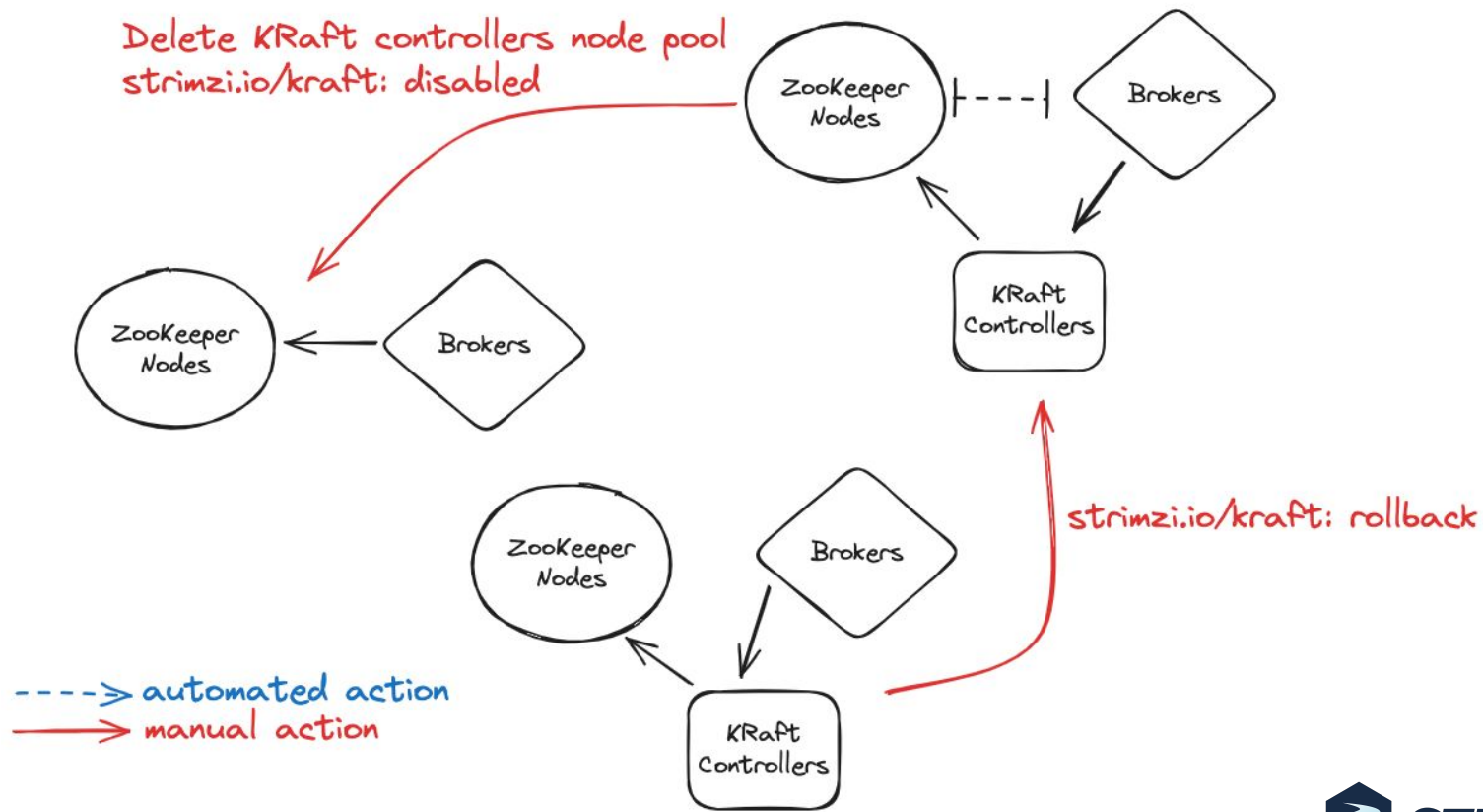
- Update controllers configuration
- Roll controller pods
- Remove ZooKeeper
- Update state



## Automating Migration

Create KRaft controllers node pool  
strimzi.io/kraft: migration







# Demo

# Summary

# Summary

- Migration can be automated
  - Changing configuration
  - Restarting nodes
  - Waiting
- Think about when and how the user should intervene
- Define and track states

# Migration Top Tips

- Migrate as soon as you can
- Perform practice migrations
- Don't forget dashboards and alerts

# Migration versions

Strimzi 0.45



✓ ZooKeeper mode  
✓ KRaft mode

Strimzi 0.46



✗ ZooKeeper mode  
✓ KRaft mode



# Thank you

KRaft migration overview:

[strimzi.io/blog/2024/03/21/kraft-migration](https://strimzi.io/blog/2024/03/21/kraft-migration)

KRaft migration in Strimzi:

[strimzi.io/blog/2024/03/22/strimzi-kraft-migration](https://strimzi.io/blog/2024/03/22/strimzi-kraft-migration)



Website: [strimzi.io](https://strimzi.io)



GitHub: [github.com/strimzi](https://github.com/strimzi)



Twitter: @strimziio



YouTube: [youtube.com/c/Strimzi](https://youtube.com/c/Strimzi)



LinkedIn: [linkedin.com/company/strimzi](https://linkedin.com/company/strimzi)