How to Start Apache Kafka 4.1: A Step-by-Step Guide

Starting Apache Kafka 4.1 is a little bit different from earlier versions.

Step 1: Generate a Cluster ID

First, you need a unique identifier for your entire cluster. This ID will be the same across all servers in your cluster.

Open a terminal and run:

```
CLUSTER_ID=$(./bin/kafka-storage.sh random-uuid)

echo "Cluster ID: $CLUSTER_ID"
```

Save this ID. You'll need it for the next steps. The output should look something like:

```
Cluster ID: F9_futKUQPKBwpQddvXsDQ
```

Step 2: Generate Log IDs for Each Server

In a production setup with three servers, you need a separate log ID for each one. This identifies each server's metadata log location.

Run these commands:

```
LOG_ID_1=$(./bin/kafka-storage.sh random-uuid)

LOG_ID_2=$(./bin/kafka-storage.sh random-uuid)

LOG_ID_3=$(./bin/kafka-storage.sh random-uuid)

echo "Log IDs: '$LOG_ID_1' '$LOG_ID_2' '$LOG_ID_3'"
```

This gives you three unique IDs. Keep these handy because you'll use them in the next step.

Step 3: Format Storage for Server 1

Now you need to prepare the storage directories. You run this command once for each server in your cluster.

For the first server, run:

```
./bin/kafka-storage.sh format -t "$CLUSTER_ID" -c
config/server-1.properties --initial-controllers
"1@localhost:9093:$LOG_ID_1,2@localhost:9095:$LOG_ID_2,3@localho
st:9097:$LOG_ID_3"
```

What this does is:

The -t flag passes your cluster ID. The -c flag points to the server configuration file. The --initial-controllers flag tells Kafka which servers form the controller group and their connection details.

Step 4: Format Storage for Server 2

Run the same command but with server-2.properties:

```
./bin/kafka-storage.sh format -t "$CLUSTER_ID" -c
config/server-2.properties --initial-controllers
"1@localhost:9093:$LOG_ID_1,2@localhost:9095:$LOG_ID_2,3@localho
st:9097:$LOG_ID_3"
```

Step 5: Format Storage for Server 3

Run it once more for server-3:

```
./bin/kafka-storage.sh format -t "$CLUSTER_ID" -c
config/server-3.properties --initial-controllers
"1@localhost:9093:$LOG_ID_1,2@localhost:9095:$LOG_ID_2,3@localho
st:9097:$LOG_ID_3"
```

Step 6: Start all three servers

Now that storage is formatted, you can start the servers. Open three terminal windows (you need one for each server).

In the first terminal, run:

```
./bin/kafka-server-start.sh config/server-1.properties
```

In the second terminal, run:

```
./bin/kafka-server-start.sh config/server-2.properties
```

In the third terminal, run:

```
./bin/kafka-server-start.sh config/server-3.properties
```

Give all three servers a moment to connect and form a cluster.

Step 9: Verify Your Cluster is Healthy

Run this command to check the cluster status:

```
./bin/kafka-metadata-quorum.sh --bootstrap-controller
localhost:9093 describe --status
```

You should see output like this:

```
ClusterId: F9_futKUQPKBwpQddvXsDQ
LeaderId: 1
LeaderEpoch: 1
HighWatermark: 733
MaxFollowerLag: 0
MaxFollowerLagTimeMs: 0
```

```
CurrentVoters: [{"id": 1, "directoryId":
"btakmqkkQpekcIPgShfi8A", "endpoints":
["CONTROLLER://localhost:9093"]}, {"id": 2, "directoryId":
"pT1IuR49QCagCQaTj0zjsA", "endpoints":
["CONTROLLER://localhost:9095"]}, {"id": 3, "directoryId":
"LFxKYibKQJe0LtCtB2K7BQ", "endpoints":
["CONTROLLER://localhost:9097"]}]
CurrentObservers: []
```

Look for these key points:

- **LeaderId** should show a number (1, 2, or 3). This means one server was elected as the leader.
- **MaxFollowerLag** should be 0. This means all servers are in sync.
- **CurrentVoters** should list all three servers. This means the cluster is complete.

If you see all these signs, your cluster is working properly.