Kafka Setup & Batch File Guide

Multi-Broker Cluster, Partitions & Replication Explained

Generated by ChatGPT (Jarvis Mode)

1. Introduction

Apache Kafka is a distributed event streaming platform. It runs on a cluster of brokers and stores data in topics. Each topic is divided into partitions, which can be replicated across multiple brokers for fault tolerance.

This guide explains how our batch (.bat) script sets up Kafka brokers, formats storage, and creates topics with custom partitions and replication factors.

2. Brokers & Controllers

- A Kafka cluster consists of one or more BROKERS.
- Each broker has a unique node.id and stores partitions of topics.
- In KRaft mode, one broker also acts as CONTROLLER to manage metadata.
- Our setup uses 3 brokers: Broker1, Broker2, Broker3.
- Broker1 runs as both broker + controller.

3. Why Format Storage?

Formatting storage assigns the Cluster ID and initializes metadata for each broker.
This step must be done once before starting brokers.

In the batch file:
- If logs folder (meta.properties) is missing → it runs 'StorageTool format'.
- Ensures all brokers belong to the same cluster.

4. Batch File Workflow

Our batch file does the following:

1) Defines Kafka home directory & cluster ID.
2) Creates separate config files for Broker1, Broker2, Broker3.
3) Formats storage if needed.
4) Starts all brokers with correct roles & ports.
5) Creates 'otp-service' topic with given partitions and replication factor.

5. Topics, Partitions & Replication

- A TOPIC is a category where messages are published.
- Each topic is split into PARTITIONS (parallelism & scaling).
- Each partition is stored on one or more brokers depending on replication factor.

Example: otp-service
    Partitions = 6
    Replication Factor = 3

Partition 0 → Leader on Broker1, Replica on Broker2 & Broker3
Partition 1 → Leader on Broker2, Replica on Broker3 & Broker1
Partition 2 → Leader on Broker3, Replica on Broker1 & Broker2

Thus all brokers share the topic and provide fault tolerance.

6. Modifying the Batch File

- To change number of BROKERS:
  Add new broker configs (server-4.properties, logs-4, etc.) in the batch file.

- To change PARTITIONS of a topic:
  Update the --partitions parameter in kafka-topics command.

- To change REPLICATION FACTOR:
  Update --replication-factor parameter. Max = number of brokers.

- To create NEW TOPICS:
  Add another kafka-topics command at the end of the batch file.

Example:

```
kafka-topics.bat --create --topic payments --partitions 8 --replication-factor 3 --bootstrap-server localhos
```

7. Summary

- Brokers store topic partitions; topics define business streams of data.
- Partitions = parallelism, Replication Factor = fault tolerance.
- Formatting assigns a cluster ID and initializes broker storage.
- Batch file automates broker setup, formatting, startup, and topic creation.
- You can modify the batch file to add brokers, change partitions, or replication factor.