Documentation on Modifying __cluster_metadata in Kafka KRaft Mode

I want to modify __cluster_metadata; can you describe the structure?

Kafka in KRaft (Kafka Raft) mode uses a dedicated internal topic named __cluster_metadata to manage critical cluster metadata.

This topic replaces the role traditionally held by Zookeeper in managing Kafka cluster metadata, configuration, and state.

Below is a detailed description of the structure and key components of the __cluster_metadata directory.

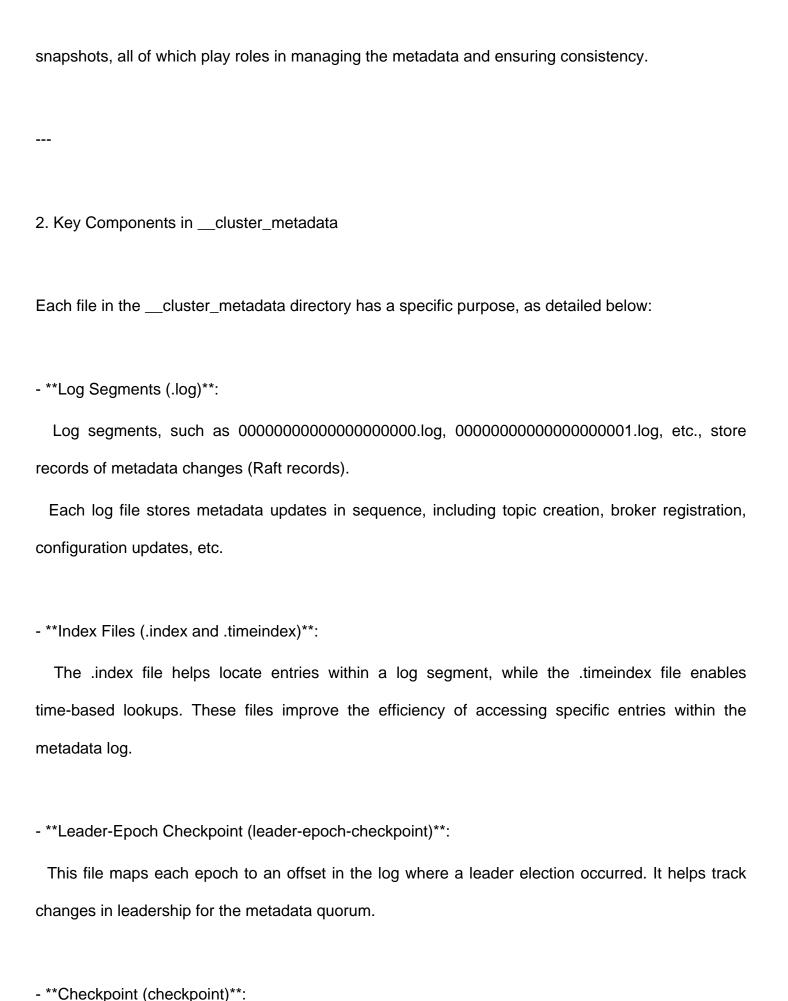
1. Introduction to __cluster_metadata Structure

In KRaft mode, Kafka's internal metadata (such as topic configurations, broker registrations, partition assignments, ACLs, and controller state) is managed by the Raft consensus protocol and stored in the __cluster_metadata directory.

This directory functions similarly to a topic partition directory but is specialized for storing the Raft log, which Kafka controllers use to synchronize and replicate the cluster's metadata.

Typical Structure of __cluster_metadata Directory

The __cluster_metadata directory contains log segments, index files, checkpoint files, and



The checkpoint file marks the most recent committed offset for the Raft quorum's metadata log,

assisting Raft followers in catching up on metadata changes.

- **Snapshot Directory (snapshot/)**:

This folder contains snapshots of the complete metadata state at specific points, allowing for efficient recovery by reducing the need to replay all logs.

3. Example Data Entries in Log Segments

Each log segment in the __cluster_metadata folder may contain records representing different metadata actions, such as:

- **Topic Creation (TopicRecord)**:
 - Fields include topic name, number of partitions, replication factor, and configurations.
 - Example:

```
{
  "record_type": "TopicRecord",
  "topic_name": "example_topic",
  "num_partitions": 3,
  "replication_factor": 2
}
```

- **Partition Assignment (PartitionRecord)**:

```
- Example:
  {
   "record_type": "PartitionRecord",
   "topic_name": "example_topic",
    "partition_id": 0,
   "replica_nodes": [1, 2]
  }
- **Configuration Update (ConfigRecord)**:
 - Changes to topic settings or broker configurations.
 - Example:
  {
    "record_type": "ConfigRecord",
   "config_key": "retention.ms",
   "config_value": "86400000"
  }
- **Broker Registration (BrokerRegistrationRecord)**:
 - Registers or deregisters brokers in the cluster.
 - Example:
  {
   "record_type": "BrokerRegistrationRecord",
   "broker_id": 1,
```

- Details about broker assignments for each partition.

