**recovery-point-offset-checkpoint** and **replication-offset-checkpoint**

In Apache Kafka, **recovery-point-offset-checkpoint** and **replication-offset-checkpoint** are checkpoint files located on brokers to manage recovery and replication. They are part of Kafka's mechanism for ensuring durability, consistency, and fast recovery during broker restarts or crashes.

**1. recovery-point-offset-checkpoint**

* **Purpose**:
  + Keeps track of the latest offsets for each partition that have been flushed to disk.
  + Ensures durability by marking a "safe point" for log recovery after a broker restart or crash.
* **Location**:
  + Found in the Kafka log directory (log.dirs).
* **Structure**:
  + The file contains a list of partitions and the offsets that have been fully flushed to disk.
* **Usage**:
  + During broker recovery, Kafka reads this file to determine the last stable offsets and avoid replaying already flushed data.

**Example File Content:**

Version: 1

Partition: TopicName-0 Offset: 500

Partition: TopicName-1 Offset: 300

Partition: TopicName-2 Offset: 1000

* **Explanation**:
  + For TopicName-0, all offsets up to 500 are considered flushed.
  + During recovery, the broker replays logs only from offsets greater than 500.

**2. replication-offset-checkpoint**

* **Purpose**:
  + Tracks the offsets replicated to a broker for each partition where the broker acts as a replica (but not the leader).
  + Helps resume replication from the correct offset after a broker restart.
* **Location**:
  + Found in the Kafka log directory (log.dirs).
* **Structure**:
  + Contains a mapping of partition to the latest replicated offset on that broker.
* **Usage**:
  + When a broker restarts, it uses this file to identify the point at which to resume fetching data from the leader.

**Example File Content:**

Version: 1

Partition: TopicName-0 Offset: 450

Partition: TopicName-1 Offset: 290

Partition: TopicName-2 Offset: 990

* **Explanation**:
  + For TopicName-0, the broker has replicated up to offset 450.
  + On restart, the broker will fetch data starting from offset 451.

**Differences Between the Two Checkpoints**

|  |  |  |
| --- | --- | --- |
| **Aspect** | **recovery-point-offset-checkpoint** | **replication-offset-checkpoint** |
| **Purpose** | Tracks flushed offsets for durability. | Tracks replicated offsets for consistency. |
| **Usage** | Used during broker recovery to avoid replaying flushed data. | Used to resume replication from the last replicated offset. |
| **Applies To** | Applies to partitions where the broker is the leader. | Applies to partitions where the broker is a follower. |
| **Recovery Scenario** | Avoids replay of data already persisted to disk. | Avoids re-fetching data already replicated. |

**Example Scenario: Broker Restart**

1. **Before Restart**:
   * A broker writes data to the disk for TopicName-0 but only offsets up to 500 are flushed.
   * As a replica, the broker has replicated up to offset 450 for TopicName-1.
2. **During Restart**:
   * Kafka reads recovery-point-offset-checkpoint to determine that offsets up to 500 for TopicName-0 are already durable.
   * Kafka reads replication-offset-checkpoint to resume fetching data for TopicName-1 from offset 451.
3. **After Restart**:
   * The broker resumes as a leader and follower, ensuring that data consistency and durability are maintained.

**Advanced Notes**

* Both files are updated periodically by Kafka based on the broker's **log.flush.interval.messages** and **log.flush.interval.ms** configurations.
* If these files are corrupted or missing, Kafka may perform a more time-consuming log recovery process, which can increase broker startup time.