

# **Message Queue CKafka Migration Description Product Documentation**



## Copyright Notice

©2013-2019 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

## Trademark Notice



All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

## Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

# Migration Description

Last updated : 2020-07-20 16:15:46

## Operation Scenarios

You can migrate data between different CKafka instances in the following steps. For more information, please see [Migrating Data to CKafka](#).

## Directions

### 1. Create an instance

Create a CKafka instance (for more information, please see [Creating Instance](#)).

### 2. Create a topic

Create a topic with the same name as that in the legacy CKafka instance (for more information, please see [Creating Topic](#)).

### 3. Switch the producer

Switch the producer to the new CKafka instance so that the legacy instance will not receive new message data.

- Existing data in the legacy CKafka instance (if any) will continue to be consumed till the end.
- If there is no existing data in the legacy CKafka instance, you need to change the `bootstrap.servers` of the consumer to the new instance's address for consumption. As the offset of the new instance will start from zero, you need to perform the following steps based on the actual business needs when switching the consumer group:

#### 3.1. Open-Source Kafka tools

You may use open-source Kafka tools when switching the consumer group.

Tool used to get the topic offset at the specified time:

#### Note :

This tool can get the offset based on the file modification time. Its accuracy is relatively low, but it is compatible with production on legacy versions (Kafka v0.9 and below).

```
./kafka-run-class.sh kafka.tools.GetOffsetShell --broker-list xxx --topic xxx --time xxx
```

Here, `time` is the Unix time in milliseconds, `-1` indicates to pull the offset of the latest version, and `-2` indicates to pull the offset of the oldest version.

- Tool used to reset the consumer group offset (available in Kafka v0.11 and below)

```
./kafka-consumer-groups.sh --new-consumer --bootstrap-server xxx --reset-offsets --to-datetime xxx  
(such as '2019-03-05T00:00:00.000') --group xxx --topic xxx -execute
```

When using this tool, if the offset is reset by a specified time, you need to make sure that the producer is Kafka v0.10 or above; otherwise, as there is no timestamp in the production on a legacy version, the time will be incorrect, and the reset position will also be incorrect.

- Reset the offset without timestamp of production in Kafka 0.10 or below

You need to use the two offset tools above together.

- i. Pull the offset.

```
./kafka-run-class.sh kafka.tools.GetOffsetShell --broker-list xxx --topic xxx --time xxx
```

Here, `time` is the Unix time in milliseconds, `-1` indicates to pull the offset of the latest version, and `-2` indicates to pull the offset of the oldest version.

- ii. Use the tool to specify the offset of each partition for reset through a file.

```
./kafka-consumer-groups.sh --new-consumer --bootstrap-server xxx --reset-offsets --from-file  
xxx --group xxx -execute
```

Here, `file` records the topic, partition, and offset as rows in CSV format.

### 3.2. If data is synced from legacy instance to new instance before switch

Offset managed by broker in consumer group:

- If you can tolerate a small amount of data loss during consumer switch, set the configuration item `auto.offset.reset` to `latest` during consumer migration (it is the default value. You do not need to change anything if this value was not set on the legacy version).
- If you cannot tolerate any data loss, you need to record the approximate time of producer switch and complete relevant settings as described in [Open-Source Kafka tools](#Open-Source Kafka tools).

Offset retained by yourself:

As the offset of the corresponding partition in the new instance will change, you need to get the corresponding offset information. You can use the tool described in 3.1 to pull the offset based on the approximate time of producer switch, update the retained offset, and restart the consumer.

### 3.3. If data is not synced from legacy instance to new instance before switch

Offset managed by broker in consumer group:

- If you can tolerate a small amount of data loss during consumer switch, you can directly restart the consumer after modifying the broker address configuration during consumer migration.
- If you cannot tolerate any data loss, set the configuration item `auto.offset.reset` to `earliest` and start the consumer.

Offset retained by yourself:

- Directly modify the value of the retained offset to 0 and restart the consumer.  
If the consumer of the new instance fails to be started after a long time and the retention time set in the instance is too short, the legacy data may be deleted, causing the offset to start from a non-zero value. In this case, you can use the corresponding tool to pull the latest offset information as instructed in [Open-Source Kafka tools](#Open-Source Kafka tools) and use it to update the retained offset and restart the consumer.

#### **Note :**

- You can also directly perform doublewrite (i.e., writing into the new and legacy instances at the same time) through the producer; however, such switch may consume the same piece of data repeatedly.
- A corresponding discount will be provided based on your estimated business scale.
- You are recommended to use the topic copy script of Python to migrate the instance to the cluster on the new version (a README document will be present after the script is decompressed).
- Download the aforementioned open-source script [here >>](#)