

# Kafka 2.8 启用 ZooKeeper

AUTHOR: 彭玲 TIME: 2022/3/10

## Kafka 2.8 启用 ZooKeeper

### 前言

#### Kafka 环境搭建 (单机版)

1. 生成集群 ID, 设置日志目录
2. 启动 Kafka 服务器
3. 创建主题
4. 验证 Kafka 环境搭建

启动生产者

启动消费者

#### Kafka 环境搭建 (集群版)

Kafka cluster 配置

生成集群 ID, 设置日志目录

启动服务器

创建主题

## 前言

2021年3月30日, Kafka 背后的企业 Confluent 发布博客表示, 在即将发布的 **2.8 版本**里, 用户可在完全不需要 ZooKeeper 的情况下运行 Kafka, 该版本将依赖于 ZooKeeper 的控制器改造成了基于 Kafka Raft 的 Quorum 控制器。

这是一次架构上的重大升级, 让一向“重量级”的 Kafka 从此变得简单了起来。

## Kafka 环境搭建 (单机版)

Kafka 3.1 中的 KRaft 模式仅用于测试, **不用于生产**。

### 1. 生成集群 ID, 设置日志目录

```
1 anxin@node38:~/pengling/kafka-v3/kafka_2.13-3.1.0$ ./bin/kafka-storage.sh
  random-uuid
2 s-JhF2xLQGminL89ivIMww
```

使用上面步骤生成的 cluster ID (`s-JhF2xLQGminL89ivIMww`) 格式化存储目录。

```
1 anxin@node38:~/pengling/kafka-v3/kafka_2.13-3.1.0$ ./bin/kafka-storage.sh
  format -t s-JhF2xLQGminL89ivIMww -c ./config/kraft/server.properties
2 Formatting /tmp/kraft-combined-logs
```

## 2. 启动 Kafka 服务器

```
1 # -daemon 后台启动
2 ./bin/kafka-server-start.sh ./config/kraft/server.properties # 前台启动
```

## 3. 创建主题

```
1 anxin@node38:~/pengling/kafka-v3/kafka_2.13-3.1.0$ ./bin/kafka-topics.sh --
  create --topic foo --partitions 1 --replication-factor 1 --bootstrap-server
  localhost:9092
2 Created topic foo.
```

## 4. 验证 Kafka 环境搭建

当在 `生产者` 下输入消息并回车后, 在 `消费者` 窗口下就能立即看到对应的消息, 这就说明环境搭建成功。

### 启动生产者

```
1 anxin@node38:~/pengling/kafka-v3/kafka_2.13-3.1.0$ ./bin/kafka-console-
  producer.sh --bootstrap-server localhost:9092 --topic foo
2 >Apache Kafka
3 >Quorum Controller
4 >Event Driven Consensus
5 >
```

### 启动消费者

```
1 anxin@node38:~/pengling/kafka-v3/kafka_2.13-3.1.0$ ./bin/kafka-console-
  consumer.sh --bootstrap-server localhost:9092 --topic foo --from-beginning
2 Apache Kafka
3 Quorum Controller
4 Event Driven Consensus
```

## Kafka 环境搭建 (集群版)

### Kafka cluster 配置

创建 3 个 集群节点:

```
1 $ cd config/kraft
2 $ cp server.properties server1.properties
3 $ cp server.properties server2.properties
4 $ cp server.properties server3.properties
```

3 个 服务的配置文件中, 注意以下不同点:

- node.id
- listeners
- log.dirs

```

1 node.id=1
2
3 process.roles=broker,controller
4
5 inter.broker.listener.name=PLAINTEXT
6
7 controller.listener.names=CONTROLLER
8
9 listeners=PLAINTEXT://:9092,CONTROLLER://:19092
10
11 log.dirs=/tmp/server1/kraft-combined-logs
12
13 listener.security.protocol.map=CONTROLLER:PLAINTEXT,PLAINTEXT:PLAINTEXT,SSL:
SSL,SASL_PLAINTEXT:SASL_PLAINTEXT,SASL_SSL:SASL_SSL
14
15 controller.quorum.voters=1@localhost:19092,2@localhost:19093,3@localhost:190
94

```

## 生成集群 ID，设置日志目录

```

1 $ ./bin/kafka-storage.sh format -t <random-uuid> -c
./config/kraft/server1.properties
2 Formatting /tmp/server1/kraft-combined-logs
3 $ ./bin/kafka-storage.sh format -t <random-uuid> -c
./config/kraft/server2.properties
4 Formatting /tmp/server2/kraft-combined-logs
5 $ ./bin/kafka-storage.sh format -t <random-uuid> -c
./config/kraft/server3.properties
6 Formatting /tmp/server3/kraft-combined-logs

```

## 启动服务器

```

1 ./bin/kafka-server-start.sh -daemon ./config/kraft/server1.properties
2 ./bin/kafka-server-start.sh -daemon ./config/kraft/server2.properties
3 ./bin/kafka-server-start.sh -daemon ./config/kraft/server3.properties

```

## 创建主题

在 3 个节点的集群中，创建 3 partitions 和 3 replicas：

```

1 ./bin/kafka-topics.sh --create --topic kraft-test --partitions 3 --
replication-factor 3 --bootstrap-server localhost:9092

```