

# 课程目标

- 1、掌握 Kafka Broker 文件存储结构分析
- 2、掌握 Kafka 消息保留和清理机制
- 3、掌握 Kafka 高可用及选举机制
- 4、掌握 Kafka 数据同步原理及故障处理

## 课前准备

### 1、CentOS Kafka 集群环境安装

#### 1.1、安装 Kafka

和单机版安装一样，分别 192.168.8.144、192.168.8.145、192.168.8.146 三台机器安装好 Kafka 环境。

具体安装过程，参考《Kafka 应用场景及架构设计详解》预习资料

#### 1.2、修改配置文件

分别修改 192.168.8.144、192.168.8.145、192.168.8.146 这 2 台机器上的配置文件

```
cd /usr/local/kafka/config  
vim server.properties
```

修改配置文件中的 `broker.id` 分别为 144、145、146

```
broker.id=144
```

`listeners` 分别设置为 `PLAINTEXT://192.168.8.144:9092`、`PLAINTEXT://192.168.8.145:9092`、`PLAINTEXT://192.168.8.146:9092`

```
listeners=PLAINTEXT://192.168.8.144:9092
```

三台机器的 `zookeeper.connect` 都修改为以下内容：

```
zookeeper.connect=192.168.8.144:2181,192.168.8.145:2181,192.168.8.146:2181
```

### 1.3、启动 3 个服务

第一步：

分别启动 ZK。

```
nohup ./bin/zookeeper-server-start.sh config/zookeeper.properties >> zookeeper.nohup &
```

再分别启动 kafka。

```
cd ../bin  
./kafka-server-start.sh -daemon ../config/server.properties
```

PS: 如果遇到 zk node exists 的问题, 先把 brokers 节点删掉 (临时解决方案)。

### 1.4、集群下创建 Topic

在 bin 目录下,

创建一个名为 gptest 的 topic, 只有一个副本, 一个分区:

```
sh kafka-topics.sh --create --topic gptest --bootstrap-server 192.168.8.144:9092
```

查看已经创建的 topic:

```
sh kafka-topics.sh --bootstrap-server 192.168.8.144:9092 --describe --topic gptest
```

### 1.5、集群下启动 Consumer

在一个新的远程窗口中:

```
cd /usr/local/kafka/bin  
sh kafka-console-consumer.sh --bootstrap-server  
192.168.8.144:9092,192.168.8.144:9092,192.168.8.144:9092 --topic gptest --from-beginning
```

### 1.6、集群下启动 Producer

打开一个新的窗口, 在 kafka 解压目录下:

```
sh kafka-console-producer.sh --broker-list 192.168.8.144:9092,192.168.8.144:9092,192.168.8.144:9092  
--topic gptest
```

## 1.7、集群下 Producer 窗口发送消息

在生产者窗口输入 hello world 回车

## 2、CentOS Kafka 单机伪集群安装

注意：

单机的 kafka 和集群的 kafka 不要混用一个 zk，否则会出现数据混乱的问题。

### 1.1、下载解压 kafka

```
cd /usr/local/

wget https://dlcdn.apache.org/kafka/3.2.0/kafka_2.13-3.2.0.tgz

tar -xzvf kafka_2.13-3.2.0.tgz

mv kafka_2.13-3.2.0 kafka

cd kafka
```

### 1.2、修改配置文件

复制 3 个配置文件

```
cd config

cp server.properties server1.properties

cp server.properties server2.properties

cp server.properties server3.properties
```

修改配置文件中的 broker.id 分别为 1、2、3

listeners 这一行取消注释，端口号分别为 9093、9094、9095

log.dirs 分别设置为 kafka-logs1、kafka-logs2、kafka-logs3（先创建）

```
mkdir -p /tmp/kafka-logs1 /tmp/kafka-logs2 /tmp/kafka-logs3
```

server1.properties 的配置：

```
broker.id=1
```

```
listeners=PLAINTEXT://192.168.8.147:9093
```

```
log.dirs=/tmp/kafka-logs1
```

server2.properties 的配置:

```
broker.id=2
```

```
listeners=PLAINTEXT://192.168.8.147:9094
```

```
log.dirs=/tmp/kafka-logs2
```

server3.properties 的配置:

```
broker.id=3
```

```
listeners=PLAINTEXT://192.168.8.147:9095
```

```
log.dirs=/tmp/kafka-logs3
```

### 1.3、启动 3 个伪集群服务

第一步:

启动 ZK。

再启动 kafka。

```
cd ../bin
```

```
./kafka-server-start.sh -daemon ../config/server1.properties
```

```
./kafka-server-start.sh -daemon ../config/server2.properties
```

```
./kafka-server-start.sh -daemon ../config/server3.properties
```

PS: 如果遇到 zk node exists 的问题, 先把 brokers 节点删掉 (临时解决方案)。

### 1.4、伪集群下创建 Topic

在 bin 目录下,

创建一个名为 gptest 的 topic, 只有一个副本, 一个分区:

```
sh kafka-topics.sh --create --topic gptest --bootstrap-server 192.168.8.147:9092
```

查看已经创建的 topic:

```
sh kafka-topics.sh --bootstrap-server 192.168.8.147:9092 --describe --topic gptest
```

### 1.5、伪集群下启动 Consumer

在一个新的远程窗口中：

```
kafka-console-consumer.sh --bootstrap-server 192.168.8.147:9093,192.168.8.147:9094,192.168.8.147:9095  
--topic gptest --from-beginning
```

## 1.6、伪集群下启动 Producer

打开一个新的窗口，在 kafka 解压目录下：

```
kafka-console-producer.sh --broker-list 192.168.8.147:9093,192.168.8.147:9094,192.168.8.147:9095  
--topic gptest
```

## 1.7、伪集群下 Producer 窗口发送消息

在生产者窗口输入 hello world 回车

咕泡出品 必属精品