尚硅谷大数据技术之Kafka

（作者：尚硅谷研究院）

版本：V3.0.0

# 第1章 Kafka快速入门

## 1.1 安装部署

### 1.1.1 集群规划

|  |  |  |
| --- | --- | --- |
| hadoop102 | hadoop103 | hadoop104 |
| zk | zk | zk |
| kafka | kafka | kafka |

### 1.1.2 集群部署

**0）**官方下载地址：<http://kafka.apache.org/downloads.html>

**1）**解压安装包

[atguigu@hadoop102 software]$ tar -zxvf kafka\_2.12-3.0.0.tgz -C /opt/module/

**2）**修改解压后的文件名称

tguigu@hadoop102 module]$ mv kafka\_2.12-3.0.0/ kafka

**3）**进入到/opt/module/kafka目录，修改配置文件

[atguigu@hadoop102 kafka]$ cd config/

[atguigu@hadoop102 config]$ vim server.properties

输入以下内容：

#broker的全局唯一编号，不能重复，只能是数字。

broker.id=0

#处理网络请求的线程数量

num.network.threads=3

#用来处理磁盘IO的线程数量

num.io.threads=8

#发送套接字的缓冲区大小

socket.send.buffer.bytes=102400

#接收套接字的缓冲区大小

socket.receive.buffer.bytes=102400

#请求套接字的缓冲区大小

socket.request.max.bytes=104857600

#kafka运行日志(数据)存放的路径，路径不需要提前创建，kafka自动帮你创建，可以配置多个磁盘路径，路径与路径之间可以用"，"分隔

log.dirs=/opt/module/kafka/datas

#topic在当前broker上的分区个数

num.partitions=1

#用来恢复和清理data下数据的线程数量

num.recovery.threads.per.data.dir=1

# 每个topic创建时的副本数，默认时1个副本

offsets.topic.replication.factor=1

#segment文件保留的最长时间，超时将被删除

log.retention.hours=168

#每个segment文件的大小，默认最大1G

log.segment.bytes=1073741824

# 检查过期数据的时间，默认5分钟检查一次是否数据过期

log.retention.check.interval.ms=300000

#配置连接Zookeeper集群地址（在zk根目录下创建/kafka，方便管理）

zookeeper.connect=hadoop102:2181,hadoop103:2181,hadoop104:2181/kafka

**4）**分发安装包

[atguigu@hadoop102 module]$ xsync kafka/

**5）**分别在hadoop103和hadoop104上修改配置文件/opt/module/kafka/config/server.properties中的broker.id=1、broker.id=2

注：broker.id不得重复，整个集群中唯一。

[atguigu@hadoop103 module]$ vim kafka/config/server.properties

修改:

# The id of the broker. This must be set to a unique integer for each broker.

broker.id=1

[atguigu@hadoop104 module]$ vim kafka/config/server.properties

修改:

# The id of the broker. This must be set to a unique integer for each broker.

broker.id=2

**6）**配置环境变量

（1）在/etc/profile.d/my\_env.sh文件中增加kafka环境变量配置

[atguigu@hadoop102 module]$ sudo vim /etc/profile.d/my\_env.sh

增加如下内容：

#KAFKA\_HOME

export KAFKA\_HOME=/opt/module/kafka

export PATH=$PATH:$KAFKA\_HOME/bin

（2）刷新一下环境变量。

[atguigu@hadoop102 module]$ source /etc/profile

（3）分发环境变量文件到其他节点，并source。

[atguigu@hadoop102 module]$ sudo /home/atguigu/bin/xsync /etc/profile.d/my\_env.sh

[atguigu@hadoop103 module]$ source /etc/profile

[atguigu@hadoop104 module]$ source /etc/profile

**7）**启动集群

（1）先启动Zookeeper集群，然后启动Kafka。

[atguigu@hadoop102 kafka]$ zk.sh start

（2）依次在hadoop102、hadoop103、hadoop104节点上启动Kafka。

[atguigu@hadoop102 kafka]$ bin/kafka-server-start.sh -daemon config/server.properties

[atguigu@hadoop103 kafka]$ bin/kafka-server-start.sh -daemon config/server.properties

[atguigu@hadoop104 kafka]$ bin/kafka-server-start.sh -daemon config/server.properties

注意：配置文件的路径要能够到server.properties。

**8）**关闭集群

[atguigu@hadoop102 kafka]$ bin/kafka-server-stop.sh

[atguigu@hadoop103 kafka]$ bin/kafka-server-stop.sh

[atguigu@hadoop104 kafka]$ bin/kafka-server-stop.sh

### 1.1.3 集群启停脚本

1）在/home/atguigu/bin目录下创建文件kf.sh脚本文件

[atguigu@hadoop102 bin]$ vim kf.sh

脚本如下：

#! /bin/bash

case $1 in

"start"){

for i in hadoop102 hadoop103 hadoop104

do

echo " --------启动 $i Kafka-------"

ssh $i "/opt/module/kafka/bin/kafka-server-start.sh -daemon /opt/module/kafka/config/server.properties"

done

};;

"stop"){

for i in hadoop102 hadoop103 hadoop104

do

echo " --------停止 $i Kafka-------"

ssh $i "/opt/module/kafka/bin/kafka-server-stop.sh "

done

};;

esac

2）添加执行权限

[atguigu@hadoop102 bin]$ chmod +x kf.sh

3）启动集群命令

[atguigu@hadoop102 ~]$ kf.sh start

4）停止集群命令

[atguigu@hadoop102 ~]$ kf.sh stop

**注意：**停止Kafka集群时，一定要等Kafka所有节点进程全部停止后再停止Zookeeper集群。因为Zookeeper集群当中记录着Kafka集群相关信息，Zookeeper集群一旦先停止，Kafka集群就没有办法再获取停止进程的信息，只能手动杀死Kafka进程了。

## 1.2 Kafka命令行操作

### 1.2.1 主题命令行操作

1）查看操作主题命令参数

[atguigu@hadoop102 kafka]$ bin/kafka-topics.sh

|  |  |
| --- | --- |
| 参数 | 描述 |
| --bootstrap-server <String: server toconnect to> | 连接的Kafka Broker主机名称和端口号。 |
| --topic <String: topic> | 操作的topic名称。 |
| --create | 创建主题。 |
| --delete | 删除主题。 |
| --alter | 修改主题。 |
| --list | 查看所有主题。 |
| --describe | 查看主题详细描述。 |
| --partitions <Integer: # of partitions> | 设置分区数。 |
| --replication-factor<Integer: replication factor> | 设置分区副本。 |
| --config <String: name=value> | 更新系统默认的配置。 |

2）查看当前服务器中的所有topic

[atguigu@hadoop102 kafka]$ bin/kafka-topics.sh --bootstrap-server hadoop102:9092 --list

3）创建first topic

[atguigu@hadoop102 kafka]$ bin/kafka-topics.sh --bootstrap-server hadoop102:9092 --create --partitions 1 --replication-factor 3 --topic first

选项说明：

--topic 定义topic名

--replication-factor 定义副本数

--partitions 定义分区数

4）查看first主题的详情

[atguigu@hadoop102 kafka]$ bin/kafka-topics.sh --bootstrap-server hadoop102:9092 --describe --topic first

5）修改分区数（注意：分区数只能增加，不能减少）

[atguigu@hadoop102 kafka]$ bin/kafka-topics.sh --bootstrap-server hadoop102:9092 --alter --topic first --partitions 3

6）再次查看first主题的详情

[atguigu@hadoop102 kafka]$ bin/kafka-topics.sh --bootstrap-server hadoop102:9092 --describe --topic first

7）删除topic（学生自己演示）

[atguigu@hadoop102 kafka]$ bin/kafka-topics.sh --bootstrap-server hadoop102:9092 --delete --topic first

### 1.2.2 生产者命令行操作

1）查看操作生产者命令参数

[atguigu@hadoop102 kafka]$ bin/kafka-console-producer.sh

|  |  |
| --- | --- |
| 参数 | 描述 |
| --bootstrap-server <String: server toconnect to> | 连接的Kafka Broker主机名称和端口号。 |
| --topic <String: topic> | 操作的topic名称。 |

2）发送消息

[atguigu@hadoop102 kafka]$ bin/kafka-console-producer.sh --bootstrap-server hadoop102:9092 --topic first

>hello world

>atguigu atguigu

### 1.2.3 消费者命令行操作

**1）查看操作消费者命令参数**

[atguigu@hadoop102 kafka]$ bin/kafka-console-consumer.sh

|  |  |
| --- | --- |
| 参数 | 描述 |
| --bootstrap-server <String: server toconnect to> | 连接的Kafka Broker主机名称和端口号。 |
| --topic <String: topic> | 操作的topic名称。 |
| --from-beginning | 从头开始消费。 |
| --group <String: consumer group id> | 指定消费者组名称。 |

**2）消费消息**

（1）消费first主题中的数据。

[atguigu@hadoop102 kafka]$ bin/kafka-console-consumer.sh --bootstrap-server hadoop102:9092 --topic first

（2）把主题中所有的数据都读取出来（包括历史数据）。

[atguigu@hadoop102 kafka]$ bin/kafka-console-consumer.sh --bootstrap-server hadoop102:9092 --from-beginning --topic first