

## Kafka分布式集群部署手册(一)

在本博客的[《Apache Kafka-0.8.1.1源码编译》](#)文章中简单地谈到如何用gradlew或sbt编译Kafka 0.8.1.1的代码。今天主要来谈谈如何部署一个分布式集群。以下本文所有的内容都是基于Kafka 0.8.1.1 (Kafka 0.7.x的操作命令和本文略有不同，请注意！) 在介绍Kafka分布式部署之前，先来了解一下Kafka的基本概念。

- (1) Kafka维护按类区分的消息，称为主题 (topic)
- (2) 生产者 (producer) 向kafka的主题发布消息
- (3) 消费者 (consumer) 向主题注册，并且接收发布到这些主题的消息
- (4) kafka以一个拥有一台或多台服务器的集群运行着，每一台服务器称为broker

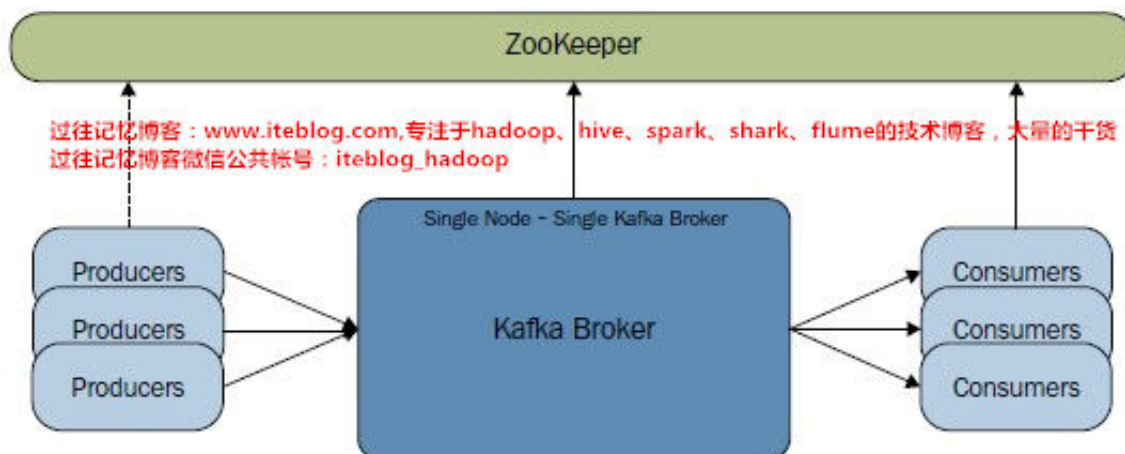
Kafka分布式集群的创建主要有三种模式：

- (1)、Single node - single broker集群；
- (2)、Single node - multiple broker集群；
- (3)、Multiple node - multiple broker集群。

下面简单地说说如何部署这三种模式的Kafka集群。

### 一、Single node - single broker集群

Single node - single broker集群的集群的体系结构图如下所示：



欢迎关注微信公共帐号：iteblog\_hadoop

- (1)、启动Zookeeper服务

在上述图中我们可以看到，图中只有一个节点且在该节点只启动了一个broker，这就是Single node – single broker模式的集群。整个集群和Zookeeper都有通信，所以在部署集群的时候，你需要

安装部署好Zookeeper（关于如何部署分布式的Zookeeper集群，请参见本博客的[《Zookeeper 3.4.5分布式安装手册》](#)。）当然，在Kafka的\$KAFKA\_HOME/bin目录下有zookeeper-server-start.sh脚本文件，我们可以利用这个脚本文件来启动Zookeeper，如下：

```
# bin/zookeeper-server-start.sh config/zookeeper.properties
```

在默认情况下，Zookeeper将会在2181端口下监听，我们可以通过修改clientPort参数来进行修改。

## （2）、启动Kafka broker服务

启动完Zookeeper之后，我们修改config/server.properties配置文件里面的以下三个配置项：

```
#####  
#####  
#  
# User: 过往记忆  
# Date: 14-6-23  
# Time: 下午17:37  
# blog:  
# 本文地址：/archives/1045  
# 过往记忆博客，专注于hadoop、hive、spark、shark、flume的技术博客，大量的干货  
# 过往记忆博客微信公共帐号：iteblog_hadoop  
#  
#####  
#####  
  
# The id of the broker. This must be set to a unique integer for each broker.  
Broker.id=0  
  
# The directory under which to store log files  
log.dir=/tmp/kafka8-logs  
  
# Zookeeper connection string  
zookeeper.connect=localhost:2181
```

Broker.id是该broker的id，这个值必须是唯一的；log.dir该目录用来配置存放log file的目录；zookeeper.connect是Zookeeper的监听地址（格式是host:port），默认情况下是localhost:2181。修

改完之后我们可以将Kafka broker服务打开，命令如下：

```
# bin/kafka-server-start.sh config/server.properties
```

这时候我们就打开了一个broker。

(3)、如何测试？

打开broker之后，我们就可以在其中创建一个topic，并用producer往该topic中发送消息；同时用consumer从中获取消息。如下所示：

创建一个topic

```
# bin/kafka-topics.sh --create --zookeeper localhost:2181 \W  
--replication-factor 1 --partitions 1 --topic wyp  
Created topic "wyp".
```

1、如果topic之前就创建了，你再重复去创建，将会出现以下的异常信息：

```
Error while executing topic command Topic "wyp" already exists.  
kafka.common.TopicExistsException: Topic "wyp" already exists.  
at kafka.admin.AdminUtils$.createOrUpdateTopicPartitionAssignmentPathInZK  
(AdminUtils.scala:171)  
at kafka.admin.AdminUtils$.createTopic(AdminUtils.scala:156)  
at kafka.admin.TopicCommand$.createTopic(TopicCommand.scala:86)  
at kafka.admin.TopicCommand$.main(TopicCommand.scala:50)  
at kafka.admin.TopicCommand.main(TopicCommand.scala)
```

2、上述的命令是基于Kafka 0.8.1.1，如果你用的是Kafka 8.x之前版本，请用以下的命令创建topic

```
# bin/kafka-create-topic.sh --zookeeper localhost:2181 \W  
--replica 1 --partition 1 --topic wyp
```

往wyp中发送消息：

```
# bin/kafka-console-producer.sh --broker-list localhost:9092 --topic wyp
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J:See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Hello Kafka
My name is wyp
This is Guowangjiyi blog
```

往wyp中拉取消息：

```
<!-----
User: 过往记忆
Date: 14-6-23
Time: 17:37
bolg:
本文地址：/archives/1045
过往记忆博客，专注于hadoop、hive、spark、shark、flume的技术博客，大量的干货
过往记忆博客微信公共帐号：iteblog_hadoop
----->
```

```
# bin/kafka-console-consumer.sh --zookeeper localhost:2181 W
--topic wyp --from-beginning
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J:See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Hello Kafka
My name is wyp
This is Guowangjiyi blog
Consumed 3 messages
```

由于文章有点长，所以Single node – multiple broker集群和Multiple node – multiple broker集群的部署将会在下一篇文章进行说明，请关注  
过往记忆博客，微信公共帐号：iteblog\_hadoop

本博客文章除特别声明，全部都是原创！  
原创文章版权归过往记忆大数据（[过往记忆](#)）所有，未经许可不得转载。  
本文链接：[【】（）](#)