基于zookeeper 搭建Kafka集成文档

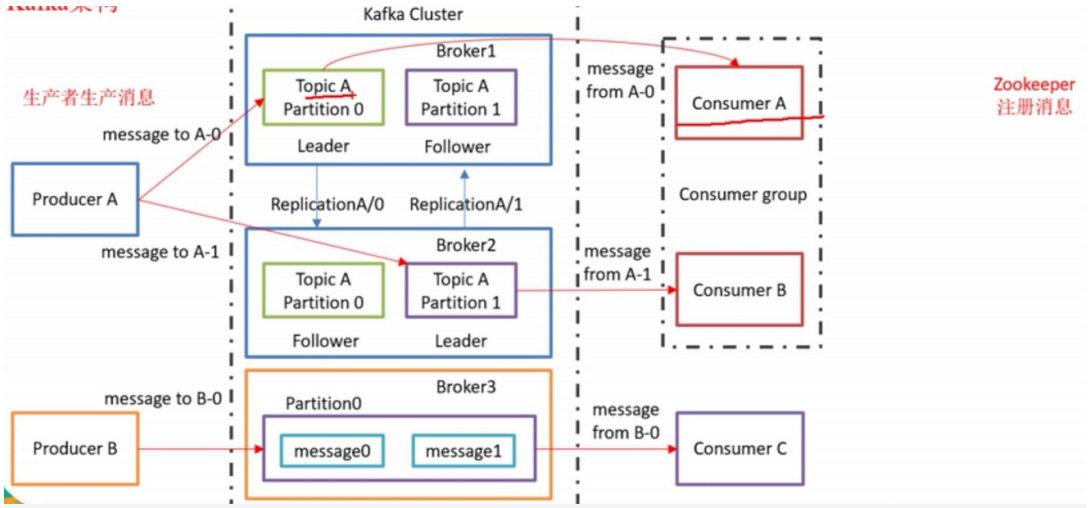
常用消息系统对比

- RabbitMQ Erlang编写，支持多协议 AMQP，XMPP，SMTP，STOMP。支持负载均衡、数据持久化。同时支持Peer-to-Peer和发布/订阅模式
- Redis 基于Key-Value对的NoSQL数据库，同时支持MQ功能，可做轻量级队列服务使用。就入队操作而言，Redis对短消息（小于10KB）的性能比RabbitMQ好，长消息的性能比RabbitMQ差。
- ZeroMQ 轻量级，不需要单独的消息服务器或中间件，应用程序本身扮演该角色，Peer-to-Peer。它实质上是一个库，需要开发人员自己组合多种技术，使用复杂度高
- ActiveMQ JMS实现，Peer-to-Peer，支持持久化、XA事务
- Kafka/Jafka 高性能跨语言的分布式发布/订阅消息系统，数据持久化，全分布式，同时支持在线和离线处理
- MetaQ/RocketMQ 纯Java实现，发布/订阅消息系统，支持本地事务和XA分布式事务

Kafka设计目标

- 高吞吐率 在廉价的商用机器上单机可支持每秒100万条消息的读写
- 消息持久化 所有消息均被持久化到磁盘，无消息丢失，支持消息重放
- 完全分布式 Producer，Broker，Consumer均支持水平扩展
- 同时满足适应在线流处理和离线批处理

Kafka框架



准备工作

1. win10中下载jdk并安装

下载地址：

https://www.oracle.com/technetwork/cn/java/javase/downloads/jdk8-downloads-2133151-zhs.html

2. zookeeper安装配置

运行kafka为什么要依赖zookeeper：
需要一个地方存元信息。zookeeper又是分布式，做配置管理比较好的。于是就用上了。
没zookeeper的话，也需要一套机制存储元数据和交换集群信息的工具。

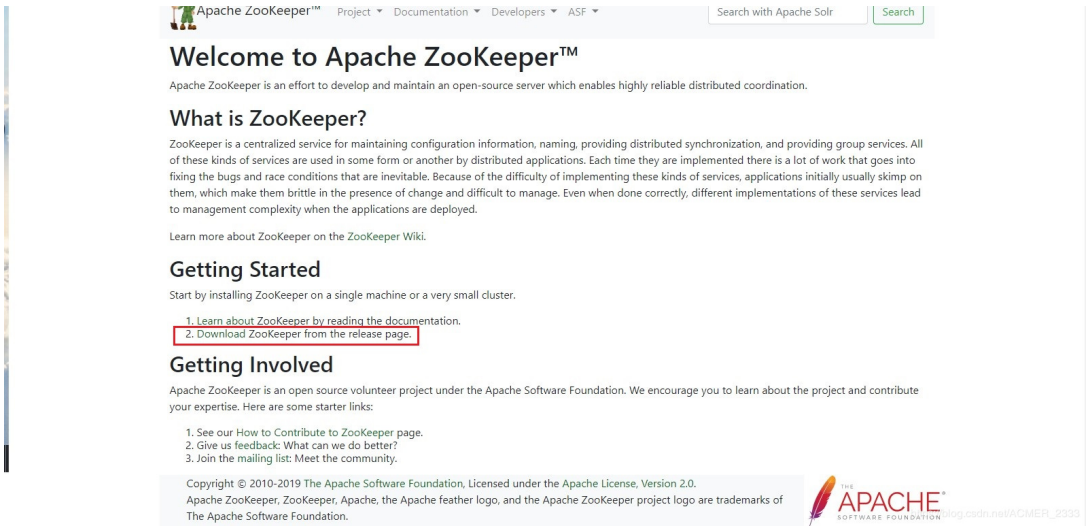
kafka的官方文档有说明。zookeeper是为了解决分布式一致性问题的工具。

至于kafka为什么使用zk，你首先要知道zk的作用，作为去中心化的集群模式。
需要要消费者知道现在那些生产者（对于消费者而言，kafka就是生产者）是可用的。

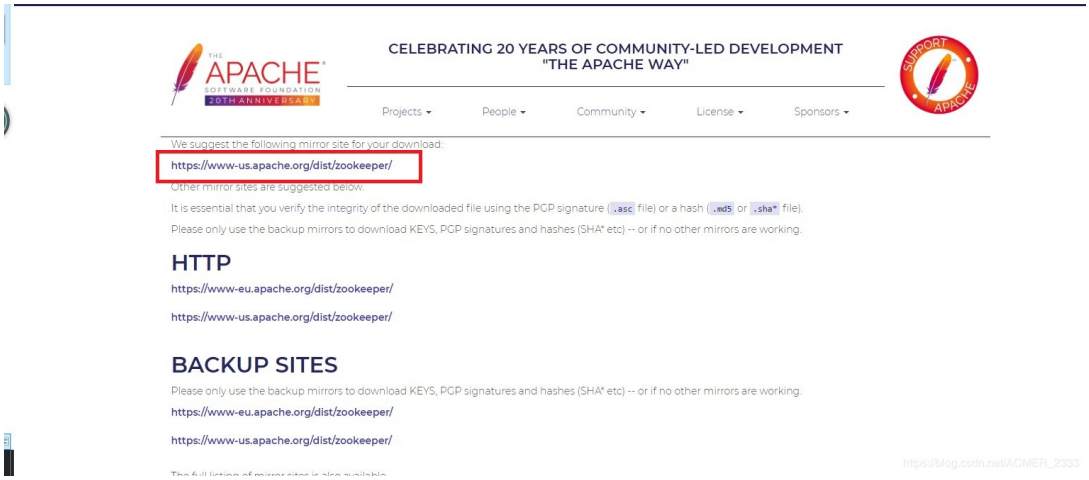
如果没了zk消费者如何知道，如果每次消费者在消费之前都去尝试连接生产者测试下是否连接成功，那么效率问题怎么解决。

所以kafka需要zk，在kafka的设计中就依赖了zk了。

zookeeper下载地址：https://zookeeper.apache.org/releases.html



点击download连接进入到下载界面：



选择对应的版本：

下载完之后，解压。解压之后的目录为:

安装:

下载后，解压放在目录D:（本文所用的目录）下，关于zookeeper以及kafka的目录，路径中最好不要出现空格，比如D:\Program Files，尽量别用，运行脚本时会有问题。

①进入zookeeper的相关设置所在的文件目录，例如本文的：D:\zookeeper-3.4.14\conf

②将"zoo_sample.cfg"复制一份并命名为"zoo.cfg"

③打开zoo.cfg(至于使用什么编辑器，根据自己喜好选即可)，找到并编辑：

将 dataDir=/tmp/zookeeper 改成 D:/zookeeper-3.4.14/data或 D:\zookeeper-3.4.14\data(路径仅为示例，具体可根据需要配置)

这里注意，路径要么是”/“分割，要么是转义字符”\”，这样会生成正确的路径(层级，子目录)。

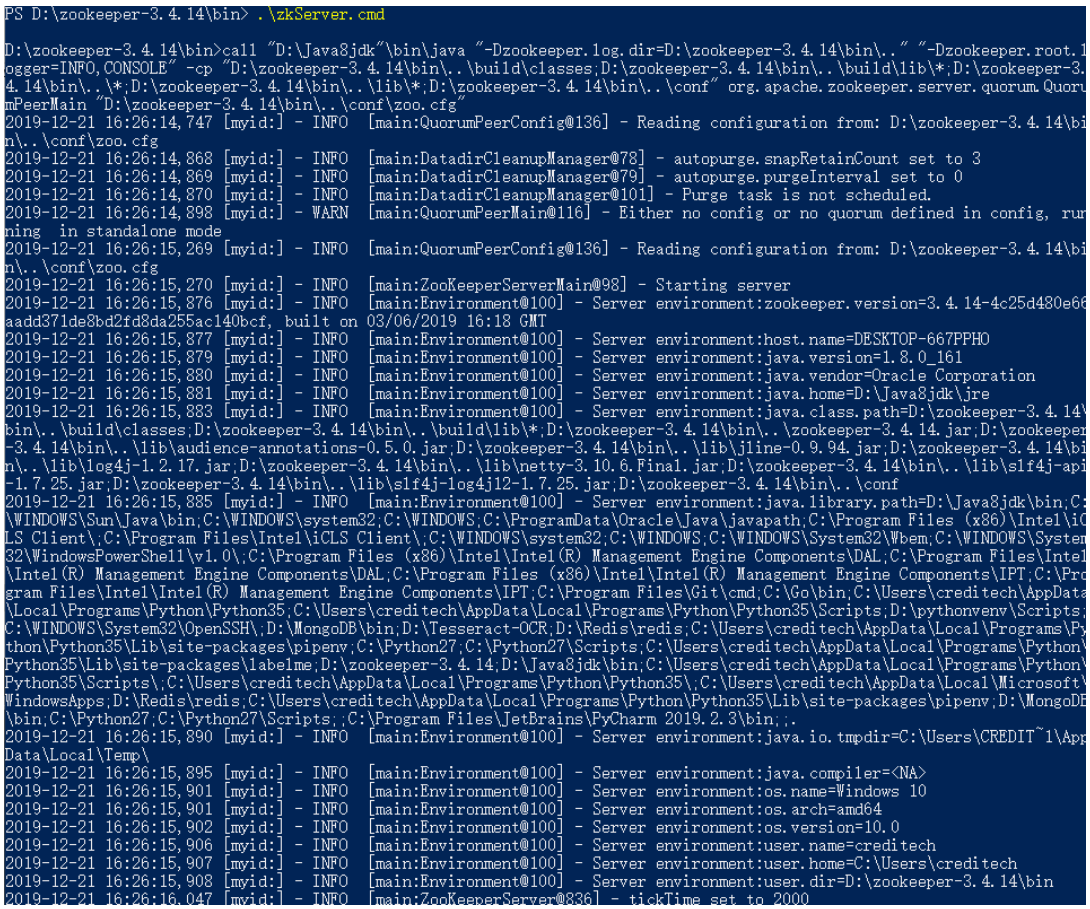
④与配置jre类似，在系统环境变量中添加：

```
a.系统变量中添加ZOOKEEPER_HOME=D:\zookeeper-3.4.14

b.编辑系统变量中的path变量，增加%ZOOKEEPER_HOME%\bin
```

⑤在zoo.cfg文件中修改默认的Zookeeper端口(默认端口2181)

⑥打开PowerShell窗口，输入zkserver.cmd，运行Zookeeper，运行结果如下：



恭喜，Zookeeper已经安装完成，已在2181端口运行。

3.安装运行kafka
kafka下载地址：http://kafka.apache.org/downloads

要下载Binary downloads这个类型，不要下载源文件，这种方便使用。下载后，解压放在D:\目录下。

①进入kafka配置文件所在目录，D:\kafka2.3.1\kafka_2.12-2.3.1\config

②编辑文件”server.properties”，找到并编辑：
将
log.dirs=/tmp/kafka-logs 改成 log.dirs=D:/kafka2.3.1/kafka_2.12-2.3.1/kafka-logs 或者 D:\kafka2.3.1\kafka_2.12-2.3.1\kafka-logs

同样注意：路径要么是”/“分割，要么是转义字符”\”，这样会生成正确的路径(层级，子目录)。错误路径情况可自行尝试，文件夹名为这种形式：bigdatakafka_2.11-0.9.0.1kafka-logs

③在server.properties文件中，zookeeper.connect=localhost:2181代表kafka所连接的zookeeper所在的服务器IP以及端口，可根据需要更改。本文在同一台机器上使用，故不用修改。

④kafka会按照默认配置，在9092端口上运行，并连接zookeeper的默认端口2181。

还有一点特别注意：在kafka安装目录中找到bin\windows目录中的kafka-run-class.bat找到179行为%CLASSPATH%加上双引号。不然运行kafka的时候回提示报错：错误: 找不到或无法加载主类 Files\Java\jdk1.7.0_80\lib;

#运行kafka并一个简单的例子
提示：请确保启动kafka服务器前，Zookeeper实例已经在运行，因为kafka的运行是需要zookeeper这种分布式应用程序协调服务。

①进入kafka安装目录D:\bigdata\kafka_2.11-0.9.0.1

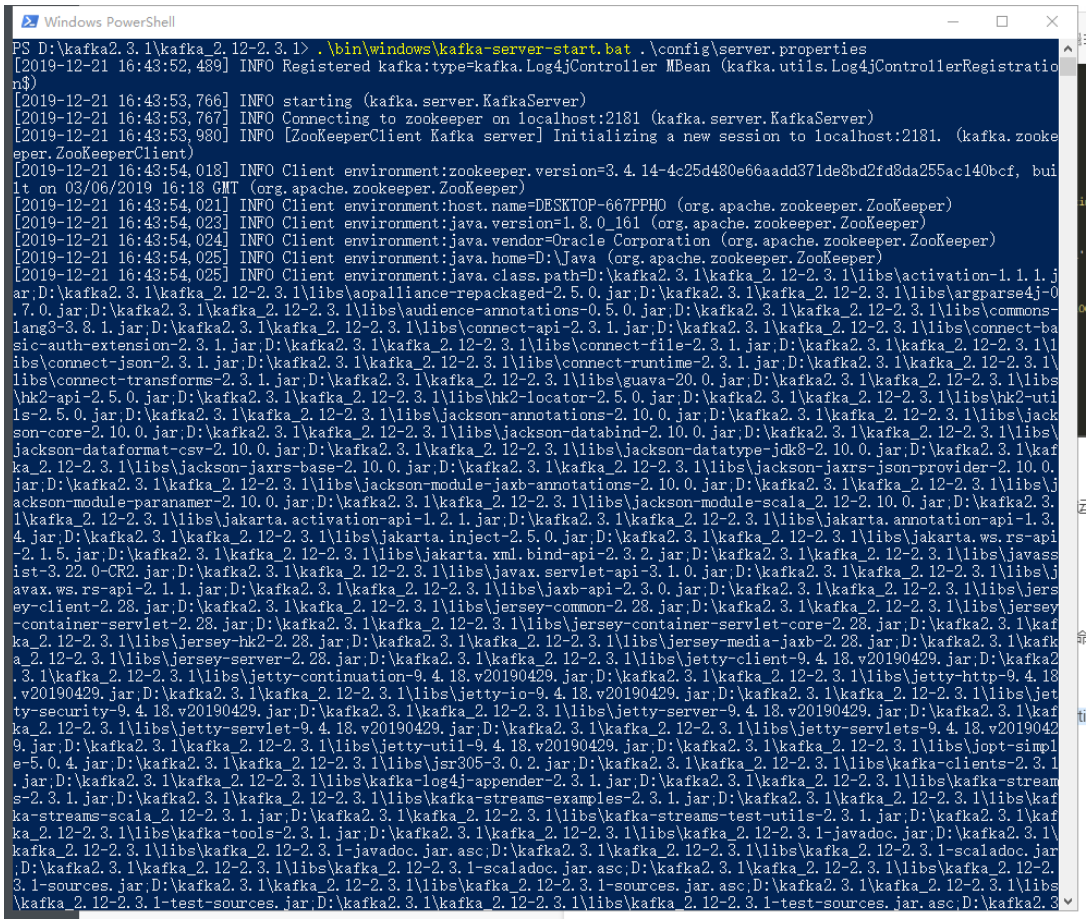
②按下shift+鼠标右键，选择”在此处打开命令窗口”，打开命令行。

③在命令行中输入：

```
.\bin\windows\kafka-server-start.bat .\config\server.properties
```

回车。

④正确运行的情况为：



到目前为止，zookeeper以及kafka都已正确运行。保持运行状态，不要关闭。

创建主题

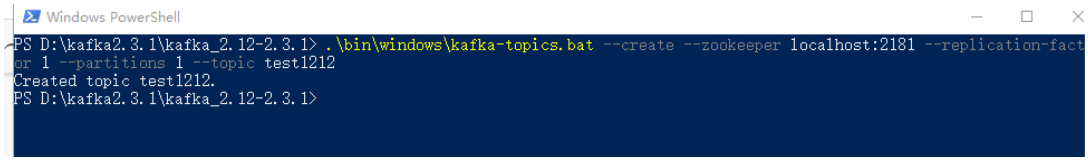
①创建主题，命名为”test1221”，replicationfactor=1(因为只有一个kafka服务器在运行)。可根据集群中kafka服务器个数来修改replicationfactor的数量，以便提高系统容错性等。

②在D:\kafka2.3.1\kafka_2.12-2.3.1\bin\windows目录下打开新的命令行

③输入命令：

```
.\bin\windows\kafka-topics.bat --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic test1212
```

运行



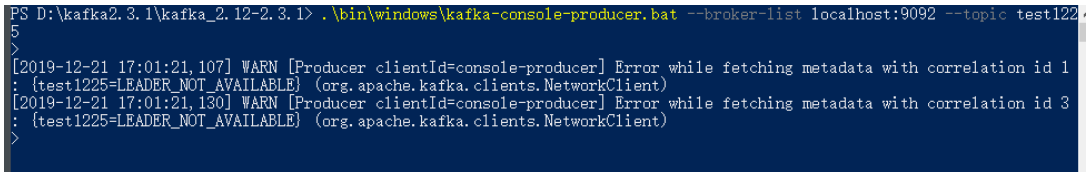
现在这样就创建了一个topic ， 这个窗口可以关闭了。

接下来 创建生产者(producer)和消费者(consumer)

①在D:\kafka2.3.1\kafka_2.12-2.3.1\bin\windows目录下打开新的命令行。

②输入命令，启动producer：

```
.\bin\windows\kafka-console-producer.bat --broker-list localhost:9092 --topic test1225
```



该窗口不要关闭。

③在同目录下打开新的命令行。

④输入命令，启动consumer：

```
.\bin\windows\kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic test1212 --from-beginning
```

####基于Console 在bin目录下，后缀 .sh
创建Topic

```
bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic test
```

Producer发送消息

```
bin/kafka-console-producer.sh --broker-list localhost:9092 --topic test
```

在控制台输入要发送的消息：

```
This is a message

This is another message
```

Consumer接收消息

```
bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic test --from-beginning
```

输入命令后可以看到控制台输出了刚才的消息：

```
This is a message
This is another message
```