

单台 windows 搭建 zookeeper 和 kafka 集群

参考链接：

zookeeper 入门 (1) 在单机上实现 ZooKeeper 伪机群/伪集群部署：

<http://blog.csdn.net/poechant/article/details/6633923>

一台机器上安装 zookeeper 和 kafka 集群：

<http://blog.csdn.net/u013244038/article/details/53938997>

操作系统版本：windows 10

zookeeper 版本：3.4.10

kafka 版本：2.12-0.10.2.1

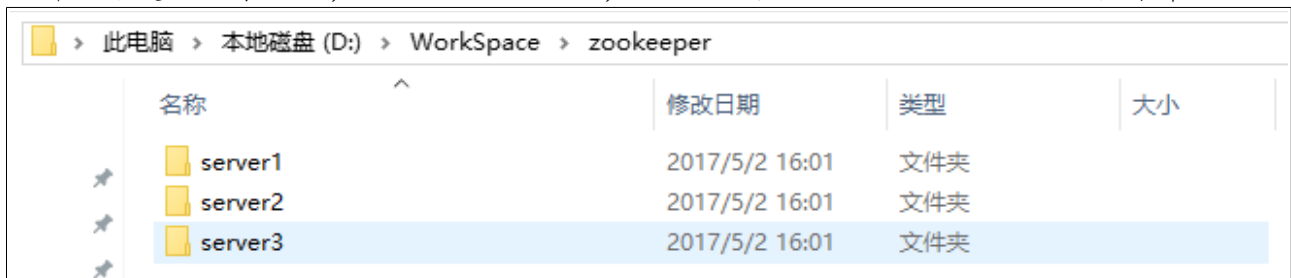
我们的目标是在一台机器上部署三个 zookeeper 的 server，并在此基础上部署三个 zookeeper 的 server。

1.zookeeper 的部署

zookeeper 的部署采用伪集群模式，即在一台电脑上部署多个 server。

1.1.建立文件夹，存放项目

在你喜欢的位置（我的目录为 D:\WorkSpace）建立一个名为 zookeeper 的文件夹，并在该文件夹下建立三个目录，分别名为 server1，server2 和 server3。完成之后如下图所示：



此电脑 > 本地磁盘 (D:) > WorkSpace > zookeeper				
名称	修改日期	类型	大小	
server1	2017/5/2 16:01	文件夹		
server2	2017/5/2 16:01	文件夹		
server3	2017/5/2 16:01	文件夹		

1.2.下载 zookeeper

下载地址：<http://apache.fayea.com/zookeeper/>。或者直接去 zookeeper 官网自己找对应的版本也是一样的。

< > ↻
 ① apache.fayea.com/zookeeper/

ZooKeeper Releases

Please make sure you're downloading from [a nearby mirror site](#), not from [www.apache.org](#).

We suggest downloading the current [stable](#) release.

Older releases are available from the [archives](#).

Name	Last modified	Size	Description
Parent Directory		-	
bookkeeper/	2015-02-18 02:50	-	
current/	2017-03-30 13:03	-	
stable/	2017-03-30 13:03	-	
zookeeper-3.3.6/	2015-02-18 02:50	-	
zookeeper-3.4.10/	2017-03-30 13:03	-	
zookeeper-3.4.6/	2016-01-11 01:11	-	
zookeeper-3.4.8/	2016-02-21 04:41	-	
zookeeper-3.4.9/	2016-09-03 12:28	-	
zookeeper-3.5.0-alpha/	2015-02-18 02:50	-	
zookeeper-3.5.1-alpha/	2015-08-31 12:36	-	
zookeeper-3.5.2-alpha/	2016-07-21 01:09	-	
zookeeper-3.5.3-beta/	2017-04-17 11:32	-	
HEADER.html	2015-12-16 23:31	420	

1.3.解压文件并创建一些目录

将下载好的压缩包解压到 server1 目录下，并在 server1 目录下建三个文件夹，分别名为 data，dataLog 和 logs。完成之后应该是这样的。

此电脑 > 本地磁盘 (D:) > Workspace > zookeeper > server1

名称	修改日期	类型	大小
data	2017/5/2 16:12	文件夹	
dataLog	2017/5/2 16:12	文件夹	
logs	2017/5/2 16:01	文件夹	
zookeeper-3.4.10	2017/5/2 16:01	文件夹	

注意：压缩包解压的时候直接选【解压到当前文件夹】即可。所以上图中的 zookeeper-3.4.10 文件夹打开之后应该是这样的：

此电脑 > 本地磁盘 (D:) > WorkSpace > zookeeper > server1 > zookeeper-3.4.10				
名称	修改日期	类型	大小	
bin	2017/5/2 16:01	文件夹		
conf	2017/5/2 16:06	文件夹		
contrib	2017/5/2 16:01	文件夹		
dist-maven	2017/5/2 16:01	文件夹		
docs	2017/5/2 16:01	文件夹		
lib	2017/5/2 16:01	文件夹		
recipes	2017/5/2 16:01	文件夹		
src	2017/5/2 16:01	文件夹		
build.xml	2017/3/23 18:14	XML 文档	83 KB	
ivy.xml	2017/3/23 18:14	XML 文档	6 KB	
ivysettings.xml	2017/3/23 18:14	XML 文档	2 KB	
LICENSE.txt	2017/3/23 18:14	文本文档	12 KB	
NOTICE.txt	2017/3/23 18:14	文本文档	4 KB	
README.txt	2017/3/23 18:14	文本文档	2 KB	
README_packaging.txt	2017/3/23 18:14	文本文档	2 KB	
zookeeper-3.4.10.jar	2017/3/23 19:24	Executable Jar File	1,423 KB	
zookeeper-3.4.10.jar.asc	2017/3/23 19:28	ASC 文件	1 KB	
zookeeper-3.4.10.jar.md5	2017/3/23 19:24	MD5 文件	1 KB	
zookeeper-3.4.10.jar.sha1	2017/3/23 19:24	SHA1 文件	1 KB	

1.4.在 server1/data 目录下，创建一个叫做 myid 的文件。

注意，这个文件不需要任何扩展名，所以显示的形式如下图所示：

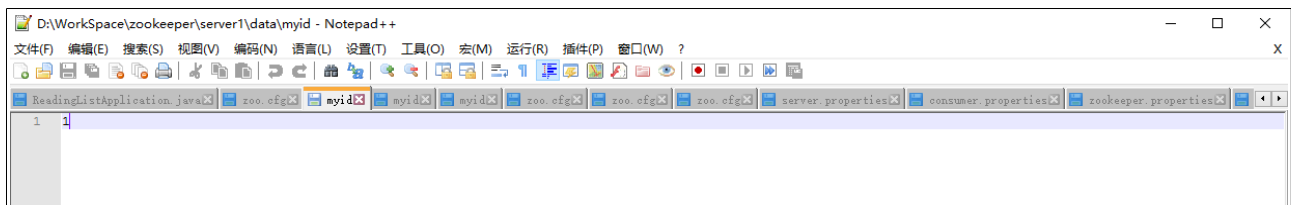
此电脑 > 本地磁盘 (D:) > WorkSpace > zookeeper > server1 > data				
名称	修改日期	类型	大小	
version-2	2017/5/3 10:22	文件夹		
myid	2017/5/2 16:02	文件	1 KB	

如果你看到的 myid 文件不是这样的，说明你的扩展名没有去除。在 windows 系统下，文件扩展名默认是被隐藏的。在【查看】选项卡里勾选【文件扩展名】，你就能看到被隐藏的扩展名了。

A screenshot of a Windows File Explorer window. The top ribbon has tabs for '文件' (File), '主页' (Home), '共享' (Share), and '查看' (View), with '查看' being the active tab. The address bar shows the path: 此电脑 > 本地磁盘 (D:) > WorkSpace > zookeeper > server1 > data. On the left, the '快速访问' (QuickTime) sidebar is visible with '桌面' (Desktop) and '下载' (Downloads) links. The main file list has columns for '名称' (Name), '修改日期' (Modified Date), '类型' (Type), and '大小' (Size). It contains two entries: 'version-2' (文件夹) and 'myid' (文件). The 'myid' file is selected and highlighted in blue.

1.5.打开 myid 文件，在里面添加对应的 id

因为我们目前配置的是 server1，所以添加 id 为 1，然后关闭即可。



这里打开 myid 文件使用的编辑器是 notepad++，如果没有安装的话，可以下载一个安装一下。下载地址：<https://notepad-plus-plus.org/>。



1.6.修改配置文件

在 server1/zookeeper-3.4.10/conf 目录下，新建 zoo.cfg 文件，加入内容如下：

```
tickTime=2000
initLimit=5
syncLimit=2
dataDir=D:/WorkSpace/zookeeper/server1/data
dataLogDir=D:/WorkSpace/zookeeper/server1/dataLog
clientPort=2181
server.1=127.0.0.1:2888:3888
server.2=127.0.0.1:2889:3889
server.3=127.0.0.1:2890:3890
```

注意，dataDir 和 dataLogDir 中的 D:/WorkSpace 目录需要改成你自己的目录。

这里对 zoo.cfg 中的内容进行简要地说明：

tickTime 为客户端与服务器之间或服务器互相之间的心跳间隔，以 ms 为单位。

initLimit 为 leader 与 follower 进行初始连接时最多能够容忍的心跳数。

syncLimit 为 follower 与 leader 请求和应答之间最多能容忍的心跳数。

dataDir 为数据文件目录。

dataLogDir 为日志文件目录。

clientPort 为客户端连接服务器的端口，zookeeper 监听此接口接收客户端访问请求。

最后三行为集群中每个服务器的 ip 和端口号。前一个端口为 leader 和 follower 通信端口，后一个端口为选举新 leader 时所用的端口。一般来说，在多台电脑的情况下，每台服务器上的端口配置是一样的。不过我们现在使用同一台电脑配置三个 server，为了避免冲突，只能把它们改成不一样。

1.7.复制 server1 中的内容到 server2 和 server3 中，并修改相应配置

将 server1 中的内容分别复制到 server2 和 server3 中，并修改/data/myid 文件和/zookeeper-3.4.10/conf/zoo.cfg 文件。
myid 文件只需要将其中的内容改成服务器的序号即可。
zoo.cfg 文件需要修改 dataDir，dataLogDir 和 clientPort。修改后的三个文件如下：
server1:

```
tickTime=2000
initLimit=5
syncLimit=2
dataDir=D:/WorkSpace/zookeeper/server1/data
dataLogDir=D:/WorkSpace/zookeeper/server1/dataLog
clientPort=2181
server.1=127.0.0.1:2888:3888
server.2=127.0.0.1:2889:3889
server.3=127.0.0.1:2890:3890
```

server2:

```
tickTime=2000
initLimit=5
syncLimit=2
dataDir=D:/WorkSpace/zookeeper/server2/data
dataLogDir=D:/WorkSpace/zookeeper/server2/dataLog
clientPort=2182
server.1=127.0.0.1:2888:3888
server.2=127.0.0.1:2889:3889
server.3=127.0.0.1:2890:3890
```

server3:

```
tickTime=2000
initLimit=5
syncLimit=2
dataDir=D:/WorkSpace/zookeeper/server3/data
dataLogDir=D:/WorkSpace/zookeeper/server3/dataLog
clientPort=2183
server.1=127.0.0.1:2888:3888
server.2=127.0.0.1:2889:3889
server.3=127.0.0.1:2890:3890
```

1.8.启动服务器

分别进入三个 server 的 /zookeeper-3.4.10/bin 目录下，运行：zkServer start。结果如下图所示：

```
选择C:\WINDOWS\system32\cmd.exe - zkServer
at java.net.SocksSocketImpl.connect(SocksSocketImpl.java:392)
at java.net.Socket.connect(Socket.java:589)
at org.apache.zookeeper.server.quorum.QuorumCnxManager.connectOne(QuorumCnxManager.java:562)
at org.apache.zookeeper.server.quorum.QuorumCnxManager.connectAll(QuorumCnxManager.java:614)
at org.apache.zookeeper.server.quorum.FastLeaderElection.lookForLeader(FastLeaderElection.java:843)
at org.apache.zookeeper.server.quorum.QuorumPeer.run(QuorumPeer.java:913)
2017-05-03 14:58:47,829 [myid:1] - INFO [QuorumPeer[myid=1]/0:0:0:0:0:0:0:2181:QuorumPeer$QuorumServer@167] - Resolved
d hostname: 127.0.0.1 to address: /127.0.0.1
2017-05-03 14:58:48,859 [myid:1] - WARN [QuorumPeer[myid=1]/0:0:0:0:0:0:0:2181:QuorumCnxManager@588] - Cannot open channel to 3 at election address /127.0.0.1:38
90
java.net.ConnectException: Connection refused: connect
at java.net.DualStackPlainSocketImpl.waitForConnect(Native Method)
at java.net.DualStackPlainSocketImpl.socketConnect(DualStackPlainSocketImpl.java:85)
at java.net.AbstractPlainSocketImpl.doConnect(AbstractPlainSocketImpl.java:350)
at java.net.AbstractPlainSocketImpl.connectToAddress(AbstractPlainSocketImpl.java:206)
at java.net.AbstractPlainSocketImpl.connect(AbstractPlainSocketImpl.java:188)
at java.net.PlainSocketImpl.connect(PlainSocketImpl.java:172)
at java.net.SocksSocketImpl.connect(SocksSocketImpl.java:392)
at java.net.Socket.connect(Socket.java:589)
at org.apache.zookeeper.server.quorum.QuorumCnxManager.connectOne(QuorumCnxManager.java:562)
at org.apache.zookeeper.server.quorum.QuorumCnxManager.connectAll(QuorumCnxManager.java:614)
at org.apache.zookeeper.server.quorum.FastLeaderElection.lookForLeader(FastLeaderElection.java:843)
at org.apache.zookeeper.server.quorum.QuorumPeer.run(QuorumPeer.java:913)
2017-05-03 14:58:49,003 [myid:1] - INFO [QuorumPeer[myid=1]/0:0:0:0:0:0:0:2181:QuorumPeer$QuorumServer@167] - Resolved
d hostname: 127.0.0.1 to address: /127.0.0.1
2017-05-03 14:58:49,019 [myid:1] - INFO [QuorumPeer[myid=1]/0:0:0:0:0:0:0:2181:FastLeaderElection@852] - Notification
time out: 6400
中文(简体) - 手心输入法 半 :
```

第一个 server 启动的时候会报错，这是由于它想要和其他两个 server 进行通信，而另外两个 server 还没有搭建起来。不用理会这个错误，继续打开新的 cmd 窗口，启动其他 server 就好。注意，原来的窗口不要关！所有 server 都启动了之后，显示如下：

```
C:\WINDOWS\system32\cmd.exe - zkServer
2017-05-03 15:00:43,207 [myid:2] - INFO [QuorumPeer[myid=2]/0:0:0:0:0:0:0:2182:Environment@100] - Server environment:
java.io.tmpdir=C:\Users\YUNTIN_1\AppData\Local\Temp\
2017-05-03 15:00:43,207 [myid:2] - INFO [QuorumPeer[myid=2]/0:0:0:0:0:0:0:2182:Environment@100] - Server environment:
java.compiler=NA>
2017-05-03 15:00:43,207 [myid:2] - INFO [QuorumPeer[myid=2]/0:0:0:0:0:0:0:2182:Environment@100] - Server environment:
os.name=Windows 10
2017-05-03 15:00:43,207 [myid:2] - INFO [QuorumPeer[myid=2]/0:0:0:0:0:0:0:2182:Environment@100] - Server environment:
os.arch=amd64
2017-05-03 15:00:43,207 [myid:2] - INFO [QuorumPeer[myid=2]/0:0:0:0:0:0:0:2182:Environment@100] - Server environment:
os.version=10.0
2017-05-03 15:00:43,207 [myid:2] - INFO [QuorumPeer[myid=2]/0:0:0:0:0:0:0:2182:Environment@100] - Server environment:
user.name=yuntingzhang
2017-05-03 15:00:43,207 [myid:2] - INFO [QuorumPeer[myid=2]/0:0:0:0:0:0:0:2182:Environment@100] - Server environment:
user.home=C:\Users\yuntingzhang
2017-05-03 15:00:43,207 [myid:2] - INFO [QuorumPeer[myid=2]/0:0:0:0:0:0:0:2182:Environment@100] - Server environment:
user.dir=D:\WorkSpace\zookeeper\server2\zookeeper-3.4.10\bin
2017-05-03 15:00:43,207 [myid:2] - INFO [QuorumPeer[myid=2]/0:0:0:0:0:0:0:2182:ZooKeeperServer@173] - Created server
with tickTime 2000 minSessionTimeout 4000 maxSessionTimeout 40000 datadir D:\WorkSpace\zookeeper\server2\dataLog\version
-2 snapdir D:\WorkSpace\zookeeper\server2\data\version-2
2017-05-03 15:00:43,207 [myid:2] - INFO [QuorumPeer[myid=2]/0:0:0:0:0:0:0:2182:Follower@64] - FOLLOWING - LEADER ELEC
TION TOOK - 59529
2017-05-03 15:00:43,207 [myid:2] - INFO [QuorumPeer[myid=2]/0:0:0:0:0:0:0:2182:QuorumPeer$QuorumServer@167] - Resolved
d hostname: 127.0.0.1 to address: /127.0.0.1
2017-05-03 15:00:43,291 [myid:2] - INFO [QuorumPeer[myid=2]/0:0:0:0:0:0:0:2182:Learner@332] - Getting a diff from the
leader 0x20000002a
2017-05-03 15:00:50,003 [myid:2] - WARN [QuorumPeer[myid=2]/0:0:0:0:0:0:0:2182:Follower@118] - Got zxid 0x300000001 e
xpected 0x1
2017-05-03 15:00:50,003 [myid:2] - INFO [SyncThread:2:FileTxnLog@203] - Creating new log file: log.300000001
中文(简体) - 手心输入法 半 :
```

如果报 Invalid arguments, exiting abnormally 的异常的话，只需要在 zkServer start 命令中去掉 start，即输入 zkServer 即可。

1.9.连接客户端

进入任意一个 zookeeper-3.4.10 目录下，用以下命令启动客户端：zkCli -server 127.0.0.1:2181，显示如下结果：

```
C:\WINDOWS\system32\cmd.exe - zkCli -server 127.0.0.1:2181
nnection to server 127.0.0.1/127.0.0.1:2181. Will not attempt to authenticate using SASL (unknown error)
2017-05-03 15:25:32,626 [myid:] - INFO [main-SendThread(127.0.0.1:2181):ClientCnxn$SendThread@876] - Socket connection
established to 127.0.0.1/127.0.0.1:2181, initiating session
[Line support is enabled]
[zk: 127.0.0.1:2181(CONNECTING) 0] 2017-05-03 15:26:02,638 [myid:] - WARN [main-SendThread(127.0.0.1:2181):ClientCnxn$S
endThread@1108] - Client session timed out, have not heard from server in 30012ms for sessionid 0x0
2017-05-03 15:26:02,638 [myid:] - INFO [main-SendThread(127.0.0.1:2181):ClientCnxn$SendThread@1156] - Client session ti
med out, have not heard from server in 30012ms for sessionid 0x0, closing socket connection and attempting reconnect
2017-05-03 15:26:04,091 [myid:] - INFO [main-SendThread(127.0.0.1:2181):ClientCnxn$SendThread@1032] - Opening socket co
nnection to server 127.0.0.1/127.0.0.1:2181. Will not attempt to authenticate using SASL (unknown error)
2017-05-03 15:26:04,091 [myid:] - INFO [main-SendThread(127.0.0.1:2181):ClientCnxn$SendThread@876] - Socket connection
established to 127.0.0.1/127.0.0.1:2181, initiating session
2017-05-03 15:26:34,093 [myid:] - WARN [main-SendThread(127.0.0.1:2181):ClientCnxn$SendThread@1108] - Client session ti
med out, have not heard from server in 30002ms for sessionid 0x0
2017-05-03 15:26:34,093 [myid:] - INFO [main-SendThread(127.0.0.1:2181):ClientCnxn$SendThread@1156] - Client session ti
med out, have not heard from server in 30002ms for sessionid 0x0, closing socket connection and attempting reconnect
2017-05-03 15:26:35,748 [myid:] - INFO [main-SendThread(127.0.0.1:2181):ClientCnxn$SendThread@1032] - Opening socket co
nnection to server 127.0.0.1/127.0.0.1:2181. Will not attempt to authenticate using SASL (unknown error)
2017-05-03 15:26:35,748 [myid:] - INFO [main-SendThread(127.0.0.1:2181):ClientCnxn$SendThread@876] - Socket connection
established to 127.0.0.1/127.0.0.1:2181, initiating session
2017-05-03 15:27:05,754 [myid:] - WARN [main-SendThread(127.0.0.1:2181):ClientCnxn$SendThread@1108] - Client session ti
med out, have not heard from server in 30006ms for sessionid 0x0
2017-05-03 15:27:05,754 [myid:] - INFO [main-SendThread(127.0.0.1:2181):ClientCnxn$SendThread@1156] - Client session ti
med out, have not heard from server in 30006ms for sessionid 0x0, closing socket connection and attempting reconnect
2017-05-03 15:27:07,499 [myid:] - INFO [main-SendThread(127.0.0.1:2181):ClientCnxn$SendThread@1032] - Opening socket co
nnection to server 127.0.0.1/127.0.0.1:2181. Will not attempt to authenticate using SASL (unknown error)
2017-05-03 15:27:07,499 [myid:] - INFO [main-SendThread(127.0.0.1:2181):ClientCnxn$SendThread@876] - Socket connection
established to 127.0.0.1/127.0.0.1:2181, initiating session
中文(简体) - 手心输入法 半 :
```

输入 help, 显示可以执行的指令。

```
C:\WINDOWS\system32\cmd.exe - zkCli -server 127.0.0.1:2181
med out, have not heard from server in 30006ms for sessionid 0x0, closing socket connection and attempting reconnect
2017-05-03 15:27:07,499 [myid:] - INFO [main-SendThread(127.0.0.1:2181):ClientCnxn$SendThread@1032] - Opening socket co
nnection to server 127.0.0.1/127.0.0.1:2181. Will not attempt to authenticate using SASL (unknown error)
2017-05-03 15:27:07,499 [myid:] - INFO [main-SendThread(127.0.0.1:2181):ClientCnxn$SendThread@876] - Socket connection
established to 127.0.0.1/127.0.0.1:2181, initiating session
help
ZooKeeper -server host:port cmd args
  stat path [watch]
  set path data [version]
  ls path [watch]
  delquota [-n|-b] path
  ls2 path [watch]
  setAcl path acl
  setquota -n|-b val path
  history
  redo cmdno
  printwatches on|off
  delete path [version]
  sync path
  listquota path
  rmr path
  get path [watch]
  create [-s] [-e] path data acl
  addauth scheme auth
  quit
  getAcl path
  close
  connect host:port
[zk: 127.0.0.1:2181(CONNECTING) 1]
中文(简体) - 手心输入法 半 :
```

做到这一步, zookeeper 集群的搭建就完成了。接下来我们要在此基础上搭建 kafka 的集群。

2.kafka 的部署

kafka 在设计的时候就依赖于 zookeeper, 对于 kafka 而言, zookeeper 相当于它的文件系统。每个节点的信息都保存在 zookeeper 对应的路径下。

(1) **Broker 注册:** Broker 在 zookeeper 中保存为一个临时节点, 节点的路径是/brokers/ids/[brokerid],每个节点会保存对应 broker 的 IP 以及端口等信息。

(2) **Topic 注册:** 在 kafka 中,一个 topic 会被分成多个区并被分到多个 broker 上, 分区的信息以及 broker 的分布情况都保存在 zookeeper 中, 根节点路径为/brokers/topics,每个 topic 都会在 topics 下建立独立的子节点, 每个 topic 节点下都会包含分区以及 broker 的对应信息。

(3) **消费者与分区的对应关系**: 对于每个消费者分组, kafka 都会为其分配一个全局唯一的 Group ID, 分组内的所有消费者会共享该 ID, kafka 还会为每个消费者分配一个 consumer ID, 通常采用 hostname:uuid 的形式。在 kafka 的设计中规定, 对于 topic 的每个分区, 最多只能被一个消费者进行消费, 也就是消费者与分区的关系是一对多的关系。消费者与分区的关系被存储在 zookeeper 中节点的路径为 /consumers/[group_id]/owners/[topic]/[broker_id-partition_id], 该节点的内容就是消费者的 Consumer ID。

(4) **消费者负载均衡**: 消费者服务启动时, 会创建一个属于消费者节点的临时节点, 节点的路径为 /consumers/[group_id]/ids/[consumer_id], 该节点的内容是该消费者订阅的 Topic 信息。每个消费者会对 /consumers/[group_id]/ids 节点注册 Watcher 监听器, 一旦消费者的数量增加或减少就会触发消费者的负载均衡。消费者还会对 /brokers/ids/[brokerid] 节点进行监听, 如果发现服务器的 Broker 服务器列表发生变化, 也会进行消费者的负载均衡。

(5) **消费者的 offset**: 在 kafka 的消费者 API 分为两种 (1) **High Level API**: 由 zookeeper 维护消费者的 offset (2) **Low Level API**: 自己的代码实现对 offset 的维护。由于自己维护 offset 往往比较复杂, 所以多数情况下都是使用 High Level 的 API offset 在 zookeeper 中的节点路径为 /consumers/[group_id]/offsets/[topic]/[broker_id-part_id], 该节点的值就是对应的 offset。
看到这里, 你就会明白 kafka 和 zookeeper 之间的关系了。我们现在就是要在已经配置好的 zookeeper 集群上配置 kafka 集群。

2.1. 建立对应目录

还是找到你喜欢的那个目录 (我的是 D:\WorkSpace), 建立一个叫做 kafka 的文件夹, 里面建立三个子目录, 分别叫做: kafka1, kafka2 和 kafka3。完成之后应该是这个样子:



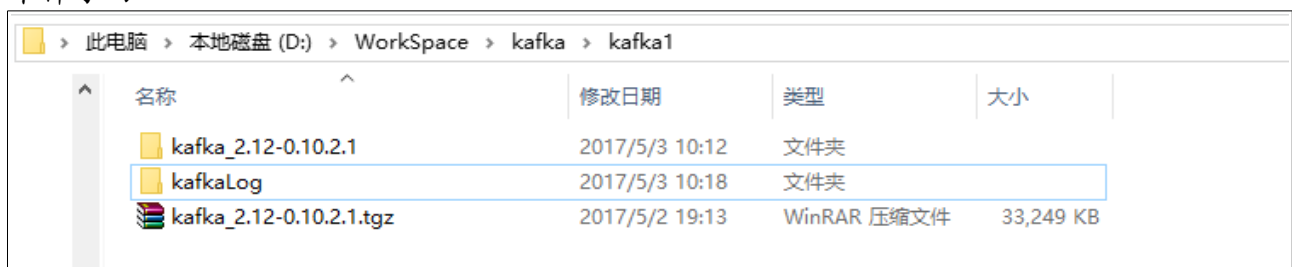
名称	修改日期	类型	大小
kafka1	2017/5/3 10:00	文件夹	
kafka2	2017/5/3 10:04	文件夹	
kafka3	2017/5/3 10:05	文件夹	

2.2. 下载 kafka 并解压到每个文件夹

kafka 下载地址:

https://www.apache.org/dyn/closer.cgi?path=/kafka/0.10.2.1/kafka_2.12-0.10.2.1.tgz

下载之后是一个压缩包, 还是直接解压到 kafka1, kafka2 和 kafka3 这三个文件夹就行。解压完后, 在 kafka1, kafka2 和 kafka3 目录下分别添加一个 kafkaLog 目录。最后的结果应该是这个样子的:



名称	修改日期	类型	大小
kafka_2.12-0.10.2.1	2017/5/3 10:12	文件夹	
kafkaLog	2017/5/3 10:18	文件夹	
kafka_2.12-0.10.2.1.tgz	2017/5/2 19:13	WinRAR 压缩文件	33,249 KB

2.3. 修改配置文件:

分别打开 kafka1, kafka2 和 kafka3 三个目录中的 kafka_2.12-0.10.2.1/config 路径下的 consumer.properties 文件, 将里面的内容改成:

```
zookeeper.connect=127.0.0.1:2181,127.0.0.1:2182,127.0.0.1:2183
group.id=logGroup
```


再打开这三个目录下的 server.properties 文件，按照下面内容进行修改：

kafka1：

```
broker.id=0
log.dirs =D:/WorkSpace/kafka/kafka1/kafkaLog
listeners=PLAINTEXT://127.0.0.1:9092
port=9092
zookeeper.connect=127.0.0.1:2181,127.0.0.1:2182,127.0.0.1:2183
message.max.byte=5242880
default.replication.factor=2
replica.fetch.max.bytes=5242880
```

kafka2：

```
broker.id=1
log.dirs =D:/WorkSpace/kafka/kafka2/kafkaLog
listeners=PLAINTEXT://127.0.0.1:9093
port=9093
zookeeper.connect=127.0.0.1:2181,127.0.0.1:2182,127.0.0.1:2183
message.max.byte=5242880
default.replication.factor=2
replica.fetch.max.bytes=5242880
```

kafka3：

```
broker.id=2
log.dirs =D:/WorkSpace/kafka/kafka3/kafkaLog
listeners=PLAINTEXT://127.0.0.1:9094
port=9094
zookeeper.connect=127.0.0.1:2181,127.0.0.1:2182,127.0.0.1:2183
message.max.byte=5242880
default.replication.factor=2
replica.fetch.max.bytes=5242880
```

2.4.启动 kafka 服务器

进入 D:/WorkSpace/kafka/kafka1/ 目录下，输入

.\bin\windows\kafka-server-start.bat .\config\server.properties 命令，注意不要漏掉命令中的“.”。得到的结果如下图所示：

