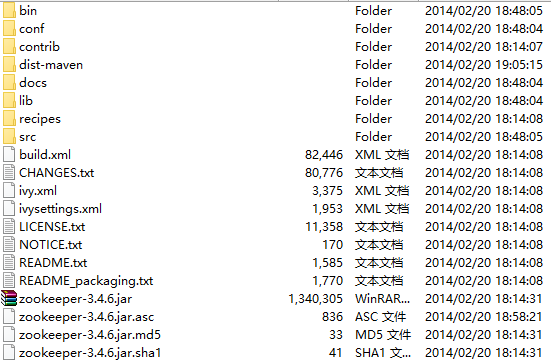
# zookeeper教程

Zookeeper运行模式有三种：单机模式、集群模式和伪集群模式，三种模式差别不大，以下会进行介绍，这里以单机模式为例。Zookeeper使用Java语言编写，运行环境需要Java环境的支持，安装前请先安装1.6及以上版本的JDK，然后到http://mirrors.hust.edu.cn/apache/zookeeper/zookeeper-3.4.6/下载zookeeper安装包zookeeper-3.4.6.tar.gz，这里使用3.4.6版本。注意下载的是部署包，不是源码包，否则启动时会报如下错误：Error: Could not find or load main class org.apache.zookeeper.server.quorum.QuorumPeerMain

## 单机模式

### 1、上传ZooKeeper部署包并解压

将zookeeper-3.4.6.tar.gz上传至/opt目录，我们将zookeeper安装在该目录下，cd到该目录下，使用tar -zxvf zookeeper-3.4.6.tar.gz进行解压，生成如下文件



### **2、配置zoo.cfg**

在confg目录下创建zoo.cfg，并配置如下内容：

tickTime=2000

dataDir=/var/lib/zookeeper/

clientPort=2181

initLimit=5

syncLimit=2

server.1=192.168.85.128:2888:3888

### **3、配置myid文件**

在/var/lib/zookeeper/目录下创建myid文件并在该文件的第一行写上一个数字1。

### **4、启动ZooKeeper并验证**

使用sh zkServer.sh start或（./zkServer.sh start）命令进行启动Zookeeper，可以使用三种方式来验证Zookeeper是否正常启动，如：

[root@localhost bin]# ./zkServer.sh start

JMX enabled by default

Using config: /opt/zookeeper-3.4.6/bin/../conf/zoo.cfg

Starting zookeeper ... STARTED

[root@localhost bin]# ./zkServer.sh status

JMX enabled by default

Using config: /opt/zookeeper-3.4.6/bin/../conf/zoo.cfg

Mode: standalone

[root@localhost bin]# telnet 192.168.85.128 2181

Trying 192.168.85.128...

Connected to 192.168.85.128.

Escape character is '^]'.

stat

Zookeeper version: 3.4.6-1569965, built on 02/20/2014 09:09 GMT

Clients:

/192.168.85.128:44159[0](queued=0,recved=1,sent=0)

Latency min/avg/max: 0/0/0

Received: 2

Sent: 1

Connections: 1

Outstanding: 0

Zxid: 0x0

Mode: standalone

Node count: 4

Connection closed by foreign host.

[root@localhost bin]# echo ruok | nc 192.168.85.128 2181

imok

使用telnet和nc命令验证是否启动成功时需要安装telnet和nc才行，否则会提示找不到命令。另外，启动完后会在bin目录下生成一个zookeeper.out日志文件，如果没有正常启动，可以来查看这个文件，如果要修改这个文件输出位置，请查看“zookeeper log4j日志输出修改”教程。

### **5、停止ZooKeeper**

使用sh zkServer.sh stop停止ZooKeeper服务

[root@localhost bin]# sh zkServer.sh stop

JMX enabled by default

Using config: /opt/zookeeper-3.4.6/bin/../conf/zoo.cfg

Stopping zookeeper ... STOPPED

## 集群模式

按照以上单机模式的步骤，为其他集群机器配置上zoo.cfg和myid文件即可，zoo.cfg文件修改如下：

tickTime=2000

dataDir=/var/lib/zookeeper/

clientPort=2181

initLimit=5

syncLimit=2

server.1=192.168.85.127:2888:3888

server.2=192.168.85.128:2888:3888

server.3=192.168.85.129:2888:3888

集群中所有机器的zoo.cfg文件的内容都应该是一致的。因此最好使用SVN或是GIT把此文件管理起来，确保每个机器都能共享到一份相同的配置。

## 伪集群模式

伪集群是指所有的集群都在同一台机器上，但是还是以集群的特性来对外提供服务。这种模式和集群模式类似，只需修改zoo.cfg配置，如：

tickTime=2000

dataDir=/var/lib/zookeeper/

clientPort=2181

initLimit=5

syncLimit=2

server.1=192.168.85.127:2888:3888

server.2=192.168.85.128:2889:3889

server.3=192.168.85.128:2890:3890

## [Zookeeper Log4j日志输出修改](http://yangyoupeng-cn-fujitsu-com.iteye.com/blog/1922459)

1、修改log4j.properties：   
%ZK\_HOME%conf/log4j.properties

将：zookeeper.root.logger=INFO, CONSOLE    
改成： zookeeper.root.logger=INFO,ROLLINGFILE

2、修改bin/zkEvn.sh文件：

将：

if [ "x${ZOO\_LOG4J\_PROP}" = "x" ]

then

    ZOO\_LOG4J\_PROP="INFO,CONSOLE"

fi    
改成：

if [ "x${ZOO\_LOG4J\_PROP}" = "x" ]

then

    ZOO\_LOG4J\_PROP="INFO,ROLLINGFILE"

fi

将：

if [ "x${ZOO\_LOG\_DIR}" = "x" ]

then

ZOO\_LOG\_DIR="."

fi

改成：

if [ "x${ZOO\_LOG\_DIR}" = "x" ]

then

ZOO\_LOG\_DIR="/opt/zookeeper-3.4.6/logs"

fi

这样再启动zkSever.sh start的时候 就会生成在/opt/zookeeper-3.4.6/logs目录中生成日志了

## zookeeper客户端脚本

### 1、启动ZooKeeper客户端

进入ZooKeeper的bin目录之后，执行如下命令：**sh zkCli.sh**

当看到如下输出信息时，表示已经成功连接上本地的ZooKeeper服务器了：



注意，上面的命令没有显示地指定ZooKeeper服务器地址，那么默认是连接本地的ZooKeeper服务器。如果希望连接指定的ZooKeeper服务器，可以通过如下方式实现：**sh zkCli.sh -server [ip]:[port]**

### **2、创建节点**

使用如下命令可以创建一个ZooKeeper节点：

create [-s] [-e] path data acl

其中，-s或-e分别指定节点特性：顺序或临时节点。默认情况下，即不添加-s或-e参数的，创建的是持久节点。create命令的最后一个参数是acl，它是用来进行权限控制的，缺省情况下，不做任何权限控制。

**示例：**

[zk: localhost:2181(CONNECTED) 0] **create /zk-book 123**

Created /zk-book

执行完上面的命令，就在ZooKeeper的根节点下创建了一个叫作/zk-book的节点，并且节点的数据内容是“123”。

### **3、读取子节点**

使用ls命令，可以列出ZooKeeper节点下的所有子节点。注意：这个命令只能看到指定节点下第一级的所有子节点。语法如下：

ls path [watch]

其中，path表示的是指定数据节点的路径，第一次部署的ZooKeeper集群，默认在根节点“/”下面有一个叫作/zookeeper的保留节点。

示例：

[zk: localhost:2181(CONNECTED) 1] **ls /**

[zk-book, zookeeper]

### **4、读取节点数据**

使用get命令，可以获取ZooKeeper指定节点的数据内容和属性信息。用法如下：

[zk: localhost:2181(CONNECTED) 2] **get /zk-book**

123

cZxid = 0x2

ctime = Sat Apr 08 07:06:05 PDT 2017

mZxid = 0x2

mtime = Sat Apr 08 07:06:05 PDT 2017

pZxid = 0x2

cversion = 0

dataVersion = 0

aclVersion = 0

ephemeralOwner = 0x0

dataLength = 3

numChildren = 0

从上面的输出信息中，我们可以看到，第一行是节点/zk-book的数据内容，其他几行则是创建该节点的事务ID（cZxid）、最后一次更新该节点的事务ID（mZxid）和最后一次更新该节点的时间（mtime）等属性。

### **5、更新节点数据内容**

使用set命令，可以更新指定节点的数据内容，用法如下：set path [version]

其中，data就是更新的新内容。注意，set命令后面还有一个version参数，在ZooKeeper中，节点的数据是有版本概念的，这个参数用于指定本次更新操作是基于ZNode的哪一个数据版本进行的。

**示例：**

[zk: localhost:2181(CONNECTED) 3] **set /zk-book 456**

cZxid = 0x2

ctime = Sat Apr 08 07:06:05 PDT 2017

mZxid = 0x3

mtime = Sat Apr 08 07:08:35 PDT 2017

pZxid = 0x2

cversion = 0

dataVersion = 1

aclVersion = 0

ephemeralOwner = 0x0

dataLength = 3

numChildren = 0

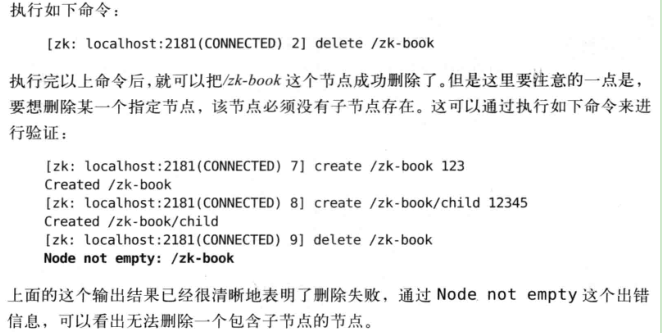
执行以上命名后，节点/zk-book的数据内容就已经被更新成“456”了。另外在输出信息中，dataVersion的值有原来的0变成了1，这是因为刚才的更新操作导致该节点的数据版本也发生了变更。

### **5、删除节点**

使用delete命令，可以删除ZooKeeper上的指定节点，用法如下：

delete path [version]

次命令中的version参数和set命令中的version参数的作用是一致的。



### 完整示例

[root@localhost bin]# **sh zkCli.sh**

Connecting to localhost:2181

2017-04-08 04:37:22,748 [myid:] - INFO [main:Environment@100] - Client environment:zookeeper.version=3.4.6-1569965, built on 02/20/2014 09:09 GMT

2017-04-08 04:37:22,753 [myid:] - INFO [main:Environment@100] - Client environment:host.name=localhost

2017-04-08 04:37:22,754 [myid:] - INFO [main:Environment@100] - Client environment:java.version=1.7.0\_79

2017-04-08 04:37:22,758 [myid:] - INFO [main:Environment@100] - Client environment:java.vendor=Oracle Corporation

2017-04-08 04:37:22,758 [myid:] - INFO [main:Environment@100] - Client environment:java.home=/opt/java/jdk1.7.0\_79/jre

2017-04-08 04:37:22,758 [myid:] - INFO [main:Environment@100] - Client environment:java.class.path=/opt/zookeeper-3.4.6/bin/../build/classes:/opt/zookeeper-3.4.6/bin/../build/lib/\*.jar:/opt/zookeeper-3.4.6/bin/../lib/slf4j-log4j12-1.6.1.jar:/opt/zookeeper-3.4.6/bin/../lib/slf4j-api-1.6.1.jar:/opt/zookeeper-3.4.6/bin/../lib/netty-3.7.0.Final.jar:/opt/zookeeper-3.4.6/bin/../lib/log4j-1.2.16.jar:/opt/zookeeper-3.4.6/bin/../lib/jline-0.9.94.jar:/opt/zookeeper-3.4.6/bin/../zookeeper-3.4.6.jar:/opt/zookeeper-3.4.6/bin/../src/java/lib/\*.jar:/opt/zookeeper-3.4.6/bin/../conf:.:/opt/java/jdk1.7.0\_79/lib/dt.jar:/opt/java/jdk1.7.0\_79/lib/tools.jar

2017-04-08 04:37:22,758 [myid:] - INFO [main:Environment@100] - Client environment:java.library.path=/usr/java/packages/lib/amd64:/usr/lib64:/lib64:/lib:/usr/lib

2017-04-08 04:37:22,759 [myid:] - INFO [main:Environment@100] - Client environment:java.io.tmpdir=/tmp

2017-04-08 04:37:22,759 [myid:] - INFO [main:Environment@100] - Client environment:java.compiler=<NA>

2017-04-08 04:37:22,759 [myid:] - INFO [main:Environment@100] - Client environment:os.name=Linux

2017-04-08 04:37:22,759 [myid:] - INFO [main:Environment@100] - Client environment:os.arch=amd64

2017-04-08 04:37:22,759 [myid:] - INFO [main:Environment@100] - Client environment:os.version=2.6.32-431.el6.x86\_64

2017-04-08 04:37:22,760 [myid:] - INFO [main:Environment@100] - Client environment:user.name=root

2017-04-08 04:37:22,760 [myid:] - INFO [main:Environment@100] - Client environment:user.home=/root

2017-04-08 04:37:22,760 [myid:] - INFO [main:Environment@100] - Client environment:user.dir=/opt/zookeeper-3.4.6/bin

2017-04-08 04:37:22,763 [myid:] - INFO [main:ZooKeeper@438] - Initiating client connection, connectString=localhost:2181 sessionTimeout=30000 watcher=org.apache.zookeeper.ZooKeeperMain$MyWatcher@2897eb44

Welcome to ZooKeeper!

2017-04-08 04:37:23,119 [myid:] - INFO [main-SendThread(localhost:2181):ClientCnxn$SendThread@975] - Opening socket connection to server localhost/0:0:0:0:0:0:0:1:2181. Will not attempt to authenticate using SASL (unknown error)

2017-04-08 04:37:23,145 [myid:] - INFO [main-SendThread(localhost:2181):ClientCnxn$SendThread@852] - Socket connection established to localhost/0:0:0:0:0:0:0:1:2181, initiating session

JLine support is enabled

2017-04-08 04:37:23,298 [myid:] - INFO [main-SendThread(localhost:2181):ClientCnxn$SendThread@1235] - Session establishment complete on server localhost/0:0:0:0:0:0:0:1:2181, sessionid = 0x15b4d3461d10000, negotiated timeout = 30000

WATCHER::

WatchedEvent state:SyncConnected type:None path:null

[zk: localhost:2181(CONNECTED) 0] **create /zk-book 123**

Created /zk-book

[zk: localhost:2181(CONNECTED) 1] **ls /**

[zk-book, zookeeper]

[zk: localhost:2181(CONNECTED) 2] **get /zk-book**

123

cZxid = 0x2

ctime = Sat Apr 08 07:06:05 PDT 2017

mZxid = 0x2

mtime = Sat Apr 08 07:06:05 PDT 2017

pZxid = 0x2

cversion = 0

dataVersion = 0

aclVersion = 0

ephemeralOwner = 0x0

dataLength = 3

numChildren = 0

[zk: localhost:2181(CONNECTED) 3] **set /zk-book 456**

cZxid = 0x2

ctime = Sat Apr 08 07:06:05 PDT 2017

mZxid = 0x3

mtime = Sat Apr 08 07:08:35 PDT 2017

pZxid = 0x2

cversion = 0

dataVersion = 1

aclVersion = 0

ephemeralOwner = 0x0

dataLength = 3

numChildren = 0

[zk: localhost:2181(CONNECTED) 4] **delete /zk-book**

[zk: localhost:2181(CONNECTED) 5] ls /

[zookeeper]

[zk: localhost:2181(CONNECTED) 6]