[【Apache ZooKeeper】命令行zkCli.sh使用指南](http://blog.csdn.net/ganglia/article/details/11606807)

2013-09-12 15:53 39913人阅读 [评论](http://blog.csdn.net/ganglia/article/details/11606807#comments)(2) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/ganglia/article/details/11606807#report)

http://static.blog.csdn.net/images/category_icon.jpg 分类：

云计算（53） http://static.blog.csdn.net/images/arrow_triangle%20_down.jpg

版权声明：本文为博主原创文章，未经博主允许不得转载。首发于http://rootsongjc.github.io

ZooKeeper命令行

ZooKeeper客户端有[**C语言**](http://lib.csdn.net/base/c)和[**Java**](http://lib.csdn.net/base/java)两个版本。

ZooKeeper的命令在/usr/lib/zookeeper/bin文件夹下。

运行Java版本的客户端使用bash zkCli.sh -server IP:port ，运行[**c语言**](http://lib.csdn.net/base/c)版本的使用./cli\_mt IP:port，下面介绍Java版本的，C语言版差不多。

查看具体结点信息

bash zkServer.sh status  
root@ubuntu:/usr/lib/zookeeper/bin# bash zkServer.sh status  
JMX enabled by default  
Using config: /usr/lib/zookeeper/bin/../conf/zoo.cfg  
Mode: standalone  
  
  
**查看哪个结点被选作leader或者follower**  
echo stat|nc 127.0.0.1 2181  
root@ubuntu:/usr/lib/zookeeper/bin# echo stat|nc 127.0.0.1 2181   
Zookeeper version: 3.4.5-cdh4.4.0--1, built on 09/04/2013 01:46 GMT  
Clients:  
 /127.0.0.1:57736[0](queued=0,recved=1,sent=0)  
  
  
Latency min/avg/max: 0/0/0  
Received: 1  
Sent: 0  
Connections: 1  
Outstanding: 0  
Zxid: 0x0  
Mode: standalone  
Node count: 4  
  
  
**测试是否启动了该Server，若回复imok表示已经启动**  
echo ruok|nc 127.0.0.1 2181  
root@ubuntu:/usr/lib/zookeeper/bin# echo ruok|nc 127.0.0.1 2181  
imok  
  
  
ZooKeeper命令行类似于shell。  
当启动 ZooKeeper 服务成功之后，输入下述命令，连接到 ZooKeeper 服务：  
bash zkCli.sh –server 192.168.255.133:2181  
  
  
root@ubuntu:/usr/lib/zookeeper/bin# bash zkCli.sh -server 192.168.255.133:2181  
Connecting to 192.168.255.133:2181  
2013-09-12 15:38:39,654 [myid:] - INFO  [main:Environment@100] - Client environment:zookeeper.version=3.4.5-cdh4.4.0--1, built on 09/04/2013 01:46 GMT  
2013-09-12 15:38:39,657 [myid:] - INFO  [main:Environment@100] - Client environment:host.name=ubuntu  
2013-09-12 15:38:39,659 [myid:] - INFO  [main:Environment@100] - Client environment:java.version=1.6.0\_43  
2013-09-12 15:38:39,660 [myid:] - INFO  [main:Environment@100] - Client environment:java.vendor=Sun Microsystems Inc.  
2013-09-12 15:38:39,662 [myid:] - INFO  [main:Environment@100] - Client environment:java.home=/usr/local/jdk1.6.0\_43/jre  
2013-09-12 15:38:39,664 [myid:] - INFO  [main:Environment@100] - Client environment:java.class.path=/usr/lib/zookeeper/bin/../build/classes:/usr/lib/zookeeper/bin/../build/lib/\*.jar:/usr/lib/zookeeper/bin/../lib/slf4j-log4j12-1.6.1.jar:/usr/lib/zookeeper/bin/../lib/slf4j-api-1.6.1.jar:/usr/lib/zookeeper/bin/../lib/netty-3.2.2.Final.jar:/usr/lib/zookeeper/bin/../lib/log4j-1.2.15.jar:/usr/lib/zookeeper/bin/../lib/jline-0.9.94.jar:/usr/lib/zookeeper/bin/../zookeeper-3.4.5-cdh4.4.0.jar:/usr/lib/zookeeper/bin/../src/java/lib/\*.jar:/usr/lib/zookeeper/bin/../conf:  
2013-09-12 15:38:39,665 [myid:] - INFO  [main:Environment@100] - Client environment:java.library.path=/usr/local/jdk1.6.0\_43/jre/lib/amd64/server:/usr/local/jdk1.6.0\_43/jre/lib/amd64:/usr/local/jdk1.6.0\_43/jre/../lib/amd64:/usr/java/packages/lib/amd64:/usr/lib64:/lib64:/lib:/usr/lib  
2013-09-12 15:38:39,667 [myid:] - INFO  [main:Environment@100] - Client environment:java.io.tmpdir=/tmp  
2013-09-12 15:38:39,668 [myid:] - INFO  [main:Environment@100] - Client environment:java.compiler=<NA>  
2013-09-12 15:38:39,669 [myid:] - INFO  [main:Environment@100] - Client environment:os.name=Linux  
2013-09-12 15:38:39,670 [myid:] - INFO  [main:Environment@100] - Client environment:os.arch=amd64  
2013-09-12 15:38:39,670 [myid:] - INFO  [main:Environment@100] - Client environment:os.version=3.2.0-51-generic  
2013-09-12 15:38:39,672 [myid:] - INFO  [main:Environment@100] - Client environment:user.name=root  
2013-09-12 15:38:39,672 [myid:] - INFO  [main:Environment@100] - Client environment:user.home=/root  
2013-09-12 15:38:39,673 [myid:] - INFO  [main:Environment@100] - Client environment:user.dir=/usr/lib/zookeeper/bin  
2013-09-12 15:38:39,674 [myid:] - INFO  [main:ZooKeeper@438] - Initiating client connection, connectString=192.168.255.133:2181 sessionTimeout=30000 watcher=org.apache.zookeeper.ZooKeeperMain$MyWatcher@2666e815  
Welcome to ZooKeeper!  
JLine support is enabled  
2013-09-12 15:38:39,720 [myid:] - INFO  [main-SendThread(192.168.255.133:2181):ClientCnxn$SendThread@966] - Opening socket connection to server 192.168.255.133/192.168.255.133:2181. Will not attempt to authenticate using SASL (Unable to locate a login configuration)  
2013-09-12 15:38:39,729 [myid:] - INFO  [main-SendThread(192.168.255.133:2181):ClientCnxn$SendThread@849] - Socket connection established to 192.168.255.133/192.168.255.133:2181, initiating session  
[zk: 192.168.255.133:2181(CONNECTING) 0] 2013-09-12 15:38:39,748 [myid:] - INFO  [main-SendThread(192.168.255.133:2181):ClientCnxn$SendThread@1207] - Session establishment complete on server 192.168.255.133/192.168.255.133:2181, sessionid = 0x141102059a70001, negotiated timeout = 30000  
  
  
WATCHER::  
  
  
WatchedEvent state:SyncConnected type:None path:null  
  
  
输入help显示帮助信息：  
  
help   
ZooKeeper -server host:port cmd args  
        connect host:port  
        get path [watch]  
        ls path [watch]  
        set path data [version]  
        rmr path  
        delquota [-n|-b] path  
        quit   
        printwatches on|off  
        create [-s] [-e] path data acl  
        stat path [watch]  
        close   
        ls2 path [watch]  
        history   
        listquota path  
        setAcl path acl  
        getAcl path  
        sync path  
        redo cmdno  
        addauth scheme auth  
        delete path [version]  
        setquota -n|-b val path  
[zk: 192.168.255.133:2181(CONNECTED) 1]

连接成功后，系统会输出 ZooKeeper 的相关环境以及配置信息，并在屏幕输出“ Welcome to ZooKeeper ”等信息。

命令行工具的一些简单操作如下：

**1 ）使用 ls 命令来查看当前 ZooKeeper 中所包含的内容：**  
[zk: 202.115.36.251:2181(CONNECTED) 1] ls /  
[zookeeper]  
  
**2 ）创建一个新的 znode ，使用 create /zk myData 。这个命令创建了一个新的 znode 节点“ zk ”以及与它关联的字符串：**  
[zk: 202.115.36.251:2181(CONNECTED) 2] create /zk "myData"  
Created /zk   
  
**3 ）我们运行 get 命令来确认 znode 是否包含我们所创建的字符串：**  
[zk: 202.115.36.251:2181(CONNECTED) 3] get /zk  
cZxid = 0x6  
ctime = Thu Sep 12 15:49:16 CST 2013  
mZxid = 0x7  
mtime = Thu Sep 12 15:49:38 CST 2013  
pZxid = 0x6  
cversion = 0  
dataVersion = 1  
aclVersion = 0  
ephemeralOwner = 0x0  
dataLength = 10  
numChildren = 0  
  
**4 ）下面我们通过 set 命令来对 zk 所关联的字符串进行设置：**  
[zk: 202.115.36.251:2181(CONNECTED) 4] set /zk "zsl"  
"myData"  
cZxid = 0x4  
ctime = Thu Sep 12 15:46:37 CST 2013  
mZxid = 0x4  
mtime = Thu Sep 12 15:46:37 CST 2013  
pZxid = 0x4  
cversion = 0  
dataVersion = 0  
aclVersion = 0  
ephemeralOwner = 0x0  
dataLength = 8  
numChildren = 0  
  
**5 ）下面我们将刚才创建的 znode 删除：**  
[zk: 202.115.36.251:2181(CONNECTED) 5] delete /zk  
  
当然，我们还可以创建有曾次的目录，比如使用create /zk/node1在zk目录下创建新的目录node1  
[zk: 192.168.255.133:2181(CONNECTED) 18] create /zk/node1 "node1"  
Created /zk/node1  
[zk: 192.168.255.133:2181(CONNECTED) 19] ls /zk  
[node1]  
  
 **使用ZooKeeper四字命令**

传递四个字母的字符串给ZooKeeper，ZooKeeper会返回一些有用的信息。

|  |  |
| --- | --- |
| **ZooKeeper 四字命令** | **功能描述** |
| conf | 输出相关服务配置的详细信息。 |
| cons | 列出所有连接到服务器的客户端的完全的连接 / 会话的详细信息。包括“接受 / 发送”的包数量、会话 id 、操作延迟、最后的操作执行等等信息。 |
| dump | 列出未经处理的会话和临时节点。 |
| envi | 输出关于服务环境的详细信息（区别于 conf 命令）。 |
| reqs | 列出未经处理的请求 |
| ruok | [**测试**](http://lib.csdn.net/base/softwaretest)服务是否处于正确状态。如果确实如此，那么服务返回“imok ”，否则不做任何相应。 |
| stat | 输出关于性能和连接的客户端的列表。 |
| wchs | 列出服务器 watch 的详细信息。 |
| wchc | 通过 session 列出服务器 watch 的详细信息，它的输出是一个与watch 相关的会话的列表。 |
| wchp | 通过路径列出服务器 watch 的详细信息。它输出一个与 session相关的路径。 |

例如：

root@ubuntu:/usr/lib/zookeeper/bin# echo conf |nc 192.168.255.133 2181      
clientPort=2181  
dataDir=/var/lib/zookeeper/version-2  
dataLogDir=/var/log/zookeeper/version-2  
tickTime=2000  
maxClientCnxns=50  
minSessionTimeout=4000  
maxSessionTimeout=40000  
serverId=0