

一、前言

在学习了 Zookeeper 相关的理论知识后，下面接着学习对 Zookeeper 的相关操作。

二、Zookeeper 部署

Zookeeper 有三种运行形式：集群模式、单机模式、伪集群模式。

三、服务端

bin 目录下常用的脚本解释

zkCleanup	清理 Zookeeper 历史数据，包括食物日志文件和快照数据文件
zkCli	Zookeeper 的一个简易客户端
zkEnv	设置 Zookeeper 的环境变量
zkServer	Zookeeper 服务器的启动、停止、和重启脚本

3.1 运行服务

进入 bin 目录，使用 **zkServer.sh start** 启动服务

```
leesf@leesf-K42JP:~/program/zookeeper/zookeeper-3.4.6/bin$ zkServer.sh start
JMX enabled by default
Using config: /home/leesf/program/zookeeper/zookeeper-3.4.6/bin/../conf/zoo.cfg
Starting zookeeper ... STARTED
```

使用 jps 命令查看，存在 QuorumPeerMain 进程，表示 Zookeeper 已经启动

```
leesf@leesf-K42JP:~/program/zookeeper/zookeeper-3.4.6/bin$ jps
3698 Jps
3672 QuorumPeerMain
```

3.2 停止服务

在 bin 目录下，使用 **zkServer.sh stop** 停止服务

```
leesf@leesf-K42JP:~/program/zookeeper/zookeeper-3.4.6/bin$ zkServer.sh stop
JMX enabled by default
Using config: /home/leesf/program/zookeeper/zookeeper-3.4.6/bin/../conf/zoo.cfg
Stopping zookeeper ... STOPPED
```

使用 jps 命令查看，QuorumPeerMain 进程已不存在，表示 Zookeeper 已经关闭

```
leesf@leesf-K42JP:~/program/zookeeper/zookeeper-3.4.6/bin$ jps
3791 Jps
```

四、客户端

4.1 打开客户端

在服务端开启的情况下，运行客户端，使用如下命令：**./zkCli.sh**

```
leesf@leesf-K42JP:~/program/zookeeper/zookeeper-3.4.6/bin$ ./zkCli.sh
Connecting to localhost:2181
2016-11-02 15:38:26,485 [myid:] - INFO [main:Environment@100] - Client env
ent:zookeeper.version=3.4.6-1569965, built on 02/20/2014 09:09 GMT
2016-11-02 15:38:26,490 [myid:] - INFO [main:Environment@100] - Client env
ent:host.name=leesf-K42JP
2016-11-02 15:38:26,490 [myid:] - INFO [main:Environment@100] - Client env
ent:java.version=1.8.0_60
2016-11-02 15:38:26,492 [myid:] - INFO [main:Environment@100] - Client env
ent:java.vendor=Oracle Corporation
2016-11-02 15:38:26,493 [myid:] - INFO [main:Environment@100] - Client env
ent:java.home=/home/leesf/program/java/jdk1.8.0_60/jre
```

```
Welcome to ZooKeeper!
2016-11-02 15:38:26,533 [myid:] - INFO [main-SendThread(localhost:2181):Cl
n$SendThread@975] - Opening socket connection to server localhost/127.0.0.
81. Will not attempt to authenticate using SASL (unknown error)
JLine support is enabled
2016-11-02 15:38:26,633 [myid:] - INFO [main-SendThread(localhost:2181):Cl
n$SendThread@852] - Socket connection established to localhost/127.0.0.1:
initiating session
[zk: localhost:2181(CONNECTING) 0] 2016-11-02 15:38:26,780 [myid:] - INFO
-SendThread(localhost:2181):ClientCnxn$SendThread@1235] - Session establish
complete on server localhost/127.0.0.1:2181, sessionId = 0x15823f944060000,
tiated timeout = 30000

WATCHER::

WatchedEvent state:SyncConnected type:None path:null
```

连接服务端成功，若连接不同的主机，可使用如下命令：**./zkCli.sh -server ip:port**

可以使用帮助命令 **help** 来查看客户端的操作

```
[zk: localhost:2181(CONNECTED) 1] 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
```

4.2 创建节点

使用 `create` 命令，可以创建一个 Zookeeper 节点，如

`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，若不指定，则表示持久节点；`acl` 用来进行权限控制。

① 创建顺序节点

使用 **`create -s /zk-test 123`** 命令创建 `zk-test` 顺序节点

```
[zk: localhost:2181(CONNECTED) 3] create -s /zk-test 123
Created /zk-test0000000019
[zk: localhost:2181(CONNECTED) 4] ls /
[controller_epoch, brokers, zookeeper, admin, consumers, zk-test0000000019,
ig]
```

可以看到创建的 `zk-test` 节点后面添加了一串数字以示区别。

② 创建临时节点

使用 **`create -e /zk-temp 123`** 命令创建 `zk-temp` 临时节点

```
[zk: localhost:2181(CONNECTED) 5] create -e /zk-temp 123
Created /zk-temp
[zk: localhost:2181(CONNECTED) 6] ls /
[controller_epoch, brokers, zookeeper, admin, consumers, zk-test0000000019,
ig, zk-temp]
```

临时节点在客户端会话结束后，就会自动删除，下面使用 **quit** 命令退出客户端

```
[zk: localhost:2181(CONNECTED) 7] quit
Quitting...
2016-11-02 16:03:45,178 [myid:] - INFO [main:ZooKeeper@684] - Session: 0x1
944060003 closed
2016-11-02 16:03:45,178 [myid:] - INFO [main-EventThread:ClientCnxn$EventT
@512] - EventThread shut down
```

再次使用客户端连接服务端，并使用 **ls /** 命令查看根目录下的节点

```
[zk: localhost:2181(CONNECTED) 0] ls /
[controller_epoch, brokers, zookeeper, admin, consumers, zk-test0000000019,
ig]
```

可以看到根目录下已经不存在 **zk-temp** 临时节点了。

③ 创建永久节点

使用 **create /zk-permanent 123** 命令创建 **zk-permanent** 永久节点

```
[zk: localhost:2181(CONNECTED) 1] create /zk-permanent 123
Created /zk-permanent
[zk: localhost:2181(CONNECTED) 2] ls /
[controller_epoch, zk-permanent, brokers, zookeeper, admin, consumers, zk-t
00000019, config]
```

可以看到永久节点不同于顺序节点，不会自动在后面添加一串数字。

4.3 读取节点

与读取相关的命令有 **ls** 命令和 **get** 命令，**ls** 命令可以列出 Zookeeper 指定节点下的所有子节点，只能查看指定节点下的第一级的所有子节点；**get** 命令可以获取 Zookeeper 指定节点的数据内容和属性信息。其用法分别如下

ls path [watch]

get path [watch]

ls2 path [watch]

若获取根节点下面的所有子节点，使用 **ls /** 命令即可

```
[zk: localhost:2181(CONNECTED) 3] ls /  
[controller_epoch, zk-permanent, brokers, zookeeper, admin, consumers, zk-t  
00000019, config]
```

若想获取根节点数据内容和属性信息，使用 **get /** 命令即可

```
[zk: localhost:2181(CONNECTED) 0] get /  
  
cZxid = 0x0  
ctime = Thu Jan 01 08:00:00 CST 1970  
mZxid = 0x0  
mtime = Thu Jan 01 08:00:00 CST 1970  
pZxid = 0xa15  
cversion = 36  
dataVersion = 0  
aclVersion = 0  
ephemeralOwner = 0x0  
dataLength = 0  
numChildren = 8
```

也可以使用 **ls2 /** 命令查看

```
[zk: localhost:2181(CONNECTED) 17] ls2 /  
[controller_epoch, zk-permanent, brokers, zookeeper, admin, consumers, zk-t  
00000019, config]  
cZxid = 0x0  
ctime = Thu Jan 01 08:00:00 CST 1970  
mZxid = 0x0  
mtime = Thu Jan 01 08:00:00 CST 1970  
pZxid = 0xa23  
cversion = 40  
dataVersion = 0  
aclVersion = 0  
ephemeralOwner = 0x0  
dataLength = 0  
numChildren = 8
```

可以看到其子节点数量为 8。

若想获取/zk-permanent 的数据内容和属性，可使用如下命令：**get /zk-permanent**

```
[zk: localhost:2181(CONNECTED) 1] get /zk-permanent
123
cZxid = 0xa15
ctime = Wed Nov 02 16:10:27 CST 2016
mZxid = 0xa15
mtime = Wed Nov 02 16:10:27 CST 2016
pZxid = 0xa15
cversion = 0
dataVersion = 0
aclVersion = 0
ephemeralOwner = 0x0
dataLength = 3
numChildren = 0
```

可以看到其数据内容为 123，还有其他的属性，之后会详细介绍。

4.4 更新节点

使用 **set** 命令，可以更新指定节点的数据内容，用法如下

set path data [version]

其中，**data** 就是要更新的新内容，**version** 表示数据版本，如将 **/zk-permanent** 节点的数据更新为 456，可以使用如下命令：**set /zk-permanent 456**

```
[zk: localhost:2181(CONNECTED) 2] set /zk-permanent 456
cZxid = 0xa15
ctime = Wed Nov 02 16:10:27 CST 2016
mZxid = 0xa1a
mtime = Wed Nov 02 16:43:32 CST 2016
pZxid = 0xa15
cversion = 0
dataVersion = 1
aclVersion = 0
ephemeralOwner = 0x0
dataLength = 3
numChildren = 0
```

现在 **dataVersion** 已经变为 1 了，表示进行了更新。

4.5 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]

其中 **version** 也是表示数据版本，使用 **delete /zk-permanent** 命令即可删除 **/zk-permanent** 节点

```
[zk: localhost:2181(CONNECTED) 10] delete /zk-permanent  
[zk: localhost:2181(CONNECTED) 11] ls /  
[controller_epoch, brokers, zookeeper, admin, consumers, zk-test0000000019,  
iq]
```

可以看到，已经成功删除/zk-permanent 节点。值得注意的是，**若删除节点存在子节点，那么无法删除该节点，必须先删除子节点，再删除父节点。**