**一、安装条件**

想要安装zookeeper，必须先在linux中安装好jdk。安装步骤见：

<https://www.cnblogs.com/expiator/p/9987351.html>

二、下载并解压zookeeper压缩包

1. 先进入/usr/local/目录，也可以是其他的目录：

**[root@localhost /]# cd /usr/local**

2. zookeeper安装包可以在官网下载。

也可以在后面这个地址下载  <http://mirror.bit.edu.cn/apache/zookeeper/zookeeper-3.4.13/zookeeper-3.4.13.tar.gz>

如果链接打不开，就先打开 [http://mirror.bit.edu.cn/apache/zookeeper](http://mirror.bit.edu.cn/apache/zookeeper/)， 再选择版本。

在此目录下载zookeeper安装包：

[root@localhost local]# wget http://mirror.bit.edu.cn/apache/zookeeper/zookeeper-3.4.13/zookeeper-3.4.13.tar.gz

3. 解压：

[root@localhost local]# **tar -zxvf zookeeper-3.4.13.tar.gz**

**三、编辑配置文件**

1.进入conf目录：

[root@localhost local]# **cd zookeeper-3.4.13/conf**

2. 将zoo\_sample.cfg这个文件复制为zoo.cfg (必须是这个文件名)

[root@localhost conf]# **cp zoo\_sample.cfg zoo.cfg**

3. 进入zoo.cfg文件进行编辑

[root@localhost conf]# **vim zoo.cfg**

4. 按 i 进入编辑模式，修改以下内容：

**dataDir=/tmp/zookeeper/data
dataLogDir=/tmp/zookeeper/log**

注意：如果想配置集群的话，请在clientPort下面添加服务器的ip。如（多服务器配置）

server.1=192.168.180.132:2888:3888

server.2=192.168.180.133:2888:3888

server.3=192.168.180.134:2888:3888

如果电脑内存比较小，zookeeper还可以设置成伪集群。也就是全部服务器采用同一个ip，但是使用不同的端口。

5. 在tmp目录创建目录。

[root@localhost conf]# **mkdir /tmp/zookeeper**

[root@localhost conf]# **mkdir /tmp/zookeeper/data**

[root@localhost conf]# **mkdir /tmp/zookeeper/log**

 6.如果是配置集群，还需要在前面配置过的dataDir路径下新增myid文件

[root@localhost conf]# **cd /tmp/zookeeper/data**

[root@localhost data]# **touch myid**

[root@localhost data]# **vim myid**

在data目录下创建文件，文件名为“myid”, 编辑该“myid”文件，并在对应的IP的机器上输入对应的编号。

如在192.168.180.132上，“myid”文件内容就是1。在192.168.180.133上，内容就是2。

**四、配置环境变量**

1.上面的操作都完事之后，我们需要配置一下环境变量，配置环境变量的命令如下：

[root@localhost zookeeper-3.4.13]# **export ZOOKEEPER\_INSTALL=/usr/local/zookeeper-3.4.13/**
[root@localhost zookeeper-3.4.13]# **export PATH=$PATH:$ZOOKEEPER\_INSTALL/bin**

**五、启动zookeeper**

1.进入bin目录，并启动zookeep。如果不是在bin目录下执行，启动zookeeper时会报错： bash: **./zkServer.sh:  No such file or directory**

注意：  ./zkServer.sh start前面的 .  不可忽略。

[root@localhost local]# cd /usr/local/zookeeper-3.4.13/bin
[root@localhost bin]# **./zkServer.sh start**

2.启动成功效果如下：

ZooKeeper JMX enabled by default
Using config: /usr/local/zookeeper-3.4.13/bin/../conf/zoo.cfg
Starting zookeeper ... STARTED

3.zookeeper的服务端启动后，还需要启动zookeeper的客户端：

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

如果是连接多个不同的主机节点，可以使用如下命令：

./zkCli.sh -server 192.168.2.30:2888

启动成功效果如下：

copycode.gif

Connecting to localhost:2181

..........

..........

..........

Welcome to ZooKeeper!
2018-10-25 21:04:54,407 [myid:] - INFO [main-SendThread(localhost:2181):ClientCnxn$SendThread@1029] - Opening socket connection to server localhost/0:0:0:0:0:0:0:1:2181. Will not attempt to authenticate using SASL (unknown error)
JLine support is enabled
2018-10-25 21:04:54,471 [myid:] - INFO [main-SendThread(localhost:2181):ClientCnxn$SendThread@879] - Socket connection established to localhost/0:0:0:0:0:0:0:1:2181, initiating session
[zk: localhost:2181(CONNECTING) 0] 2018-10-25 21:04:54,501 [myid:] - INFO [main-SendThread(localhost:2181):ClientCnxn$SendThread@1303] - Session establishment complete on server localhost/0:0:0:0:0:0:0:1:2181, sessionid = 0x10000712e6f0000, negotiated timeout = 30000
WATCHER::
WatchedEvent state:SyncConnected type:None path:null

copycode.gif

4.查看状态：

[root@localhost bin]# **./zkServer.sh status**
ZooKeeper JMX enabled by default
Using config: /usr/local/zookeeper-3.4.13/bin/../conf/zoo.cfg
Mode: standalone

**遇到问题怎么解决？**

zookeeper的出错日志会记录在 **zookeeper.out。**

**当前处于哪个目录，执行完zkServer.sh start命令， zookeeper.out就会写在哪个目录。**

**vim zookeeper.out 可以查看报错信息。然后再搜索解决。**

**六、zookeeper使用**

通过 ./zkCli.sh 进入客户端后，就可以使用命令来操作zookeeper了。

1.创建节点

**使用create命令，可以创建一个zookeeper节点。**

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

**其中-s表示顺序节点，-e表示临时节点。默认情况下，创建的是持久节点。**

**path是节点路径，data是节点数据，acl是用来进行权限控制的。**

**如下：**

**创建一个叫做/zk-test的节点，内容是"123"**

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

**Created /zk-test**

**创建/zk-test的子节点book，内容是"233"**

**[zk: localhost:2181(CONNECTED) 7] create /zk-test/book 233
Created /zk-test/book**

**2.查看节点内容**

**使用get命令，可以获取zookeeper指定节点的内容和属性信息。**

**如下：**

copycode.gif

**[zk: localhost:2181(CONNECTED) 1] get /zk-test**

**123
cZxid = 0x3a
ctime = Sun Nov 11 21:50:44 CST 2018
mZxid = 0x3a
mtime = Sun Nov 11 21:50:44 CST 2018
pZxid = 0x3a
cversion = 0
dataVersion = 0
aclVersion = 0
ephemeralOwner = 0x0
dataLength = 3
numChildren = 0**

copycode.gif

**3.查看子节点**

**使用ls命令可以查看指定节点下的所有子节点**

**以下查看根目录下的所有子节点：**

**[zk: localhost:2181(CONNECTED) 2] ls /**

**[zk-test, zookeeper]**

**查看zk-test节点的子节点：**

**[zk: localhost:2181(CONNECTED) 3] ls /zk-test**

**[book]**

**4.更新节点内容**

**使用set命令，更新节点内容。格式为：**

**set   path  data**

**其中的data就是要更新的新内容。**

copycode.gif

**[zk: localhost:2181(CONNECTED) 4] set /zk-test 456
cZxid = 0x3a
ctime = Sun Nov 11 21:50:44 CST 2018
mZxid = 0x3b
mtime = Sun Nov 11 22:05:20 CST 2018
pZxid = 0x3a
cversion = 0
dataVersion = 1
aclVersion = 0
ephemeralOwner = 0x0
dataLength = 3
numChildren = 0**

copycode.gif

**在输出的信息中，可以发现，dataVersion的值由原来的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变更。**

**6.删除节点**

**使用delete命令来删除节点，如下：**

**[zk: localhost:2181(CONNECTED) 11] delete /zk-test**

**Node not empty: /zk-test**

**可以发现，一个节点存在子节点时，无法删除该节点。**

**删除子节点/zk-test/book，如下：**

**[zk: localhost:2181(CONNECTED) 12] delete /zk-test/book
WATCHER::
WatchedEvent state:SyncConnected type:NodeDeleted path:/zk-test/book**

**zookeeper中的watcher会监控节点，当子节点发生变化时会发出通知。此时提示子节点 /zk-test/book删除成功。**

**继续尝试删除节点 /zk-test，**

**[zk: localhost:2181(CONNECTED) 13] ls /zk-test
[]
[zk: localhost:2181(CONNECTED) 14] delete /zk-test
[zk: localhost:2181(CONNECTED) 15] ls /
[]**

**删除成功。**