# 1 **Zookeeper**

## 1.1**介绍**

Zookeeper是Apacahe Hadoop的子项目，是一个树型的目录服务，支持变更推送，适合作为Dubbo服务的注册中心，工业强度较高，可用于生产环境。是Hadoop的核心项目之一。

扩展：Hadoop的核心项目（9点）

1. HDFS: Hadoop分布式文件系统(Distributed File System) － HDFS (Hadoop Distributed File System)

2. MapReduce：并行计算框架，0.20前使用 org.apache.hadoop.mapred 旧接口，0.20版本开始引入org.apache.hadoop.mapreduce的新API

3. HBase: 类似Google BigTable的分布式NoSQL列数据库。（HBase和Avro已经于2010年5月成为顶级 Apache 项目）

4. Hive：数据仓库工具，由Facebook贡献。

5. Zookeeper：分布式锁设施，提供类似Google Chubby的功能，由Facebook贡献。

6. Avro：新的数据序列化格式与传输工具，将逐步取代Hadoop原有的IPC机制。

7. Pig: 大数据分析平台，为用户提供多种接口。

8. Ambari：Hadoop管理工具，可以快捷的监控、部署、管理集群。

9. Sqoop：于在HADOOP与传统的数据库间进行数据的传递。

**Zookeeper：**

1、可以作为集群的管理工具使用。

2、可以集中管理配置文件。

## 1.2工作原理：

① ZooKeeper的核心是原子广播，这个机制保证了各个Server之间的同步。实现这个机制的协议叫做Zab协议。Zab协议有两种模式，它们分别是恢复模式和广播模式。当服务器启动或者领导者崩溃后，Zab就进入了恢复模式，当领导者被选举出来，且大多数Server完成了和Leader的状态同步以后，恢复模式就结束了。状态同步保证了Leader和Server具有相同的系统状态。

② 一旦leader已经和多数的follower进行了状态同步后，他就可以开始广播消息了，即进入广播状态。这时候当一个server加入zookeeper服务中，它会在恢复模式下启动，发现leader，并和leader进行状态同步。待到同步结束，它也参与消息广播。Zookeeper服务一直维持在Broadcast状态，直到leader崩溃了或者leader失去了大部分的followers支持。

③ 广播模式需要保证proposal被按顺序处理，因此zk采用了递增的事务id号(zxid)来保证。所有的提议(proposal)都在被提出的时候加上了zxid。实现中zxid是一个64为的数字，它高32位是epoch用来标识leader关系是否改变，每次一个leader被选出来，它都会有一个新的epoch。低32位是个递增计数。

④ 当leader崩溃或者leader失去大多数的follower，这时候zk进入恢复模式，恢复模式需要重新选举出一个新的leader，让所有的server都恢复到一个正确的状态。

## 1.3 Zookeeper的安装环境

Linux：centos7

JDK: 1.8.0\_181

Zookeeper-3.5.4-beta (用了最新公测版，发现了它的)

Zookeeper是java开发的可以运行在windows、linux等环境下。需要先安装jdk。

| **系统** | **开发环境** | **生产环境** |
| --- | --- | --- |
| Linux | 支持 | 支持 |
| Solaris | 支持 | 支持 |
| FreeBSD | 支持 | 支持 |
| Windows | 支持 | 不支持 |
| MacOS | 支持 | 不支持 |

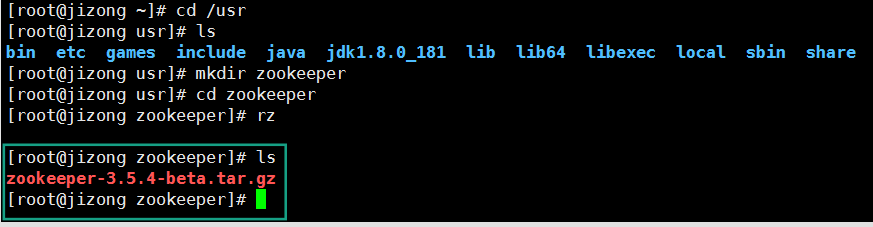
表1：ZooKeeper支持的运行平台

## 1.4 安装步骤：

安装步骤：

第一步：安装jdk（jdk1.8.0\_181）, 创建文件夹：

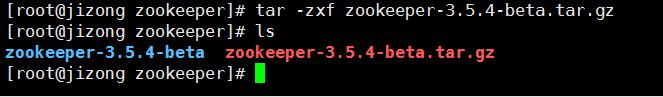
[root@jizong usr]# mkdir zookeeper zookeeper-3.5.4-beta



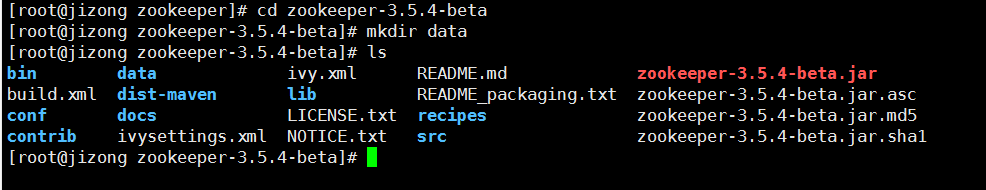
第二步：把zookeeper的压缩包上传到linux虚拟机上。

第三步：解压缩压缩包

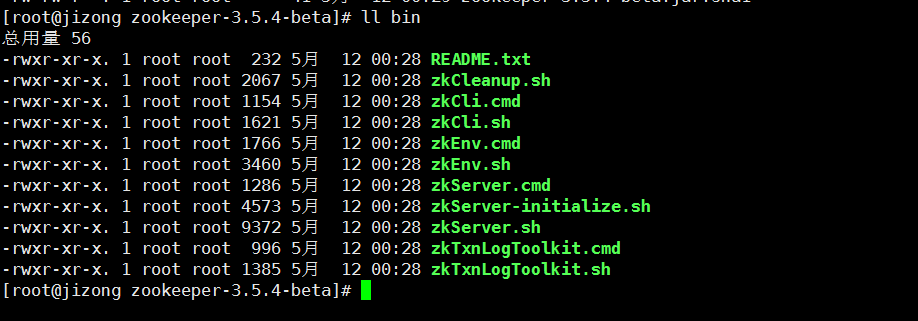
tar -zxf zookeeper-3.5.4-beta



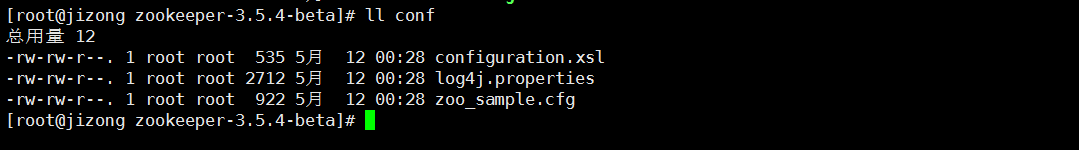
第四步：进入zookeeper-3.5.4-beta目录，创建data文件夹。ll或ls查看当前目录文件

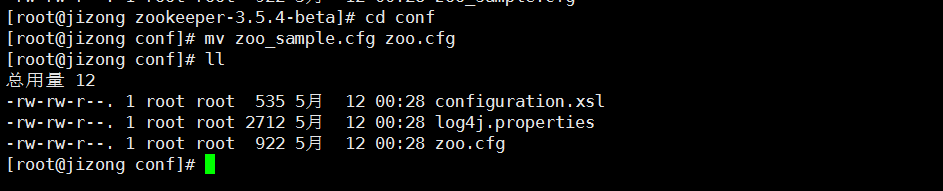


Data路径：/usr/zookeeper/zookeeper-3.5.4-beta/data



第五步：把zoo\_sample.cfg改名为zoo.cfg





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

第六步：修改data属性：

[root@jizong zookeeper]# cd zookeeper-3.5.4-beta/data/

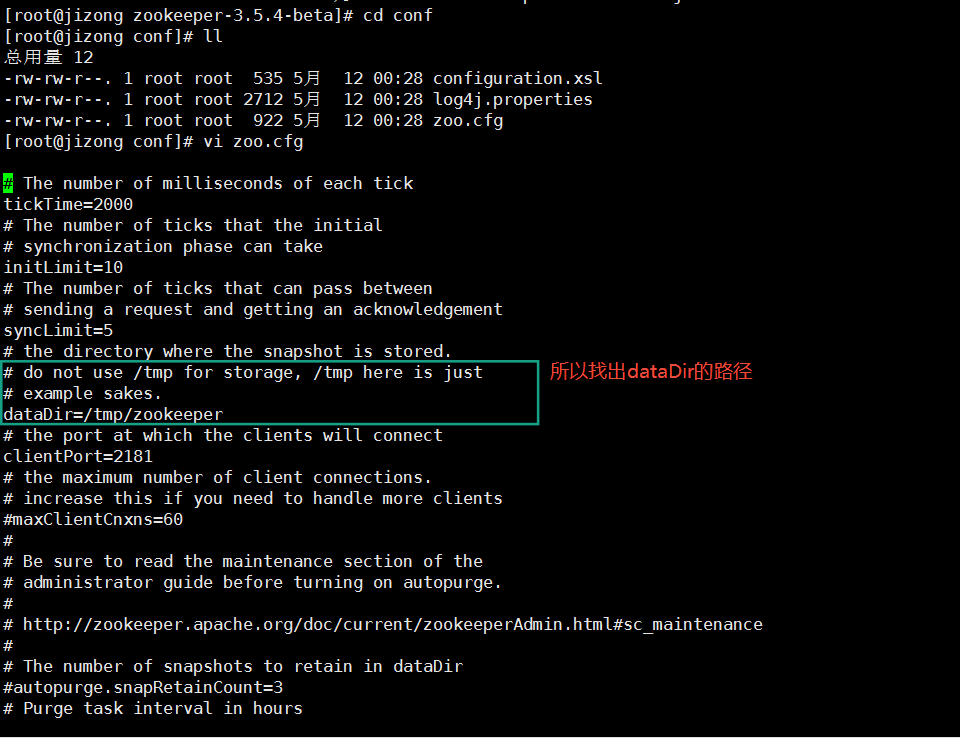
[root@jizong data]# pwd

/usr/zookeeper/zookeeper-3.5.4-beta/data

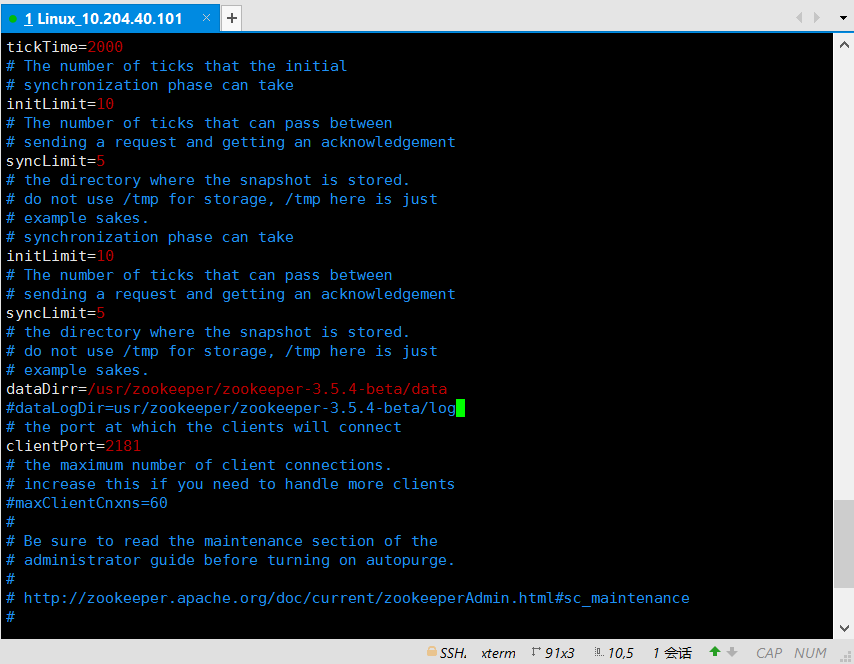
进到zoo.cfg

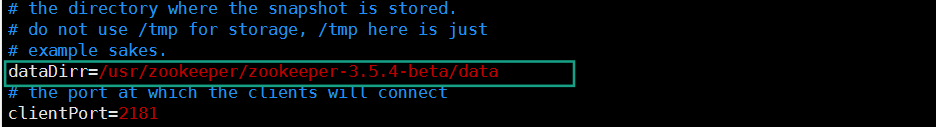
vim /usr/zookeeper/zookeeper-3.5.4-beta/conf/zoo.cfg

# do not use /tmp for storage,xxxxxx



补充：zoo.cfg 的配置说明



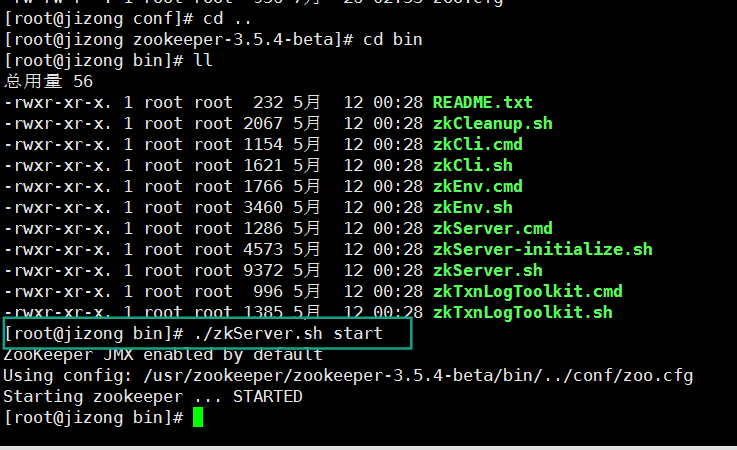


dataDirr

Esc--:wq (保存退出)

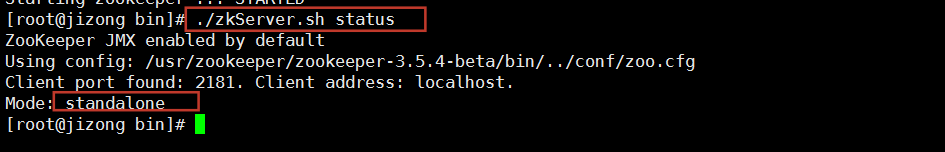
第七步：启动zookeeper

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



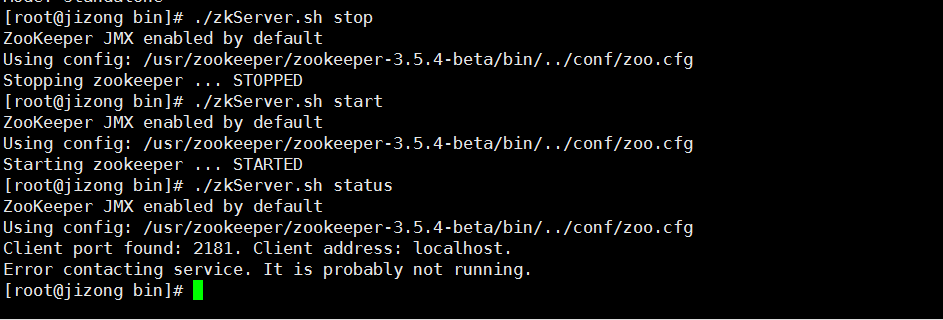
关闭：[root@localhost bin]# ./zkServer.sh stop

查看状态：[root@localhost bin]# ./zkServer.sh status



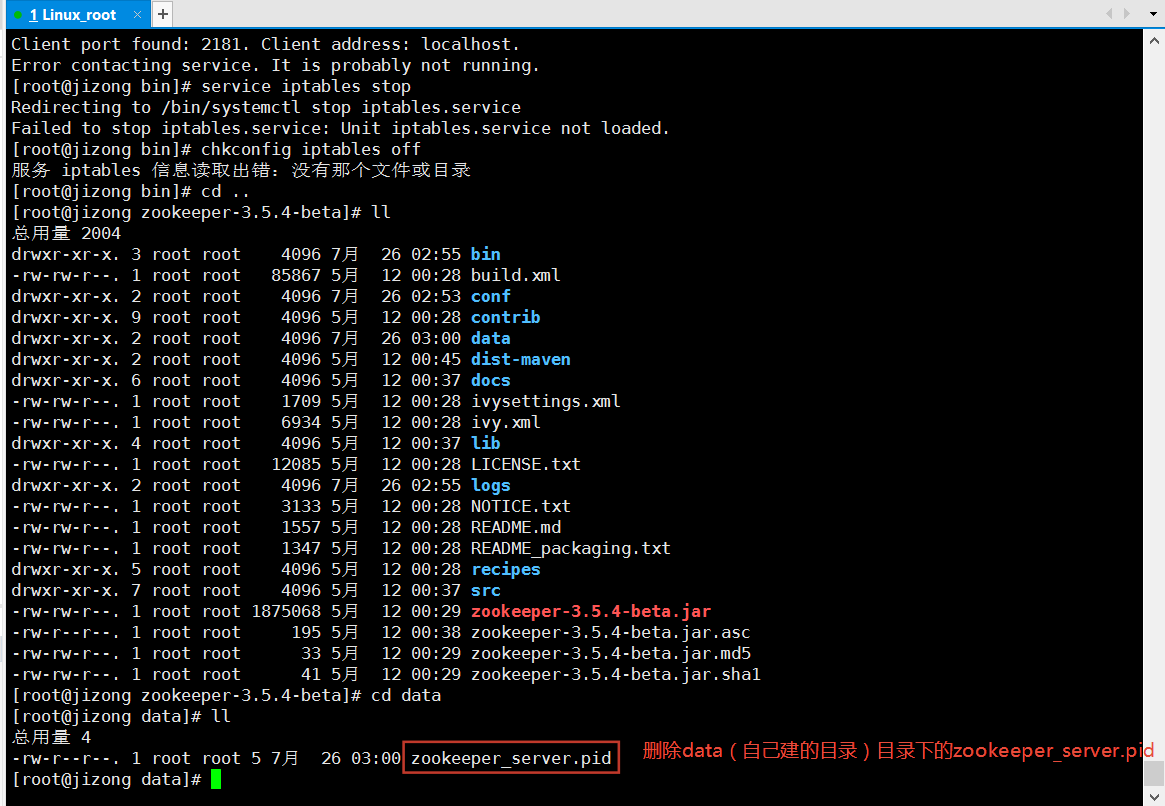
启动报错：

原因：[2.1 zookeeper 的默认端口问题](#_2.1_zookeeper_的默认端口问题)



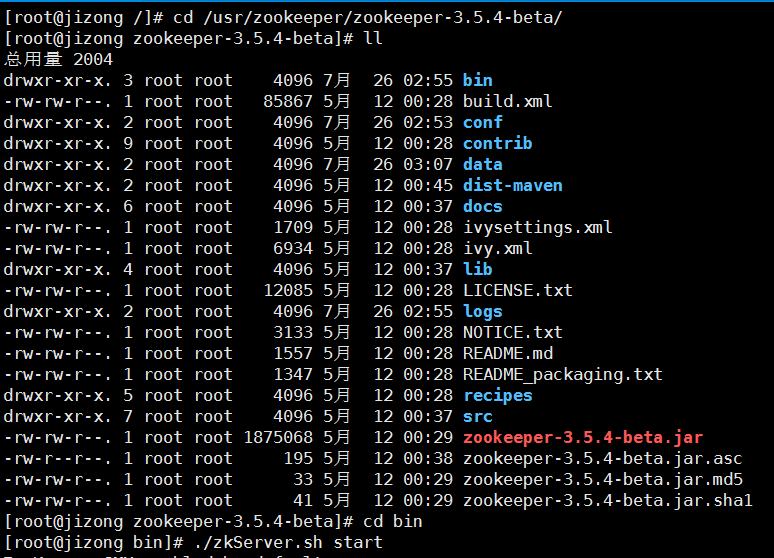
解决方法：

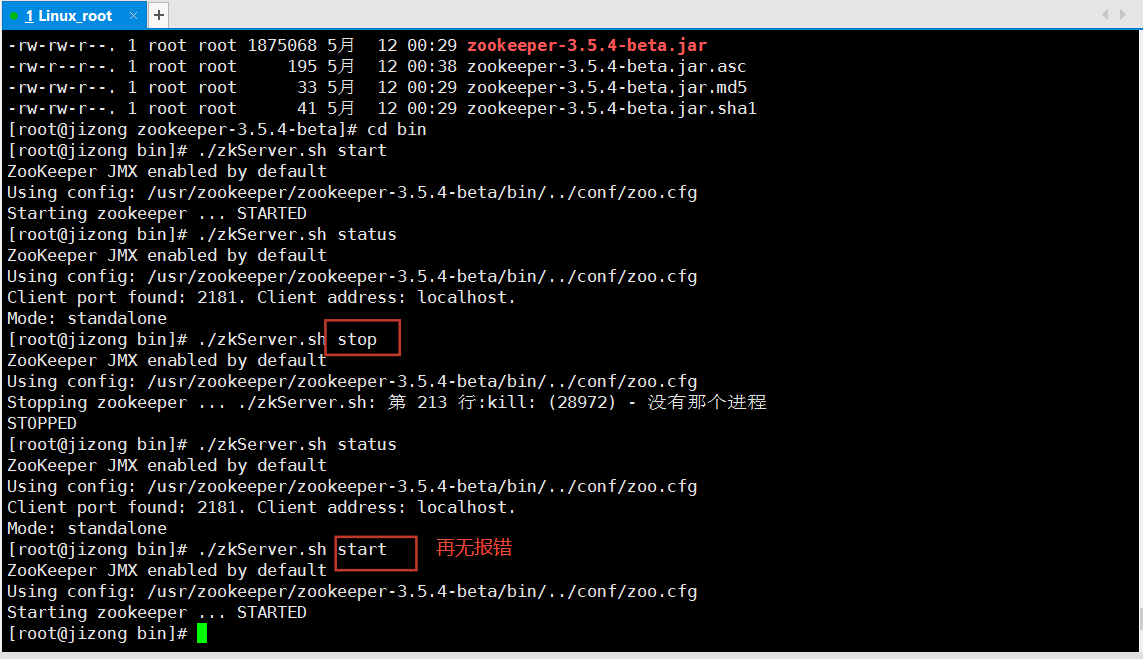
cd /usr/zookeeper/zookeeper-3.5.4-beta/data



重新启动：

[root@jizong /]# cd /usr/zookeeper/zookeeper-3.5.4-beta/





[root@jizong data]# rm zookeeper\_server.pid

rm：是否删除普通文件 "zookeeper\_server.pid"？y

[root@jizong data]# ll

总用量 0

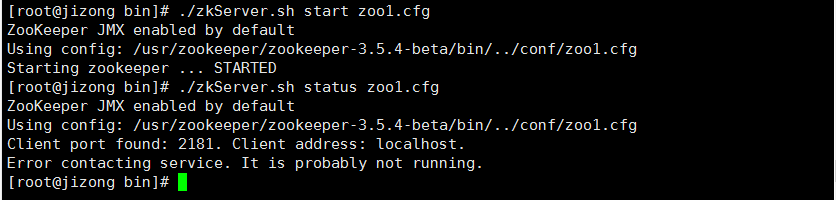
**注意：需要关闭防火墙。**

# 2 单机的伪集群

参考文档：<https://zookeeper.apache.org/doc/>

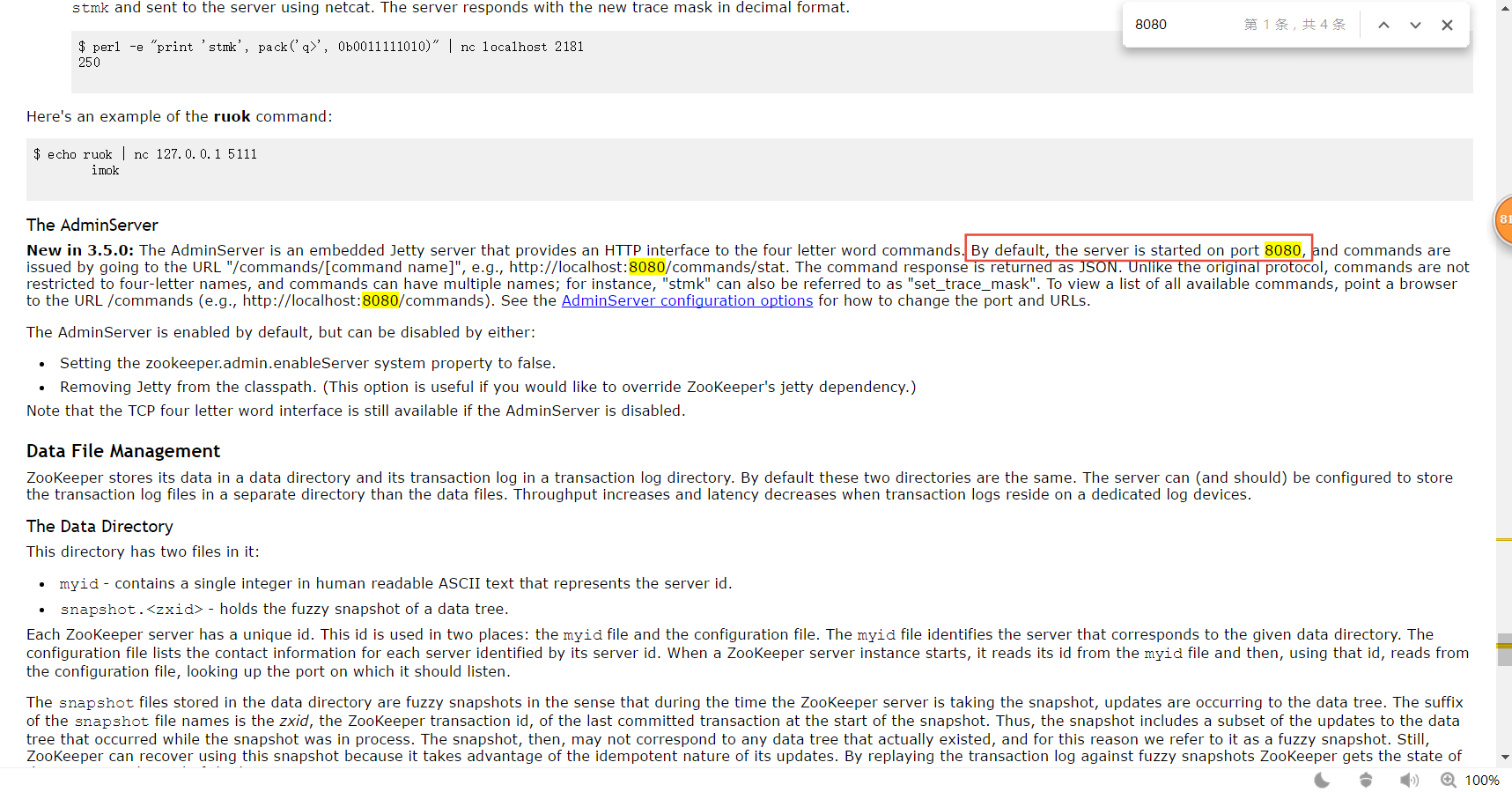
## 2.1 zookeeper 的默认端口问题

错误1：

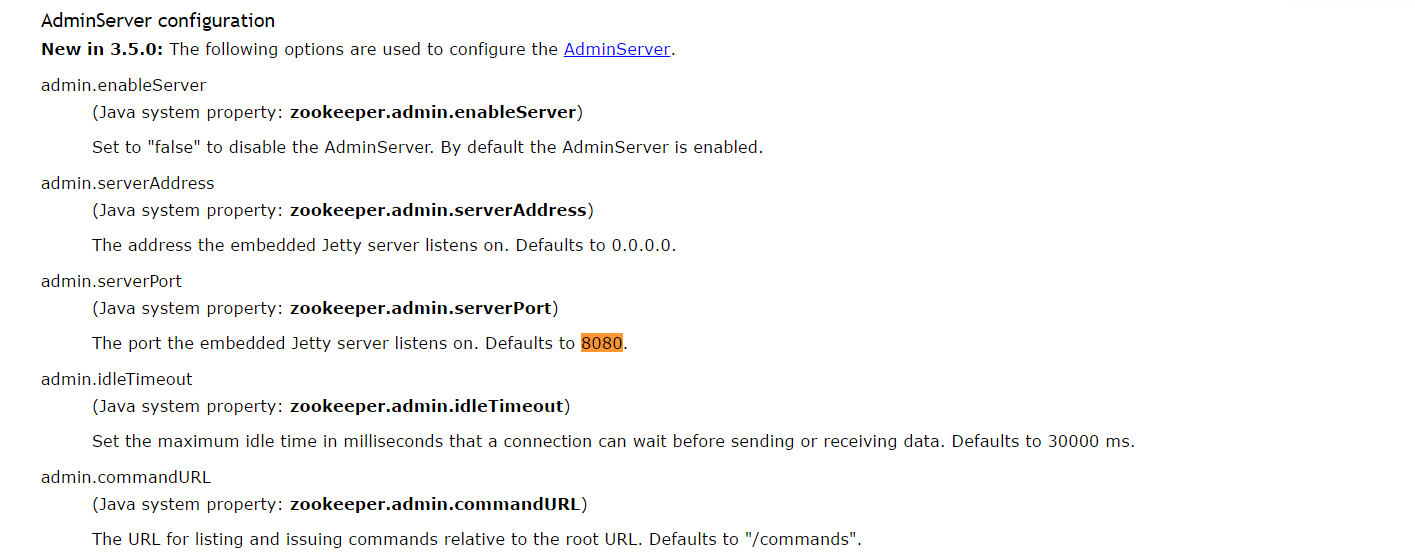


原因之一：

[https://zookeeper.apache.org/doc/r3.5.4-beta/zookeeperAdmin.html#Single+Machine+Requirements](https://zookeeper.apache.org/doc/r3.5.4-beta/zookeeperAdmin.html%23Single+Machine+Requirements)



配置属性如下



解决方法：在zoo.cfg配置文件中，添加：

admin.serverPort=端口号 ，防止默认端口和tomcat冲突。

原因之二：zoo.cfg中的配置dataDirr=XX（正确：dataDir）

## 2.2 复制三份 zookeeper

在目录/usr/zookeeper/zookeeper-3.5.4-beta 下面操作

①建立3个文件夹：zookeeper-3.5.4-1， zookeeper-3.5.4-2， zookeeper-3.5.4-3。然后每个文件夹里面解压一个zookeeper包。把zookeeper-3.5.4-beta文件目录下的文件mv到上一层，再创建一个“data”目录，因为conf-zooX.cfg中的dataDir=XXX/data。  
② 进入data目录，创建一个myid的文件，里面写入一个数字（zoo.cfg中有配置的server.X），server.1,就写一个1，server.2对应myid文件就写入2，server.3对应myid文件就写个3  
③ zookeeper的默认配置文件为zookeeper/conf/zoo\_sample.cfg，需要将其修改为zoo.cfg。其中各配置项的含义，解释如下：  
① tickTime：CS通信心跳时间  
Zookeeper 服务器之间或客户端与服务器之间维持心跳的时间间隔，也就是每个 tickTime 时间就会发送一个心跳。tickTime以毫秒为单位。  
tickTime=2000    
② initLimit：LF初始通信时限  
集群中的follower服务器(F)与leader服务器(L)之间初始连接时能容忍的最多心跳数（tickTime的数量）。  
initLimit=5    
③ syncLimit：LF同步通信时限  
集群中的follower服务器与leader服务器之间请求和应答之间能容忍的最多心跳数（tickTime的数量）。  
syncLimit=2     
④ dataDir：数据文件目录  
Zookeeper保存数据的目录，默认情况下，Zookeeper将写数据的日志文件也保存在这个目录里。  
dataDir=/home/michael/opt/zookeeper/data    
⑤ clientPort：客户端连接端口  
客户端连接 Zookeeper 服务器的端口，Zookeeper 会监听这个端口，接受客户端的访问请求。

clientPort=2181

命令：vim /usr/zookeeper/zookeeper-3.5.4-1/conf/zoo1.cfg

更改后：

tickTime=2000

initLimit=10

syncLimit=5

dataDir=/usr/zookeeper/zookeeper-3.5.4-1/data

clientPort=2181

admin.serverPort=8901

server.1=127.0.0.1:2891:3891

server.2=127.0.0.1:2892:3892

server.3=127.0.0.1:2893:3893

命令：vim /usr/zookeeper/zookeeper-3.5.4-2/conf/zoo2.cfg

更改后：

tickTime=2000

initLimit=10

syncLimit=5

dataDir=/usr/zookeeper/zookeeper-3.5.4-2/data

clientPort=2182

admin.serverPort=8902

server.1=127.0.0.1:2891:3891

server.2=127.0.0.1:2892:3892

server.3=127.0.0.1:2893:3893

命令：vim /usr/zookeeper/zookeeper-3.5.4-3/conf/zoo3.cfg

更改后

tickTime=2000

initLimit=10

syncLimit=5

dataDir=/usr/zookeeper/zookeeper-3.5.4-3/data

clientPort=2183

admin.serverPort=8903

server.1=127.0.0.1:2891:3891

server.2=127.0.0.1:2892:3892

server.3=127.0.0.1:2893:3893

两点说明：

* server.id=host:port1:port2   
   其中id为一个数字，表示zk进程的id，这个id也是dataDir目录下的[myid文件](#_2.3_myid)内容。   
   host是该zk进程所在的IP地址，port1表示follower和leader交换消息所使用的端口，port2表示选举leader所使用的端口。port2 表示的是万一集群中的 Leader 服务器挂了，需要一个端口来重新进行选举，选出一个新的 Leader，而这个端口就是用来执行选举时服务器相互通信的端口。如果是伪集群的配置方式，由于 host都是一样，所以不同的 Zookeeper 实例通信端口号不能一样。
* host

配置hosts文件：编辑/etc/hosts，加入如下内容：

127.0.0.1 a1

127.0.0.1 a2

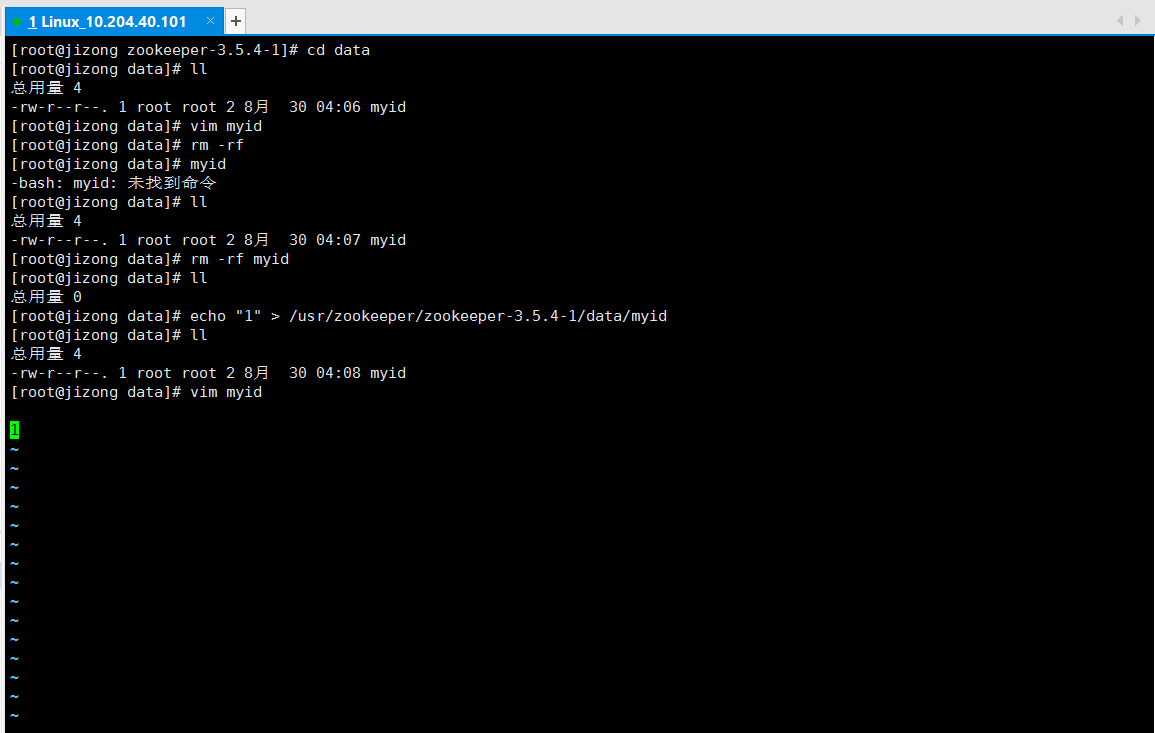
127.0.0.1 a3

单机伪集群，IP都是127.0.0.1，配置对应关系后，host 可以用a1,a2,a3 代替。

## 2.3 myid

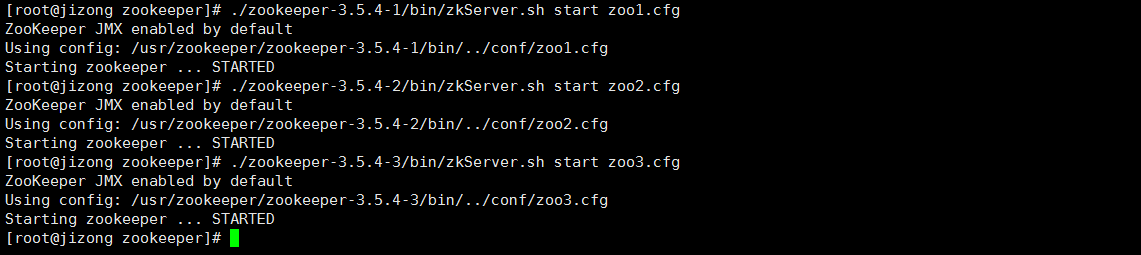
在zookeeper-3.5.4-(1~3) 中的 data目录下修改 myid 文件内容分别为：1，2，3

[root@jizong data]# echo "1" > /usr/zookeeper/zookeeper-3.5.4-1/data/myid



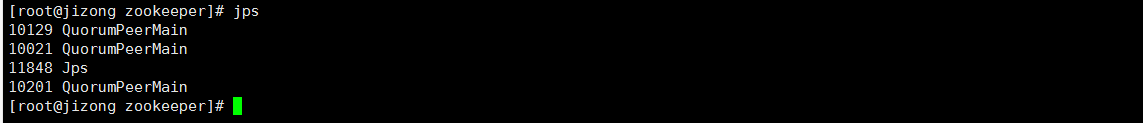
## 2.4 启动

命令：[root@jizong zookeeper]# ./zookeeper-3.5.4-1/bin/zkServer.sh start zoo1.cfg



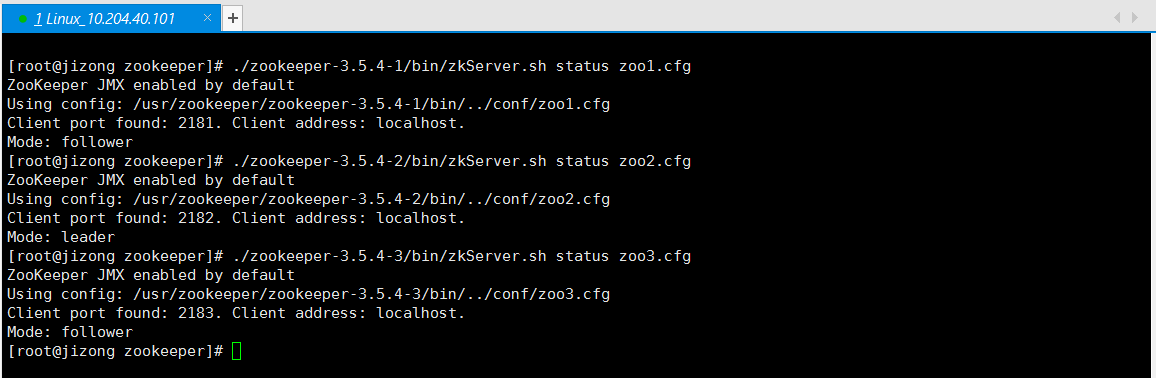
查看节点

命令：[root@jizong zookeeper]# jps



查看状态

命令：[root@jizong zookeeper]# ./zookeeper-3.5.4-1/bin/zkServer.sh status zoo1.cfg



我惊奇的发现zookeeper-3.5.4-1 不是 leader，而是zookeeper-3.5.4-2 ？又去了解了一下

参考网址<https://blog.csdn.net/z69183787/article/details/79368177>

自己消化了下，结合本配置的原因如下：

zookeeper中算法是基于Leader选举：意思是当得票数超过半数以上时，成为leader,其他成为follower。

比如我的这三个服务器分别是zookeeper-3.5.4-1（A）, zookeeper-3.5.4-2（B）, zookeeper-3.5.4-3（C）。

1.我们依次启动A   ->  B  ->  C

2.启动A时，A会给自己投票，然后会把信息发送给其他服务器，但是现在就A一个服务器，无法通信，所以A会处于Looking状态

3.启动B时，B会给自己投票，因为B的id为2，权重比A大，所以B成为老大，但是此时投票数超过半数，所以B成为leader。

4.启动C时，C给自己投票，C权重比A和B都大，但是这个时候B已经成为leader，所以C只能成为follower,A和C成为follower

同理：如果A,B出故障，当运行到C时，C权重大，成为老大，投票数超过半数成为leader，虽然D，E服务器ID（权重）比C大，但是由于C已经成为leader，那么DE只能被迫成为follower，前提是：C比DE先启动。

注意事项：

①在一台机器上部署了3个zookeeper-3.5.4；需要注意的是clientPort这个端口，如果在1台机器上部署多个zookeeper-3.5.4,那么每台机器都要不同的clientPort，比如 zookeeper-3.5.4-1是2181, zookeeper-3.5.4-2是2182，zookeeper-3.5.4-3是2183。

②dataDir ：数据目录，默认包过日记目录。

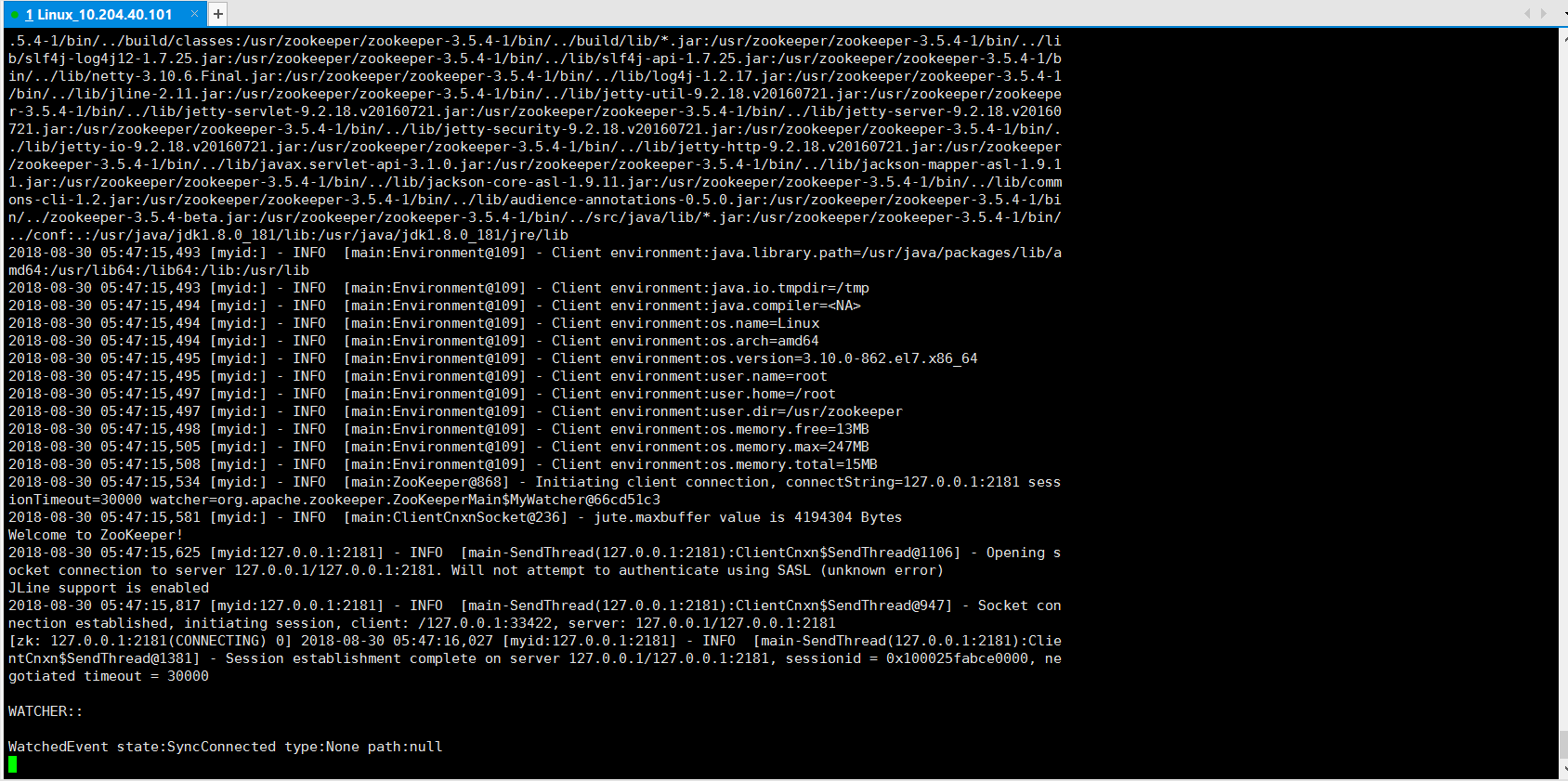
dataLogDir：配置日记目录。

## 2.5 连接客户端

通过客户端连接ZooKeeper的集群，可以任意的zookeeper是进行连接

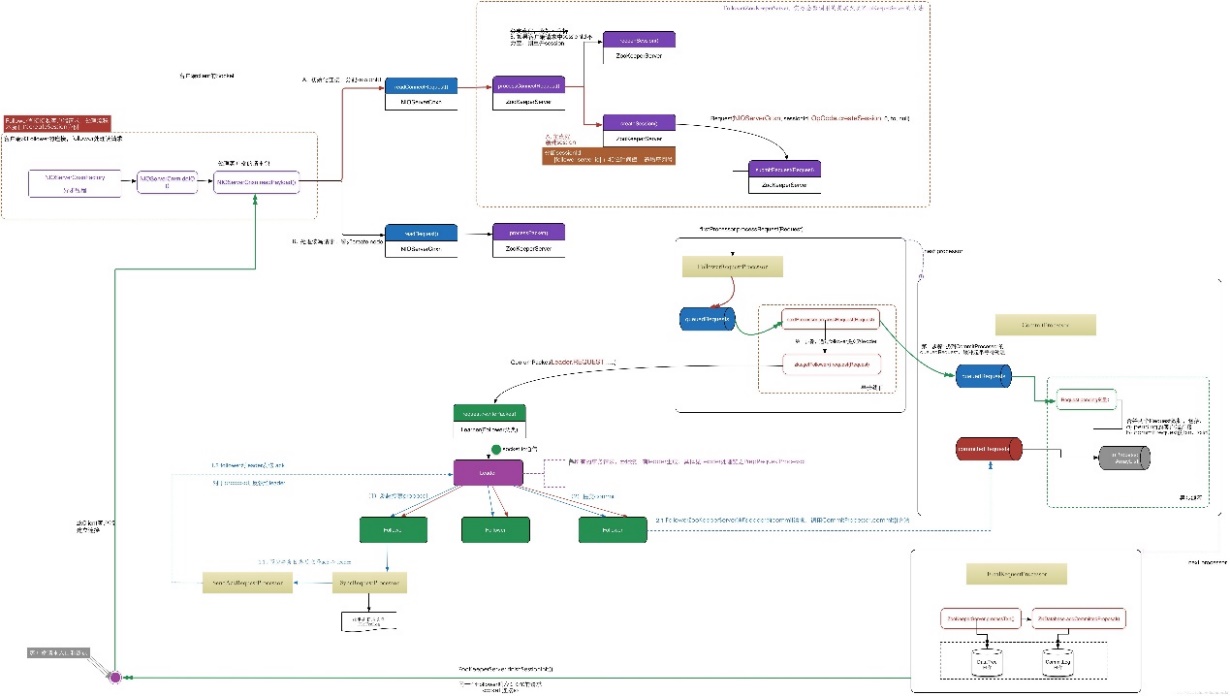
命令：

[root@jizong zookeeper]# /usr/zookeeper/zookeeper-3.5.4-1/bin/zkCli.sh -server 127.0.0.1:2181



# 3 补充说明

## 3．1 Leader—Follower 客户端连接并创建session（以后再消化）

参考：<https://blog.csdn.net/huyangyamin/article/details/77746683>

## 3．2 vi 和vim

它们都是多模式编辑器，不同的是vim 是vi的升级版本，它不仅兼容vi的所有指令，而且还有一些新的特性在里面。

vim的这些优势主要体现在以下几个方面：

1、多级撤消

在vi里，按 u只能撤消上次命令，而在vim里可以无限制的撤消。

2、易用性

vi只能运行于unix中，而vim不仅可以运行于unix,windows ,mac等多操作平台。

3、语法加亮

vim可以用不同的颜色来加亮你的代码。

4、可视化操作

就是说vim不仅可以在终端运行，也可以运行于x window、 mac os、 windows。

5、对vi的完全兼容