**zookeeper集群环境搭建详细图文教程**

**zhoubang @ 2018-01-02**

**文档目录**

**[友情介绍](#_Toc10923)** [1](#_Toc10923)

**[软件环境](#_Toc337)** [2](#_Toc337)

**[注意点](#_Toc18336)** [2](#_Toc18336)

**[环境安装](#_Toc28957)** [2](#_Toc28957)

[1、 新建用于存储安装包以及软件安装的目录 2](#_Toc31454)

[2、 下载安装zookeeper 3](#_Toc6308)

[3、 解压zookeeper压缩包 4](#_Toc25894)

**[单节点zookeeper配置](#_Toc18770)** [4](#_Toc18770)

[1、 配置zoo.cfg文件 4](#_Toc20011)

[2、 配置zookeeper的数据存储目录 5](#_Toc16439)

[3、 新建myid文件 6](#_Toc27197)

[4、 配置zookeeper集群节点 6](#_Toc9695)

[5、 启动zookeeper服务 7](#_Toc22343)

[6、 检查zookeeper服务是否成功启动 8](#_Toc8659)

**[多节点zookeeper配置](#_Toc17111)** [8](#_Toc17111)

[1、不同虚拟机下的zookeeper配置 9](#_Toc18177)

[2、 验证不同虚拟机下的zookeeper服务 10](#_Toc24500)

**[zookeeper客户端连接测试](#_Toc9097)** [11](#_Toc9097)

[1、 连接节点自身 13](#_Toc25760)

[2、 跨节点连接 13](#_Toc1349)

**[停止任意zk节点，观察zk自动选举leader效果](#_Toc6787)** [15](#_Toc6787)

**[恢复停止的zk节点，观察zk集群节点最新角色分配情况](#_Toc19574)** [17](#_Toc19574)

**[总结](#_Toc7176)** [18](#_Toc7176)

**[经验补充](#_Toc3519)** [19](#_Toc3519)

**友情介绍**

QQ技术交流群 **470414533**

个人网站 **[www.2b2b92b.com](http://www.2b2b92b.com)**

开源中国 **<https://gitee.com/zhoubang85>**

**在这里，你可以学习到什么？**

* 你可以完整的学会zookeeper集群环境的搭建和配置
* 通过测试，清楚的知道zookeeper对于存在宕机节点情况下的leader自动选举

该文档不仅仅包含zookeeper集群环境搭建的内容，也补充了zookeeper的主从节点等相关的介绍内容，方便学习研究，加深对zookeeper的学习。

文档内容文字较多，图文丰富，请耐心阅读，不会让你失望。

## 软件环境

如果自己没有服务器的话，可以在本机安装WMware虚拟机，如何安装虚拟机，请自行解决。

VMware-workstation-full-14虚拟机下载地址(里面有永久注册码)

<https://pan.baidu.com/s/1eR9T9lw>

CentOS-6.5\_64位下载地址

<https://pan.baidu.com/s/1bZegMi>

Zookeeper\_3.4.9下载地址

<https://archive.apache.org/dist/zookeeper/zookeeper-3.4.9/zookeeper-3.4.9.tar.gz>

## 注意点

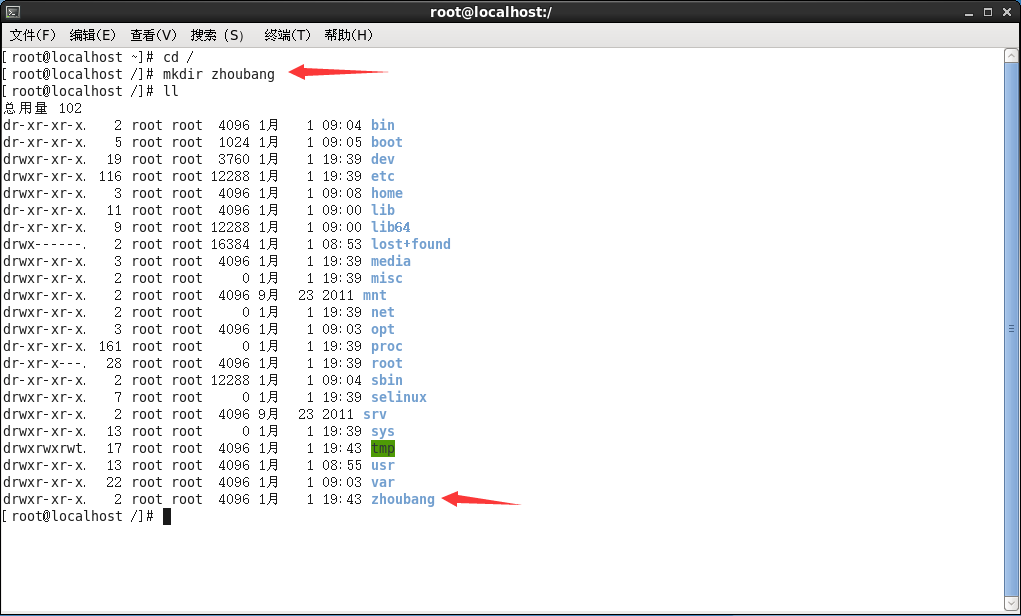
在登录虚拟机中的Linux系统的时候，登录用户请使用系统自带的root，密码即为安装过程中自定义的密码。

不要使用自定义的用户名登录，不然有些目录是没有写的权限的，避免出现一些意外的情况，建议使用root最高权限账户登录.

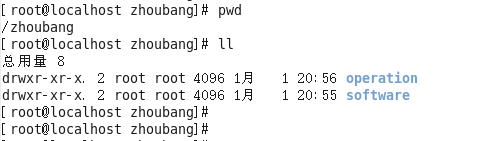
## 环境安装

**我们先在一台虚拟机中安装配置好单个ZK服务，成功后我们克隆出多个虚拟机，就无需在其他虚拟机中重新下载安装zookeeper等重复操作。**

### 新建用于存储安装包以及软件安装的目录

在 / 目录下创建一个以我的名字zhoubang为名称的目录，个人习惯罢了，这个请自行命名；

同时在zhoubang目录下再新建2个目录software（存储安装包）、operation（软件安装的目录），这些都是个人习惯，你们也可以自行选择目录创建。



### 下载安装zookeeper

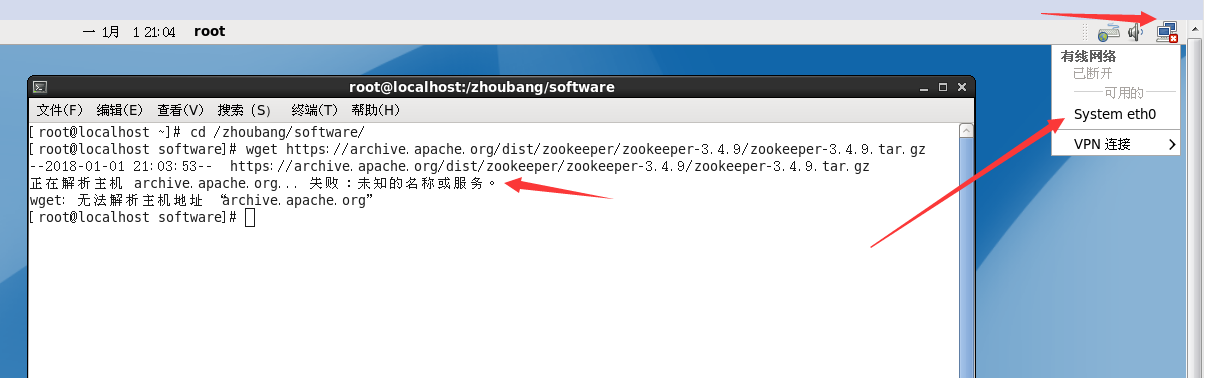
* 在线下载：推荐使用此方式，简单方便。
* 本地上传安装包 ：前提是本地已经有zookeeper的安装包了。你可以通过FTP等上传工具直接上传到对应目录下.

**zk在线下载与安装**：

通过命令 cd /zhoubang/software 进入安装包的存储目录，使用wget命令下载zookeeper（下载地址文档上面已经提供）；如下图，回车即可下载：

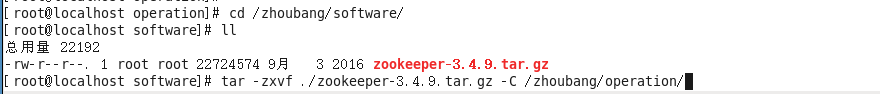


回车之后如果出现下图所示的<未知的名称或服务>错误，请检查当前虚拟机下的Linux系统的网络连接是否正常；

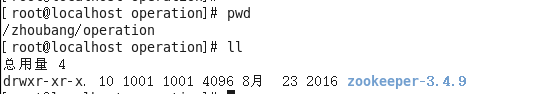


### 3、解压zookeeper压缩包

进入 /zhoubang/software目录， 通过命令 tar -zxvf ./zookeeper-3.4.9.tar.gz -C /zhoubang/operation/ 解压zookeeper包到指定目录，这里我解压到 /zhoubang/operation 目录下，如图：

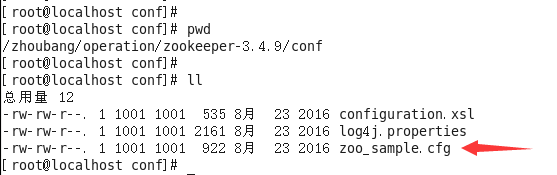


此时，在/zhoubang/operation目录下，就会有zookeeper的资源，如下图：



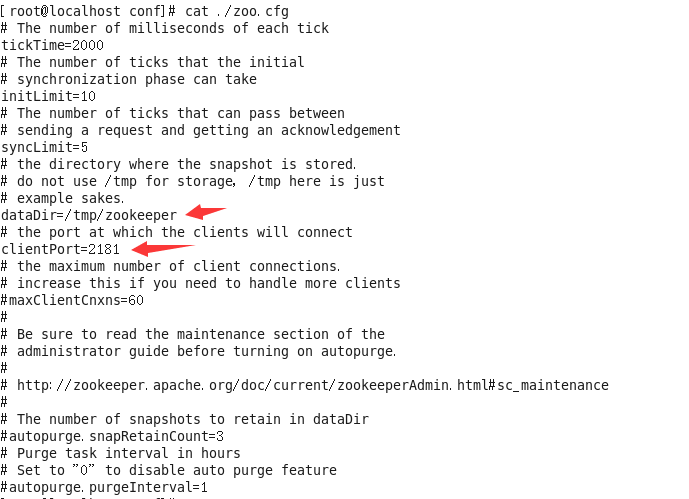
## 单节点zookeeper配置

### 1、配置zoo.cfg文件



如上图，zk提供的配置文件名称为zoo\_sample.cfg ，首先我们需要将此文件重命名为zoo.cfg。因为zookeeper默认会加载zoo.cfg的文件，这是zk的规定。

查看一下zoo.cfg配置文件里面的内容有什么，如下图：



其中，最常用的配置有以下几个：

· **tickTime**：这个时间是作为 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。

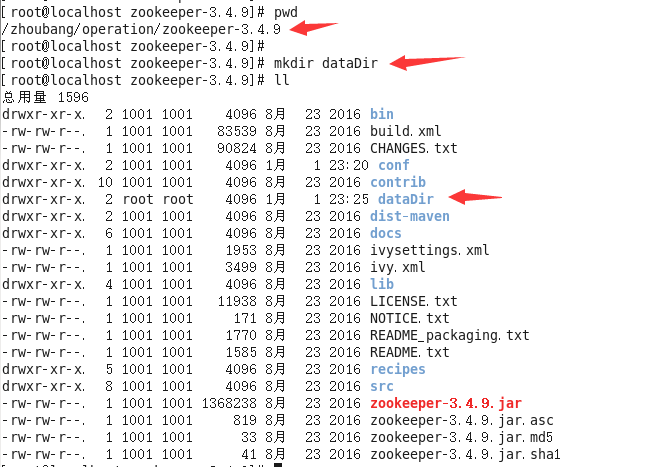
· **dataDir**：顾名思义就是 Zookeeper 保存数据的目录，默认情况下，Zookeeper的日志文件是在bin目录下，有一个zookeeper.out文件。

· **clientPort**：这个端口就是客户端连接 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。伪集群模式下，这个端口需要配置成不同的。如果是多台虚拟机或者服务器下，则无需更改。

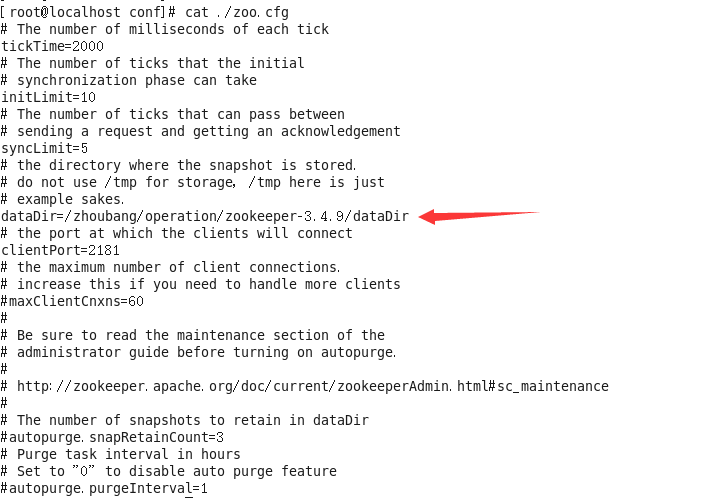
### 2、配置zookeeper的数据存储目录

这里我就在zookeeper的安装根目录下，新建了一个dataDir目录，用于存储zookeeper的数据。

如下图：

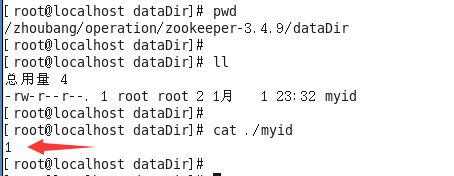


修改zoo.cfg中的dataDir配置，即为刚刚新建的dataDir的目录地址。最终修改的结果如下图：



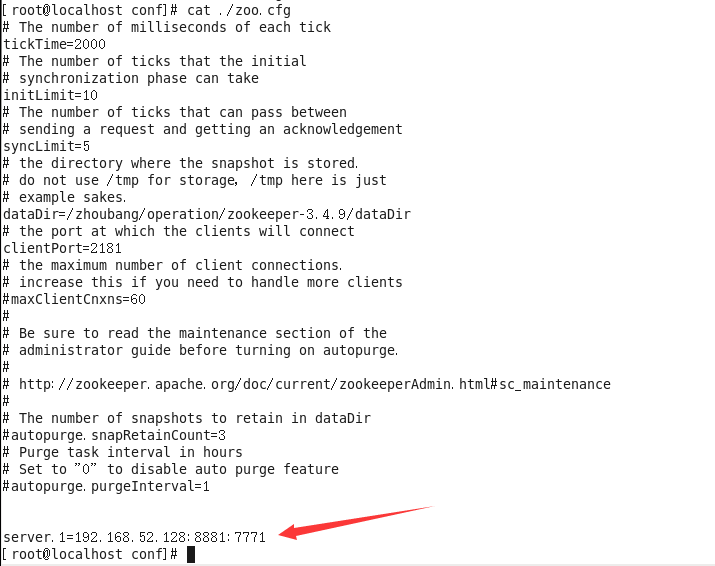
### 新建myid文件

在刚刚新建的dataDir目录下，新建一个myid文件，该文件里面的内容，这里我就填写为1，至于数值1有何作用和意义？是否可以为其他数字？当然，我也不多做玄乎了，这数值是可以自定义的，单范围只能是1-255之间。具体的含义，下面会介绍。如下图：



### 配置zookeeper集群节点

在zoo.cfg文件中的最下面，配置zk集群节点信息；当前配置的是单台虚拟机，所以配置内容目前只有一个server节点（等将其他虚拟机克隆并启动之后，就可以将其他虚拟机信息添加到此），效果如图：



**【server】的配置含义说明：**

server.A=B:C:D

其中 A 是一个数字，表示这个是第几号服务器；

B 是这个服务器的 ip 地址；

C 表示的是这个服务器与集群中的 Leader 服务器交换信息的端口；

D 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（也就是在一台服务器上搭建多个ZK服务节点），由于 一台服务器下的IP地址肯定一样，所以不同的 Zookeeper 实例通信端口号不能一样，故需要给它们分配不同的端口号。

相反，如果是在不同的服务器下（不同的虚拟机，IP地址不同）搭建ZK的话，由于IP地址不同，则C与D对应的通信端口可以保持一致。当然也可以配置成其他的端口。

**【注意点】：**

其中的server.1里面的数字1，是我们在myid文件中指定的数值。代表不同的zk节点。

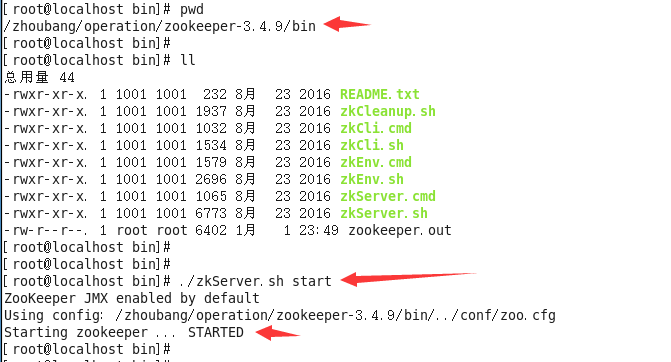
同时，IP地址配置也要与server.1所在的服务器节点的IP地址一致。

后面的2个通信端口，随意指定，只要不与其他端口冲突就行。

### 启动zookeeper服务

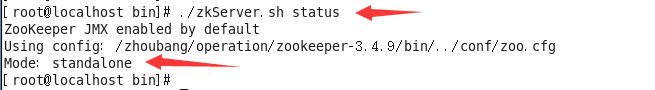
进入zookeeper的bin目录下，执行命令启动zookeeper服务

如下图：



### 检查zookeeper服务是否成功启动

在bin目录下，通过命令 ./zkServer.sh status 查看启动状态，如图：



由于我只启动了一个ZK服务，所以Mode后面的信息为standalone，意思独立节点。

## 多节点zookeeper配置

到上面为止，我们只是在单台虚拟机中成功搭建了zookeeper服务。接下来我们就开始在其他的虚拟机节点中配置zookeeper集群节点信息。

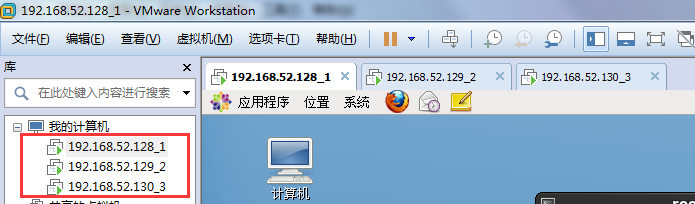
我本机电脑上克隆出了3个虚拟机（当然你也可以克隆很多个，为了学习方便，只配置3个节点就够了）。IP地址分别是：

192.168.52.128

192.168.52.129

192.168.52.130

虚拟机克隆情况，如图所示：



\* 这里虚拟机的命名，是方便大家查看的。 后面的数值1、2、3分别是我在zookeeper的myid文件中指定的数值。

\* 每台虚拟机中的系统环境以及zookeeper的安装配置，都与上面的单节点的zookeeper配置一致。

### 1、不同虚拟机下的zookeeper配置

需要更改的地方有以下几点：

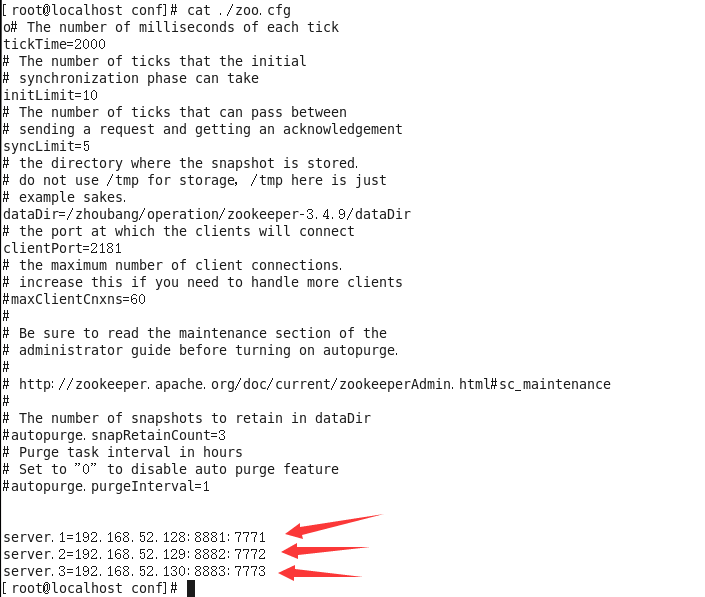
* **myid文件**
* **zoo.cfg 文件**

1.1首先我们修改192.168.52.128 对应的虚拟机中的zookeeper配置。

myid文件中的数值为1（当前第一台虚拟机下的zookeeper可以不用修改，只修改另外2台即可）

然后修改zoo.cfg文件，在最下面添加ZK节点配置.

最终配置效果如下：

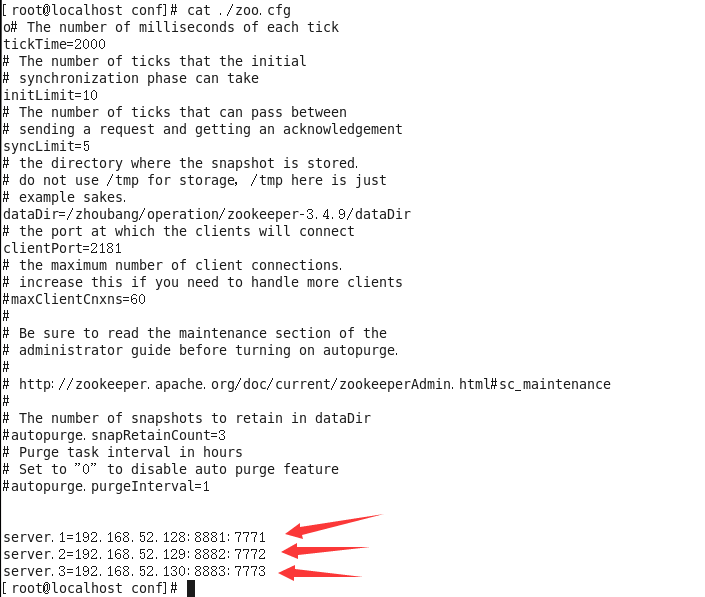


通过配置我们可以看到，需要将其他的zookeeper节点信息配置进来。

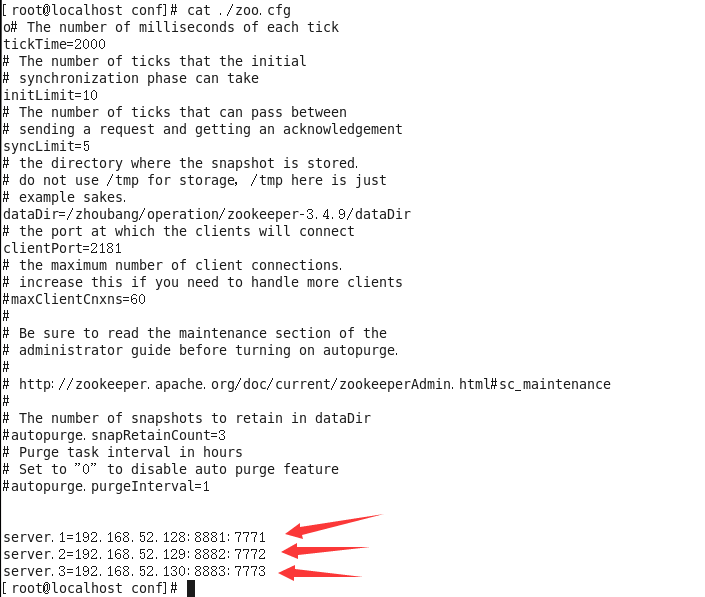
再强调一遍，server.1、server.2、server.3 中的数值1、2、3，对应的是dataDir的myid文件中的数值，不要配置错了，包括IP也要对应。

按照上面的操作步骤，分别对192.168.52.129、192.168.52.130的虚拟机中的zookeeper进行配置。如下：

1.2、将192.168.52.129虚拟机中的zookeeper下的myid中的数值填写 2 ，对应的zoo.cfg文件最下面的配置，和上图一样的server配置，可以直接copy过去，最终配置如下：



1.3、将192.168.52.130虚拟机中的zookeeper下的myid中的数值填写 3 ，对应的zoo.cfg文件最下面的配置，和上图一样的server配置，可以直接copy过去，最终配置如下：

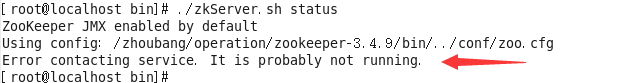


至此，我们3个zookeeper节点的配置已经完毕。下面我们就启动zookeeper服务，验证集群是否搭建成功！

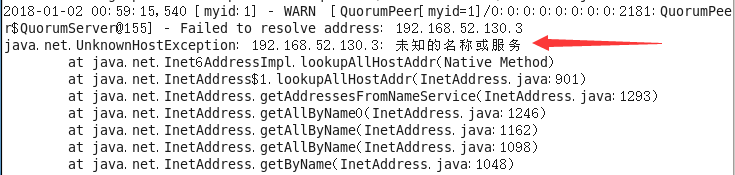
### 2、验证不同虚拟机下的zookeeper服务

分别进入每个虚拟机下的zookeeper的bin目录下，执行命令 ./zkServer.sh start 启动ZK服务！

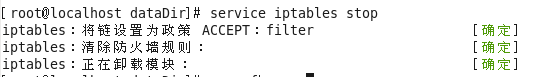
通过命令 ./zkServer.sh status查看ZK启动状态。如果出现了以下错误：



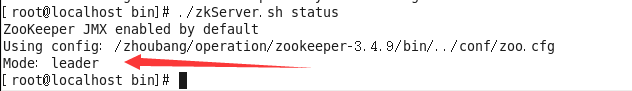
我们去查看一下日志（bin目录下zookeeper.out日志文件）。查看里面的内容，有一段错误信息如下：

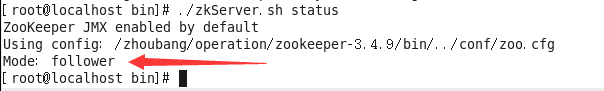


这种情况下，网上很多人都说是防火墙开启导致端口无法通信造成的；那我就在每个虚拟机上，把防火墙关闭；可以使用命令 service iptables stop 临时修改防火墙状态，关闭防火墙。



防火墙关闭了，这时候全部重启zookeeper，再通过 ./zkServer.sh status 命令查看启动状态，如果出现下面的任意一个结果，说明ZK集群环境已经搭建成功，如下图：





到此，zookeeper集群环境已经搭建完毕！

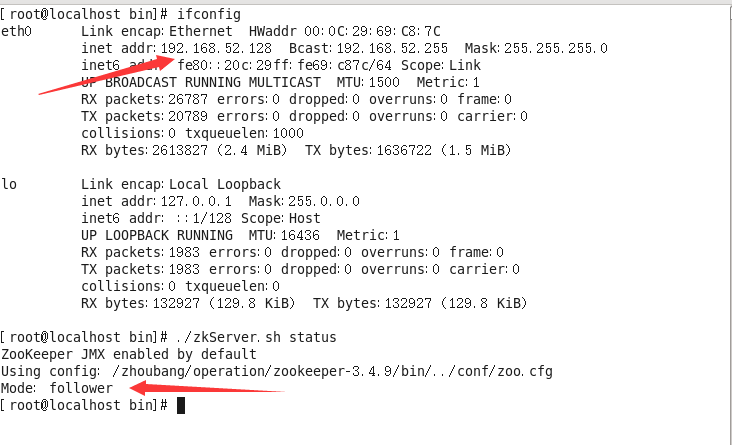
## zookeeper客户端连接测试

我们可以通过zkCli.sh提供的命令进行客户端连接测试，检测是否可以跨节点连接。

**先看一下各zookeeper节点的角色关系**

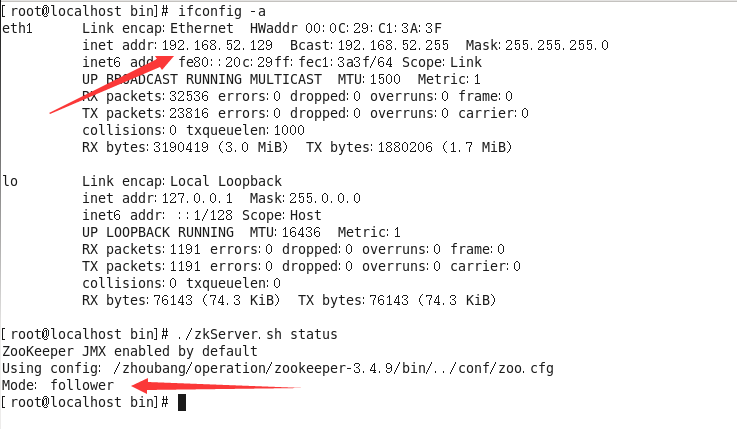
**192.168.52.128 follower**

如下图：



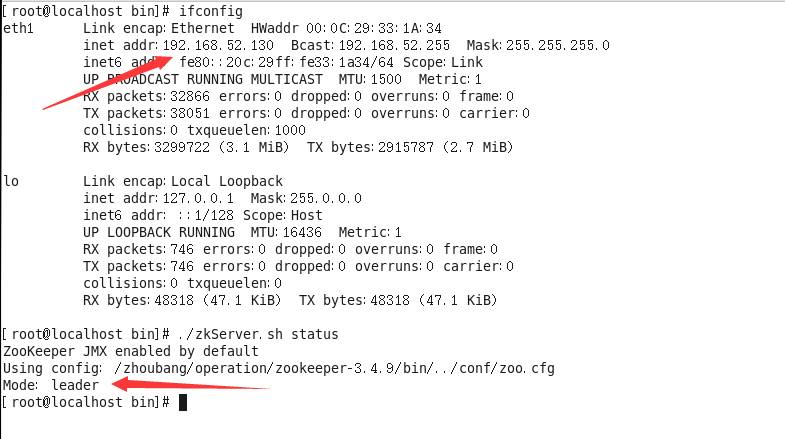
**192.168.52.129 follower**

如下图：



**192.168.52.130 leader**

如下图：



### 连接节点自身

我们先在任意一个zookeeper节点的bin目录，通过命令./zkCli.sh -server 192.168.52.129:2181连接自己本身（这里我就在IP为192.168.52.129的服务器下做客户端连接测试），

如图：



出现上图所示内容，说明zookeeper客户端连接OK。退出命令为 quit

### 跨节点连接

既然是集群环境，那肯定可以在任意zookeeper节点下，去连接任意的zookeeper节点。

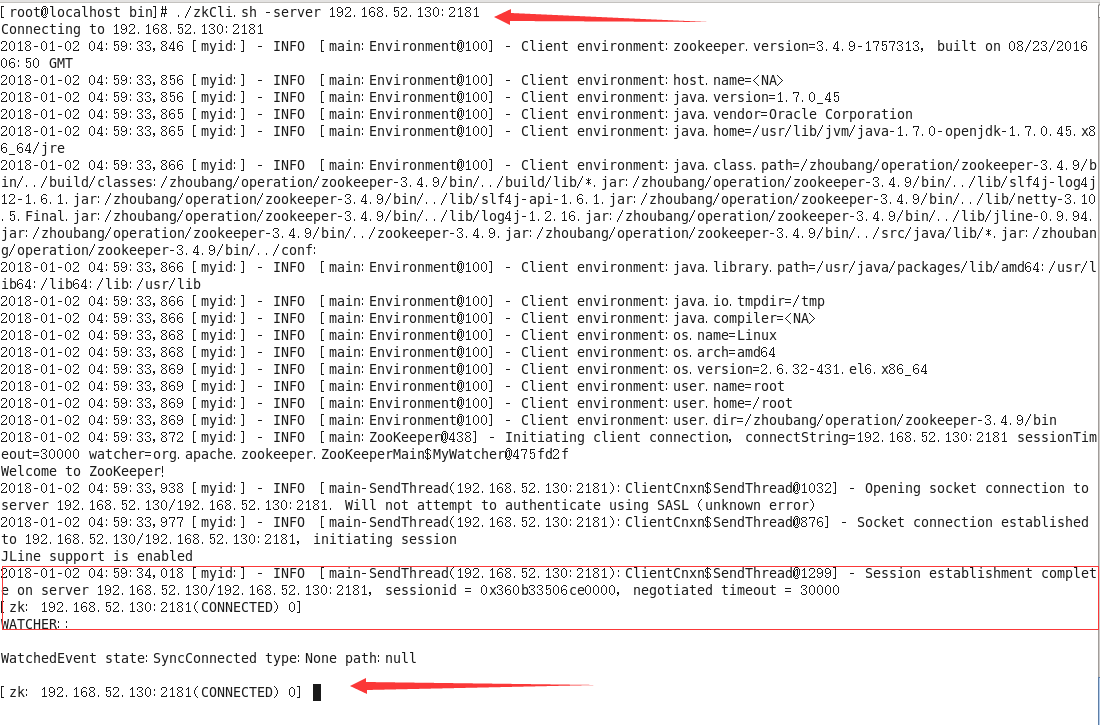
我们试一下就知道了，同样的，通过命令./zkCli.sh -server 192.168.52.129:2181连接任意节点。上面已经做了连接自身的测试了，现在我们就来操作，在192.168.52.129的服务器上去连接其他的2台服务器上的zookeeper服务。

* **先连接192.168.52.128服务器上的zookeeper服务，如图：**



从图中可以看出，跨服务节点的连接是OK的！

* **连接192.168.52.130服务器上的zookeeper服务，如图**



从图中可以看出，跨服务节点的连接是OK的！

## 停止任意zk节点，观察zk自动选举leader效果

既然是集群，就必然脱离不了主从的角色，zookeeper自身会根据节点的生存、死亡状态，自动的进行选举切换。

这里我们将角色为leader的解决给停止掉，然后看一下其余的2个处于正常情况的角色分配情况，看看zookeeper会不会自动进行选举leader并切换。

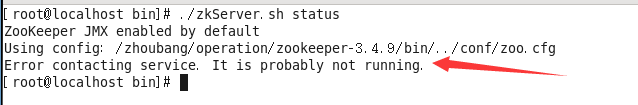
再说明一下当前3个zk节点的角色关系：

**192.168.52.128 follower**

**192.168.52.129 follower**

**192.168.52.130 leader**

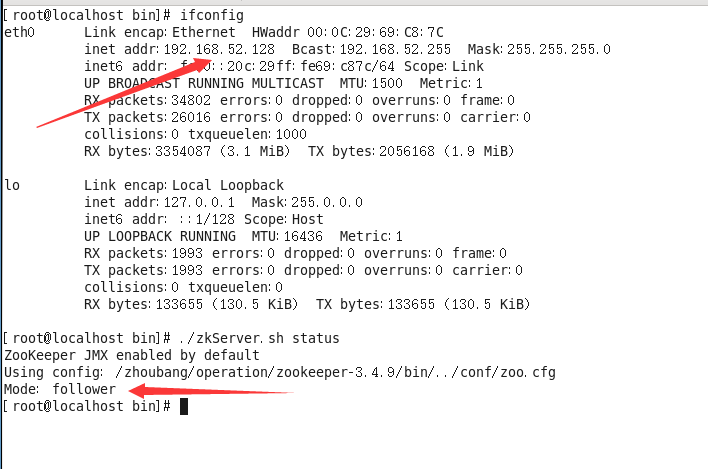
我们先停止192.168.52.130这个leader主节点，进入zookeeper的bin目录，执行命令 ./zkServer.sh stop 停止服务。然后通过命令 ./zkServer.sh status 查看当前zk节点状态是否已经停止，如图：



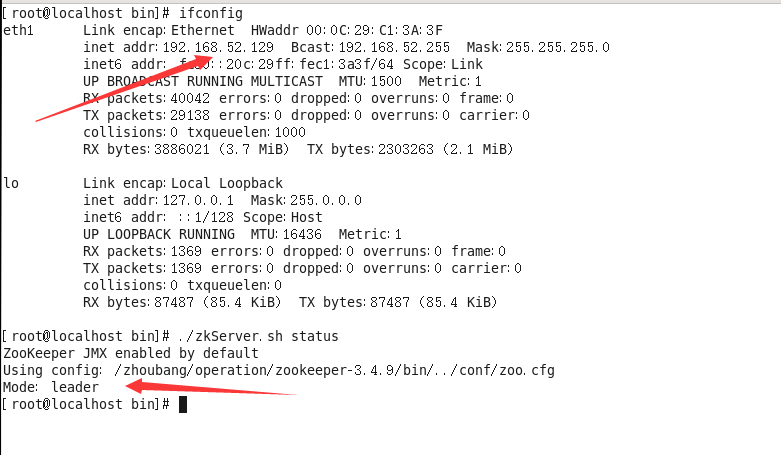
说明之前的leader角色的节点已经停止成功了！

乘胜追击，大约等待5到30秒左右，我们去看一下另外2个zk节点的角色状态，经过查看，最新的角色分配情况如下：

**192.168.52.128 follower**



**192.168.52.129 leader**



从以上2张图中可以看到，192.168.52.129这个服务器上的zk节点的角色，从之前的**follower**变成了**leader，**192.168.52.128角色没有发生变化。

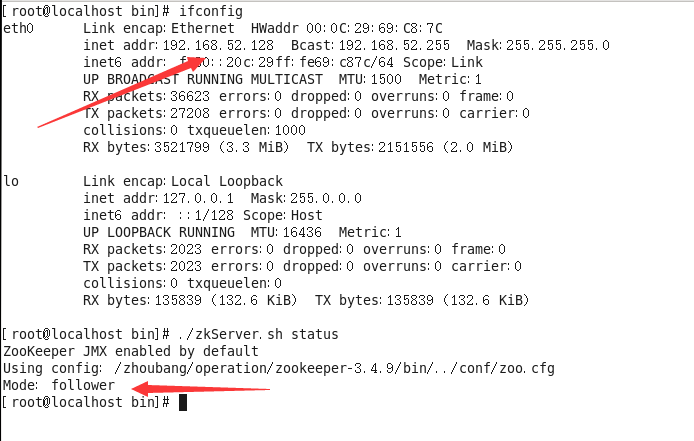
这就证明了，zookeeper会自动进行leader选举，而异常停止的zk节点会被剔除在外，不再提供服务。

### 恢复停止的zk节点，观察zk集群节点最新角色分配情况

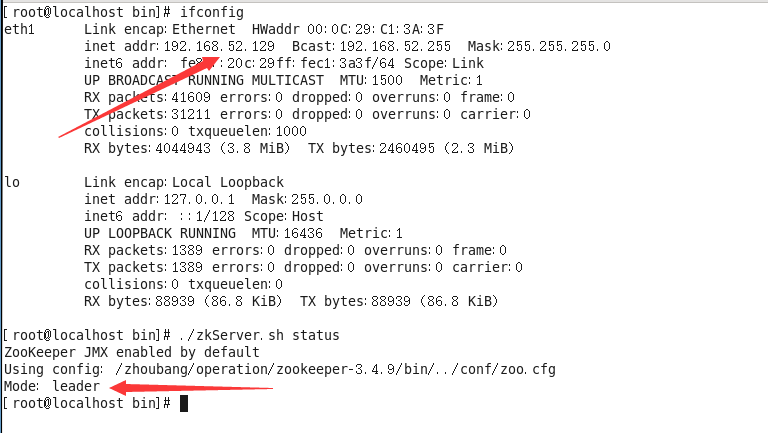
上面的操作，我们知道了zk可以自动进行leader选举，剔除死亡的服务节点；那么，当我们恢复刚刚停止的zk节点的话，那这3个节点之间的觉得分配情况又是如何呢？

启动刚刚停止的zk节点（192.168.52.130），然后在每一个zk的bin目录下，通过命令 ./zkServer.sh status 查看角色分配情况，结果如下图：

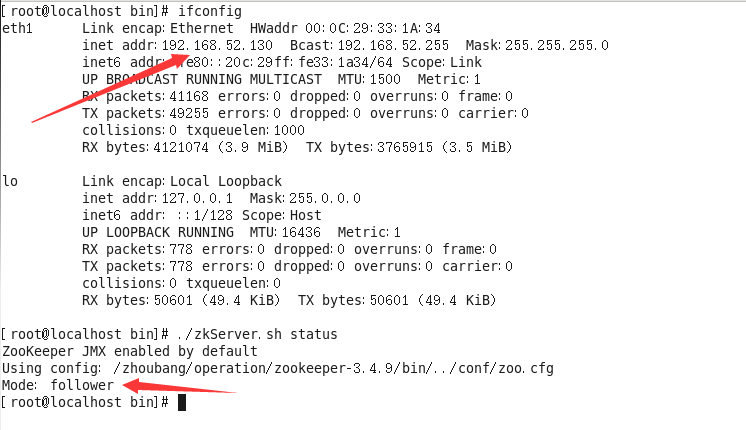
**192.168.52.128 follower**



**192.168.52.129 leader**



**192.168.52.130 follower**



从图中可以观察到，我们把之前停止的zk节点（192.168.52.130）恢复后，这个节点的角色为follower，并没有恢复原先停止之前的leader角色。

这说明，当zk进行重新leader选举后，除非leader意外宕机，否则角色不会发生变化。

## 总结

经过了详细的操作以及测试，我相信对于新手而言，肯定会学习到不少知识点。这也不枉费我的心血付出啊，写文档好累的啊~不过一切值得，文档完成的那一刻，将会永久留存在互联网中，供他人学习使用。

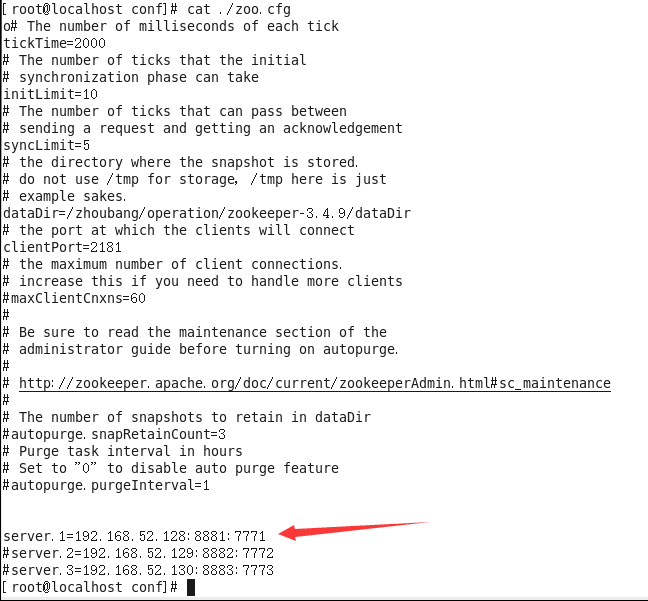
其实zookeeper的集群环境搭建以及配置，是很简单的。只有亲身操作，才感受到很简单，没有那么复杂。不要被“集群”等专业术语所吓到。

本文档写的很详细了，不仅仅只包含环境搭建和配置，也对zk的一些特点以及主从选举进行了案例测试和说明，加深此文档的知识储备水平。

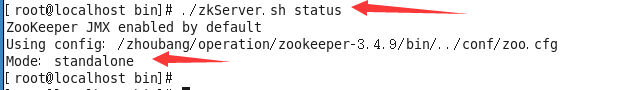
好了，就说到这吧，祝各位学习顺利~

## 【经验补充】

### 如果在zoo.cfg文件中只配置了一个server节点，且server的内容为当前服务器的信息；是否可以启动成功呢？比如下面的情况：



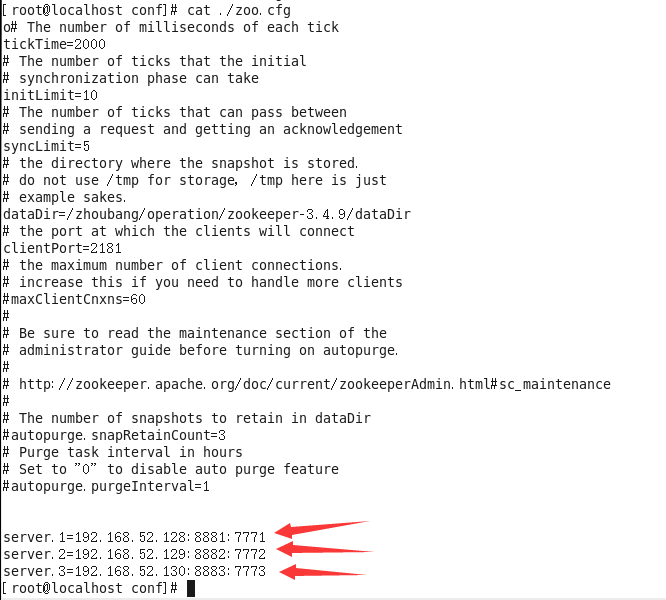
则在通过命令 ./zkServer.sh start 启动zk服务的时候，是可以正常启动的。如下图：



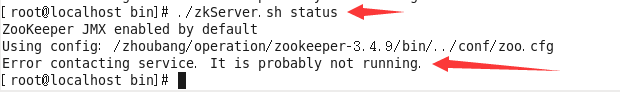
可以看到，Mode：standalone 的信息，说明是ZK单点。因为我们在zoo.cfg中只配置了自己的server信息.

当然，在zoo.cfg中如果一个server配置都没有的话，也是这样的效果，可以成功启动。

### 与上面第一点相反，如果在zoo.cfg中配置了大于1个的server的配置，如果只启动一个ZK实例，会启动成功吗？如下图：



这时候如果上面3个ZK没有一个启动的话（都是停止状态），如果此时在任意一个ZK的bin目录下通过命令 ./zkServer.sh start启动的话，再使用命令 ./zkServer status 查看启动状态，你会发现，会一直报一个错误，如图：



这种错误的原因，就是因为你只启动了一个ZK实例导致的；为什么会这样呢？

你想想，既然是在zoo.cfg中配置了3个server，说明是集群的配置，既然是集群，那么必然会有2种角色，一个是leader、一个是follower；这就说明，我们的ZK服务节点必须至少有2台是正常服务的。

如果只有一台实例启动，请问这台是leader角色还是follower角色呢？都不是！原因很简单，如果唯一启动的这台ZK突然宕机了，请问，下一个接手的ZK服务节点是谁呢？肯定没有。这样的话，也就失去了集群的意义。

因此，ZK启动的时候，本身会进行验证，必须启动至少2台ZK实例，才可以正常提供leader选举服务。否则就会一直出现上图的错误信息；

1. **如下图所示，在zoo.cfg中server配置中，是否可以将其中的IP地址使用域名或者其他易于识别的名称代替？如图：**



思考：我们都知道，一个服务器正常都是使用域名来访问的，使用域名的好处就是防止IP发生变化导致服务器频繁修改IP引用。在上图的zookeeper集群的配置中，使用的是IP地址的配置访问，其实使用IP来这么做的一个缺陷，肯定就是刚刚说的IP变化所带来的服务不可用。所以，线上环境一般建议通过域名来配置，方便扩展维护。

问题：如何将192.168.52.128、192.168.52.129、192.168.52.130通过域名或者其他易于识别的别名替换？

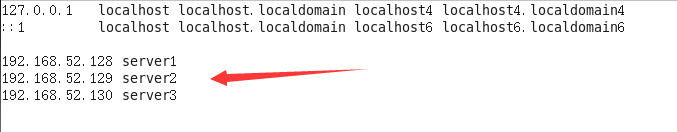
答案：解决方案肯定是有的。

我们就拿其中一个zookeeper节点进行讲解配置，其他的节点配置都是一样的。

我这里选择了 192.168.52.128这个服务器进行zookeeper集群信息的优化配置；

1. **首先，肯定需要设置IP地址与域名的映射关系，类似于window系统中的host。**

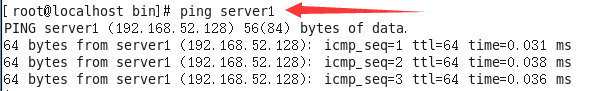
通过命令vi编辑 /etc/hosts 文件，进入编辑区域。最终配置如图：



文件内部最顶层是系统默认有的配置，我们不用理会。我们只需要添加zookeeper的3台服务器节点的IP与别名的映射关系即可。

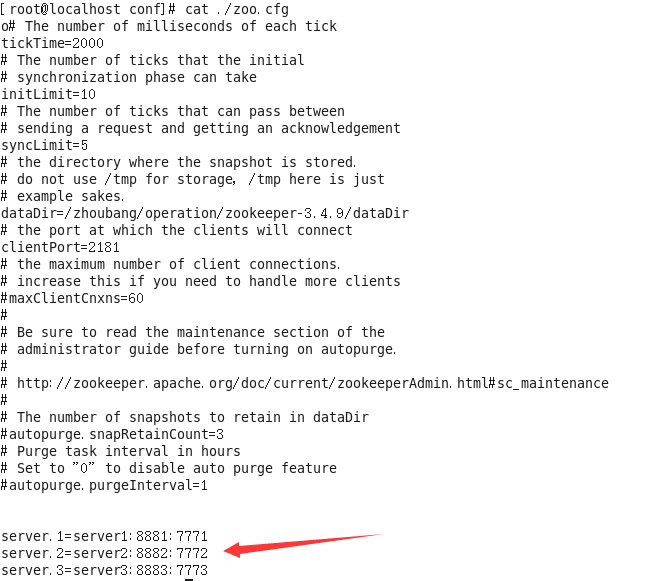
IP地址右侧的配置，可以随意指定别名或者域名，方便区分标识即可。

保存之后，你也可以通过ping命令来测一下网络状态，如图：

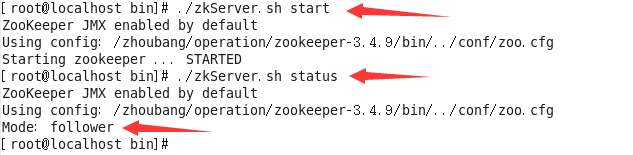


图中所示结果，说明配置的没有问题。

1. **修改zookeeper的zoo.cfg文件，将里面的server配置部分中的IP地址，替换为上面对应配置的别名，zoo.cfg最终配置如下：**



1. **重启zookeeper，进行测试验证，是否可以成功启动**



* 从上图会看到，zookeeper重启后，zookeeper服务是正常的。故：按照上面的/etc/hosts的映射配置，是可以实现IP地址替换为别名的。
* 当然，有的朋友或许会问， /etc/hosts 文件下不还是有IP地址的配置吗，有什么提升和好处？

其实很容易理解，如果在zoo.cfg中使用IP配置，当zookeeper节点启动之后，如果IP发生了变化，如果zookeeper服务不重启的话，zookeeper始终会使用之前配置文件中的IP进行通信，就造成服务始终不可用，还是需要手动去调整IP才可以。

而通过在 /etc/host 中进行IP地址的映射，即使IP发生了变化，我们只需要在 /etc/hosts 里面配置即可，zookeeper服务是不用重新重启的，实现了无缝对接和维护扩展。

这也就是软件架构中最常见的优化思想。