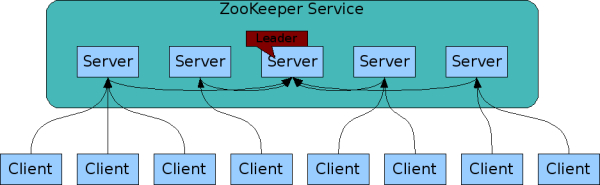
**Zookeeper学习文档**

# Zookeeper简介

ZooKeeper是一个开源的分布式框架，提供了协调分布式应用的基本服务。它向外部应用暴露一组通用服务——分布式同步（Distributed Synchronization）、命名服务（Naming Service）、集群维护（Group Maintenance）等，简化分布式应用协调及其管理的难度。

它是Google的Chubby一个开源的实现。它本身可以搭建成一个集群，这个zk集群用来对应用程序集群进行管理，监视应用程序集群中各个节点的状态，并根据应用程序集群中各个节点提交的反馈信息决定下一步的合理操作。



# Zookeeper下载和安装

ZooKeeper的下载很简单，安装就是解压一下，也很简单。

https://archive.apache.org/dist/zookeeper/zookeeper-3.4.10/zookeeper-3.4.10.tar.gz

# 部署3个节点的Zookeeper伪分布式集群

在同一台服务器上，部署一个3个ZooKeeper节点组成的集群，这样的集群叫伪分布式集群，而如果集群中的3个节点分别部署在3个服务器上，那么这种集群就叫真正的分布式集群。

这里，记录一下搭建一个3节点的伪分布式集群的过程，真正的分布式集群的搭建过程和伪分布式的过程类似，稍有不同，我会在下面指出来的。

首先，建立一个集群安装的目录，就叫zookeeper。其次，在这个目录的下面解压三份ZooKeeper，形成3个节点，每一个目录中的ZooKeeper就代表一个节点。

这样就形成了如下的安装目录结构：

/root/zookeeper

          |----zookeeper1/

          |----zookeeper2/

          |----zookeeper3/

### 3.1 解压zookeeper安装包

首先在要安装集群的目录中解压zk。

tar -zxvf zookeeper-3.4.10.tar.gz

之后把解压的zk复制出三份来，分别命名为zookeeper1，zookeeper2，zookeeper3，这三个目录中的zk就当成是集群中的3个节点。

### 3.2 为每个节点建立data目录、logs目录和myid文件

在3个节点目录中分别建立data目录、logs目录和myid文件。

下面是zookeeper1上的：

新建目录data：mkdir data

新建目录logs：mkdir logs

新建文件myid：touch data/myid

myid文件的内容是节点在集群中的编号，zookeeper1节点的编号就写成1，后边的zookeeper2的编号是2，zookeeper3的编号就是3。

按照同样的方法，依次在zookeeper2和zookeeper3上都建立以上目录和文件。

### 3.3 为每个节点创建配置文件

在zookeeper1下新增配置文件：cp zoo\_sample.cfg zoo.cfg

配置文件zoo.cfg的内容如下：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | tickTime=2000  initLimit=10  syncLimit=5  dataDir=/root/zookeeper/zookeeper1/data  dataLogDir=/root/zookeeper/zookeeper1/logs  clientPort=2181  server.1=127.0.0.1:8880:7770  server.2=127.0.0.1:8881:7771  server.3=127.0.0.1:8882:7772 |

配置文件中的配置项的含义参见下面的介绍。

用同样的方法，在zookeeper2和zookeeper3的相应位置创建zoo.cfg，文件内容复制zookeeper1的zoo.cfg。只不过需要改动clientport、dataDir、dataLogDir三个配置项，zookeeper2的clientport改为2182，zookeeper3的clientport改为2183，而dataDir和dataLogDir都修改为相应的目录，就好了。

经过以上的配置，zk集群中的3个节点就全部配置好了。

### 3.4 启动zk集群

进入zookeeper集群的第一个节点zookeeper0的bin目录下，启动服务：

|  |  |
| --- | --- |
| 1  2  3 | [root@localhost zookeeper]$  [root@localhost zookeeper]$ cd zookeeper1/  [root@localhost zookeeper1]$ bin/zkServer.sh start |

然后，按照同样的方法，依次启动zookeeper1和zookeeper2的服务。

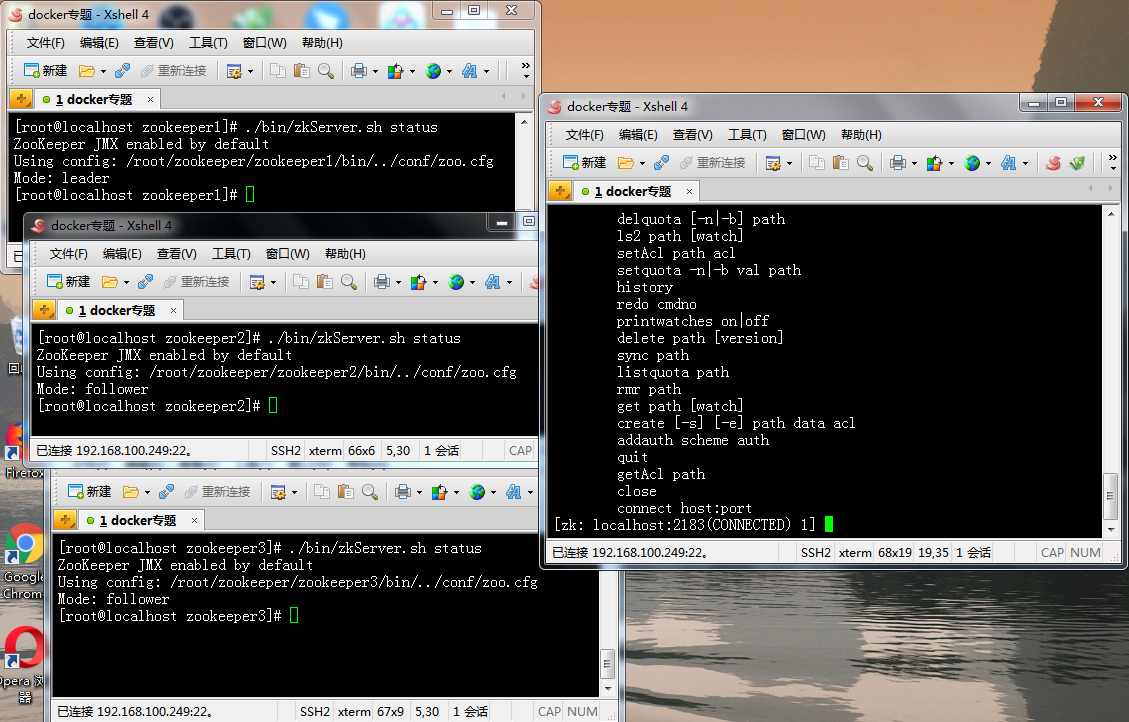
这样zookeeper集群的3个节点都启动起来了。

### 3.5 客户端接入集群

进入zookeeper集群中任意一个节点的bin目录下，启动一个客户端，接入已经启动好的zookeeper集群。这里的server可以填写集群中的任何一个节点的ip，端口号是对应ip的节点的配置文件中clientport的值。

|  |  |
| --- | --- |
| 1 | ./zkCli.sh –server 127.0.0.1:2181 |

下面的截图是我成功之后的3个节点和客户端的图：



以上就是zk集群的搭建过程。

# 真实分布式集群需要注意的地方

真正的分布式集群和伪分布式集群不一样的地方在于配置文件。

1、clientport端口各个节点一样就行。

2、server.1=127.0.0.1:8880:7770中的ip要修改成对应的server的ip，后边的两个端口号不需要不同，各个节点都一样就可以了。

其他地方伪分布式和真正分布式都是一样的。

# ZooKeeper配置文件中的配置项的含义

配置文件中配置项的含义：

* tickTime: zookeeper中使用的基本时间单位，毫秒值，比如可以设为1000，那么基本时间单位就是1000ms，也就是1s。
* initLimit: zookeeper集群中的包含多台server，其中一台为leader，集群中其余的server为follower，initLimit参数配置初始化连接时，follower和leader之间的最长心跳时间。如果该参数设置为5，就说明时间限制为5倍tickTime，即5\*1000=5000ms=5s。
* syncLimit: 该参数配置leader和follower之间发送消息，请求和应答的最大时间长度。如果该参数设置为2，说明时间限制为2倍tickTime，即2000ms。
* dataDir: 数据目录. 可以是任意目录，一般是节点安装目录下data目录。
* dataLogDir: log目录, 同样可以是任意目录，一般是节点安装目录下的logs目录。如果没有设置该参数，将使用和dataDir相同的设置。
* clientPort: 监听client连接的端口号。
* server.X=A:B:C 其中X是一个数字, 表示这是第几号server，它的值和myid文件中的值对应。A是该server所在的IP地址。B是配置该server和集群中的leader交换消息所使用的端口。C配置选举leader时所使用的端口。由于配置的是伪集群模式，所以各个server的B, C参数必须不同，如果是真正分布式集群，那么B和C在各个节点上可以相同，因为即使相同由于节点处于不同的服务器也不会导致端口冲突。

# Zookeeper常用命令

### Zookeeper服务命令

在准备好相应的配置之后，可以直接通过zkServer.sh 这个脚本进行服务的相关操作

1. 启动ZK服务: bin/zkServer.sh start

2. 查看ZK服务状态: bin/zkServer.sh status

3. 停止ZK服务: bin/zkServer.sh stop

4. 重启ZK服务: bin/zkServer.sh restart

5. 连接服务器: zkCli.sh -server 127.0.0.1:2181

### 连接Zookeeper

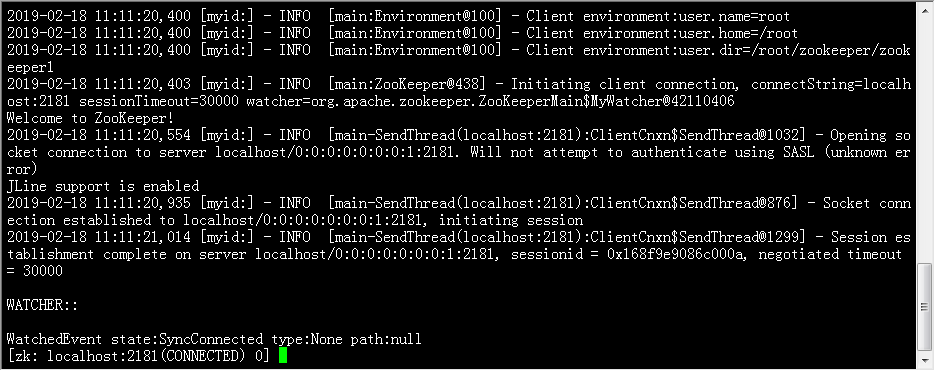
启动ZooKeeper服务之后，我们可以使用如下命令连接到 ZooKeeper 服务：

zkCli.cmd -server 127.0.0.1:2181

Linux环境下：

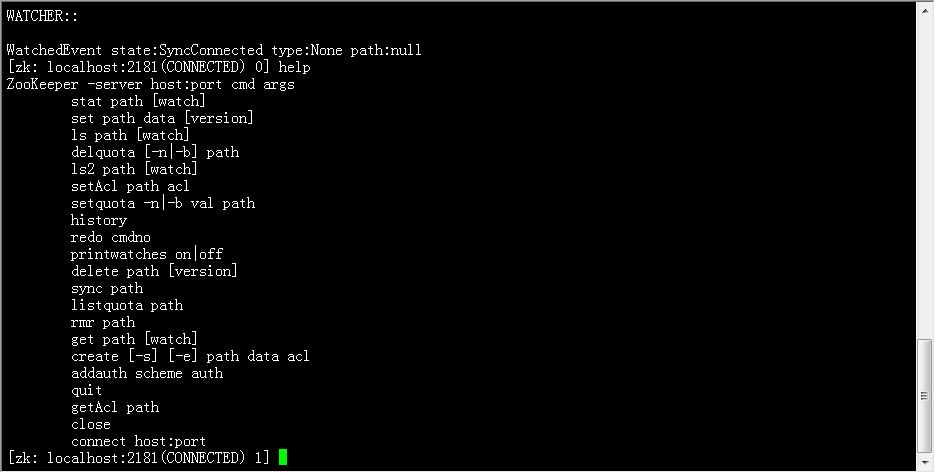
zkCli.sh -server 127.0.0.1:2181

连接成功后，系统会输出 ZooKeeper 的相关环境以及配置信息，如下：



### Zookeeper客户端命令

我们可以使用 help命令来查看帮助：



#### 创建节点

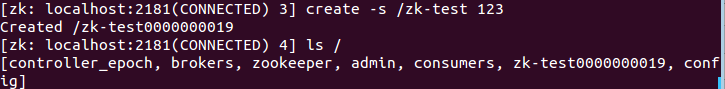
使用create命令，可以创建一个Zookeeper节点， 如

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。

* 创建顺序节点

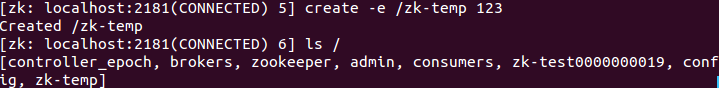
使用 create -s /zk-test 123 命令创建zk-test顺序节点



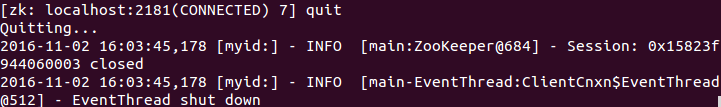
可以看到创建的zk-test节点后面添加了一串数字以示区别。

* 创建顺临时节点

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



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



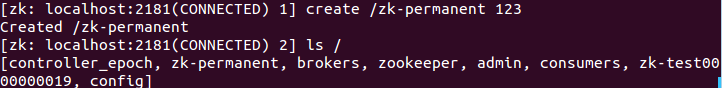
再次使用客户端连接服务端，并使用ls / 命令查看根目录下的节点

IMG_256

可以看到根目录下已经不存在zk-temp临时节点了。

* 创建永久节点

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



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。

#### 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ZK指定节点的数据内容和属性信息。其用法分别如下

ls path [watch]

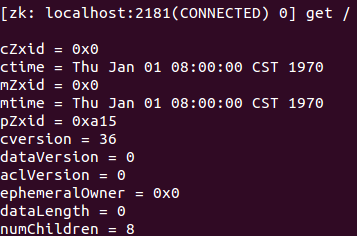
get path [watch]

ls2 path [watch]

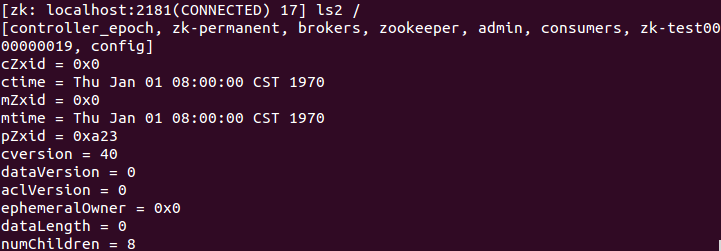
若获取根节点下面的所有子节点，使用ls / 命令即可

IMG_256

若想获取根节点数据内容和属性信息，使用get / 命令即可



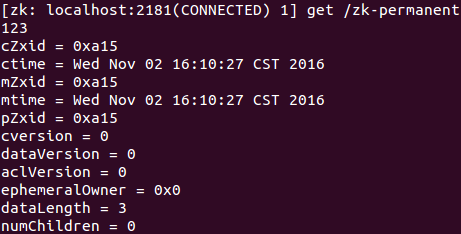
也可以使用ls2 / 命令查看



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



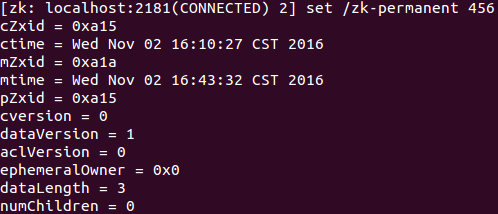
可以看到其数据内容为123，还有其他的属性。

#### 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



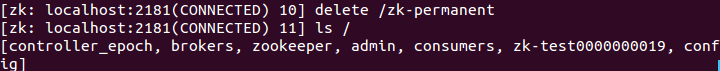
现在dataVersion已经变为1了，表示进行了更新。

#### 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



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。