

Zookeeper安装手册

本手册主要针对没有接触过Zookeeper的同学，介绍Zookeeper的概念，如何下载安装zookeeper，如何搭建zookeeper集群。

Zookeeper的概念

万事万物都有源头，Zookeeper的源头就是它的[官网](#)。官网是这样介绍zookeeper的，Zookeeper是一个开源的分布式的分布式应用程序协调服务。最开始由雅虎公司模仿google的chubby开发而来，后来作为了apache的项目，用在大数据、分布式服务领域。

安装Zookeeper

安装

安装1.6版本以上JDK，配置环境变量 下载[zookeeper 4.3.13版本](#)

解压后的conf目录，增加配置文件，复制zoo_sample.cfg改名为默认配置文件zoo.cfg 启动服务端
bin/zkServer.sh start 使用ps命令检测zookeeper是否启动成功

解压后的conf目录，增加配置文件，复制zoo_sample.cfg改名为默认配置文件zoo.cfg 启动服务端
bin/zkServer.sh start

这个操作命令如下：

```
wget https://archive.apache.org/dist/zookeeper/zookeeper-3.4.13/zookeeper-3.4.13.tar.gz
tar -xvzf zookeeper-3.4.13.tar.gz
sudo mv zookeeper-3.4.13 /usr/local/zookeeper
cd /usr/local/zookeeper
sudo cp conf/zoo_sample.cfg conf/zoo.cfg
sudo bin/zkserver.sh start
```

使用ps命令检测zookeeper是否启动成功

```
ps -ef|grep zookeeper
```

关键配置

```
tickTime=2000          # 一个心跳的时间，单位毫秒
dataDir=/var/lib/zookeeper # zookeeper数据存放的地方
clientPort=2181         # zookeeper客户端访问端口号
```

测试

客户端连接： bin/zkCli.sh -server 127.0.0.1:2181

```
[zk: localhost:2181(CONNECTED) 0] 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
```

客户端连接成功

Zookeeper集群搭建

zookeeper集群要求由奇数个服务节点组成，下面我们就来搭建一个本地3节点伪集群。

搭建

复制3份zookeeper安装目录：zookeeper、zookeeper-01、zookeeper-02

分别修改zoo.cfg配置

```
tickTime=2000
dataDir=/tmp/zookeeper01/data # 单机伪集群，目录要不一致
dataLogDir=/tmp/zookeeper01/log # 单机伪集群，目录要不一致
clientPort=2181 # 单机伪集群，端口号要不一致
# 集群中的follower服务器(F)与leader服务器(L)之间完成初始化同步连接时能容忍的最多心跳数
# (tickTime的数量)。
# 如果zk集群环境数量确实很大，同步数据的时间会变长，因此这种情况下可以适当调大该参数。
initLimit=5
# 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
# 集群节点信息配置
# 格式: server.id=host:port:port
# id, 通过在各自的dataDir目录下创建一个名为myid的文件来为每台机器赋予一个服务器id。
# 两个端口号，第一个跟随者用来连接到领导者，第二个用来选举领导者。
server.1=127.0.0.1:2881:3881 # 单机伪集群，两个端口号要不一致
server.2=127.0.0.1:2882:3882
server.3=127.0.0.1:2883:3883
```

分别在三个节点的dataDir目录下创建myid文件，内容为对应的id编号

分别启动三个节点服务

测试

连接任意节点

```
# 创建一个新节点
bin/zkCli.sh -server 127.0.0.1:2181
[zk: localhost:2181(CONNECTED) 0] create /dongnao good
Created /dongnao
```

切换到另一节点

```
# 查询2181节点创建的内容
bin/zkCli.sh -server 127.0.0.1:2182
[zk: localhost:2182(CONNECTED) 0] get /dongnao
good
cZxid = 0x4
ctime = Wed Aug 07 16:51:39 CST 2019
mZxid = 0x4
mtime = Wed Aug 07 16:51:39 CST 2019
pZxid = 0x4
cversion = 0
dataVersion = 0
aclVersion = 0
ephemeralOwner = 0x0
dataLength = 4
numChildren = 0
```

集群间数据同步了，集群搭建完成

注意点

正式环境中，Zookeeper集群的节点部署在独立的服务器上，启动集群的时候，经常碰到集群间的Zookeeper节点连接不上的问题。这个问题总得来说，是因为节点间的端口连接不上，教大家一个排查法。

- Zookeeper服务未正常启动

因为每个Zookeeper节点都部署在不同的独立环境中，可能因为**缺少依赖**，**端口号被占用**等原因导致服务启动失败，可以通过**日志文件错误信息**，**使用ps命令检查进程**，两个方式来进行排查。

- 对应的端口号未开放

每台服务器有**三个端口号**，用telnet命令检测对应的端口是否开放；某些操作系统**可能装了两个防火墙**，需要检查一下；

我们学习Zookeeper，通过搭建伪集群来加快我们的学习步伐。