

centos7下zookeeper集群安装部署

centos7下zookeeper集群 安装部署

应用场景：ZooKeeper是一个分布式的，开放源码的分布式应用程序协调服务，是Google的Chubby一个开源的实现，是Hadoop和Hbase的重要组件。

它是一个为分布式应用提供一致性服务的软件，提供的功能包括：配置维护、域名服务、分布式同步、组服务等。[摘自百度百科]

官网：<http://zookeeper.apache.org/>

安装环境：Zookeeper集群最好至少安装3个节点，这里端口采用默认zookeeper默认端口：2181

Host	IP	Software	Port	Usage	OS
server-1	192.168.1.21	zookeeper 3.4.11	2181	node-1	CentOS 7.3
server-2	192.168.1.22	zookeeper 3.4.11	2181	node-2	CentOS 7.3
server-3	192.168.1.23	zookeeper 3.4.11	2181	node-3	CentOS 7.3

安装步骤：

1. 下载zookeeper并解压

官网推荐下载地址：<https://mirrors.tuna.tsinghua.edu.cn/apache/zookeeper/>

```
~]# tar -zxvf zookeeper-3.4.11.tar.gz -C /opt/ //这里解压至/opt目录中使用
```

```
bin          //启动运行文件
build.xml
conf         //配置文件
contrib
dist-maven
docs
ivysettings.xml
ivy.xml
```

```
lib
LICENSE.txt
NOTICE.txt
README.md
README_packaging.txt
recipes
src
zookeeper-3.4.11.jar
zookeeper-3.4.11.jar.asc
zookeeper-3.4.11.jar.md5
zookeeper-3.4.11.jar.sha1
```

解压后目录

2. 编辑配置文件

进入conf目录，cp生成一个zk能识别的配置文件名：zoo.cfg，如下

```
[root@server-1 conf]# cp zoo_sample.cfg zoo.cfg
```

```
[root@server-1 conf]# cp zoo_sample.cfg zoo.cfg
[root@server-1 conf]# ll
total 16
-rw-r--r-- 1 root root 535 Nov 2 02:47 configuration.xml
-rw-r--r-- 1 root root 2161 Nov 2 02:47 log4j.properties
-rw-r--r-- 1 root root 922 Jan 13 22:27 zoo.cfg
-rw-r--r-- 1 root root 922 Nov 2 02:47 zoo_sample.cfg
```

修改配置文件zoo.cfg，如下：

```
[root@server-1 conf]# vim zoo.cfg
```

```
tickTime=2000
# The number of ticks that the initial
# synchronization phase can take
initLimit=10
# The number of ticks that can pass between
# sending a request and getting an acknowledgement
syncLimit=5
# the directory where the snapshot is stored.
# do not use /tmp for storage, /tmp here is just
# example sake.
dataDir=/data/zookeeper //这里修改了路径
# the port at which the clients will connect
clientPort=2181 //这里默认端口不变
# the maximum number of client connections.
# increase this if you need to handle more clients
#maxClientCnxns=60
#
# Be sure to read the maintenance section of the
# administrator guide before turning on autopurge.
#
# http://zookeeper.apache.org/doc/current/zookeeperAdmin.html#sc_m
#
# The number of snapshots to retain in dataDir
#autopurge.snapRetainCount=3
# Purge task interval in hours
# Set to "0" to disable auto purge feature
#autopurge.purgeInterval=1

server.1=192.168.1.21:2888:3888
server.2=192.168.1.22:2888:3888 //新增这三行，三台节点新增的内容都一样
server.3=192.168.1.23:2888:3888
```

保存，退出；

三台节点配置一样，其他两台略。

3. 添加myid文件

说明：除了修改 zoo.cfg 配置文件，集群模式下还要**新增**一个名叫**myid**的文件，这个文件放在上述**dataDir**指定的**目录下**，这个文件里面就

只有一个数据，就是上图配置中server.x的这个x（1,2,3）值，zookeeper启动时会读取这个文件，拿到里面的数据与 zoo.cfg 里面

的配置信息比较从而判断到底是那个server（节点）。

```
[root@server-1 ~]# cat /data/zookeeper/myid
1 // node-1的myid值就是 1
```

```
[root@server-2 ~]# cat /data/zookeeper/myid
2 //node-2的myid内容就是 2
```

```
[root@server-3 ~]# cat /data/zookeeper/myid
3 //node-3的myid内容就是 3
```

4. 启动节点

切换到bin目录中查看一下：

```
[root@server-1 bin]# ls -l
total 36
-rwxr-xr-x 1 root root 232 Nov  2 02:47 README.txt
-rwxr-xr-x 1 root root 1937 Nov  2 02:47 zkCleanup.sh
-rwxr-xr-x 1 root root 1056 Nov  2 02:47 zkCli.cmd
-rwxr-xr-x 1 root root 1534 Nov  2 02:47 zkCli.sh //客户端连接脚本
-rwxr-xr-x 1 root root 1628 Nov  2 02:47 zkEnv.cmd
-rwxr-xr-x 1 root root 2696 Nov  2 02:47 zkEnv.sh
-rwxr-xr-x 1 root root 1089 Nov  2 02:47 zkServer.cmd
-rwxr-xr-x 1 root root 6773 Nov  2 02:47 zkServer.sh //服务启动查看脚本
```

启动方法查看：

```
[root@server-1 bin]# ./zkServer.sh //直接回车可以查看用法
ZooKeeper JMX enabled by default
Using config: /opt/zookeeper-3.4.11/bin/../conf/zoo.cfg
Usage: ./zkServer.sh {start|start-foreground|stop|restart|status|up}
```

启动节点1：

```
[root@server-1 bin]# ./zkServer.sh start
```

// 默认会到同级目录conf中寻找zoo.cfg文件，所以默认不用加配置文件

```
[root@server-1 bin]# ./zkServer.sh start
ZooKeeper JMX enabled by default
Using config: /opt/zookeeper-3.4.11/bin/../conf/zoo.cfg
Starting zookeeper ... STARTED //显示STARTED不代表成功，一定要确认端口起来
[root@server-1 bin]# ss -tnl
State      Recv-Q Send-Q           Local Address:Port
LISTEN     0      128          *:22
LISTEN     0      100      127.0.0.1:25
LISTEN     0      50          :::45226
LISTEN     0      50      ::ffff:192.168.1.21:3888
LISTEN     0      128          :::22
LISTEN     0      100          :::1:25
LISTEN     0      50          :::2181
```

同样的，启动节点2，节点3：

```
[root@server-2 bin]# ./zkServer.sh start
```

```
[root@server-3 bin]# ./zkServer.sh start
```

启动后，会在当前bin目录下生成一个日志zookeeper.out，如果启动不成功里面会有错，可供排错使用。

5. 查看节点状态

当三台节点都启动完毕后可以查看他们各自在集群中的状态：

```
[root@server-1 bin]# ./zkServer.sh status
```

```
[root@server-1 bin]# ./zkServer.sh status
ZooKeeper JMX enabled by default
Using config: /opt/zookeeper-3.4.11/bin/../conf/zoo.cfg
Mode: follower //该节点为从节点
```

```
[root@server-2 bin]# ./zkServer.sh status
ZooKeeper JMX enabled by default
Using config: /opt/zookeeper-3.4.11/bin/../conf/zoo.cfg
Mode: leader //节点2被选举为主节点
```

```
[root@server-3 bin]# ./zkServer.sh status
ZooKeeper JMX enabled by default
Using config: /opt/zookeeper-3.4.11/bin/../conf/zoo.cfg
Mode: follower //节点3为从节点
```

至此，zookeeper集群部署成功。

排错：如果查看集群状态的时候报“*Error contacting service. It is probably not running.*”请检查防火墙是不是阻挡了contacting。

```
[root@server-1 bin]# ./zkServer.sh status
ZooKeeper JMX enabled by default //如果报错如下，请检查Selinux状态和防火墙状态
Using config: /opt/zookeeper-3.4.11/bin/../conf/zoo.cfg
Error contacting service. It is probably not running.
```

6. zookeeper客户端使用

使用bin目录下的客户端登录脚本：zkCli.sh

```
ZooKeeper -server host:port cmd args
stat path [watch]
set path data [version]
ls path [watch]
delquota [-n|-b] path
ls2 path [watch]
//一些用法
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

```

比方说：登录、查看、退出。

```
[root@server-1 bin]# ./zkCli.sh -server 192.168.1.21:2181 //连本地节点
```

```
[root@server-1 bin]# ./zkCli.sh -server 192.168.1.21 //不带端口就采用默认2181端口
```

```
[root@server-1 bin]# ./zkCli.sh //不带参数回车默认连接本地IP和2181端口
```

```
[root@server-1 bin]# ./zkCli.sh -server 192.168.1.22 //也可以连接节点2的zk
```

登录后操作：

```

[zk: 192.168.1.21(CONNECTED) 0] ls /
[zookeeper]
[zk: 192.168.1.21(CONNECTED) 1] ls /zookeeper
[quota]
[zk: 192.168.1.21(CONNECTED) 2] get /zookeeper
cZxid = 0x0
ctime = Thu Jan 01 08:00:00 CST 1970
mZxid = 0x0
mtime = Thu Jan 01 08:00:00 CST 1970
pZxid = 0x0
cversion = -1
dataVersion = 0
aclVersion = 0
ephemeralOwner = 0x0
dataLength = 0
numChildren = 1
[zk: 192.168.1.21(CONNECTED) 3] quit
Quitting...
2018-01-14 00:56:25,737 [myid:] - INFO [main:ZooKeeper@687] - Session 100009709f20007
2018-01-14 00:56:25,742 [myid:] - INFO [main-EventThread:ClientCn
[root@server-1 bin]#

```

说明：客户端连接后，用get / 命令可以发现此时只有zookeeper一项；如果此Zookeeper用于对Kafka或JStorm等提供服务，

则还会有相应的其他目录，后面在介绍kafka时会有相关内容。

kafka集群安装：<http://www.cnblogs.com/ding2016/p/8282907.html>

7. 集群测试

这里就模拟断掉"leader"节点——node-2，另外两台会通过之前设定的3888这个端口重新选举，结果如下：

米重新选举leader，结果如下：

```
[root@server-1 bin]# ./zkServer.sh status
ZooKeeper JMX enabled by default
Using config: /opt/zookeeper-3.4.11/bin/../conf/zoo.cfg
Mode: follower //node-1仍然为 从
```

```
[root@server-3 bin]# ./zkServer.sh status
ZooKeeper JMX enabled by default
Using config: /opt/zookeeper-3.4.11/bin/../conf/zoo.cfg
Mode: leader // node-3已经被选举为 主
```

结束.
