

Zookeeper集群无状态平滑升级、迁移

1、前言

前面我们学习了Zookeeper的集群搭建和管理，也学习和探讨了它的选举原理，我们这一章节来学习zookeeper集群的升级和迁移

在学习集群的升级和迁移之前，我们先要知道为什么会有这两个操作？

我们的zookeeper集群可能因为某一些原因，比如：

当前版本功能不够强大或者不够稳定，升级功能更加强大或者稳定的版本；

当前集群容量太小，需要增加服务节点，升级负载均衡能力；

当前集群服务器配置太低，需要迁移到配置更高的服务器上；

老板发神经等等.....

知道为什么要升级迁移后，再来分析下升级迁移的概念和分类

升级：主要包括两种情况--->

- 1)、版本升级：zookeeper集群节点数量和拓扑结构不变，只改变zookeeper的版本
- 2)、容量升级：一般指的是集群增加服务节点，提高负载均衡性能

迁移：一般来说指的是将服务从集群A迁移到集群B，这里也主要包括两种情况---->

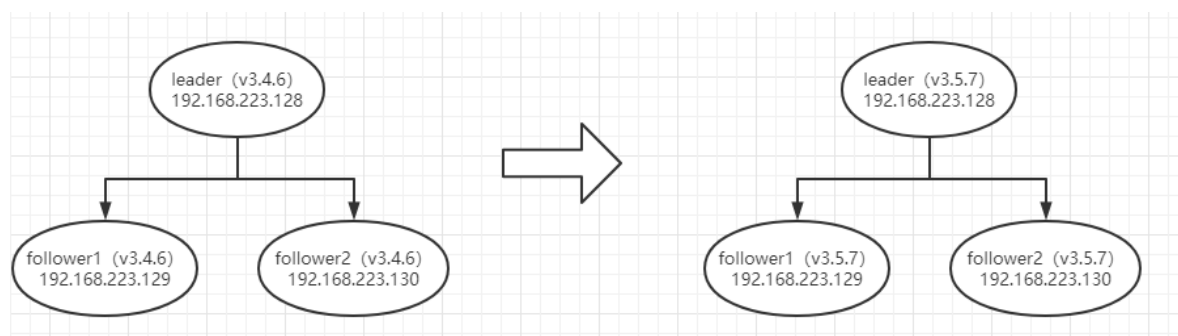
- 1)、部分迁移，比如集群1 (A,B,C) 迁移到集群2 (A,D,E)
- 2)、整体迁移，比如集群1 (A,B,C) 迁移到集群2 (C,D,E)

但是：不管是升级迁移，也不管是哪种升级，哪种迁移，我们都要尽可能的保证一点：在线的服务尽可能不断开，必须持续提供服务，使用户无感知或者说极小可忽略的停顿，并且最重要的是数据不丢失！这也就是所谓的平滑升级迁移！

2、升级迁移拓扑图

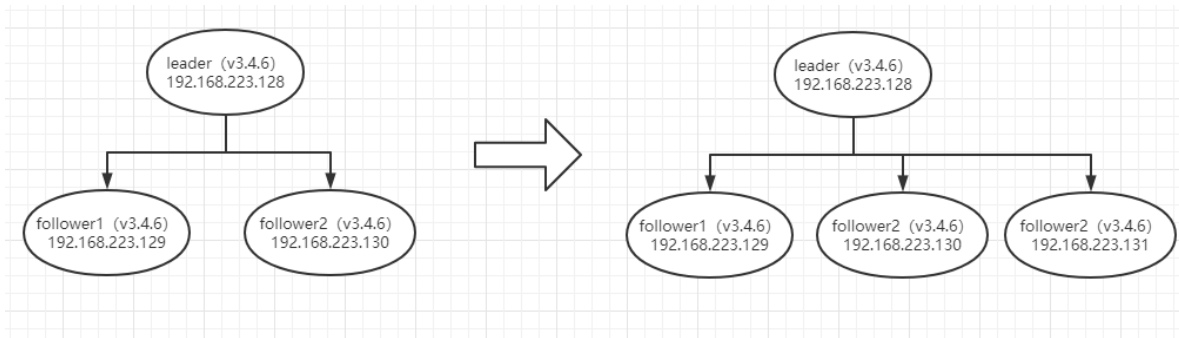
2.1 zookeeper集群升级

2.1.1 版本升级



服务器和节点数量不变，主从节点拓扑结构不变，只改变版本

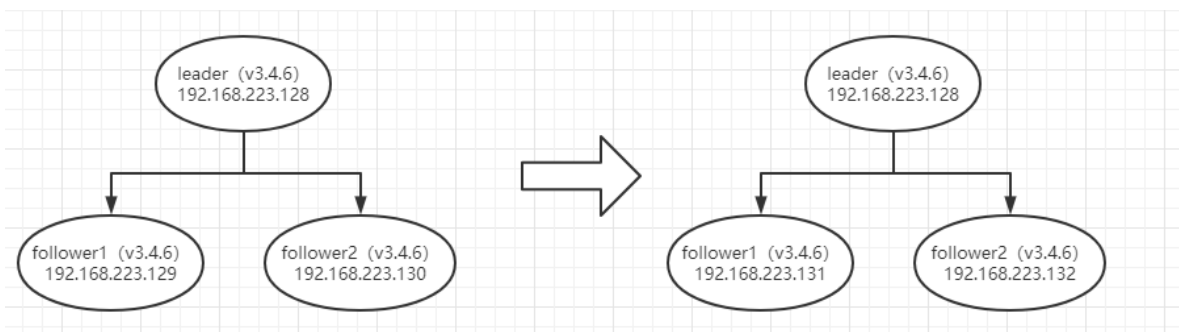
2.1.2 容量升级



主从节点拓扑结构不变，版本不变，只是增加从节点，提高负载均衡能力

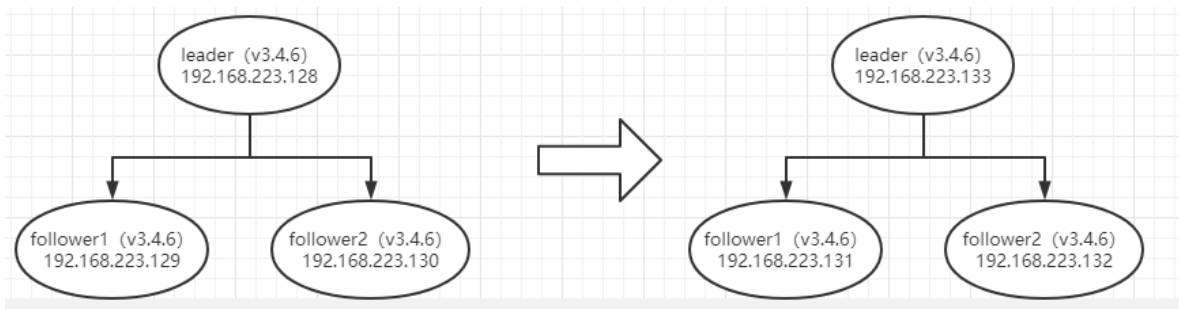
2.2 zookeeper集群迁移

2.2.1 局部迁移



一般来说部分迁移主节点尽可能不变，只替换从节点

2.2.2 整体迁移



整个节点全部替换！

3、zookeeper应用

3.1 客户端开启DNS域名解析

PS：因为我们后面都是通过虚拟机hostname来进行访问，方便升级迁移无缝对接，所以设置本地DNS域名解析Hosts：C:\Windows\System32\drivers\etc\hosts（如果是Linux虚拟机充当客户端也是一样的修改etc下hosts和hostname文件）

```
192.168.223.128 ydt1
192.168.223.129 ydt2
192.168.223.130 ydt3
192.168.223.131 ydt4
192.168.223.132 ydt5
192.168.223.133 ydt6
```

打开CMD窗口可以通过命令: telnet ydt1 2181 连接通, 标识配置成功!

3.2 dubbo-admin管理控制台

下载一个dubbo-admin管理控制台, 这里主要是用来监控zookeeper服务是否可用!

下载地址: <https://github.com/dangdangdotcom/dubbox/tree/master/dubbo-admin>, 本课程已经为大家下载了, 直接使用即可

将下载的dubbo-admin-*项目复制到tomcat目录下的webapps目录下



进入该项目下的WEB-INF目录下, 修改dubbo.properties文件如下:

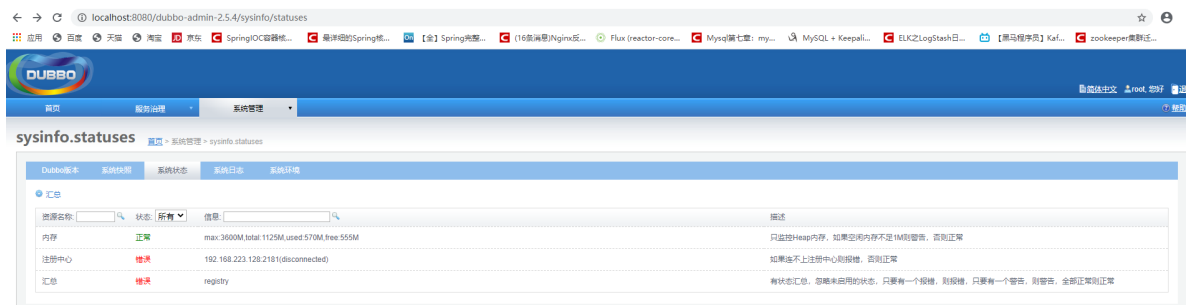
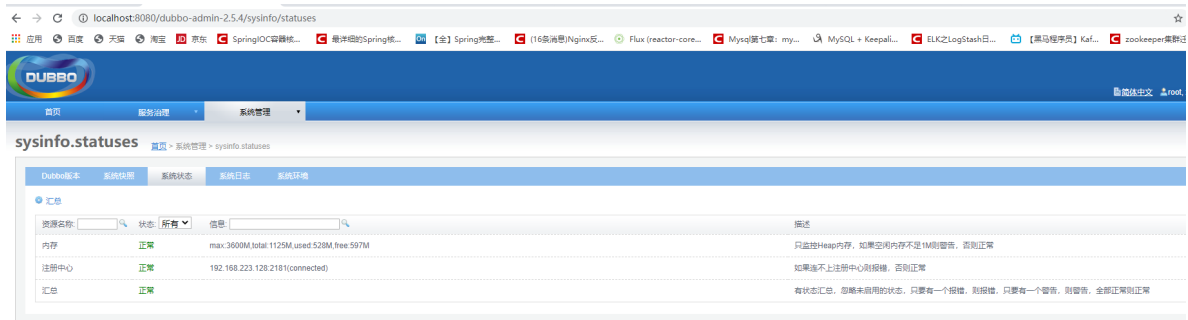
```
#zookeeper集群地址, 逗号分隔, 129位leader, 128,130备份节点
dubbo.registry.address=zookeeper://ydt1:2181?backup=ydt2:2181,ydt3:2181
#root账号密码都是root
dubbo.admin.root.password=root
#guest账号密码都是guest
dubbo.admin.guest.password=guest
```

3.3 启动dubbo-admin

- 1)、启动之前配置的128,129,130三台zookeeper集群, 不清楚的同学查看第一章《Zookeeper集群及配置》
- 2)、正常启动tomcat, 进入bin目录直接双击startup.bat, 如果不是管理账号右键以管理员身份运行!

```
Tomcat
Creating shared instance of singleton bean 'mvcUriComponentsContributor'
ControllerAdvice beans: 0 @ModelAttribute, 0 @InitBinder, 1 RequestBodyAdvice, 1 ResponseBodyAdvice
Creating shared instance of singleton bean 'org.springframework.web.servlet.mvc.method.annotation.ExceptionHandlerExceptionHandlerResolver#0'
ControllerAdvice beans: 0 @ExceptionHandler, 1 ResponseBodyAdvice
Creating shared instance of singleton bean 'org.springframework.web.servlet.mvc.annotation.ResponseStatusExceptionHandlerResolver#0'
Creating shared instance of singleton bean 'org.springframework.web.servlet.mvc.support.DefaultHandlerExceptionHandlerResolver#0'
Creating shared instance of singleton bean 'org.springframework.web.servlet.handler.BeanNameUrlHandlerMapping'
Creating shared instance of singleton bean 'org.springframework.web.servlet.mvc.HttpRequestHandlerAdapter'
Creating shared instance of singleton bean 'org.springframework.web.servlet.mvc.SimpleControllerHandlerAdapter'
Looking up JNDI object with name [java:comp/env/spring.liveBeansView.mbeanDomain]
Converted JNDI name [java:comp/env/spring.liveBeansView.mbeanDomain] not found - trying original name [spring.liveBeansView.mbeanDomain]. javax.naming.NameNotFoundException: 名称[spring.liveBeansView.mbeanDomain]未在此上下文中绑定。找不到[spring.liveBeansView.mbeanDomain]。
Looking up JNDI object with name [spring.liveBeansView.mbeanDomain]
JNDI lookup for name [spring.liveBeansView.mbeanDomain] threw NamingException with message: 名称[spring.liveBeansView.mbeanDomain]未在此上下文中绑定。找不到[spring.liveBeansView.mbeanDomain]。 Returning null.
八月 03, 2020 5:47:56 下午 org.springframework.web.servlet.FrameworkServlet initServletBean
信息: Completed initialization in 1010 ms
八月 03, 2020 5:47:56 下午 org.apache.catalina.startup.HostConfig deployDirectory
信息: Web应用程序目录[D:\tools\apache-tomcat-7.0.104\webapps\ROOT]的部署已在[2,274]毫秒内完成
八月 03, 2020 5:47:56 下午 org.apache.coyote.AbstractProtocol start
信息: 开始协议处理句柄["http-apr-8080"]
八月 03, 2020 5:47:56 下午 org.apache.coyote.AbstractProtocol start
信息: 开始协议处理句柄["http-nio-8443"]
八月 03, 2020 5:47:56 下午 org.apache.catalina.startup.Catalina start
信息: Server startup in 32539 ms
```

访问dubbo-admin管理控制台：<http://localhost:8080/dubbo-admin-2.5.4>（注意项目名，建议改成dubbo-admin,省的输版本）



当集群可用和不可用的时候，我们可以看得到控制台有不同的输出，我们后面就用这些信息来作为服务可用不可用的标识

```
INFO zkclient.ZkClient - zookeeper state changed (Disconnected)
INFO zkclient.ZkClient - zookeeper state changed (SyncConnected)
```

4、升级部署

4.1 版本升级

4.1.1 环境准备

在128,129,130 上下载准备升级用的zookeeper版本

```
#旧版本为3.4.6，我们升级为3.4.14，三台机器分别执行如下命令：
wget https://archive.apache.org/dist/zookeeper/zookeeper-3.4.14/zookeeper-3.4.14.tar.gz
tar -zxvf zookeeper-3.4.14.tar.gz #解压
rm zookeeper-3.4.14.tar.gz -f #解压了就删掉吧，多浪费空间啊
```

4.1.2 修改配置测试

配置内容跟之前zookeeper集群一样，不过我们利用的是zookeeper选举同步这个特点来进行逐步替换，并且为了保证集群服务的最小影响，我们需要从从节点开始替换！-----当前129是leader

```
[root@ydt2 zookeeper-3.4.6]# ./bin/zkServer.sh status
JMX enabled by default
Using config: /usr/local/zookeeper-3.4.6/bin/../conf/zoo.cfg
Mode: leader
```

dubbo-admin也连接正常：

```
earDomain]未在此上下文中绑定。找不到[spring.liveBeansView.mbeanDomain]。 Returning null.
八月 03, 2020 10:24:00 下午 org.springframework.web.servlet.FrameworkServlet initServletBean
信息: Completed initialization in 1004 ms
八月 03, 2020 10:24:00 下午 org.apache.catalina.startup.HostConfig deployDirectory
信息: Web应用程序目录[D:\tools\apache-tomcat-7.0.104\webapps\ROOT]的部署已在[3, 541]毫秒内完成
八月 03, 2020 10:24:00 下午 org.apache.coyote.AbstractProtocol start
信息: 开始协议处理句柄["http-apr-8080"]
八月 03, 2020 10:24:00 下午 org.apache.coyote.AbstractProtocol start
信息: 开始协议处理句柄["http-nio-8443"]
八月 03, 2020 10:24:00 下午 org.apache.catalina.startup.Catalina start
信息: Server startup in 35441 ms
```

1) 先将128节点旧版本的zookeeper服务，dubbo-admin连接正常（节点数少于一半挂掉）：

```
八月 03, 2020 10:27:36 下午 org.springframework.web.servlet.FrameworkServlet initServletBean
信息: Completed initialization in 1006 ms
八月 03, 2020 10:27:36 下午 org.apache.catalina.startup.HostConfig deployDirectory
信息: Web应用程序目录[D:\tools\apache-tomcat-7.0.104\webapps\ROOT]的部署已在[2, 305]毫秒内完成
八月 03, 2020 10:27:36 下午 org.apache.coyote.AbstractProtocol start
信息: 开始协议处理句柄["http-apr-8080"]
八月 03, 2020 10:27:36 下午 org.apache.coyote.AbstractProtocol start
信息: 开始协议处理句柄["http-nio-8443"]
八月 03, 2020 10:27:36 下午 org.apache.catalina.startup.Catalina start
信息: Server startup in 36708 ms
```

然后启动新版本zookeeper服务，达到替换原先从节点，同时将数据同步过来的目的，新版本zookeeper服务的配置与之前旧版本一样：

```
cd zookeeper-3.4.14/
mv conf/zoo_sample.cfg conf/zoo.cfg

#在zookeeper-3.4.6目录下分别创建一个data和logs目录
mkdir data
mkdir logs

#修改zoo.cfg文件配置
vim conf/zoo.cfg
-----#配置如下#-----
-----

#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。
tickTime=2000
#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=10
#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（tick#Time的数量）。
syncLimit=5
#数据持久化目录
dataDir=/usr/local/zookeeper-3.4.14/data #数据持久化目录
#日志目录
dataLogDir=/usr/local/zookeeper-3.4.14/logs
#客户端连接 Zookeeper 服务器的端口 用默认就行
clientPort=2181

server.1=ydt1:4000:5000
server.2=ydt2:4000:5000
server.3=ydt3:4000:5000

#创建多个节点集群时，在dataDir目录下必须创建myid文件，myid文件用于zookeeper验证server序号#等，myid文件只有一行，并且为当前server的序号，例如server.1的myid就是1，server.2的myid就是2等。

#server.A=B:C:D;其中 A 是一个数字，表示这个第几号服务器；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，由于 #B 都是一样，所以不同的 Zookeeper 实例通信端口号不能一样，所以要给它们分配不同的端口号
```

启动后你会发现通过客户端发现，它拥有了跟129,130节点同样的数据，说明数据热同步成功！

同时我们的dubbo-admin控制显示的还是正常状态!

同时129节点还是从节点状态：

```
[root@ydt1 zookeeper-3.4.14]# ./bin/zkServer.sh status
ZooKeeper JMX enabled by default
Using config: /usr/local/zookeeper-3.4.14/bin/../conf/zoo.cfg
Mode: follower
```

![image-20200803220156999](C:\Users\18588\AppData\Roaming\Typora\typora-user-images\image-20200803220156999.png)

2)、现在我们把130的旧版本的zookeeper服务关掉，服务依然可用，再一次说明129的新版本替换成功！

```
beanDomain)未在此上下文中绑定。找不到[org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerMethod]。 Returning null.
八月 03, 2020 10:27:36 下午 org.springframework.web.servlet.FrameworkServlet initServletBean
信息: Completed initialization in 1006 ms
八月 03, 2020 10:27:36 下午 org.apache.catalina.startup.HostConfig deployDirectory
信息: Web应用程序目录[D:\tools\apache-tomcat-7.0.104\webapps\ROOT]的部署已在[2,305]毫秒内完成
八月 03, 2020 10:27:36 下午 org.apache.coyote.AbstractProtocol start
信息: 开始协议处理句柄["http-apr-8080"]
八月 03, 2020 10:27:36 下午 org.apache.coyote.AbstractProtocol start
信息: 开始协议处理句柄["http-nio-8443"]
八月 03, 2020 10:27:36 下午 org.apache.catalina.startup.Catalina start
信息: Server startup in 36708 ms
```

130机器上如法炮制（这里就不展开说了，相信自己，你最美丽，老胡给你加油！）

最后轮到129主节点了，配置跟从节点一样，先将旧版本的服务关掉，dubbo-admin会先断开，然后马上连上了：

```
INFO zkclient.ZkClient - zookeeper state changed (Disconnected)
INFO zkclient.ZkClient - zookeeper state changed (SyncConnected)
```

选举的时候是不能提供服务的！

最后再启动新版本的zookeeper服务，当然，现在的leader肯定已经换人了：

```
Using config: /usr/local/zookeeper-3.4.14/bin/../conf/zoo.cfg
Starting zookeeper ... STARTED
[root@ydt2 zookeeper-3.4.14]# ./bin/zkServer.sh status
ZooKeeper JMX enabled by default
Using config: /usr/local/zookeeper-3.4.14/bin/../conf/zoo.cfg
Mode: follower
```

总结：如果忽略选举时间，基本上已经完全实现了热替换，数据也能够同步过来！

4.2 容量升级

容量升级不能做到完全的热部署，因为新增节点必须在客户端配置！

进入该项目下的WEB-INF目录下，修改dubbo.properties文件如下：

```
#zookeeper集群地址，逗号分隔，129位leader，128,130备份节点
dubbo.registry.address=zookeeper://ydt1:2181?backup=ydt2:2181,ydt3:2181
#root账号密码都是root
dubbo.admin.root.password=root
#guest账号密码都是guest
dubbo.admin.guest.password=guest
```

4.2.1 启动原始集群

启动128,129,130三个节点的集群，当前leader为129，follower为128,130

4.2.2 新增节点并配置

新增一个节点131（将之前三台虚拟机整体复制过来），配置zoo.cfg，配置比已经启动的原始集群中节点配置多一个本身的配置：

```
server.1=ydt1:4000:5000
server.2=ydt2:4000:5000
server.3=ydt3:4000:5000
#新增节点配置,ydt4为131节点域名
server.4=ydt4:4000:5000
```

```
-----
vim data/myid
#修改myid为4
```

4.2.3 启动新节点

加入集群并称为从节点

```
[root@ydt4 zookeeper-3.4.6]# ./bin/zkServer.sh start
JMX enabled by default
Using config: /usr/local/zookeeper-3.4.6/bin/../conf/zoo.cfg
Starting zookeeper ... STARTED
[root@ydt4 zookeeper-3.4.6]# ./bin/zkServer.sh status
JMX enabled by default
Using config: /usr/local/zookeeper-3.4.6/bin/../conf/zoo.cfg
Mode: follower
```

数据也同步过来了：

```
zk: localhost:2181(CONNECTED) 0 ls /
test200000000014, zk-getChildren-sync-test, curator, zookeeper, test100000000009, zk-getChildren-async-test, test100000000007, create, dubbo, test11
```

4.2.4 原始集群处理

原始集群中三个节点zoo.cfg配置中一次增加：server.4=192.168.223.131:4000:5000 并重启，为了尽可能的少选举，当然先从从节点开始处理

5、迁移部署

5.1 局部迁移

局部迁移跟容量升级有点类似，只不过替换节点不需要配置被替换节点的集群配置

5.1.1 关闭被替换节点

关闭需要替换的130节点，服务可以正常连接！

```
INFO rule.FallbackModuleMappingRule - Initialized control:FallbackModuleMappingRule with cache en
INFO rule.DirectTemplateMappingRule - Initialized control.template:DirectTemplateMappingRule with
INFO zkclient.ZkEventThread - Starting ZkClient event thread.
INFO zkclient.ZkClient - zookeeper state changed (SyncConnected)
INFO zkclient.ZkEventThread - Starting ZkClient event thread.
INFO zkclient.ZkClient - zookeeper state changed (Disconnected)
INFO zkclient.ZkClient - zookeeper state changed (SyncConnected)
INFO zkclient.ZkClient - zookeeper state changed (Disconnected)
INFO zkclient.ZkClient - zookeeper state changed (SyncConnected)
INFO zkclient.ZkClient - zookeeper state changed (Disconnected)
INFO zkclient.ZkClient - zookeeper state changed (SyncConnected)
INFO zkclient.ZkClient - zookeeper state changed (Disconnected)
INFO zkclient.ZkClient - zookeeper state changed (SyncConnected)
INFO zkclient.ZkClient - zookeeper state changed (Disconnected)
INFO zkclient.ZkClient - zookeeper state changed (SyncConnected)
INFO zkclient.ZkClient - zookeeper state changed (Disconnected)
INFO zkclient.ZkClient - zookeeper state changed (SyncConnected)
INFO zkclient.ZkClient - zookeeper state changed (Disconnected)
INFO zkclient.ZkClient - zookeeper state changed (SyncConnected)
```

5.1.2 开启替换节点


```
192.168.223.128 ydt1
192.168.223.129 ydt2
192.168.223.131 ydt3 #该节点为域名解析替换，节点之间互通使用的为hostname域名
```

```
server.1=ydt1:4000:5000
server.2=ydt2:4000:5000
server.4=ydt3:4000:5000
#这里我准备替换130，所以就不要去配置该节点了，当然，你也可以最后去掉
```

[illegible]

```
vim etc/hosts
----->
192.168.223.131 ydt1 #该节点为域名解析替换，域名为被替换机器域名，IP为本机131 IP
192.168.223.129 ydt2
192.168.223.130 ydt3

vim etc/hostname
----->
ydt1 #改为需要被替换机器的域名
```

第二：需要修改zoo.cfg的集群配置跟128保持一致，然后启动zookeeper服务

```
server.4=ydt1:4000:5000 #这个地方现在是131了，注意myid要设置为4，比之前的要大
server.2=ydt2:4000:5000
server.3=ydt3:4000:5000
```

#如果使用过该节点，把这些陈年老酿删掉，保持纯洁

```
[root@ydt1 zookeeper-3.4.6]# rm data/version-2/ -rf
[root@ydt1 zookeeper-3.4.6]# rm data/zookeeper_server.pid -f
[root@ydt1 zookeeper-3.4.6]# rm logs/version-2/ -rf
```

```
[root@ydt1 zookeeper-3.4.6]# ./bin/zkServer.sh start
JMX enabled by default
Using config: /usr/local/zookeeper-3.4.6/bin/../conf/zoo.cfg
Starting zookeeper ... STARTED
[root@ydt1 zookeeper-3.4.6]# ./bin/zkServer.sh status
JMX enabled by default
Using config: /usr/local/zookeeper-3.4.6/bin/../conf/zoo.cfg
Mode: follower
```

5.2.2 迁移node2(129->132)

1)、修改130的DNS域名解析，重启网络服务，修改130节点zoo.cfg配置重启；

#修改130节点域名解析：vim/etc/hosts

```
192.168.223.131 ydt1
192.168.223.129 ydt2
192.168.223.130 ydt3
```

#重启130网络服务

```
service network restart
```

#修改130集群配置，然后重启

```
server.4=ydt1:4000:5000 #这个地方现在是131了，否则129关掉后，集群超过半数挂掉，game over
server.2=ydt2:4000:5000
server.3=ydt3:4000:5000
```

关闭129节点zookeeper服务，保证剩下的130,131（新）能持续提供服务,能看到服务仍然能够使用：

The screenshot shows the Dubbo Admin console interface. The top navigation bar includes '首页' (Home), '服务治理' (Service Governance), and '系统管理' (System Management). The main content area is titled 'sysinfo.statuses' and shows a summary of system information. The summary table includes the following data:

资源名称	状态	信息
内存	正常	max:3600M,total:1065M,used:348M,free:717M
注册中心	正常	ydt1:2181(connection)
汇总	正常	

2)、打开132虚拟机，替换掉关闭的129节点

```
#132节点需要的配置跟131类似，域名改为原先129节点域名：ydt2，域名解析如下：
192.168.223.131 ydt1
192.168.223.132 ydt2
192.168.223.130 ydt3

#zoo.cfg集群配置
server.4=ydt1:4000:5000
server.5=ydt2:4000:5000 #这个地方现在是132了，注意myid要设置为5，比之前的要大
server.3=ydt3:4000:5000

#重启虚拟机，让hostname: ydt2生效

#如果使用过该节点，把这些陈年老酿删掉，保持纯洁
[root@ydt1 zookeeper-3.4.6]# rm data/version-2/ -rf
[root@ydt1 zookeeper-3.4.6]# rm data/zookeeper_server.pid -f
[root@ydt1 zookeeper-3.4.6]# rm logs/version-2/ -rf

#启动zookeeper服务
```

```
[root@ydt2 zookeeper-3.4.6]# ./bin/zkServer.sh start
JMX enabled by default
Using config: /usr/local/zookeeper-3.4.6/bin/../conf/zoo.cfg
Starting zookeeper ... STARTED
[root@ydt2 zookeeper-3.4.6]# ./bin/zkServer.sh status
JMX enabled by default
Using config: /usr/local/zookeeper-3.4.6/bin/../conf/zoo.cfg
Mode: follower
```

5.2.3 迁移node3(130->133)

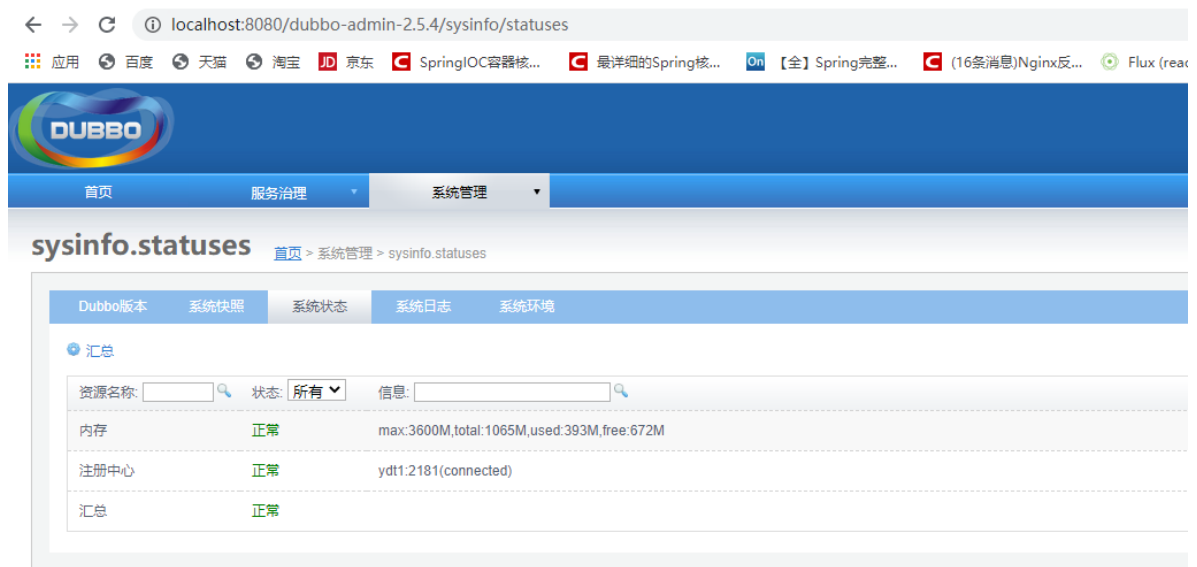
1)、修改131的DNS域名解析，重启网络服务，修改131节点zoo.cfg节点集群配置，重启zookeeper服务

```
#修改131节点域名解析：vim/etc/hosts
192.168.223.131 ydt1
192.168.223.132 ydt2
192.168.223.130 ydt3

#重启131网络服务
service network restart

#修改131节点zoo.cfg节点集群配置，重启zookeeper服务
server.4=ydt1:4000:5000
server.5=ydt2:4000:5000 #这个地方现在是132了，否则130关掉后，集群超过半数挂掉，game over
server.3=ydt3:4000:5000
```

2)、关闭130节点虚拟机，保证剩下的131（新），132（新）能持续提供服务！



3)、修改 131,132机器上的域名映射解析如下，方便数据同步：

```
192.168.223.131 ydt1
192.168.223.132 ydt2
192.168.223.133 ydt3

#重启131、132网络服务
service network restart
```

4)、打开133虚拟机，替换掉关闭的130节点，启动133节点zookeeper服务

```
#133节点需要的配置跟132类似，域名改为原先130节点域名：ydt3，域名解析如下：
192.168.223.131 ydt1
192.168.223.132 ydt2
192.168.223.133 ydt3

#zoo.cfg集群配置
server.4=ydt1:4000:5000
server.5=ydt2:4000:5000
server.6=ydt3:4000:5000 #这个地方现在是133了，注意myid要设置为6，比之前的要大

#如果使用过该节点，把这些陈年老酿删掉，保持纯洁
[root@ydt1 zookeeper-3.4.6]# rm data/version-2/ -rf
[root@ydt1 zookeeper-3.4.6]# rm data/zookeeper_server.pid -f
[root@ydt1 zookeeper-3.4.6]# rm logs/version-2/ -rf

#重启虚拟机，让hostname生效后再启动zookeeper服务
```

```
[root@ydt3 zookeeper-3.4.6]# ./bin/zkServer.sh start
JMX enabled by default
Using config: /usr/local/zookeeper-3.4.6/bin/../conf/zoo.cfg
Starting zookeeper ... ^[[ASTARTED
[root@ydt3 zookeeper-3.4.6]# ./bin/zkServer.sh status
JMX enabled by default
Using config: /usr/local/zookeeper-3.4.6/bin/../conf/zoo.cfg
Mode: follower
```

5.2.4 收官

最后再将131，132的集群配置如下，依次重启，完美收官

```
server.4=ydt1:4000:5000  
server.5=ydt2:4000:5000  
server.6=ydt3:4000:5000
```

6、写到最后

我们在做zookeeper迁移也好，数据库迁移也罢，需要遵循两点：

- 1)、数据完整性是最重要的，如果不能保证，那就不满足迁移的前提！
- 2)、服务持续提供是次重要的，如果不能保证，提前跟客户方说明！
- 3)、因为redis集群选举期间服务不可用，所以迁移时尽可能的先迁移从节点
- 4)、请保证所有要加入的节点都是干净纯洁的，如果不是，请删掉

数据版本数据信息：rm -rf data/version-2

日志版本数据信息：rm -rf logs/version-2

zookeeper服务id文件：rm -f data/zookeeper_server.pid