该文章搭建之前需要具备的软件及工具：

1.一台云服务器[我的是阿里云的centos]

1. 云服务器安装好JDK

[没安装的参考这篇：

<https://blog.csdn.net/weixin_38964895/article/details/81078247>]

1. 云服务器安装好Tomcat

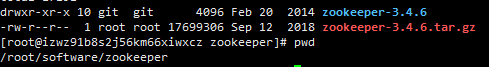
[没安装的参考这篇：

<https://blog.csdn.net/weixin_38964895/article/details/81078276>]

4.zookeeper，dubbo-admin[我用的版本：zookeeper3.4.6，dubbo-admin2.4.1]【文章中贴有下载地址链接】

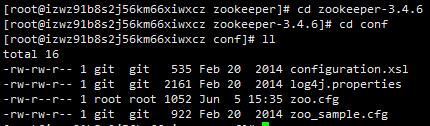
1.搭建zookeeper

下载zookeeper，安装并解压，此次我用的是：3.4.6，路径如下

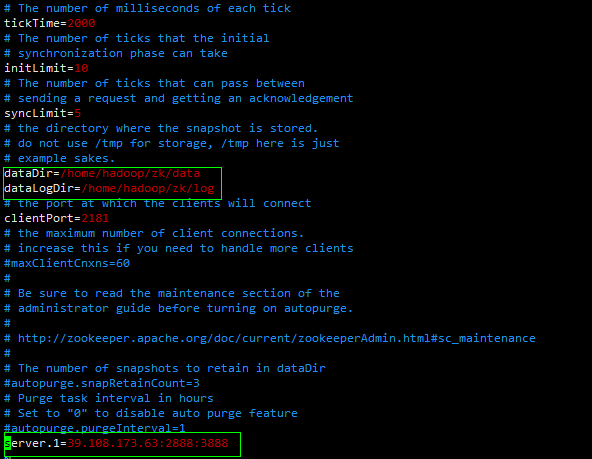


下载地址： <https://download.csdn.net/download/weixin_38964895/11229209>

复制一份 zoo\_sample.cfg  为 zoo.cfg 如下



vim zoo.cfg



默认进来之后，会有

Server.1=Master:3333:4444

Server.2=slave1:3333:4444

Server.3=slave2:3333:4444

注意点1：更改成自己服务器的IP，具体几个server可根据需要添加，由于服务器性能原因，我这里只保留一个，就是单机模式

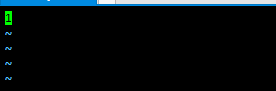
注意点2：上面的dataDir  dataLogDir  先手动去确认一下，没有就手动  mkdir

注意点3：没有myid文件，需要在上图的：dataDir路径下生成一个【vi myid就行】，同时写入上面的server.X中 的 X

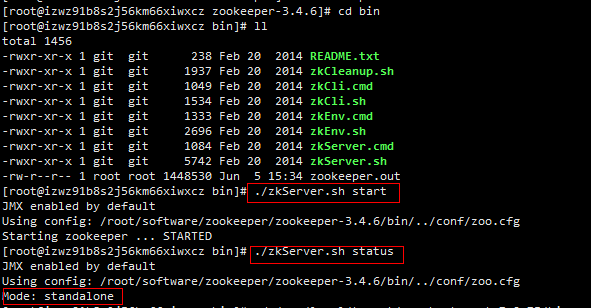
【Server.1=Master:3333:4444   就在myid中写个1

Server.1=Master:3333:4444  Server.2=slave1:3333:4444   就写1  2 】

【集群模式下除了多个zookeeper外，在myid文件中也需要添加server.X中的X】



修改完之后保存退出，进入bin目录下，启动



 ./zkServer.sh start   启动指令

./zkServer.sh status 查看启动状态的指令

2.搭建dubbo-admin

首先下载dubbo-admin的war包，没有的可以在下面的链接进行下载

<https://download.csdn.net/download/weixin_38964895/11227295>

下载完之后，打开找到里面的dubbo.Properties，需要更改一点点[2181端口对应的zookeeper监听的端口]

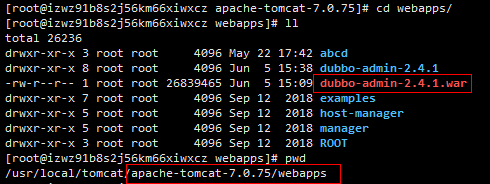
dubbo.registry.address=zookeeper://39.108.173.63:2181

dubbo.admin.root.password=root

dubbo.admin.guest.password=guest

把红色区域的IP更换成自己的即可

然后把war包放在tomcat的webapps路径下



以上基本就完成了dubbo-admin的相关工作，启动tomcat即可【启动有问题的可以看下第四点：常见问题】

启动完成之后访问

http://39.108.173.63:8080/dubbo-admin-2.4.1/



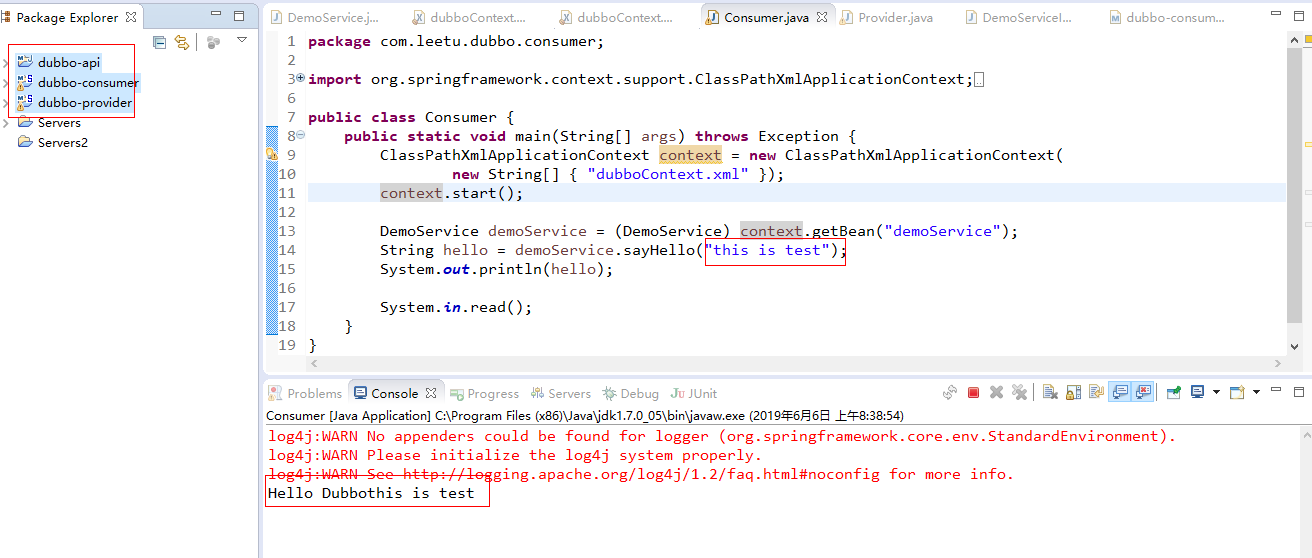
root/root登陆

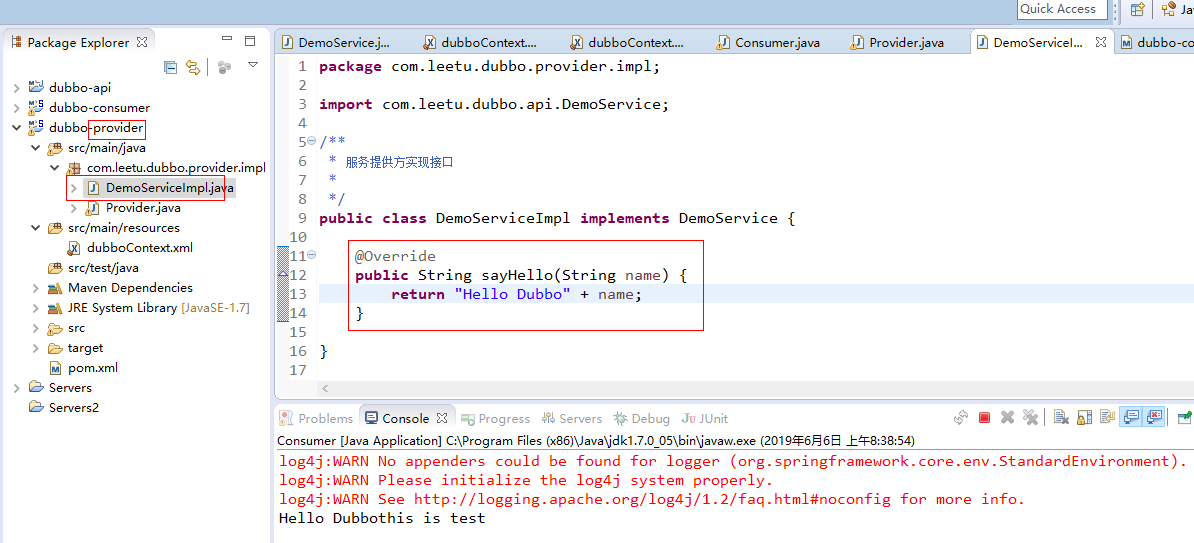


3.创建Spring项目并访问

Spring项目的创建Demo，是可以直接拿来使用的，导入Eclipse之后更改里面的公网IP即可，对应的代码下载地址：

<https://download.csdn.net/download/weixin_38964895/11229138>











使用时候三个项目分别放在三个tomcat

先启动api

再启动provider

再启动consumer

再启动provider中的main

再启动consumer中的main

4.常见问题及解决方案

  在搭建过程中，zookeeper搭建没有遇到问题，遇到也都是可以百度到的，下面针对dubbo启动的一次错误日志，给出我此次的解决方案：

错误日志：

java.net.UnknownHostException: izwz91b8s2j56kmxiwxcz: izwz91b8s2j56kmxiwxcz

at java.net.InetAddress.getLocalHost(InetAddress.java:1473)

at com.alibaba.dubbo.common.utils.NetUtils.getLocalAddress0(NetUtils.java:203)

at com.alibaba.dubbo.common.utils.NetUtils.getLocalAddress(NetUtils.java:190)

at com.alibaba.dubbo.common.utils.NetUtils.getLocalHost(NetUtils.java:154)

at com.alibaba.dubbo.governance.sync.RegistryServerSync.<clinit>(RegistryServerSync.java:46)

at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)

at sun.reflect.NativeConstructorAccessorImpl.newInstance(NativeConstructorAccessorImpl.java:57)

at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAccessorImpl.java:45)

at java.lang.reflect.Constructor.newInstance(Constructor.java:526)

at org.springframework.beans.BeanUtils.instantiateClass(BeanUtils.java:100)

at org.springframework.beans.factory.support.SimpleInstantiationStrategy.instantiate(SimpleInstantiationStrategy.java:61)

at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.instantiateBean(AbstractAutowireCapableBeanFactory.java:877)

at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.createBeanInstance(AbstractAutowireCapableBeanFactory.java:839)

at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.doCreateBean(AbstractAutowireCapableBeanFactory.java:440)

at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory$1.run(AbstractAutowireCapableBeanFactory.java:409)

at java.security.AccessController.doPrivileged(Native Method)

at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.createBean(AbstractAutowireCapableBeanFactory.java:380)

at org.springframework.beans.factory.support.AbstractBeanFactory$1.getObject(AbstractBeanFactory.java:264)

at org.springframework.beans.factory.support.DefaultSingletonBeanRegistry.getSingleton(DefaultSingletonBeanRegistry.java:222)

at org.springframework.beans.factory.support.AbstractBeanFactory.doGetBean(AbstractBeanFactory.java:261)

at org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:185)

at org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:164)

at org.springframework.beans.factory.support.DefaultListableBeanFactory.findAutowireCandidates(DefaultListableBeanFactory.java:671)

at org.springframework.beans.factory.support.DefaultListableBeanFactory.resolveDependency(DefaultListableBeanFactory.java:610)

at org.springframework.beans.factory.annotation.AutowiredAnnotationBeanPostProcessor$AutowiredFieldElement.inject(AutowiredAnnotationBeanPostProcessor.java:412)

at org.springframework.beans.factory.annotation.InjectionMetadata.injectFields(InjectionMetadata.java:105)

at org.springframework.beans.factory.annotation.AutowiredAnnotationBeanPostProcessor.postProcessAfterInstantiation(AutowiredAnnotationBeanPostProcessor.java:240)

at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.populateBean(AbstractAutowireCapableBeanFactory.java:959)

at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.doCreateBean(AbstractAutowireCapableBeanFactory.java:472)

at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory$1.run(AbstractAutowireCapableBeanFactory.java:409)

at java.security.AccessController.doPrivileged(Native Method)

at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.createBean(AbstractAutowireCapableBeanFactory.java:380)

at org.springframework.beans.factory.support.AbstractBeanFactory$1.getObject(AbstractBeanFactory.java:264)

at org.springframework.beans.factory.support.DefaultSingletonBeanRegistry.getSingleton(DefaultSingletonBeanRegistry.java:222)

at org.springframework.beans.factory.support.AbstractBeanFactory.doGetBean(AbstractBeanFactory.java:261)

at org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:185)

at org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:164)

at org.springframework.beans.factory.support.DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.java:429)

at org.springframework.context.support.AbstractApplicationContext.finishBeanFactoryInitialization(AbstractApplicationContext.java:728)

at org.springframework.context.support.AbstractApplicationContext.refresh(AbstractApplicationContext.java:380)

at org.springframework.web.context.ContextLoader.createWebApplicationContext(ContextLoader.java:255)

at org.springframework.web.context.ContextLoader.initWebApplicationContext(ContextLoader.java:199)

at com.alibaba.citrus.webx.context.WebxComponentsLoader.initWebApplicationContext(WebxComponentsLoader.java:117)

at org.springframework.web.context.ContextLoaderListener.contextInitialized(ContextLoaderListener.java:45)

at org.apache.catalina.core.StandardContext.listenerStart(StandardContext.java:5118)

at org.apache.catalina.core.StandardContext.startInternal(StandardContext.java:5634)

at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:145)

at org.apache.catalina.core.ContainerBase.addChildInternal(ContainerBase.java:899)

at org.apache.catalina.core.ContainerBase.addChild(ContainerBase.java:875)

at org.apache.catalina.core.StandardHost.addChild(StandardHost.java:652)

at org.apache.catalina.startup.HostConfig.deployWAR(HostConfig.java:1092)

at org.apache.catalina.startup.HostConfig$DeployWar.run(HostConfig.java:1984)

at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:471)

at java.util.concurrent.FutureTask.run(FutureTask.java:262)

at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1145)

at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:615)

at java.lang.Thread.run(Thread.java:745)

Caused by: java.net.UnknownHostException: izwz91b8s2j56km66xiwxcz

at java.net.Inet4AddressImpl.lookupAllHostAddr(Native Method)

at java.net.InetAddress$1.lookupAllHostAddr(InetAddress.java:901)

at java.net.InetAddress.getAddressesFromNameService(InetAddress.java:1293)

at java.net.InetAddress.getLocalHost(InetAddress.java:1469)

... 56 more

WARN utils.ConfigUtils - [DUBBO] No dubbo.properties found on the class path., dubbo version: 2.4.1, current host: 172.19.20.275

错误日志主要看下最后，发现他提到：current host: 172.19.20.275，

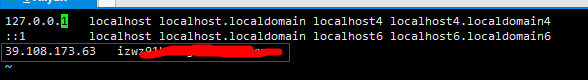
而上面又出现一个：UnknownHostException: izwz91b8s2j56kmxiwxcz

前者对应我阿里云的公网IP，后者对应我的阿里云实例ID，

我的理解是：他默认映射的是私有IP而非共有IP，所以需要修改host文件，增加一组映射

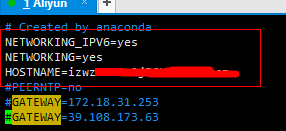
具体修改方案如下：

vi /etc/hosts



遮挡部分对应自己的实例ID即可

vim /etc/sysconfig/network



修改完之后需要重启Linux，然后再依次启动zookeeper，dubbo。就可以出现第三步的效果了

---------------------

原文：https://blog.csdn.net/weixin\_38964895/article/details/91043574