**camel 环境搭建**

**1、Vagrant 环境**

**1.1、Vagrant是什么**

[**Vagrant**](https://www.vagrantup.com/)**是一款用来构建虚拟开发环境的工具，它底层支持VirtualBox、VMware甚至AWS作为虚拟机系统，提供易于配置，重复性好，便携式的工作环境。也可以和puppet，chef结合，实现虚拟机管理的自动化。**

**1.2、Vagrant能做什么**

* **统一开发环境。一次配置打包，统一分发给团队成员，统一团队开发环境，解决诸如“编码问题”，“缺少模块”，“配置文件不同”带来的问题；**
* **避免重复搭建开发环境。新员工加入，不用浪费时间搭建开发环境，快速加入开发，减少时间成本的浪费；**
* **多个相互隔离开发环境。可以在不用box里跑不同的语言，或者编译安装同一语言不同版本，搭建多个相互隔离的开发环境，卸载清除时也很快捷轻松**

**1.3、Vagrant相关软件下载**

**下载vagrant**

[**http://www.vagrantup.com/downloads.html**](http://www.vagrantup.com/downloads.html)

**下载VirtualBox**

[**https://www.virtualbox.org/wiki/Downloads**](https://www.virtualbox.org/wiki/Downloads)

**注意：以下以安装在windows上为例，VirtualBox和Vagrant不要装在同一分区里，Vagrant默认选项安装到C盘**

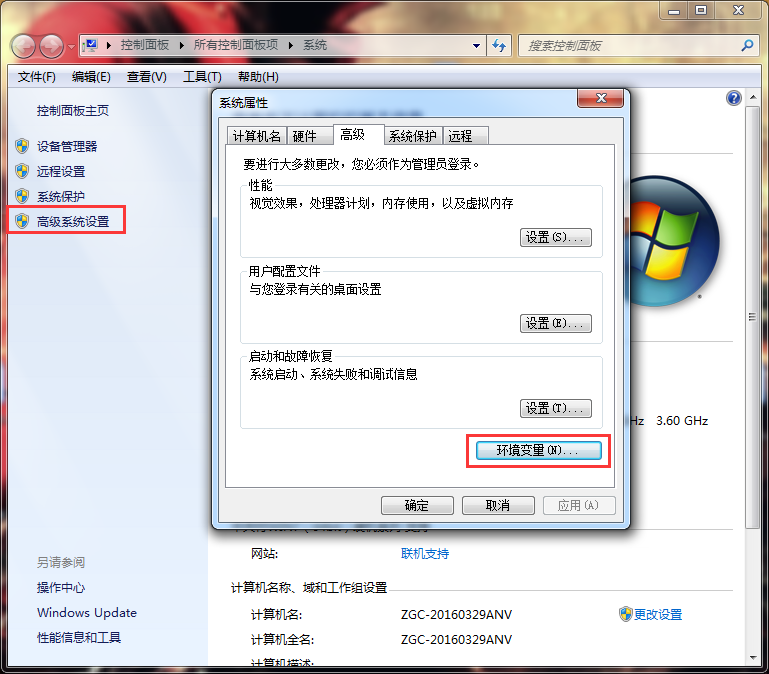
**1.4、Vagrant配置环境变量（windows）**

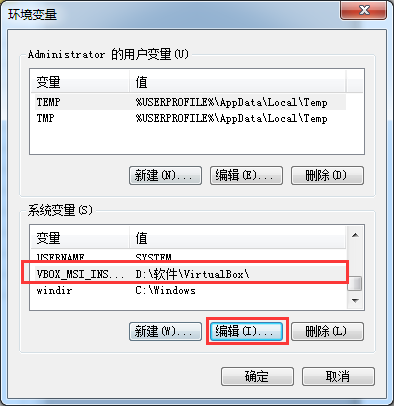
**1.4.1、VirtualBox**

**VirtualBox: 将%VirtualBox\_homt%添加到Path中，这样Vagrant才能被识别**

**变量名：VBOX\_MSI\_INSTALL\_PATH**

**变量值：D:\软件\VirtualBox\**





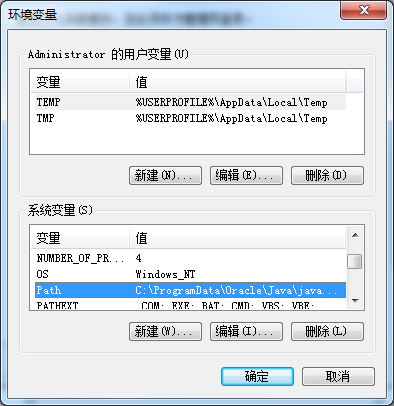


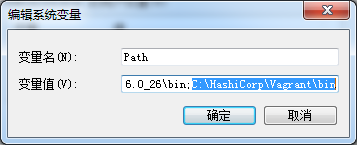
**1.4.2、Vagrant**

**Vagrant: 安装成功后，自动在Path中添加%Vagrant\_home%/bin，检查一下**

**变量名：Path**

**变量值：…;C:\HashiCorp\Vagrant\bin**

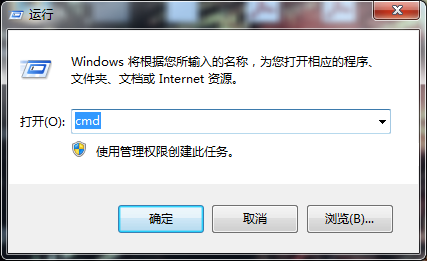




**1.5、启动Vagrant**

**1.5.1、创建vagrant环境目录**

**进入运行命令模式，创建测试文件夹（vagrant环境所在目录）**



**创建vagrantdemo目录**

**C:\Users\Administrator>md vagrantdemo**

**C:\Users\Administrator>cd vagrantdemo**

**1.5.2、vagrant box下载**

**box是一个zip包，包含了vagrant的配置信息和VirtualBox的虚拟机镜像文件**

**下载box**

[**http://www.vagrantbox.es/**](http://www.vagrantbox.es/)

**1.5.3、使用box方式安装系统**

**命令格式**

**vagrant box add "box\_name" remoteUrl or localFile**

**" box\_name "可以是任意字符，用于标识box**

**使用remoteUrl（远程地址）添加box**

**vagrant box add "centos7"**

**https://github.com/holms/vagrant-centos7-box/releases/download/7.1.1503.001/CentOS-7.1.1503-x86\_64-netboot.box**

**这种方式需要即时下载，较慢，建议下载下来box，在本地添加**

**或是**

**使用localFile（本地box文件）添加box**

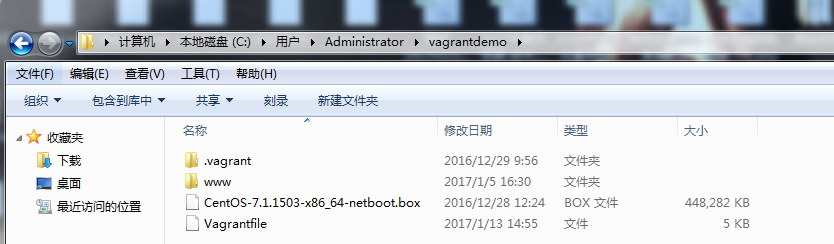
**后面加绝对路径或进入同层目录**

**vagrant box add "centos7" CentOS-7.1.1503-x86\_64-netboot.box**

**1.5.4、初始化和Vagrantfile**

**vagrant init "box\_name"**

**初始化后会在当前目录（C:\Users\Administrator\vagrantdemo）生成以一个Vagrantfile文件**



**Vagrantfile详细使用请自行查阅资料或官方文档**

**这里环境使用的Vagrantfile，可以复制使用**

**# -\*- mode: ruby -\*-**

**# vi: set ft=ruby :**

**Vagrant.configure("2") do |config|**

**config.vm.define :admin do |admin|**

**admin.vm.provider "virtualbox" do |v|**

**v.customize ["modifyvm", :id, "--name", "admin", "--memory", "1024"]**

**end**

**admin.vm.box = "centos7"**

**admin.vm.hostname = "camel-admin"**

**admin.vm.network "public\_network"**

**end**

**config.vm.define :agent1 do |agent1|**

**agent1.vm.provider "virtualbox" do |v|**

**v.customize ["modifyvm", :id, "--name", "agent1", "--memory", "512"]**

**end**

**agent1.vm.box = "centos7"**

**agent1.vm.hostname = "agent1"**

**agent1.vm.network "public\_network"**

**end**

**config.vm.define :agent2 do |agent2|**

**agent2.vm.provider "virtualbox" do |v|**

**v.customize ["modifyvm", :id, "--name", "agent2", "--memory", "512"]**

**end**

**agent2.vm.box = "centos7"**

**agent2.vm.hostname = "agent2"**

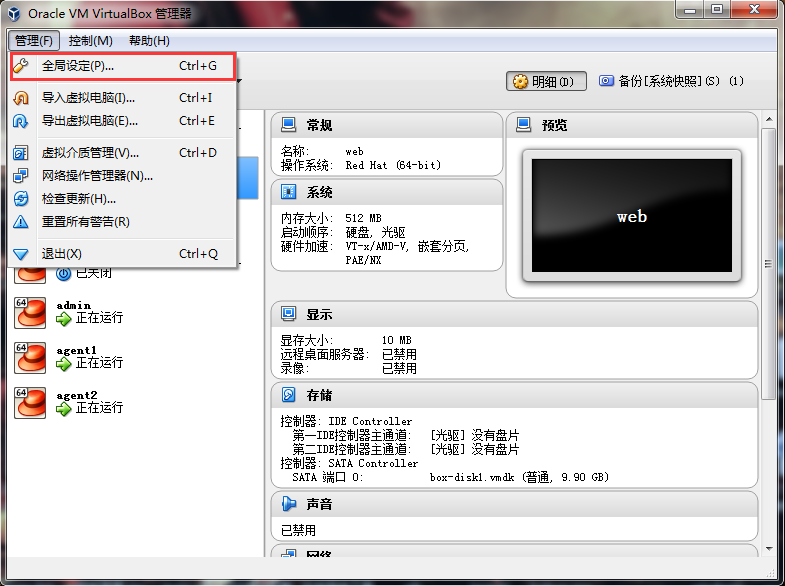
**agent2.vm.network "public\_network"**

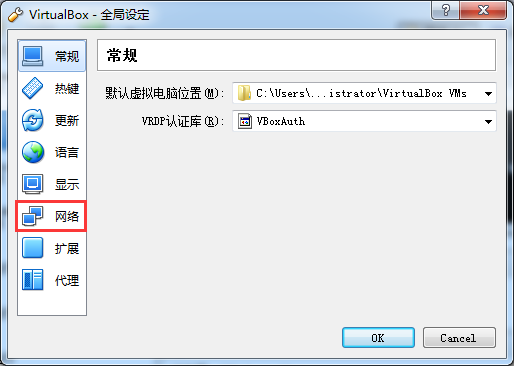
**end**

**end**

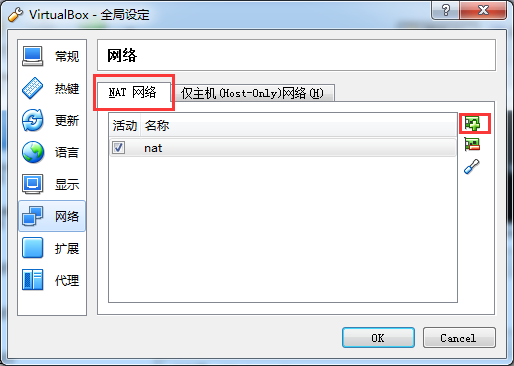
**1.5.5、public\_network配置**

**配置Virtualbox网络，使vagrant虚拟机获得本地dhcp分发的ip**

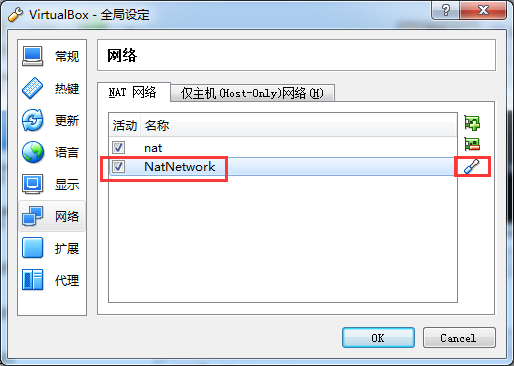
****

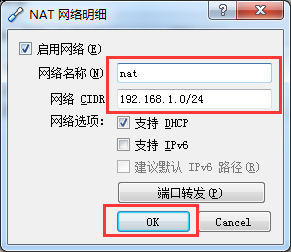
****

**添加nat网络**

****

**编辑nat网络**

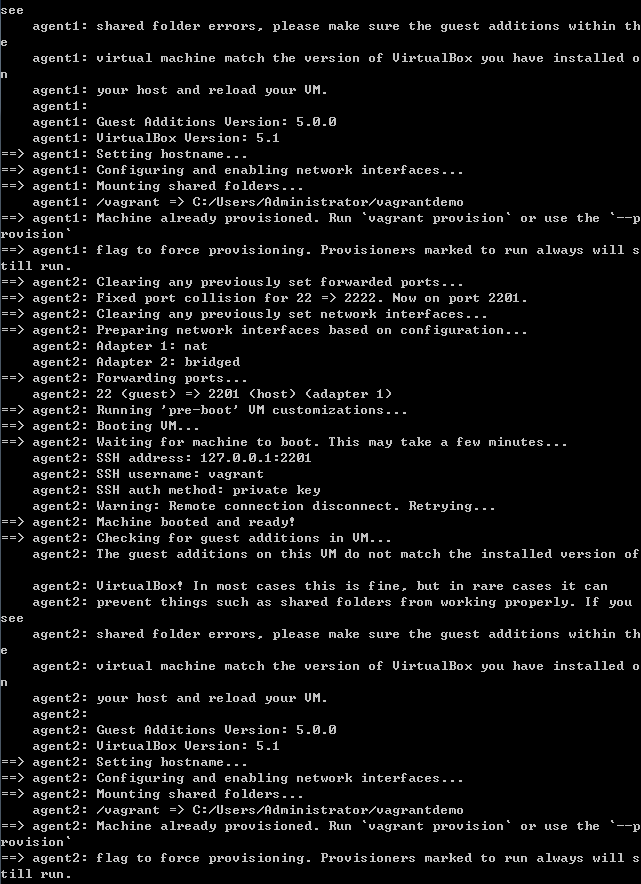
****

****

**1.5.6、启动虚拟机**

**vagrant up**





**启动后会打开3太虚拟机admin，agent1，agent2**

**对应127.0.0.1的2222、2200、2201端口**

**并获得本地ip地址**

**1.6、使用ssh登录到虚拟机**

**windows默认没有ssh命令，可以安装babun或cmder**

**使用如下命令连接**

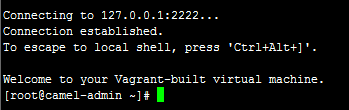
**C:\Users\Administrator\vagrantdemo>vagrant ssh**

**下面使用xshell连接，初始账号密码都是vagrant**

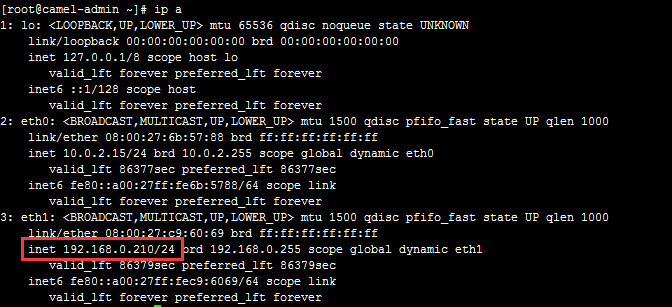
**连接camel-admin**

**ssh 127.0.0.1 2222**

**连接后使用sudo passwd root修改root密码**

****

**使用ip a查看获取到的局域网dhcp ip**

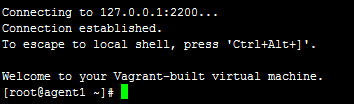
****

**再次登录**

**ssh 192.168.1.210连接admin**

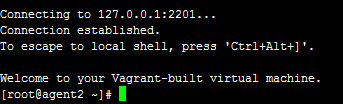
**连接agent1**

**ssh 127.0.0.1 2200**



**连接agent2**

**ssh 127.0.0.1 2201**



**其他具体操作同admin操作**

**2、camel 环境**

**admin端需要安装camel-admin和Dengine**

**agent端需要安装camel-agent和Dengine**

**2.1、admin端**

**2.1.1、系统初始化和下载camel**

**清除vagrant默认防火墙策略**

**chmod 755 /etc/rc.d/rc.local**

**echo "/usr/sbin/iptables -F" >> /etc/rc.local**

**iptables -F**

**关闭也可以**

**systemctl stop firewalld**

**systemctl disable firewalld**

**依赖包安装**

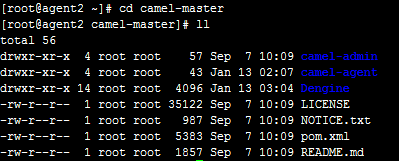
**yum install -y vim unzip maven net-tools git gcc gcc-c++ automake autoconf libtool make ncurses-devel zlib zlib-devel libtermcap-devel libevent-devel readline-devel patch**

**下载camel-master.zip**

**项目地址**

**https://github.com/dianping/camel**

**unzip camel-master.zip**



**2.1.2、安装mariadb**

**yum -y install mariadb mariadb-server**

**systemctl start mariadb**

**systemctl enable mariadb**

**mariadb初始化**

**mysql -uroot**

**MariaDB [(none)]> grant all on \*.\* to root@localhost identified by '123456';**

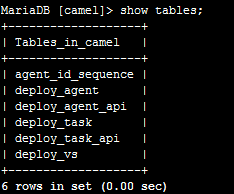
**MariaDB [(none)]> flush privileges;**

**MariaDB [(none)]> create database camel;**

**MariaDB [(none)]> use camel;**

**MariaDB [(none)]>**

**source /root/camel-master/camel-admin/src/main/resources/init-data/create\_table.sql**

****

**2.1.3、安装mongodb**

**下载mongodb**

[**https://www.mongodb.com/**](https://www.mongodb.com/)

**注意：下载3.0.7版本，3.0.7以上版本有问题最新3.4.1启动报错**

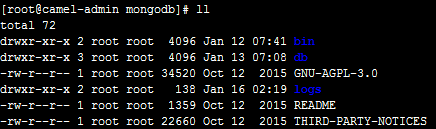
**mongodb初始化**

**tar zxvf mongodb-linux-x86\_64-rhel70-3.0.7.tgz**

**mv mongodb-linux-x86\_64-rhel70-3.0.7 mongodb**

**cd mongodb**

**mkdir db logs**



**cd bin**

**vim mongodb.conf**

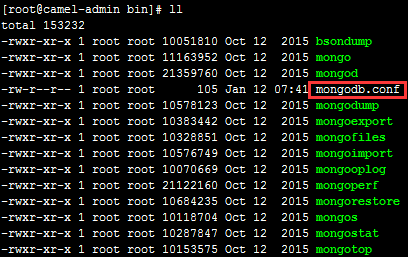
**dbpath=/root/mongodb/db**

**logpath=/root/mongodb/logs/mongodb.log**

**port=27017**

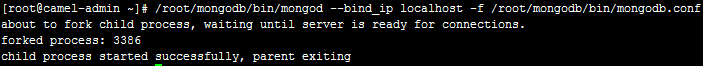
**fork=true**

**nohttpinterface=true**



**启动mongodb**

**/root/mongodb/bin/mongod --bind\_ip localhost -f /root/mongodb/bin/mongodb.conf**

****

**mongodb建库**

**/root/mongodb/bin/mongo**

**> use camel\_runtime**

**switched to db camel\_runtime**

**> db.movie.insert({"name":"tutorials yiibai"})**

**WriteResult({ "nInserted" : 1 })**

**> use camel\_nginx\_log**

**switched to db camel\_nginx\_log**

**> db.movie.insert({"name":"tutorials yiibai"})**

**WriteResult({ "nInserted" : 1 })**

**> use camel\_config**

**switched to db camel\_config**

**> db.movie.insert({"name":"tutorials yiibai"})**

**WriteResult({ "nInserted" : 1 })**

**> show dbs**

**admin 0.000GB**

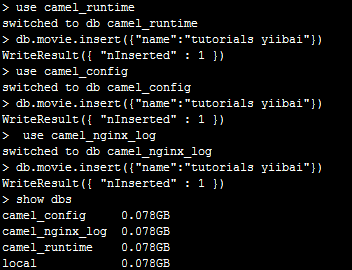
**camel\_config 0.000GB**

**camel\_nginx\_log 0.000GB**

**camel\_runtime 0.000GB**

**local 0.000GB**

**> exit**

****

**2.1.4、配置camel-admin和生成camel-admin的war包**

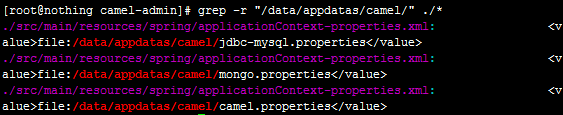
**camel-admin配置文件修改**

**grep -r "/data/appdatas/camel/" ./\***

**./src/main/resources/spring/applicationContext-properties.xml: <value>file:/data/appdatas/camel/jdbc-mysql.properties</value>**

**./src/main/resources/spring/applicationContext-properties.xml: <value>file:/data/appdatas/camel/mongo.properties</value>**

**./src/main/resources/spring/applicationContext-properties.xml: <value>file:/data/appdatas/camel/camel.properties</value>**

****

**mkdir -p /data/appdatas/camel/**

**cp /root/camel-master/camel-admin/src/main/resources/init-data/camel.properties /data/appdatas/camel/**

**vim camel.properties**

**# threshold of local nginx config check**

**local.nginx.config.check=true**

**cp /root/camel-master/camel-admin/src/main/resources/init-data/jdbc-mysql.properties /data/appdatas/camel/**

**vim /data/appdatas/camel/jdbc-mysql.properties**

**jdbc.driverClassName=com.mysql.jdbc.Driver**

**jdbc.username=root**

**jdbc.password=123456**

**jdbc.maxPoolSize=50**

**jdbc.minPoolSize=1**

**jdbc.initialPoolSize=1**

**jdbc.idleConnectionTestPeriod=1800**

**jdbc.maxIdleTime=3600**

**jdbc.checkoutTimeout=5000**

**jdbc.url=jdbc:mysql://127.0.0.1:3306/camel?useUnicode=true&characterEncoding=utf8&zeroDateTimeBehavior=convertToNull&noAccessToProcedureBodies=true&socketTimeout=5000&connectTimeout=5000**

**红色camel为之前mariadb新建的数据库名**

**cp src/main/resources/init-data/mongo.properties /data/appdatas/camel/**

**vim /data/appdatas/camel/mongo.properties**

**mongodb.url=127.0.0.1:27017**

**mongodb.dbname\_config=camel\_config**

**mongodb.dbname\_nginx\_log=camel\_nginx\_log**

**mongodb.dbname\_runtime=camel\_runtime**

**isCluster=false**

**connections-per-host=1800**

**slave-ok=false**

**添加dns**

**否则后面下载无法解析**

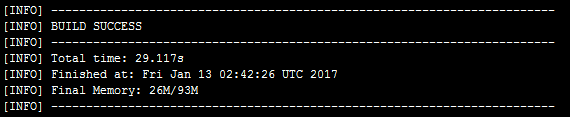
**echo " nameserver 8.8.8.8" >> /etc/resolv.conf**

**构建camel-admin**

**cd /root/camel-master/camel-admin/**

**mvn clean compile**

**时间较长，最后看到BUILD SUCCESS构建成功**

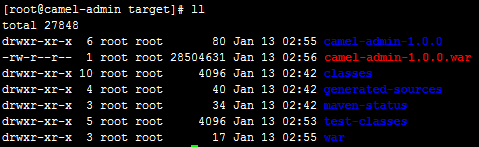
****

**生成camel-admin 的war包**

**cd /root/camel-master/camel-admin/**

**mvn install package -Dmaven.test.skip=true**

**在/root/camel-master/camel-admin/target/目录下生成war包**

****

**2.1.5、安装tomcat**

**这里使用tomcat7，jdk在之前camel-admin构建过程中已安装，为openjdk**

**将之前生成的camel-admin 的war包放到tomcat项目目录**

**cp camel-admin-1.0.0.war /opt/apache-tomcat-7.0.59/webapps/**

**2.1.6、安装Dengine**

**注意：Dengine默认安装到/usr/loca/nginx，安装Dengine之前，删除系统中安装在/usr/loca/nginx目录的nginx，其他nginx不要安装到/usr/loca/nginx目录**

**cd /root/camel-master/Dengine/**

**./install\_dengine**

**权限确认**

**/usr/local/nginx/conf/phoenix-slb/权限777**

**chmod 777 /usr/local/nginx/conf/phoenix-slb/**

**访问策略配置**

**vim /usr/local/nginx/conf/nginx\_status.conf**

**req\_status\_zone server "$host:$server\_addr:$server\_port" 10M;**

**check\_shm\_size 50M;**

**req\_status server;**

**server {**

**listen 6666;**

**server\_name aaabbbccc;**

**location /status {**

**check\_status;**

**access\_log off;**

**allow 192.168.0.210;**

**# deny all;**

**}**

**location /degrade{**

**upstream\_degrade\_interface;**

**access\_log off;**

**allow 192.168.0.210;**

**# deny all;**

**}**

**location / {**

**return 444;**

**}**

**}**

**server {**

**listen 80 default\_server;**

**server\_name aaabbbccc;**

**location /status {**

**echo "ok";**

**default\_type text/plain;**

**access\_log off;**

**allow 192.168.0.210;**

**# deny all;**

**}**

**location /reqstatus {**

**req\_status\_show;**

**access\_log off;**

**allow 192.168.0.210;**

**# deny all;**

**}**

**location / {**

**return 444;**

**}**

**error\_page 404 403 =444 @static;**

**location @static{**

**return 444;**

**}**

**}**

**2.1.7、启动Dengine和tomcat**

**启动Dengine**

**/usr/local/nginx/sbin/nginx**

**开机自启动**

**echo "/usr/local/nginx/sbin/nginx" >> /etc/rc.local**

**启动tomcat**

**cd /opt/apache-tomcat-7.0.59/bin/**

**./startup.sh**

**开机自启动**

**echo "/opt/apache-tomcat-7.0.59/bin/startup.sh" >> /etc/rc.local**

**2.1.8、访问管理端首页**

**http://192.168.0.210:8080/camel-admin-1.0.0/**

****

**2.2、agent端（agent1/agent2）**

**2.2.1、系统初始化**

**清除vagrant默认防火墙策略**

**chmod 755 /etc/rc.d/rc.local**

**echo "/usr/sbin/iptables -F" >> /etc/rc.local**

**iptables -F**

**关闭也可以**

**systemctl stop firewalld**

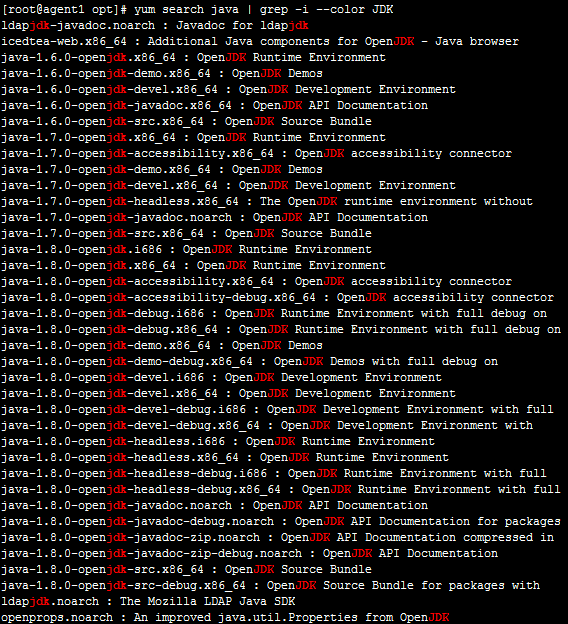
**systemctl disable firewalld**

**2.2.2、安装依赖包**

**yum -y install maven git gcc gcc-c++ automake autoconf libtool make ncurses-devel zlib zlib-devel libtermcap-devel libevent-devel readline-devel patch**

**2.2.3、安装JDK**

**yum search java | grep -i --color JDK**

****

**yum install java-1.7.0-openjdk**

**2.2.4、添加dns**

**vim /etc/resolv.conf**

**nameserver 8.8.8.8**

**2.2.5、运行camel-agent**

**cd /root/camel-master/camel-agent**

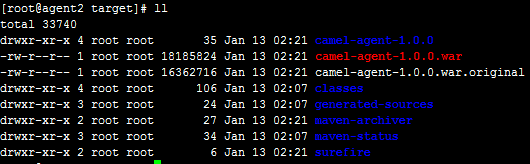
**运行agent**

**mvn spring-boot:run > /root/camel-agent.out 2>&1 &**

**生成war包（可选）**

**mvn clean package**

**可以将/root/camel-master/camel-agent/target目录下生成的war包放到tomcat工程目录运行**

****

**2.2.6、安装tomcat（可选）**

**注意：如果已经使用mvn spring-boot:run方式启动agent，跳过此步**

**这里使用tomcat7，jdk为java-1.7.0-openjdk**

**将之前生成的camel-admin 的war包放到tomcat项目目录**

**cp /root/camel-master/camel-agent/target/camel-agent-1.0.0.war /opt/apache-tomcat-7.0.59/webapps/**

**cd /opt/apache-tomcat-7.0.59/bin**

**./startup.sh**

**2.2.7、安装Dengine**

**注意：Dengine默认安装到/usr/loca/nginx，安装Dengine之前，删除系统中安装在/usr/loca/nginx目录的nginx，其他nginx不要安装到/usr/loca/nginx目录**

**cd /root/camel-master/Dengine/**

**./install\_dengine**

**权限确认**

**/usr/local/nginx/conf/phoenix-slb/权限777**

**chmod 777 /usr/local/nginx/conf/phoenix-slb/**

**访问策略配置**

**vim /usr/local/nginx/conf/nginx\_status.conf**

**req\_status\_zone server "$host:$server\_addr:$server\_port" 10M;**

**check\_shm\_size 50M;**

**req\_status server;**

**server {**

**listen 6666;**

**server\_name aaabbbccc;**

**location /status {**

**check\_status;**

**access\_log off;**

**allow 192.168.0.210;**

**# deny all;**

**}**

**location /degrade{**

**upstream\_degrade\_interface;**

**access\_log off;**

**allow 192.168.0.210;**

**# deny all;**

**}**

**location / {**

**return 444;**

**}**

**}**

**server {**

**listen 80 default\_server;**

**server\_name aaabbbccc;**

**location /status {**

**echo "ok";**

**default\_type text/plain;**

**access\_log off;**

**allow 192.168.0.210;**

**# deny all;**

**}**

**location /reqstatus {**

**req\_status\_show;**

**access\_log off;**

**allow 192.168.0.210;**

**# deny all;**

**}**

**location / {**

**return 444;**

**}**

**error\_page 404 403 =444 @static;**

**location @static{**

**return 444;**

**}**

**}**

**cd /root/camel-master/Dengine/**

**./install\_dengine**

**2.2.8、启动Dengine**

**启动Dengine**

**/usr/local/nginx/sbin/nginx**

**开机自启动**

**echo "/usr/local/nginx/sbin/nginx" >> /etc/rc.local**

**开机自启动camel-agent**

**echo " mvn spring-boot:run > /root/camel-agent.out 2>&1 &" >> /etc/rc.local**

**3、camel使用**

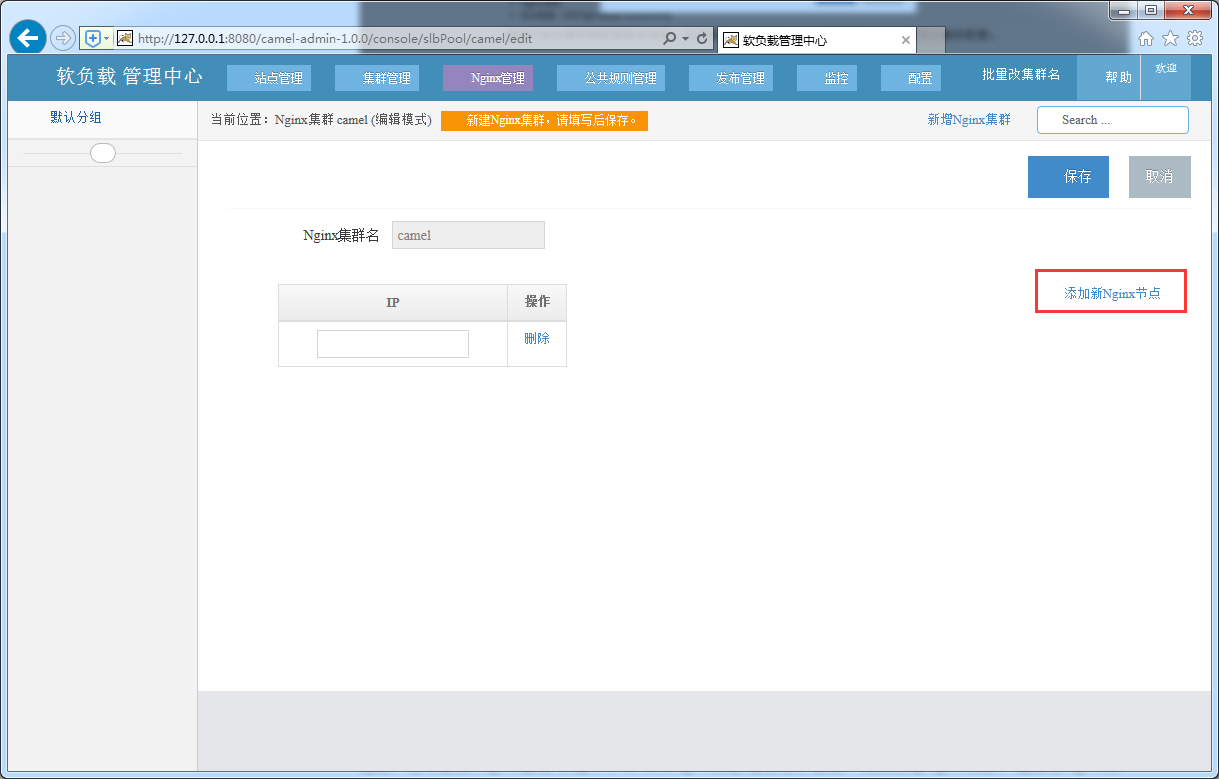
**3.1、创建nginx集群**

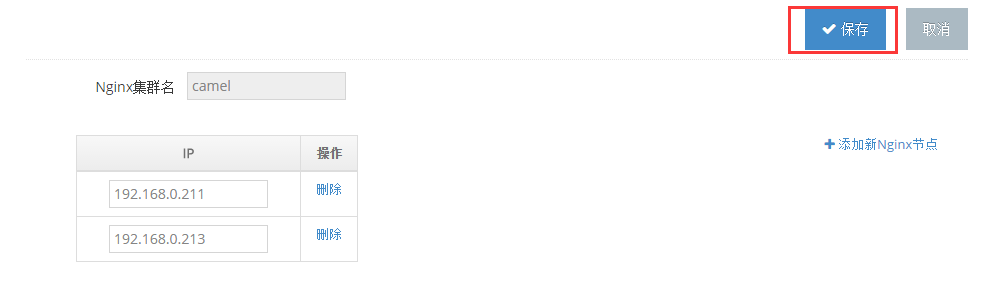


**3.2、nginx集群重命名**



**3.3、nginx集群添加节点**

****



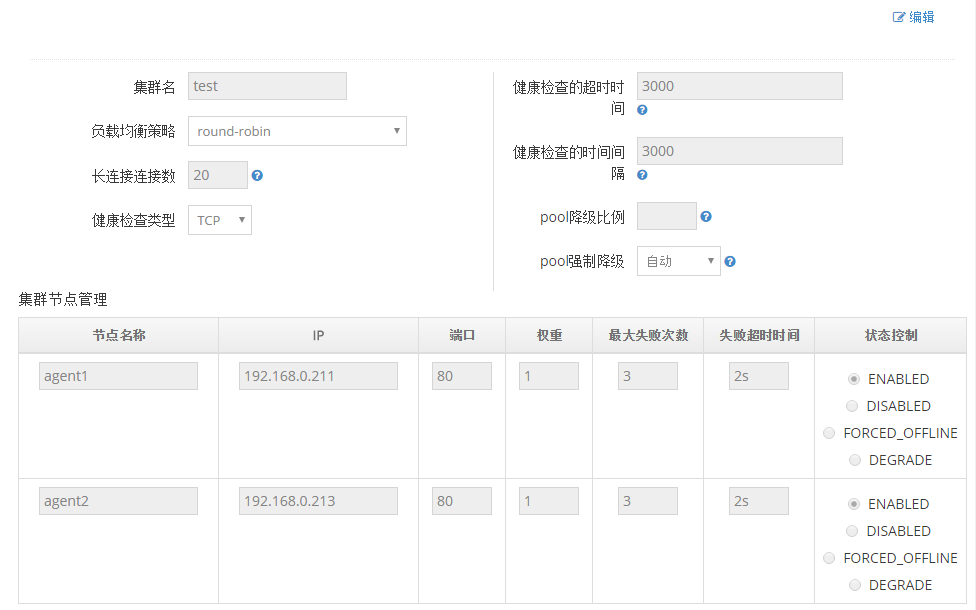
**3.4、新增集群**



**3.5、集群重命名**

****

****



**3.6、新增站点**



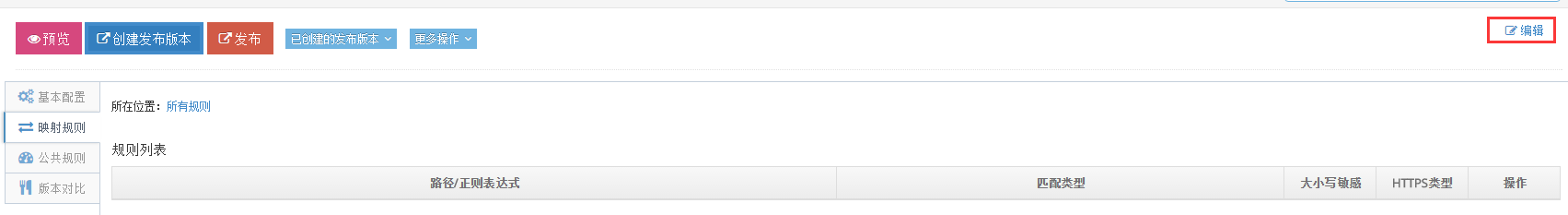
**3.7、站点命名**



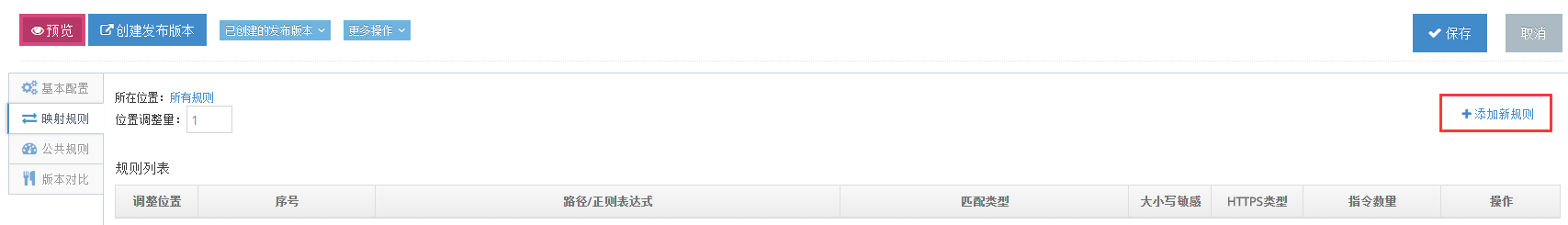


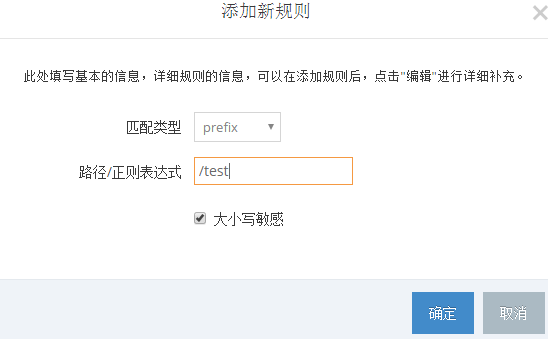
**3.8、映射规则**

**编辑**

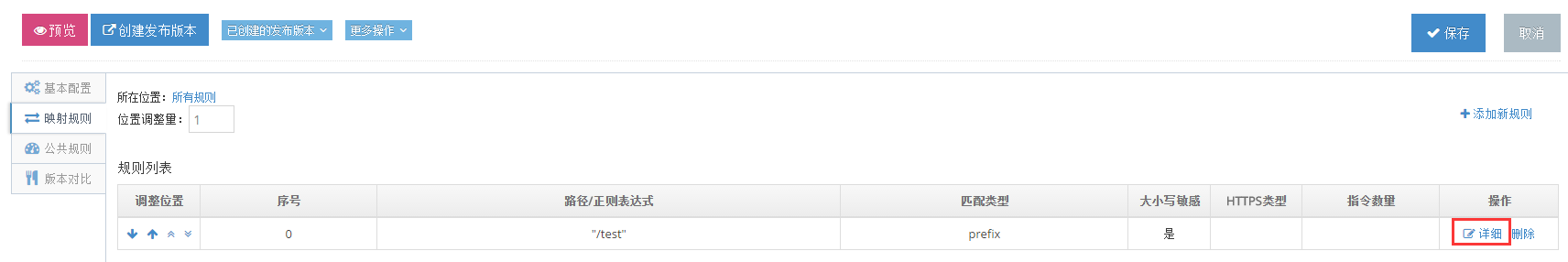


**添加新规则**

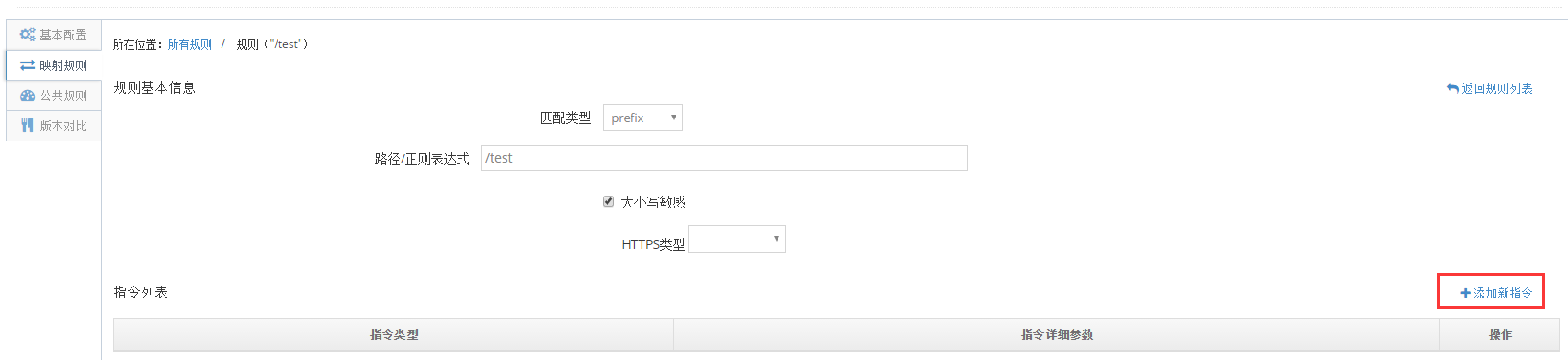


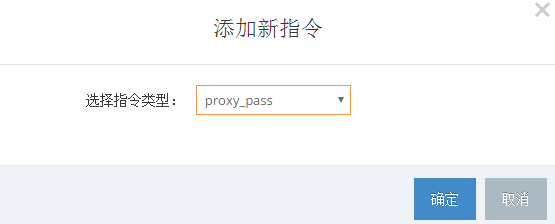


**详细**

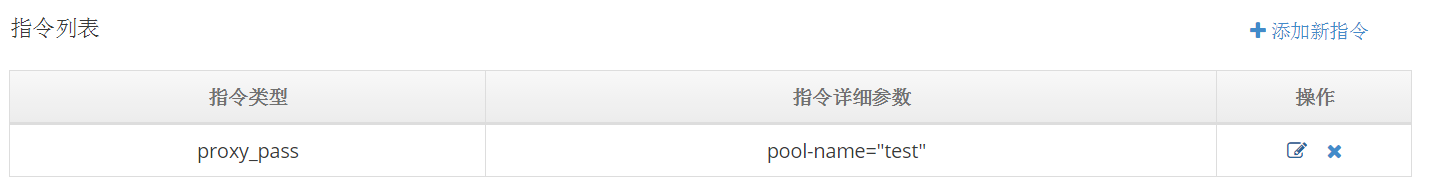
****

**添加新指令**







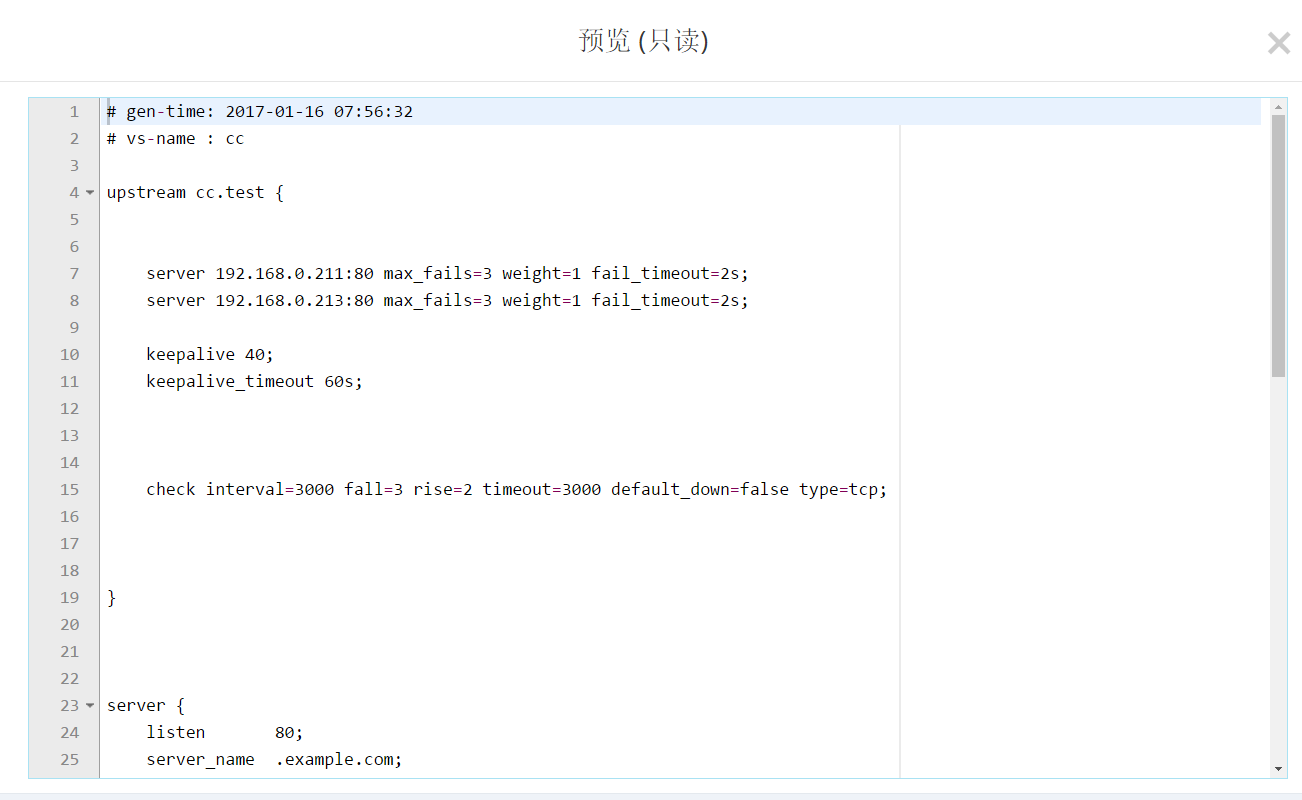


**保存**



**3.9、预览**









**3.10、创建发布版本**





**3.11、发布**



**选择发布版本**



**3.12、最终发布页面**



