毫米波集中查验技术预研

1、技术预研内容

1. 、项目应用开发服务器部署
2. 、Redis集群搭建部署架构

3>、RabbitMQ集群搭建部署架构

4>、RSA和AES加密混合加密流程算法和应用场景

5>、zxbbix部署

2、预研详细信息

2.1、项目开发应用服务器部署（jenkins+docker）

环境构建描述

在CentOS 7.4安装maven、eureka工程服务通过git以及推送gitlab。

jenkins+docker部署eureka服务，实现开发环境自动化部署代码

2.1.1、安装和启动docker

2.1.1.1安装和启动docker

yum update -y

yum install -y yum-utils

yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo

yum -y install docker-ce

systemctl start docker

查看可安装的版本

yum list docker-ce --showduplicates | sort -r

安装到指定版本

yum install docker-ce-18.06.3.ce-3.el7 -y

版本号查询

docker --version

2.1.1.2、设置镜像

vi /etc/docker/daemon.json

{

"registry-mirrors": ["https://aj2rgad5.mirror.aliyuncs.com"]

}

2.1.1.3、开放管理端口映射

vi /lib/systemd/system/docker.service

将第11行的ExecStart=/usr/bin/dockerd，替换为：

ExecStart=/usr/bin/dockerd -H tcp://0.0.0.0:2375 -H unix:///var/run/docker.sock -H tcp://0.0.0.0:13154

2375是管理端口，13154是备用端口

在~/.bashrc中写入docker管理端口

export DOCKER\_HOST=tcp://0.0.0.0:2375

source ~/.bashrc

2.1.1.4、重启docker

systemctl daemon-reload && service docker restart

2.1.1.5、测试docker是否正常安装和运行

docker run hello-world

Hello from Docker!

This message shows that your installation appears to be working correctly.

2.1.2、安装maven

wget http://mirror.bit.edu.cn/apache/maven/maven-3/3.6.2/binaries/apache-maven-3.6.2-bin.tar.gz

tar -zxvf apache-maven-3.5.0-bin.tar.gz

vi ~/.bashrc

export MAVEN\_HOME=/usr/local/apache-maven-3.5.0

export PATH=$PATH:$MAVEN\_HOME/bin

source ~/.bashrc

mvn -version

cd /usr/local/apache-maven-3.5.0/conf

vi settings.xml

<mirror>

<id>nexus-aliyun</id>

<mirrorOf>\*</mirrorOf>

<name>Nexus aliyun</name>

<url>http://maven.aliyun.com/nexus/content/groups/public</url>

</mirror>

2.1.3、安装git

yum install -y git

git --version

http://updates.jenkins-ci.org/download/war/

jenkins安装使用

2.198版本

将jenkins.war放到/usr/local中

执行java -jar httpPort=135135 jenkins.war，启动jenkins服务器

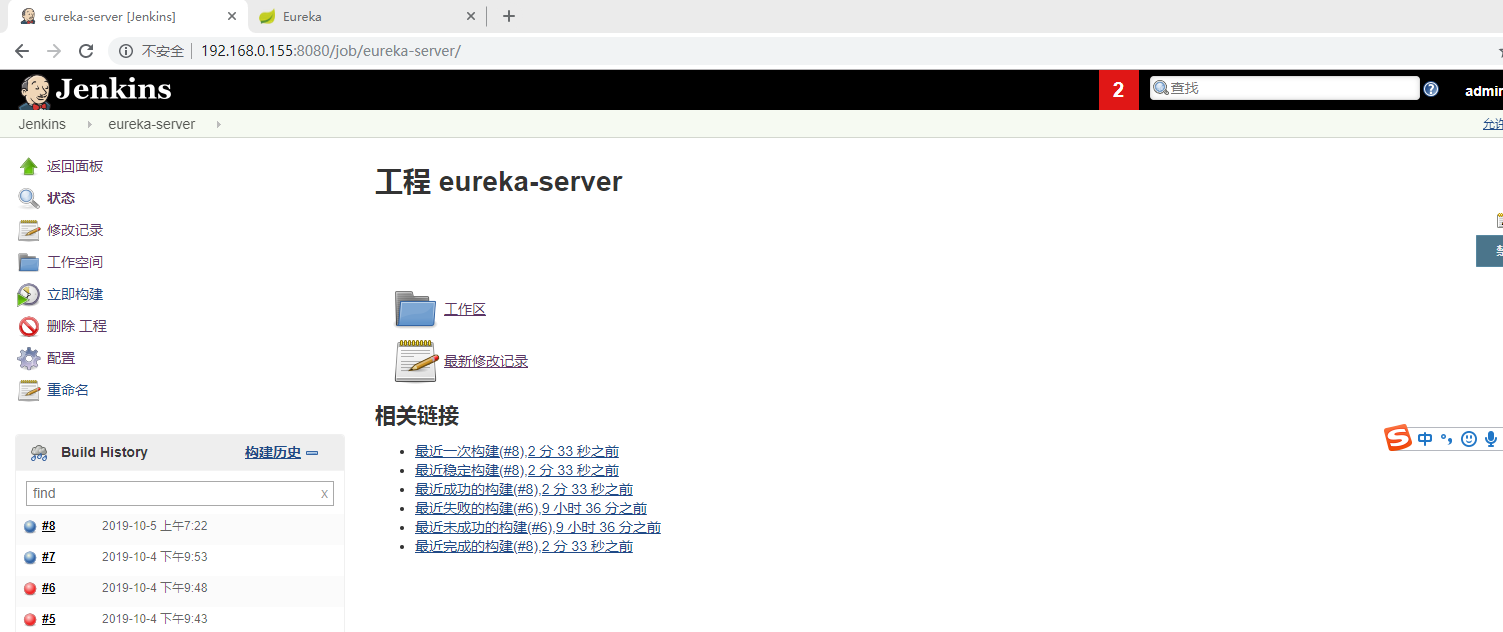
访问135135端口，可以看到jenkins的界面，需要输入秘钥

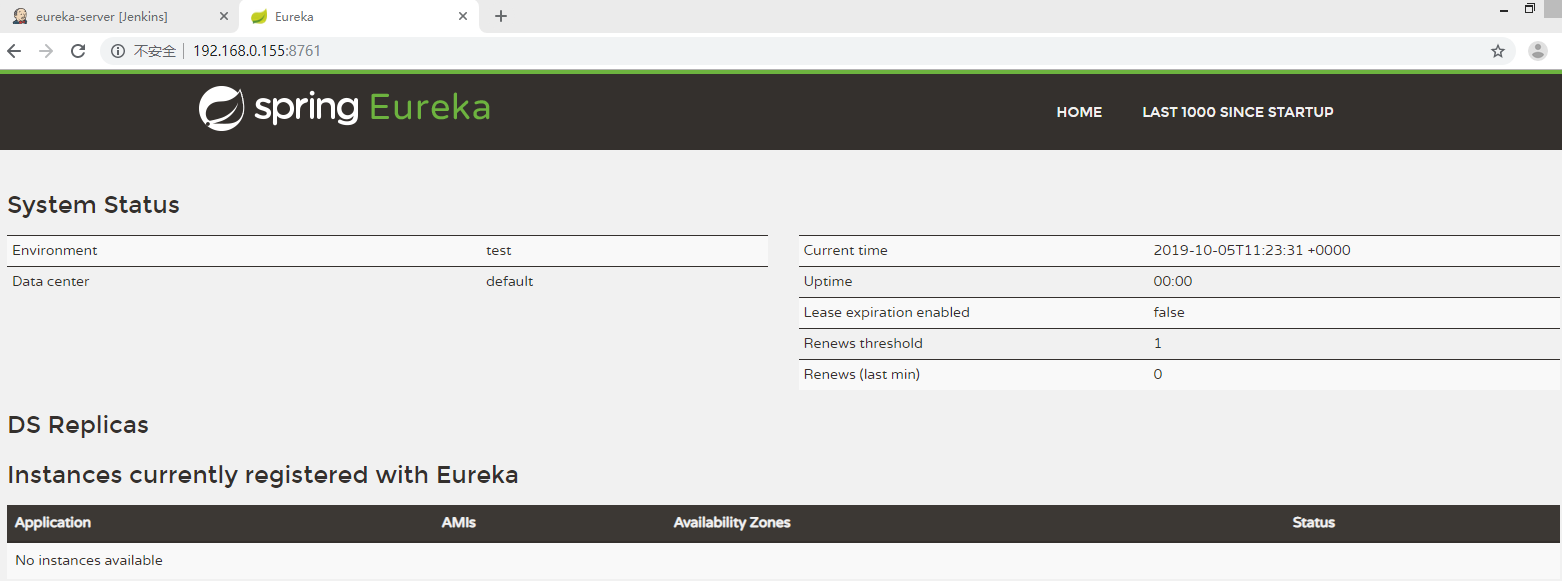
在控制台可以找到秘钥，也可以再/root/.jenkins/secrets/initialAdminPassword中找到

复制秘钥，进入界面

2.1.4、jenkins+docker部署eureka服务页面效果输出







2.2、Redis集群搭建部署架构

2.3、RabbitMQ集群搭建部署架构

2.3.1环境搭建：

安装rabbitmq 3.6.12，目前比较稳定的版本

2.3.1.1、安装编译工具

安装依赖包

yum install -y ncurses ncurses-base ncurses-devel ncurses-libs ncurses-static ncurses-term ocaml-curses ocaml-curses-devel

yum install -y openssl-devel zlib-devel

yum install -y make ncurses-devel gcc gcc-c++ unixODBC unixODBC-devel openssl openssl-devel

2.3.1.2、安装erlang

下载erlang：http://erlang.org/download/otp\_src\_20.0.tar.gz

tar -zxvf otp\_src\_20.0.tar.gz

cd otp\_src\_20.0

./configure --prefix=/usr/local/erlang --with-ssl -enable-threads -enable-smmp-support -enable-kernel-poll --enable-hipe --without-javac

make && make install

ln -s /usr/local/erlang/bin/erl /usr/local/bin/erl

vi ~/.bashrc

ERLANG\_HOME=/usr/local/erlang

PATH=$ERLANG\_HOME/bin:$PATH

source ~/.bashrc

erl

2.3.1.3、安装rabbitmq

http://www.rabbitmq.com/releases/rabbitmq-server/v3.6.12/rabbitmq-server-generic-unix-3.6.12.tar.xz

yum install -y xz

xz -d rabbitmq-server-generic-unix-3.6.12.tar.xz

tar -xvf rabbitmq-server-generic-unix-3.6.12.tar

mv rabbitmq\_server-3.6.1 rabbitmq-3.6.12

开启管理页面的插件

cd rabbitmq-3.6.12/sbin/

./rabbitmq-plugins enable rabbitmq\_management

后台启动rabbitmq server

./rabbitmq-server -detached

关闭rabbitmq server

./rabbitmqctl stop

添加管理员账号

./rabbitmqctl add\_user rabbitadmin 123456

./rabbitmqctl set\_user\_tags rabbitadmin administrator

进入管理页面

15672端口号，输入用户名和密码

### 2.3.2集群节点安装

2.3.2.1、修改集群用户与连接心跳检测

|  |
| --- |
| 注意修改vim /usr/lib/rabbitmq/lib/rabbitmq\_server-3.6.12/ebin/rabbit.app文件  修改：loopback\_users 中的 <<"guest">>,只保留guest  修改：heartbeat 为1 |

2.3.2.2、安装管理插件

|  |
| --- |
| //首先启动服务  /etc/init.d/rabbitmq-server start stop status restart  //查看服务有没有启动： lsof -i:5672  rabbitmq-plugins enable rabbitmq\_management  //可查看管理端口有没有启动： lsof -i:15672 或者 netstat -tnlp|grep 15672 |

2.3.2.3、服务指令

|  |
| --- |
| /etc/init.d/rabbitmq-server start stop status restart  验证单个节点是否安装成功：[http://192.168.0.131:15672/](http://192.168.11.71:15672/)   * 1. Ps：以上操作三个节点（131、132、133）同时进行操作 |

2.3.2.4、文件同步步骤

PS:选择131、132、133任意一个节点为Master（这里选择131为Master），也就是说我们需要把131的Cookie文件同步到132、133节点上去，进入/var/lib/rabbitmq目录下，把/var/lib/rabbitmq/.erlang.cookie文件的权限修改为13134，原来是400；然后把.erlang.cookie文件copy到各个节点下；最后把所有cookie文件权限还原为400即可。

|  |
| --- |
| /etc/init.d/rabbitmq-server stop  //进入目录修改权限；远程copy132、133节点，比如：  scp /var/lib/rabbitmq/.erlang.cookie 到192.168.0.132和192.168.0.133中 |

2.3.2.5、组成集群步骤

1、停止MQ服务

PS:我们首先停止3个节点的服务

|  |
| --- |
| rabbitmqctl stop |

2、组成集群操作

PS:接下来我们就可以使用集群命令，配置131、132、133为集群模式，3个节点（131、132、133）执行启动命令，后续启动集群使用此命令即可。

|  |
| --- |
| rabbitmq-server -detached |

3、slave加入集群操作（重新加入集群也是如此，以最开始的主节点为加入节点）

|  |
| --- |
| //注意做这个步骤的时候：需要配置/etc/hosts 必须相互能够寻址到  deatil132：rabbitmqctl stop\_app  deatil132：rabbitmqctl join\_cluster --ram rabbit@deatil131  deatil132：rabbitmqctl start\_app  deatil133：rabbitmqctl stop\_app  deatil133：rabbitmqctl join\_cluster rabbit@deatil131  deatil133：rabbitmqctl start\_app  //在另外其他节点上操作要移除的集群节点  rabbitmqctl forget\_cluster\_node rabbit@deatil24 |

4、修改集群名称

PS:修改集群名称（默认为第一个node名称）：

|  |
| --- |
| rabbitmqctl set\_cluster\_name rabbitmq\_cluster1 |

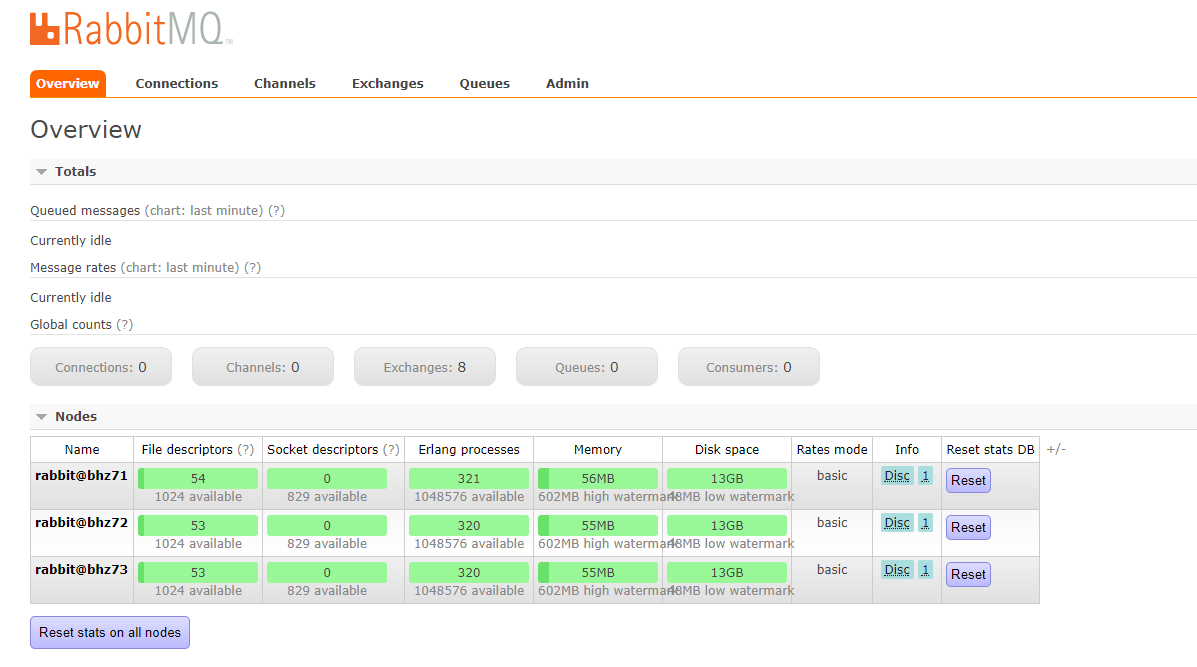
5、查看集群状态

PS:最后在集群的任意一个节点执行命令：查看集群状态

|  |
| --- |
| rabbitmqctl cluster\_status |
|  |

6、管控台界面

PS: 访问任意一个管控台节点：[http://192.168.0.131:15672](http://192.168.11.71:15672) 如图所示



2.3.2.6、配置镜像队列

PS:设置镜像队列策略（在任意一个节点上执行）

|  |
| --- |
| rabbitmqctl set\_policy ha-all "^" '{"ha-mode":"all"}' |

PS:将所有队列设置为镜像队列，即队列会被复制到各个节点，各个节点状态一致，RabbitMQ高可用集群就已经搭建好了,我们可以重启服务，查看其队列是否在从节点同步。

2.3.2.7、安装Ha-Proxy

1、Haproxy简介

HAProxy是一款提供高可用性、负载均衡以及基于TCP和HTTP应用的代理软件，HAProxy是完全免费的、借助HAProxy可以快速并且可靠的提供基于TCP和HTTP应用的代理解决方案。

HAProxy适用于那些负载较大的web站点，这些站点通常又需要会话保持或七层处理。

HAProxy可以支持数以万计的并发连接,并且HAProxy的运行模式使得它可以很简单安全的整合进架构中，同时可以保护web服务器不被暴露到网络上。

2、Haproxy安装

PS:134、135节点同时安装Haproxy，下面步骤统一

|  |
| --- |
| //下载依赖包  yum install gcc vim wget  //下载[haproxy](http://www.linuxea.com/tag/haproxy/)  wget http://www.[haproxy](http://www.linuxea.com/tag/haproxy/).org/download/1.6/src/haproxy-1.6.5.tar.gz  //解压  tar -zxvf haproxy-1.6.5.tar.gz -C /usr/local  //进入目录、进行编译、安装  cd /usr/local/haproxy-1.6.5  make TARGET=linux31 PREFIX=/usr/local/haproxy  make install PREFIX=/usr/local/haproxy  mkdir /etc/haproxy  //赋权  groupadd -r -g 149 haproxy  useradd -g haproxy -r -s /sbin/nologin -u 149 haproxy  //创建haproxy配置文件  touch /etc/haproxy/haproxy.cfg |

3、Haproxy配置

PS:haproxy 配置文件haproxy.cfg详解

|  |
| --- |
| vim /etc/haproxy/haproxy.cfg |

|  |
| --- |
| #logging options  global  log 1134.0.0.1 local0 info  maxconn 5120  chroot /usr/local/haproxy  uid 99  gid 99  daemon  quiet  nbproc 20  pidfile /var/run/haproxy.pid  defaults  log global  #使用4层代理模式，”mode http”为7层代理模式  mode tcp  #if you set mode to tcp,then you nust change tcplog into httplog  option tcplog  option dontlognull  retries 3  option redispatch  maxconn 2000  contimeout 5s  **##客户端空闲超时时间为 60秒 则HA 发起重连机制**  clitimeout 60s  **##服务器端链接超时时间为 15秒 则HA 发起重连机制**  srvtimeout 15s  #front-end IP for consumers and producters  listen rabbitmq\_cluster  bind 0.0.0.0:5672  #配置TCP模式  mode tcp  #balance url\_param userid  #balance url\_param session\_id check\_post 64  #balance hdr(User-Agent)  #balance hdr(host)  #balance hdr(Host) use\_domain\_only  #balance rdp-cookie  #balance leastconn  #balance source //ip  #简单的轮询  balance roundrobin  #rabbitmq集群节点配置 #inter 每隔五秒对mq集群做健康检查， 2次正确证明服务器可用，2次失败证明服务器不可用，并且配置主备机制  server deatil131 192.168.0.131:5672 check inter 5000 rise 2 fall 2  server deatil132 192.168.0.132:5672 check inter 5000 rise 2 fall 2  server deatil133 192.168.0.133:5672 check inter 5000 rise 2 fall 2  #配置haproxy web监控，查看统计信息  listen stats  bind 192.168.0.134:8100  mode http  option httplog  stats enable  #设置haproxy监控地址为http://localhost:8100/rabbitmq-stats  stats uri /rabbitmq-stats  stats refresh 5s |

4、启动haproxy

|  |
| --- |
| /usr/local/haproxy/sbin/haproxy -f /etc/haproxy/haproxy.cfg  //查看haproxy进程状态  ps -ef | grep haproxy |

5、访问haproxy

PS:访问如下地址可以对rmq节点进行监控：[http://192.168.1.134:8100/rabbitmq-stats](http://192.168.1.27:8100/rabbitmq-stats)

6、关闭haproxy

|  |
| --- |
| killall haproxy  ps -ef | grep haproxy |

2.3.2.8、安装KeepAlived

1、Keepalived简介

Keepalived，它是一个高性能的服务器高可用或热备解决方案，Keepalived主要来防止服务器单点故障的发生问题，可以通过其与Nginx、Haproxy等反向代理的负载均衡服务器配合实现web服务端的高可用。Keepalived以VRRP协议为实现基础，用VRRP协议来实现高可用性（HA）.VRRP（Virtual Router Redundancy Protocol）协议是用于实现路由器冗余的协议，VRRP协议将两台或多台路由器设备虚拟成一个设备，对外提供虚拟路由器IP（一个或多个）。

2、Keepalived安装

PS:下载地址：<http://www.keepalived.org/download.html>

|  |
| --- |
| //安装所需软件包  yum install -y openssl openssl-devel  //下载  wget http://www.keepalived.org/software/keepalived-1.2.18.tar.gz  //解压、编译、安装  tar -zxvf keepalived-1.2.18.tar.gz -C /usr/local/  cd keepalived-1.2.18/ && ./configure --prefix=/usr/local/keepalived  make && make install  //将keepalived安装成Linux系统服务，因为没有使用keepalived的默认安装路径（默认路径：/usr/local）,安装完成之后，需要做一些修改工作  //首先创建文件夹，将keepalived配置文件进行复制：  mkdir /etc/keepalived  cp /usr/local/keepalived/etc/keepalived/keepalived.conf /etc/keepalived/  //然后复制keepalived脚本文件：  cp /usr/local/keepalived/etc/rc.d/init.d/keepalived /etc/init.d/  cp /usr/local/keepalived/etc/sysconfig/keepalived /etc/sysconfig/  ln -s /usr/local/sbin/keepalived /usr/sbin/  ln -s /usr/local/keepalived/sbin/keepalived /sbin/  如果软链接不对  rm -f /usr/sbin/keepalived  //可以设置开机启动：chkconfig keepalived on，到此我们安装完毕!  chkconfig keepalived on |

3、Keepalived配置

PS:修改keepalived.conf配置文件

|  |
| --- |
| vim /etc/keepalived/keepalived.conf |

PS: 134节点（Master）配置如下

|  |
| --- |
| ! Configuration File for keepalived  global\_defs {  router\_id deatil134 ##标识节点的字符串，通常为hostname  }  vrrp\_script chk\_haproxy {  script "/etc/keepalived/haproxy\_check.sh" ##执行脚本位置  interval 2 ##检测时间间隔  weight -20 ##如果条件成立则权重减20  }  vrrp\_instance VI\_1 {  state MASTER ## 主节点为MASTER，备份节点为BACKUP  interface eth0 ## 绑定虚拟IP的网络接口（网卡），与本机IP地址所在的网络接口相同（我这里是eth0）  virtual\_router\_id 134 ## 虚拟路由ID号（主备节点一定要相同）  mcast\_src\_ip 192.168.0.134 ## 本机ip地址  priority 100 ##优先级配置（0-254的值）  nopreempt  advert\_int 1 ## 组播信息发送间隔，俩个节点必须配置一致，默认1s  authentication { ## 认证匹配  auth\_type PASS  auth\_pass deatil  }  track\_script {  chk\_haproxy  }  virtual\_ipaddress {  192.168.0.136 ## 虚拟ip，可以指定多个  }  } |

PS: 135节点（backup）配置如下

|  |
| --- |
| ! Configuration File for keepalived  global\_defs {  router\_id deatil135 ##标识节点的字符串，通常为hostname  }  vrrp\_script chk\_haproxy {  script "/etc/keepalived/haproxy\_check.sh" ##执行脚本位置  interval 2 ##检测时间间隔  weight -20 ##如果条件成立则权重减20  }  vrrp\_instance VI\_1 {  state BACKUP ## 主节点为MASTER，备份节点为BACKUP  interface eno1613213236 ## 绑定虚拟IP的网络接口（网卡），与本机IP地址所在的网络接口相同（我这里是eno1613213236）  virtual\_router\_id 134 ## 虚拟路由ID号（主备节点一定要相同）  mcast\_src\_ip 192.168.0.135 ## 本机ip地址  priority 90 ##优先级配置（0-254的值）  nopreempt  advert\_int 1 ## 组播信息发送间隔，俩个节点必须配置一致，默认1s  authentication { ## 认证匹配  auth\_type PASS  auth\_pass deatil  }  track\_script {  chk\_haproxy  }  virtual\_ipaddress {  192.168.1.136 ## 虚拟ip，可以指定多个  }  } |

4、执行脚本编写

PS:添加文件位置为/etc/keepalived/haproxy\_check.sh（134、135两个节点文件内容一致即可）

|  |
| --- |
| #!/bin/bash  COUNT=`ps -C haproxy --no-header |wc -l`  if [ $COUNT -eq 0 ];then  /usr/local/haproxy/sbin/haproxy -f /etc/haproxy/haproxy.cfg  sleep 2  if [ `ps -C haproxy --no-header |wc -l` -eq 0 ];then  killall keepalived  fi  fi |

5、执行脚本赋权

PS:haproxy\_check.sh脚本授权,赋予可执行权限.

|  |
| --- |
| chmod +x /etc/keepalived/haproxy\_check.sh |

6、启动keepalived

PS:当我们启动俩个haproxy节点以后，我们可以启动keepalived服务程序：

|  |
| --- |
| //启动两台机器的keepalived  service keepalived start | stop | status | restart  //查看状态  ps -ef | grep haproxy  ps -ef | grep keepalived |

7、高可用测试

PS:vip在134节点上

PS:134节点宕机测试：停掉134的keepalived服务即可。

PS:查看135节点状态：我们发现VIP漂移到了135节点上，那么135节点的haproxy可以继续对外提供服务！

2.3.2.9、集群配置文件

创建如下配置文件位于：/etc/rabbitmq目录下（这个目录需要自己创建）

环境变量配置文件：rabbitmq-env.conf

配置信息配置文件：rabbitmq.config（可以不创建和配置，修改）

rabbitmq-env.conf配置文件：

---------------------------------------关键参数配置-------------------------------------------

RABBITMQ\_NODE\_IP\_ADDRESS=本机IP地址

RABBITMQ\_NODE\_PORT=5672

RABBITMQ\_LOG\_BASE=/var/lib/rabbitmq/log

RABBITMQ\_MNESIA\_BASE=/var/lib/rabbitmq/mnesia

配置参考参数如下：

RABBITMQ\_NODENAME=FZTEC-240088 节点名称

RABBITMQ\_NODE\_IP\_ADDRESS=1134.0.0.1 监听IP

RABBITMQ\_NODE\_PORT=5672 监听端口

RABBITMQ\_LOG\_BASE=/data/rabbitmq/log 日志目录

RABBITMQ\_PLUGINS\_DIR=/data/rabbitmq/plugins 插件目录

RABBITMQ\_MNESIA\_BASE=/data/rabbitmq/mnesia 后端存储目录

更详细的配置参见： [http://www.rabbitmq.com/configure.html#configuration-file](http://www.rabbitmq.com/configure.html" \l "configuration-file)

配置文件信息修改：

/usr/lib/rabbitmq/lib/rabbitmq\_server-3.6.4/ebin/rabbit.app和rabbitmq.config配置文件配置任意一个即可，我们进行配置如下：

vim /usr/lib/rabbitmq/lib/rabbitmq\_server-3.6.4/ebin/rabbit.app

-------------------------------------关键参数配置----------------------------------------

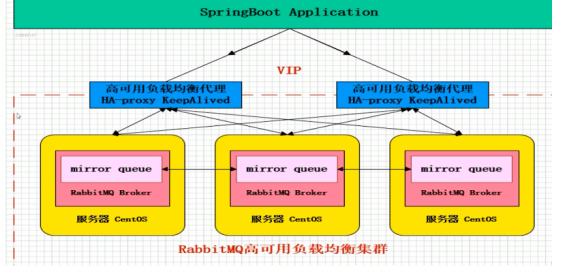
tcp\_listerners 设置rabbimq的监听端口，默认为[5672]。  
disk\_free\_limit 磁盘低水位线，若磁盘容量低于指定值则停止接收数据，默认值为{mem\_relative, 1.0},即与内存相关联1：1，也可定制为多少byte.  
vm\_memory\_high\_watermark，设置内存低水位线，若低于该水位线，则开启流控机制，默认值是0.4，即内存总量的40%。  
hipe\_compile 将部分rabbimq代码用High Performance Erlang compiler编译，可提升性能，该参数是实验性，若出现erlang vm segfaults，应关掉。  
force\_fine\_statistics， 该参数属于rabbimq\_management，若为true则进行精细化的统计，但会影响性能

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

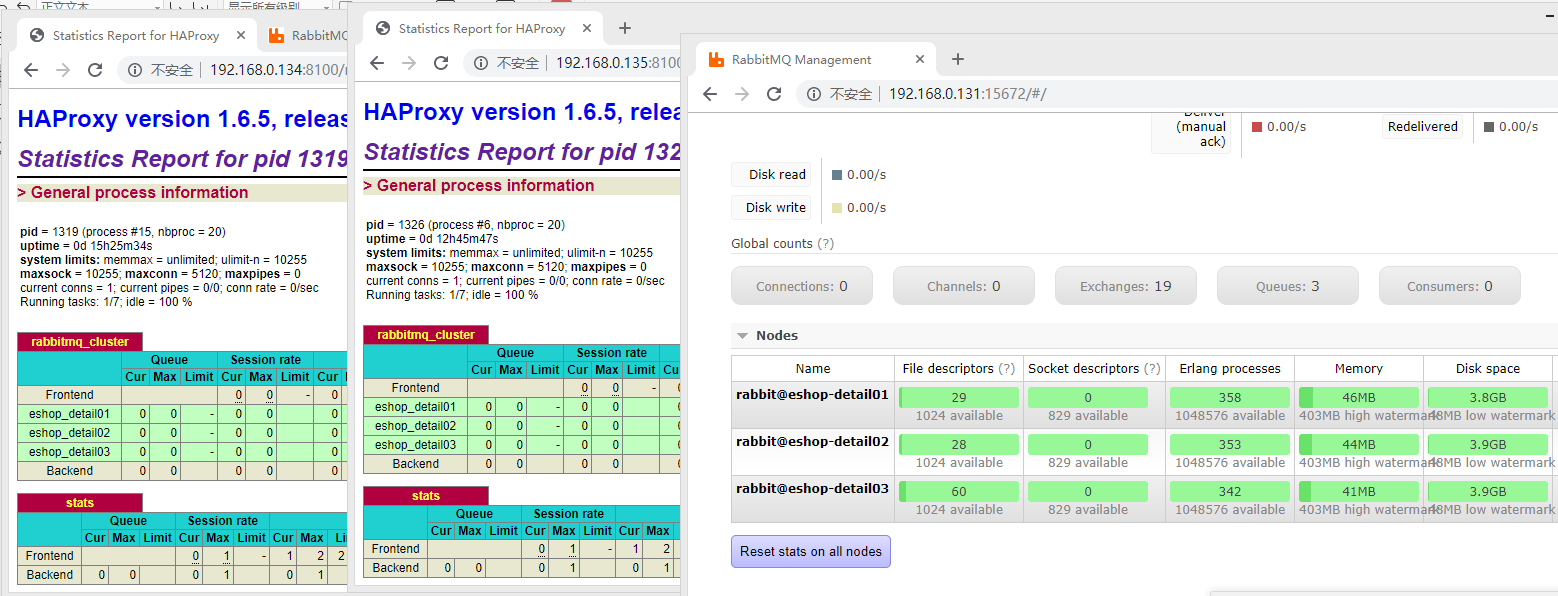
更详细的配置参见：http://www.rabbitmq.com/configure.html

### 2.3.3集群效果输出

2.3.3.1、RabbitMQ集群架构

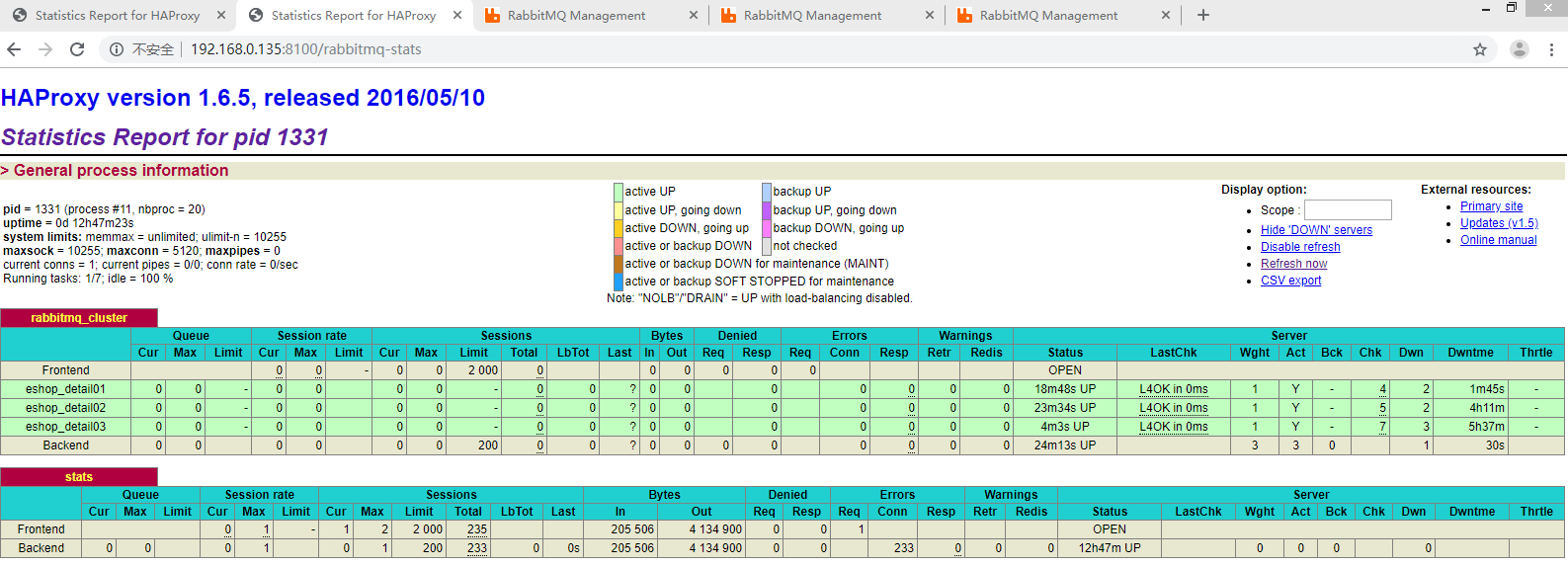


2.3.3.2、MQ集群示意图

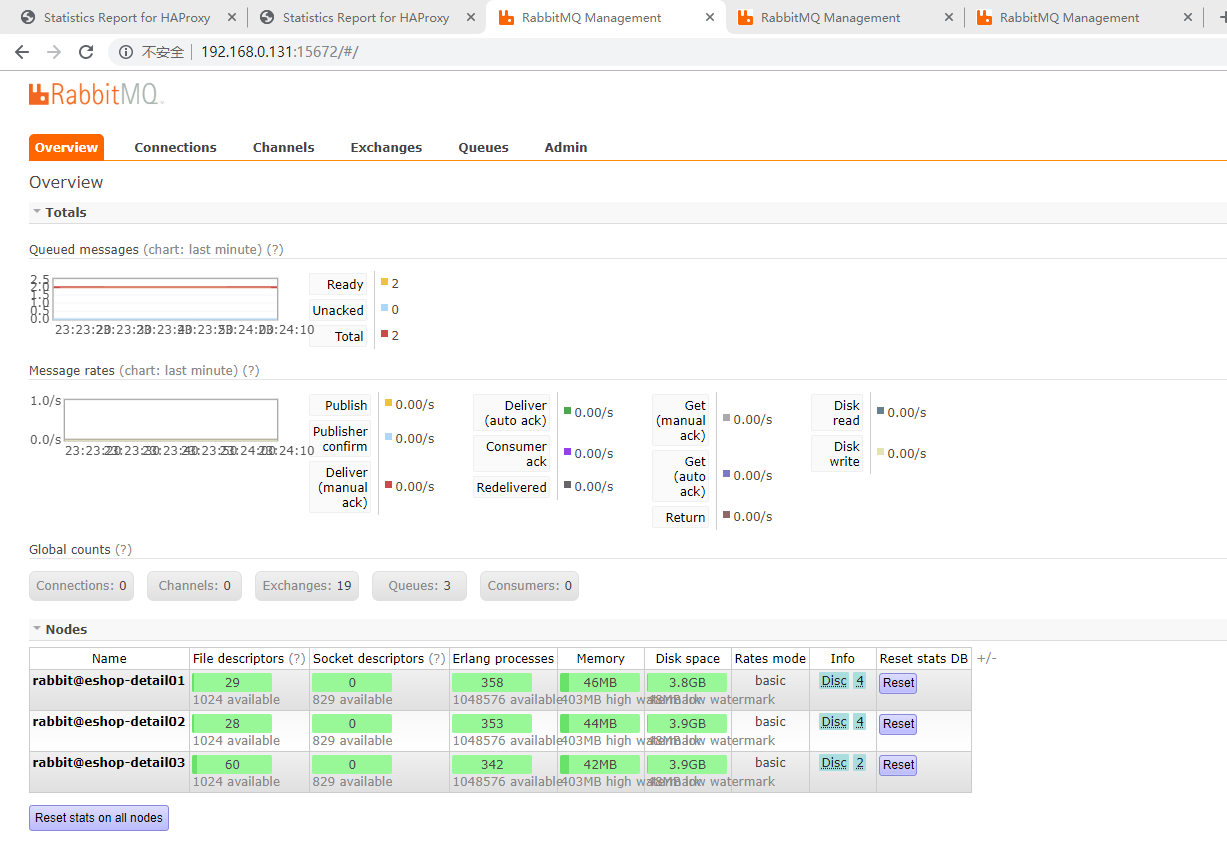


2.3.3.3、HAProxy示意图

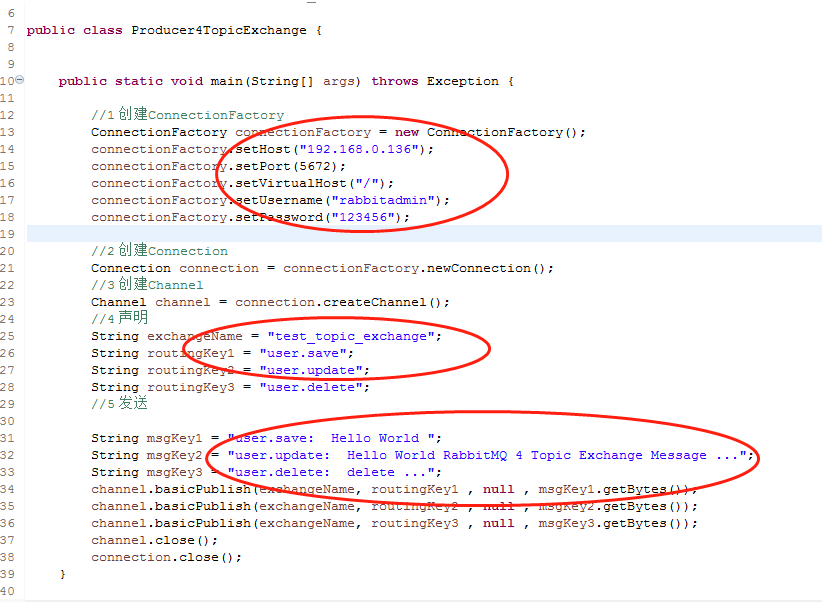
   HAProxy提供高可用性、负载均衡以及基于TCP和HTTP应用的代理，支持虚拟主机，它是免费、快速并且可靠的一种解决方案。HAProxy特别适用于那些负载特大的web站点，这些站点通常又需要会话保持或七层处理



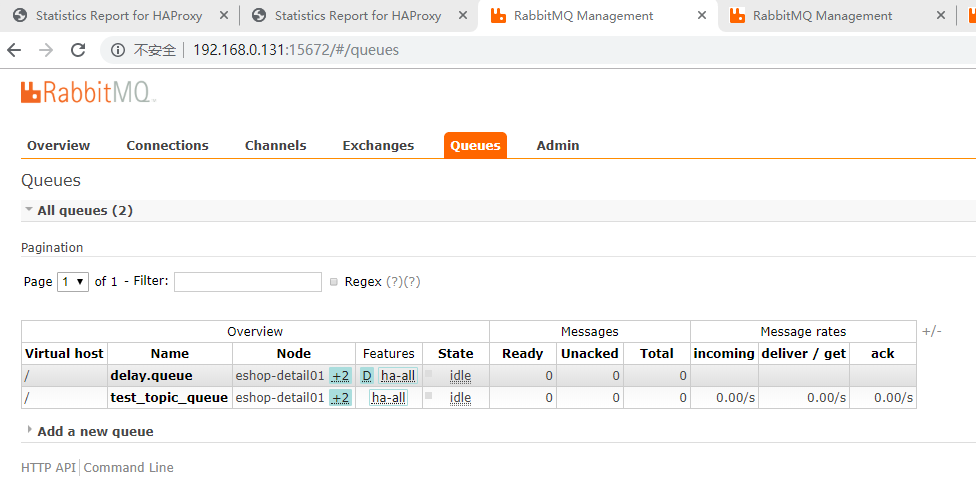
2.3.3.4、rabbitMQ示意图



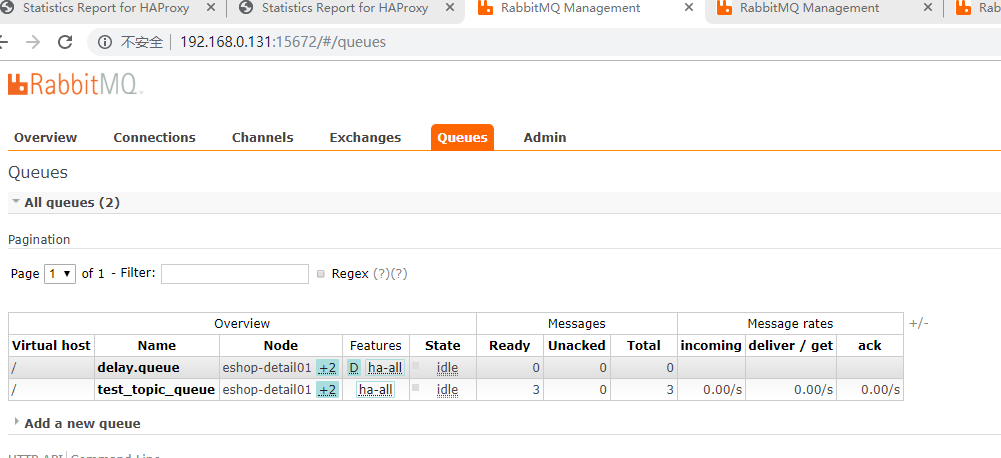
2.3.3.5代码生产者发送消息例子



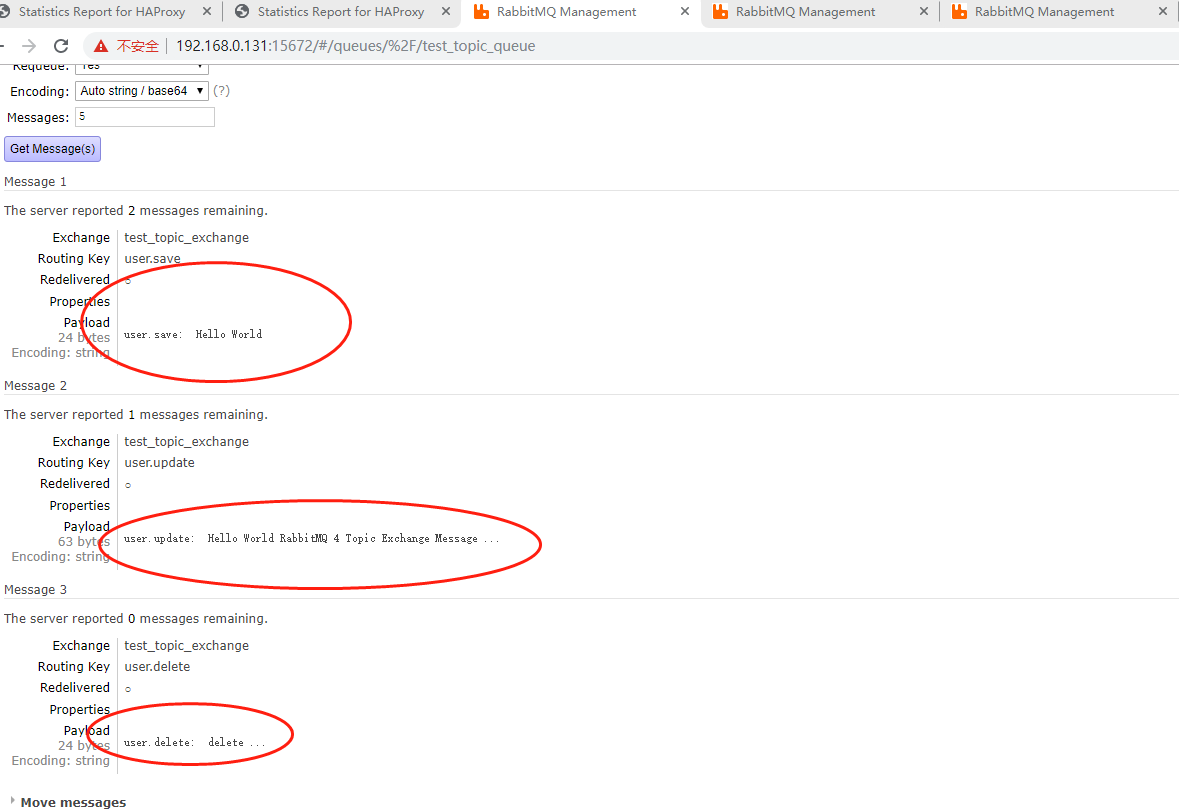
2.3.3.6消息发送队列前



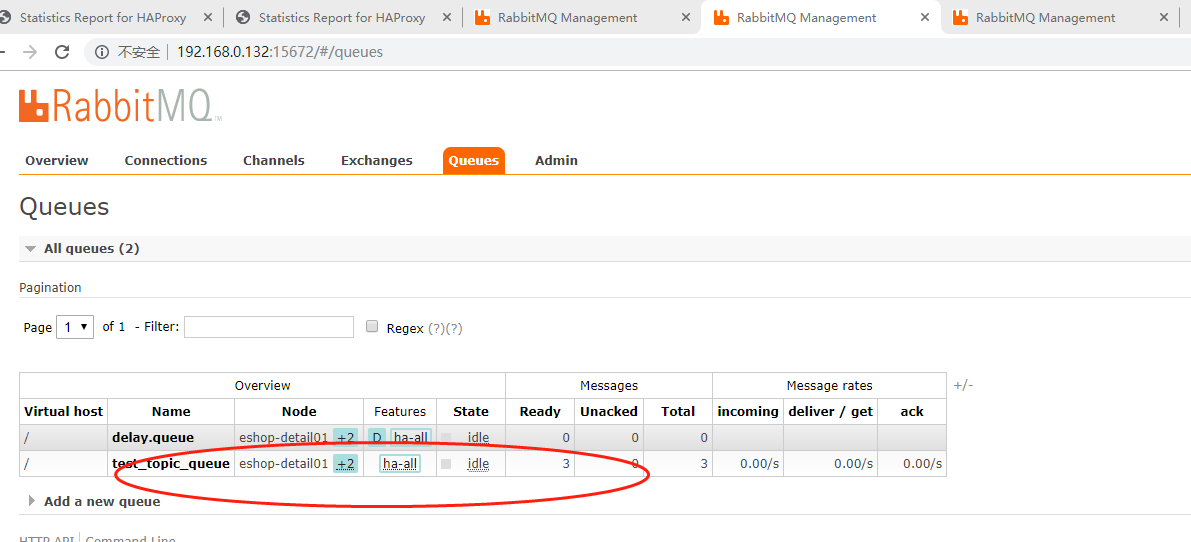
2.3.3.7消息发送队列后

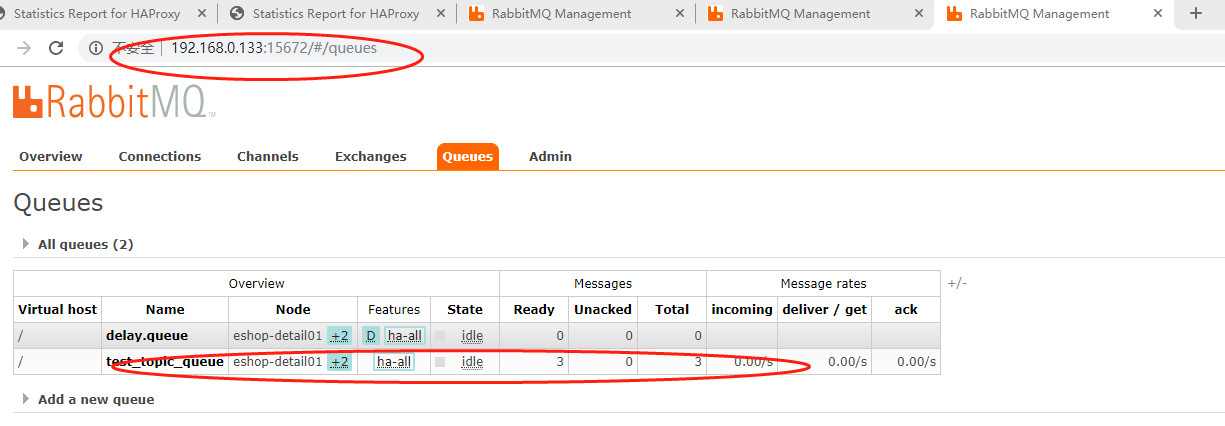


2.3.3.8数据与代码发送消息内容一致

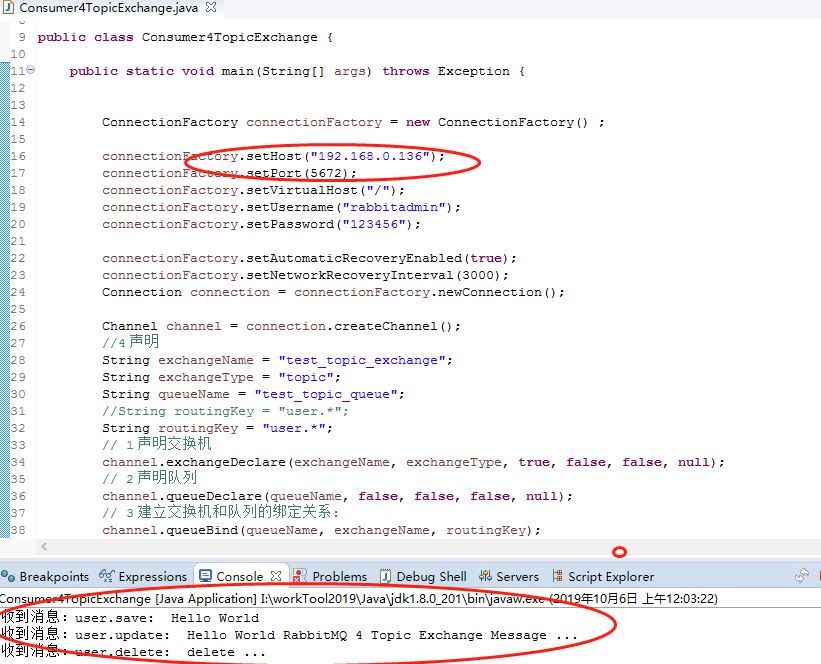


2.3.3.9另外两个rabbitmq数据一致

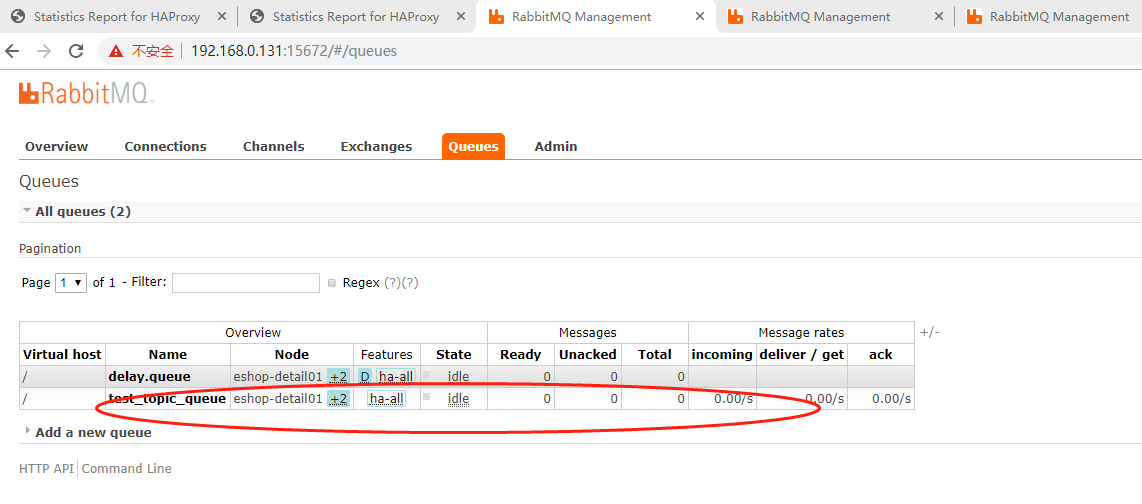




2.3.3.10、消费端接收消息



2.3.3.11、mq被用户消费调页面



2.4、RSA和AES加密混合加密流程算法

2.4.1 RSA和AES加密混合流程简述



一般情况下，RSA时需要和AES结合使用的。

客户端使用RSA + AES对重要信息进行加密

客户端加密过程主要分为以下三个步骤：

1、客户端获取AES的密钥；

2、对身份证信息（重要信息）进行AES加密；

3、通过使用RSA对AES密钥进行公钥加密。

服务端使用RSA + AES对重要信息进行解密

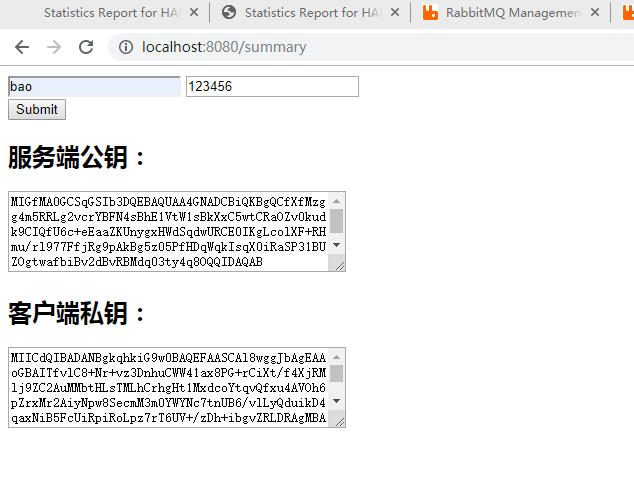
服务端解密过程主要分为以下两个步骤：

1、获取AES秘钥

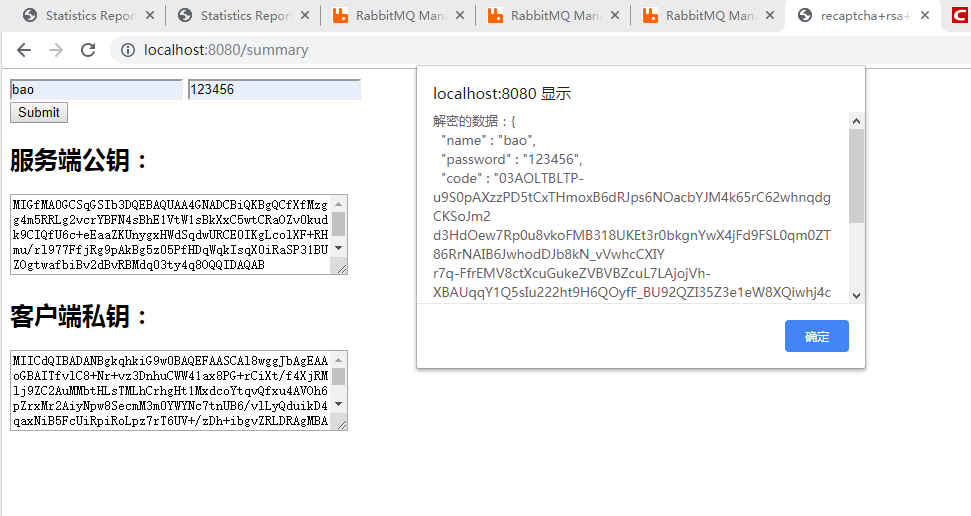
2、 对加密后的AES密钥进行RSA私钥解密，拿到密钥原文；

3、 对加密后的重要信息进行AES解密，拿到原始内容。

2.4.2模拟用户信息提交页面request信息进行aes 、rsa加密

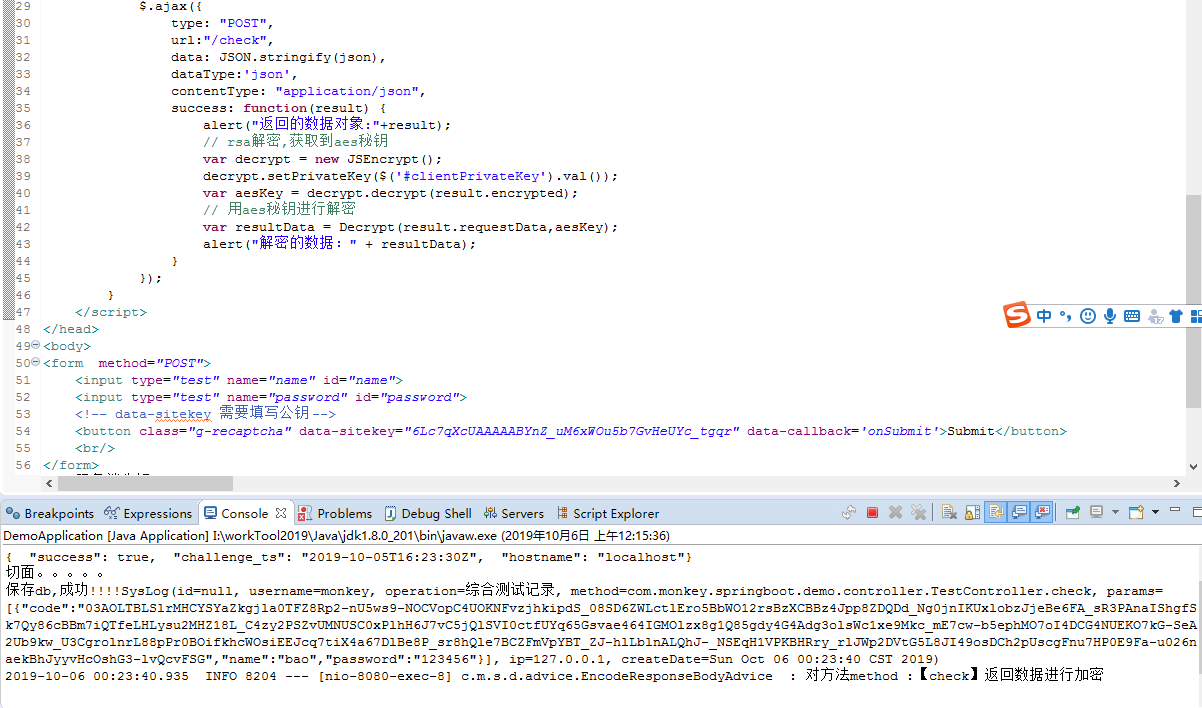


2.4.3模拟用户信息提交页面reponse信息进行aes 、rsa解密



2.4.4代码如下





2.5、zxbbix部署

2.5.1[使用docker部署，zabbix](https://www.cnblogs.com/root0/p/9812789.html)官方地址

官方写的很详细并且是中文的,一步步按照操作就可以

<https://www.zabbix.com/documentation/3.4/zh/manual/installation/containers>

2.5.2 启动一个空的Mysql服务器实例

docker run --name mysql-server -t \

-e MYSQL\_DATABASE="zabbix" \

-e MYSQL\_USER="zabbix" \

-e MYSQL\_PASSWORD="zabbix" \

-e MYSQL\_ROOT\_PASSWORD="zabbix" \

-d mysql:5.7 \--character-set-server=utf8 --collation-server=utf8\_bin

2.5.3 启动Zabbix server实例，并关联这个实例到已创建的MySQL服务器实例

docker run --name zabbix-server-mysql -t \

-e DB\_SERVER\_HOST="mysql-server" \

-e MYSQL\_DATABASE="zabbix" \

-e MYSQL\_USER="zabbix" \

-e MYSQL\_PASSWORD="zabbix" \

-e MYSQL\_ROOT\_PASSWORD="zabbix" \

--link mysql-server:mysql \

-p 10051:10051 \

-d zabbix/zabbix-server-mysql:latest

2.5.4 启动Zabbix web 接口，并将它与MySQL服务器实例和Zabbix server实例关联

docker run --name zabbix-web-nginx-mysql -t \

-e DB\_SERVER\_HOST="mysql-server" \

-e MYSQL\_DATABASE="zabbix" \

-e MYSQL\_USER="zabbix" \

-e MYSQL\_PASSWORD="zabbix" \

-e MYSQL\_ROOT\_PASSWORD="zabbix" \

--link mysql-server:mysql \

--link zabbix-server-mysql:zabbix-server \

-p 8080:80 \

-d zabbix/zabbix-web-nginx-mysql:latest

2.5.5 查看容器启动情况

[root@zabbix\_server ~]# docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

98cbe8d8a6bd zabbix/zabbix-web-nginx-mysql:latest "docker-entrypoint.sh" 6 seconds ago Up 5 seconds 443/tcp, 0.0.0.0:8080->80/tcp zabbix-web-nginx-mysql

de040d43d60f zabbix/zabbix-server-mysql:latest "docker-entrypoint.sh" 59 seconds ago Up 59 seconds 0.0.0.0:10051->10051/tcp zabbix-server-mysql

3276f18def8d mysql:5.7 "docker-entrypoint.s…" About a minute ago Up About a minute 3306/tcp mysql-server

2.5.6 查看docker镜像

[root@zabbix\_server ~]# docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

zabbix/zabbix-web-nginx-mysql latest 4db891b4393a 10 hours ago 177MB

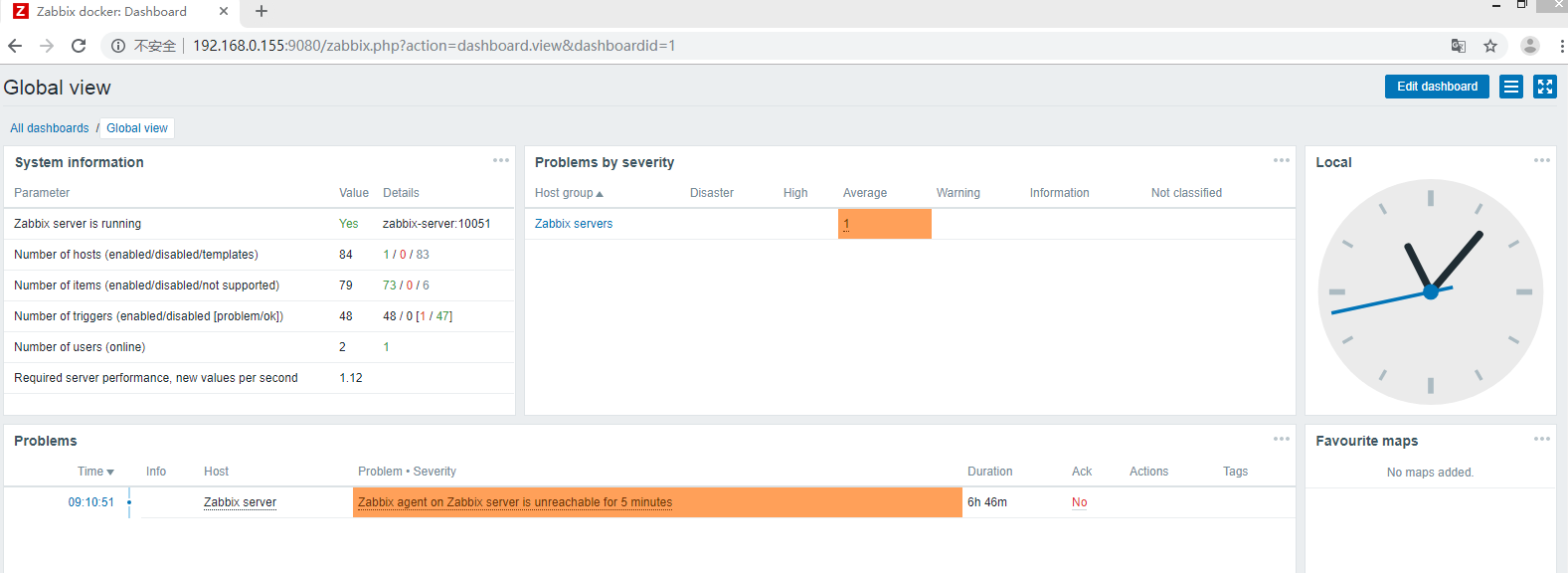
zabbix/zabbix-server-mysql latest f5e58dafe9ac 10 hours ago 62.2MB

mysql 5.7 f0f3956a9dd8 7 days ago 409MB

2.5.7 访问web界面

http://192.168.0.155:9080

帐号密码: Admin zabbix

已经可以访问了