第三节课:Rabbitmq镜像队列搭建 开发应用场景

一:集群节点安装

1.1)**集群节点安装（老师的是**192.168.159.86;192.168.159.87;192.168.159.88)

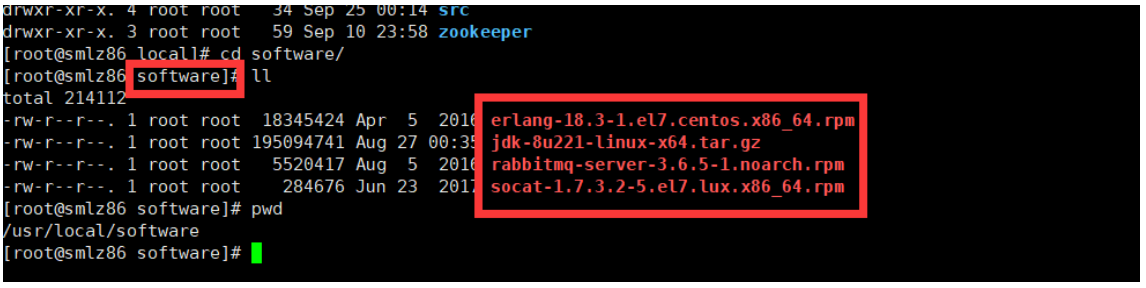
在这里老师是以我的86**服务器单台演示**

①:安装rabbitmq所需要的依赖包

yum install build-essential openssl openssl-devel unixODBC unixODBC-devel make gcc gcc-c++ kernel-devel m4 ncurses-devel tk tc xz

②:下载安装包 (PS:老师的下载包的目录是　cd /usr/local/software)

```
wget www.rabbitmq.com/releases/erlang/erlang-18.3-1.el7.centos.x86_64.rpm
wget http://repo.iotti.biz/CentOS/7/x86_64/socat-1.7.3.2-5.el7.lux.x86_64.rpm
wget www.rabbitmq.com/releases/rabbitmq-server/v3.6.5/rabbitmq-server-3.6.5-1.noarch.rpm
```



③:安装服务命令

```
#第一步:安装erlang语言环境
rpm -ivh erlang-18.3-1.el7.centos.x86_64.rpm
#第二步:安装socat加解密软件
rpm -ivh socat-1.7.3.2-5.el7.lux.x86_64.rpm
#第三步:最后安装rabbitmq
rpm -ivh rabbitmq-server-3.6.5-1.noarch.rpm
```

④:修改集群用户与连接心跳检测

注意修改vim /usr/lib/rabbitmq/lib/rabbitmq_server-3.6.5/ebin/rabbit.app**文件**

修改：loopback_users 中的 <<"guest">>,只保留guest（不修改只能通过localhost访问)

```
        {channel_max, 0},
        {heartbeat, 60},
        {msg_store_file_size_limit, 16777216},
        {fhc_write_buffering, true},
        {fhc_read_buffering, false},
        {queue_index_max_journal_entries, 32768},
        {queue_index_embed_msgs_below, 4096},
        {default_user, <<"guest">>},
        {default_pass, <<"guest">>},
        {default_user_tags, [administrator]},
        {default_vhost, <<"/">>},
        {default_permissions, [<<".*">>, <<".*">>, <<".*">>]},
        {loopback_users, [guest]},
        {password_hashing_module, rabbit_password_hashing_sha256},
        {cluster_nodes, {[], disc}},
        {server_properties, []},
        {collect_statistics, none},
        {collect_statistics_interval, 5000},
        {mnesia_table_loading_timeout, 30000},
        {auth_mechanisms, ['PLAIN', 'AMQPLAIN']},
        {auth_backends, [rabbit_auth_backend_internal]},
        {delegate_count, 16},
        {trace_vhosts, []},
        {log_levels, [{connection, info}]},
        {ssl_cert_login_from, distinguished_name},
        {ssl_handshake_timeout, 5000},
```

⑤:修改 本机系统文件

a:修改 vim /etc/rabbitmq/rabbitmq-env.conf

添加: NODENAME=rabbit

```
        ]}]}.
[root@smlz86 software]# cat /etc/rabbitmq/rabbitmq-env.conf
NODENAME=rabbit
[root@smlz86 software]# ^C
```

b:修改 vim /etc/hostname

老师的是smlz86 你自己的根据情况来

```
[root@smlz86 software]# cat /etc/hostname
smlz86
[root@smlz86 software]#
```

c:修改本地 vim /etc/hosts文件

```
[root@smlz86 software]# cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.159.86   smlz86
192.168.159.87   smlz87
192.168.159.88   smlz88
[root@smlz86 software]#
```

⑥:验证单台服务器是可用的

进入到/usr/local**目录下** cd /usr/local**目录下**

输入 rabbitmqctl start_app

启动插件: rabbitmq-plugins enable rabbitmq_management

```
[root@smlz86 local]# rabbitmqctl start_app
Starting node rabbit@smlz86 ...
[root@smlz86 local]#
```

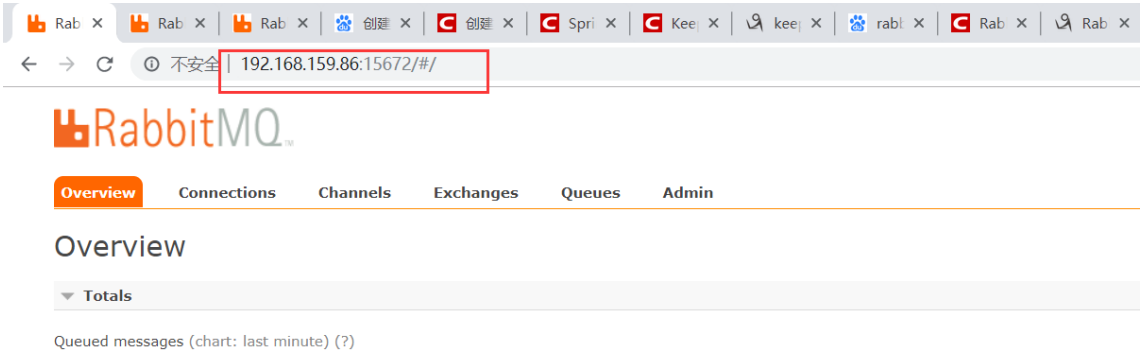
检查端口: lsof -i:5672

```
[root@smlz86 local]# rabbitmqctl start_app
Starting node rabbit@smlz86 ...
[root@smlz86 local]# lsof -i:5672
COMMAND  PID     USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
beam    9807 rabbitmq   50u  IPv6 145303      0t0  TCP *:amqp (LISTEN)
[root@smlz86 local]#
```

通过 ps -ef|grep rabbitmq

```
COMMAND  PID     USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
beam    9807 rabbitmq   50u  IPv6 145303      0t0  TCP *:amqp (LISTEN)
[root@smlz86 local]# ps -ef|grep rabbitmq
rabbitmq   9769      1  0 01:26 ?        00:00:00 /usr/lib64/erlang/erts-7.3/bin/epmd -daemon
rabbitmq   9807      1  0 01:26 ?        00:02:19 /usr/lib64/erlang/erts-7.3/bin/beam -W w -A 64 -P 1048576 -t 5000000 -stbt db -K true -- -root
a /usr/lib/rabbitmq/lib/rabbitmq_server-3.6.5/ebin -noshell -noinput -s rabbit boot -sname rabbit -boot start_sasl -kernel inet_default_connect_
_logger false -rabbit error_logger {file,"/var/log/rabbitmq/rabbit.log"} -rabbit sasl_error_logger {file,"/var/log/rabbitmq/rabbit-sasl.log"} -r
t plugins_dir "/usr/lib/rabbitmq/lib/rabbitmq_server-3.6.5/plugins" -rabbit plugins_expand_dir "/var/lib/rabbitmq/mnesia/rabbit-plugins-expand"
tart_memsup false -mnesia dir "/var/lib/rabbitmq/mnesia/rabbit" -kernel inet_dist_listen_min 25672 -kernel inet_dist_listen_max 25672 -noshell -
rabbitmq   9878   9807  0 01:26 ?        00:00:00 inet_gethost 4
rabbitmq   9879   9878  0 01:26 ?        00:00:01 inet_gethost 4
root      20348  14835  0 07:17 pts/1    00:00:00 grep --color=auto rabbitmq
[root@smlz86 local]#
```

访问地址:http://192.168.159.86:15672



⑦:然后87,88节点按照以上的步骤进行想同的操作

ps:**等87,88服务都成功安装了服务后，现在我们三台服务器都安装成功了**

我们需要从86 87 88**节点中选择一个主节点**(master) **老师选择的是**86**服务器**

也就是说我们需要把86的Cookie文件同步到87、88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777(chmod 777 /var/lib/rabbitmq/.erlang.cookie)，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。

copy 86的.erlang.cookie文件到 87 88上

scp /var/lib/rabbitmq/.erlang.cookie 192.168.159.87:/var/lib/rabbitmq

scp /var/lib/rabbitmq/.erlang.cookie 192.168.159.88:/var/lib/rabbitmq

⑧:启用集群命令 同样进入 /usr/local**目录下**

a:**在**86,87,88**机器上 先执行停止命令**

rabbitmqctl stop

b:**然后在三台服务器上**86,87,88**上执行 下面的命令(启动集群命令)**

rabbitmq-server -detached

c:**切换到87的机器上执行下面三条命令（同样目录都是再/usr/local下执行的）**

rabbitmqctl stop_app

rabbitmqctl join_cluster rabbit@smlz86

rabbitmqctl start_app

D:**切换到88的机器上执行下面三条命令（同样目录都是再/usr/local下执行的）**

rabbitmqctl stop_app

rabbitmqctl join_cluster rabbit@smlz86

rabbitmqctl start_app

E:**修改集群名称: 在86上执行给命令**(/usr/local)

rabbitmqctl set_cluster_name rabbitmq_cluster_smlz

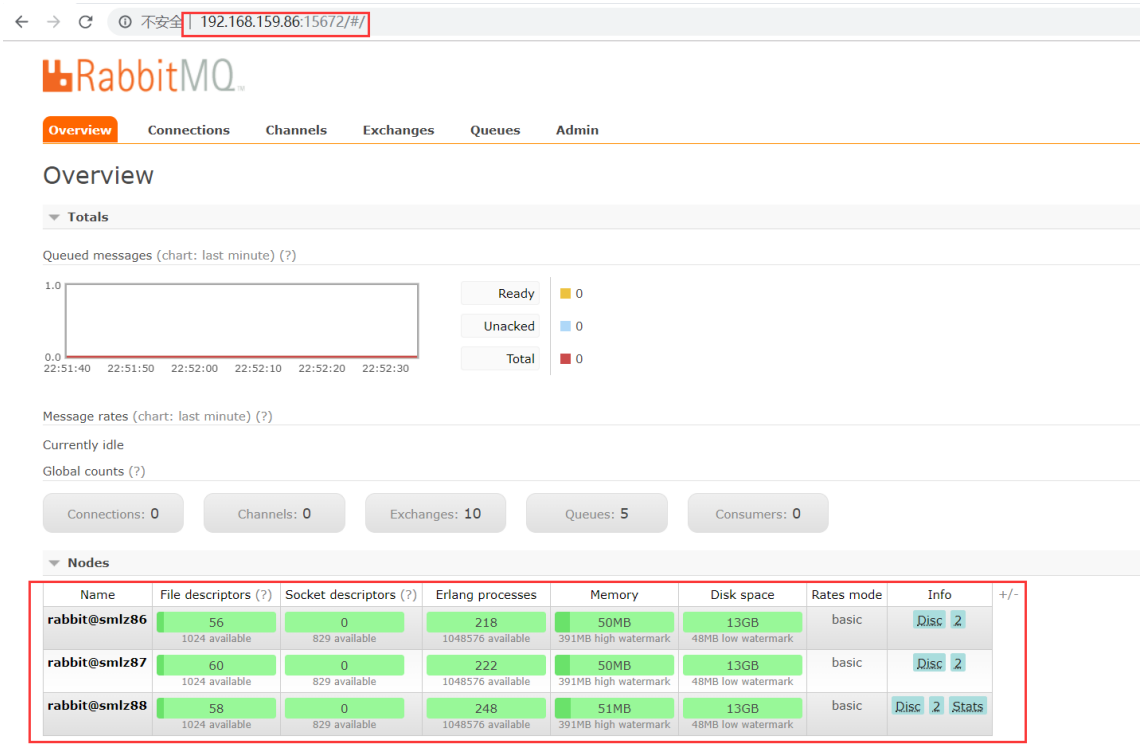
F:**查看集群状态**

rabbitmqctl cluster_status

```
[root@smlz86 local]# rabbitmqctl cluster_status
Cluster status of node rabbit@smlz86 ...
[{nodes,[{disc,[rabbit@smlz86,rabbit@smlz87,rabbit@smlz88]}]},
 {running_nodes,[rabbit@smlz87,rabbit@smlz88,rabbit@smlz86]},
 {cluster_name,<<"smlz_rmq_cluster">>},
 {partitions,[]},
 {alarms,[{rabbit@smlz87,[]},{rabbit@smlz88,[]},{rabbit@smlz86,[]}]}]
[root@smlz86 local]#
```

G:**三台服务随意访问地址**

访问任意一个管控台节点：http://192.168.159.86:15672　这里我访问的是86节点



H:**配置镜像队列**

在任意节点上执行 rabbitmqctl set_policy ha-all "^" '{"ha-mode":"all"}'

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

剔除节点命令

PS:rabbitmqctl forget_cluster_node rabbit@**节点名称(比如**smlz87 smlz88) **剔除节点**

二:安装 HaProxy（192.168.159.89)

2.1)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

2.2)Haproxy**安装**

1:**下载依赖包**

yum install gcc vim wget

2:)**下载**haproxy

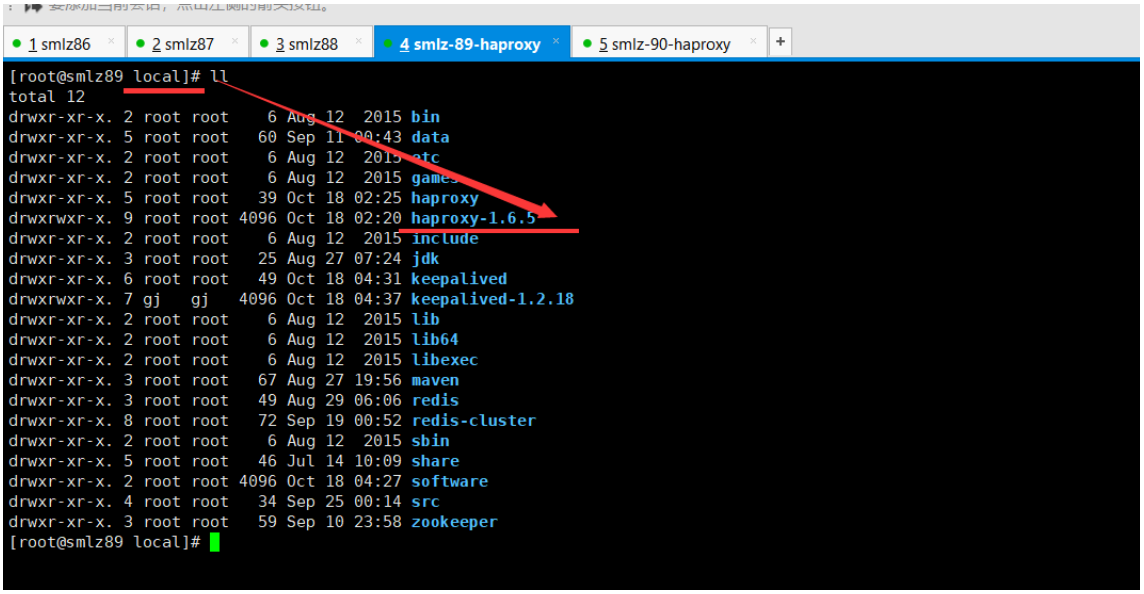
wget http://www.haproxy.org/download/1.6/src/haproxy-1.6.5.tar.gz（**貌似国内访问不了这个连接**）**老师会提供这个安装包**

```
[root@smlz89 local]# cd software/
[root@smlz89 software]# ll
total 215964
-rw-r--r--. 1 root root  18345424 Apr  5  2016 erlang-18.3-1.el7.centos.x86_64.rpm
-rw-r--r--. 1 root root   1563272 Oct 18 01:58 haproxy-1.6.5.tar.gz
-rw-r--r--. 1 root root 195094741 Aug 27 00:35 jdk-8u221-linux-x64.tar.gz
-rw-r--r--. 1 root root    330361 Dec  1  2016 keepalived-1.2.18.tar.gz
-rw-r--r--. 1 root root   5520417 Aug  5  2016 rabbitmq-server-3.6.5-1.noarch.rpm
-rw-r--r--. 1 root root    284676 Jun 23  2017 socat-1.7.3.2-5.el7.lux.x86_64.rpm
[root@smlz89 software]# pwd
/usr/local/software
[root@smlz89 software]#
```

3)cd /usr/local/software **进行解压**

tar -zxvf haproxy-1.6.5.tar.gz -C /usr/local

解压后会在 /usr/local **中生成一个文件** haproxy-1.6.5**文件夹**



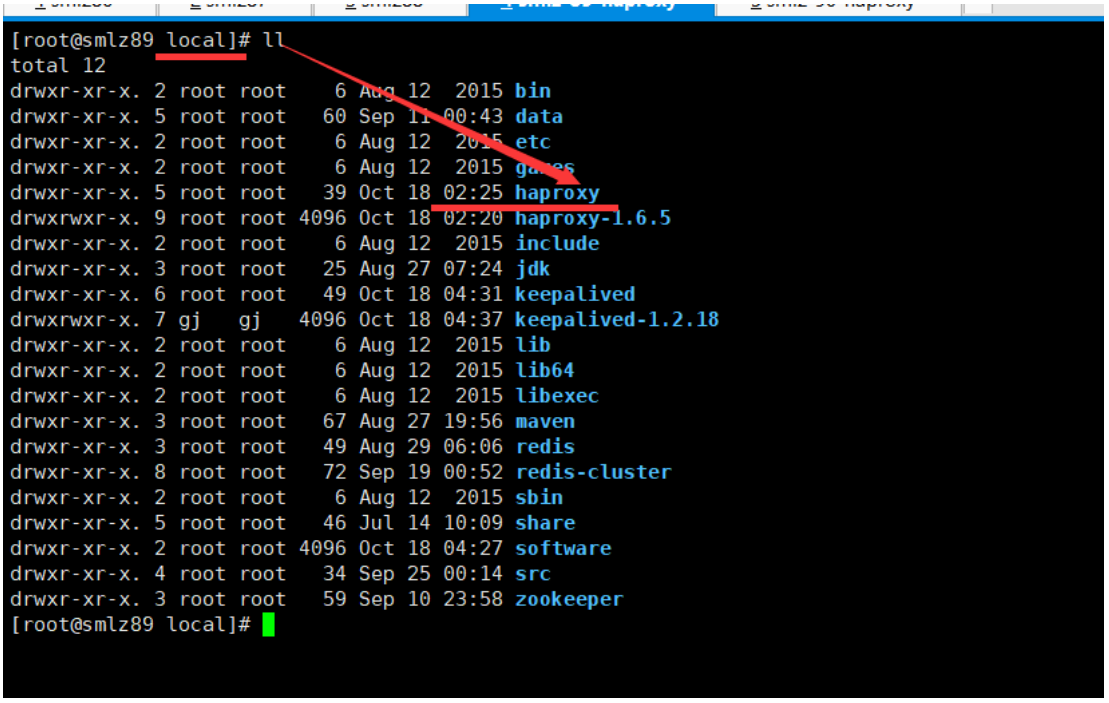
4)**进入目录、进行编译、安装**

4.1)cd /usr/local/haproxy-1.6.5 进入解压目录

4.2)编译 make TARGET=linux31 PREFIX=/usr/local/haproxy

4.3) 安装: make install PREFIX=/usr/local/haproxy

安装目录:



4.4)**创建 一个**haproxy**的目录 用于存放**haproxy**的配置文件**:

创建目录: mkdir /etc/haproxy

赋权:

groupadd -r -g 149 haproxy

useradd -g haproxy -r -s /sbin/nologin -u 149 haproxy

创建配置文件

touch /etc/haproxy/haproxy.cfg

5:)haproxy.cfg**配置文件详解**

```
#lgging options
global
        log 127.0.0.1 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
    #监听的端口
        bind 0.0.0.0:5672
        #配置TCP模式
        mode tcp
        #简单的轮询
        balance roundrobin
        ##==========================================超级 超级 重要防止应用程序连接关闭==============
        timeout client 3h
        timeout server 3h
        #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
        server smlz86 192.168.159.86:5672 check inter 5000 rise 2 fall 2
        server smlz87 192.168.159.87:5672 check inter 5000 rise 2 fall 2
        server smlz88 192.168.159.88:5672 check inter 5000 rise 2 fall 2
#配置haproxy web监控，查看统计信息
listen stats
        bind 192.168.159.89:8100
        mode http
        option httplog
        stats enable
        #设置haproxy监控地址为http://192.168.159.89:8100/rabbitmq-stats
        stats uri /rabbitmq-stats
        stats refresh 5s
```

6)**启动**haproxy

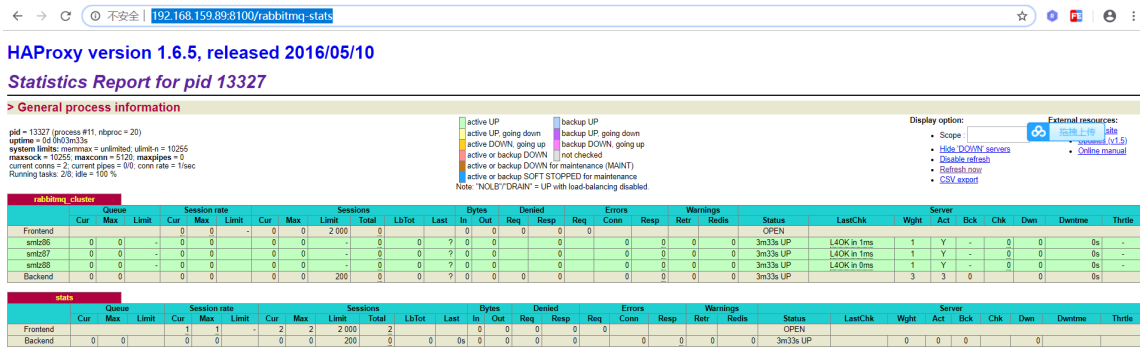
/usr/local/haproxy/sbin/haproxy -f /etc/haproxy/haproxy.cfg

7）**查看**haproxy**是否启动**?　ps-ef|grep rabbitmq

```
[root@smlz89 local]# ps -ef|grep haproxy
nobody    13317      1  0 00:18 ?        00:00:00 /usr/local/haproxy/sbin/haproxy -f /etc/haproxy/haproxy.cfg
nobody    13318      1  0 00:18 ?        00:00:00 /usr/local/haproxy/sbin/haproxy -f /etc/haproxy/haproxy.cfg
nobody    13319      1  0 00:18 ?        00:00:00 /usr/local/haproxy/sbin/haproxy -f /etc/haproxy/haproxy.cfg
nobody    13320      1  0 00:18 ?        00:00:00 /usr/local/haproxy/sbin/haproxy -f /etc/haproxy/haproxy.cfg
nobody    13321      1  0 00:18 ?        00:00:00 /usr/local/haproxy/sbin/haproxy -f /etc/haproxy/haproxy.cfg
nobody    13322      1  0 00:18 ?        00:00:00 /usr/local/haproxy/sbin/haproxy -f /etc/haproxy/haproxy.cfg
nobody    13323      1  0 00:18 ?        00:00:00 /usr/local/haproxy/sbin/haproxy -f /etc/haproxy/haproxy.cfg
nobody    13324      1  0 00:18 ?        00:00:00 /usr/local/haproxy/sbin/haproxy -f /etc/haproxy/haproxy.cfg
nobody    13325      1  0 00:18 ?        00:00:00 /usr/local/haproxy/sbin/haproxy -f /etc/haproxy/haproxy.cfg
nobody    13326      1  0 00:18 ?        00:00:00 /usr/local/haproxy/sbin/haproxy -f /etc/haproxy/haproxy.cfg
nobody    13327      1  0 00:18 ?        00:00:00 /usr/local/haproxy/sbin/haproxy -f /etc/haproxy/haproxy.cfg
nobody    13328      1  0 00:18 ?        00:00:00 /usr/local/haproxy/sbin/haproxy -f /etc/haproxy/haproxy.cfg
nobody    13329      1  0 00:18 ?        00:00:00 /usr/local/haproxy/sbin/haproxy -f /etc/haproxy/haproxy.cfg
nobody    13330      1  0 00:18 ?        00:00:00 /usr/local/haproxy/sbin/haproxy -f /etc/haproxy/haproxy.cfg
nobody    13331      1  0 00:18 ?        00:00:00 /usr/local/haproxy/sbin/haproxy -f /etc/haproxy/haproxy.cfg
nobody    13332      1  0 00:18 ?        00:00:00 /usr/local/haproxy/sbin/haproxy -f /etc/haproxy/haproxy.cfg
nobody    13333      1  0 00:18 ?        00:00:00 /usr/local/haproxy/sbin/haproxy -f /etc/haproxy/haproxy.cfg
nobody    13334      1  0 00:18 ?        00:00:00 /usr/local/haproxy/sbin/haproxy -f /etc/haproxy/haproxy.cfg
nobody    13335      1  0 00:18 ?        00:00:00 /usr/local/haproxy/sbin/haproxy -f /etc/haproxy/haproxy.cfg
nobody    13336      1  0 00:18 ?        00:00:00 /usr/local/haproxy/sbin/haproxy -f /etc/haproxy/haproxy.cfg
root      13433   3479  0 00:20 pts/0    00:00:00 grep --color=auto haproxy
[root@smlz89 local]#
```

haparoxy**的监控台访问**

http://192.168.159.89:8100/rabbitmq-stats



关闭haproxy命令

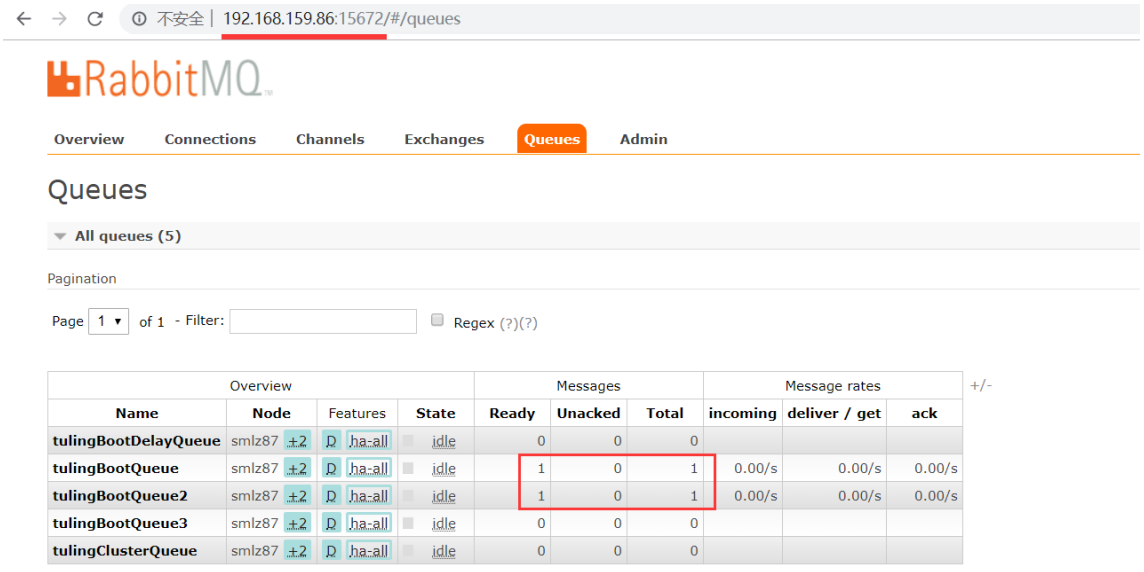
killall haproxy

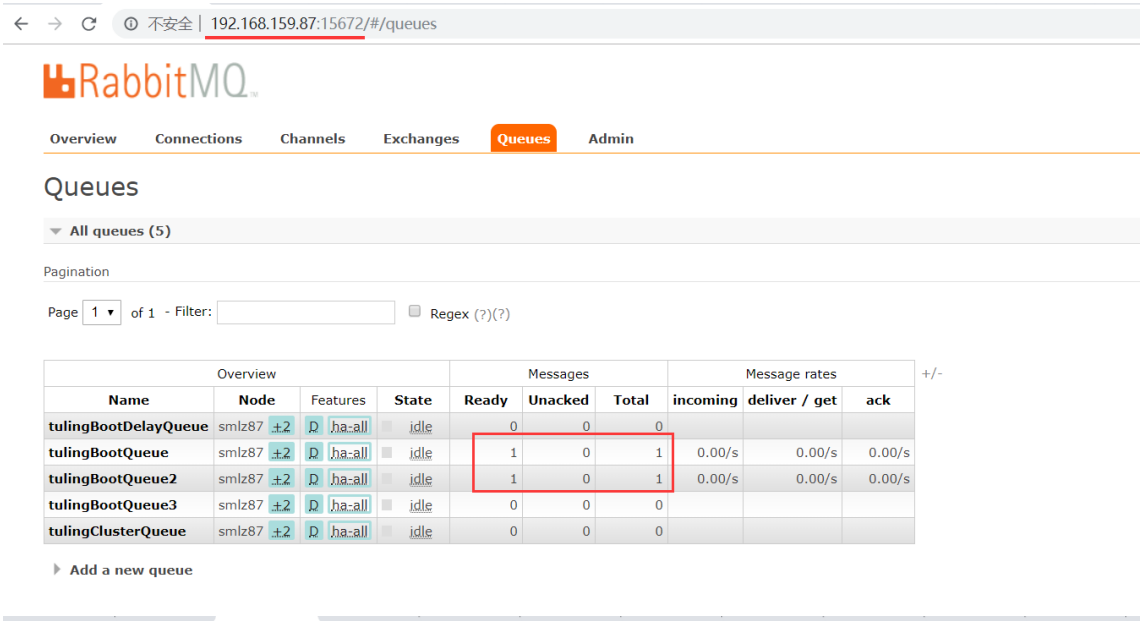
三:集群测试

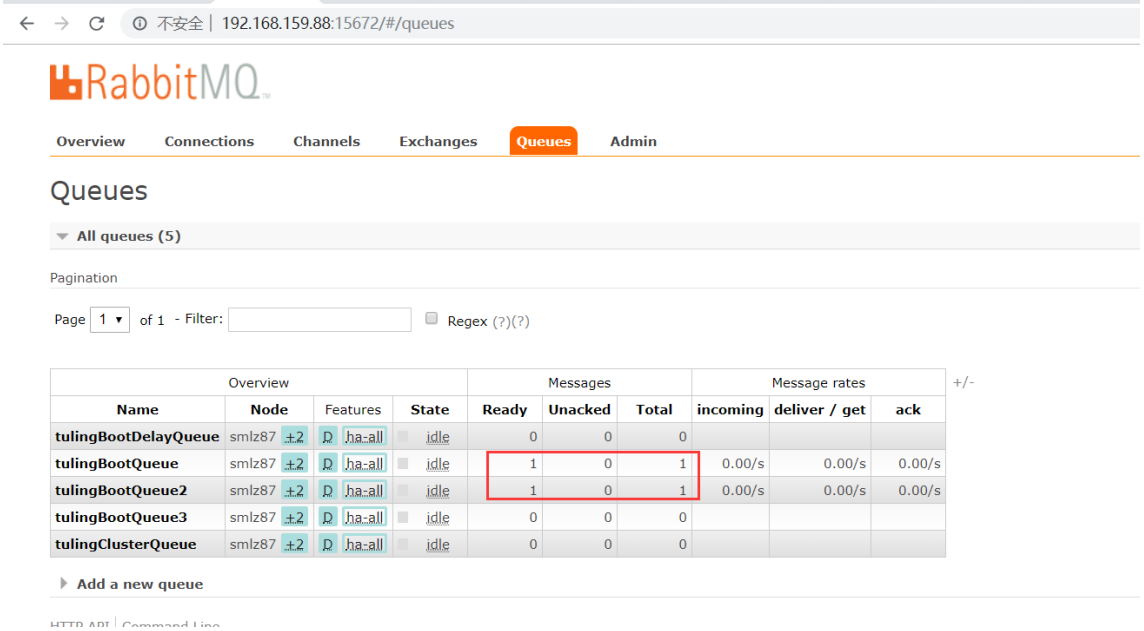
应用程序客户端连接地址:如下（生产端）

```
spring.rabbitmq.host=192.168.159.89
spring.rabbitmq.port=5672
spring.rabbitmq.virtual-host=/
spring.rabbitmq.username=guest
spring.rabbitmq.password=guest
```

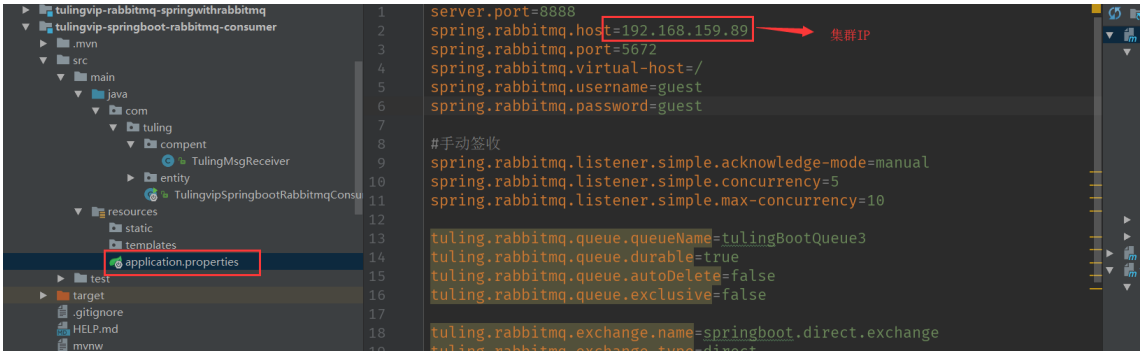
3.1）**发送消息到集群上**



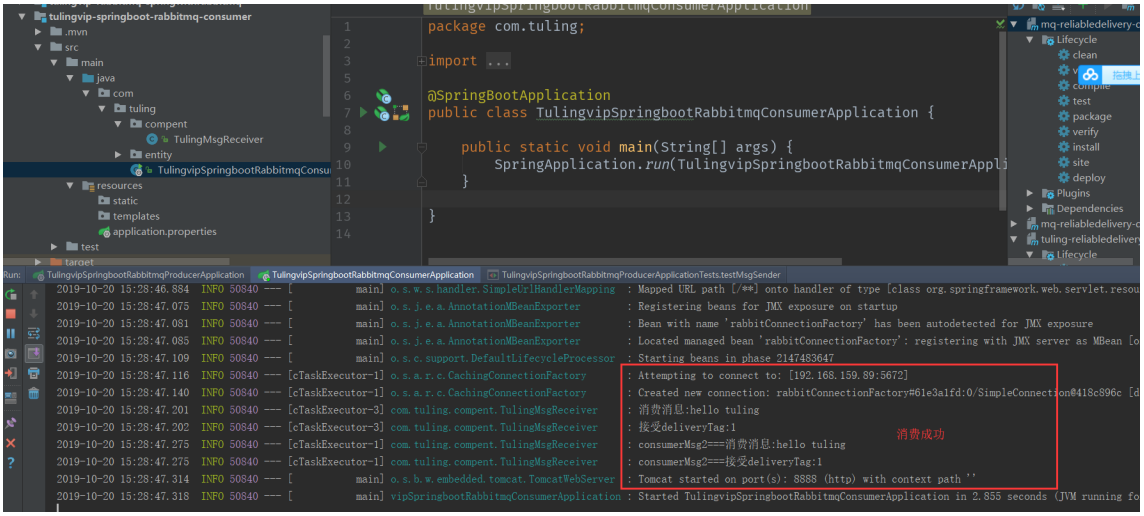




消费端:连接配置:

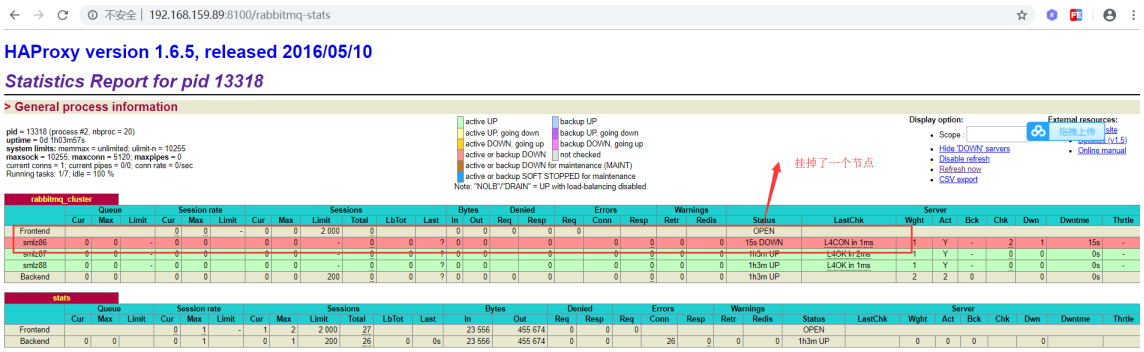


启动集群:

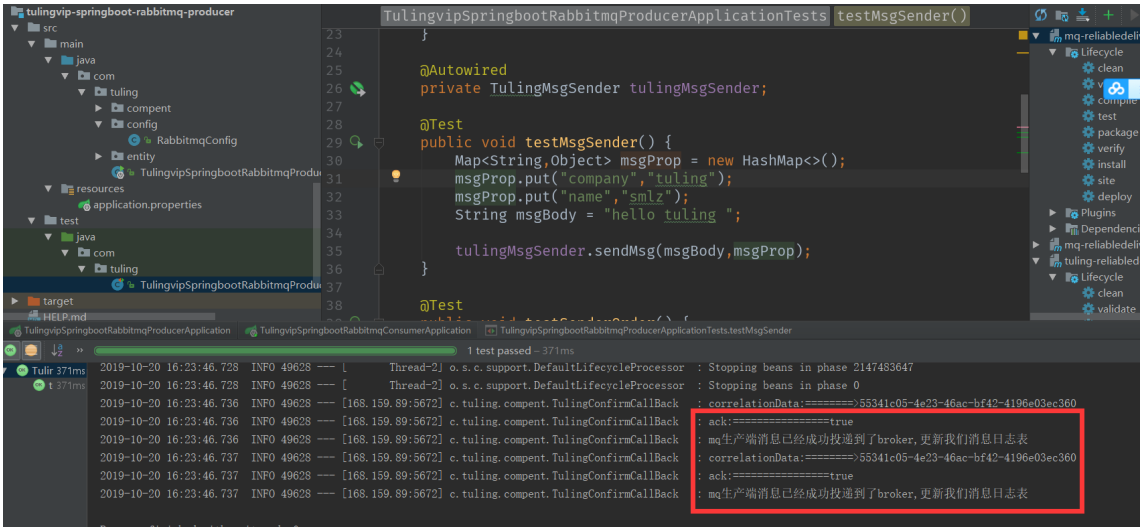


四:模拟集群服务器故障

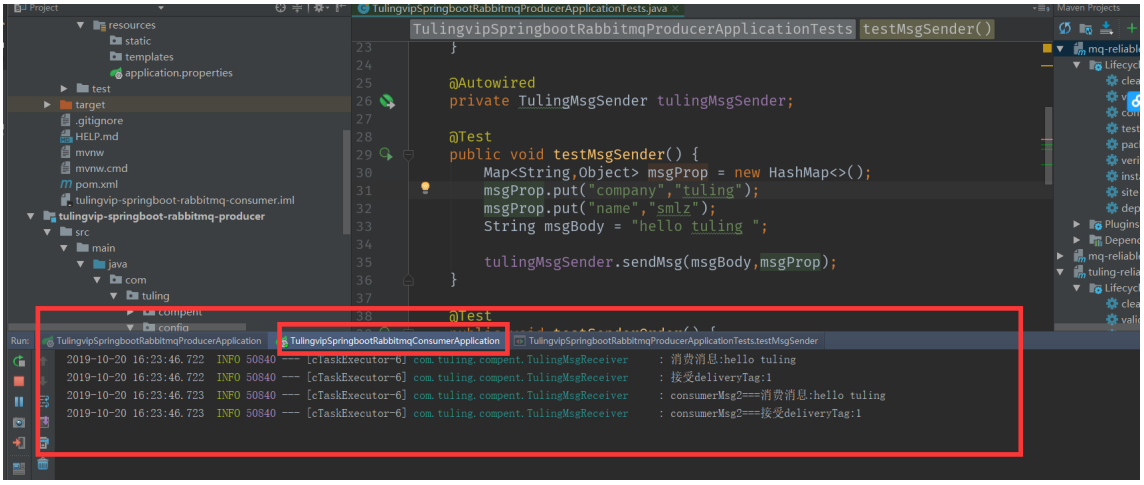
1)**停止主节点86节点**



生产端:成功发送



消费端:成功接收


TulingvipSpringbootRabbitmqProducerApplicationTests.java
TulingvipSpringbootRabbitmqProducerApplicationTests
testMsgSender()
@Autowired
private TulingMsgSender tulingMsgSender;
@Test
public void testMsgSender() {
Map<String,Object> msgProp = new HashMap<>();
msgProp.put("company","tuling");
msgProp.put("name","smlz");
String msgBody = "hello tuling ";
tulingMsgSender.sendMsg(msgBody,msgProp);
}
TulingvipSpringbootRabbitmqProducerApplication
TulingvipSpringbootRabbitmqConsumerApplication
TulingvipSpringbootRabbitmqProducerApplicationTests.testMsgSender
2019-10-20 16:23:46.722 INFO 50840 --- [cTaskExecutor-6] com.tuling.compent.TulingMsgReceiver : 消费消息:hello tuling
2019-10-20 16:23:46.722 INFO 50840 --- [cTaskExecutor-6] com.tuling.compent.TulingMsgReceiver : 接受deliveryTag:1
2019-10-20 16:23:46.723 INFO 50840 --- [cTaskExecutor-6] com.tuling.compent.TulingMsgReceiver : consumerMsg2===消费消息:hello tuling
2019-10-20 16:23:46.723 INFO 50840 --- [cTaskExecutor-6] com.tuling.compent.TulingMsgReceiver : consumerMsg2===接受deliveryTag:1