<https://www.jianshu.com/p/42e53443940f>

系统环境：*****Centos7.6*****

Docker version： 1.13.1, build 7d71120/1.13.1

Docker Compose version： v2.2.2

三个节点：

10.10.11.79 （这一台做rabbitmq集群根节点）

10.10.11.80 (这台做haproxy+keepalived高可用 master节点)

10.10.11.81 (这台做haproxy+keepalived高可用 backup节点)

[RabbitMQ](https://so.csdn.net/so/search?q=RabbitMQ&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/qq_31745863/article/details/_blank) 集群中节点包括****内存节点(RAM)、 磁盘节点(Disk， 消息持久化)****， ****集群中至少有一个 Disk 节点。****

10.10.11.79,10.10.11.80做磁盘节点，10.10.11.81做内存节点。

下面开始先搭建普通集群模式，此方案只适用非持久化非持久化队列。

# 安装rabbitmq：

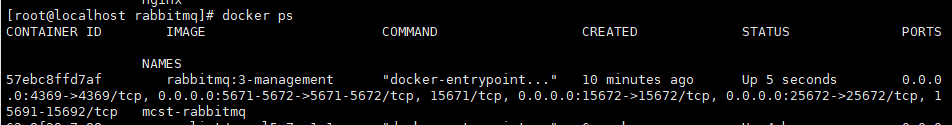
三个节点都创建文件夹/home/rabbitmq,注意挂载目录的权限问题。

节点1 yaml文件（此节点作集群根节点）：

注意：三个节点 RABBITMQ\_ERLANG\_COOKIE 保持一致。一定要有 extra\_hosts 配置，否则在搭建集群的过程中会连接不到其他 rabbitmq 节点服务。此节点作为集群根节点。

启动命令：

docker-compose -f mcst-rabbitmq-node1.yaml up -d



停止：docker-compose -f mcst-rabbitmq-node1.yaml stop

节点2 yaml文件（从节点）：

另外在同目录下面需要编辑一个脚本node2-rabbitmq.sh：

rabbitmqctl stop\_app

rabbitmqctl reset

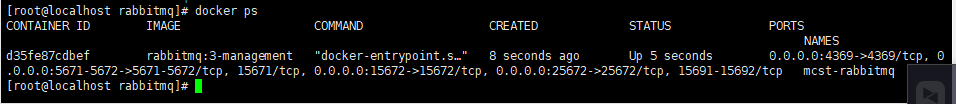
rabbitmqctl join\_cluster rabbit@rabbitmq1

rabbitmqctl start\_app

启动命令：

docker-compose -f mcst-rabbitmq-node2.yaml up -d

停止：docker-compose -f mcst-rabbitmq-node2.yaml stop



进入容器内部：

docker exec -it mcst-rabbitmq bash

执行脚本将当前节点添加到集群中（注意文件执行权限）：

cd home

sh node2-rabbitmq.sh

我执行脚本的时候报了这个错误：

Error: unable to perform an operation on node 'rabbit@rabbitmq2'

把节点2的防火墙关闭：

systemctl stop firewalld

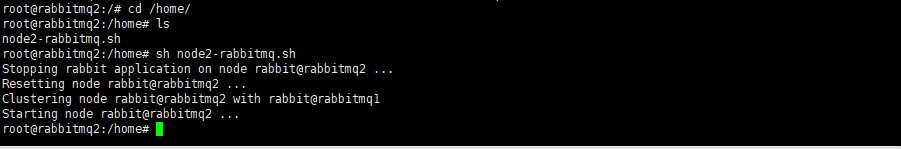
重启docker：

systemctl restart docker

重启容器：

docker start mcst-rabbitmq

进入容器重新执行脚本即可：



节点3 yaml文件（从节点）：

同目录下建一个脚本rabbitmq-ram.sh：

rabbitmqctl stop\_app

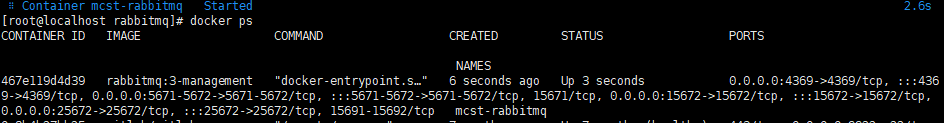
rabbitmqctl reset

rabbitmqctl join\_cluster --ram rabbit@rabbitmq1

rabbitmqctl start\_app

启动命令：

docker-compose -f mcst-rabbitmq-node3.yaml up -d



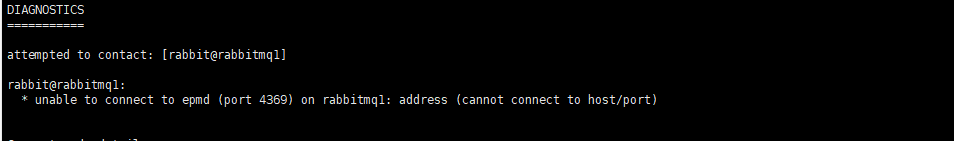
进入容器内部：

docker exec -it mcst-rabbitmq bash

执行脚本， 把内存节点到集群中：

sh rabbitmq-ram.sh

我第三个几点启动的时候报链接不上根节点：



把根节点（10.10.11.79）的网络端口打开：

查看端口是否开放

firewall-cmd --query-port=4369/tcp

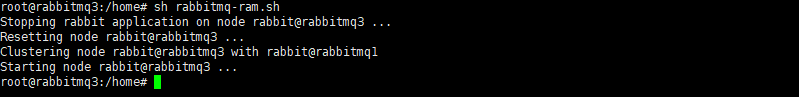
开放端口

firewall-cmd --permanent --add-port=4369/tcp

重启防火墙

firewall-cmd --reload

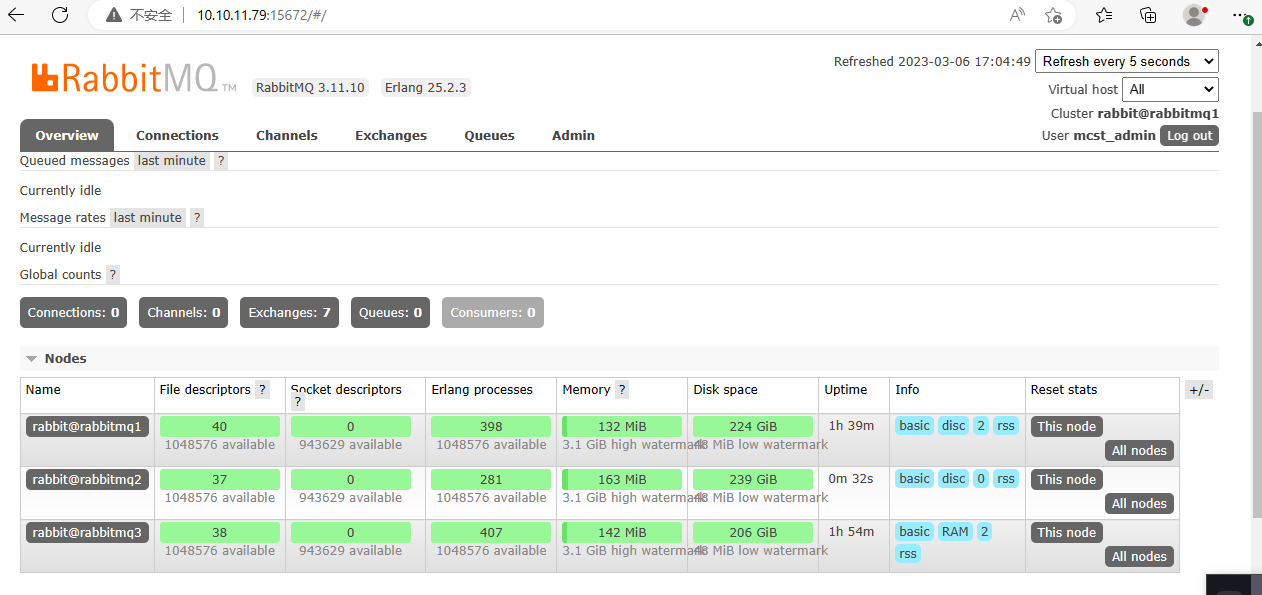
重试：



查看：

http://10.10.11.79:15672 进入管理端查看集群状态。

这三节点的个ip 都可以访问，用户：mcst\_admin 密码：mcst\_admin\_123



将rabbitmq2，rabbitmq3的节点加入rabbitmq1中创建普通集群，集群就已经搭建好了。

Springboot链接一下：

这里使用addresses 链接多个ip。

spring:

rabbitmq:

# rabbitmq集群地址

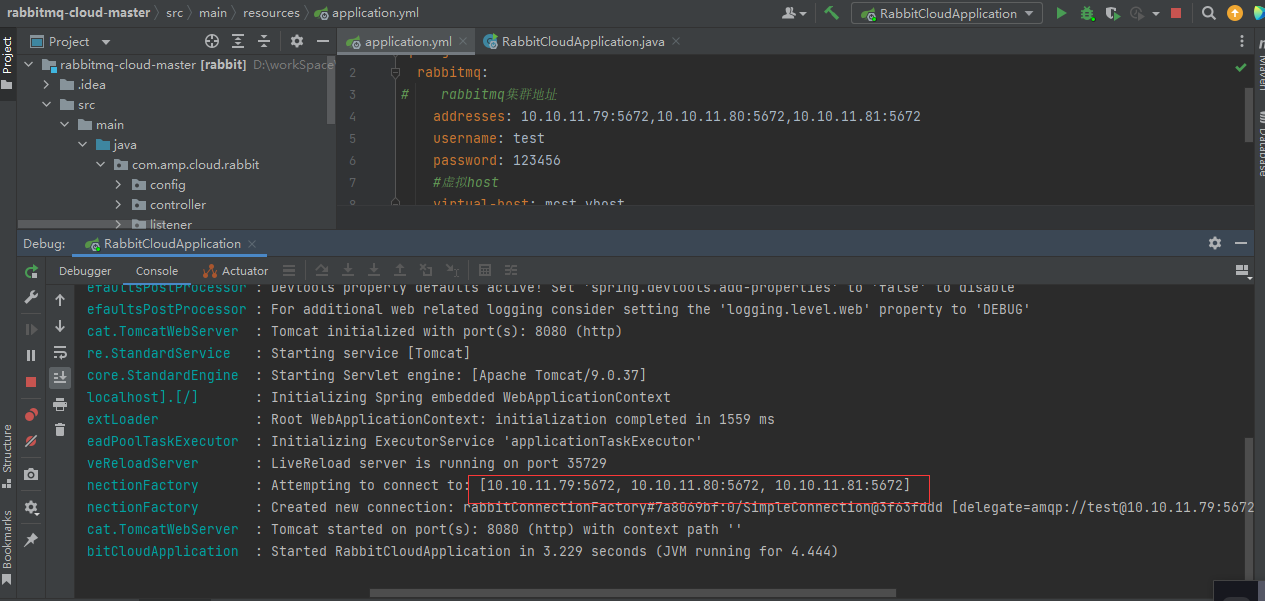
addresses: 10.10.11.79:5672,10.10.11.80:5672,10.10.11.81:5672

username: test

password: 123456

#虚拟host

virtual-host: mcst\_vhost



在普通模式的基础上，还把需要的队列做成镜像队列，存在于多个节点来实现高可用(HA)：

# **haproxy+keepalived高可用集群**

## # 安装**HAProxy**

使用haproxy 做负载均衡

因为79是集群的根节点，所以我们在80，81两个节点上建立文件夹/home/haproxy

需要创建两个文件，

1.创建启动容器的mcst-haproxy.yaml 文件：

设置 extra\_hosts（rabbitmq 集群节点 ip） 和 volumes（使用自定义的配置文件

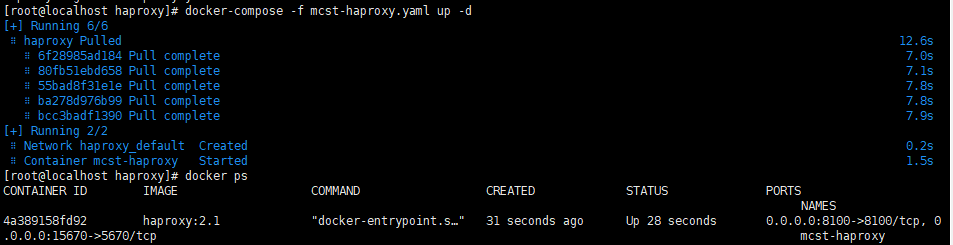
）

1. yaml文件同级目录下面创建haproxy 配置文件haproxy.cfg：

两个节点使用相同的配置。

两个节点都要启动，部署命令：

docker-compose -f mcst-haproxy.yaml up -d



HAProxy Master：10.10.11.80

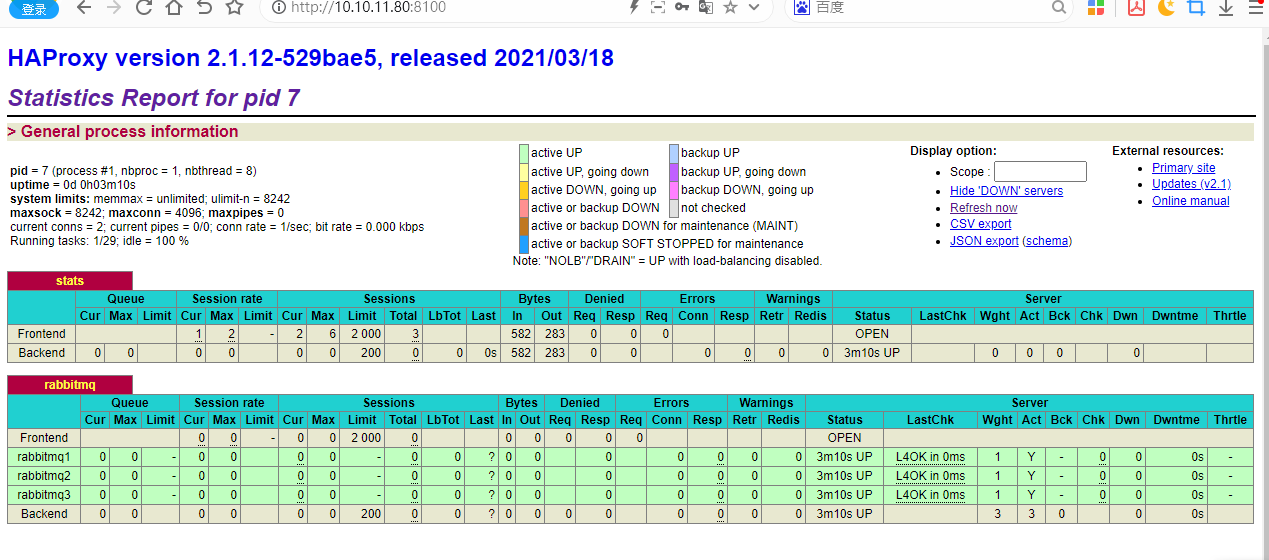
HAProxy Backup：10.10.11.81

登录 HAProxy 的管理端查看集群状态：

用户：admin

密码：admin123

<http://10.10.11.80:8100/>



# # 安装 Keepalived

如果其中一个节点的HAProxy 服务挂了，我们也要保证服务是可用的，这样就需要健康检测服务Keepalived，使用keepalived搭建双主热备。

## **## yum安装：**

yum clean all

yum install -y keepalived

## ## **keepalived常用命令：**

systemctl start keepalived.service #启动keepalived

systemctl stop keepalived.service #停止keepalived

systemctl status keepalived.service #查看keepalived状态

ifconfig 命令查看服务器网卡，找到和本机服务 ip 对应的网卡：

ifconfig

### Master （80节点）节点配置：

创建配置文件 /etc/keepalived/keepalived.conf

### Backup（81节点）节点配置：

创建配置文件 /etc/keepalived/keepalived.conf

创建完配置，启动 keepalived：

systemctl restart keepalived

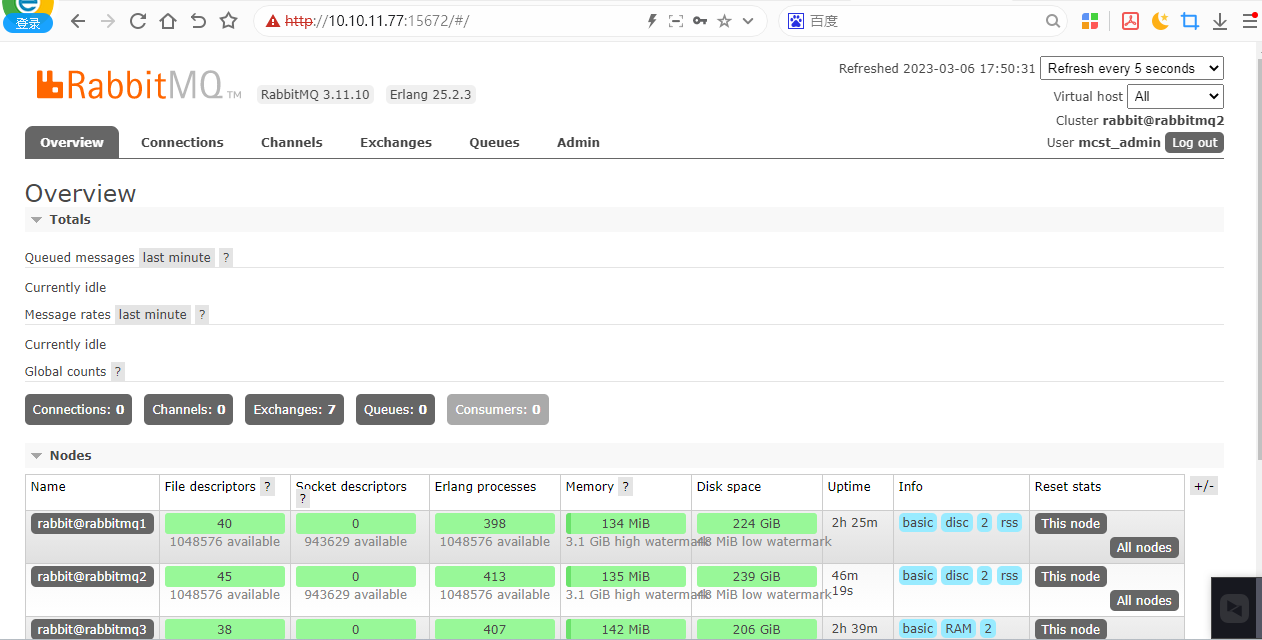
关闭：

systemctl stop keepalived

通过虚拟ip访问：

http://10.10.11.77:15672

用户：mcst\_admin 密码：mcst\_admin\_123



# springboot链接测试：

使用keepalived的虚拟ip链接

spring:

rabbitmq:

# rabbitmq集群地址

host: 10.10.11.77

port: 5672

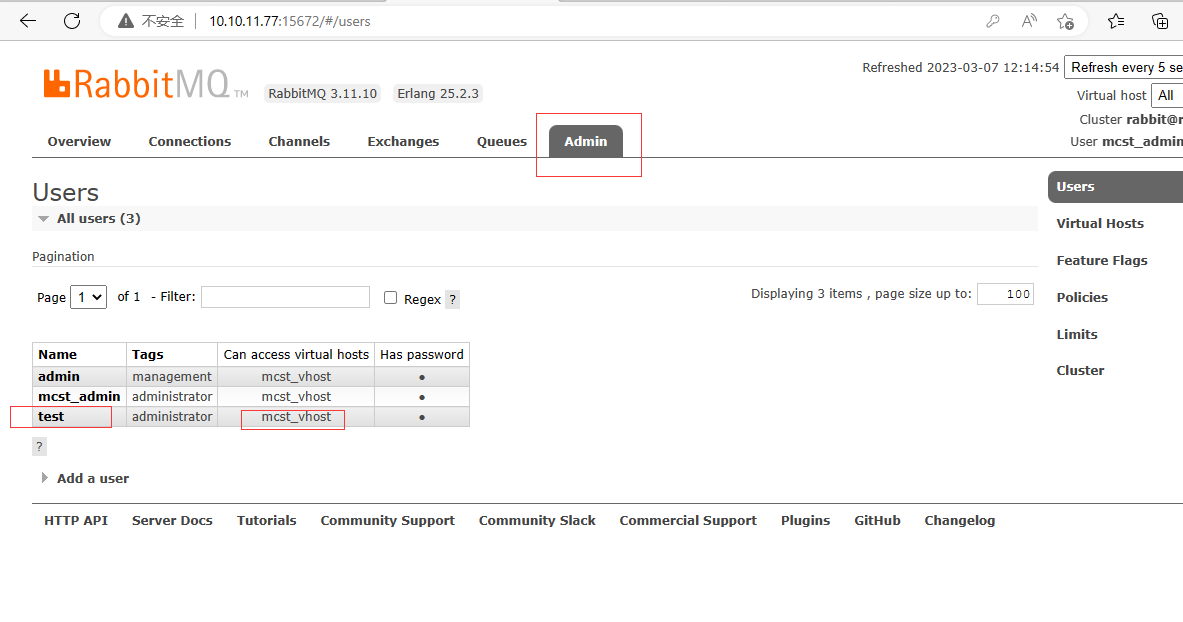
username: test

password: 123456

#虚拟host

virtual-host: mcst\_vhost

如果链接不上，检查一下下面几个地方：



1. mq的端口号，是5672，不是访问浏览器时候的15672
2. 不要使用默认账户，新建用户添加相关的权限
3. Springboot链接配置种是否配置了virtual-host
4. ip+5672 查看一下服务器网络端口是不是开放了

显示链接成功了：

