

从所有教程的词条中查询...

Java / 第六章RabbitMQ集群入门

全部开发者教程

RabbitMQ理论+实战精讲

每章总结

第二章RabbitMQ快速入门回顾

第三章RabbitMQ基础总结

第四章RabbitMQ高级特性总结

第五章RabbitMQ和SpringBoot适配

第六章RabbitMQ集群入门

第七章RabbitMQ集群高可用



Moody • 更新于 2020-10-10

◀ 上一节 第五章RabbitM... 第七章RabbitM... 下一节 ▶

单节点安装Rabbitmq指南

基于 CentOS 8 版本

设置主机名称，注意将星号替换为数字

```
hostnamectl set-hostname mq0*.localdomain
```

在hosts文件中，前两行里加入主机名称

```
vi /etc/hosts
```

安装epel

```
sudo yum install epel-release -y
```

安装erlang

```
sudo yum install erlang -y
```

安装socat

```
yum install socat -y
```

安装wget

```
yum install wget -y
```

下载rabbitmq安装包

```
wget https://github.com/rabbitmq/rabbitmq-server/releases/download/v3.8.8/rab
```

导入rabbitmq密钥

```
rpm -import https://www.rabbitmq.com/rabbitmq-release-signing-key.asc
```

安装rabbitmq

```
rpm -ivh rabbitmq-server-3.8.8-1.el8.noarch.rpm
```

启动rabbitmq

```
systemctl start rabbitmq-server
```

查看rabbitmq服务状态

```
systemctl status rabbitmq-server
```

✎ 意见反馈

♡ 收藏教程

🔖 标记书签

启用管控台插件

```
rabbitmq-plugins enable rabbitmq_management
```

关闭系统防火墙

```
systemctl stop firewalld.service  
systemctl disable firewalld.service
```

添加测试账户

```
rabbitmqctl add_user test test  
rabbitmqctl set_user_tags test administrator  
rabbitmqctl set_permissions -p / test ".*" ".*" ".*"
```

RabbitMQ集群配置指南

在集群所有节点安装rabbitmq

编辑hosts,使得节点间可以通过主机名互相访问

```
vi /etc/hosts
```

修改.erlang.cookie权限

```
chmod 777 /var/lib/rabbitmq/.erlang.cookie
```

将主节点的.erlang.cookie文件传输至集群所有节点

```
scp /var/lib/rabbitmq/.erlang.cookie root@mq02:/var/lib/rabbitmq
```

复原.erlang.cookie权限

```
chmod 400 /var/lib/rabbitmq/.erlang.cookie
```

加入集群

```
rabbitmqctl stop_app  
rabbitmqctl join_cluster --ram rabbit@mq01  
rabbitmqctl start_app
```

镜像队列配置指南

在主节点增加镜像队列配置

```
rabbitmqctl set_policy ha-all "^" '{"ha-mode":"all"}'
```

haproxy负载均衡安装指南

安装haproxy

```
yum install haproxy -y
```

编辑hosts, 使得haproxy能够通过主机名访问集群节点

```
vi /etc/hosts
```

编辑haproxy配置文件

```
vi /etc/haproxy/haproxy.cfg
```

意见反馈

收藏教程

标记书签

```
global
# 日志输出配置、所有日志都记录在本机，通过 local0 进行输出
log 127.0.0.1 local0 info
# 最大连接数
maxconn 4096
daemon
# 默认配置
defaults
# 应用全局的日志配置
log global
# 使用4层代理模式，7层代理模式则为"http"
mode tcp
# 日志类别
option tcplog
# 不记录健康检查的日志信息
option dontlognull
# 3次失败则认为服务不可用
retries 3
# 每个进程可用的最大连接数
maxconn 2000
# 连接超时
timeout connect 5s
# 客户端超时
timeout client 120s
# 服务端超时
timeout server 120s

# 绑定配置
listen rabbitmq_cluster
bind :5672
# 配置TCP模式
mode tcp
# 采用加权轮询的机制进行负载均衡
balance roundrobin
# RabbitMQ 集群节点配置
server mq01 mq01:5672 check inter 5000 rise 2 fall 3 weight 1
server mq02 mq02:5672 check inter 5000 rise 2 fall 3 weight 1
server mq03 mq03:5672 check inter 5000 rise 2 fall 3 weight 1

# 配置监控页面
listen monitor
bind *:8100
mode http
option httplog
stats enable
stats uri /rabbitmq
stats refresh 5s
```

设置seLinux

```
sudo setsebool -P haproxy_connect_any=1
```

关闭防火墙

```
systemctl stop firewalld.service
systemctl disable firewalld.service
```

启动haproxy

```
systemctl start haproxy
```

[意见反馈](#)[收藏教程](#)[标记书签](#)

keepalived配置指南

安装keepalived

```
yum install keepalived -y
```

编辑keepalived配置文件

```
vi /etc/keepalived/keepalived.conf
```

主机配置文件:

```
! Configuration File for keepalived

global_defs {
    router_id mq04
    vrrp_skip_check_adv_addr
    vrrp_strict
    vrrp_garp_interval 0
    vrrp_gna_interval 0
}

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
    interface ens33
    virtual_router_id 51
    mcast_src_ip 192.168.57.133
    priority 100
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 1111
    }
    virtual_ipaddress {
        192.168.57.233
    }
    track_script {
        chk_haproxy
    }
}
```

热备机配置文件:

```
global_defs {
    router_id mq04
    vrrp_skip_check_adv_addr
    vrrp_strict
    vrrp_garp_interval 0
    vrrp_gna_interval 0
}

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
}
```

意见反馈

收藏教程

标记书签

```
vrrp_instance VI_1 {
    state BACKUP
    interface ens33
    virtual_router_id 51
    mcast_src_ip 192.168.57.131
    priority 50
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 1111
    }
    virtual_ipaddress {
        192.168.57.233
    }
    track_script {
        chk_haproxy
    }
}
```

健康检测脚本

```
vi /etc/keepalived/haproxy_check.sh

#!/bin/bash
COUNT=`ps -C haproxy --no-header |wc -l`
if [ $COUNT -eq 0 ];then
    systemctl start haproxy
    sleep 2
    if [ `ps -C haproxy --no-header |wc -l` -eq 0 ];then
        systemctl stop keepalived
    fi
fi
```

修改健康检测脚本执行权限

```
chmod +x /etc/keepalived/haproxy_check.sh
```

启动keepalived

```
systemctl start keepalived
```

做故障转移实验时，关闭keepalived即可

```
systemctl stop keepalived
```

Federation 配置指南

首先需要开启federation插件

```
rabbitmq-plugins enable rabbitmq_federation_management
```

federation的配置一共有三个层次

- Upstreams - 每个upstream定义怎么连接到其他broker
- Upstream sets - 把每一个upstream设置在一个组中，使upstreams使用federation。
- Policies 可以将Upstreams和Upstream sets按照规则配置到exchange和queue中

实际上，在最简单的使用情况下，你可以忽略已经存在的upstream设置，因为有一个隐含的默认upstream叫做“all”，他会添加所有的upstream

意见反馈

收藏教程

标记书签

upstreams和upstream set都是实例的参数，就像exchanges、queues、virtual host都有

实验环境

把ubuntuTest01的数据自动复制到ubuntuTest02

- 10.20.112.26 ubuntuTest01
- 10.20.112.27 ubuntuTest02

首先确保两台机器的federation插件已经安装，参照上面步骤。

在浏览器登陆ubuntuTest02的rabbitmq的ui管理界面：<http://10.20.112.27:15672/#/>

创建exchange: test.exchange, 使用默认配置

UI操作: Exchanges->Add a new exchange

在rabbitmq中会有一些默认的exchange, 创建完毕后如图:

创建queue: test.queue, 绑定到test.exchange, key使用test。

UI操作: Queues->Add a new queue

绑定到test.exchange, 并设置key

UI操作: Queues->All queues(test.queue[单击])>Bindings

创建upstream: upstream1

UI操作: Admin->Federation Upstreams->Add a new upstream

创建Parameters: mqcluster

UI操作: Admin->Federation Upstreams->Parameters

创建policy: mypolicy

UI操作: Admin->Policies->Add / update a policy

状态图

观察26上面的连接

