## RabbitMQ安装配置

## 安装erlang

下载地址：http://www.erlang.org/downloads

yum install ncurses-devel

tar xf otp\_src\_20.1.tar.gz

cd otp\_src\_20.1

./configure --prefix=/usr/local/erlang20 --without-javac

make

make install

erl验证

## 安装RabbitMQ

安装python

yum install python -y

安装simplejson

yum install xmlto -y

yum install python-simplejson -y

下载源码：http://www.rabbitmq.com/download.html

Generic Unix -> rabbitmq-server-generic-unix-3.6.14.tar.xz

xz -d rabbitmq-server-generic-unix-3.6.14.tar.xz

tar xf rabbitmq-server-generic-unix-3.6.14.tar

mv rabbitmq\_server-3.6.14 /usr/local/rabbitmq

## 设置环境变量

export PATH=$PATH:/usr/local/ruby/bin:/usr/local/erlang20/bin:/usr/local/rabbitmq/sbin

source /etc/profile

./rabbitmq-server启动rabbitMQ server 5672端口监听

rabbitmqctl stop 停止

## 设置guest可以远程连接

修改rabbitmq的配置/usr/local/rabbitmq/etc/rabbitmq/rabbitmq.config

添加：[{rabbit, [{loopback\_users, []}]}].

重新启动

<http://www.rabbitmq.com/access-control.html>

## 启用管理控制台

./sbin/rabbitmq-plugins enable rabbitmq\_management

重启rabbitmq

打开浏览器输入： <http://127.0.0.1:15672/>，用户名guest，密码guest

Topic：\*1个单词，#0个或者多个单词。

## 四中路由模式

pom.xml:

<!-- mq -->

<dependency>

<groupId>com.rabbitmq</groupId>

<artifactId>amqp-client</artifactId>

<version>3.5.1</version>

</dependency>

<dependency>

<groupId>org.springframework.amqp</groupId>

<artifactId>spring-rabbit</artifactId>

<version>1.4.5.RELEASE</version>

</dependency>





## 如何保证重启服务器消息也不丢失

1. Exchage持久化
2. Queue持久化
3. 发送消息的时候，设置MessageDeliveryMode为MessageDeliveryMode.PERSISTENT，默认行为
4. 消息手动确认

## CentOS6.7安装报错：{missing\_dependencies,[crypto,ssl]}

（1）重新安装下openssl：

tar -zvxf openssl-1.0.1s.tar.gz

cd openssl-1.0.1s

./config --prefix=/usr/local/openssl

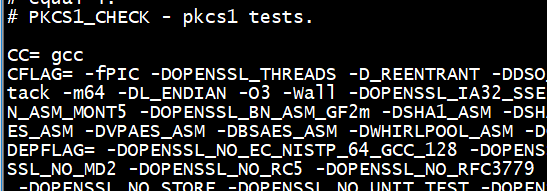
修改Makefile：

vi Makefile

将原来的：CFLAG= -DOPENSSL\_THREADS

修改为： CFLAG= -fPIC -DOPENSSL\_THREADS

也就是添加-fPIC



执行执行：

make && make install

（2）重新安装erlang：

tar xf otp\_src\_20.1.tar.gz

cd otp\_src\_20.1

./configure --prefix=/usr/local/erlang20 --without-javac --with-ssl=/usr/local/openssl

Make && make install

（3）启动rabbitmq-server, 现在应该就可以了！

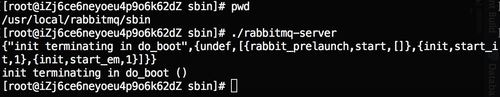
或者更简单的：

yum install openssl

yum install openssl-devel

然后再重新安装erlang，未经证实

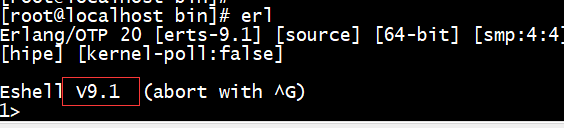
## RabbitMQ安装报错：{init terminating in do\_boot



这个有可能是系统自带的erlang版本太低导致的。

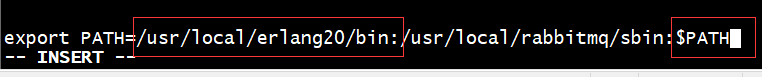
在命令行直接输入：erl

如果输出的版本不是V9.1：

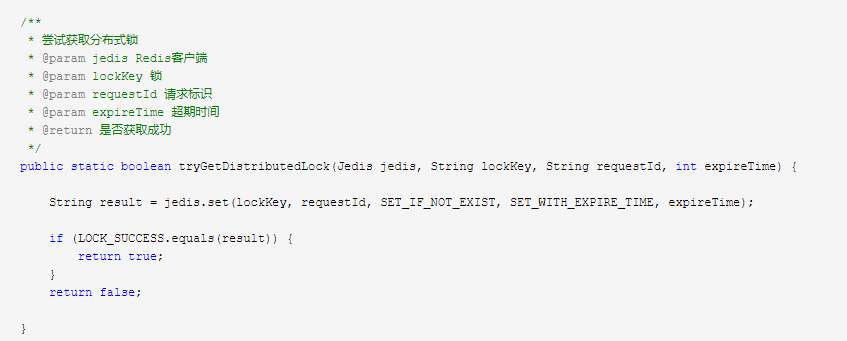


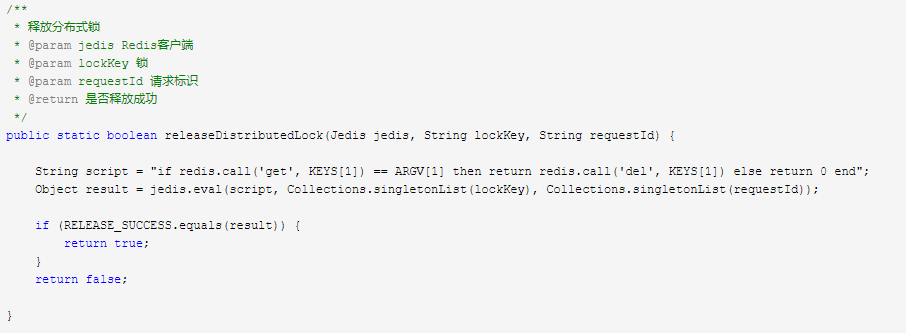
说明没有使用我们自己安装的erlang。只需要调整一下path就可以了：

vi /etc/profile

注意，我们自己安装的放在PATH前面。

## 如何用Redis实现分布式锁？





参考：<https://www.cnblogs.com/linjiqin/p/8003838.html>

# 集群搭建

## 修改hosts

在两台机器上配置响应的hosts如下：

192.168.220.133 centos6\_7

192.168.220.134 centos7\_2

## 2. 复制.erlang.cookie

首先启动两台机器的rabbitmq，此时会在HOME目录（默认情况下）生成cookie文件.erlang.cookie，用于erlang在分布式节点之间进行通讯，此cookie文件的内容必须相同，否则无法通讯。

1.启动指令：

RABBITMQ\_NODENAME=rabbit@centos6\_7 rabbitmq-server -detached

RABBITMQ\_NODENAME=rabbit@centos7\_2 rabbitmq-server -detached

2. 复制cookie

改变了节点cookie之后不能对节点通讯，此时用kill指令关掉rabbitmq进程重启即可，当然也可以优雅的先关闭然后修改cookie:

关闭：rabbitmqctl -n rabbit@centos6\_7 stop，然后修改cookie

注意：如果修改了节点名，接后续操作必须-n指明，否则erlang无法找到节点

## 3. 将节点加入集群:

## 在修改了cookie之后，启动rabbitmq：

RABBITMQ\_NODENAME=rabbit@centos6\_7 rabbitmq-server –detached

关闭app并reset：

rabbitmqctl -n rabbit@centos6\_7 stop\_app

rabbitmqctl -n rabbit@centos6\_7 reset

将节点加入集群:

rabbitmqctl -n rabbit@centos6\_7 join\_cluster rabbit@centos7\_2

输出：Clustering node rabbit@rabbit2 with rabbit@rabbit1 ..

## 4.查看集群状态

[root@localhost ~]# rabbitmqctl -n rabbit@centos7\_2 cluster\_status

Cluster status of node rabbit@centos7\_2

[{nodes,[{disc,[rabbit@centos6\_7,rabbit@centos7\_2]}]},

{running\_nodes,[rabbit@centos7\_2]},

{cluster\_name,<<"rabbit@localhost">>},

{partitions,[]},

{alarms,[{rabbit@centos7\_2,[]}]}]

[root@localhost ~]#

## 用户角色权限管理

新建用户：rabbitmqctl add\_user root root123

删除用户：rabbitmqctl delete\_user root

验证用户：rabbitmqctl authenticate\_user root root321

用户的角色分为5种类型：

none：无任何角色。新创建的用户的角色默认为none。

management：可以访问Web管理页面。

policymaker：包含management的所有权限，并且可以管理策略（policy）和参数（parameter）。

monitoring：包含management的所有权限，并且可以看到所有连接（connections）、信道（channels）以及节点相关的信息。

administartor：包含monitoring的所有权限，并且可以管理用户、虚拟主机、权限、策略、参数等等。administator代表了最高的权限。

授角色：rabbitmqctl set\_user\_tags root administrator

添加vhost：rabbitmqctl add\_vhost vhost1

删除vhost：rabbitmqctl delete\_vhost vhost1

列举vhost：rabbitmqctl list\_vhosts

授权：rabbitmqctl set\_permissions [-p vhost] {user} {conf} {write} {read}

vhost：授予用户访问权限的vhost名称，可以缺省，即vhost为”/”。

user：可以访问指定vhost的用户名称。

conf：一个用于匹配用户在哪些资源上拥有可配置权限的正则表达式。

write：一个用于匹配用户在哪些资源上拥有可写权限的正则表达式。

read：一个用于匹配用户在哪些资源上有用可读权限的正则表达式。

rabbitmqctl set\_permissions -p vhost1 root ".\*" ".\*" ".\*"