Rabbitmq linux安装&集群高可用

### rabbitmq linux centos7下安装

在安装(搭建集群)之前，确定两个点 1:防火墙关掉 2:打开网络（方便操作）

关闭防火墙 systemctl stop firewalld.service

禁止开机自启 systemctl disable firewalld.service

首先还是安装erlang

下载erlang wget http://www.rabbitmq.com/releases/erlang/erlang-18.2-1.el6.x86\_64.rpm

安装erlang rpm -ihv http://www.rabbitmq.com/releases/erlang/erlang-18.2-1.el6.x86\_64.rpm

安装完erlang之后开始rabbitmq，还是再提醒一下先装erlang 再装rabbitmq

装Rabbitmq之前先装一个公钥 :

rpm --import https://dl.bintray.com/rabbitmq/Keys/rabbitmq-release-signing-key.asc

装好公钥之后下载Rabbitmq:

wget http://www.rabbitmq.com/releases/rabbitmq-server/v3.6.6/rabbitmq-server-3.6.6-1.el7.noarch.rpm

安装:

rpm -ihv rabbitmq-server-3.6.6-1.el7.noarch.rpm

安装中途可能会提示你需要一个叫socat的插件

如果提示了就先安装socat 再装rabbitmq

安装socat:

yum install socat

至此就装好了Rabbitmq了，可以执行以下命令启动Rabbitmq:

service rabbitmq-server start

或

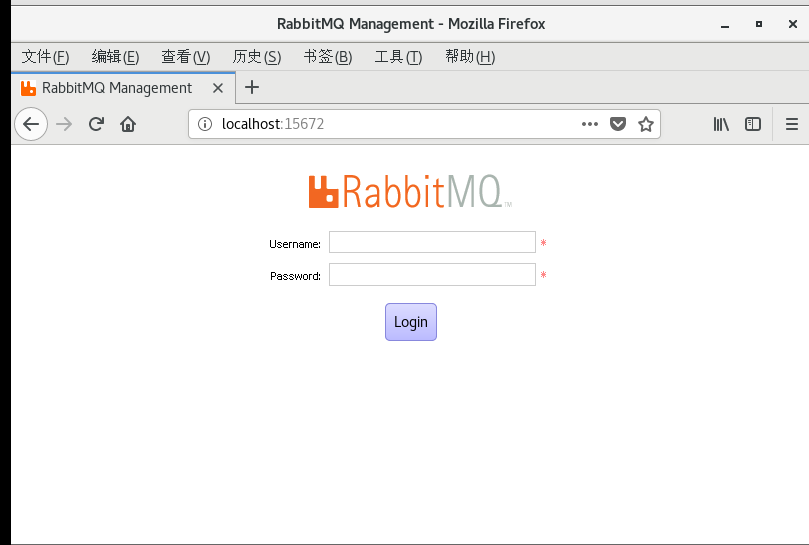
systemctl start rabbitmq-server

和windows环境下一样rabbitmq对于linux也提供了他的管理插件

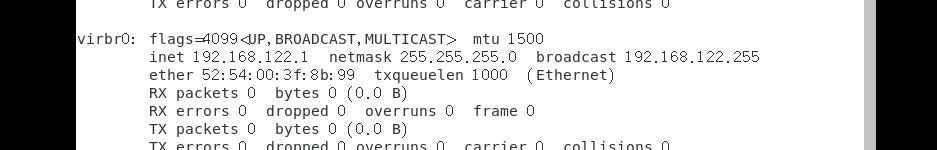
安装rabbitmq管理插件:

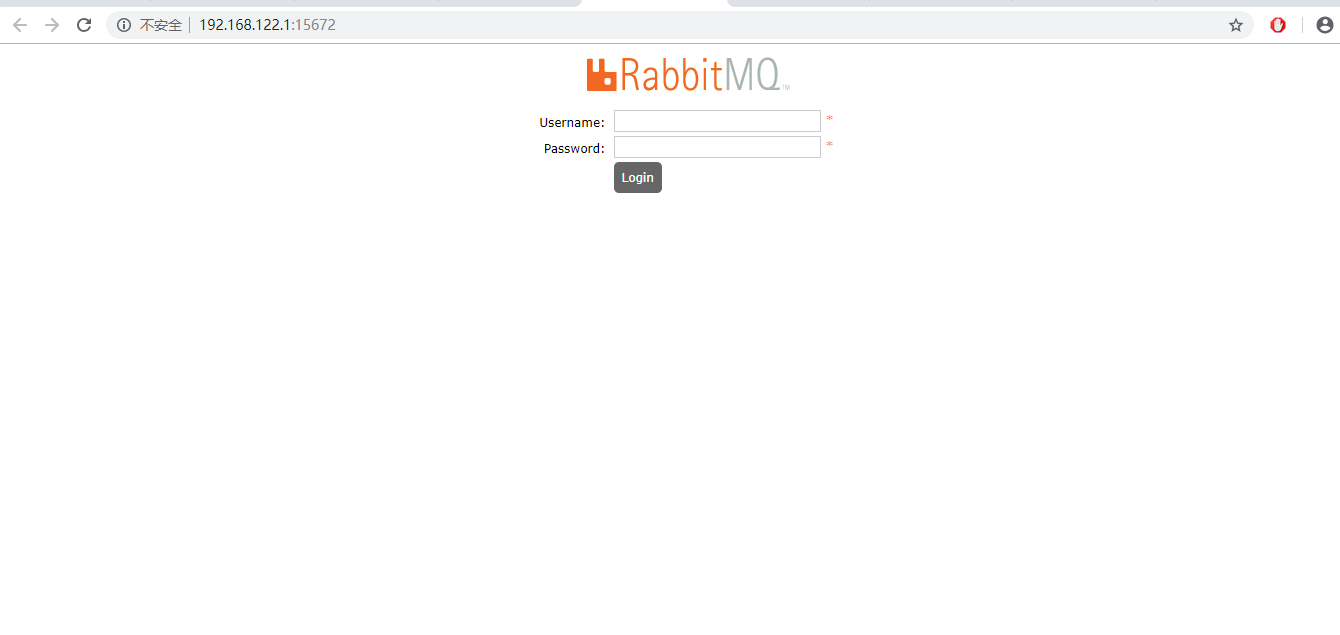
rabbitmq-plugins enable rabbitmq\_management

安装完管理插件之后，如果有装了浏览器的话比如火狐，可以和windows一样 访问一下：localhost:15672 可以看到一个熟悉的页面：



当然如果你和我一样，是用虚拟机搭建的linux的话，可以用主机访问一下也是没问题的！比如说我这里虚拟机的地址为:





那么安装我们就讲到这，后面我们来讲集群环境的搭建以及一些问题。

### rabbitmq集群搭建，配置

rabbbitmq由于是由erlang语言开发的，天生就支持分布式

rabbitmq 的集群分两种模式，**一种是默认模式，一种是镜像模式**

当然所谓的镜像模式是基于默认模式加上一定的配置来的

在rabbitmq集群当中所有的节点（一个rabbitmq服务器）会被归为两类，**一类是磁盘节点，一类是内存节点**

磁盘节点会把集群的所有信息(比如交换机,队列等信息)持久化到磁盘当中，而内存节点只会将这些信息保存到内存当中，讲白了重启一遍就没了。

为了可靠性考虑rabbitmq官方强调集群环境至少需要有一个磁盘节点，而且为了高可用的话，必须至少要有2个磁盘节点，因为如果只有一个磁盘节点而刚好这唯一的磁盘节点宕机了的话，集群虽然虽然可以运作，但是不能对集群进行任何的修改操作（比如队列添加，交换机添加，增加/移除新的节点等）。

具体想让rabbitmq实现集群，我们首先需要改一下系统的hostname (因为rabbitmq集群节点名称是读取hostname的)

这里 我们模拟3个节点 :

rabbitmq1

rabbitmq2

rabbitmq3

linux修改hostname命令: hostnamectl set-hostname [name]

修改后重启一下让rabbitmq重新读取节点名字

然后我们需要让每个节点通过hostname能ping通（记得关闭防火墙）这里我们可以修改修改一下hosts文件

关闭防火墙:

关闭防火墙 systemctl stop firewalld.service

禁止开机自启 systemctl disable firewalld.service

接下来,我们需要将各个节点的.erlang.cookie文件内容保持一致(文件路径/var/lib/rabbitmq/.erlang.cookie)

因为我是采用虚拟机的方式来模拟集群环境，所以如果像我一样是克隆的虚拟机的话同步.erlang.cookie文件这个操作在克隆的时候就已经完成了。

上面这些步骤完成之后 我们就可以开始来构建集群了

我们先让rabbitmq2加入 rabbitmq1与他构建为一个集群

执行命令( ram:使rabbitmq2成为一个内存节点，默认为:disk磁盘节点)：

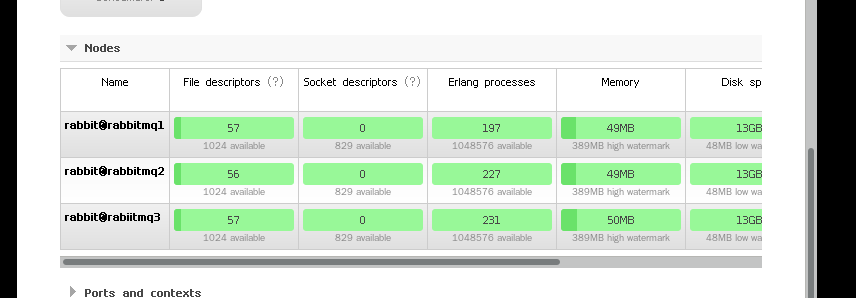
rabbitmqctl stop\_app

rabbitmqctl join\_cluster rabbit@rabbitmq1 --ram

rabbitmqctl start\_app

在构建的时候我们需要先停掉rabbitmqctl应用才能构建，等构建完毕之后再启动，我们把rabbitmq2添加完之后在rabbitmq3节点上也执行同样的代码使他也加入进去，当然我们也可以让rabbitmq3也作为一个磁盘节点

当执行完操作以后我们来看看效果:



随便在哪个节点打开管理页面都能看到集群环境各节点的信息;

有关集群的其他命令:

rabbitmq-server start -detached 启动RabbitMQ节点

rabbitmqctl startapp 启动RabbitMQ应用，而不是节点

rabbitmqctl stopapp 停止

rabbitmqctl status 查看状态

rabbitmqctl add\_user mq 123456 新增账户

rabbitmqctl set\_user\_tags mq administrator 账户tags

rabbitmq-plugins enable rabbitmqmanagement 启用管理控制页面RabbitMQManagement rabbitmqctl clusterstatus 集群状态

rabbitmqctl forgetclusternode rabbit@[nodeName] 节点摘除 rabbitmqctl reset application 重置

普通模式的rabbitmq集群搭建好后，我们来说一下镜像模式

在普通模式下的rabbitmq集群，他会把所有节点的交换机信息和队列的元数据(队列数据分为两种，一种为队列里面的消息，另外一种是队列本身的信息，比如队列的最大容量，队列的名称等等配置信息，后者称之为元数据) 进行复制，确保所有节点都有一份。

而镜像模式，则是把所有的队列数据完全同步（当然对性能肯定会有一定影响） 当对数据可靠性要求高时可以使用镜像模式

实现镜像模式也非常简单有2种方式，一种是直接在管理台控制，另外一种是在声明队列的时候控制

声明队列的时候可以加入镜像队列参数，在上一节参数列表当中有解释，我们来讲一下管理台控制

镜像队列配置命令解释：

rabbitmqctl set\_policy [-p Vhost] Name Pattern Definition [Priority]

-p Vhost：可选参数，针对指定vhost下的queue进行设置

Name: policy的名称

Pattern: queue的匹配模式(正则表达式)

Definition：镜像定义，包括三个部分ha-mode, ha-params, ha-sync-mode

ha-mode:指明镜像队列的模式，有效值为 all/exactly/nodes

all：表示在集群中所有的节点上进行镜像

exactly：表示在指定个数的节点上进行镜像，节点的个数由ha-params指定 nodes：表示在指定的节点上进行镜像，节点名称通过ha-params指定

ha-params：ha-mode模式需要用到的参数

ha-sync-mode：进行队列中消息的同步方式，有效值为automatic和manual

这里举个例子，如果想配置所有名字开头为policy的队列进行镜像且镜像数量为1那么命令如下:

rabbitmqctl setpolicy hapolicy "^policy\_" '{"ha-mode":"exactly","ha-params":1,"ha-sync-mode":"automatic"}'

