

注意: 请使用CentOS7安装

1. 安装依赖环境

在线安装依赖环境:

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

2. 安装Erlang

上传

erlang-18.3-1.el7.centos.x86_64.rpm

socat-1.7.3.2-5.el7.linux.x86_64.rpm

rabbitmq-server-3.6.5-1.noarch.rpm

```
# 安装  
rpm -ivh erlang-18.3-1.el7.centos.x86_64.rpm
```

如果出现如下错误

```
Warning: erlang-22.0.7-1.el7.x86_64.rpm: Header V4 RSA/SHA1 Signature, key ID 6026dfca: NOKEY  
error: Failed dependencies:  
    libc.so.6(GLIBC_2.14)(64bit) is needed by erlang-22.0.7-1.el7.x86_64  
    libc.so.6(GLIBC_2.15)(64bit) is needed by erlang-22.0.7-1.el7.x86_64  
    libcrypto.so.10(OPENSSL_1.0.2)(64bit) is needed by erlang-22.0.7-1.el7.x86_64
```

说明glibc 版本太低。我们可以查看当前机器的glibc 版本

```
strings /lib64/libc.so.6 | grep GLIBC
```

```
GLIBC_2.2.5  
GLIBC_2.2.6  
GLIBC_2.3  
GLIBC_2.3.2  
GLIBC_2.3.3  
GLIBC_2.3.4  
GLIBC_2.4  
GLIBC_2.5  
GLIBC_2.6  
GLIBC_2.7  
GLIBC_2.8  
GLIBC_2.9  
GLIBC_2.10  
GLIBC_2.11  
GLIBC_2.12  
GLIBC_PRIVATE
```

当前最高版本2.12, 需要2.15.所以需要升级glibc

- 使用yum更新安装依赖

```
sudo yum install zlib-devel bzip2-devel openssl-devel ncurses-devel sqlite-  
devel readline-devel tk-devel gcc make -y
```

- 下载rpm包

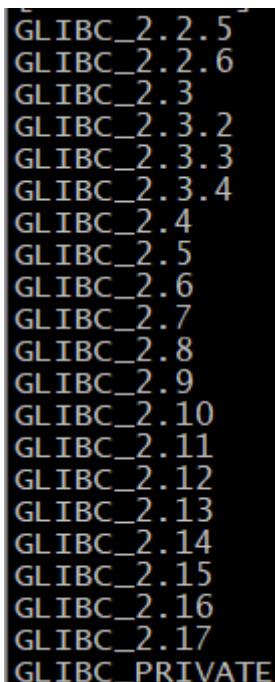
```
wget http://copr-be.cloud.fedoraproject.org/results/mosquito/myrepo-  
el6/epel-6-x86_64/glibc-2.17-55.fc20/glibc-utils-2.17-55.el6.x86_64.rpm &  
wget http://copr-be.cloud.fedoraproject.org/results/mosquito/myrepo-  
el6/epel-6-x86_64/glibc-2.17-55.fc20/glibc-static-2.17-55.el6.x86_64.rpm &  
wget http://copr-be.cloud.fedoraproject.org/results/mosquito/myrepo-  
el6/epel-6-x86_64/glibc-2.17-55.fc20/glibc-2.17-55.el6.x86_64.rpm &  
wget http://copr-be.cloud.fedoraproject.org/results/mosquito/myrepo-  
el6/epel-6-x86_64/glibc-2.17-55.fc20/glibc-common-2.17-55.el6.x86_64.rpm &  
wget http://copr-be.cloud.fedoraproject.org/results/mosquito/myrepo-  
el6/epel-6-x86_64/glibc-2.17-55.fc20/glibc-devel-2.17-55.el6.x86_64.rpm &  
wget http://copr-be.cloud.fedoraproject.org/results/mosquito/myrepo-  
el6/epel-6-x86_64/glibc-2.17-55.fc20/glibc-headers-2.17-55.el6.x86_64.rpm &  
wget http://copr-be.cloud.fedoraproject.org/results/mosquito/myrepo-  
el6/epel-6-x86_64/glibc-2.17-55.fc20/nscd-2.17-55.el6.x86_64.rpm &
```

- 安装rpm包

```
sudo rpm -Uvh socat-1.7.3.2-1.1.el7.x86_64.rpm --force --nodeps
```

- 安装完毕后再查看glibc版本,发现glibc版本已经到2.17了

```
strings /lib64/libc.so.6 | grep GLIBC
```



```
GLIBC_2.2.5  
GLIBC_2.2.6  
GLIBC_2.3  
GLIBC_2.3.2  
GLIBC_2.3.3  
GLIBC_2.3.4  
GLIBC_2.4  
GLIBC_2.5  
GLIBC_2.6  
GLIBC_2.7  
GLIBC_2.8  
GLIBC_2.9  
GLIBC_2.10  
GLIBC_2.11  
GLIBC_2.12  
GLIBC_2.13  
GLIBC_2.14  
GLIBC_2.15  
GLIBC_2.16  
GLIBC_2.17  
GLIBC_PRIVATE
```

3. 安装RabbitMQ

```
# 安装
rpm -ivh socat-1.7.3.2-1.1.el7.x86_64.rpm --force --nodeps

# 安装
rpm -ivh rabbitmq-server-3.6.5-1.noarch.rpm
```

4. 开启管理界面及配置

```
# 开启管理界面
rabbitmq-plugins enable rabbitmq_management

# 修改默认配置信息
vim /usr/lib/rabbitmq/lib/rabbitmq_server-3.6.5/ebin/rabbit.app

# 比如修改密码、配置等等，例如：loopback_users 中的 <<"guest">>,只保留guest
```

5. 启动

```
service rabbitmq-server start # 启动服务
service rabbitmq-server stop # 停止服务
service rabbitmq-server restart # 重启服务
```

- 设置配置文件

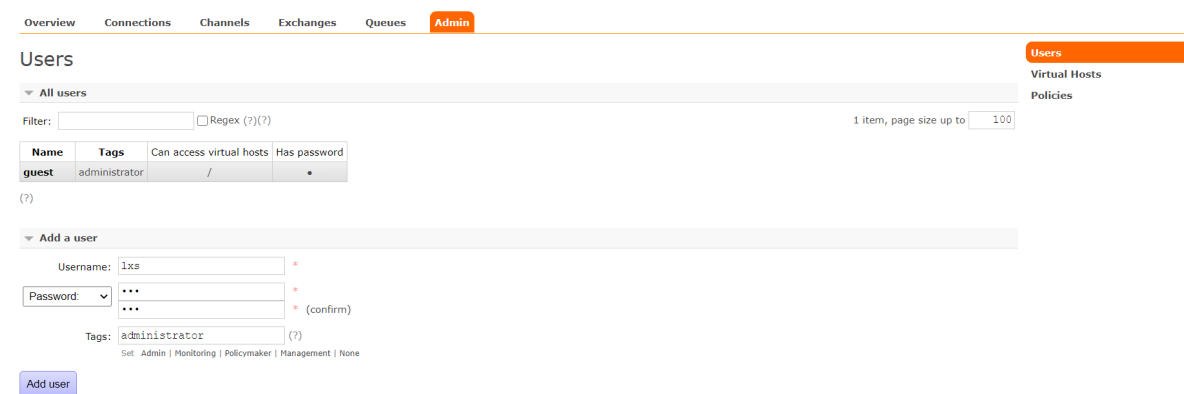
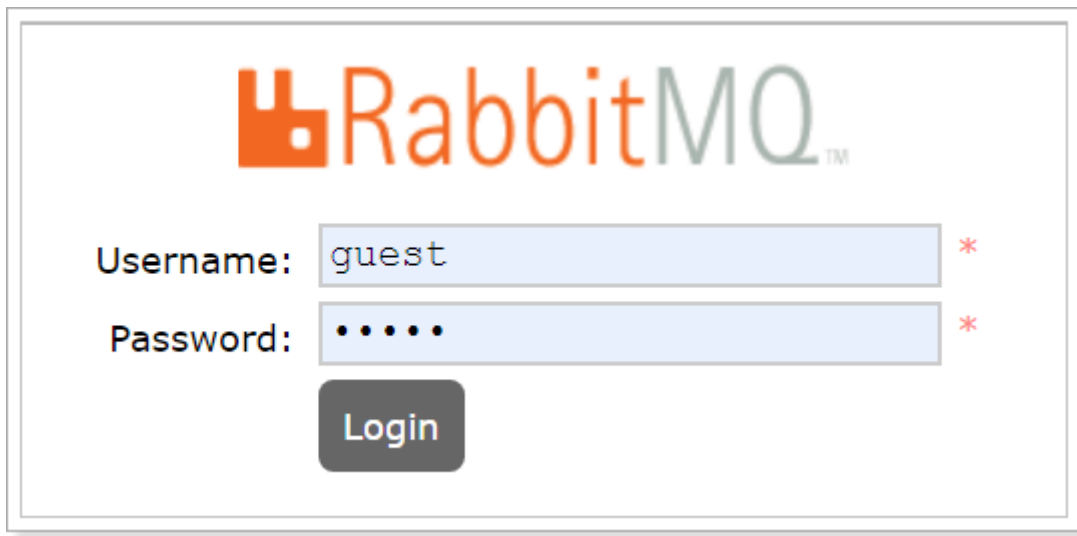
```
cd /usr/share/doc/rabbitmq-server-3.6.5/

cp rabbitmq.config.example /etc/rabbitmq/rabbitmq.config
```

6. 配置虚拟主机及用户

6.1. 用户角色

RabbitMQ在安装好后，可以访问 `http://ip地址:15672`；其自带了guest/guest的用户名和密码；如果需要创建自定义用户；那么也可以登录管理界面后，如下操作：



角色说明：

1、 超级管理员(administrator)

可登陆管理控制台，可查看所有的信息，并且可以对用户，策略(policy)进行操作。

2、 监控者(monitoring)

可登陆管理控制台，同时可以查看rabbitmq节点的相关信息(进程数，内存使用情况，磁盘使用情况等)

3、 策略制定者(policymaker)

可登陆管理控制台, 同时可以对policy进行管理。但无法查看节点的相关信息(上图红框标识的部分)。

4、 普通管理者(management)

仅可登陆管理控制台，无法看到节点信息，也无法对策略进行管理。

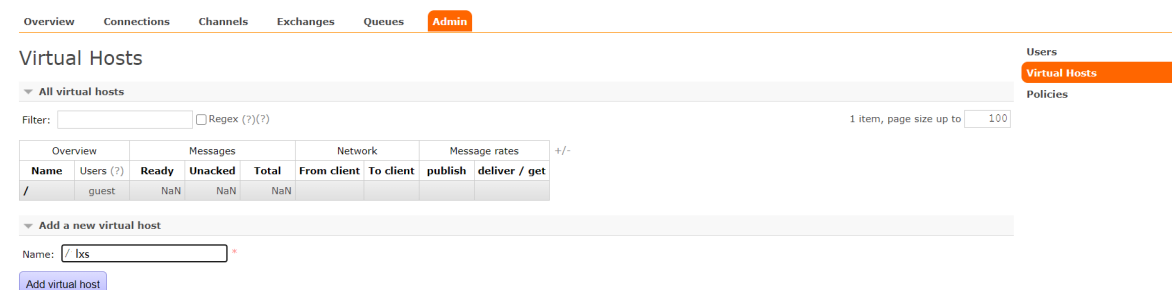
5、 其他

无法登陆管理控制台，通常就是普通的生产者和消费者。

6.2. Virtual Hosts配置

像mysql拥有数据库的概念并且可以指定用户对库和表等操作的权限。RabbitMQ也有类似的权限管理；在RabbitMQ中可以虚拟消息服务器Virtual Host，每个Virtual Hosts相当于一个相对独立的RabbitMQ服务器，每个VirtualHost之间是相互隔离的。exchange、queue、message不能互通。相当于mysql的db。Virtual Name一般以/开头。

6.2.1. 创建Virtual Hosts



Overview Connections Channels Exchanges Queues **Admin**

Virtual Hosts

▼ All virtual hosts

Filter: ☐ Regexp (?) 1 item, page size up to 100

Overview		Messages			Network		Message rates	
Name	Users (?)	Ready	Unacked	Total	From client	To client	publish	deliver / get
/	guest	NaN	NaN	NaN				

▼ Add a new virtual host

Name:

Add virtual host

6.2.2. 设置Virtual Hosts权限



Overview Connections Channels Exchanges Queues **Admin**

Cluster rabbit@itcast
User guest **Log out**

Virtual Hosts

▼ All virtual hosts

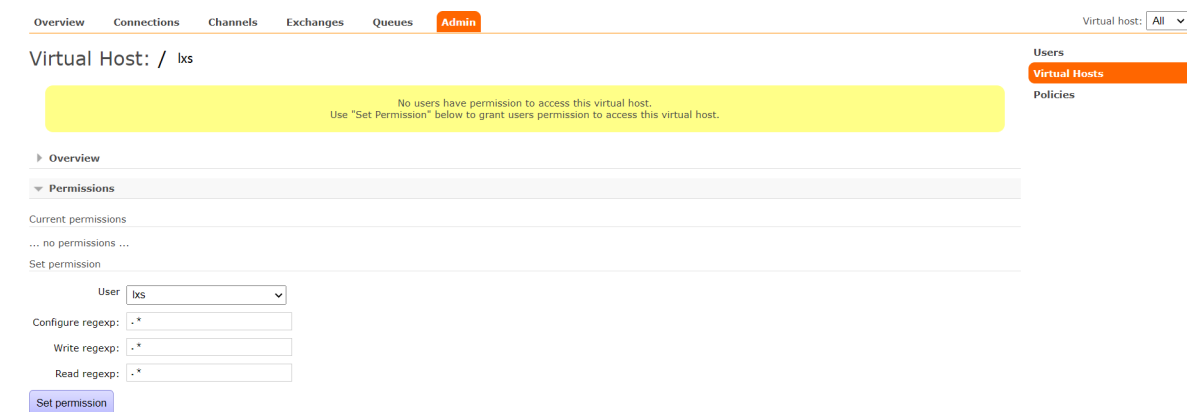
Filter: ☐ Regexp ? 2 items, page size up to 100

Overview		Messages			Network		Message rates		
Name	Users ?	State	Ready	Unacked	Total	From client	To client	publish	deliver / get
/	guest	running	NaN	NaN	NaN				
/ lxs	guest	running	NaN	NaN	NaN				

▼ Add a new virtual host

Name:

Add virtual host



Overview Connections Channels Exchanges Queues **Admin**

Virtual host: All ▼

Virtual Host: / lxs

No users have permission to access this virtual host.
Use "Set Permission" below to grant users permission to access this virtual host.

► Overview

▼ Permissions

Current permissions

... no permissions ...

Set permission

User

Configure regexp:

Write regexp:

Read regexp:

Set permission

7. docker 安装 rabbitmq

1.docker search rabbitmq

2.docker pull rabbitmq:版本号 -management
例如: docker pull rabbitmq:3.7.7-management

```
3.docker run -d --hostname my-rabbit --name rabbit -p 15672:15672 -p 5673:5672 rabbitmq
```

-d 后台运行容器；

--name 指定容器名；

-p 指定服务运行的端口（5672：应用访问端口；15672：控制台web端口号）；

-v 映射目录或文件；

--hostname 主机名（RabbitMQ的一个重要注意事项是它根据所谓的“节点名称”存储数据，默认为主机名）；

-e 指定环境变量；（RABBITMQ_DEFAULT_VHOST：默认虚拟机名；RABBITMQ_DEFAULT_USER：默认的用户名；

RABBITMQ_DEFAULT_PASS：默认用户名的密码）

4. 查看运行容器

```
docker ps
```

5. 进入容器内部

```
docker exec -it 容器id /bin/bash
```

6.http://自己的ip:15672/