|  |
| --- |
| 本帖最后由 craaazy123 于 2014-07-19 17:04 编辑  CentOS 6.4 32bit minimal下安装 第一种安装方式（源代码编译安装）： 1. 下载erlang源代码及RabbitMQ rpm安装包      $ wget [http://www.erlang.org/download/otp\_src\_R16B02.tar.gz](http://www.erlang.org/download/otp_src_R16B02.tar.gz" \t "http://bbs.chinaunix.net/_blank)       $ wget [http://www.rabbitmq.com/releases ... -3.2.0-1.noarch.rpm](http://www.rabbitmq.com/releases/rabbitmq-server/v3.2.0/rabbitmq-server-3.2.0-1.noarch.rpm" \t "http://bbs.chinaunix.net/_blank)  2. 编译安装erlang      解压otp\_src\_R16B02.tar.gz：     $ tar -zxvf  otp\_src\_R16B02.tar.gz      利用yum安装erlang编译环境：     $ yum -y install make ncurses-devel gcc gcc-c++ unixODBC unixODBC-devel openssl openssl-devel      配置erlang：     $ cd opt\_src\_R16B02     $ ./configure      \        --prefix=/usr/local/erlang     \        --enable-smp-support          \        --enable-threads                 \        --enable-sctp                      \        --enable-kernel-poll            \        --enable-hipe                      \        --with-ssl            编译安装erlang：      $ make && make install       3. 设置环境变量      $ vi /etc/profile            ERL\_HOME=/usr/local/erlang      PATH=$ERL\_HOME/bin: $PATH （*复制党注意*  :与$加了空格，主要是论坛自动替换成表情了）      export ERL\_HOME PATH       让环境变量立即生效：      $ source /etc/profile  4. 安装RabbitMQ     $ rpm -i --nodeps rabbitmq-server-3.2.0-1.noarch.rpm           注意：如果直接用rpm -i rabbitmq-server-3.2.0-1.noarch.rpm，会提示缺少erlang 13B的环境，前面安装了erlang 15B02所以      加上--nodeps则能安装成功。 5. 启动rabbitmq     $ rabbitmq-server start  6. 添加rabbitmq到启动项     $ chkconfig rabbitmq-server on  第二种安装方式（yum源安装）：  1. 安装配置epel源    $ rpm -Uvh [http://dl.fedoraproject.org/pub/ ... ease-6-8.noarch.rpm](http://dl.fedoraproject.org/pub/epel/6/i386/epel-release-6-8.noarch.rpm" \t "http://bbs.chinaunix.net/_blank)  2. 安装erlang    $ yum -y install erlang  3. 安装RabbitMQ    $ yum -y install rabbitmq-server    注意：用yum安装的版本可能不是最新的。如有需要，安装rabbitmq部分可以参照第一种方式。  4. 启动/停止rabbitmq服务器  IMG_256 |

配置web 访问：

缺省配置先启动看看：

# /sbin/service rabbitmq-server start

Starting rabbitmq-server: SUCCESS

* 1
* 2
* 1
* 2

如果启动报这个错误：epmd error for host “demo”: timeout ，那么只需要修改一下hosts文件，增加你的主机名，注意，比如你的主机叫 demo.woplus，在hosts中配了 127.0.0.1 demo.woplus 也不行，你需要在hosts中再加一个 127.0.0.1 demo。

## 二、配置

原本想直接先HelloWorld的，但是不行，如果没有做一些必要的权限配置，后面那个HelloWorld是跑不起来的。所以，先说说安装之后要做的几件事。

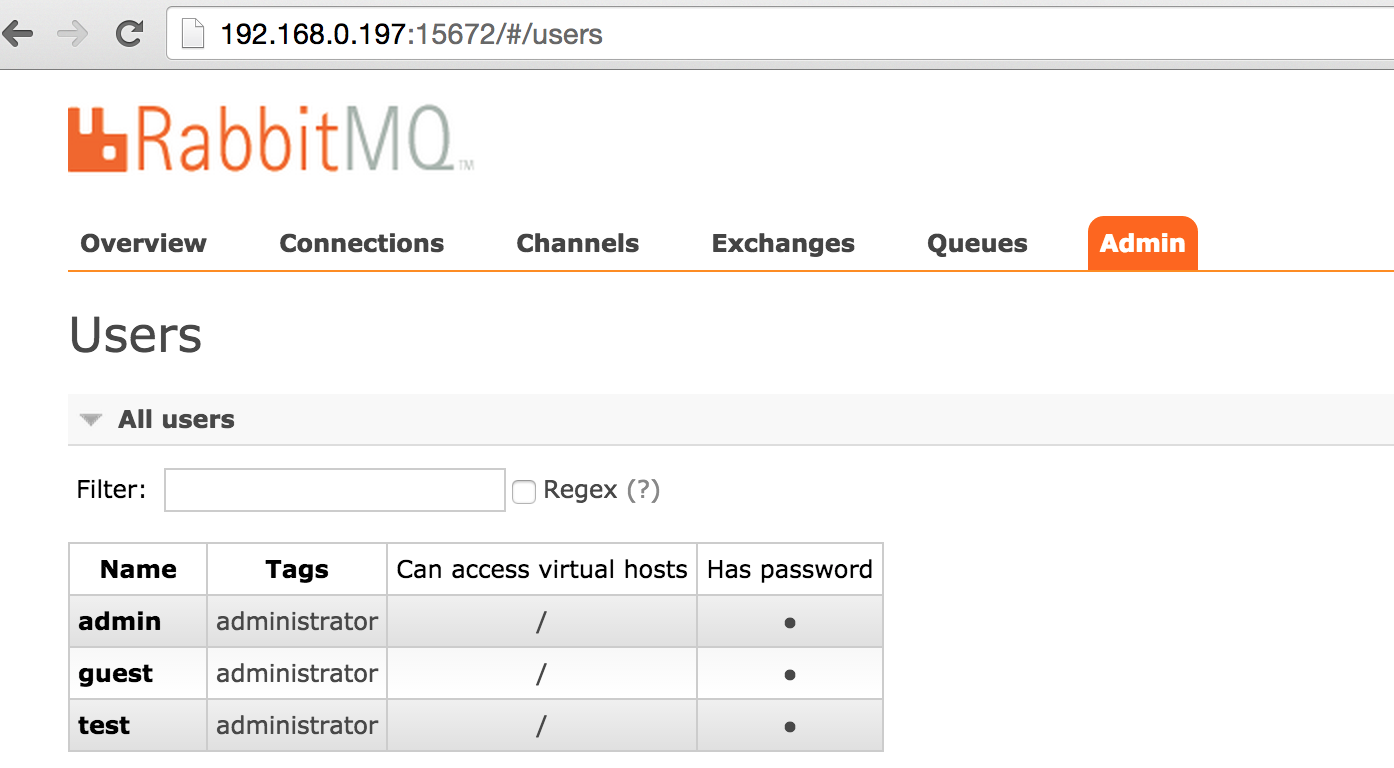
第一件事要创建用户，因为缺省的guest/guest用户只能在本地登录，所以先用命令行创建一个admin/admin123，并让他成为管理员。

# ./rabbitmqctl add\_user admin admin123#

./rabbitmqctl set\_user\_tags admin administrator

然后，我们启用WEB管理。

# ./rabbitmq-plugins enable rabbitmq\_management

现在可用使用浏览器访问管理台，就用刚才创建的admin登录即可，端口是15672。   


在管理台我们，我们需要为刚才的admin配置权限，见上图，点名字进去即可，conf/write/read 都配成缺省显示的那个.\* 即可。

然后我们新建一个用户 test/test123，一会儿用它来写HelloWorld。

我并没有做其它配置，这意味着我们使用的是缺省配置，基本够用了，如果以后有需要再修改吧，可以参考官网上的帮助：[http://www.rabbitmq.com/configure.html#config-items](http://www.rabbitmq.com/configure.html" \l "config-items" \t "http://blog.csdn.net/sharetop/article/details/_blank)

## 三、Java Client

现在可以来个 Say HelloWorld了。

我用Maven新建一个工程，修改POM文件，增加如下：

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>com.rabbitmq</groupId>

<artifactId>amqp-client</artifactId>

<version>3.5.6</version>

</dependency>

</dependencies>

使用官方的**[Java](http://lib.csdn.net/base/java" \o "Java 知识库" \t "http://blog.csdn.net/sharetop/article/details/_blank)**-client即可。然后代码如下：

public class App

{

private final static String QUEUE\_NAME = "hello";

static boolean isBreak =false;

public static void main( String[] args )

{

System.out.println( "Hello World!" );

ConnectionFactory factory = new ConnectionFactory();

factory.setUsername("test");

factory.setPassword("test123");

factory.setHost("192.168.0.197");

factory.setVirtualHost("/");

factory.setPort(5672);

try {

Connection connection = factory.newConnection();

Channel channel = connection.createChannel();

Consumer consumer = new DefaultConsumer(channel) {

@Override

public void handleDelivery(String consumerTag, Envelope envelope, AMQP.BasicProperties properties, byte[] body)

throws IOException {

String message = new String(body, "UTF-8");

System.out.println(" [x] Received '" + message + "'");

isBreak=true;

}

};

channel.basicConsume(QUEUE\_NAME, true, consumer);

channel.queueDeclare(QUEUE\_NAME, false, false, false, null);

String message = "Hello World!";

channel.basicPublish("", QUEUE\_NAME, null, message.getBytes());

System.out.println(" [x] Sent '" + message + "'");

while(!isBreak){

try {

Thread.sleep(1000);

} catch (InterruptedException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

channel.close();

connection.close();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (TimeoutException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

大概说明一下，ConnectionFactory先填上创建连接要的参数（主机地址、登录账户等），然后就是newConnection()得到Connection，并创建出一个频道Channel。   
从Channel我们可以构造出一个消费者Consumer用来监听收到的消息，也可以使用channel发布消息。RabbitMQ的消息是byte[]，所以需要将字串转成字节数组才能发送。