**Centos7部署RabbitMQ 3.7.6**

原

荐

[[阿dai](https://my.oschina.net/adailinux)](https://my.oschina.net/adailinux)

[阿dai](https://my.oschina.net/adailinux) 发布于 07/10 00:16

字数 3761

阅读 995

收藏 42

点赞 5

[评论 6](https://my.oschina.net/adailinux/blog/1843137#comments)

[RabbitMQ](https://my.oschina.net/adailinux?q=RabbitMQ)[CentOS](https://my.oschina.net/adailinux?q=CentOS)[Linux](https://my.oschina.net/adailinux?q=Linux)[Erlang](https://my.oschina.net/adailinux?q=Erlang)

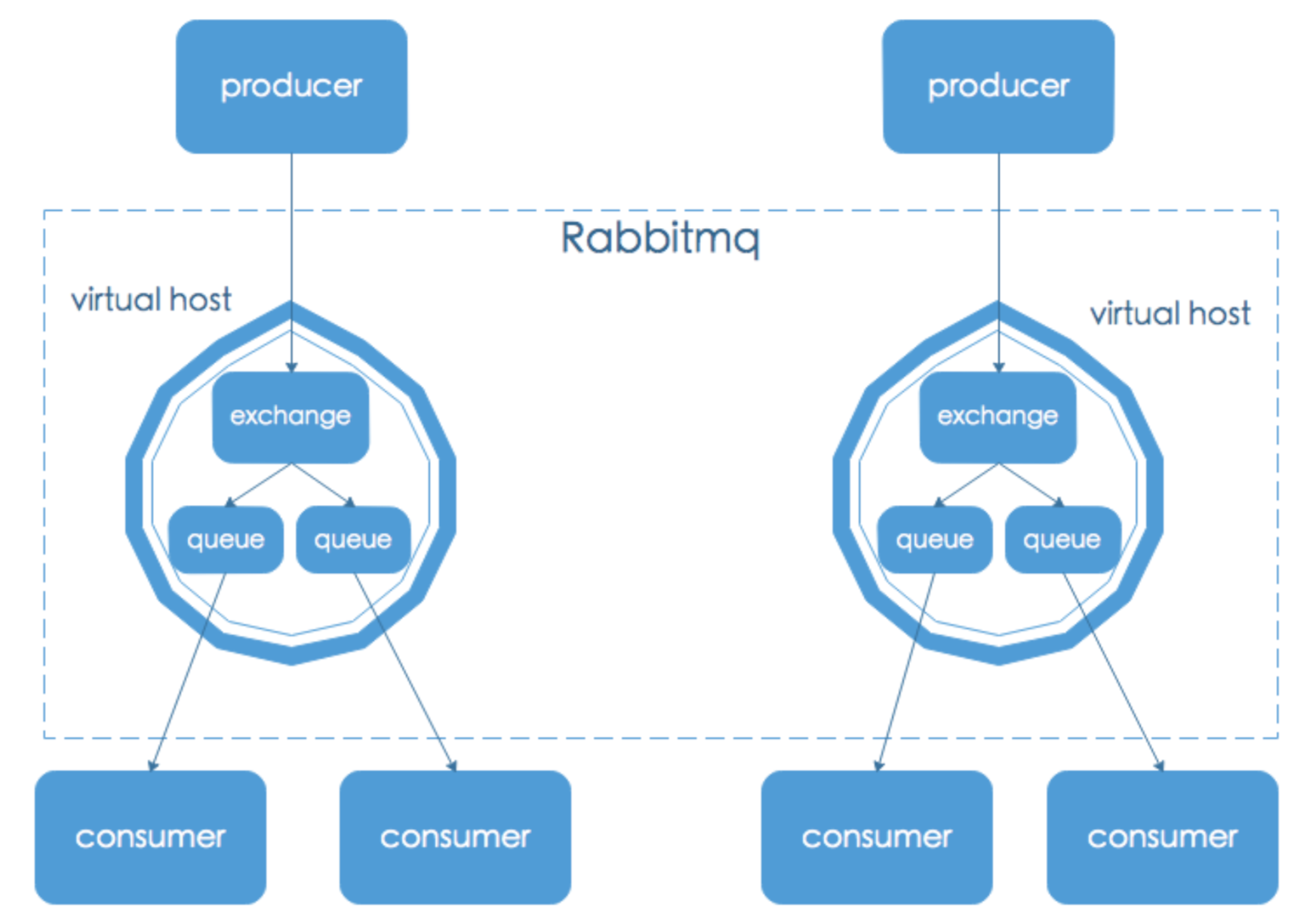
[**Kubernetes官方论坛来袭，开源中国购票通道开启全网最大优惠！ >>>**](https://my.oschina.net/u/2982602/blog/1982223)  https://www.oschina.net/img/hot3.png

简介

MQ全称为Message Queue, 消息队列（MQ）是一种应用程序对应用程序的通信方法。应用程序通过读写出入队列的消息（针对应用程序的数据）来通信，而无需专用连接来链接它们。消息传递指的是程序之间通过在消息中发送数据进行通信，而不是通过直接调用彼此来通信，直接调用通常是用于诸如远程过程调用的技术。排队指的是应用程序通过 队列来通信。队列的使用除去了接收和发送应用程序同时执行的要求。其中较为成熟的MQ产品有IBM WEBSPHERE MQ等等。

rabbitmq是采用Erlang（一种通用的面向并发的编程语言）编写的符合AMQP（Advanced Message Queuing Protocol）规范的消息中间件。诞生于金融行业，现在广泛应用于企业级的分布式应用中。

消息模型

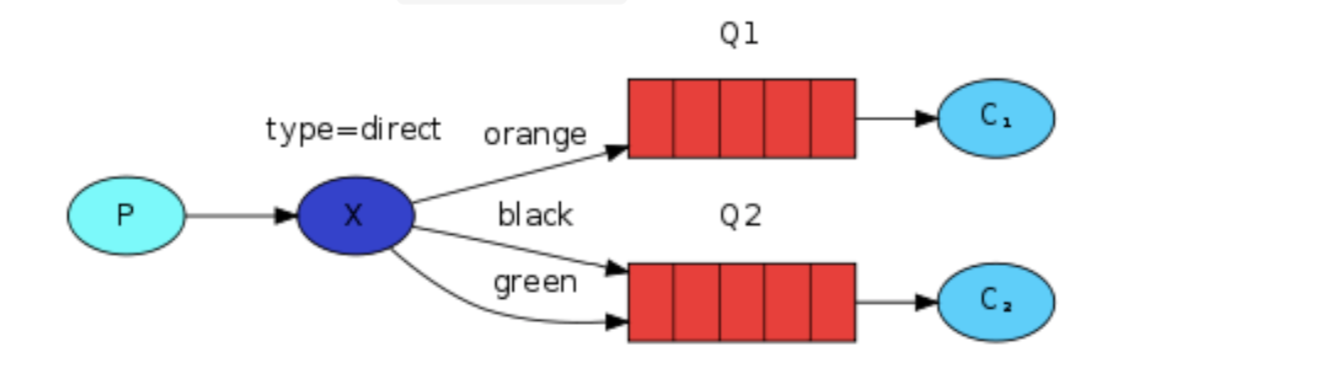
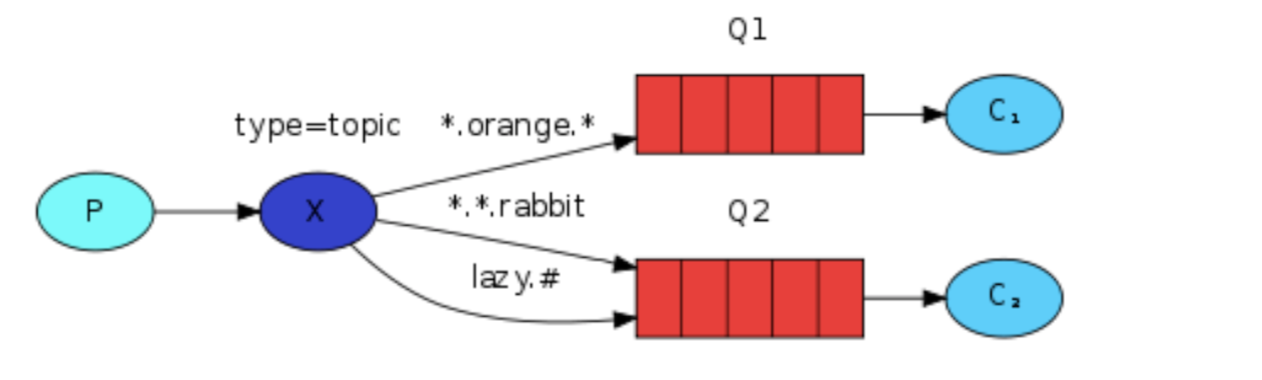


从模型抽象上来说其工作过程：生产者（producer）创建消息，然后发布到队列（queue）中，最后将消息发送到监听的消费者（consumer）。

* producer（publisher）：消息的生产者，也是一个向交换器发布消息的客户端应用程序。
* exchange：交换器，实现对消息的路由，生产者投递消息后最先到达exchange，由exchange决定将消息投递到那个queue，消息正确投递之前需要将exchange和queue进行绑定。
* queue：消息队列，用来保存消息直到发送给消费者。它是消息的容器，也是消息的终点。一个消息可投入一个或多个队列。消息一直在队列里面，等待消费者连接到这个队列将其取走。
* virtual host：虚拟主机，表示一批交换器、消息队列和相关对象。虚拟主机是共享相同的身份认证和加密环境的独立服务器域。每个 vhost 本质上就是一个 mini 版的 RabbitMQ 服务器，拥有自己的队列、交换器、绑定和权限机制。vhost 是 AMQP 概念的基础，必须在连接时指定，RabbitMQ 默认的 vhost 是 / 。
* consumer：消息的消费者，表示一个从消息队列中取得消息的客户端应用程序。

exchange模式

Exchange分发消息的类型有四种：

* Fanout：即广播模式，exchange收到消息后把消息投递到每个和它绑定的队列。
* direct：消息按照指定的routing key（exchange和queue绑定时可指定）直接投递到队列 
* topic：复杂的发布/订阅模式。routing key由.号分隔的字符串指定，如quick.orange.rabbit，每个分隔的字符串最好能对应一定的业务含义。支持模糊匹配，如\*.orange.rabbit或如quick.orange.#。\*精确匹配一个字符串，#匹配0个或多个字符串。 
* header：按照消息头投递到队列，不考虑routing key。

关于本节内容可参考：<https://www.jianshu.com/p/79ca08116d57>

安装RabbitMQ

系统环境：centos 7.3

内网ip：192.168.228.130

安装erlang

$ yum install -y erlang

此处直接yum安装的版本为erlang-R16B-03.18.el7.x86\_64 在后面安装Rabbitmq时会报错：

错误：软件包：rabbitmq-server-3.7.6-1.el7.noarch (/rabbitmq-server-3.7.6-1.el7.noarch)

需要：erlang >= 19.3

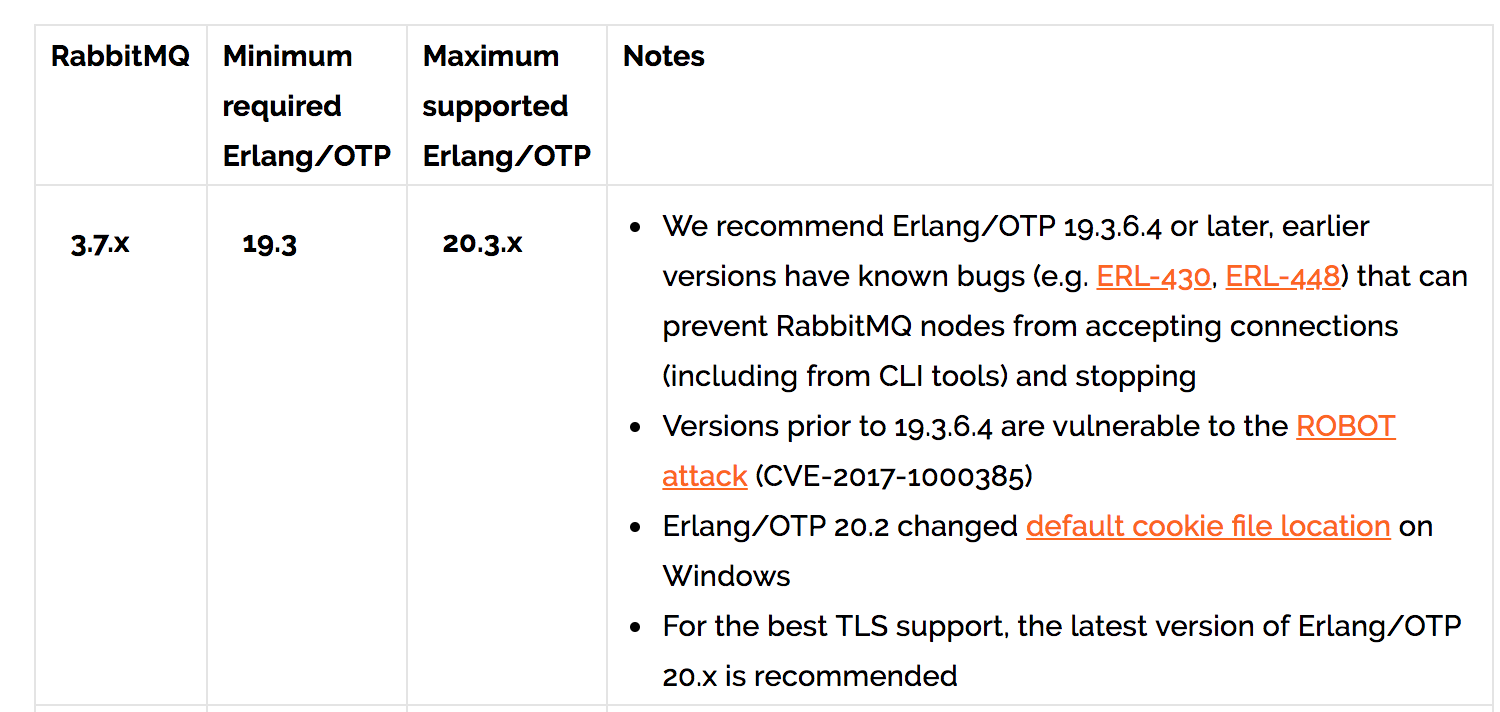
已安装: erlang-R16B-03.18.el7.x86\_64 (@epel)

erlang = R16B-03.18.el7

您可以尝试添加 --skip-broken 选项来解决该问题

您可以尝试执行：rpm -Va --nofiles --nodigest

* 原因是：rabbitmq-server-3.7.6依赖erlang版本要19.3=<erlang<=20.3.x（第一次根据官方提供的方法安装的21.0版本——成功入坑，详情继续向下看）。
* 解决办法（**此处有坑**）：
* # 下载erlang官方镜像源
* $ wget https://packages.erlang-solutions.com/erlang-solutions-1.0-1.noarch.rpm
* # 配置镜像源
* $ rpm -Uvh erlang-solutions-1.0-1.noarch.rpm
* # 安装erlang
* $ yum install -y erlang
* # 查看版本
* $ rpm -q erlang
* erlang-21.0-1.el7.centos.x86\_64 # 即安装的为21.0版本
* 再次入坑（使用上述办法安装erlang，在手动添加rabbitmq配置文件后），报错信息：
* Could not start application logger: Logger.App.start(:normal, []) returned an error: shutdown: failed to start child: Logger.ErrorHandler
* \*\* (EXIT) no process: the process is not alive or there's no process currently associated with the given name, possibly because its application isn't started

按官方提供的方法安装好erlang后，当手动添加rabbitmq配置文件后再次启动出现上面的报错信息，经过分析发现（太坑了）： 

* 再次爬坑： 根据上图中的提示，只能再找办法安装19.3~20.3.x之间版本的erlang了，但是官方提供的方法只能安装最新版的erlang，经过一番搜索找到如下办法可以安装19.3~20.3.x之间版本的erlang：
* # 手动配置erlang的repo源
* # In /etc/yum.repos.d/rabbitmq-erlang.repo
* [rabbitmq-erlang]
* name=rabbitmq-erlang
* baseurl=https://dl.bintray.com/rabbitmq/rpm/erlang/20/el/7
* gpgcheck=1
* gpgkey=https://dl.bintray.com/rabbitmq/Keys/rabbitmq-release-signing-key.asc
* repo\_gpgcheck=0
* enabled=1
* # 清理原有rpm包缓存
* $ yum clean all
* # 重新安装20.3版本的erlang
* $ yum install -y erlang
* # 查看包状态
* $ rpm -q erlang
* erlang-20.3-1.el7.centos.x86\_64 # 完成！
  + 更多版本的安装包请参考：<http://packages.erlang-solutions.com/rpm/centos/7/x86_64/> (感谢楼主)
* **注意1：** 因为之前安装过旧版本的erlang会导致冲突，所以在安装新版本erlang时需要将旧版本erlang有关的包全部卸载掉yum remove -y erlang（使用rpm -e erlang --nodeps只能卸载erlang包，无法卸载其相关的依赖包）。
* **注意2：** 如果你之前像我一样根据官方提示安装过erlang-21.0，那么肯定也入坑了，爬坑过程中卸载erlang的时候注意观察会发现rabbitmq-server会同时被卸载，所以安装好erlang-20.3后需要再次安装rabbitmq-server，方法如下。

参考erlang官方文档：<https://www.erlang-solutions.com/resources/download.html>

如果你安装的是低版本的rabbitmq会比较省事，官方提供了erlang的rpm包：<http://www.rabbitmq.com/releases/erlang/>

安装RabbitMQ Server

# download rpm.repo

$ wget https://dl.bintray.com/rabbitmq/all/rabbitmq-server/3.7.6/rabbitmq-server-3.7.6-1.el7.noarch.rpm

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

# this example assumes the CentOS 7 version of the package

$ yum install -y rabbitmq-server-3.7.6-1.el7.noarch.rpm

# Done!

$ rpm -q rabbitmq-server

rabbitmq-server-3.7.6-1.el7.noarch

如果报错请返回 **“erlang 安装”** 。

管理服务

* centos7可以直接使用系统工具管理服务
* $ systemctl start/status/restart/stop rabbitmq-server
* # 查看rabbimq启动的端口
* $ netstat -lntp
* tcp 0 0 0.0.0.0:4369 0.0.0.0:\* LISTEN 96803/epmd
* tcp 0 0 0.0.0.0:25672 0.0.0.0:\* LISTEN 96660/beam.smp
* tcp6 0 0 :::4369 :::\* LISTEN 96803/epmd
* tcp6 0 0 :::5672 :::\* LISTEN 96660/beam.smp

参考rabbitmq官方文档：<http://www.rabbitmq.com/install-rpm.html>

RabbitMQ文件位置及配置管理

文件位置

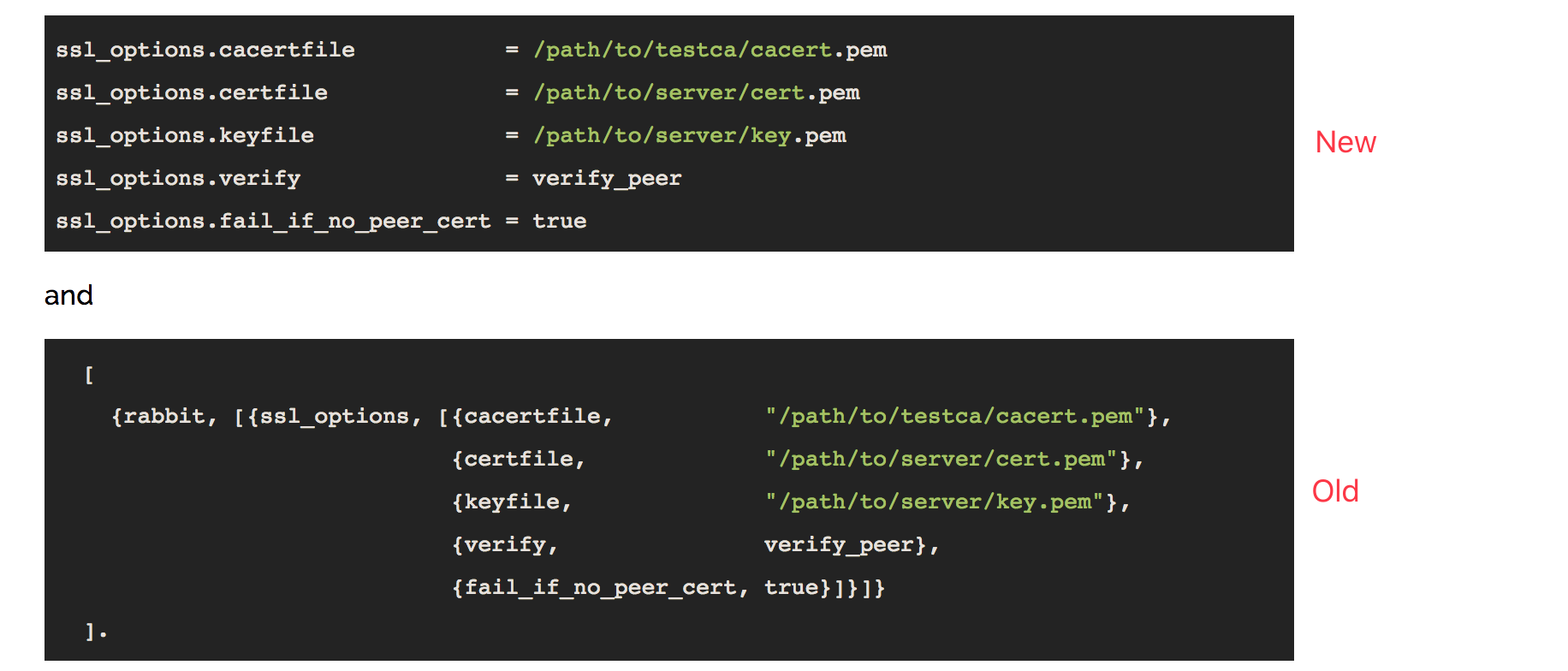
* 文件结构：
* /var/lib/rabbitmq/
* ├── config # 存放配置文件
* ├── mnesia # 存放节点的数据、信息、状态、日志等文件
* └── schema # 存放rabbitmq的纲要文件
  + 更多文件结构信息相关内容可参考：<http://www.rabbitmq.com/relocate.html>
* 怎么寻找RabbitMQ配置文件位置:
  + 每个节点的配置信息都可以在log\_file（首行）找到，内容如下
  + node : rabbit@host3
  + home dir : /var/lib/rabbitmq
  + config file(s) : (none) # 在此发现rabbit@host1这个节点并没有配置文件(后面补充)
  + cookie hash : FFGcfZmAr6Q+pjUEHEiUEw==
  + log(s) : /var/log/rabbitmq/rabbit@host3.log
  + : /var/log/rabbitmq/rabbit@host3\_upgrade.log
  + database dir : /var/lib/rabbitmq/mnesia/rabbit@host3
  + 可以再Management\_UI界面找到，前提是开启UI插件，开启方法如下：
  + $ rabbitmq-plugins enable rabbitmq\_management
    - 默认监听端口：15672（在3.0版本之前是55672端口）
    - 访问方法：http://*server-name*:15672/
    - 默认用户/密码：guest/guest
* 关于Management\_UI用户类型及权限管理：
  + none：没有任何权限
  + management：进行符合AMQP协议的所有操作，以及查看所有virtual\_host，查看所有的exchange、queue，管理所binding的virtual\_host、所有的channel和connections；
  + policymaker：所有“management”角色能做的事，以及对所有该用户能登录的virtual\_host进行策略的写改删查；
  + monitoring：所有“management”角色能做的事，以及查看所有virtual\_host、所有用户的全部信息；
  + administrator：为所欲为！
  + 以上请参考：<http://www.rabbitmq.com/management.html>

配置管理

官方关于如何分配服务器资源的 **建议** ：<http://www.rabbitmq.com/production-checklist.html>

* virtual\_host分配：
  + 单机单实例：使用默认virtual\_host即可；
  + 单机多实例：使用独立的virtual\_host， \*\*e.g. \*\* project1\_development, project1\_production, project2\_development, project2\_production...
* 用户管理：
  + 删除默认用户guest；
  + 禁止远程连接，单独创建一个配有复杂密码的管理用户；
  + 为每个APP配置单独的用户及密码；
* 资源控制和监控
* 内存管理及优化建议：
  + rabbitmq默认在超出可用内存的40%后就不接受任何message，配置参数{vm\_memory\_high\_watermark, 0.4}；
  + 每个节点至少分配128M内存；
  + vm\_memory\_high\_watermark范围为0.4~0.66；
  + 更多参考：<http://www.rabbitmq.com/memory.html>
* 磁盘空间管理建议：
  + 默认的50M磁盘空间就可以顺利运行，配置参数disk\_free\_limit；
  + 针对特殊的环境进行个性化配置，磁盘空间不足会导致节点故障；
  + 为保证节点正常运行，生产环境内存和磁盘资源关系需要配置告警：
    - {disk\_free\_limit, {mem\_relative, 1.0}}，即当剩余的磁盘空间和分配的内存空间相同时告警，这是保证节点正常运行的最低内存、磁盘空间比例；
    - {disk\_free\_limit, {mem\_relative, 1.5}}，当剩余磁盘空间为分配的内存空间的1.5倍时告警，这样配置比较安全；
    - {disk\_free\_limit, {mem\_relative, 2.0}}，当剩余磁盘空间为分配的内存空间的2倍时告警，这样配置最谨慎。
* fd控制：
  + 确保rabbitmq最少可用fd数量为50000个；
  + fd限制数量算法：并发数\*95%\*2+总队列数
  + 生产环境可以配置为500000，不会占用太多硬件资源。
* 集群相关重要内容：
  + 集群规模
  + 集群隔离控制策略
  + 节点时间同步
* 本节更多相关内容请参考：<http://www.rabbitmq.com/production-checklist.html>

创建配置文件

自3.7.0版本开始，rabbitmq的配置文件命名为rabbitmq.config，是一种标准的erlang配置文件，同时也兼容旧版本的配置文件（advanced.config）格式，与旧格式的配置文件相比，新格式配置文件更简介、更清晰、更易于管理，且可以使用Chef、Puppet等自动化部署工具生成。新旧版本配置文件对比**e.g.**：

新版配置文件格式的结构可以总结为以下三点：

* 每个单一的配置信息都在一行；
* 每行的结构为key = value模式；
* 以“#”开头的为注释信息。

第一个rabbitmq.config

listeners.tcp.default = 5673

或，

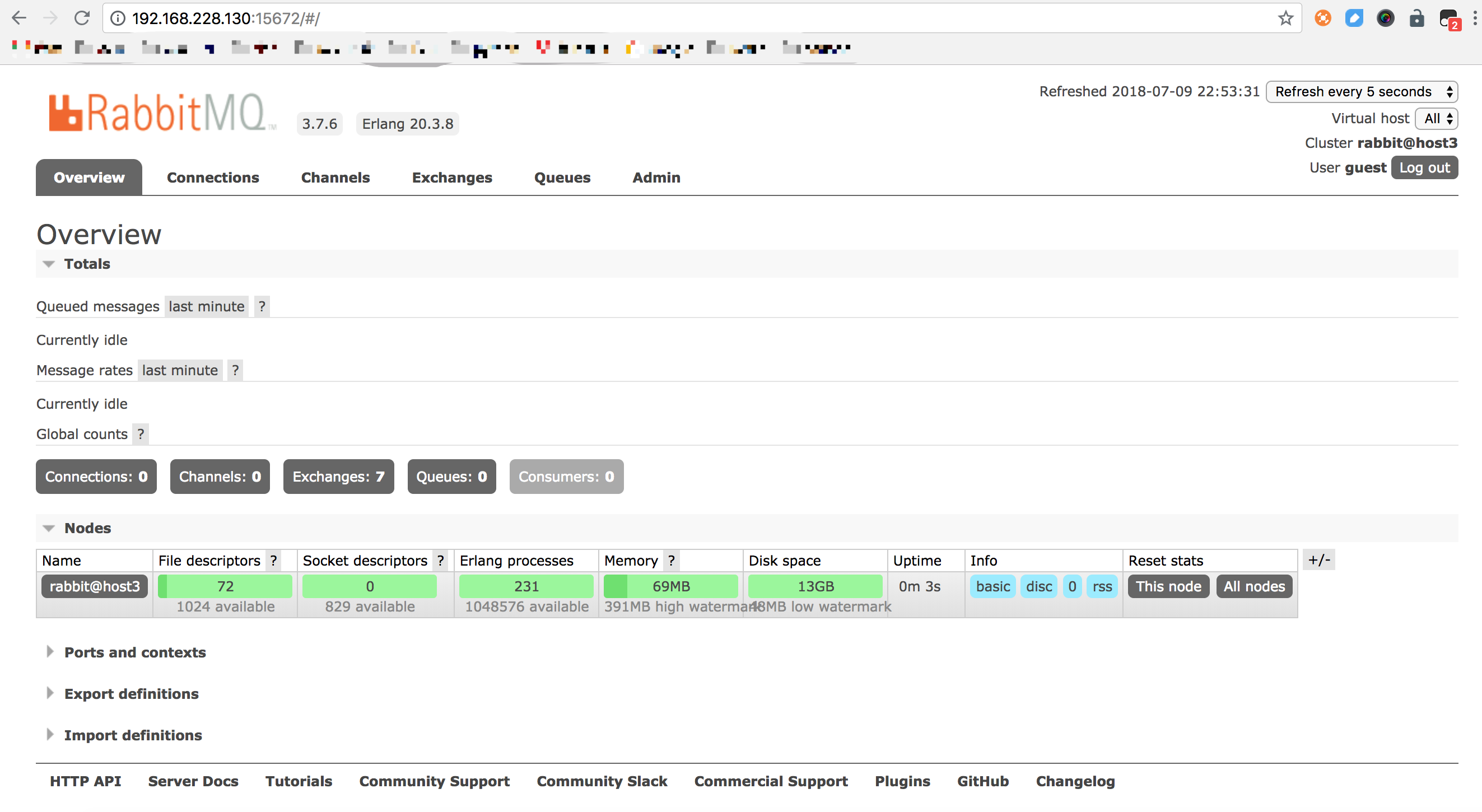
[

{rabbit, [{tcp\_listeners, [5673]}]}

].

* 配置完成后启动rabbitmq：
* # 重启
* $ systemctl restart rabbitmq-server
* # 查看端口
* $ netstat -lntp
* tcp6 0 0 :::5673 :::\* LISTEN 44435/beam.smp
* # 查看日志
* $ less /var/log/rabbitmq/rabbit@host3.log
* 2018-07-09 21:15:37.692 [info] <0.215.0>
* node : rabbit@host3
* home dir : /var/lib/rabbitmq
* config file(s) : /etc/rabbitmq/rabbitmq.conf # 此时新配置文件生效
* cookie hash : p40SfKMD4r8HemoR0EOJWw==
* log(s) : /var/log/rabbitmq/rabbit@host3.log
* : /var/log/rabbitmq/rabbit@host3\_upgrade.log
* database dir : /var/lib/rabbitmq/mnesia/rabbit@host3
* 自定义rabbitmq的配置文件路径：
* $ vim /usr/lib/systemd/system/rabbitmq-server.service
* [Unit]
* Description=RabbitMQ broker
* After=syslog.target network.target
* [Service]
* Type=notify
* User=rabbitmq
* Group=rabbitmq
* UMask=0027
* NotifyAccess=all
* TimeoutStartSec=3600
* # RABBITMQ\_CONFIG\_FILE=/etc/rabbitmq/rabbitmq.conf # 指定配置文件
* # Un-comment this setting if you need to increase RabbitMQ's
* # open files limit
* # LimitNOFILE=16384
* #
* # Note: systemd on CentOS 7 complains about in-line comments,
* # so only append them here
* #
* # Restart:
* # The following setting will automatically restart RabbitMQ
* # in the event of a failure. systemd service restarts are not a
* # replacement for service monitoring. Please see
* # http://www.rabbitmq.com/monitoring.html
* Restart=on-failure
* RestartSec=10
* WorkingDirectory=/var/lib/rabbitmq
* ExecStart=/usr/sbin/rabbitmq-server
* ExecStop=/usr/sbin/rabbitmqctl shutdown
* # See rabbitmq/rabbitmq-server-release#51
* SuccessExitStatus=69
* [Install]
* WantedBy=multi-user.target
* 配置文件详解：
  + 常用参数详解：<http://www.rabbitmq.com/configure.html>
  + 示例文件详解：<https://github.com/rabbitmq/rabbitmq-server/blob/master/docs/rabbitmq.conf.example>
* 说明：
  + rabbitmq-env.conf和rabbitmq.conf都可以手动创建；
  + 更改配置文件后，重启服务生效。

web管理界面

* 开启管理插件：
* $ rabbitmq-plugins enable rabbitmq\_management
* <输出内容>
* The following plugins have been configured:
* rabbitmq\_management
* rabbitmq\_management\_agent
* rabbitmq\_web\_dispatch
* Applying plugin configuration to rabbit@host3...
* The following plugins have been enabled:
* rabbitmq\_management
* rabbitmq\_management\_agent
* rabbitmq\_web\_dispatch
* started 3 plugins.
* <开启成功>
* 访问web界面：（默认端口为15672）[http://server\_ip:15672](http://server_ip:15672/) 
* 为guest用户授权：loopback\_users = none （默认只能是本机访问，**localhost或者127.0.0.1**，从外部访问需要添加该配置）
  + **访问控制：** <https://www.rabbitmq.com/access-control.html>
* 重启后访问，登录管理页面： 
* 管理页面能做的操作：
* list and delete exchanges, queues, bindings, users, virtual hosts and permissions.
* Monitor queue length, message rates globally and per channel, data rates per connection, etc.
* Monitor resource usage, such as file descriptors, memory use, available disk space.
* Manage users (provided administrative permissions of the current user).
* Export and import object definitions (vhosts, users, permissions, queues, exchanges, bindings, parameters, policies) to JSON.
* Force close connections, purge queues.
* Send and receive messages (useful in development environments and for troubleshooting).

命令行管理

因为guest为默认用户，对外开放并不安全，但是通过localhost来访问又不方便（通过配置代理也可以实现外网访问），所以在此还可以使用命令行进行管理，在此准备了两种管理工具：rabbitmqctl、rabbitmqadmin（两者不冲突）。

方法1——rabbitmqctl：

rabbitmqctl为rabbitmq内置管理工具，功能很多，详细的使用方法可以参考下面的链接：<http://www.rabbitmq.com/man/rabbitmqctl.8.html>。

**用法示例：**

* 添加用户：
* # rabbitmqctl add\_user <username> <newpassword>
* $ rabbitmqctl add\_user adai 123456
* Adding user "adai" ...
* 查看用户信息：
* $ rabbitmqctl list\_users
* Listing users ...
* adai []
* guest [administrator]
* 修改用户密码：
* # rabbitmqctl change\_password <username> <newpassword>
* $ rabbitmqctl change\_password adai 111111
* Changing password for user "adai" ...
* 为用户授权：
* # 格式： rabbitmqctl set\_permissions [-pvhostpath] {user} {conf} {write} {read}
* ## vhost The name of the virtual host to which to grant the user access, defaulting to "/".
* ## user The name of the user to grant access to the specified virtual host.
* ## conf A regular expression matching resource names for which the user is granted configure permissions.
* ## write A regular expression matching resource names for which the user is granted write permissions.
* ## read A regular expression matching resource names for which the user is granted read permissions.
* # 示例：授权给用户adai默认vhost("/")的所有配置的读、写权限
* $ rabbitmqctl set\_permissions -p "/" adai ".\*" ".\*" ".\*"
* Setting permissions for user "adai" in vhost "/" ...
* 查看用户权限：
* # rabbitmqctl list\_permissions [-p vhost]
* $ rabbitmqctl list\_permissions -p /
* Listing permissions for vhost "/" ...
* guest .\* .\* .\*
* adai .\* .\* .\*
* 清理用户权限：
* # rabbitmqctl clear\_permissions [-p vhost] username
* $ rabbitmqctl clear\_permissions -p / adai
* Clearing permissions for user "adai" in vhost "/" ...
* $ rabbitmqctl list\_permissions -p /
* Listing permissions for vhost "/" ...
* guest .\* .\* .\*
* 查看某用户的权限：
* $ rabbitmqctl list\_user\_permissions adai
* Listing permissions for user "adai" ...
* $ rabbitmqctl set\_permissions -p "/" adai ".\*" ".\*" ".\*"
* Setting permissions for user "adai" in vhost "/" ...
* $ rabbitmqctl list\_user\_permissions adai
* Listing permissions for user "adai" ...
* / .\* .\* .\*
* 设置用户角色（上面有讲，共五种角色类型）：
* $ rabbitmqctl set\_user\_tags adai administartor # 将adai设置为管理员
* Setting tags for user "adai" to [administartor] ...
* $ rabbitmqctl list\_users
* Listing users ...
* adai [administartor]
* guest [administrator]
* 删除用户：
* # rabbitmqctl delete\_user <username>
* $ rabbitmqctl add\_user test monitoring
* Adding user "test" ...
* $ rabbitmqctl list\_users
* Listing users ...
* adai [administartor]
* guest [administrator]
* test []
* $ rabbitmqctl delete\_user test
* Deleting user "test" ...
* $ rabbitmqctl list\_users
* Listing users ...
* adai [administartor]
* guest [administrator]

方法2——rabbitmqadmin：

* 该工具来自第三方，获取方法：
  + 如果你安装了UI\_management插件，可以直接通过下面方法获取
  + wget http://server\_ip:15672/cli/rabbitmqadmin
  + # 注意：
  + ## The tool supports：
  + ## Python 3.x
  + ## Python 2.6 or later for HTTP connections
  + ## Python 2.7.9 or later for HTTPS connections
  + 还可以通过github仓库获取，地址<https://raw.githubusercontent.com/rabbitmq/rabbitmq-management/v3.7.6/bin/rabbitmqadmin>
* 获取rabbitmqadmin后，将其加入系统环境：/usr/local/bin
* mv rabbitmqadmin /usr/local/bin/
* # 赋予其执行权限
* chmod 755 /usr/local/bin/rabbitmqadmin
* # 激活rabbitmqadmin命令
* rabbitmqadmin --help
* **注意：** rabbitmqadmin的版本要好rabbitmq的版本对应，否则无法使用。
* rabbitmqadmin工具支持在bash中以列表的输出信息:
* # 开启rabbitmqadmin的列表支持功能
* rabbitmqadmin --bash-completion > /etc/bash\_completion.d/rabbitmqadmin
* 详细用法参考：<http://www.rabbitmq.com/management-cli.html>