名称

rabbitmqctl -
用于管理RabbitMQ节点的工具

概要

rabbitmqctl [-q] [[-s] [[-l] [[-n *节点*] [[-t *超时*] *命令* [*COMMAND_OPTIONS*]

描述

RabbitMQ是一个开源的多协议消息代理。
rabbitmqctl是一个用于管理RabbitMQ服务器节点的命令行工具。它通过在专用CLI工具通信端口上连接到目标RabbitMQ节点并使用共享密钥（称为cookie文件）进行身份验证来执行所有操作。如果连接失败，目标节点未运行或rabbitmqctl无法成功向目标节点进行身份验证，则会显示诊断信息。要了解更多信息，请参阅 RabbitMQ CLI工具指南和 RabbitMQ网络指南

OPTIONS

-n *节点*
: 默认节点是“ *rabbit @ target-hostname* ”，其中*target-hostname*是本地主机。在名为“myserver.example.com”的主机上，节点名称通常为“rabbit @ myserver”（除非RABBITMQ_NODENAME已被覆盖）。“hostname -s”的输出通常是在“@”符号后使用的正确后缀。有关 配置RabbitMQ节点的详细信息，请参阅 rabbitmq-server（8）。

-q, --quiet
: 选择安静的输出模式。安静模式生效时，信息消息会减少。

-s, --silent
: 选择静音输出模式。当静默模式生效时，信息消息会减少，表头会被抑制。

--no-table-headers
: 不要为表格数据输出标题。

--dry-run
: 不要运行命令。仅打印信息消息。

-t *超时*, --timeout *超时*
: 操作超时（秒）。并非所有命令都支持超时。默认是infinity。

-l, --longnames

必须在将群集配置为使用长（FQDN）节点名称时指定。要了解更多信息，请参阅 RabbitMQ群集指南

--erlang-cookie *曲奇饼*

用于向目标节点进行身份验证的共享密钥。首选使用本地文件或 RABBITMQ_ERLANG_COOKIE环境变量，而不是在命令行上指定此选项。要了解更多信息，请参阅 RabbitMQ CLI工具指南

COMMANDS

help [
-l
] [[
COMMAND_NAME
]

打印所有可用命令的用法。

-l, --list-commands

List命令仅用于命令，不带参数说明。

COMMAND_NAME

打印指定命令的用法。

应用管理

force_reset

强制将RabbitMQ节点返回到其原始状态。

该force_reset命令的不同之处在于 reset，无论当前管理数据库状态和群集配置如何，它都无条件地重置节点。如果数据库或群集配置已损坏，则只应将其用作最后的手段。

对于reset并 force_reset获得成功的RabbitMQ的应用程序必须已经停止，例如用 stop_app。

例如，要重置RabbitMQ节点：

```
rabbitmqctl force_reset
```

hipe_compile *目录*

在给定目录中执行HiPE编译并缓存生成的 .beam -files。

如有必要，将创建父目录。在编译之前，将自动删除目录中的任何现有 .beam文件。

要使用此预编译文件，应将RABBITMQ_SERVER_CODE_PATH环境变量设置 为hipe_compileinvokation中指定的目录 。

例如，对HiPE-compile模块并将它们存储到 / tmp / rabbit-hipe / ebin目录：

```
rabbitmqctl hipe_compile /tmp/rabbit-hipe/ebin
```

reset

将RabbitMQ节点返回到其原始状态。

从其所属的任何群集中删除该节点，从管理数据库中删除所有数据，例如已配置的用户和vhost，并删除所有持久性消息。

对于reset并 force_reset获得成功的RabbitMQ的应用程序必须已经停止，例如用 stop_app。

例如，要重置RabbitMQ节点：

```
rabbitmqctl reset
```

rotate_logs

指示RabbitMQ节点执行内部日志轮换。

根据配置文件中指定的更大设置执行日志轮换。

请注意，在外部日志轮换的情况下无需调用此命令（例如，从logrotate（8）），因为lager检测到重命名并自动重新打开日志文件。

例如，此命令启动内部日志轮换过程：

```
rabbitmqctl rotate_logs
```

旋转是异步执行的，因此无法保证在此命令返回时它将完成。

shutdown

关闭节点，RabbitMQ及其运行时。该命令是阻塞的，并将在运行时进程退出后返回。如果RabbitMQ无法停止，它将返回非零退出代码。此命令推断目标节点的OS PID，因此只能用于关闭在同一主机上运行的节点（或者广泛地说，在同一操作系统中，例如在同一个VM或容器中）
与stop命令不同，shutdown命令：

- 不需要*pid_file*来等待运行时进程退出
- 如果RabbitMQ节点未运行，则返回非零退出代码

例如，这将使用默认节点名关闭本地运行的RabbitMQ节点：

```
rabbitmqctl shutdown
```

start_app

启动RabbitMQ应用程序。
此命令通常在执行需要停止RabbitMQ应用程序的其他管理操作后运行，例如 reset。
例如，要指示RabbitMQ节点启动RabbitMQ应用程序：

```
rabbitmqctl start_app
```

stop [*pid_file*]

停止运行RabbitMQ的Erlang节点。要重新启动节点，请按照安装指南中的“运行服务器”的说明进行操作 。
如果指定了*pid_file*，则还要等待指定的进程终止。有关wait此文件的详细信息，请参阅该 命令的说明。
例如，要指示RabbitMQ节点终止：

```
rabbitmqctl stop
```

stop_app

停止RabbitMQ应用程序，让runtme（Erlang VM）运行。
该命令通常在执行需要停止RabbitMQ应用程序的其他管理操作之前运行，例如 reset。
例如，要指示RabbitMQ节点停止RabbitMQ应用程序：

```
rabbitmqctl stop_app
```

wait *pid_file*， wait --pid *pid*

等待RabbitMQ应用程序启动。
此命令将等待RabbitMQ应用程序从节点开始。如果指定了*pidfile*，它将等待创建pid文件，然后等待pid文件或--pid参数中指定了pid的进程，然后在该进程中启动RabbitMQ应用程序。如果进程在没有启动RabbitMQ应用程序的情况下终止，它将失败。
如果未创建指定的pidfile或未在--timeout命令内启动erlang节点，则将失败。默认超时为10秒。
合适的pid文件由 rabbitmq-server（8） 脚本创建。默认情况下，它位于Mnesia目录中。修改 RABBITMQ_PID_FILE环境变量以更改位置。
例如，当RabbitMQ节点启动时，此命令将返回：

```
rabbitmqctl wait /var/run/rabbitmq/pid
```

集群管理

join_cluster *clusternode* [--ram]

CLUSTERNODE
要集群的节点。

--ram
如果提供，节点将作为RAM节点加入群集。

指示节点成为指定节点所在群集的成员。在群集之前，节点已重置，因此使用此命令时请务必小心。要使此命令成功，RabbitMQ应用程序必须已停止，例如使用stop_app。
群集节点可以有两种类型：光盘或RAM。磁盘节点在RAM和磁盘上复制数据，从而在节点发生故障和从全局事件（例如所有节点的电源故障）恢复时提供冗余。RAM节点仅在RAM中复制数据（队列内容除外，如果队列是持久的或者太大而不能容纳在内存中，它可以驻留在磁盘上）并且主要用于可伸缩性。只有在管理资源（例如添加/删除队列，交换或绑定）时，RAM节点才更具性能。群集必须始终至少有一个磁盘节点，并且通常应该有多个磁盘节点。
默认情况下，该节点将是一个磁盘节点。如果要创建RAM节点，请提供--ram标志。
执行join_cluster命令后，每当RabbitMQ应用程序在当前节点上启动时，它将尝试在节点关闭时连接到群集中的节点。
离开群集，reset节点。您还可以使用该forget_cluster_node命令远程删除节点 。
有关详细信息，请参阅“ 群集指南” 。
例如，此命令指示RabbitMQ节点加入“hare @ elena”所属的群集，作为ram节点：

```
rabbitmqctl join_cluster hare@elena --ram
```

cluster_status

显示按节点类型分组的群集中的所有节点以及当前运行的节点。
例如，此命令显示集群中的节点：

```
rabbitmqctl cluster_status
```

change_cluster_node_type *类型*

更改群集节点的类型。
该*类型*必须是下列之一：

- disc
- ram

必须停止节点才能使此操作成功，并且在将节点转换为RAM节点时，节点不能是群集中唯一的磁盘节点。
例如，此命令将RAM节点转换为磁盘节点：

```
rabbitmqctl change_cluster_node_type disc
```

forget_cluster_node [
--offline
]

--offline

允许从脱机节点中删除节点。这仅适用于所有节点都处于脱机状态且最后一个要关闭的节点无法联机的情况，从而阻止整个群集启动。它不应该用于任何其他情况，因为它可能导致不一致。

远程删除群集节点。正在删除的节点必须处于脱机状态，而我们要删除的节点必须处于联机状态，除非使用该--offline标志。
使用该--offline标志时， rabbitmqctl不会尝试正常连接节点; 相反，它将暂时成为节点以进行更改。如果无法正常启动节点，这将非常有用。在这种情况下，节点将成为群集元数据的规范源（例如，存在哪些队列），即使之前不存在。因此，您应该在最新节点上使用此命令，以便尽可能关闭。
例如，此命令将从节点“hare @ mcnulty”中删除节点“rabbit @ stringer”：

```
rabbitmqctl -n hare@mcnulty forget_cluster_node rabbit@stringer
```

rename_cluster_node *oldnode1 newnode1* [
oldnode2 newnode2 ...
]

支持在本地数据库中重命名集群节点。
此子命令导致rabbitmqctl临时成为节点以进行更改。因此必须完全停止本地群集节点; 其他节点可以在线或离线。

此子命令采用偶数个参数，成对表示节点的旧名称和新名称。您必须为此节点以及同时停止和重命名的任何其他节点指定旧名称和新名称。
可以停止所有节点并同时重命名它们（在这种情况下，必须为每个节点提供所有节点的旧名称和新名称），或者一次停止和重命名一个节点（在这种情况下，每个节点只需要被告知它自己的名字如何变化）。
例如，此命令将节点“rabbit @ misshelpful”重命名为节点“rabbit @ cordelia”
```
rabbitmqctl rename_cluster_node rabbit@misshelpful rabbit@cordelia
```
请注意，此命令仅更改本地数据库。可能还需要重命名本地数据库目录，并配置新节点名称。
例如：

1. 停止节点：
   ```
   rabbitmqctl stop rabbit@misshelpful
   ```
2. 重命名本地数据库中的节点：
   ```
   rabbitmqctl rename_cluster_node rabbit@misshelpful rabbit@cordelia
   ```
3. 重命名本地数据库目录（注意，如果已设置RABBITMQ_MNESIA_DIR环境变量，则无需执行此操作）：

   ```
   mv
     / var / lib / rabbitmq / mnesia / rabbit \ @misshelpful \
     的/ var / lib中/ RabbitMQ的/ Mnesia的/兔\ @cordelia
   mv
     / var / lib / rabbitmq / mnesia / rabbit \ @ misshelpful-rename \
     在/ var / lib中/的RabbitMQ / Mnesia的/兔子\ @科黛拉，重命名
   mv
     / var / lib / rabbitmq / mnesia / rabbit \ @ misshelpful-plugins-expand \
     在/ var / lib中/的RabbitMQ / Mnesia的/兔子\ @科黛拉-插件-扩大
   ```

4. 如果您有 */etc/rabbitmq/rabbitmq-env.conf*并在那里配置了节点名称，请更新此配置。
5. 准备好后启动节点

update_cluster_nodes *CLUSTERNODE*

CLUSTERNODE
要查询最新信息的节点。

指示已聚集的节点在唤醒时将*clusternode*联系 到群集。这不同于join_cluster 它不加入任何集群 - 它检查节点是否已经在具有*clusternode*的集群中。
对该命令的需求是由于在节点离线时集群可以改变的事实。考虑节点 *A*和*B* 聚类的情况。 *A*下降， *C*群集与 *B*，然后 *B*离开群集。当 *A*醒来时，它会尝试联系 *B*，但这会失败，因为 *B*不再在群集中。以下命令将解决此问题：
```
update_cluster_nodes -n A C
```
force_boot
确保节点下次启动，即使它不是最后关闭的节点。
通常，当你完全关闭RabbitMQ集群时，你重新启动的第一个节点应该是最后一个节点，因为它可能已经看到其他节点没有发生的事情。但有时这是不可能的：例如，如果整个集群断电，那么所有节点都可能认为它们不是最后关闭的。
在这种情况下，您可以force_boot 在节点关闭时调用。这将告诉节点在下次要求时无条件启动。如果在此节点关闭后群集发生任何更改，它们将丢失。
如果最后一个要关闭的节点永久丢失，那么您应forget_cluster_node --offline优先使用 此命令，因为它将确保在丢失节点上掌握的镜像队列得到提升。
例如，这将强制节点在下次启动时不等待其他节点：
```
rabbitmqctl force_boot
```
sync_queue [
-p *虚拟主机*

] *队列*

队列
要同步的队列的名称。

使用不同步的镜像（跟随者副本）指示镜像队列以同步它们。同步发生时队列将阻塞（所有发布者和使用该队列的消费者将阻止或暂时看不到任何活动）。此命令只能与镜像队列一起使用。要了解更多信息，请参阅 RabbitMQ镜像指南
请注意，具有不同步副本和活动使用者的队列最终将同步（假设消费者取得进展）。此命令主要用于没有活动使用者的队列。

cancel_sync_queue [
-p *虚拟主机*
] *队列*

队列
要取消同步的队列的名称。

指示同步镜像队列以停止自身同步。

purge_queue [
-p *虚拟主机*
] *队列*

队列
要清除的队列的名称。

清除队列（删除其中的所有消息）。

set_cluster_name *名称*
将群集名称设置为*name*。群集名称在连接时通知客户端，并由联合和铲插件用于记录消息的位置。默认情况下，群集名称是从群集中第一个节点的主机名派生的，但可以更改。
例如，这会将群集名称设置为“london”：

```
rabbitmqctl set_cluster_name london
```

用户管理

请注意，所有用户管理命令 rabbitmqctl只能管理内部RabbitMQ数据库中的用户。无法使用这些命令检查或管理来自任何其他身份验证后端（如LDAP）的用户。 rabbitmqctl。

add_user *用户名 密码*

用户名
要创建的用户的名称。
密码
创建的用户用于登录代理的密码。

例如，此命令指示RabbitMQ代理使用（初始）密码“changeit”创建名为“janeway”的（非管理）用户：

```
rabbitmqctl add_user janeway changeit
```

delete_user *用户名*

用户名
要删除的用户的名称。

例如，此命令指示RabbitMQ代理删除名为“janeway”的用户：

```
rabbitmqctl delete_user janeway
```

change_password *用户名 newpassword*

用户名
要更改其密码的用户的名称。
新密码
用户的新密码。

例如，此命令指示RabbitMQ代理将名为“janeway”的用户的密码更改为“newpass”：

```
rabbitmqctl change_password janeway newpass
```

clear_password *用户名*

用户名
要清除其密码的用户的名称。

例如，此命令指示RabbitMQ代理清除名为“janeway”的用户的密码：

```
rabbitmqctl clear_password janeway
```

此用户现在无法使用密码登录（但如果已配置，则可以通过例如SASL EXTERNAL）。

authenticate_user *用户名 密码*

用户名
用户名。
密码
用户的密码。

例如，此命令指示RabbitMQ代理使用密码“verifyit”验证名为“janeway”的用户：

```
rabbitmqctl authenticate_user janeway verifyit
```

set_user_tags *用户名* [*标签 ...*]

用户名
要设置其标记的用户的名称。
标签
零，一个或多个要设置的标签。将删除任何现有标记。

例如，此命令指示RabbitMQ代理确保名为“janeway”的用户是管理员：

```
rabbitmqctl set_user_tags janeway administrator
```

当用户通过AMQP登录时，这不起作用，但可以用于允许用户在用户通过其他方式（例如使用管理插件）登录时管理用户，虚拟主机和权限。
此命令指示RabbitMQ代理从名为“janeway”的用户中删除任何标记：

```
rabbitmqctl set_user_tags janeway
```

list_users
列出用户。每个结果行将包含用户名，后跟为该用户设置的标记列表。
例如，此命令指示RabbitMQ代理列出所有用户：

```
rabbitmqctl list_users
```

访问控制

请注意，`rabbitmqctl`管理RabbitMQ内部用户数据库。来自任何备用授权后端的用户权限将不可见`rabbitmqctl`。

add_vhost *虚拟主机*

虚拟主机
要创建的虚拟主机条目的名称。

创建虚拟主机。
例如，此命令指示RabbitMQ代理创建名为“test”的新虚拟主机：
```
rabbitmqctl add_vhost test
```
delete_vhost *虚拟主机*

虚拟主机
要删除的虚拟主机条目的名称。

删除虚拟主机。
删除虚拟主机会删除其所有交换，队列，绑定，用户权限，参数和策略。
例如，此命令指示RabbitMQ代理删除名为“test”的虚拟主机：
```
rabbitmqctl delete_vhost test
```
list_vhosts [
vhostinfoitem ...
]
列出虚拟主机。
所述*vhostinfoitem*参数用于指示在结果中包含哪些虚拟主机的信息项。结果中的列顺序将与参数的顺序匹配。 *vhostinfoitem*可以从以下列表中获取任何值：

name
具有非ASCII字符的虚拟主机的名称在C中转义。
tracing
是否为此虚拟主机启用了跟踪。

如果未指定*vhostinfoitem*，则显示vhost名称。
例如，此命令指示RabbitMQ代理列出所有虚拟主机：
```
rabbitmqctl list_vhosts name tracing
```
set_permissions [
-p *虚拟主机*
] *用户 CONF 写入 读*

虚拟主机
要授予用户访问权限的虚拟主机的名称，默认为“/”。
用户
用于授予对指定虚拟主机的访问权限的用户的名称。
CONF
正则表达式匹配为其授予用户配置权限的资源名称。
写
正则表达式，匹配为用户授予写入权限的资源名称。
读
正则表达式，匹配为其授予用户读取权限的资源名称。

设置用户权限。
例如，此命令指示RabbitMQ代理向名为“janeway”的用户授予对名为“my-vhost”的虚拟主机的访问权限，并对名称以“janeway-”开头的所有资源具有配置权限，并具有写入和读取权限。所有资源：
```
rabbitmqctl set_permissions -p my-vhost janeway "^janeway-.*" ".*" ".*"
```
clear_permissions [
-p *虚拟主机*
] *用户名*

虚拟主机
拒绝用户访问的虚拟主机的名称，默认为“/”。
用户名
拒绝访问指定虚拟主机的用户名。

设置用户权限。
例如，此命令指示RabbitMQ代理拒绝名为“janeway”的用户访问名为“my-vhost”的虚拟主机：
```
rabbitmqctl clear_permissions -p my-vhost janeway
```
list_permissions [
-p *虚拟主机*
]

虚拟主机
要列出已授予其访问权限的用户及其权限的虚拟主机的名称。默认为“/”。

列出虚拟主机中的权限。
例如，此命令指示RabbitMQ代理列出已被授予对名为“my-vhost”的虚拟主机的访问权限的所有用户，以及他们对该虚拟主机中资源的操作所具有的权限。请注意，空字符串表示没有授予权限：
```
rabbitmqctl list_permissions -p my-vhost
```
list_user_permissions *用户名*

用户名
要列出权限的用户的名称。

列出用户权限。
例如，此命令指示RabbitMQ代理列出已授予用户名为“janeway”的所有虚拟主机，以及用户对这些虚拟主机中的资源的操作所具有的权限：
```
rabbitmqctl list_user_permissions janeway
```
set_topic_permissions [
-p *虚拟主机*
] *用户 交换 写入 读取*

虚拟主机
要授予用户访问权限的虚拟主机的名称，默认为“/”。
用户
权限在目标虚拟主机中应用的用户的名称。
交换
将应用授权检查的主题交换的名称。
写
与已发布消息的路由键匹配的正则表达式。
读
与消耗的消息的路由密钥匹配的正则表达式。

设置用户主题权限。
例如，此命令指示RabbitMQ代理让名为“janeway”的用户发布并使用以“janeway-”开头的路由密钥通过“my-vhost”虚拟主机的“amp.topic”交换消息：
```
rabbitmqctl set_topic_permissions -p my-vhost janeway amq.topic “^janeway-.*” “^janeway-.*”
```
主题权限支持以下变量的变量扩展：username，vhost和client_id。请注意，仅在使用MQTT时才会扩展client_id。通过使用“^ {username} - 。*”可以使前面的示例更通用：
```
rabbitmqctl set_topic_permissions -p my-vhost janeway amq.topic “^{username}-.*” “^{username}-.*”
```

clear_topic_permissions [
-p *虚拟主机*
] *用户名* [
交换
]

虚拟主机
要清除主题权限的虚拟主机的名称，默认为“/”。
用户名
用于清除指定虚拟主机的主题权限的用户的名称。
交换
主题交换的名称，用于清除主题权限，默认为给定用户具有主题权限的所有主题交换。

清除用户主题权限。
例如，此命令指示RabbitMQ代理删除名为“janeway”的用户的主题权限，用于名为“my-vhost”的虚拟主机中的主题交换“amq.topic”：

```
rabbitmqctl clear_topic_permissions -p my-vhost janeway amq.topic
```

list_topic_permissions [
-p *虚拟主机*
]

虚拟主机
要列出用户主题权限的虚拟主机的名称。默认为“/”。

列出虚拟主机中的主题权限。
例如，此命令指示RabbitMQ代理列出已在虚拟主机中被授予主题权限的所有用户，名为“my-vhost：”

```
rabbitmqctl list_topic_permissions -p my-vhost
```

list_user_topic_permissions *用户名*

用户名
要列出主题权限的用户的名称。

列出用户主题权限。
例如，此命令指示RabbitMQ代理列出已授予用户名为“janeway”的所有虚拟主机，以及用户在这些虚拟主机中拥有的主题权限：

```
rabbitmqctl list_topic_user_permissions janeway
```

参数管理

RabbitMQ的某些功能（例如联合插件）由动态的群集范围*参数控制*。有两种参数：作用于虚拟主机的参数和全局参数。每个vhost-scoped参数由组件名称，名称和值组成。组件名称和名称是字符串，值是有效的JSON文档。全局参数由名称和值组成。名称是一个字符串，值是任意Erlang数据结构。可以设置，清除和列出参数。通常，您应该参考相关功能的文档来了解如何设置参数。

set_parameter [
-p *虚拟主机*
] *component_name 名称 值*
设置参数。

COMPONENT_NAME
要为其设置参数的组件的名称。

名称
要设置的参数的名称。
值
参数的值，作为JSON术语。在大多数shell中，您很可能需要引用它。

例如，此命令将默认虚拟主机中“federation-upstream”组件的参数“node01”设置为以下JSON“guest”：
```
rabbitmqctl set_parameter federation-upstream node01 '{"uri":"amqp://user:password@server/%2F","ack-mode":"on-publish"}'
```

clear_parameter [-p *虚拟主机*] *component_name 键*
清除参数。

COMPONENT_NAME
要为其清除参数的组件的名称。
名称
要清除的参数的名称。

例如，此命令清除默认虚拟主机中“federation-upstream”组件的参数“node01”：
```
rabbitmqctl clear_parameter federation-upstream node01
```

list_parameters [-p *虚拟主机*]
列出虚拟主机的所有参数。
例如，此命令列出默认虚拟主机中的所有参数：
```
rabbitmqctl list_parameters
```

set_global_parameter *名称 值*
设置全局运行时参数。这类似于 set_parameter但键值对不依赖于虚拟主机。

名称
要设置的全局运行时参数的名称。
值
全局运行时参数的值，作为JSON术语。在大多数shell中，您很可能需要引用它。

例如，此命令将全局运行时参数“mqtt_default_vhosts”设置为JSON术语{“O = client，CN = guest”：“/”}：
```
rabbitmqctl set_global_parameter mqtt_default_vhosts '{"O=client,CN=guest":"/"}'
```

clear_global_parameter *名称*
清除全局运行时参数。这类似于 clear_parameter但键值对不依赖于虚拟主机。

名称
要清除的全局运行时参数的名称。

例如，此命令清除全局运行时参数“mqtt_default_vhosts”：
```
rabbitmqctl clear_global_parameter mqtt_default_vhosts
```

list_global_parameters
列出所有全局运行时参数。这类似于 list_parameters但是全局运行时参数不依赖于任何虚拟主机。
例如，此命令列出所有全局参数：
```
rabbitmqctl list_global_parameters
```

政策管理

策略用于在群集范围内控制和修改队列和交换的行为。策略适用于给定的vhost，包括名称，模式，定义和可选优先级。可以设置，清除和列出策略。

set_policy [
-p *虚拟主机*
] [[
--priority *优先*
] [[
--apply-to *适用于*
] *名称 模式 定义*

设置策略。

名称

政策的名称。

图案

正则表达式，在匹配给定资源时会导致应用策略。

定义

策略的定义，作为JSON术语。在大多数shell中，您很可能需要引用它。

优先

策略的优先级为整数。数字越大表示优先级越高。默认值为0。

适用于

此策略应适用于哪些类型的对象。可能的值是：

- queues
- exchanges
- all

默认是 all ..

例如，此命令在默认虚拟主机中设置策略“federate-me”，以便联合内置交换：

```
rabbitmqctl set_policy federate-me ^amq. '{"federation-upstream-set":"all"}'
```

clear_policy [
-p *虚拟主机*
] *名字*

清除政策。

名称

要清除的策略的名称。

例如，此命令清除默认虚拟主机中的“federate-me”策略：

```
rabbitmqctl clear_policy federate-me
```

list_policies [
-p *虚拟主机*
]

列出虚拟主机的所有策略。

例如，此命令列出默认虚拟主机中的所有策略：

```
rabbitmqctl list_policies
```

set_operator_policy [
-p *虚拟主机*
] [[
--priority *优先*
] [[

--apply-to *适用于*
] *名称 模式 定义*
设置一个覆盖用户策略中参数子集的运算符策略。参数与那些相同 set_policy。
支持的参数是：

- 到期
- 消息的TTL
- 最长长度
- 最大长度的字节

clear_operator_policy [
-p *虚拟主机*
] *名字*
清除运营商政策。参数与那些相同 clear_policy。
list_operator_policies [
-p *虚拟主机*
]
列出虚拟主机的操作员策略覆盖。参数与那些相同 list_policies。

虚拟主机限制

可以对虚拟主机强制执行某些限制。

set_vhost_limits [
-p *虚拟主机*
] *定义*
设置虚拟主机限制。

定义
限制的定义，作为JSON术语。在大多数shell中，您很可能需要引用它。
公认的限制是：

- 最大的连接
- 最大队列

使用负值指定“无限制”。

例如，此命令将vhost“qa_env”中的最大并发连接数限制为64：
```
rabbitmqctl set_vhost_limits -p qa_env '{"max-connections": 64}'
```
此命令将vhost“qa_env”中的最大队列数限制为256：
```
rabbitmqctl set_vhost_limits -p qa_env '{"max-queues": 256}'
```
此命令清除vhost“qa_env”中的最大连接数限制：
```
rabbitmqctl set_vhost_limits -p qa_env '{"max-connections": -1}'
```
此命令禁用vhost“qa_env”中的客户端连接：
```
rabbitmqctl set_vhost_limits -p qa_env '{"max-connections": 0}'
```
clear_vhost_limits [
-p *虚拟主机*
]
清除虚拟主机限制。
例如，此命令清除vhost“qa_env”中的vhost限制：
```
rabbitmqctl clear_vhost_limits -p qa_env
```
list_vhost_limits [

-p *虚拟主机*
] [[
--global
]

显示配置的虚拟主机限制。

--global

显示所有虚拟主机的限制。抑制 -p参数。

拓扑反思

拓扑内省命令列出具有制表符分隔列的拓扑实体（例如队列）。一些命令（ list_queues, list_exchanges， list_bindings和 list_consumers）接受一个可选的 *虚拟主机*参数。
和list_queues， list_exchanges和 list_bindings命令接受可显示结果的可选虚拟主机参数。默认值为“/”。

list_queues [
-p *虚拟主机*
] [[
--offline| --online| --local
] [[
queueinfoitem ...
]

返回队列详细信息 如果-p标志不存在，则返回“/”虚拟主机的队列详细信息。该-p标志可用于覆盖此默认值。
显示的队列可以使用以下互斥选项之一按其状态或位置进行过滤：

--offline

仅列出当前不可用的持久队列（更具体地说，它们的主节点不可用）。

--online

列出当前可用的队列（其主节点是）。

--local

仅列出主进程位于当前节点上的队列。

所述*queueinfoitem*参数用于指示哪个队列信息项中的结果，包括。结果中的列顺序将与参数的顺序匹配。 *queueinfoitem*可以从以下列表中获取任何值：

name

带有非ASCII字符的队列名称在C中转义。

durable

队列是否在服务器重新启动后仍然存在。

auto_delete

是否在不再使用时自动删除队列。

arguments

队列参数。

policy

队列的有效策略名称。

pid

队列的Erlang进程标识符。

owner_pid

连接的Erlang进程的Id，它是队列的独占所有者。如果队列是非独占的，则清空。

exclusive

如果队列是独占的（即具有owner_pid），则为True，否则为false。

exclusive_consumer_pid

Erlang进程的Id，表示订阅此队列的独占消费者的频道。如果没有独家消费者，则清空。

exclusive_consumer_tag
: 订阅此队列的独占消费者的消费者标签。如果没有独家消费者，则清空。

messages_ready
: 准备传递给客户端的消息数。

messages_unacknowledged
: 传递给客户端但尚未确认的消息数。

messages
: 准备好和未确认消息的总和（队列深度）。

messages_ready_ram
: 来自messages_ready的邮件数量，它们驻留在ram中。

messages_unacknowledged_ram
: 来自messages_unacknowledged的邮件数量，它们驻留在ram中。

messages_ram
: ram中驻留的消息总数。

messages_persistent
: 队列中的持久性消息总数（对于瞬态队列，总是为0）。

message_bytes
: 队列中所有邮件正文的大小总和。这不包括消息属性（包括标头）或任何开销。

message_bytes_ready
: 喜欢message_bytes但只计算准备交付给客户的那些消息。

message_bytes_unacknowledged
: 像message_bytes，但只计算交付给客户的这些消息，但尚未确认。

message_bytes_ram
: 像message_bytes，但只计算那些目前在RAM中的消息。

message_bytes_persistent
: 喜欢message_bytes但只计算那些持久的消息。

head_message_timestamp
: 队列中第一条消息的timestamp属性（如果存在）。消息的时间戳仅在处于分页状态时才会出现。

disk_reads
: 自该队列启动以来该队列从磁盘读取消息的总次数。

disk_writes
: 自启动以来此队列将消息写入磁盘的总次数。

consumers
: 消费者数量。

consumer_utilisation
: 队列能够立即向消费者传递消息的时间分数（介于0.0和1.0之间）。如果消费者受到网络拥塞或预取计数的限制，则可以小于1.0。

memory
: 运行时为队列分配的内存字节数，包括堆栈，堆和内部结构。

slave_pids
: 如果队列是镜像的，则列出镜像的ID（跟随者副本）。要了解更多信息，请参阅 RabbitMQ镜像指南

synchronised_slave_pids
: 如果镜像队列，则会提供与主（领导者）同步的镜像（跟随者副本）的ID。要了解更多信息，请参阅 RabbitMQ镜像指南

state
: 队列的状态。通常“正在运行”，但如果队列正在同步，则可能是“{syncing, *message_count*}”。
位于当前关闭的群集节点上的队列将显示状态为“关闭”（并且大多数其他*queueinfoitem*将不可用）。

如果未指定*queueinfoitem*，则显示队列名称和深度。
例如，此命令显示名为“my-vhost”的虚拟主机的每个队列的使用者的深度和数量。

```
rabbitmqctl list_queues -p my-vhost messages consumers
```

list_exchanges [
-p *虚拟主机*
] [[
exchangeinfoitem ...
]

返回交换详情。如果没有-p 标志，则返回“/”虚拟主机的交换详细信息。该-p标志可用于覆盖此默认值。
该*exchangeinfoitem*参数用于指示交换信息项的结果中包含。结果中的列顺序将与参数的顺序匹配。 *exchangeinfoitem*可以从以下列表中获取任何值：

name
: 带有非ASCII字符的交换的名称在C中转义。

type
: 交换类型，例如：

- 直接
- 话题
- 头
- 扇出

durable
: 交换机是否在服务器重启后仍然存在。

auto_delete
: 交换机是否会在不再使用时自动删除。

internal
: 交换是否是内部的，即不能由客户直接发布。

arguments
: 交换论点。

policy
: 申请交易所的政策名称。

如果没有*exchangeinfoitem*指定然后显示交换名称和类型。
例如，此命令显示名为“my-vhost”的虚拟主机的每次交换的名称和类型：

```
rabbitmqctl list_exchanges -p my-vhost name type
```

list_bindings [
-p *虚拟主机*
] [[
bindinginfoitem ...
]

返回绑定详细信息 默认情况下，返回“/”虚拟主机的绑定。该-p标志可用于覆盖此默认值。
所述*bindinginfoitem*参数用于指示哪个绑定信息项中的结果包括。结果中的列顺序将与参数的顺序匹配。 *bindinginfoitem*可以从以下列表中获取任何值：

source_name
: 附加绑定的消息源的名称。使用非ASCII字符转义，如在C.

source_kind
: 附加绑定的消息源的类型。目前总是交流。使用非ASCII字符转义，如在C.

destination_name
: 附加绑定的消息的目标名称。使用非ASCII字符转义，如在C.

destination_kind
: 附加绑定的消息的目标类型。使用非ASCII字符转义，如在C.

routing_key
: 绑定的路由密钥，非ASCII字符在C中转义。

arguments

绑定的参数。

如果未指定*bindinginfoitem*，则显示以上所有项目。
例如，此命令显示名为“my-vhost”的虚拟主机中绑定的交换名称和队列名称
```
rabbitmqctl list_bindings -p my-vhost exchange_name queue_name
```

list connections [
connectioninfoitem ...
]

返回TCP / IP连接统计信息。
所述*connectioninfoitem*参数用于指示在结果中包括的连接信息项。结果中的列顺序将与参数的顺序匹配。 *connectioninfoitem*可以从以下列表中获取任何值：

pid
与连接关联的Erlang进程的ID。
name
连接的可读名称。
port
服务器端口。
host
通过反向DNS获取的服务器主机名，或者如果反向DNS失败或被禁用，则为其IP地址。
peer port
同行港口。
peer host
通过反向DNS获取的对等主机名，或者如果反向DNS失败或未启用，则为其IP地址。
ssl
布尔值，指示连接是否使用SSL保护。
ssl protocol
SSL协议（例如“tlsv1”）。
ssl key exchange
SSL密钥交换算法（例如“rsa”）。
ssl cipher
SSL密码算法（例如“aes_256_cbc”）。
ssl hash
SSL散列函数（例如“sha”）。
peer cert subject
对等方SSL证书的主题，RFC4514格式。
peer cert issuer
对等方SSL证书的颁发者，采用RFC4514格式。
peer cert validity
对等方SSL证书有效的时间段。
state
连接状态; 之一：

- 开始
- 调音
- 开盘
- 赛跑
- 流
- 闭塞
- 受阻
- 关闭
- 关闭

channels
使用连接的通道数。

protocol
正在使用的AMQP协议版本; 目前之一：

- {0,9,1}
- {0,8,0}

请注意，如果客户端请求AMQP 0-9连接，我们将其视为AMQP 0-9-1。

auth_mechanism
使用SASL身份验证机制，例如“PLAIN”。

user
与连接关联的用户名。

vhost
带有非ASCII字符的虚拟主机名在C中转义。

timeout
连接超时/协商的心跳间隔，以秒为单位。

frame_max
最大帧大小（字节）。

channel_max
此连接上的最大通道数。

client_properties
在连接建立期间由客户端传输的信息属性。

recv_oct
收到八位字节。

recv_cnt
收到的数据包。

send_oct
八位字节发送。

send_cnt
发送的数据包。

send_pend
发送队列大小。

connected_at
建立此连接的日期和时间，作为时间戳。

如果未指定*connectioninfoitem*，则显示用户，对等主机，对等端口，自流控制以来的时间和内存块状态。
例如，此命令显示每个连接的发送队列大小和服务器端口：

```
rabbitmqctl list_connections send_pend port
```

list_channels [
channelinfoitem ...
]

返回有关所有当前通道的信息，即执行大多数AMQP命令的逻辑容器。这包括作为普通AMQP连接一部分的通道，以及由各种插件和其他扩展创建的通道。
所述*channelinfoitem*参数用于指示在结果中包括频道信息项。结果中的列顺序将与参数的顺序匹配。 *channelinfoitem*可以从以下列表中获取任何值：

pid
与连接关联的Erlang进程的ID。

connection
与通道所属的连接关联的Erlang进程的ID。

name
频道的可读名称。

number
通道的编号，在连接中唯一标识它。

user
与频道关联的用户名。

vhost
通道运行的虚拟主机。
transactional
如果通道处于事务模式，则为True，否则为false。
confirm
如果通道处于确认模式，则为True，否则为false。
consumer_count
通过通道检索消息的逻辑AMQP使用者数。
messages_unacknowledged
通过此频道发送但尚未确认的消息数。
messages_uncommitted
尚未提交的事务中收到的消息数。
acks_uncommitted
尚未提交的交易中收到的确认数量。
messages_unconfirmed
已确认的已发布消息数。在未处于确认模式的通道上，此值保持为0。
prefetch_count
新消费者的QoS预取限制，如果无限制则为0。
global_prefetch_count
整个通道的QoS预取限制，如果不受限制则为0。

如果未指定*channelinfoitem*，则假定为pid，user，consumer_count和messages_unacknowledged。
例如，此命令显示每个通道的连接过程和未确认消息的计数：
```
rabbitmqctl list_channels connection messages_unacknowledged
```

list_consumers [
-p *虚拟主机*
]
列出消费者，即对队列消息流的订阅。打印的每一行显示，由制表符分隔，订阅的队列的名称，创建和管理订阅的通道进程的id，唯一标识通道内订阅的使用者标记，指示是否是布尔值的布尔值对于传递给此使用者的消息，应该是确认，一个表示预取限制的整数（0表示“无”），以及此使用者的任何参数。

status
显示代理状态信息，例如当前Erlang节点上运行的应用程序，RabbitMQ和Erlang版本，操作系统名称，内存和文件描述符统计信息。（请参阅 cluster_status命令以找出哪些节点已群集并正在运行。）
例如，此命令显示有关RabbitMQ代理的信息：
```
rabbitmqctl status
```

node_health_check
对目标节点执行多次运行状况检查。
验证Rabbit应用程序正在运行且未设置警报，然后检查节点上的每个队列和通道是否都可以发出基本统计信息。
例：
```
rabbitmqctl node_health_check -n rabbit@hostname
```

environment
显示每个正在运行的应用程序的应用程序环境中每个变量的名称和值。

report
生成服务器状态报告，其中包含所有服务器状态信息的串联以用于支持目的。在提供支持请求时，应将输出重定向到文件。
例如，此命令创建一个服务器报告，该报告可以附加到支持请求电子邮件：
```
rabbitmqctl report > server_report.txt
```

eval *EXPR*
评估任意Erlang表达式。
例如，此命令返回rabbitmqctl已连接的节点的名称：
```
rabbitmqctl eval “node().”
```

杂

close_connection *connectionpid 解释*

connectionpid
与关闭连接关联的Erlang进程的ID。
说明
说明字符串。

指示代理关闭与Erlang进程ID connectionpid关联的*连接*（另请参阅 list_connections命令），将*解释*字符串作为AMQP连接关闭协议的一部分传递 给连接的客户端。
例如，此命令指示RabbitMQ代理关闭与Erlang进程ID“<rabbit@tanto.4262.0>”关联的连接，并将解释“go away”传递给连接的客户端：

```
rabbitmqctl close_connection “<rabbit@tanto.4262.0>” “go away”
```

close_all_connections [
-p *虚拟主机*
] [[
--global
] [[
--per-connection-delay *延迟*
] [[
--limit *限制*
] *解释*

-p *虚拟主机*
应为其关闭连接的虚拟主机的名称。--global指定时忽略。
--global
如果所有vhosts的连接都应该关闭。覆盖 -p
--per-connection-delay *延迟*
每次连接关闭后等待的时间（以毫秒为单位）。
--limit *限制*
要关闭的连接数。仅适用于每个虚拟主机。--global指定时忽略 。
说明
说明字符串。

指示代理关闭指定vhost或整个RabbitMQ节点的所有连接。
例如，此命令指示RabbitMQ代理关闭“qa_env”vhost上的10个连接，并传递说明“请关闭”：

```
rabbitmqctl close_all_connections -p qa_env --limit 10 ’Please close’
```

此命令指示代理关闭与节点的所有连接：

```
rabbitmqctl close_all_connections --global
```

trace_on [
-p *虚拟主机*
]

虚拟主机
要开始跟踪的虚拟主机的名称。

开始跟踪。请注意，跟踪状态不是持久的; 如果重新启动服务器，它将恢复为关闭状态。

trace_off [
-p *虚拟主机*
]

虚拟主机
要停止跟踪的虚拟主机的名称。

停止跟踪。

set_vm_memory_high_watermark *分数*

分数
触发流量控制的新内存阈值分数，作为大于或等于0的浮点数。

set_vm_memory_high_watermark absolute *memory_limit的*

memory_limit的
触发流量控制的新内存限制，以字节为单位表示为大于或等于0的整数或带有内存单位的字符串（例如512M或1G）。可用单位是：

k, kiB
kibibytes（2 ^ 10字节）
M, MiB
mebibytes（2 ^ 20字节）
G, GiB
gibibytes（2 ^ 30字节）
kB
千字节（10 ^ 3字节）
MB
兆字节（10 ^ 6字节）
GB
千兆字节（10 ^ 9字节）

set_disk_free_limit *disk_limit*

disk_limit
下限为整数（以字节为单位）或带有内存单位的字符串（请参阅vm_memory_high_watermark），例如512M或1G。一旦可用磁盘空间达到限制，将设置磁盘警报。

set_disk_free_limit mem_relative *分数*

分数
相对于可用RAM的总量限制为非负浮点数。低于1.0的值可能很危险，应谨慎使用。

encode *值 密码* [
--cipher *暗号*
] [[
--hash *哈希*
] [[
--iterations *迭代*
]

价值 密码
加密和密码短语的价值。
例如：
rabbitmqctl encode '<<"guest">>' mypassphrase
--cipher *密码* --hash *哈希* --iterations *迭代*

用于指定加密设置的选项。它们可以单独使用。
例如：
```
rabbitmqctl encode --cipher blowfish_cfb64 --hash sha256 --iterations 10000 '<<"guest">>'
mypassphrase
```

decode *值 密码* [
--cipher *暗号*
] [[
--hash *哈希*
] [[
--iterations *迭代*
]

价值 密码
要解密的值（由encode命令生成）和密码。
例如：
```
rabbitmqctl decode '{encrypted, <<"...">>}' mypassphrase
```
--cipher *密码* --hash *哈希* --iterations *迭代*
用于指定解密设置的选项。它们可以单独使用。
例如：
```
rabbitmqctl decode --cipher blowfish_cfb64 --hash sha256 --iterations 10000 '{encrypted,
<<"...">>}' mypassphrase
```

list_hashes
列出编码命令支持的哈希函数。
例如，此命令指示RabbitMQ代理列出编码命令支持的所有散列函数：
```
rabbitmqctl list_hashes
```
list_ciphers
列出编码命令支持的密码套件。
例如，此命令指示RabbitMQ代理列出编码命令支持的所有密码套件：
```
rabbitmqctl list_ciphers
```

插件命令

RabbitMQ插件可以扩展rabbitmqctl工具以在启用时添加新命令。当前可用的命令可以在`rabbitmqctl help`输出中找到 。以下命令由RabbitMQ插件添加，默认分发中提供：

铲插件

shovel_status
打印配置的铲子列表
delete_shovel [
-p *虚拟主机*
] *名字*
指示RabbitMQ节点按*名称*删除配置的铲 。

联合插件

federation_status [
--only-down
]
打印联合链接列表。

--only-down
仅列出未运行的联合链接。

restart federation link *link_id*
指示RabbitMQ节点使用指定的*link_id*重新启动联合链接 。

AMQP-1.0插件

list amqp10 connections [
amqp10_connectioninfoitem ...
]
与list_connections 命令类似，但返回对AMQP-1.0连接有意义的字段。
*amqp10_connectioninfoitem*参数用于指示要包含在结果中的连接信息项。结果中的列顺序将与参数的顺序匹配。*amqp10_connectioninfoitem*可以从以下列表中获取任何值：

pid
与连接关联的Erlang进程的ID。
auth_mechanism
使用SASL身份验证机制，例如“PLAIN”。
host
通过反向DNS获取的服务器主机名，或者如果反向DNS失败或被禁用，则为其IP地址。
frame_max
最大帧大小（字节）。
timeout
连接超时/协商的心跳间隔，以秒为单位。
user
与连接关联的用户名。
state
连接状态; 之一：

- 开始
- waiting_amqp0100
- 确保
- 赛跑
- 闭塞
- 受阻
- 关闭
- 关闭

recv_oct
收到八位字节。
recv_cnt
收到的数据包。
send_oct
八位字节发送。
send_cnt
发送的数据包。
ssl
布尔值，指示连接是否使用SSL保护。
ssl_protocol
SSL协议（例如“tlsv1”）。
ssl_key_exchange
SSL密钥交换算法（例如“rsa”）。
ssl_cipher
SSL密码算法（例如“aes_256_cbc”）。

ssl_hash
: SSL散列函数（例如“sha”）。

peer_cert_subject
: 对等方SSL证书的主题，RFC4514格式。

peer_cert_issuer
: 对等方SSL证书的颁发者，采用RFC4514格式。

peer_cert_validity
: 对等方SSL证书有效的时间段。

node
: 建立连接的RabbitMQ节点的节点名称。

MQTT插件

list_mqtt_connections [
mqtt_connectioninfoitem
]

与list_connections 命令类似，但返回对MQTT连接有意义的字段。 *mqtt_connectioninfoitem* 参数用于指示要包含在结果中的连接信息项。结果中的列顺序将与参数的顺序匹配。
*mqtt_connectioninfoitem*可以从以下列表中获取任何值：

host
: 通过反向DNS获取的服务器主机名，或者如果反向DNS失败或被禁用，则为其IP地址。

port
: 服务器端口。

peer_host
: 通过反向DNS获取的对等主机名，或者如果反向DNS失败或未启用，则为其IP地址。

peer_port
: 同行港口。

protocol
: MQTT协议版本，可以是以下内容：

  - {'MQTT'，N / A}
  - {'MQTT'，3.1.0}
  - {'MQTT'，3.1.1}

channels
: 使用连接的通道数。

channel_max
: 此连接上的最大通道数。

frame_max
: 最大帧大小（字节）。

client_properties
: 在连接建立期间由客户端传输的信息属性。

ssl
: 布尔值，指示连接是否使用SSL保护。

ssl_protocol
: SSL协议（例如“tlsv1”）。

ssl_key_exchange
: SSL密钥交换算法（例如“rsa”）。

ssl_cipher
: SSL密码算法（例如“aes_256_cbc”）。

ssl_hash
: SSL散列函数（例如“sha”）。

conn_name
: 连接的可读名称。

connection_state

连接状态; 之一：

- 开始
- 赛跑
- 受阻

connection
: 与内部amqp直接连接关联的Erlang进程的ID。

consumer_tags
: QOS0和QOS1的消费者标签元组。

message_id
: 控制消息中发送的最后一个数据包ID。

client_id
: 连接的MQTT客户端标识符。

clean_sess
: MQTT清理会话标志。

will_msg
: MQTT将在CONNECT帧中发送消息。

exchange
: Exchange以路由在rabbitmq_mqtt应用程序环境中配置的MQTT消息。

ssl_login_name
: SSL对等证书身份验证名称

retainer_pid
: 与保留连接存储关联的Erlang进程的ID。

user
: 与连接关联的用户名。

vhost
: 带有非ASCII字符的虚拟主机名在C中转义。

STOMP插件

list_stomp_connections [
stomp_connectioninfoitem
]
: 与`list_connections` 命令类似，但返回对STOMP连接有意义的字段。
*stomp_connectioninfoitem*参数用于指示要包含在结果中的连接信息项。结果中的列顺序将与参数的顺序匹配。*stomp_connectioninfoitem*可以从以下列表中获取任何值：

conn_name
: 连接的可读名称。

connection
: 与内部amqp直接连接关联的Erlang进程的ID。

connection_state
: 连接状态; 之一：

- 赛跑
- 闭塞
- 受阻

session_id
: STOMP协议会话标识符

channel
: 与连接关联的AMQP通道

version
: 协商用于连接的STOMP协议版本。

implicit_connect

指示是否使用隐式连接建立连接（没有CONNECT帧）

auth_login
连接的有效用户名。

auth_mechanism
STOMP授权机制。可以是以下之一：

- 配置
- SSL
- stomp_headers

port
服务器端口。

host
通过反向DNS获取的服务器主机名，或者如果反向DNS失败或未启用，则为其IP地址。

peer_port
同行港口。

peer_host
通过反向DNS获取的对等主机名，或者如果反向DNS失败或未启用，则为其IP地址。

protocol
STOMP协议版本，可以是以下内容：

- {'STOMP'，0}
- {'STOMP'，1}
- {'STOMP'，2}

channels
使用连接的通道数。

channel_max
此连接上的最大通道数。

frame_max
最大帧大小（字节）。

client_properties
客户端在连接期间传输的信息属性

ssl
布尔值，指示连接是否使用SSL保护。

ssl_protocol
SSL协议（例如“tlsv1”）。

ssl_key_exchange
SSL密钥交换算法（例如“rsa”）。

ssl_cipher
SSL密码算法（例如“aes_256_cbc”）。

ssl_hash
SSL散列函数（例如“sha”）。

管理代理插件

reset_stats_db [
--all
]
重置RabbitMQ节点的管理统计数据库。

--all
重置群集中所有节点的统计数据库。

也可以看看

rabbitmq-diagnostics（8）， rabbitmq-plugins（8）， rabbitmq-server（8）， rabbitmq-service（8）， rabbitmq-env.conf（5）， rabbitmq-echopid（8）

作者

RabbitMQ团队 < info@rabbitmq.com >