

分布式消息服务 RabbitMQ 版

用户指南

文档版本 01
发布日期 2024-10-25



版权所有 © 华为云计算技术有限公司 2024。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

目录

1 RabbitMQ 业务使用流程	1
2 权限管理	3
2.1 创建用户并授权使用 DMS for RabbitMQ	3
3 购买 RabbitMQ 实例	7
4 配置 Vhost	14
4.1 创建 RabbitMQ Vhost	14
4.2 创建 RabbitMQ Exchange	17
4.3 绑定 RabbitMQ Exchange	19
4.4 创建 RabbitMQ Queue	21
4.5 绑定 RabbitMQ Queue	23
4.6 管理 RabbitMQ Vhost	24
4.6.1 查看 RabbitMQ Vhost 信息	24
4.6.2 删除 RabbitMQ Vhost	25
4.7 管理 RabbitMQ Exchange	27
4.7.1 解绑 RabbitMQ Exchange	27
4.7.2 删除 RabbitMQ Exchange	29
4.8 管理 RabbitMQ Queue	29
4.8.1 查看 RabbitMQ Queue 信息	29
4.8.2 清空 RabbitMQ Queue 消息	30
4.8.3 解绑 RabbitMQ Queue	32
4.8.4 配置 RabbitMQ 镜像队列	33
4.8.5 配置 RabbitMQ 惰性队列	35
4.8.6 配置 RabbitMQ 仲裁队列	36
4.8.7 配置 RabbitMQ 单一活跃消费者	40
4.8.8 删除 RabbitMQ Queue	42
5 连接实例	46
5.1 配置 RabbitMQ 网络连接	46
5.1.1 连接 RabbitMQ 网络要求	46
5.1.2 配置 RabbitMQ 实例的公网访问	47
5.2 配置 RabbitMQ 访问控制	49
5.2.1 开启 RabbitMQ ACL 访问	49
5.2.2 配置 RabbitMQ ACL 用户	50

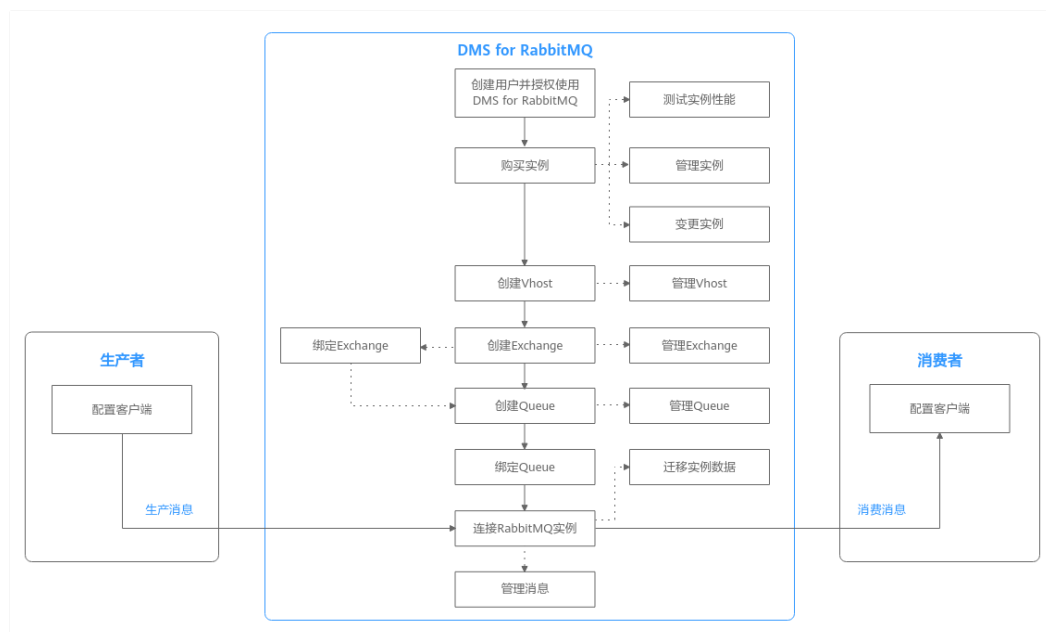
5.3 配置 RabbitMQ 客户端心跳超时时间.....	53
5.4 使用客户端连接 RabbitMQ（关闭 SSL）.....	54
5.5 使用客户端连接 RabbitMQ（开启 SSL）.....	56
6 管理消息.....	60
6.1 查看 RabbitMQ 消息.....	60
6.2 配置 RabbitMQ 死信消息.....	63
6.3 配置 RabbitMQ 消息确认机制.....	64
6.4 配置 RabbitMQ 消息预取值.....	65
7 配置高级特性.....	67
7.1 配置 RabbitMQ 持久化.....	67
7.2 配置 RabbitMQ TTL.....	71
8 管理实例.....	73
8.1 查看和修改 RabbitMQ 实例基本信息.....	73
8.2 查看 RabbitMQ 客户端连接地址.....	76
8.3 配置 RabbitMQ 实例标签.....	77
8.4 配置 RabbitMQ 实例回收站策略.....	78
8.5 重置 RabbitMQ 实例密码.....	82
8.6 开启 RabbitMQ 实例插件.....	82
8.7 导出 RabbitMQ 实例列表.....	84
8.8 删除 RabbitMQ 实例.....	84
8.9 登录 RabbitMQ WebUI.....	85
9 变更 RabbitMQ 实例规格.....	88
10 迁移 RabbitMQ 业务.....	91
11 测试实例性能.....	95
11.1 测试 RabbitMQ 生产速率和消费速率.....	95
12 申请扩大 RabbitMQ 配额.....	105
13 查看监控指标与配置告警.....	107
13.1 查看 RabbitMQ 监控数据.....	107
13.2 RabbitMQ 支持的监控指标.....	108
13.3 配置 RabbitMQ 监控告警.....	121
14 查看 RabbitMQ 审计日志.....	125

1 RabbitMQ 业务使用流程

分布式消息服务RabbitMQ版完全兼容开源RabbitMQ，为您提供即开即用、消息特性丰富、灵活路由、高可用、监控和告警等特性，广泛应用于秒杀、流控、系统解耦等场景。

使用RabbitMQ实例生产消费消息的流程如下图所示。

图 1-1 RabbitMQ 业务使用流程



- 1. 创建用户并授权使用DMS for RabbitMQ**
创建IAM用户，并授予DMS for RabbitMQ的权限，以达到用户之间权限的隔离。
- 2. 购买RabbitMQ实例**
RabbitMQ实例采用物理隔离的方式部署，租户独占RabbitMQ实例。
- 3. 创建Vhost**
生产者和消费者连接RabbitMQ实例时，需要指定一个Vhost。
- 4. 创建Exchange**

生产者向RabbitMQ实例发送消息时，不会直接将消息发送到Queue，而是先将消息发送到Exchange中，Exchange根据路由键将消息发送到对应的Queue。

5. **创建Queue**

Queue用于存储消息，每个消息都会被发送到一个或多个Queue中。

6. **绑定Queue**

Exchange收到消息后，根据路由键将消息发送到绑定的Queue。

7. **连接实例**

客户端使用内网/公网连接RabbitMQ实例，并生产消费消息。

2 权限管理

2.1 创建用户并授权使用 DMS for RabbitMQ

如果您需要对您所拥有的DMS for RabbitMQ服务进行精细的权限管理，您可以使用[统一身份认证服务](#)（Identity and Access Management，简称IAM），通过IAM，您可以：

- 根据企业的业务组织，在您的华为账号中，给企业中不同职能部门的员工创建IAM用户，让员工拥有唯一安全凭证，并使用DMS for RabbitMQ资源。
- 根据企业用户的职能，设置不同的访问权限，以达到用户之间的权限隔离。
- 将DMS for RabbitMQ资源委托给更专业、高效的其他华为账号或者云服务，这些账号或者云服务可以根据权限进行代运维。

如果华为账号已经能满足您的要求，不需要创建独立的IAM用户，您可以跳过本章节，不影响您使用DMS for RabbitMQ服务的其它功能。

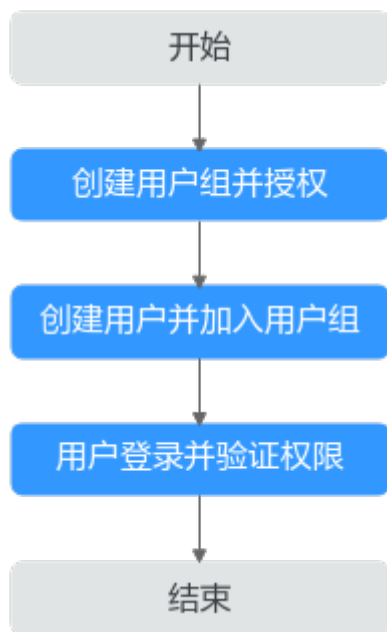
本章节为您介绍对用户授权的方法，操作流程如[图2-1](#)所示。

前提条件

给用户组授权之前，请您了解用户组可以添加的DMS for RabbitMQ系统策略，并结合实际需求进行选择，DMS for RabbitMQ支持的系统策略及策略间的对比，请参见：[DMS for RabbitMQ系统策略](#)。若您需要对除DMS for RabbitMQ之外的其它服务授权，IAM支持服务的所有策略请参见[系统权限](#)。

示例流程

图 2-1 给用户授权 DMS for RabbitMQ 权限流程



1. 创建用户组并授权

在IAM控制台创建用户组，并授予DMS for RabbitMQ的只读权限“DMS ReadOnlyAccess”。

2. 创建用户并加入用户组

在IAM控制台创建用户，并将其加入1中创建的用户组。

3. 用户登录并验证权限

新创建的用户登录控制台，切换至授权区域，验证权限：

- 在“服务列表”中选择分布式消息服务RabbitMQ版，进入RabbitMQ实例主界面，单击右上角“购买RabbitMQ实例”，尝试购买RabbitMQ实例，如果无法购买RabbitMQ实例（假设当前权限仅包含DMS ReadOnlyAccess），表示“DMS ReadOnlyAccess”已生效。
- 在“服务列表”中选择云硬盘（假设当前策略仅包含DMS ReadOnlyAccess），若提示权限不足，表示“DMS ReadOnlyAccess”已生效。
- 在“服务列表”中选择分布式消息服务RabbitMQ版（假设当前策略仅包含DMS ReadOnlyAccess），进入RabbitMQ实例主界面，如果能够查看RabbitMQ实例列表，表示“DMS ReadOnlyAccess”已生效。

DMS for RabbitMQ 自定义策略样例

如果系统预置的DMS for RabbitMQ权限，不满足您的授权要求，可以创建自定义策略。自定义策略中可以添加的授权项（Action）请参考[细粒度策略支持的授权项](#)。

目前华为云支持以下两种方式创建自定义策略：

- 可视化视图创建自定义策略：无需了解策略语法，按可视化视图导航栏选择云服务、操作、资源、条件等策略内容，可自动生成策略。

- JSON视图创建自定义策略：可以在选择策略模板后，根据具体需求编辑策略内容；也可以直接在编辑框内编写JSON格式的策略内容。

具体创建步骤请参见：[创建自定义策略](#)。下面为您介绍常用的DMS for RabbitMQ自定义策略样例。

📖 说明

DMS for RabbitMQ的权限与策略基于分布式消息服务DMS，因此在IAM服务中为DMS for RabbitMQ分配用户与权限时，请选择并使用“DMS”的权限与策略。

- 示例1：授权用户创建实例和删除实例

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "dms:instance:create",
        "dms:instance:delete"
      ]
    }
  ]
}
```

- 示例2：拒绝用户删除实例

拒绝策略需要同时配合其他策略使用，否则没有实际作用。用户被授予的策略中，一个授权项的作用如果同时存在Allow和Deny，则遵循**Deny优先原则**。

如果您给用户授予DMS FullAccess的系统策略，但不希望用户拥有DMS FullAccess中定义的删除实例权限，您可以创建一条拒绝删除实例的自定义策略，然后同时将DMS FullAccess和拒绝策略授予用户，根据Deny优先原则，则用户可以对DMS for RabbitMQ执行除了删除实例外的所有操作。拒绝策略示例如下：

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "dms:instance:delete"
      ]
    }
  ]
}
```

DMS for RabbitMQ 资源

资源是服务中存在的对象。在DMS for RabbitMQ中，资源包括：rabbitmq，您可以在创建自定义策略时，通过指定资源路径来选择特定资源。

表 2-1 DMS for RabbitMQ 的指定资源与对应路径

指定资源	资源名称	资源路径
rabbitmq	实例	<p>【格式】 DMS:*:*:rabbitmq:实例ID</p> <p>【说明】 对于实例资源，IAM自动生成资源路径前缀 DMS:*:*:rabbitmq: 通过实例ID指定具体的资源路径，支持通配符*。例如： DMS:*:*:rabbitmq:*表示任意RabbitMQ实例。</p>

DMS for RabbitMQ 请求条件

您可以在创建自定义策略时，通过添加“请求条件”（Condition元素）来控制策略何时生效。请求条件包括条件键和运算符，条件键表示策略语句的 Condition元素，分为全局级条件键和服务级条件键。[全局级条件键](#)（前缀为g:）适用于所有操作，服务级条件键（前缀为服务缩写，如dms:）仅适用于对应服务的操作。运算符与条件键一起使用，构成完整的条件判断语句。

DMS for RabbitMQ通过IAM预置了一组条件键，例如，您可以先使用dms:ssl条件键检查RabbitMQ实例是否开启SSL，然后再允许执行操作。下表显示了适用于DMS for RabbitMQ服务特定的条件键。

表 2-2 DMS for RabbitMQ 请求条件

DMS for RabbitMQ条件键	运算符	描述
dms:publicIP	Bool Null	是否开启公网
dms:ssl	Bool Null	是否开启SSL

3 购买 RabbitMQ 实例

RabbitMQ是一款基于AMQP（即Advanced Message Queuing Protocol）协议的开源服务，用于在分布式系统中存储转发消息。

RabbitMQ实例采用物理隔离的方式部署，租户独占RabbitMQ实例。支持用户自定义规格和自定义特性，您可以根据业务需要定制相应计算能力和存储空间的RabbitMQ实例。

准备实例依赖资源

购买实例前，请提前准备好如[表3-1](#)所示资源。

表 3-1 RabbitMQ 实例依赖资源

资源名称	要求	创建方法
VPC和子网	<p>RabbitMQ实例可以使用当前账号下已创建的VPC和子网，也可以使用新创建的VPC和子网，还可以使用共享VPC和子网，请根据实际需要进行配置。</p> <p>共享VPC是基于资源访问管理（Resource Access Manager，简称RAM）服务的机制，VPC的所有者可以将VPC内的子网共享给一个或者多个账号使用。通过共享VPC功能，可以简化网络配置，帮助您统一配置和运维多个账号下的资源，有助于提升资源的管控效率，降低运维成本。有关VPC子网共享的更多信息，请参见共享VPC。</p> <p>在创建VPC和子网时应注意：创建的VPC与RabbitMQ实例在相同的区域。</p>	<p>创建VPC和子网的操作指导请参考创建虚拟私有云和子网，若需要在已有VPC上创建和使用新的子网，请参考为虚拟私有云创建新的子网。</p>

资源名称	要求	创建方法
安全组	不同的RabbitMQ实例可以重复使用相同的安全组，也可以使用不同的安全组，请根据实际需要进行配置。 连接RabbitMQ实例前，请根据连接方式配置对应的安全组，具体请参考表5-2或表5-3。	创建安全组的操作指导请参考 创建安全组 ，为安全组添加规则的操作指导请参考 添加安全组规则 。
弹性IP地址	如果客户端使用公网连接RabbitMQ实例，请提前创建弹性IP地址。 在创建弹性IP地址时，应注意如下要求： <ul style="list-style-type: none"> • 创建的弹性IP地址与RabbitMQ实例在相同的区域。 • RabbitMQ控制台无法识别开启IPv6转换功能的弹性IP地址。 	创建弹性IP地址的操作指导请参考 申请弹性公网IP 。

购买 RabbitMQ 实例（3.8.35 版本）

步骤1 进入[购买RabbitMQ实例页面](#)。

步骤2 选择“计费模式”。

- 包年/包月：您创建RabbitMQ实例时，可以根据需要设置购买时长，系统会一次性按照购买价格进行扣费。
- 按需付费：您创建RabbitMQ实例时，无需设置购买时长，系统会根据消费时长进行扣费。

步骤3 选择“区域”。

不同区域的云服务产品之间内网互不相通。请就近选择靠近您业务的区域，可减少网络时延，提高访问速度。

步骤4 选择“项目”。

每个区域默认对应一个项目，这个项目由系统预置，用来隔离物理区域间的资源（计算资源、存储资源和网络资源）。

步骤5 选择“可用区”。

可用区指在同一区域下，电力、网络隔离的物理区域，可用区之间内网互通，不同可用区之间物理隔离。

根据实际情况选择1个或者3个及以上可用区。实例创建成功后，不支持更换可用区。

步骤6 设置“实例名称”。

名称支持自定义，但需要符合命名规则：长度为4~64个字符，由英文字母开头，只能由英文字母、数字、中划线、下划线组成。

步骤7 设置“企业项目”。

该参数针对企业用户使用。企业项目是一种云资源管理方式，企业项目管理服务提供统一的云资源按项目管理，以及项目内的资源管理、成员管理，默认项目为default。

步骤8 设置实例信息。

1. 版本：RabbitMQ的版本号，选择“3.8.35”。
2. 部署架构：支持“单机”和“集群”。
 - 单机：表示部署一个RabbitMQ代理。
 - 集群：表示部署多个RabbitMQ代理，实现高可靠的消息存储。
3. 代理规格：根据实际情况选择规格。更多关于规格的介绍，请参见[产品规格](#)。

说明

为了保证服务的稳定可靠，RabbitMQ采用了默认的40%高水位配置。当内存占用率到达40%后，可能会触发内存高水位，内存高水位会触发流控，导致生产者发送消息被阻塞。为了避免高水位的产生，请及时消费积压在队列中的消息。

4. 代理数量：根据实例情况选择代理个数。
5. 单个代理存储空间：选择RabbitMQ实例的磁盘类型和磁盘大小。

如何选择磁盘类型请参考[磁盘类型及性能介绍](#)。

- 单机实例的取值范围：100GB~30000GB。
 - 集群实例的取值范围：代理数*（100GB~30000GB）。
6. 虚拟私有云：选择已经创建好的虚拟私有云和子网，或者共享的虚拟私有云和子网。

虚拟私有云可以为您的RabbitMQ实例构建隔离的、能自主配置和管理的虚拟网络环境。

说明

虚拟私有云和子网在RabbitMQ实例创建完成后，不支持修改。

7. 安全组：选择已经创建好的安全组。

安全组是一组对RabbitMQ实例的访问规则的集合。您可以单击右侧的“管理安全组”，跳转到网络控制台的安全组页面，查看或创建安全组。
8. 设置“SSL”。

客户端连接实例时SSL认证的开关。开启SSL，则数据加密传输，安全性更高。**SSL开关在实例创建完成后不支持修改，请明确是否需要开启。**

图 3-1 设置实例信息

The screenshot shows the configuration interface for a RabbitMQ instance. It includes sections for deployment architecture (单机/集群), instance specifications (代理规格), instance quantity (代理数量), storage (单个代理存储空间), virtual private cloud (虚拟私有云), security group (安全组), and SSL settings.

规格名称	单个代理最大连接数	单个代理建议队列数
<input checked="" type="radio"/> rabbitmq.2u4g.single	3,000	200
<input type="radio"/> rabbitmq.4u8g.single	4,500	400
<input type="radio"/> rabbitmq.8u16g.single	7,500	800
<input type="radio"/> rabbitmq.16u32g.single	12,000	1,600
<input type="radio"/> rabbitmq.24u48g.single	15,000	2,400

当前选择规格: rabbitmq.2u4g.single | 单个代理最大连接数 3,000 | 单个代理建议队列数 200

代理数量: 1

单个代理存储空间: 100 GB

虚拟私有云: sg-53d4, subnet-38a0 (192.168.0.0/24) 可用 IP 251 个

安全组: default

SSL: 关闭

步骤9 设置连接RabbitMQ实例的用户名和密码。

用户名需要符合以下命名规则：由英文字母开头，且只能由英文字母、数字、中划线、下划线组成，长度为4~64个字符。

密码需要符合以下命名规则：

- 长度为8~32个字符。
- 至少包含以下字符中的3种：大写字母、小写字母、数字、特殊字符`~!@#%&*()-_+=\|{[]:;'"<.>?`和空格，并且不能以-开头。
- 不能与用户名或倒序的用户名相同。

步骤10 设置实例购买时长。

当选择了“包年/包月”计费模式时，页面才显示“购买时长”参数，您需要根据业务需要选择。

步骤11 单击“更多配置”，设置更多相关信息。

1. 设置“公网访问”。

您可以选择是否打开公网访问开关。

如果选择了开启，表示访问RabbitMQ实例可以通过弹性IP访问。这时页面会显示“弹性IP地址”，在“弹性IP地址”区域，您可下拉选择已有的弹性IP。另外，您可单击右侧的“创建弹性IP”，跳转到网络控制台的弹性公网IP页面，购买弹性公网IP。

说明

- 公网访问与VPC内访问相比，可能存在网络丢包和抖动等情况，且访问时延有所增加，因此建议仅在业务开发测试阶段开启公网访问进行调试。
- 如果用户在虚拟私有云的服务页面手动解绑定或删除EIP，相应RabbitMQ实例的公网访问功能会自动关闭。

2. 设置“标签”。

标签用于标识云资源，当您拥有相同类型的许多云资源时，可以使用标签按各种维度（例如用途、所有者或环境）对云资源进行分类。

如您的组织已经设定分布式消息服务RabbitMQ版的相关标签策略，则需按照标签策略规则为RabbitMQ实例添加标签。标签如果不符合标签策略的规则，则可能会导致RabbitMQ实例创建失败，请联系组织管理员了解标签策略详情。

- 如果您已经预定义了标签，在“标签键”和“标签值”中选择已经定义的标签键值对。另外，您可以单击右侧的“查看预定义标签”，系统会跳转到标签管理服务页面，查看已经预定义的标签，或者创建新的标签。
- 您也可以直接在“标签键”和“标签值”中设置标签。

当前每个RabbitMQ实例最多支持设置20个不同标签，标签的命名规格，请参考[配置RabbitMQ实例标签](#)章节。

3. 设置实例的描述信息。

步骤12 填写完上述信息后，单击“立即购买”，进入“规格确认”页面。

步骤13 确认实例信息无误且阅读并同意《华为云用户协议》后，如果“计费模式”选择“包年/包月”，单击“去支付”。根据界面提示信息，支付费用。如果“计费模式”选择“按需付费”，单击“提交”。

步骤14 在实例列表页面查看实例是否创建成功。

创建实例大约需要3到15分钟，此时实例的“状态”为“创建中”。

- 当实例的“状态”变为“运行中”时，说明实例创建成功。
- 当实例的“状态”变为“创建失败”，请参考[删除RabbitMQ实例](#)，删除创建失败的RabbitMQ实例，然后重新购买。如果重新购买仍然失败，请联系客服。

---结束

购买 RabbitMQ 实例（AMQP-0-9-1 版本）

步骤1 进入[购买RabbitMQ实例页面](#)。

步骤2 选择“计费模式”。

- 包年/包月：您创建RabbitMQ实例时，可以根据需要设置购买时长，系统会一次性按照购买价格进行扣费。
- 按需付费：您创建RabbitMQ实例时，无需设置购买时长，系统会根据消费时长进行扣费。

步骤3 选择“区域”。

不同区域的云服务产品之间内网互不相通。请就近选择靠近您业务的区域，可减少网络时延，提高访问速度。

步骤4 选择“项目”。

每个区域默认对应一个项目，这个项目由系统预置，用来隔离物理区域间的资源（计算资源、存储资源和网络资源）。

步骤5 选择“可用区”。

可用区指在同一区域下，电力、网络隔离的物理区域，可用区之间内网互通，不同可用区之间物理隔离。

实例创建成功后，不支持更换可用区。

步骤6 设置“实例名称”。

名称支持自定义，但需要符合命名规则：长度为4~64个字符，由英文字母开头，只能由英文字母、数字、中划线、下划线组成。

步骤7 设置“企业项目”。

该参数针对企业用户使用。企业项目是一种云资源管理方式，企业项目管理服务提供统一的云资源按项目管理，以及项目内的资源管理、成员管理，默认项目为default。

步骤8 设置实例信息。

1. 版本：RabbitMQ的版本号，选择“AMQP-0-9-1”。
2. 实例类型：选择“专业版”。
3. 部署架构：支持“单机”和“集群”。
 - 单机：表示部署一个RabbitMQ代理。
 - 集群：表示部署多个RabbitMQ代理，实现高可靠的消息存储。
4. 实例规格：根据实际情况选择规格。
5. 存储空间：选择RabbitMQ实例所有代理的磁盘类型和存储空间总和。
如何选择磁盘类型请参考[磁盘类型及性能介绍](#)。
 - 单机实例的取值范围：100GB~30000GB。
 - 集群实例的取值范围：200GB~60000GB。
6. 虚拟私有云：选择已经创建好的虚拟私有云和子网，或者共享的虚拟私有云和子网。
虚拟私有云可以为您的RabbitMQ实例构建隔离的、能自主配置和管理的虚拟网络环境。

说明

虚拟私有云和子网在RabbitMQ实例创建完成后，不支持修改。

7. 安全组：选择已经创建好的安全组。
安全组是一组对RabbitMQ实例的访问规则的集合。您可以单击右侧的“管理安全组”，跳转到网络控制台的安全组页面，查看或创建安全组。
8. ACL访问控制：通过ACL（Access Control List）可以实现对RabbitMQ实例中生产者和服务消费者身份的精确识别，为不同用户赋予不同的Vhost和Vhost下不同资源的权限，以达到不同用户之间权限隔离的目的。开启ACL权限控制功能后，生产消息和消费消息时，需要鉴权。

步骤9 设置实例购买时长。

当选择了“包年/包月”计费模式时，页面才显示“购买时长”参数，您需要根据业务需要选择。

步骤10 单击“更多配置”，设置更多相关信息。

1. 设置“公网访问”。
RabbitMQ AMQP-0-9-1版本实例不支持公网访问。
2. 设置“标签”。
标签用于标识云资源，当您拥有相同类型的许多云资源时，可以使用标签按各种维度（例如用途、所有者或环境）对云资源进行分类。

如果您的组织已经设定分布式消息服务RabbitMQ版的相关标签策略，则需按照标签策略规则为RabbitMQ实例添加标签。标签不符合标签策略的规则，则可能会导致RabbitMQ实例创建失败，请联系组织管理员了解标签策略详情。

- 如果您已经预定义了标签，在“标签键”和“标签值”中选择已经定义的标签键值对。另外，您可以单击右侧的“查看预定义标签”，系统会跳转到标签管理服务页面，查看已经预定义的标签，或者创建新的标签。
- 您也可以直接在“标签键”和“标签值”中设置标签。

当前每个RabbitMQ实例最多支持设置20个不同标签，标签的命名规格，请参考[配置RabbitMQ实例标签](#)章节。

3. 设置实例的描述信息。

步骤11 填写完上述信息后，单击“立即购买”，进入“规格确认”页面。

步骤12 确认实例信息无误且阅读并同意《华为云用户协议》后，如果“计费模式”选择“包年/包月”，单击“去支付”。根据界面提示信息，支付费用。如果“计费模式”选择“按需付费”，单击“提交”。

步骤13 在实例列表页面查看实例是否创建成功。

创建实例大约需要3到15分钟，此时实例的“状态”为“创建中”。

- 当实例的“状态”变为“运行中”时，说明实例创建成功。
- 当实例的“状态”变为“创建失败”，请参考[删除RabbitMQ实例](#)，删除创建失败的RabbitMQ实例，然后重新购买。如果重新购买仍然失败，请联系客服。

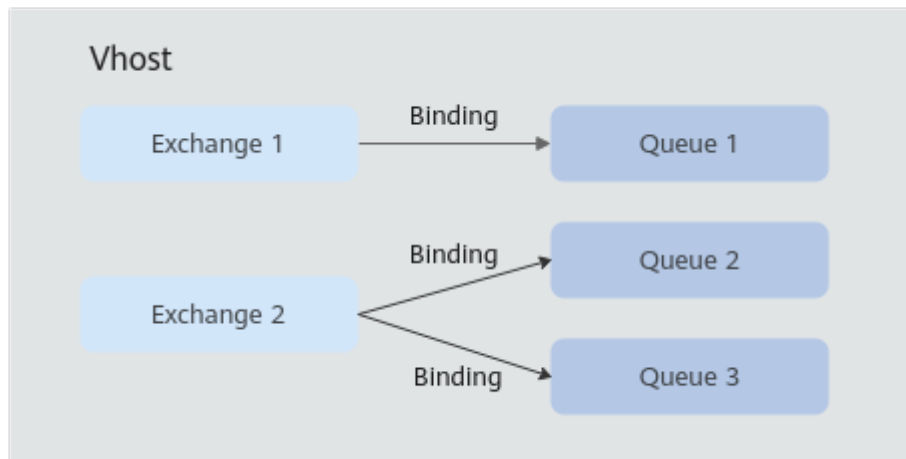
---结束

4 配置 Vhost

4.1 创建 RabbitMQ Vhost

每个Vhost（Virtual Hosts）相当于一个相对独立的RabbitMQ服务器。Vhost用作逻辑隔离，分别管理Exchange、Queue和Binding，使得应用安全地运行在不同的Vhost上，相互之间不会干扰。一个实例下可以有多个Vhost，一个Vhost里可以有若干个Exchange和Queue。生产者和消费者连接RabbitMQ实例时，需要指定一个Vhost。Vhost的相关介绍，请参考官网文档[Virtual Hosts](#)。

图 4-1 Vhost 架构图



本章节主要介绍创建Vhost的操作，有以下几种方式，您可以根据实际情况选择任意一种方式：

- [创建RabbitMQ Vhost（控制台）](#)
- [创建RabbitMQ Vhost（RabbitMQ WebUI）](#)


📖 说明

实例创建后，会自动创建一个名为“/”的Vhost。

- RabbitMQ 3.x.x版本支持以上两种创建Vhost的方式，RabbitMQ AMQP-0-9-1版本只支持在控制台创建Vhost。
- 实例创建后，RabbitMQ 3.x.x版本会自动创建一个名为“/”的Vhost，RabbitMQ AMQP-0-9-1版本会自动创建一个名为“default”的Vhost。


创建 RabbitMQ Vhost (控制台)

步骤1 登录管理控制台。

步骤2 在管理控制台左上角单击 ，选择区域。

📖 说明

此处请选择RabbitMQ实例所在的区域。

步骤3 在管理控制台左上角单击 ，选择“应用中间件 > 分布式消息服务RabbitMQ版”，进入分布式消息服务RabbitMQ专享版页面。

步骤4 单击实例名称，进入实例详情页面。

步骤5 在左侧导航栏选择“Vhost列表”，进入Vhost列表页面。

步骤6 单击“创建Vhost”，弹出“创建Vhost”对话框。

步骤7 设置Vhost的名称，单击“确定”。

Vhost创建成功后，无法修改名称，在Vhost列表页面显示创建成功的Vhost。

“Tracing”表示是否开启消息追踪功能，仅RabbitMQ 3.x.x版本支持。开启消息追踪后，您可以跟踪消息的转发路径。

图 4-2 Vhost 列表 (RabbitMQ 3.x.x 版本)

<input type="checkbox"/> 名称	Tracing	操作
<input type="checkbox"/> /	否	删除
<input type="checkbox"/> Vhost-68650386	否	删除

图 4-3 Vhost 列表 (RabbitMQ AMQP-0-9-1 版本)

<input type="checkbox"/> 名称	操作
<input type="checkbox"/> default	删除
<input type="checkbox"/> Vhost-19498975	删除

----结束

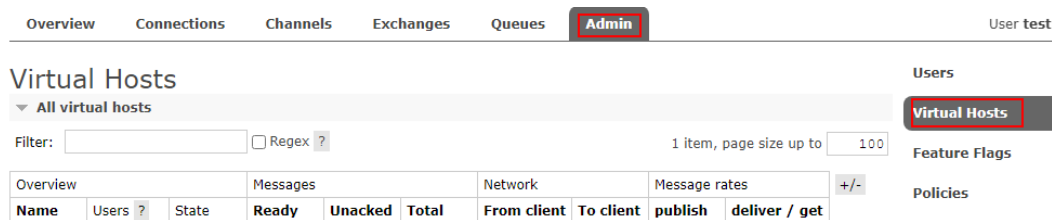
创建 RabbitMQ Vhost (RabbitMQ WebUI)

步骤1 登录RabbitMQ WebUI。

步骤2 在顶部导航栏选择“Admin”，进入Admin页面。

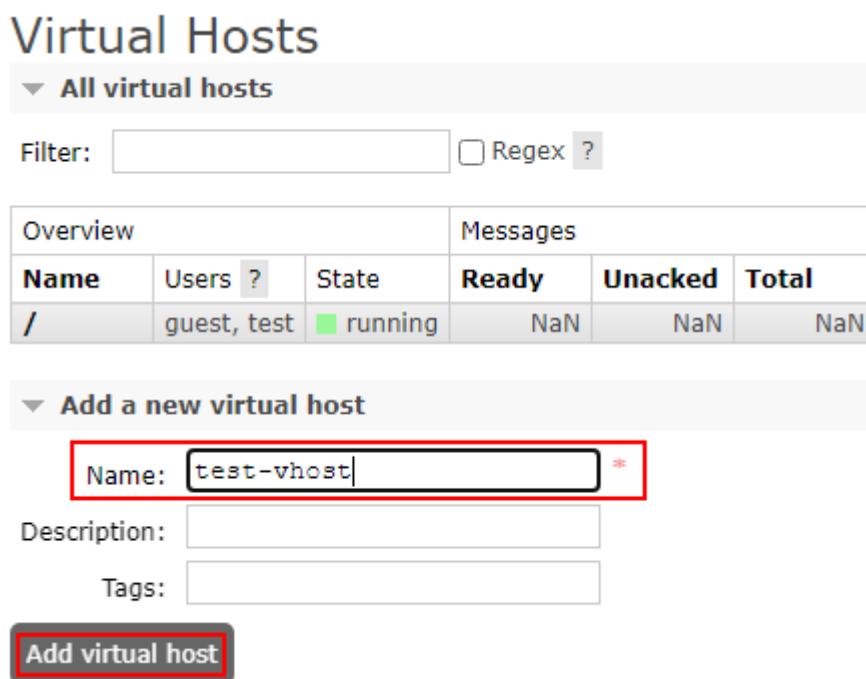
步骤3 在右侧导航栏选择“Virtual Hosts”，进入Virtual Hosts页面。

图 4-4 Virtual Hosts



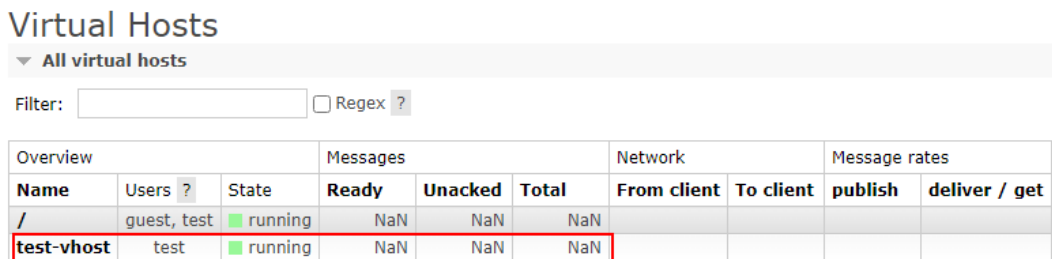
步骤4 在“Add a new virtual host”区域，输入Vhost名称，单击“Add virtual host”。

图 4-5 创建 Vhost (WebUI)



创建成功后，在“All virtual hosts”区域，显示创建成功的Vhost。

图 4-6 Vhost 列表 (WebUI)



----结束

4.2 创建 RabbitMQ Exchange

Exchange用于接收、分配消息。生产者向分布式消息服务RabbitMQ版发送消息时，不会直接将消息发送到Queue，而是先将消息发送到Exchange中，Exchange根据路由键查找Queue，如果查找到，将消息存放到Queue中，如果未查找到，将消息丢弃。


本章节指导如何在控制台创建Exchange。RabbitMQ 3.x.x版本在创建Vhost后会创建7个默认Exchange：(AMQP default)、amq.direct、amq.fanout、amq.headers、amq.match、amq.rabbitmq.trace、amq.topic。

前提条件

已创建Vhost。


创建 RabbitMQ Exchange

步骤1 登录管理控制台。

步骤2 在管理控制台左上角单击 ，选择区域。

说明

此处请选择RabbitMQ实例所在的区域。

步骤3 在管理控制台左上角单击 ，选择“应用中间件 > 分布式消息服务RabbitMQ版”，进入分布式消息服务RabbitMQ专享版页面。

步骤4 单击实例名称，进入实例详情页面。

步骤5 在左侧导航栏选择“Vhost列表”，进入Vhost列表页面。

步骤6 单击Vhost名称，进入Vhost详情页。

步骤7 在“Exchange”页签中，单击“创建Exchange”，弹出“创建Exchange”对话框。

步骤8 参考表4-1，设置Exchange名称和配置信息。

表 4-1 Exchange 参数说明

参数	说明
名称	系统为您自动生成了Exchange名称，您可以根据需要修改。Exchange名称需要符合以下规则：长度为3~128个字符，只能由英文字母、数字、百分号、竖线、中划线、下划线、斜杠、反斜杠组成。 创建Exchange后不能修改名称。

参数	说明
类型	<p>选择路由类型。关于Exchange类型更详细的介绍，请参见Exchange类型介绍。</p> <ul style="list-style-type: none"> • direct: 该类型Exchange会将消息路由到Routing Key完全匹配的Queue中。 • fanout: 该类型Exchange会将消息路由到所有与其绑定的Queue中。 • topic: 该类型Exchange将Routing Key进行通配符匹配，然后将消息路由到匹配成功的Queue中。 • headers: 该类型Exchange与Routing Key无关，而与消息中的Headers属性信息相关。Exchange根据消息中的Headers属性键值对和绑定的属性键值对进行匹配，根据匹配情况路由消息。 • x-delayed-message: 该类型Exchange的作用为延迟投递消息，根据Exchange类型的路由规则路由消息。 • x-consistent-hash: 该类型Exchange会根据Routing Key计算出一个hash值，按照hash值将消息路由到对应的Queue中。
Exchange类型	<p>“类型”设置为“x-delayed-message”时，页面显示此参数。x-delayed-message Exchange根据设定的Exchange类型的路由规则路由消息。</p> <ul style="list-style-type: none"> • direct: 该类型Exchange会将消息路由到Routing Key完全匹配的Queue中。 • fanout: 该类型Exchange会将消息路由到所有与其绑定的Queue中。 • topic: 该类型Exchange将Routing Key进行通配符匹配，然后将消息路由到匹配成功的Queue中。 • headers: 该类型Exchange与Routing Key无关，而与消息中的Headers属性信息相关。Exchange根据消息中的Headers属性键值对和绑定的属性键值对进行匹配，根据匹配情况路由消息。
自动删除	<p>是否开启自动删除Exchange。</p> <ul style="list-style-type: none"> • 开启: 当最后一个绑定的Queue与Exchange解除绑定后，该Exchange会被自动删除。 • 不开启: 当最后一个绑定的Queue与Exchange解除绑定后，不会删除该Exchange。
持久化	<p>RabbitMQ 3.x.x版本需要设置此参数。RabbitMQ AMQP-0-9-1版本Exchange默认开启持久化。</p> <p>是否开启Exchange持久化。</p> <ul style="list-style-type: none"> • 开启: 该Exchange在服务器重启后仍然存在。 • 不开启: 该Exchange在服务器重启后会被删除，需要重新创建。

参数	说明
Internal	<p>仅RabbitMQ 3.x.x版本包含此参数。 Exchange是否为RabbitMQ内部使用。</p> <ul style="list-style-type: none"> 是：该Exchange只能绑定其他Exchange，不能绑定Queue。 否：该Exchange可以绑定Exchange和Queue。

步骤9 单击“确定”，完成Exchange的创建。

在“Exchange”页签中，查看新创建的Exchange。

----结束

4.3 绑定 RabbitMQ Exchange

绑定Exchange是将Exchange和Exchange/Queue关联起来，在生产者发送消息到Exchange后，Exchange根据路由规则将消息发送到与其关联的Exchange/Queue中。

本章节指导如何在控制台绑定Exchange，支持为Exchange绑定目标Exchange，或者为Queue绑定源Exchange。一个Exchange可以绑定多个目标Exchange，一个Queue可以绑定多个源Exchange。

约束与限制


- RabbitMQ 3.x.x版本中，名为“(AMQP default)”的Exchange不能绑定任何Exchange。
- RabbitMQ AMQP-0-9-1版本的Exchange不支持绑定Exchange，只支持绑定Queue。
- “Internal”为“是”的Exchange只能绑定Exchange，不能绑定Queue。

前提条件

已[创建Exchange](#)。


为 Exchange 绑定目标 Exchange

步骤1 登录管理控制台。

步骤2 在管理控制台左上角单击 ，选择区域。

说明

此处请选择RabbitMQ实例所在的区域。

步骤3 在管理控制台左上角单击 ，选择“应用中间件 > 分布式消息服务RabbitMQ版”，进入分布式消息服务RabbitMQ专享版页面。

步骤4 单击实例名称，进入实例详情页面。

步骤5 在左侧导航栏选择“Vhost列表”，进入Vhost列表页面。



- 步骤6** 单击Vhost名称，进入Vhost详情页。
- 步骤7** 在“Exchange”页签中，在待绑定的Exchange后，单击“绑定”，弹出“绑定”页面。
- 步骤8** 单击“添加绑定”，弹出“添加绑定”对话框。
- 步骤9** 参考表4-2，设置绑定参数。

表 4-2 绑定参数说明

参数	说明
类型	仅RabbitMQ 3.x.x版本包含此参数，RabbitMQ AMQP-0-9-1版本只支持绑定Queue。 选择Exchange绑定的类型。绑定Exchange时选择“Exchange”。
绑定目标	在下拉框中选择需要绑定的目标Exchange。
Routing Key	设置Key值字符串，用于告知Exchange应该将消息投递到哪些目标Exchange中。 <ul style="list-style-type: none"> Direct Exchange和Topic Exchange需要设置Routing Key，Exchange会将消息路由到Routing Key匹配成功的目标Exchange中。如果不设置Routing Key，Exchange会将消息路由到所有与它绑定的目标Exchange中。 x-consistent-hash Exchange需要设置Routing Key，Routing Key表示Queue的权重。数值越大，Queue的权重越大，分发消息时接收到的消息越多。 Fanout Exchange和Header Exchange无需设置Routing Key，Routing Key对于这两种Exchange不起作用。

- 步骤10** 单击“确定”，完成Exchange的绑定。
在“绑定”页面，查看新绑定的Exchange。
----结束

为 Queue 绑定源 Exchange

- 步骤1** 登录管理控制台。
- 步骤2** 在管理控制台左上角单击 ，选择区域。
- 说明**
此处请选择RabbitMQ实例所在的区域。
- 步骤3** 在管理控制台左上角单击 ，选择“应用中间件 > 分布式消息服务RabbitMQ版”，进入分布式消息服务RabbitMQ专享版页面。
- 步骤4** 单击实例名称，进入实例详情页面。
- 步骤5** 在左侧导航栏选择“Vhost列表”，进入Vhost列表页面。

- 步骤6** 单击Vhost名称，进入Vhost详情页。
- 步骤7** 在“Queue”页签的待绑定的Queue所在行，单击“查看详情”，弹出“查看详情”页面。
- 步骤8** 在“绑定信息”页签，单击“添加绑定”，弹出“添加绑定”对话框。
- 步骤9** 参考表4-3，设置绑定参数。

表 4-3 绑定参数说明

参数	说明
绑定源端	在下拉框中选择需要绑定的Exchange。
Routing Key	设置Key值字符串，用于告知Exchange应该将消息投递到哪些Queue中。 <ul style="list-style-type: none">• Direct Exchange和Topic Exchange需要设置Routing Key，Exchange会将消息路由到Routing Key匹配成功的Queue中。如果不设置Routing Key，Exchange会将消息路由到所有与它绑定的Queue中。• x-consistent-hash Exchange需要设置Routing Key，Routing Key表示Queue的权重。数值越大，Queue的权重越大，分发消息时接收到的消息越多。• Fanout Exchange和Header Exchange无需设置Routing Key，Routing Key对于这两种Exchange不起作用。

- 步骤10** 单击“确定”，完成Exchange的绑定。
在“绑定信息”页签，查看新绑定的Exchange。
----结束

4.4 创建 RabbitMQ Queue

Queue用于存储消息，每个消息都会被发送到一个或多个Queue中，生产者生产消息并最终投递到Queue中，消费者可以从Queue中拉取消息进行消费。


多个消费者可以订阅同一个Queue，此时Queue中的消息会被平均分摊给多个消费者进行处理，而不是每个消费者都收到所有的消息。

本章节指导如何在控制台创建Queue。

前提条件

已创建Vhost。

创建 RabbitMQ Queue

- 步骤1** 登录管理控制台。
- 步骤2** 在管理控制台左上角单击 ，选择区域。

说明

此处请选择RabbitMQ实例所在的区域。


- 步骤3** 在管理控制台左上角单击, 选择“应用中间件 > 分布式消息服务RabbitMQ版”, 进入分布式消息服务RabbitMQ专享版页面。
- 步骤4** 单击实例名称, 进入实例详情页面。
- 步骤5** 在左侧导航栏选择“Vhost列表”, 进入Vhost列表页面。
- 步骤6** 单击Vhost名称, 进入Vhost详情页。
- 步骤7** 在“Queue”页签中, 单击“创建Queue”, 弹出“创建Queue”对话框。
- 步骤8** 参考表4-4, 设置Queue的名称和配置信息。

表 4-4 Queue 参数说明

参数	说明
名称	系统为您自动生成了Queue名称, 您可以根据需要修改。Queue名称需要符合以下规则: 长度为3~128个字符, 只能由英文字母、数字、百分号、竖线、中划线、下划线、斜杠、反斜杠组成。 创建Queue后不能修改名称。
持久化	RabbitMQ 3.x.x版本需要设置此参数。RabbitMQ AMQP-0-9-1版本Queue默认开启持久化。 是否开启Queue持久化。 <ul style="list-style-type: none"> 开启: 该Queue在服务器重启后仍然存在。 不开启: 该Queue在服务器重启后会被删除, 需要重新创建。
自动删除	是否开启自动删除Queue。 <ul style="list-style-type: none"> 开启: 在订阅该Queue消息的最后一个消费者取消订阅该Queue的消息后, 该Queue会被自动删除。 不开启: 在订阅该Queue消息的最后一个消费者取消订阅该Queue的消息后, 不会删除该Queue。
死信交换器	在下拉框中选择死信消息发送的Exchange。
死信路由键	设置死信消息的Routing Key, 即死信Exchange会将消息发送至匹配该死信Routing Key的Binding Key所对应的Queue。
消息过期时间	消息的存活时间, 单位为ms。超过消息过期时间, 消息还未被成功消费, 就会变为死信消息, 该消息将会被发送到死信Exchange。
最大优先级	仅RabbitMQ AMQP-0-9-1版本包含此参数。 设置Queue的优先级, 取值范围为1~9。数值越大, 优先级越高。

参数	说明
惰性队列模式	<p>仅RabbitMQ 3.x.x版本包含此参数。</p> <p>如果需要设置Queue为惰性队列，请输入“lazy”。</p> <p>惰性队列会尽可能的将消息存入磁盘中，在消费者消费到相应的消息时才会被加载到内存中，这样可以减少内存的消耗。</p>

步骤9 单击“确定”，完成Queue的创建。

在“Queue”页签中，查询新创建的Queue。

----结束

4.5 绑定 RabbitMQ Queue

绑定Queue是将Exchange和Queue关联起来，在生产者发送消息到Exchange后，Exchange根据路由规则将消息发送到与其关联的Queue中。

本章节指导如何在控制台为Exchange绑定Queue，只有绑定Queue后，Exchange才能把消息路由并存放到Queue。一个Exchange可以绑定多个Queue。

约束与限制


- RabbitMQ 3.x.x版本中，名为“(AMQP default)”的Exchange不能绑定任何Queue。
- “Internal”为“是”的Exchange只能绑定Exchange，不能绑定Queue。

前提条件

- 已[创建Exchange](#)。
- 已[创建Queue](#)。


为 Exchange 绑定 Queue

步骤1 登录管理控制台。

步骤2 在管理控制台左上角单击 ，选择区域。

说明

此处请选择RabbitMQ实例所在的区域。

步骤3 在管理控制台左上角单击 ，选择“应用中间件 > 分布式消息服务RabbitMQ版”，进入分布式消息服务RabbitMQ专享版页面。

步骤4 单击实例名称，进入实例详情页面。

步骤5 在左侧导航栏选择“Vhost列表”，进入Vhost列表页面。

步骤6 单击Vhost名称，进入Vhost详情页。

步骤7 在“Exchange”页签中，在待绑定的Exchange后，单击“绑定”，弹出“绑定”页面。

步骤8 单击“添加绑定”，弹出“添加绑定”对话框。

步骤9 参考表4-5，设置绑定参数。

表 4-5 绑定参数说明

参数	说明
类型	仅RabbitMQ 3.x.x版本包含此参数。RabbitMQ AMQP-0-9-1版本只支持绑定Queue。 选择Exchange绑定的类型。绑定Queue时选择“Queue”。
绑定目标	在下拉框中选择需要绑定的目标Queue。
Routing Key	设置Key值字符串，用于告知Exchange应该将消息投递到哪些Queue中。 <ul style="list-style-type: none"> • Direct Exchange和Topic Exchange需要设置Routing Key，Exchange会将消息路由到Routing Key匹配成功的Queue中。如果不设置Routing Key，Exchange会将消息路由到所有与它绑定的Queue中。 • x-consistent-hash Exchange需要设置Routing Key，Routing Key表示Queue的权重。数值越大，Queue的权重越大，分发消息时接收到的消息越多。 • Fanout Exchange和Header Exchange无需设置Routing Key，Routing Key对于这两种Exchange不起作用。

步骤10 单击“确定”，完成Queue的绑定。

在“绑定”页面，查看新绑定的Queue。

----结束


4.6 管理 RabbitMQ Vhost

4.6.1 查看 RabbitMQ Vhost 信息

Vhost创建成功后，可以在控制台查看Vhost中包含的Exchange和Queue的数量，以及详细信息。


查看 RabbitMQ Vhost 信息

步骤1 登录管理控制台。

步骤2 在管理控制台左上角单击 ，选择区域。

说明

此处请选择RabbitMQ实例所在的区域。

步骤3 在管理控制台左上角单击 ，选择“应用中间件 > 分布式消息服务RabbitMQ版”，进入分布式消息服务RabbitMQ专享版页面。

- 步骤4** 单击实例名称，进入实例详情页面。
- 步骤5** 在左侧导航栏选择“Vhost列表”，进入Vhost列表页面。
- 步骤6** 单击Vhost名称，进入Vhost详情页。
- 步骤7** 在顶部Exchange、Queue下显示的数字表示此Vhost下的Exchange、Queue数量，在“Exchange”/“Queue”页签中查看Exchange、Queue的详细信息。

图 4-7 Vhost 详情页



----结束

4.6.2 删除 RabbitMQ Vhost


本章节指导如何删除Vhost。删除Vhost后，该Vhost下的所有资源（包括Exchange、Queue）也会被删除，且无法恢复。

删除Vhost的操作有以下几种方式，您可以根据实际情况选择任意一种方式：

- [删除Vhost（控制台）](#)
- [删除Vhost（RabbitMQ WebUI）](#)


RabbitMQ 3.x.x版本支持以上两种删除Vhost的方式，RabbitMQ AMQP-0-9-1版本只支持在控制台删除Vhost。

删除 Vhost（控制台）

- 步骤1** 登录管理控制台。
- 步骤2** 在管理控制台左上角单击 ，选择区域。

说明

此处请选择RabbitMQ实例所在的区域。

- 步骤3** 在管理控制台左上角单击 ，选择“应用中间件 > 分布式消息服务RabbitMQ版”，进入分布式消息服务RabbitMQ专享版页面。
- 步骤4** 单击实例名称，进入实例详情页面。
- 步骤5** 在左侧导航栏选择“Vhost列表”，进入Vhost列表页面。
- 步骤6** 通过以下任何一种方法，删除Vhost。
 - 勾选Vhost名称左侧的方框，可选一个或多个，单击信息栏左上侧的“删除Vhost”。

- 在待删除的Vhost所在行，单击“删除”。
- 单击待删除的Vhost名称，进入Vhost详情页。在右上角单击“删除”。

说明

- 创建实例时创建的默认Vhost，不支持删除。
- 删除Vhost时，该Vhost下的所有资源（包括Exchange、Queue）会被同步删除，且无法恢复。

步骤7 在弹出的确认删除对话框中，单击“确定”。

----结束

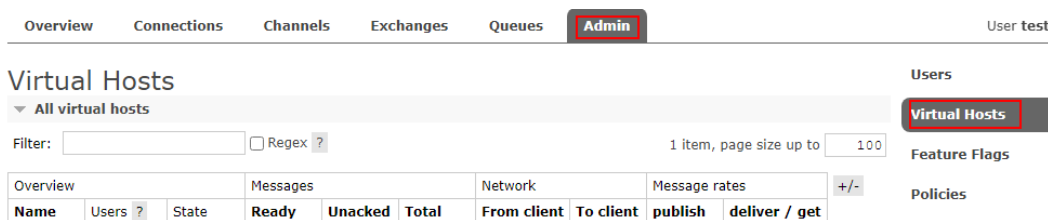
删除 Vhost (RabbitMQ WebUI)

步骤1 登录RabbitMQ WebUI。

步骤2 在顶部导航栏选择“Admin”，进入Admin页面。

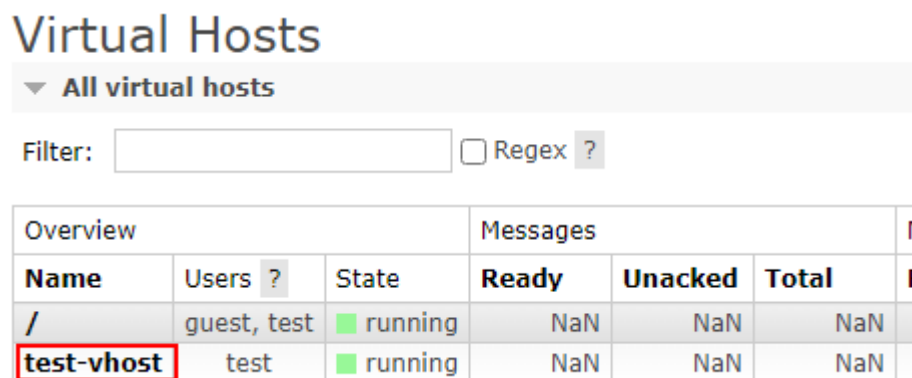
步骤3 在右侧导航栏选择“Virtual Hosts”，进入Virtual Hosts页面。

图 4-8 Virtual Hosts 页面



步骤4 单击待删除的Vhost名称，进入Vhost详情页。

图 4-9 待删除的 Vhost



步骤5 在“Delete this vhost”区域，单击“Delete this virtual host”，弹出确认删除对话框。

图 4-10 删除 Vhost

Virtual Host: test-vhost

▶ Overview

▶ Permissions

▶ Topic permissions

▼ Delete this vhost

Delete this virtual host

步骤6 单击“确定”，完成Vhost的删除。

----结束

4.7 管理 RabbitMQ Exchange

4.7.1 解绑 RabbitMQ Exchange


本章节指导如何在控制台解除绑定Exchange，支持为Exchange解绑目标Exchange，或者为Queue解绑源Queue。

前提条件

- 已[创建Exchange](#)。
- Exchange或Queue已[绑定Exchange](#)。


为 Exchange 解绑目标 Exchange

步骤1 登录管理控制台。

步骤2 在管理控制台左上角单击 ，选择区域。

说明

此处请选择RabbitMQ实例所在的区域。

步骤3 在管理控制台左上角单击 ，选择“应用中间件 > 分布式消息服务RabbitMQ版”，进入分布式消息服务RabbitMQ专享版页面。

步骤4 单击实例名称，进入实例详情页面。

步骤5 在左侧导航栏选择“Vhost列表”，进入Vhost列表页面。

步骤6 单击Vhost名称，进入Vhost详情页。

步骤7 在“Exchange”页签中，在Exchange后，单击“绑定”，弹出“绑定”页面。

步骤8 在待解绑的Exchange后，单击“解除绑定”，弹出“解除绑定”对话框。

须知


解除绑定后，该路由将不再提供服务，且无法恢复，请谨慎操作。

步骤9 单击“是”，完成Exchange的解绑。

----结束


为 Queue 解绑源 Exchange

步骤1 登录管理控制台。

步骤2 在管理控制台左上角单击 ，选择区域。

说明

此处请选择RabbitMQ实例所在的区域。

步骤3 在管理控制台左上角单击 ，选择“应用中间件 > 分布式消息服务RabbitMQ版”，进入分布式消息服务RabbitMQ专享版页面。

步骤4 单击实例名称，进入实例详情页面。

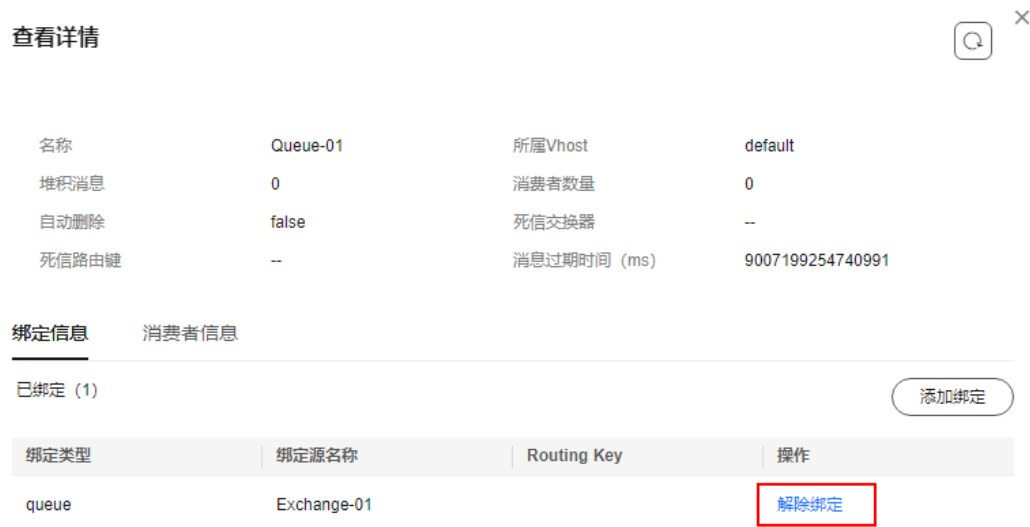
步骤5 在左侧导航栏选择“Vhost列表”，进入Vhost列表页面。

步骤6 单击Vhost名称，进入Vhost详情页。

步骤7 在“Queue”页签的待解绑的Queue所在行，单击“查看详情”，弹出“查看详情”页面。

步骤8 在待解绑的Exchange后，单击“解除绑定”，弹出“解除绑定”对话框。

图 4-11 Queue 详情



须知

解除绑定后，该路由将不再提供服务，且无法恢复，请谨慎操作。

步骤9 单击“是”，完成Exchange的解绑。

----结束

4.7.2 删除 RabbitMQ Exchange

本章节指导如何在控制台删除Exchange。删除Exchange后，该Exchange下的所有配置（包括Exchange和Exchange的绑定关系、Exchange和Queue的绑定关系）也会被删除，且无法恢复。


RabbitMQ 3.x.x版本中，默认Exchange不支持删除。

前提条件

已创建Exchange。


删除 RabbitMQ Exchange

步骤1 登录管理控制台。

步骤2 在管理控制台左上角单击 ，选择区域。

说明

此处请选择RabbitMQ实例所在的区域。

步骤3 在管理控制台左上角单击 ，选择“应用中间件 > 分布式消息服务RabbitMQ版”，进入分布式消息服务RabbitMQ专享版页面。

步骤4 单击实例名称，进入实例详情页面。

步骤5 在左侧导航栏选择“Vhost列表”，进入Vhost列表页面。

步骤6 单击Vhost名称，进入Vhost详情页。

步骤7 在“Exchange”页签中，通过以下任意一种方法，删除Exchange。

- 勾选Exchange名称左侧的方框，可选一个或多个，单击信息栏左上侧的“删除Exchange”。
- 在待删除的Exchange所在行，单击“删除”。

步骤8 单击“确定”，完成Exchange的删除。

----结束

4.8 管理 RabbitMQ Queue

4.8.1 查看 RabbitMQ Queue 信息


Queue创建成功后，可以在控制台查看Queue基本信息、绑定关系和消费者信息。

前提条件

已创建Queue。


查看 RabbitMQ Queue 信息

步骤1 登录管理控制台。

步骤2 在管理控制台左上角单击 ，选择区域。

说明

此处请选择RabbitMQ实例所在的区域。

步骤3 在管理控制台左上角单击 ，选择“应用中间件 > 分布式消息服务RabbitMQ版”，进入分布式消息服务RabbitMQ专享版页面。

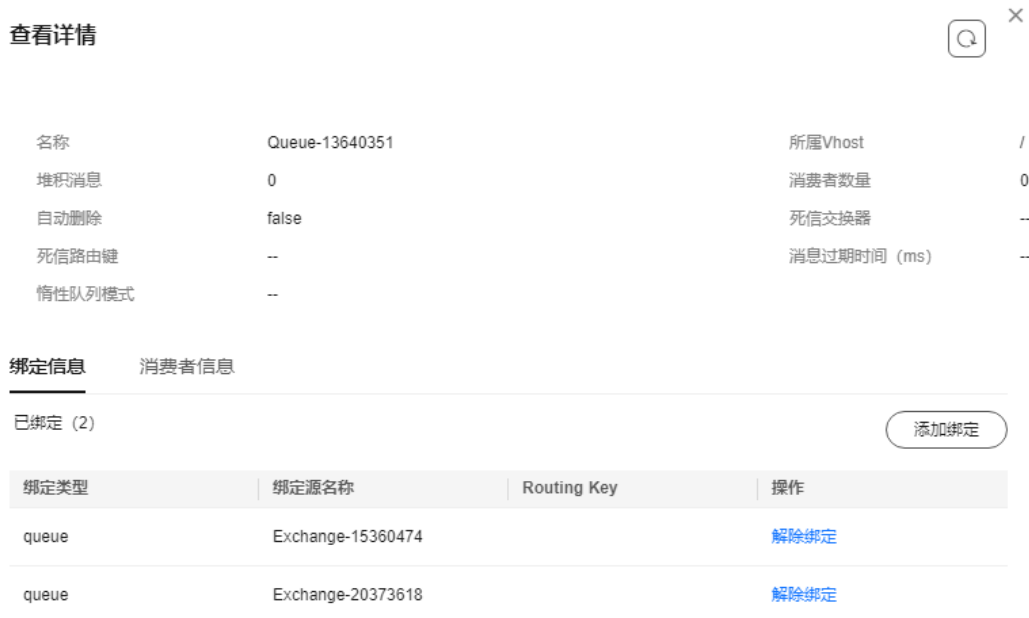
步骤4 单击实例名称，进入实例详情页面。

步骤5 在左侧导航栏选择“Vhost列表”，进入Vhost列表页面。

步骤6 单击Vhost名称，进入Vhost详情页。

步骤7 在“Queue”页签的待查看信息的Queue所在行，单击“查看详情”，弹出“查看详情”页面，查看Queue的基本信息、绑定信息和消费者信息。

图 4-12 Queue 详情



----结束

4.8.2 清空 RabbitMQ Queue 消息

本章节指导如何清空Queue中的所有消息。清空消息的操作有以下几种方式，您可以根据实际情况选择任意一种方式：

- [清空Queue消息（控制台）](#)
- [清空Queue消息（RabbitMQ WebUI）](#)

RabbitMQ 3.x.x版本支持以上两种清空Queue中所有消息的方式，RabbitMQ AMQP-0-9-1版本只支持在控制台清空Queue中所有消息。

须知


清空Queue消息后，Queue中的所有消息会被永久删除，且不可恢复，请谨慎操作。

前提条件

已[创建Queue](#)。


清空 Queue 消息（控制台）

步骤1 登录管理控制台。

步骤2 在管理控制台左上角单击 ，选择区域。

说明

此处请选择RabbitMQ实例所在的区域。

步骤3 在管理控制台左上角单击 ，选择“应用中间件 > 分布式消息服务RabbitMQ版”，进入分布式消息服务RabbitMQ专享版页面。

步骤4 单击实例名称，进入实例详情页面。

步骤5 在左侧导航栏选择“Vhost列表”，进入Vhost列表页面。

步骤6 单击Vhost名称，进入Vhost详情页。

步骤7 在“Queue”页签中，在待清空消息的Queue所在行，单击“清空消息”，弹出“清空消息”对话框。

步骤8 单击“确定”，完成Queue的清空消息。

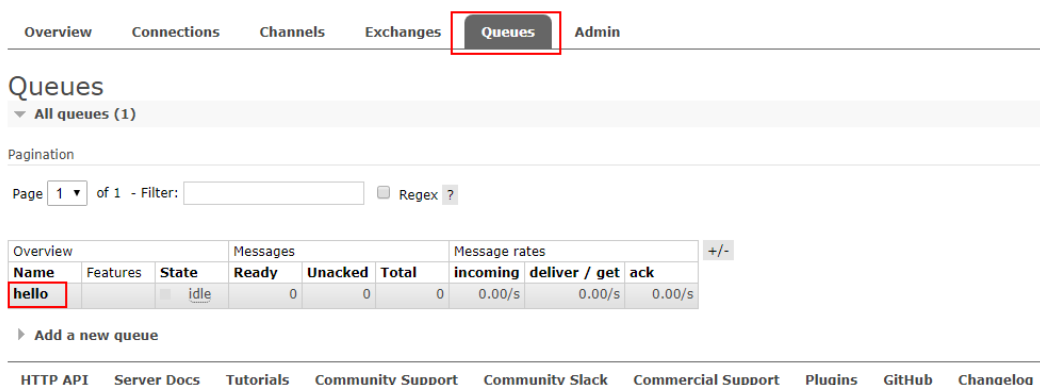
----结束

清空 Queue 消息（RabbitMQ WebUI）

步骤1 登录[RabbitMQ WebUI](#)。

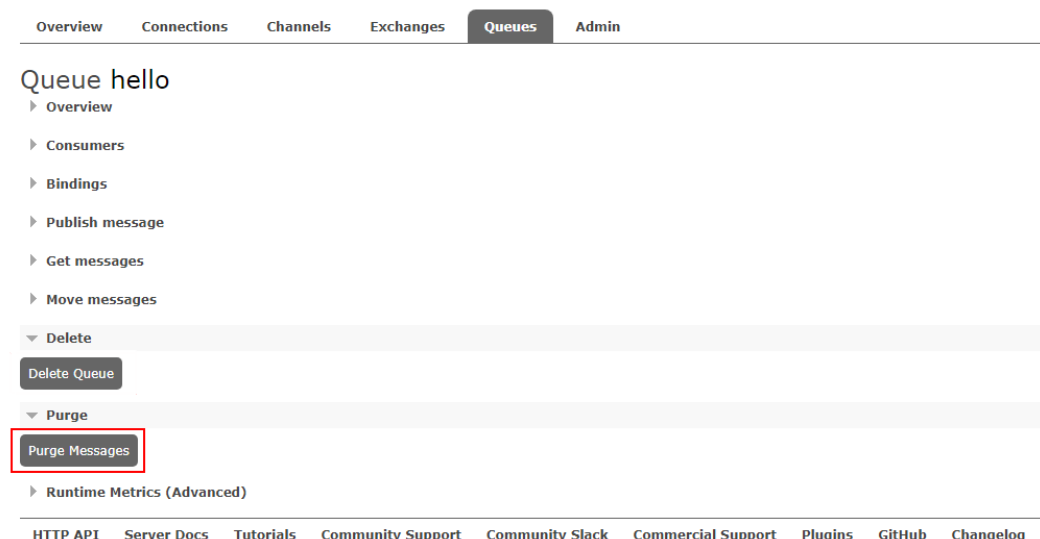
步骤2 在“Queues”页签，单击需要清空数据的队列名称，进入队列详情页面。

图 4-13 Queues



步骤3 单击“Purge Messages”，清空队列数据。

图 4-14 清空队列数据



----结束

4.8.3 解绑 RabbitMQ Queue


本章节指导如何在控制台为Exchange解绑Queue。解绑Queue后，Exchange将无法把消息路由并存放放到该Queue。

前提条件

- 已创建Exchange。
- 已创建Queue。
- Exchange已绑定Queue。


为 Exchange 解绑 Queue

步骤1 登录管理控制台。

步骤2 在管理控制台左上角单击 ，选择区域。

说明

此处请选择RabbitMQ实例所在的区域。

步骤3 在管理控制台左上角单击 ，选择“应用中间件 > 分布式消息服务RabbitMQ版”，进入分布式消息服务RabbitMQ专享版页面。

步骤4 单击实例名称，进入实例详情页面。

步骤5 在左侧导航栏选择“Vhost列表”，进入Vhost列表页面。

步骤6 单击Vhost名称，进入Vhost详情页。

步骤7 在“Exchange”页签中，在Exchange后，单击“绑定”，弹出“绑定”页面。

步骤8 在待解绑的Queue后，单击“解除绑定”，弹出“解除绑定”对话框。

须知

解除绑定后，该路由将不再提供服务，且无法恢复，请谨慎操作。

步骤9 单击“是”，完成Queue的解绑。

----结束

4.8.4 配置 RabbitMQ 镜像队列

镜像队列是指允许集群将队列镜像到其他节点上，当集群某一节点宕机后，队列能自动切换到镜像中的其他节点，保证服务的可用性。

本章节指导如何在RabbitMQ WebUI为Vhost配置镜像队列策略，满足策略条件的队列将成为镜像队列。

说明

RabbitMQ AMQP-0-9-1版本不支持镜像队列。

前提条件

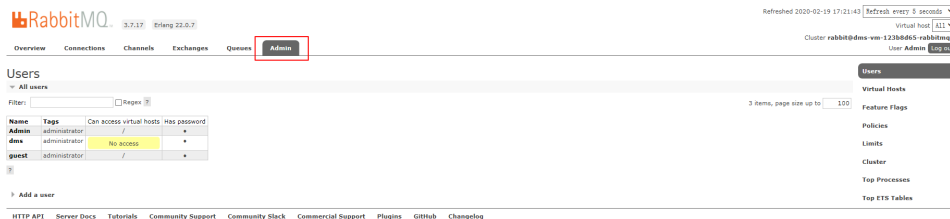
已创建集群版RabbitMQ实例。

配置 RabbitMQ 镜像队列

步骤1 登录RabbitMQ实例的Web UI。

步骤2 在菜单栏，选择“Admin”。

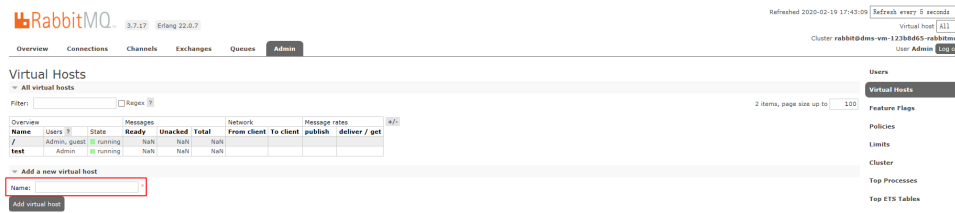
图 4-15 选择 Admin 菜单



步骤3 (可选) 如果您需要设置指定Vhost, 请执行本步骤; 如果不需要, 请直接执行**步骤4**。

选择右侧导航栏“Virtual Hosts”, 然后输入“Name”, 单击“Add virtual host”, 创建Vhost。

图 4-16 创建 Vhost



步骤4 选择右侧导航栏“Policies”, 为Vhost设置策略。

图 4-17 设置 Vhost 策略

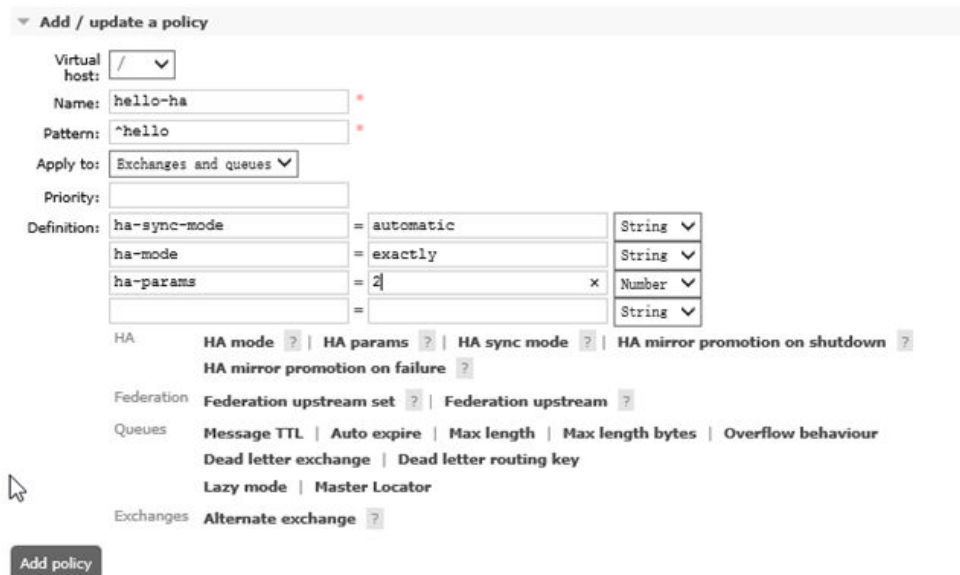


表 4-6 策略参数说明

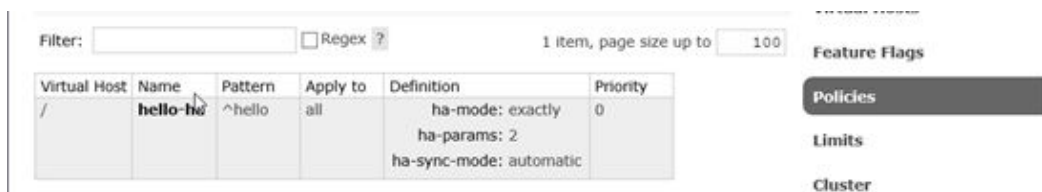
参数	说明
Virtual Host	设置策略所应用的Vhost。如果为指定的Vhost设置, 请在“Virtual Host”选择 步骤3 创建的Vhost; 如果没有, 则默认为“/”。
Name	策略的名称, 用户自定义。
Pattern	Queue的匹配模式(正则表达式)。
Apply to	策略所适用的目标。
Priority	策略的优先级, 数字越大, 优先级越高。

参数	说明
Definition	<p>镜像定义，包括三个部分ha-sync-mode、ha-mode、ha-params。</p> <ul style="list-style-type: none"> ha-sync-mode: 表示镜像队列中消息的同步方式，有效取值范围为：automatic和manual。 <ul style="list-style-type: none"> automatic: 表示自动向master同步数据。 manual: 表示手动向master同步数据。 ha-mode: 指明镜像队列的模式，有效取值范围为：all、exactly和nodes。 <ul style="list-style-type: none"> all: 表示在集群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。

步骤5 单击“Add policy”。

策略添加成功后如下图所示。

图 4-18 Vhost 策略



----结束

4.8.5 配置 RabbitMQ 惰性队列

默认情况下，RabbitMQ生产者生产的消息存储在内存中，当需要释放内存时，会将内存中的消息换页至磁盘中。换页操作会消耗较长的时间，且换页过程中队列无法处理消息。

如果生产速度过快（例如执行批处理任务），或者消费者由于各种原因（例如消费者下线、宕机）长时间内无法消费消息，导致消息大量堆积，使得内存使用率过高，换页频繁，可能会影响其他队列的消息收发。这种场景下，建议您启用惰性队列。

惰性队列（Lazy Queue）会尽可能的将消息存入磁盘中，在消费者消费到相应的消息时才会被加载到内存中，这样可以减少内存的消耗，但是会增加I/O的使用，影响单个队列的吞吐量。惰性队列的一个重要的设计目标是能够支持更长的队列，即支持更多的消息存储/消息堆积。

在以下情况下，推荐使用惰性队列：

- 队列可能会产生消息堆积。
- 队列对性能（吞吐量）的要求不是非常高，例如TPS 1万以下的场景。

- 希望队列有稳定的生产消费性能，不受内存影响而波动。

处于以下情况时，无需使用惰性队列：

- RabbitMQ需要高性能的场景。
- 队列总是很短（即队列中没有消息堆积）。
- 设置了最大长度策略。

更多关于惰性队列的说明，请参考[Lazy Queues](#)。

📖 说明

仅RabbitMQ 3.8.35版本支持惰性队列。

配置惰性队列

队列具备两种模式：default和lazy，默认模式为default。lazy模式即为惰性队列的模式，可以通过调用channel.queueDeclare方法的时候在参数中设置，也可以通过Policy的方式设置。如果一个队列同时使用这两种方式设置的话，Policy的方式具备更高的优先级。

- 以下示例演示在Java客户端通过调用channel.queueDeclare设置惰性队列。

```
Map<String, Object> args = new HashMap<String, Object>();
args.put("x-queue-mode", "lazy");
channel.queueDeclare("myqueue", false, false, false, args);
```
- 以下示例演示在RabbitMQ WebUI页面通过Policy的方式设置惰性队列。

图 4-19 通过 Policy 的方式设置惰性队列



4.8.6 配置 RabbitMQ 仲裁队列

仲裁队列（Quorum Queues）提供队列复制的能力，保障数据的高可用和安全性。使用仲裁队列可以在RabbitMQ节点间进行队列数据的复制，在一个节点宕机时，队列依旧可以正常运行。

仲裁队列适用于队列长时间存在，对队列容错和数据安全要求高，对延迟和队列特性要求相对低的场景。在可能出现消息大量堆积的场景，不推荐使用仲裁队列，因为仲裁队列的写入会造成成倍的磁盘占用。

仲裁队列的消息会优先保存在内存中，使用仲裁队列时，建议定义队列最大长度和最大内存占用，在消息堆积超过阈值时从内存转移到磁盘，以免造成内存高水位。

更多关于仲裁队列的说明，请参考[Quorum Queues](#)。

说明

仅RabbitMQ 3.8.35版本支持仲裁队列。

仲裁队列与镜像队列的差异

仲裁队列是RabbitMQ 3.8版本引入的队列类型，它与镜像队列拥有类似的功能，为RabbitMQ提供高可用的队列。镜像队列有一些设计上的缺陷，这也是RabbitMQ提供仲裁队列的原因。

镜像队列主要的缺陷在于消息同步的性能低。

- 镜像队列包含一个主队列和多个从队列，当生产者向主队列发送一条消息，主队列会将消息同步给从队列，所有的从队列都保存消息后，主队列才会向生产者发送确认。
- RabbitMQ使用集群部署时，如果其中一个节点故障下线，待它消除故障重新上线后，它保存的所有从队列的数据都会丢失。此时运维人员需要选择是否同步主队列的数据到从队列中，如果不同步数据，会增加消息丢失的风险。如果同步数据，同步时队列是阻塞的，无法对其进行操作。当队列中存在大量堆积消息时，同步会导致队列几分钟、几小时或者更长时间不可用。

仲裁队列解决了镜像队列的性能和同步问题。

- 仲裁队列的算法是基于Raft共识算法的一个变种，提供更好的消息吞吐量。仲裁队列包含一个主副本和多个从副本，当生产者向主副本发送一条消息，主副本会将消息同步给从副本，超过半数的副本保存消息后，主副本才会向生产者发送确认。这意味着少部分比较慢的从副本不会影响整个队列的性能。同样地，主副本的选举也需要超过半数的副本同意，这会避免出现网络分区时，队列存在2个主副本。由此可见，仲裁队列相对于可用性更看重一致性。
- RabbitMQ使用集群部署时，如果其中一个节点故障下线，待它消除故障重新上线后，它保存的数据不会丢失，主副本会直接从从副本中断的地方开始复制消息。复制的过程是非阻塞的，整个队列不会因为新的副本加入而受到影响。

仲裁队列相比镜像队列，缺少了一些特性，如表4-7所示，且消耗更多的内存和磁盘。

表 4-7 仲裁队列与镜像队列特性差异

特性	镜像队列	仲裁队列
非持久化队列	支持	不支持
排他队列	支持	不支持
每条消息的持久化	每条消息	永远
队列重平衡	自动	手动
消息超时时间	支持	不支持
队列超时时间	支持	支持
队列长度限制	支持	支持（除x-overflow: reject-publish-dlx）
惰性队列	支持	限制队列长度后支持

特性	镜像队列	仲裁队列
消息优先级	支持	不支持
消费优先级	支持	支持
死信交换器	支持	支持
动态Policy	支持	支持
毒药消息（让消费者无限循环消费）处理	不支持	支持
全局消息预取（Qos）	支持	不支持

配置仲裁队列

在声明队列时，将队列的“x-queue-type”参数设置为“quorum”。此参数只能在声明队列时设置，不能通过Policy设置。

仲裁队列默认的复制因子是5。

- 以下示例演示在Java客户端设置仲裁队列。

```
ConnectionFactory factory = newConnectionFactory();
factory.setRequestedHeartbeat(30);
factory.setHost(HOST);
factory.setPort(PORT);
factory.setUsername(USERNAME);
factory.setPassword(PASSWORD);

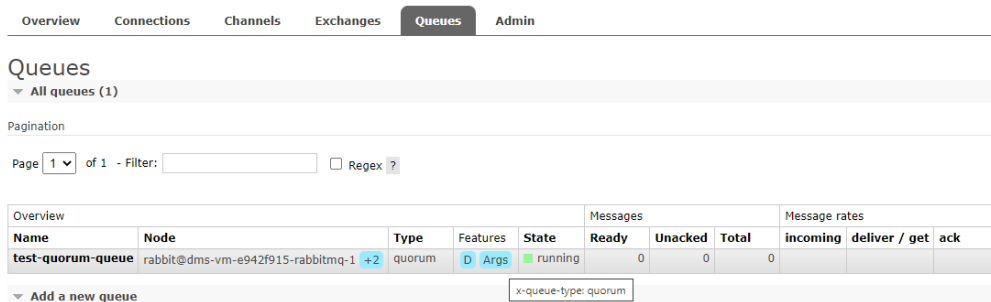
finalConnection connection = factory.newConnection();
finalChannel channel = connection.createChannel();
// 创建队列参数Map
Map<String, Object> arguments = newHashMap<>();
arguments.put("x-queue-type", "quorum");
// 声明仲裁队列
channel.queueDeclare("test-quorum-queue", true, false, false, arguments);
```

- 以下示例演示在RabbitMQ WebUI页面设置仲裁队列。

图 4-20 设置仲裁队列

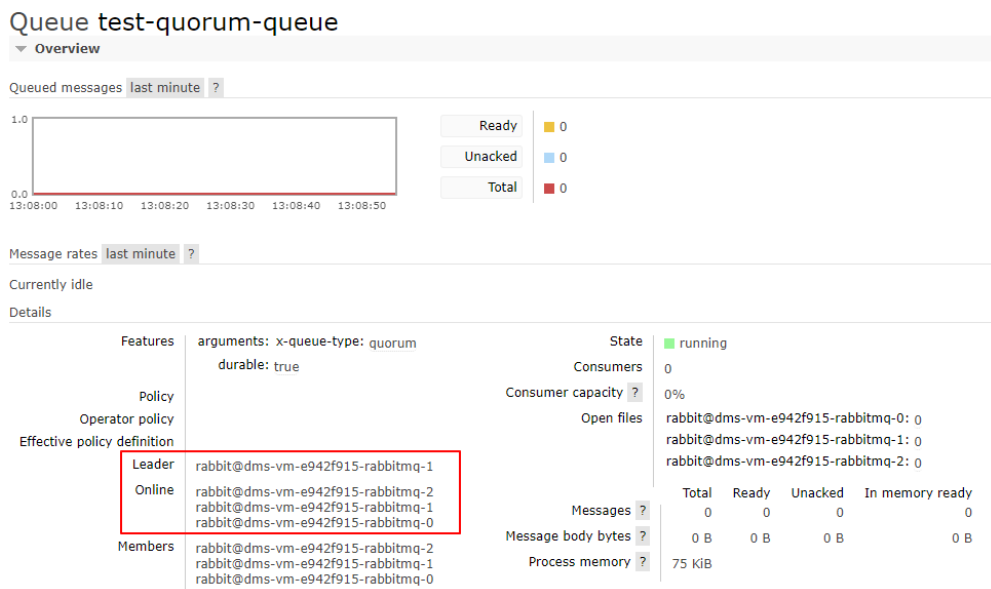
设置完成后，在“Queues”页面查看队列类型是否为“quorum”，如图4-21所示。“Node”中的“+2”表示该队列有2个副本，蓝色表示这两个副本消息同步已经完成，如果为红色则表示部分消息还未同步。

图 4-21 查看队列类型



在“Queues”页面，单击队列名称，进入队列详情页。查看当前仲裁队列主副本所在节点和在线副本所在节点。

图 4-22 队列详情页



配置仲裁队列的长度

通过配置Policy或者队列属性的方式可以限制仲裁队列的长度和在内存中保存的长度。

- x-max-length: 仲裁队列最大消息数。如果超过则丢弃消息，或者发送到死信交换器。
- x-max-length-bytes: 仲裁队列最大总消息大小（字节数）。如果超过则丢弃消息，或者发送到死信交换器。
- x-max-in-memory-length: 限制仲裁队列的内存中最大消息数量。
- x-max-in-memory-bytes: 限制仲裁队列的内存中的最大总消息大小（字节数）。

以下举例说明通过配置Policy或者队列属性的方式限制内存中保存的仲裁队列长度。

- 通过Policy方式配置，推荐使用此方式。
在Policy中通过设置x-max-in-memory-bytes参数，限制仲裁队列的长度。

图 4-23 使用 Policy 设置 x-max-in-memory-bytes

Policies

▼ User policies

Filter: Regex ?

... no policies ...

▼ Add / update a policy

Name: *

Pattern: *

Apply to:

Priority:

Definition: = *

=

Queues [All types] [Max length](#) | [Max length bytes](#) | [Overflow behaviour](#) ? | [Auto expire](#)
[Dead letter exchange](#) | [Dead letter routing key](#)

Queues [Classic] [HA mode](#) ? | [HA params](#) ? | [HA sync mode](#) ?
[HA mirror promotion on shutdown](#) ? | [HA mirror promotion on failure](#) ?
[Message TTL](#) | [Lazy mode](#) | [Master Locator](#)

Queues [Quorum] [Max in memory length](#) ? | **Max in memory bytes** ? | [Delivery limit](#) ?

Exchanges [Alternate exchange](#) ?

Federation [Federation upstream set](#) ? | [Federation upstream](#) ?

- 通过队列属性方式配置。
新增队列时，在队列属性设置x-max-in-memory-length参数，限制仲裁队列的长度。

图 4-24 使用队列属性设置 x-max-in-memory-length

▼ Add a new queue

Type:

Name: *

Node:

Arguments: =

=

Add [Auto expire](#) ? | [Max length](#) ? | [Max length bytes](#) ? | [Delivery limit](#) ?
[Overflow behaviour](#) ?
[Dead letter exchange](#) ? | [Dead letter routing key](#) ? | [Single active consumer](#) ? | **Max in memory length** ?
[Max in memory bytes](#) ?

4.8.7 配置 RabbitMQ 单一活跃消费者

单一活跃消费者（Single Active Consumer）表示队列中可以注册多个消费者，但是只允许一个消费者消费消息，只有在此消费者出现异常时，才会自动转移到另一个消费者进行消费。单一活跃消费者适用于需要保证消息消费顺序性，同时提供高可靠能力的场景。

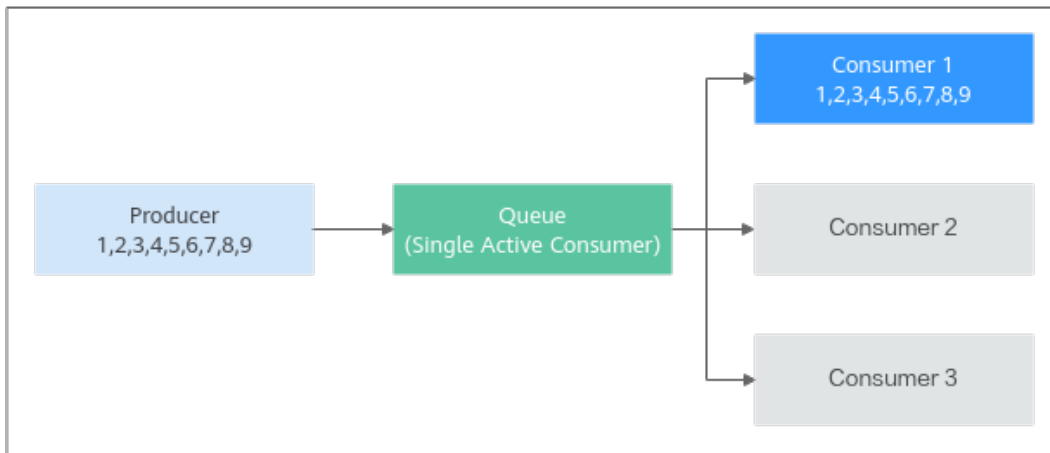
说明

仅RabbitMQ 3.8.35版本支持单一活跃消费者特性。

如图4-25所示，Producer生产9条消息，由于队列设置了单一活跃消费者特性，只有Consumer 1在消费消息。

更多关于单一活跃消费者的说明，请参考[Single Active Consumer](#)。

图 4-25 单一活跃消费者消费流程



配置单一活跃消费者

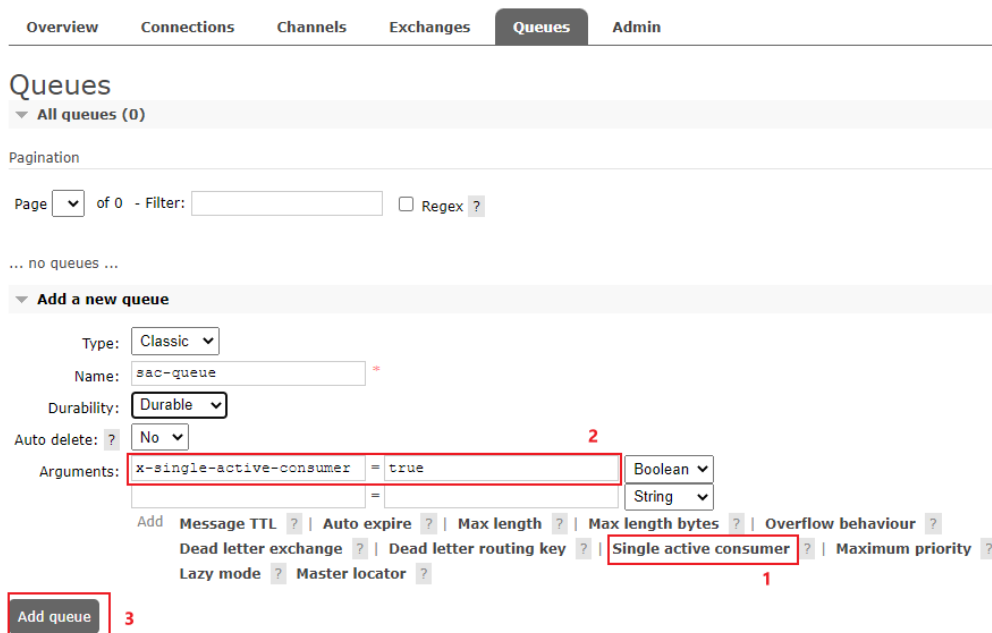
在声明队列时，可以配置单一活跃消费者，只需要将队列的“x-single-active-consumer”参数设置为“true”。

- 以下示例演示在Java客户端设置单一活跃消费者。

```
Channel ch = ...;
Map<String, Object> arguments = new HashMap<String, Object>();
arguments.put("x-single-active-consumer", true);
ch.queueDeclare("my-queue", false, false, false, arguments);
```

- 以下示例演示在[RabbitMQ WebUI](#)页面设置单一活跃消费者。

图 4-26 设置单一活跃消费者



设置完成后，在“Queues”页面查看队列特性是否包含单一活跃消费者。如图 4-27 所示，“SAC”即表示队列已设置单一活跃消费者。

图 4-27 查看队列特性

Queues

▼ All queues (1)

Pagination

Page 1 of 1 - Filter: Regex ?

Overview				Messages			Message rates			+/-
Name	Type	Features	State	Ready	Unacked	Total	incoming	deliver / get	ack	
sac-queue	classic	D SAC Args	idle	0	0	0				

▼ Add a new queue

4.8.8 删除 RabbitMQ Queue

本章节指导如何删除 Queue。删除 Queue 后，该 Queue 下的所有配置（包括 Exchange 与 Queue 的绑定关系）也会被删除，且无法恢复。

删除 Queue 的操作有以下几种方式，您可以根据实际情况选择任意一种方式：

- 删除 Queue（控制台）
- 删除单个 Queue（RabbitMQ WebUI）
- 批量删除 Queue（RabbitMQ WebUI）

📖 说明


RabbitMQ 3.x.x 版本支持以上三种删除 Queue 的方式，RabbitMQ AMQP-0-9-1 版本只支持在控制台删除 Queue。

前提条件

已创建 Queue。


删除 Queue（控制台）

步骤1 登录管理控制台。

步骤2 在管理控制台左上角单击 ，选择区域。

📖 说明

此处请选择 RabbitMQ 实例所在的区域。

步骤3 在管理控制台左上角单击 ，选择“应用中间件 > 分布式消息服务 RabbitMQ 版”，进入分布式消息服务 RabbitMQ 专享版页面。

步骤4 单击实例名称，进入实例详情页面。

步骤5 在左侧导航栏选择“Vhost 列表”，进入 Vhost 列表页面。

步骤6 单击 Vhost 名称，进入 Vhost 详情页。

步骤7 在“Queue”页签中，通过以下任意一种方法，删除Queue。

- 勾选Queue名称左侧的方框，可选一个或多个，单击信息栏左上侧的“删除Queue”。
- 在待删除的Queue所在行，单击“删除”。

步骤8 单击“确定”，完成Queue的删除。

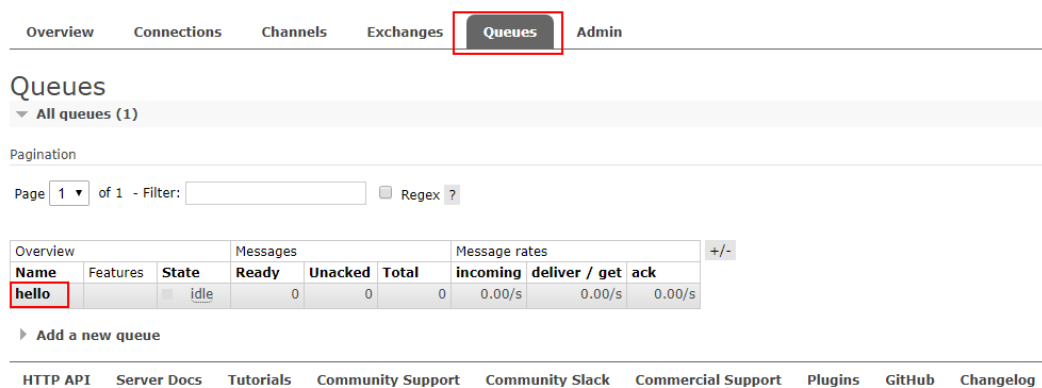
----结束

删除单个 Queue (RabbitMQ WebUI)

步骤1 [登录RabbitMQ WebUI页面](#)。

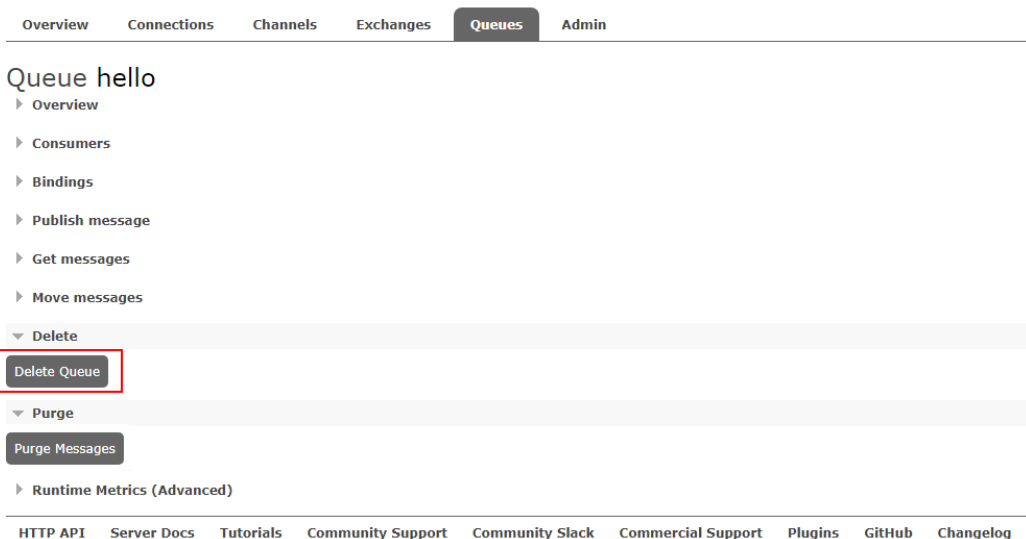
步骤2 在“Queues”页签，单击需要删除的队列名称，进入队列详情页面。

图 4-28 队列列表



步骤3 单击“Delete Queue”，删除单个队列。

图 4-29 删除单个队列



----结束

批量删除 Queue (RabbitMQ WebUI)

新增与待删除队列的前缀名称相同、且队列TTL为1毫秒的策略，通过此策略实现批量删除队列。

步骤1 [登录RabbitMQ WebUI页面](#)。

步骤2 在“Admin > Policies”页面中，新增一条策略。

图 4-30 通过 Policy 批量删除队列

The screenshot shows the 'Policies' page in the RabbitMQ WebUI. The 'Admin' tab is selected. Under 'User policies', there is a section for 'Add / update a policy'. The form contains the following fields and values:

- Name: Delete queues
- Pattern: .*
- Apply to: Queues
- Priority: (empty)
- Definition: expires = 1

Below the form, there are several expandable sections for additional configuration:

- HA: HA mode, HA params, HA sync mode, HA mirror promotion on shutdown
- Federation: Federation upstream set, Federation upstream
- Queues: Message TTL, Auto expire, Max length, Max length bytes, Overflow behaviour, Dead letter exchange, Dead letter routing key, Lazy mode, Master Locator
- Exchanges: Alternate exchange

An 'Add policy' button is located at the bottom left of the form area.

表 4-8 策略参数说明

参数	说明
Name	策略的名称，用户自定义。
Pattern	队列匹配模式，填写队列名称，会匹配包含此队列名称的队列。例如：设置为“.”时，表示匹配所有队列。设置为“.*queue-name”时，表示匹配队列名称包含queue-name的所有队列。
Apply to	策略所适用的目标，选择“Queues”。
Priority	策略的优先级，数字越大，优先级越高。
Definition	定义TTL，单位为毫秒。填写“expires”参数，值设置为“1”，表示队列过期时间为1毫秒。

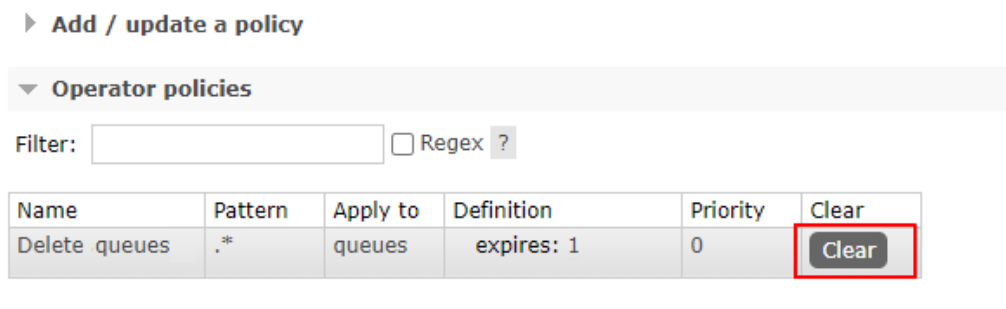
步骤3 单击“Add policy”。

在“Queues”页签，查看队列是否成功删除。

步骤4 队列成功删除后，在“Admin > Policies”页面中，在**步骤2**中新增的策略后，单击“Clear”，删除策略。

如果保留此策略，它对后续新建的队列依然生效，可能会出现误删除队列的情况。

图 4-31 删除策略



----结束

5 连接实例

5.1 配置 RabbitMQ 网络连接

5.1.1 连接 RabbitMQ 网络要求

客户端可以通过公网连接RabbitMQ实例，也可以通过内网连接RabbitMQ实例。使用内网连接时，注意以下几点：

- 如果客户端和RabbitMQ实例部署在同一个VPC内，网络默认互通。
- 如果客户端和RabbitMQ实例部署在不同VPC中，由于VPC之间逻辑隔离，客户端和RabbitMQ实例不能直接通信，需要打通VPC之间的网络。

表 5-1 连接方式说明

连接方式	实现方式	参考文档
公网连接	在RabbitMQ控制台开启公网访问，配置弹性公网IP，客户端通过弹性公网IP访问RabbitMQ实例。	配置RabbitMQ实例的公网访问
内网连接	客户端和RabbitMQ实例部署在同一个VPC中，此时网络默认互通。	-
	客户端和RabbitMQ实例部署在同一个Region的不同VPC中，利用VPC对等连接将两个VPC间的网络打通，实现跨VPC访问。	对等连接

客户端连接RabbitMQ实例前，需要放通如下安全组，否则会连接失败。

说明

- 创建安全组后，系统默认添加入方向“允许安全组内的弹性云服务器彼此通信”规则和出方向“放通全部流量”规则，此时使用内网通过同一个VPC访问RabbitMQ实例，无需添加表5-2或表5-3的规则。
- 表5-2和表5-3中的源地址以全网段放通为例，可根据实际安全需要修改为客户端的IP地址。

表 5-2 安全组规则（RabbitMQ 3.x.x 版本）

方向	类型	协议	端口	源地址	说明
入方向	IPv4	TCP	5672	0.0.0.0/0	客户端使用IPv4地址访问RabbitMQ实例（关闭SSL加密）
入方向	IPv4	TCP	5671	0.0.0.0/0	客户端使用IPv4地址访问RabbitMQ实例（开启SSL加密）
入方向	IPv4	TCP	15672	0.0.0.0/0	访问Web界面UI地址（关闭SSL加密）
入方向	IPv4	TCP	15671	0.0.0.0/0	访问Web界面UI地址（开启SSL加密）

表 5-3 安全组规则（RabbitMQ AMQP-0-9-1 版本）

方向	类型	协议	端口	源地址	说明
入方向	IPv4	TCP	5672	0.0.0.0/0	访问RabbitMQ实例

5.1.2 配置 RabbitMQ 实例的公网访问

当您需要通过公网地址访问RabbitMQ实例时，可开启实例的公网访问功能，并设置弹性IP地址。当业务不再使用公网访问功能时，也可以关闭实例的公网访问功能。

须知


公网访问与VPC内访问相比，可能存在网络丢包和抖动等情况，且访问时延有所增加，因此建议仅在业务开发测试阶段开启公网访问RabbitMQ实例。

前提条件

仅状态为“运行中”的实例，可以开启公网访问功能。


开启公网访问（RabbitMQ 3.x.x 版本）

步骤1 登录管理控制台。

步骤2 在管理控制台左上角单击 ，选择区域。


说明


此处请选择RabbitMQ实例所在的区域。

步骤3 在管理控制台左上角单击 ，选择“应用中间件 > 分布式消息服务RabbitMQ版”，进入分布式消息服务RabbitMQ专享版页面。

步骤4 单击待开启公网访问的实例名称，进入实例详情页面。

步骤5 单击“公网访问”右侧的 ，打开公网访问开关。

步骤6 从“弹性IP地址”下拉列表中选择一个弹性IP，然后单击 ，开启公网访问功能。

如果“弹性IP地址”下拉列表没有值，可单击“创建弹性IP”，跳转到弹性公网IP页面，您可以申请一个新的弹性IP。弹性IP申请完后，返回RabbitMQ控制台，单击“弹性IP地址”后的 ，然后在下拉列表中选择新申请的弹性IP。

开启公网访问功能大约需要10~30秒，请耐心等待。开启公网访问后，页面会自动跳转到“后台任务管理”页签，当任务状态为“成功”时，表示开启公网访问成功。

说明

开启公网访问后，有如下注意事项：

- 如果实例未开启SSL，修改实例的安全组策略，增加入方向规则，允许端口5672和15672的访问。

访问RabbitMQ管理面：输入地址http://{RabbitMQ实例公网IP地址}:15672，然后输入自己配置的用户名和密码。

Client方式：请使用5672端口。

- 如果实例开启SSL，修改实例的安全组策略，增加入方向规则，允许端口5671和15671的访问。


访问RabbitMQ管理面：输入地址https://{RabbitMQ实例公网IP地址}:15671，然后输入自己配置的用户名和密码。

Client方式：请使用5671端口。

----结束


关闭公网访问（RabbitMQ 3.x.x 版本）

步骤1 登录管理控制台。

步骤2 在管理控制台左上角单击 ，选择区域。


说明

此处请选择RabbitMQ实例所在的区域。

步骤3 在管理控制台左上角单击 ，选择“应用中间件 > 分布式消息服务RabbitMQ版”，进入分布式消息服务RabbitMQ专享版页面。

步骤4 单击待关闭公网访问的实例名称，进入实例详情页面。

步骤5 单击“公网访问”右侧的 ，关闭公网访问开关。

步骤6 单击 ，关闭公网访问功能。

关闭公网访问功能大约需要10~30秒，请耐心等待。关闭公网访问后，页面会自动跳转到“后台任务管理”页签，当任务状态为“成功”时，表示关闭公网访问成功。

----结束

5.2 配置 RabbitMQ 访问控制

5.2.1 开启 RabbitMQ ACL 访问

通过ACL可以实现对RabbitMQ实例中生产者和消费者身份的精确识别，为不同用户赋予不同的Vhost和Vhost下不同资源的权限，以达到不同用户之间权限隔离的目的。开启ACL权限控制功能后，生产消息和消费消息时，需要鉴权。

说明


仅RabbitMQ AMQP-0-9-1版本支持在控制台开启ACL访问控制功能。RabbitMQ 3.x.x版本默认支持用户权限管理，您可以在RabbitMQ WebUI页面创建用户，并赋予相应的权限。

前提条件

已购买RabbitMQ AMQP-0-9-1版本的实例。


开启 ACL 访问控制

步骤1 登录管理控制台。


步骤2 在管理控制台左上角单击 ，选择区域。

说明

此处请选择RabbitMQ实例所在的区域。

步骤3 在管理控制台左上角单击 ，选择“应用中间件 > 分布式消息服务RabbitMQ版”，进入分布式消息服务RabbitMQ专享版页面。

步骤4 单击实例名称，进入实例详情页面。

步骤5 在“连接信息”区域，单击“ACL访问控制”后的 ，弹出“确认”对话框。

步骤6 单击“确定”，开启ACL访问控制。

须知

开启ACL访问控制会导致没有配置认证信息的客户端连接中断。

----结束

5.2.2 配置 RabbitMQ ACL 用户

RabbitMQ实例开启ACL访问控制后，生产和消费消息时需要进行ACL用户鉴权，鉴权成功后才能生产和消费消息。

本章节介绍如何创建、编辑和删除用户。

说明


仅RabbitMQ AMQP-0-9-1版本支持在控制台配置RabbitMQ ACL用户。RabbitMQ 3.x.x版本默认支持用户权限管理，您可以在RabbitMQ WebUI页面创建用户，并赋予相应的权限。

前提条件

- 已购买RabbitMQ AMQP-0-9-1版本的实例。
- 已[开启ACL访问控制](#)。


创建用户

步骤1 登录管理控制台。

步骤2 在管理控制台左上角单击 ，选择区域。

说明

此处请选择RabbitMQ实例所在的区域。

步骤3 在管理控制台左上角单击 ，选择“应用中间件 > 分布式消息服务RabbitMQ版”，进入分布式消息服务RabbitMQ专享版页面。

步骤4 单击实例名称，进入实例详情页面。

步骤5 在左侧导航栏，单击“用户管理”，进入“用户管理”页面。

步骤6 单击“创建用户”，弹出“创建用户”对话框。

步骤7 参考[表5-4](#)，设置用户名称和配置信息。

表 5-4 用户参数说明

参数	说明
用户名	用户名支持自定义，但需要符合命名规则：长度为7~64个字符，由英文字母开头，只能由英文字母、数字、中划线、下划线组成。 用户名创建成功后，不可修改。
密码	设置用户的密码。 密码需要符合以下命名规则： <ul style="list-style-type: none">• 长度为8~32个字符。• 至少包含以下字符中的3种：大写字母、小写字母、数字、特殊字符`~!@#\$\$%^&*()-_+=\ [{ } ; : " , < . > ?` 和空格，并且不能以-开头。• 不能与用户名或倒序的用户名相同。

参数	说明
确认密码	再次输入密码，确认本次输入的密码和“密码”中输入的一致。
Vhost	<ul style="list-style-type: none"> 可访问的Vhost：在下拉框中选择用户可以访问的Vhost。 可配置的资源：为用户授予Vhost下资源的权限，使用正则表达式匹配资源。例如输入“^test-.*”，表示为用户授予Vhost下所有名称以“test-”开头的资源的权限。 可写的资源：为用户授予Vhost下资源的写权限，使用正则表达式匹配资源。例如输入“.*”，表示为用户授予Vhost下所有资源的写权限。 可读的资源：为用户授予Vhost下资源的读权限，使用正则表达式匹配资源。例如输入“.*”，表示为用户授予Vhost下所有资源的读权限。 <p>如果需要授予用户多个Vhost的访问权限，单击“添加”，继续添加其他Vhost信息。</p>

图 5-1 创建用户

创建用户

* 用户名

* 密码

* 确认密码

* Vhost	可访问的Vhost	可配置的资源	可写的资源	可读的资源	操作
	default	<input type="text"/>	<input type="text"/>	<input type="text"/>	

+ 添加

步骤8 单击“确定”，完成用户的创建。

步骤9 开启ACL访问控制后，消息生产者和消费者都需要增加用户认证信息（代码中需包含username和password），具体操作如下：

- [Java代码示例](#)
- [Python代码示例](#)

----结束


编辑用户

步骤1 登录管理控制台。

步骤2 在管理控制台左上角单击 ，选择区域。


📖 说明

此处请选择RabbitMQ实例所在的区域。

- 步骤3** 在管理控制台左上角单击，选择“应用中间件 > 分布式消息服务RabbitMQ版”，进入分布式消息服务RabbitMQ专享版页面。
- 步骤4** 单击实例名称，进入实例详情页面。
- 步骤5** 在左侧导航栏选择“用户管理”，进入用户列表页面。
- 步骤6** 在待修改信息的用户所在行，单击“编辑”，弹出“编辑用户”对话框。
- 步骤7** 如果要编辑密码，在“密码”后，单击“编辑”，输入新密码。如果要编辑Vhost相关信息，参考如下信息，进行编辑。
- 可访问的Vhost：在下拉框中选择用户可以访问的Vhost。
 - 可配置的资源：为用户授予Vhost下资源的权限，使用正则表达式匹配资源。例如输入“^test-.*”，表示为用户授予Vhost下所有名称以“test-”开头的资源的权限。
 - 可写的资源：为用户授予Vhost下资源的写权限，使用正则表达式匹配资源。例如输入“.*”，表示为用户授予Vhost下所有资源的写权限。
 - 可读的资源：为用户授予Vhost下资源的读权限，使用正则表达式匹配资源。例如输入“.*”，表示为用户授予Vhost下所有资源的读权限。
- 步骤8** 编辑完用户信息后，单击“确定”。


----结束

删除用户

- 步骤1** 登录管理控制台。
- 步骤2** 在管理控制台左上角单击，选择区域。

📖 说明

此处请选择RabbitMQ实例所在的区域。

- 步骤3** 在管理控制台左上角单击，选择“应用中间件 > 分布式消息服务RabbitMQ版”，进入分布式消息服务RabbitMQ专享版页面。
- 步骤4** 单击实例名称，进入实例详情页面。
- 步骤5** 在左侧导航栏选择“用户管理”，进入用户列表页面。
- 步骤6** 在待删除的用户所在行，单击“删除”，弹出“删除用户”对话框。
- 步骤7** 单击“确定”，完成用户的删除。

须知

删除用户后会删除授权关系，原来使用该用户认证的请求将无法访问。

----结束

5.3 配置 RabbitMQ 客户端心跳超时时间

客户端连接RabbitMQ集群实例时，如果存在消息收发时间间隔大于90秒的场景，请在客户端开启心跳并设置小于90秒的心跳超时时间，防止客户端断连。

什么是心跳

RabbitMQ实例提供了心跳功能，以确保应用程序层及时发现中断的连接和完全无响应的对端。心跳还可以防止某些网络设备在一段时间内由于没有活动而中断TCP连接。**开启心跳的方法为在连接上指定心跳超时时间。**

心跳超时时间定义了对等TCP连接在多长时间后被服务端和客户端视为关闭。服务端和客户端会对配置的心跳超时时间进行协商，客户端必须配置该值来发送心跳。RabbitMQ官方团队维护的3个客户端（Java、.NET、Erlang语言）的心跳超时时间协商逻辑如下：

- 服务端和客户端设置的心跳超时时间都不为0时，两者间较小的值生效。
- 服务端和客户端任意一端设置的心跳超时时间为0，另一端不为0时，非0的值生效。
- 服务端和客户端的心跳超时时间都设置为0时，表示禁用心跳。

配置心跳超时时间后，RabbitMQ服务端和客户端都会向对方发送AMQP心跳帧作为心跳，发送的时间间隔为心跳超时时间的一半。客户端在两次错过心跳后，会被认为是不可达的，TCP连接将被关闭。当客户端检测到服务端由于心跳而无法访问时，需要重新连接。更多关于心跳的说明，请参考[Detecting Dead TCP Connections with Heartbeats and TCP Keepalives](#)。

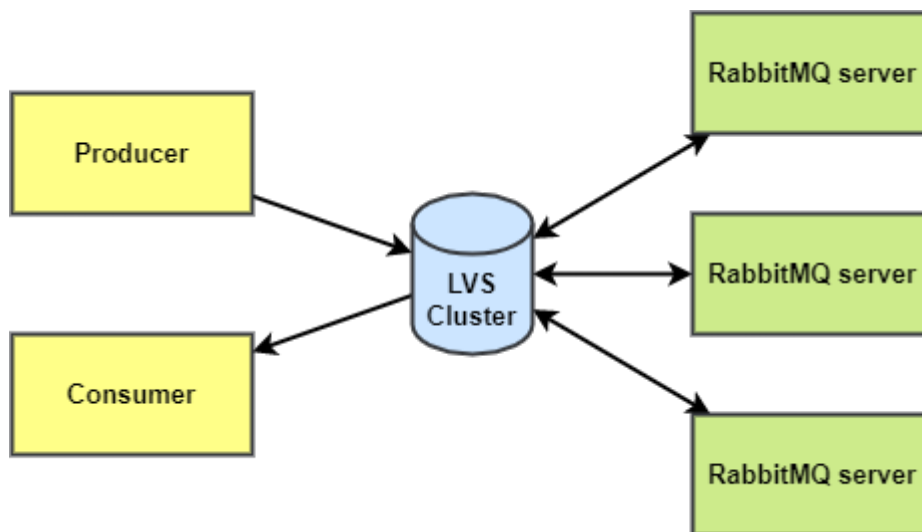
📖 说明

一些客户端（如C语言客户端）没有发送心跳的逻辑，即使配置了心跳超时时间，开启了心跳，仍然无法发送心跳。此时需要额外启动一个线程，编写发送心跳的逻辑。

LVS 的心跳超时时间

RabbitMQ集群实例使用LVS进行负载均衡，如图5-2所示，单节点实例不涉及LVS。

图 5-2 集群实例的负载均衡



LVS对客户端连接设置了心跳超时时间，默认为90秒。如果客户端在90秒内没有向LVS发送心跳（AMQP心跳帧或消息收发），LVS会主动断开与客户端的连接，此时客户端需要重新连接。

如果存在消息收发时间间隔大于90秒的场景，请在客户端开启心跳并设置小于90秒的心跳超时时间。

配置客户端心跳超时时间

- 在Java客户端配置心跳超时时间。

在创建连接前使用**ConnectionFactory#setRequestedHeartbeat**进行设置，示例如下。

```
ConnectionFactory cf = new ConnectionFactory();  
// 将心跳超时时间设置为60秒  
cf.setRequestedHeartbeat(60);
```

- 在.NET客户端配置心跳超时时间，示例如下。

```
var cf = new ConnectionFactory();  
// 将心跳超时设置为60秒  
cf.RequestedHeartbeat = TimeSpan.FromSeconds(60);
```

- 在Python pika客户端配置心跳超时时间，示例如下。

```
# 设置心跳时间为60秒  
params = pika.ConnectionParameters(host='host', heartbeat=60,  
credentials=pika.PlainCredentials('username', 'passwd'))  
connection = pika.BlockingConnection(params)  
  
while True:  
    channel.basic_publish(exchange="", routing_key='hello', body='Hello World!')  
    print("[x] Sent 'Hello World!'")  
    # 生产者需要使用connection.sleep()才能触发心跳，使用时间.sleep()不会触发心跳  
    connection.sleep(200)
```

- 在PHP客户端配置心跳超时时间，示例如下。

```
# 设置心跳时间为60秒  
$connection = new AMQPStreamConnection(RMQ_HOST, RMQ_PORT, RMQ_USER, RMQ_PASS,  
RMQ_vhost, ['heartbeat'=> 60]);
```

5.4 使用客户端连接 RabbitMQ（关闭 SSL）

本章节以分布式消息服务RabbitMQ版提供的demo为例，介绍RabbitMQ客户端如何连接未开启SSL的RabbitMQ实例，并生产和消费消息。

前提条件

- 已[购买RabbitMQ实例](#)，并记录创建时输入的用户名和密码，实例未开启SSL。
- 在实例详情中查看并记录“内网连接地址/公网连接地址”。
- 客户端所在服务器和RabbitMQ实例之间网络已互通，具体网络要求参见[连接RabbitMQ网络要求](#)。
- 客户端所在服务器已安装[Java Development Kit 1.8.111或以上版本](#)，并配置JAVA_HOME与PATH环境变量，环境变量配置方法如下：

使用执行用户在用户家目录下修改“.bash_profile”，添加如下行。其中“/opt/java/jdk1.8.0_151”为JDK的安装路径，请根据实际情况修改。

```
export JAVA_HOME=/opt/java/jdk1.8.0_151  
export PATH=$JAVA_HOME/bin:$PATH
```

执行**source .bash_profile**命令使修改生效。

- RabbitMQ实例中已创建Vhost、Exchange和Queue，且配置Exchange和Queue的绑定。

命令行模式连接实例

步骤1 登录客户端所在服务器。

步骤2 下载RabbitMQ-Tutorial.zip示例工程代码。

```
wget https://dms-demo.obs.cn-north-1.myhuaweicloud.com/RabbitMQ-Tutorial.zip
```

步骤3 解压RabbitMQ-Tutorial.zip压缩包。

```
unzip RabbitMQ-Tutorial.zip
```

步骤4 进入RabbitMQ-Tutorial目录，该目录下包含预编译好的jar文件。

```
cd RabbitMQ-Tutorial
```

步骤5 运行生产消息示例。

```
java -cp ../rabbitmq-tutorial.jar Send {host} {port} {user} {password}
```

参数说明如下：

- **{host}**：从前提条件中获取的连接地址。
- **{port}**：RabbitMQ实例的连接端口，输入5672。
- **{user}**：从前提条件中获取的用户名。
- **{password}**：从前提条件中获取的密码。

生产消息示例如下：

```
[root@ecs-test RabbitMQ-Tutorial]# java -cp ../rabbitmq-tutorial.jar Send 192.168.xx.40 5672 test Zxxxxxxs  
[x] Sent 'Hello World!'  
[root@ecs-test RabbitMQ-Tutorial]# java -cp ../rabbitmq-tutorial.jar Send 192.168.xx.40 5672 test Zxxxxxxs  
[x] Sent 'Hello World!'
```

步骤6 运行消费消息示例。

```
java -cp ../rabbitmq-tutorial.jar Recv {host} {port} {user} {password}
```

参数说明如下：

- **{host}**：从前提条件中获取的连接地址。
- **{port}**：RabbitMQ实例的连接端口，输入5672。
- **{user}**：从前提条件中获取的用户名。
- **{password}**：从前提条件中获取的密码。

消费消息示例如下：

```
[root@ecs-test RabbitMQ-Tutorial]# java -cp ../rabbitmq-tutorial.jar Recv 192.168.xx.40 5672 test Zxxxxxxs  
[*] Waiting for messages. To exit press CTRL+C  
[x] Received 'Hello World!'  
[x] Received 'Hello World!'
```

如需停止消费使用Ctrl+C命令退出。

----结束

示例代码（Java）

- 连接实例并生产消息示例代码：

- **VHOST_NAME**: 消息要发送的Queue所在的Vhost名称。
- **QUEUE_NAME**: 消息要发送的Queue名称。
- **Hello World!**: 要发送的消息，根据实际需要修改。

```
ConnectionFactory factory = new ConnectionFactory();
factory.setHost(host);
factory.setPort(port);
factory.setVirtualHost("VHOST_NAME");

factory.setUsername(user);
factory.setPassword(password);
Connection connection = factory.newConnection();
Channel channel = connection.createChannel();

channel.queueDeclare(QUEUE_NAME, false, false, false, null);

String message = "Hello World!";
channel.basicPublish("", QUEUE_NAME, null, message.getBytes("UTF-8"));
System.out.println(" [x] Sent '" + message + "'");

channel.close();
connection.close();
```

- 连接实例并消费消息示例代码：
 - **VHOST_NAME**: 要消费消息的Queue所在的Vhost名称。
 - **QUEUE_NAME**: 要消费消息的Queue名称。

```
ConnectionFactory factory = new ConnectionFactory();
factory.setHost(host);
factory.setPort(port);
factory.setVirtualHost("VHOST_NAME");
factory.setUsername(user);
factory.setPassword(password);
Connection connection = factory.newConnection();
Channel channel = connection.createChannel();

channel.queueDeclare(QUEUE_NAME, false, false, false, null);
System.out.println(" [*] Waiting for messages. To exit press CTRL+C");

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 + "'");
    }
};
channel.basicConsume(QUEUE_NAME, true, consumer);
```

5.5 使用客户端连接 RabbitMQ（开启 SSL）

本章节以分布式消息服务RabbitMQ版提供的demo为例，介绍RabbitMQ客户端如何连接开启SSL的RabbitMQ实例，并生产和消费消息。开启SSL访问，则数据加密传输，安全性更高。

前提条件

- 已购买RabbitMQ实例，并记录创建时输入的用户名和密码，实例开启SSL。
- 在实例详情中查看并记录“内网连接地址/公网连接地址”。

- 客户端所在服务器和RabbitMQ实例之间网络已互通，具体网络要求参见[连接RabbitMQ网络要求](#)。
- 客户端所在服务器已安装[Java Development Kit 1.8.111或以上版本](#)，并配置JAVA_HOME与PATH环境变量，环境变量配置方法如下：
使用执行用户在用户家目录下修改“.bash_profile”，添加如下行。其中“/opt/java/jdk1.8.0_151”为JDK的安装路径，请根据实际情况修改。

```
export JAVA_HOME=/opt/java/jdk1.8.0_151
export PATH=$JAVA_HOME/bin:$PATH
```

执行source .bash_profile命令使修改生效。
- RabbitMQ实例中已[创建Vhost](#)、[Exchange](#)和[Queue](#)，且[配置Exchange和Queue的绑定](#)。

命令行模式连接实例

步骤1 登录客户端所在服务器。

步骤2 下载RabbitMQ-Tutorial-SSL.zip示例工程代码。

```
wget https://dms-demo.obs.cn-north-1.myhuaweicloud.com/RabbitMQ-Tutorial-SSL.zip
```

步骤3 解压RabbitMQ-Tutorial-SSL.zip压缩包。

```
unzip RabbitMQ-Tutorial-SSL.zip
```

步骤4 进入RabbitMQ-Tutorial-SSL目录，该目录下包含预编译好的jar文件。

```
cd RabbitMQ-Tutorial-SSL
```

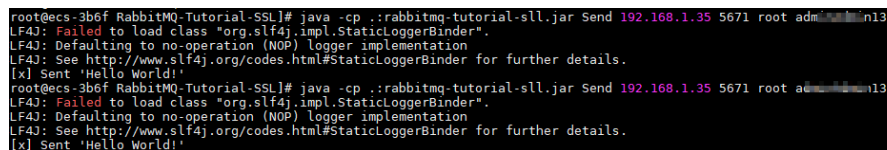
步骤5 运行生产消息示例。

```
java -cp ../rabbitmq-tutorial-sll.jar Send {host} {port} {user} {password}
```

参数说明如下：

- **{host}**：从[前提条件](#)中获取的连接地址。
- **{port}**：RabbitMQ实例的连接端口，输入5671。
- **{user}**：从[前提条件](#)中获取的用户名。
- **{password}**：从[前提条件](#)中获取的密码。

图 5-3 生产消息示例



```
root@ecs-3b6f RabbitMQ-Tutorial-SSL]# java -cp ../rabbitmq-tutorial-sll.jar Send 192.168.1.35 5671 root admin123
LF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
LF4J: Defaulting to no-operation (NOP) logger implementation
LF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
[x] Sent 'Hello World!'
root@ecs-3b6f RabbitMQ-Tutorial-SSL]# java -cp ../rabbitmq-tutorial-sll.jar Send 192.168.1.35 5671 root admin123
LF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
LF4J: Defaulting to no-operation (NOP) logger implementation
LF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
[x] Sent 'Hello World!'
```

步骤6 运行消费消息示例。

```
java -cp ../rabbitmq-tutorial-sll.jar Recv {host} {port} {user} {password}
```

参数说明如下：

- **{host}**：从[前提条件](#)中获取的连接地址。
- **{port}**：RabbitMQ实例的连接端口，输入5671。
- **{user}**：从[前提条件](#)中获取的用户名。
- **{password}**：从[前提条件](#)中获取的密码。

图 5-4 消费消息示例

```
root@ecs-3b6f RabbitMQ-Tutorial-SSL]# java -cp ./rabbitmq-tutorial-sll.jar Recv 192.168.1.35 5671 root ad
LF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
LF4J: Defaulting to no-operation (NOP) logger implementation
LF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
[*] Waiting for messages. To exit press CTRL+C
[x] Received 'Hello World!'
[x] Received 'Hello World!'
C[root@ecs-3b6f RabbitMQ-Tutorial-SSL]#
```

如需停止消费使用Ctrl+C命令退出。

----结束

示例代码（Java）

- 连接实例并生产消息示例代码：
 - **VHOST_NAME**: 消息要发送的Queue所在的Vhost名称。
 - **QUEUE_NAME**: 消息要发送的Queue名称。
 - **Hello World!**: 要发送的消息，根据实际需要修改。

```
ConnectionFactory factory = new ConnectionFactory();
factory.setHost(host);
factory.setPort(port);
factory.setVirtualHost("VHOST_NAME");

factory.setUsername(user);
factory.setPassword(password);
factory.useSslProtocol();
Connection connection = factory.newConnection();
Channel channel = connection.createChannel();

channel.queueDeclare(QUEUE_NAME, false, false, false, null);

String message = "Hello World!";
channel.basicPublish("", QUEUE_NAME, null, message.getBytes("UTF-8"));
System.out.println(" [x] Sent '" + message + "'");

channel.close();
connection.close();
```

- 连接实例并消费消息示例代码：
 - **VHOST_NAME**: 要消费消息的Queue所在的Vhost名称。
 - **QUEUE_NAME**: 要消费消息的Queue名称。

```
ConnectionFactory factory = new ConnectionFactory();
factory.setHost(host);
factory.setPort(port);
factory.setVirtualHost("VHOST_NAME");
factory.setUsername(user);
factory.setPassword(password);
factory.useSslProtocol();
Connection connection = factory.newConnection();
Channel channel = connection.createChannel();

channel.queueDeclare(QUEUE_NAME, false, false, false, null);
System.out.println(" [*] Waiting for messages. To exit press CTRL+C");

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 + "'");
    }
}
```

```
    }  
};  
channel.basicConsume(QueueName, true, consumer);
```

6 管理消息

6.1 查看 RabbitMQ 消息


生产者向Queue生产消息后，可以在控制台查看指定Vhost和Queue的消息内容和消息轨迹。

说明

仅RabbitMQ AMQP-0-9-1版本支持在控制台查看消息。


查看 RabbitMQ 消息

步骤1 登录管理控制台。

步骤2 在管理控制台左上角单击 ，选择区域。

说明

此处请选择RabbitMQ实例所在的区域。

步骤3 在管理控制台左上角单击 ，选择“应用中间件 > 分布式消息服务RabbitMQ版”，进入分布式消息服务RabbitMQ专享版页面。

步骤4 单击RabbitMQ实例名称，进入实例详情页面。

步骤5 在左侧导航栏选择“消息查询”，进入消息列表页面。

步骤6 参考表6-1，设置查询参数。

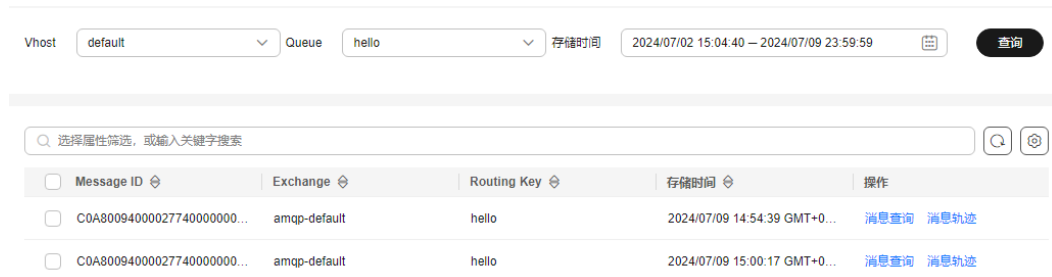
表 6-1 查询消息参数说明

参数名称	说明
Vhost	选择待查询消息所属的Vhost。
Queue	选择待查询消息所属的Queue。
存储时间	选择消息存储到Queue的时间。

步骤7 单击“查询”，查询消息。

查询结果如下：

图 6-1 查询消息



消息的参数说明如**表6-2**所示。

表 6-2 消息参数说明

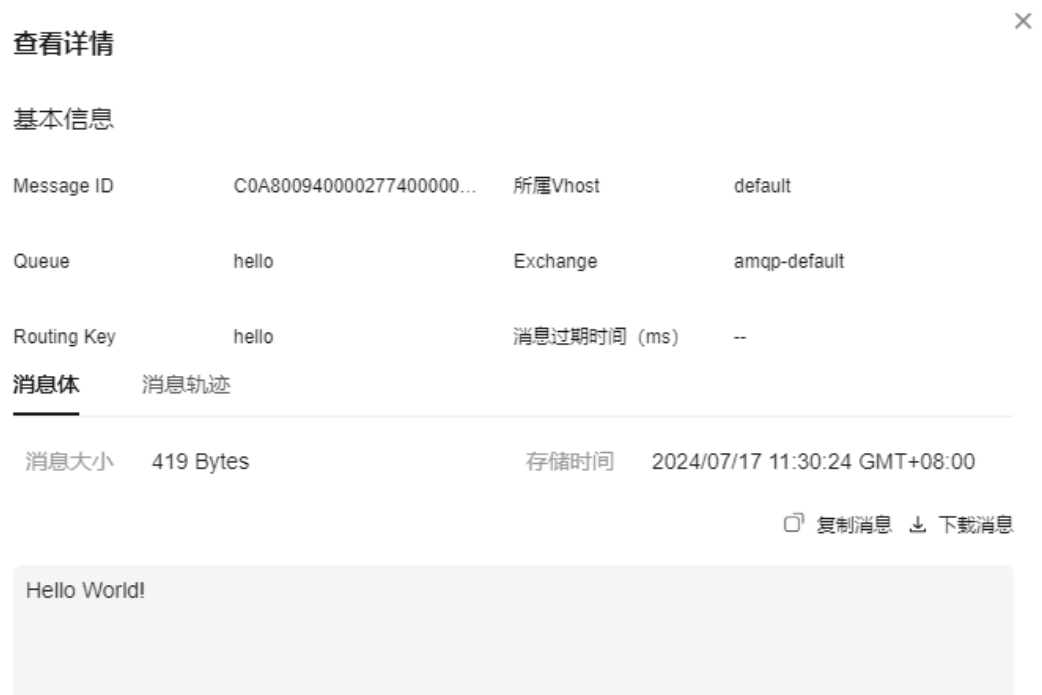
参数名称	说明
Message ID	消息标识符。
Exchange	消息所属的Exchange。
Routing Key	将消息从Exchange路由到Queue的关键字。
存储时间	消息存储到Queue的时间。

步骤8 单击消息所在行的“消息查询”，弹出“查看详情”对话框，在“消息体”页签中查看消息大小、存储时间和消息内容。

说明

控制台只能展示4KB以内的消息，超过4KB的消息，请单击“下载消息”，下载到本地查看消息完整的内容。

图 6-2 消息体



步骤9 单击消息所在行的“消息轨迹”，弹出“查看详情”对话框，在“消息轨迹”页签中查看消息是否消费成功。

图 6-3 消息轨迹



消息轨迹的参数说明如表6-3所示。

表 6-3 消息轨迹的参数说明

参数	参数说明
生产者状态	生产者状态如下： 发送成功：消息发送成功，服务端已经成功存储消息。

参数	参数说明
生产耗时	生产者发送消息的耗时，单位毫秒。
地址	生产者的IP地址。
消费者状态	消费者状态如下： <ul style="list-style-type: none"> • 消费成功 • 消费超时 • 消费异常 • 消费返回NULL • 消费失败
消费时间	消费消息的时间。
消费耗时	消费者消费消息的耗时，单位毫秒。
地址	消费者的IP地址。

----结束

6.2 配置 RabbitMQ 死信消息

死信是RabbitMQ中的一种消息机制，在消费消息时，如果队列里的消息符合以下任何一种情况，那么该消息将成为“死信”。

- “requeue”被设置为“false”，消费者使用“basic.reject”或“basic.nack”否定应答（NACK）消息。
- 消息在队列的存活时间超过设置的TTL时间。
- 队列的消息数量已经超过最大队列长度。

死信消息会被RabbitMQ进行特殊处理，如果配置了死信队列，该消息将会被存储到死信队列中，如果没有配置死信队列，该消息将会被丢弃。

更多关于死信的说明，请参考[Dead Letter Exchanges](#)。

须知

在RabbitMQ中，使用死信可能会对性能产生负面影响，请慎用。

使用队列参数配置死信交换机和路由

为队列配置死信交换机，并在创建队列时指定“x-dead-letter-exchange”和“x-dead-letter-routing-key”参数。队列根据“x-dead-letter-exchange”将死信消息发送到死信交换机中，并根据“x-dead-letter-routing-key”为死信消息设置死信路由Key。

以下示例演示在Java客户端配置死信交换机和路由：

```
channel.exchangeDeclare("some.exchange.name", "direct");
```

```
Map<String, Object> args = new HashMap<String, Object>();
args.put("x-dead-letter-exchange", "some.exchange.name");
args.put("x-dead-letter-routing-key", "some-routing-key");
channel.queueDeclare("myqueue", false, false, false, args);
```

6.3 配置 RabbitMQ 消息确认机制

RabbitMQ的消息确认机制分为生产者确认和消费者确认。在使用RabbitMQ时，生产者确认和消费者确认对于确保数据可靠性至关重要。如果连接失败，传输中的消息可能会丢失，需要重新传输。消息确认机制可以让服务端和客户端知道何时重新传输消息。客户端可以在收到消息时确认消息，也可以在客户端完全处理完消息后确认。

生产者确认会影响性能，如果需要很高的吞吐量，应禁用生产者确认。注意，不使用生产者确认会导致可靠性下降。

更多关于消息确认机制的说明，请参考[Consumer Acknowledgements and Publisher Confirms](#)。

生产者确认

生产者确认，即服务端在收到来自生产者的消息时进行确认。

以下示例演示在Java客户端配置生产者确认：

```
try {
    channel.confirmSelect(); //将信道置为publisher confirm模式
    //之后正常发送消息
    channel.basicPublish("exchange", "routingKey", null, "publisher confirm test".getBytes());
    if (!channel.waitForConfirms()) {
        System.out.println("send message failed ");
        // do something else....
    }
} catch (InterruptedException e) {
    e.printStackTrace();
}
```

调用`channel.waitForConfirms`方法之后，会等待服务端确认，这是一种同步等待的方式，会对性能产生影响。如果生产者要满足`at least once`，就必须使用同步等待方式。

消费者确认

消费者确认是指服务端通过确认消息是否成功被消费者接收，来判断是否删除队列中的此消息。

消费者确认对数据可靠性十分重要，接收重要消息的消费应用程序在未处理完消息前不应确认消息，以便消费者有足够的时间处理消息，无需担心消息处理过程中由于消费者进程异常（如工作程序崩溃、重启等）导致消息丢失。

消费者确认在客户端上配置，通过配置`basicConsume`方法启用确认。在channel中启用消费者确认适用于大多数场景。

以下示例演示在Java客户端配置消费者确认（使用`Channel#basicAck`设置`basic.ack`为肯定）：

```
// this example assumes an existing channel instance

boolean autoAck = false;
channel.basicConsume(queueName, autoAck, "a-consumer-tag",
    new DefaultConsumer(channel) {
```

```
@Override
public void handleDelivery(String consumerTag,
                           Envelope envelope,
                           AMQP.BasicProperties properties, byte[] body)
    throws IOException
{
    long deliveryTag = envelope.getDeliveryTag();
    // positively acknowledge a single delivery, the message will
    // be discarded
    channel.basicAck(deliveryTag, false);
}
});
```

未确认的消息缓存在内存中，如果未确认的消息过多，会导致内存使用率过高，此时可以在客户端配置预取值来限制消费者预取的消息数量，具体方法请参见[配置 RabbitMQ 消息预取值](#)。

6.4 配置 RabbitMQ 消息预取值

设置预取值可以限制未被确认的消息个数，一旦消费者中未被确认的消息数量达到设置的预取值，服务端将不再向此消费者发送消息，除非至少有一个未被确认的消息被确认。设置预取值本质上是一种对消费者进行流控的方法。

设置预取值时，需要考虑多种因素：

- 预取值设置太小可能会损害性能，RabbitMQ 会一直在等待获得发送消息的权限。
- 预取值设置太大可能会导致从队列中取出大量消息传递给一个消费者，而使其他消费者处于空闲状态。另外还需要考虑消费者的配置，消费者在处理消息时会将所有消息保存在内存中，太大的预取值会对消费者的性能产生负面影响，甚至可能会导致消费者崩溃。

更多关于预取值的说明，请参考[Consumer Prefetch](#)。

预取值设置建议

- 如果您只有一个或很少几个消费者在处理消息，建议一次预取多条消息，尽量让客户端保持忙碌。如果您的处理时间和网络状态稳定，则只需将总往返时间除以每条消息在客户端的处理时间即可获得估计的预取值。
- 在消费者多且处理时间短的情况下，建议使用较低的预取值。过低的预取值会使消费者闲置，因为消费者在处理完消息后需要等待下一批的消息到达。过高的值可能会使单个消费者忙碌，其他消费者处于空闲状态。
- 在消费者多且处理时间很长的情况下，建议您将预取值设置为 1，以便消息在所有消费者间均匀分布。

说明

如果客户端配置的消息确认机制为自动确认，则设置的预取值无效，已确认的消息会从队列中删除。

设置预取值

以下示例演示在 Java 客户端为单个消费者设置预取值为 10。

```
ConnectionFactory factory = new ConnectionFactory();

Connection connection = factory.newConnection();
Channel channel = connection.createChannel();
```

```
//设置预取值为10。  
channel.basicQos(10, false);  
  
QueueingConsumer consumer = new QueueingConsumer(channel);  
channel.basicConsume("my_queue", false, consumer);
```

在Java客户端中，**global**的默认值为**false**，因此以上示例可以简单地写为**channel.basicQos(10)**。

global取值的含义如下：

- false：分别作用于通道上的每个新消费者。
- true：在通道上的所有消费者之间所共享。

7 配置高级特性

7.1 配置 RabbitMQ 持久化

默认情况下，RabbitMQ生产者生产的消息存储在内存中，当节点宕机或重启时，会导致消息丢失。RabbitMQ提供持久化机制，可在节点宕机或重启时确保消息不丢失。持久化包括Exchange持久化、Queue持久化和Message持久化。

持久化是将内存中的消息写入到磁盘中，以防异常情况导致内存中的消息丢失。但是磁盘的读写速度远不如内存，开启消息持久化后，RabbitMQ的性能会下降。与惰性队列不同，持久化消息会在磁盘和内存中各存储一份，只有在内存空间不够时，才会将内存中的消息删除，存储到磁盘中。

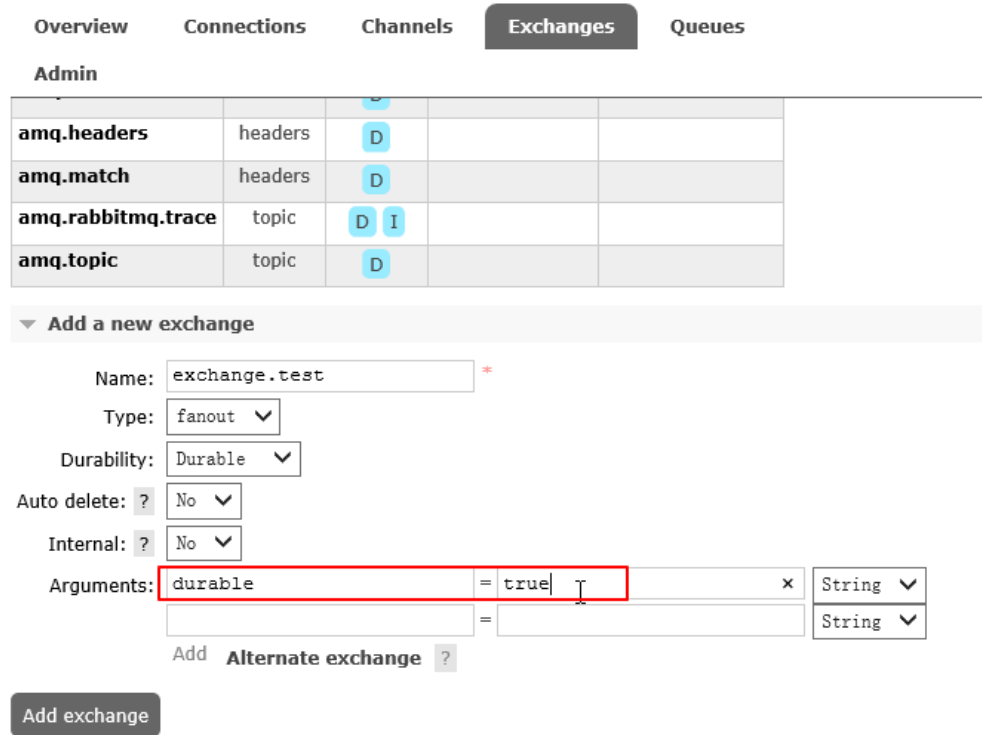
说明

- 非持久化的Queue、Exchange在重启之后会丢失。
- 非持久化的Message在重启之后会丢失（经过持久化Queue/Exchange的消息不会自动变为持久化消息）。
- 持久化消息在尚未完成持久化时，如果服务器重启，消息会丢失。
- RabbitMQ AMQP-0-9-1版本Exchange、Queue、Message默认开启持久化。

设置 Exchange 持久化

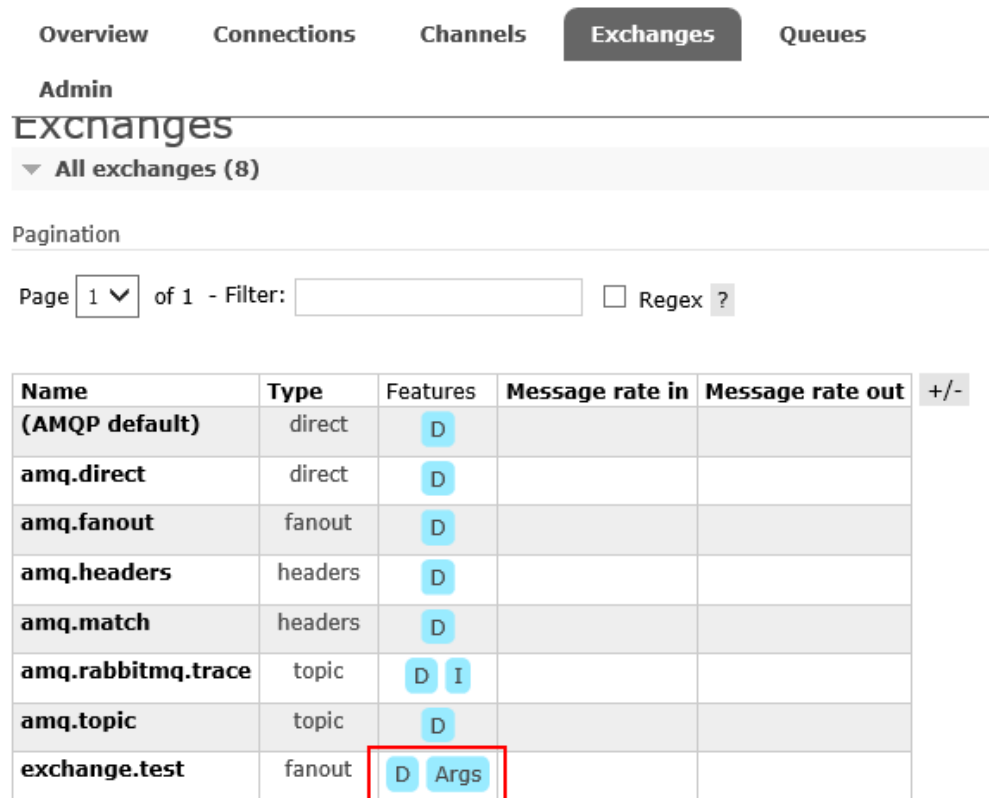
- 在[RabbitMQ WebUI页面](#)设置Exchange持久化。
创建Exchange时，设置“durable”为“true”，如[图7-1](#)所示。

图 7-1 设置 Exchange 持久化 (WebUI)



设置成功后如图7-2所示。

图 7-2 持久化的 Exchange (WebUI)



- 在RabbitMQ实例控制台设置Exchange持久化。
创建Exchange时，设置Exchange持久化，如图7-3所示。

图 7-3 设置 Exchange 持久化（控制台）

创建Exchange

i 1.Exchange是消息队列RabbitMQ的消息路由代理。
2.Exchange根据Binding key、Routing key以及Headers属性路由消息。
3.生产者向分布式消息服务RabbitMQ版发送消息时，不会直接将消息发送到Queue，而是先将消息发送到Exchange，由Exchange将消息路由到一个或多个Queue。

* 名称

Exchange创建成功后，其名称不可修改。

* 类型 direct fanout topic headers

* 自动删除

* 持久化

* Internal

设置成功后如图7-4所示。

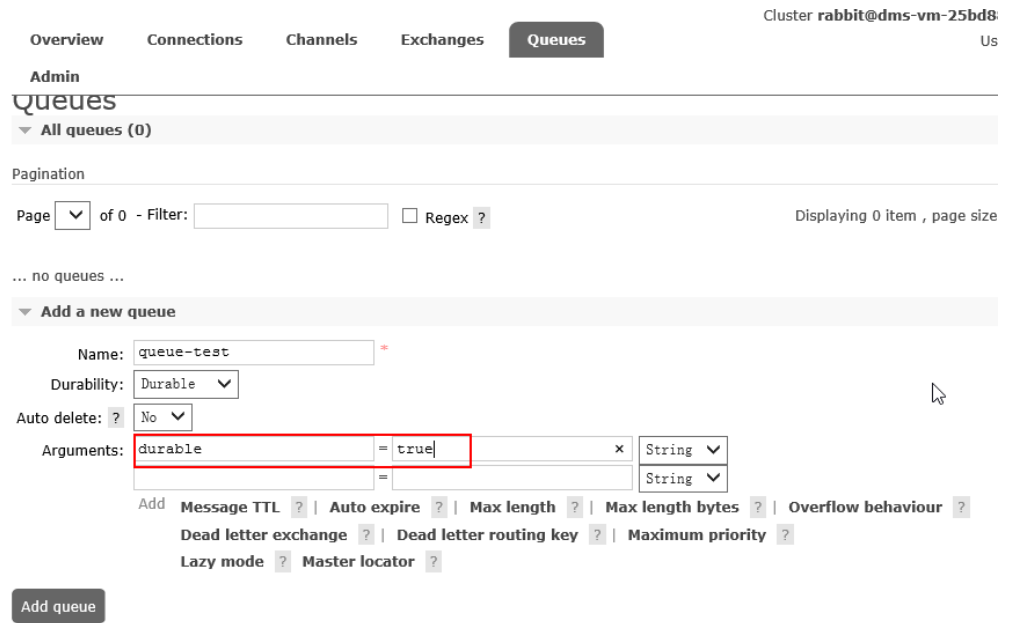
图 7-4 持久化的 Exchange（控制台）

<input type="checkbox"/> 名称	<input type="checkbox"/> 默认Exchange	<input type="checkbox"/> 类型	<input type="checkbox"/> 持久化	<input type="checkbox"/> Internal	<input type="checkbox"/> 自动删除	操作
<input type="checkbox"/> (AMQP default)	是	direct	是	否	否	绑定 删除
<input type="checkbox"/> Exchange-01	否	direct	是	否	否	绑定 删除
<input type="checkbox"/> amq.direct	是	direct	是	否	否	绑定 删除

设置 Queue 持久化

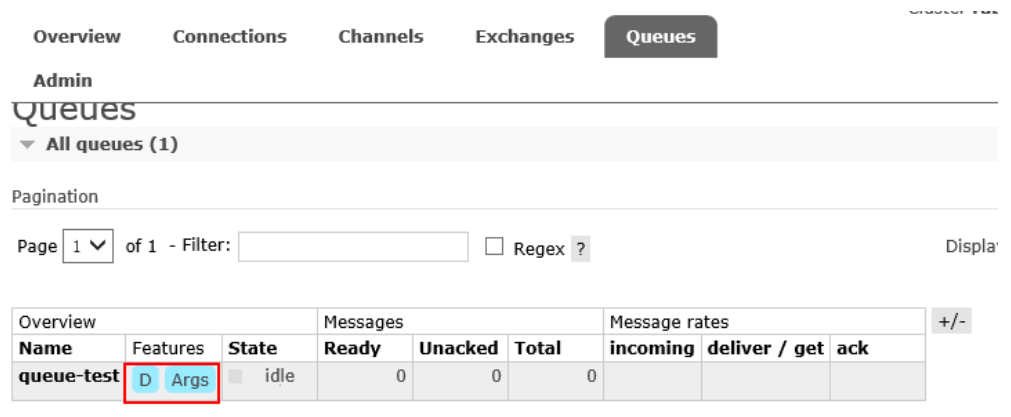
- 在RabbitMQ WebUI页面设置Queue持久化。
创建Queue时，设置“durable”为“true”，如图7-5所示。

图 7-5 设置 Queue 持久化 (WebUI)



设置成功后如图7-6所示。

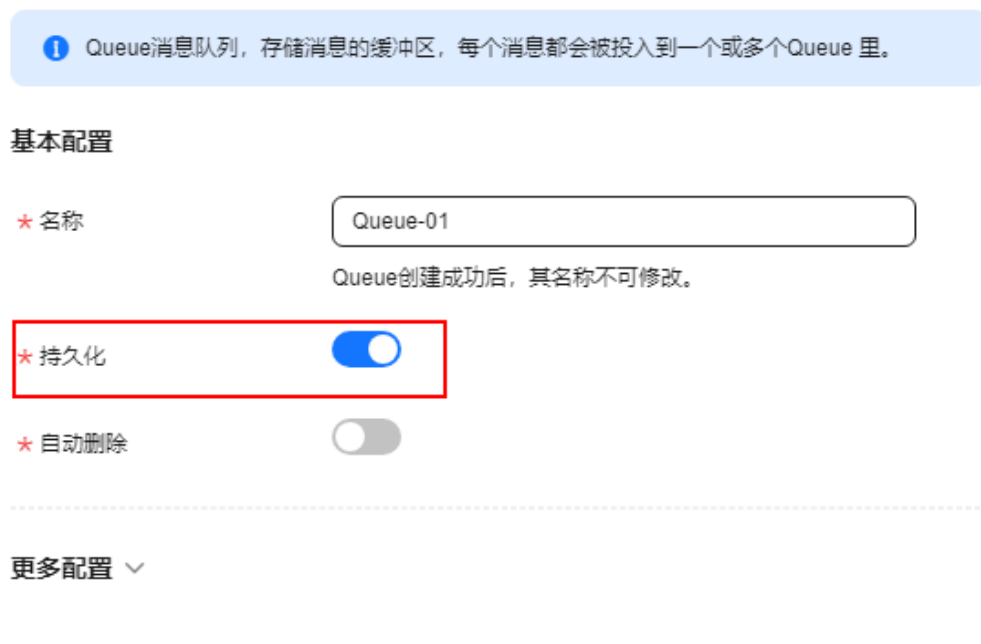
图 7-6 持久化的 Queue (WebUI)



- 在RabbitMQ实例控制台设置Queue持久化。
创建Queue时，设置Queue持久化，如图7-7所示。

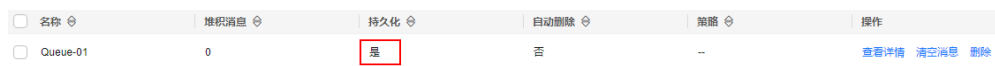
图 7-7 设置 Queue 持久化（控制台）

创建Queue



设置成功后如图7-8所示。

图 7-8 持久化的 Queue（控制台）



设置 Message 持久化

Queue设置为持久化后，可通过客户端设置“MessageProperties”为“PERSISTENT_TEXT_PLAIN”，向Queue发送持久消息。

以下示例演示在Java客户端设置Message持久化：

```
import com.rabbitmq.client.MessageProperties;  
channel.basicPublish("", "my_queue", MessageProperties.PERSISTENT_TEXT_PLAIN, message.getBytes());
```

7.2 配置 RabbitMQ TTL

TTL（Time To Live）即过期时间。消息在队列中的生存时间超过了TTL后，消息会被丢弃，如果队列设置了死信交换机，丢弃的消息会被转发到死信交换机，由死信交换机将其路由到死信队列。更多关于TTL的说明，请参考[TTL](#)。

RabbitMQ支持设置消息和队列的TTL，消息的TTL可以通过以下两种方法设置：

- 通过队列属性设置：队列中所有消息的具有相同的过期时间。
- 对消息本身单独设置：每条消息可以设置不同的TTL。

如果两种方法同时使用，以较小的TTL为准。

须知

TTL是RabbitMQ中需要慎用的特性，它可能会对性能产生负面影响。

设置队列 TTL

通过**channel.queueDeclare**方法中的“x-expires”参数控制队列被自动删除前处于未使用状态的时间。未使用是指队列中没有任何消费者，也没有被重新声明，并且在过期时间前也未调用过**Basic.Get**命令。“x-expires”参数的值必须为非零整数，单位为毫秒。

以下示例演示在Java客户端设置队列TTL。

```
Map<String, Object> args = new HashMap<String, Object>();  
args.put("x-expires", 1800000); //设置队列TTL为1800000毫秒  
channel.queueDeclare("myqueue", false, false, false, args);
```

设置消息 TTL

- 通过队列属性设置消息TTL：

在**channel.queueDeclare**方法中加入“x-message-ttl”参数，此参数的值必须为非零整数，单位为毫秒。

以下示例演示在Java客户端通过队列属性设置消息TTL。

```
Map<String, Object> arg = new HashMap<String, Object>();  
arg.put("x-message-ttl", 6000); //设置队列的消息TTL为6000毫秒  
channel.queueDeclare("normalQueue", true, false, false, arg);
```

- 对消息本身单独设置TTL：

在**channel.basicPublish**方法中加入“expiration”参数，此参数的值必须为非零整数，单位为毫秒。

以下示例演示在Java客户端对消息本身单独设置TTL。

```
byte[] messageBodyBytes = "Hello, world!".getBytes();  
AMQP.BasicProperties properties = new AMQP.BasicProperties.Builder()  
    .expiration("60000") //设置消息的TTL为60000毫秒  
    .build();  
channel.basicPublish("my-exchange", "routing-key", properties, messageBodyBytes);
```

8 管理实例

8.1 查看和修改 RabbitMQ 实例基本信息

本节介绍如何在控制台查看RabbitMQ实例的详细信息，以及修改RabbitMQ实例的基本信息。


创建RabbitMQ实例成功后，您可以根据自己的业务情况对RabbitMQ实例的部分配置信息进行调整，包括实例名称、描述、安全组等。

前提条件

RabbitMQ实例的状态为“运行中”时，才能修改实例的基本信息。


查看 RabbitMQ 实例的详细信息

步骤1 登录管理控制台。

步骤2 在管理控制台左上角单击 ，选择区域。

说明

此处请选择RabbitMQ实例所在的区域。

步骤3 在管理控制台左上角单击 ，选择“应用中间件 > 分布式消息服务RabbitMQ版”，进入分布式消息服务RabbitMQ专享版页面。

步骤4 RabbitMQ实例支持通过筛选来查询对应的RabbitMQ实例。当前支持的筛选条件为状态、名称、版本、实例类型、规格、已用/可用存储空间、计费模式、连接地址、弹性IP地址、企业项目和标签。企业用户才能使用企业项目进行筛选。RabbitMQ实例状态请参见[表8-1](#)。

表 8-1 RabbitMQ 实例状态说明

状态	说明
创建中	创建RabbitMQ实例后，在RabbitMQ实例状态进入运行中之前的状态。

状态	说明
创建失败	RabbitMQ实例处于创建实例失败的状态。
运行中	RabbitMQ实例正常运行状态。 在这个状态的实例可以运行您的业务。
故障	RabbitMQ实例处于故障的状态。
启动中	RabbitMQ实例从已冻结到运行中的中间状态。
变更中	RabbitMQ实例正在进行规格变更操作。
变更失败	RabbitMQ实例处于规格变更操作失败的状态。
已冻结	RabbitMQ实例处于已冻结状态。
冻结中	RabbitMQ实例从运行中到已冻结的中间状态。
升级中	RabbitMQ实例正在进行升级操作。
回滚中	RabbitMQ实例正在进行回滚操作。
回收中	RabbitMQ实例正在进行临时删除操作。
回收站中	RabbitMQ实例处于回收站中。
恢复中	RabbitMQ实例正在从回收站中恢复。

步骤5 单击RabbitMQ实例的名称，进入该RabbitMQ实例的基本信息页面，查看RabbitMQ实例的详细信息。

表8-2和**表8-3**为连接实例的相关参数，其他参数，请查看页面显示。

表 8-2 连接参数说明（RabbitMQ 3.x.x 版本）

参数	说明
内网连接地址	未开启公网访问时，连接实例的地址。
Web界面UI地址	未开启公网访问时，访问实例管理工具的地址。
公网访问	是否开启公网访问开关。
公网连接地址	开启公网访问后，连接实例的地址。
公网访问Web界面UI地址	开启公网访问后，访问实例管理工具的地址。

表 8-3 连接参数说明（RabbitMQ AMQP-0-9-1 版本）


参数	说明
内网连接地址	未开启公网访问时，连接实例的地址。

参数	说明
ACL访问控制	是否开启ACL访问控制开关。 开启ACL访问控制后，生产消息和消费消息时，需要鉴权。

---结束


修改 RabbitMQ 实例的基本信息

步骤1 登录管理控制台。

步骤2 在管理控制台左上角单击 ，选择区域。

说明









此处请选择RabbitMQ实例所在的区域。



步骤3 在管理控制台左上角单击 ，选择“应用中间件 > 分布式消息服务RabbitMQ版”，进入分布式消息服务RabbitMQ专享版页面。

步骤4 单击RabbitMQ实例的名称，进入实例详情页面。

步骤5 以下参数支持修改。

表 8-4 RabbitMQ 实例支持修改的参数

参数名称	修改方法	查看修改结果
实例名称	单击  ，输入修改后的实例名称，单击  。 实例名称需要符合命名规则：长度为4~64个字符，由英文字母开头，只能由英文字母、数字、中划线、下划线组成。	页面右上角直接提示修改结果。
企业项目	单击  ，在下拉框中选择新的企业项目，单击  。 该参数针对企业用户使用。修改企业项目，不会重启实例。	页面右上角直接提示修改结果。
描述	单击  ，输入修改后的描述，单击  。 长度为0~1024个字符。	页面右上角直接提示修改结果。
安全组	单击  ，在下拉框中选择新的安全组，单击  。 修改安全组，不会重启实例。	页面右上角直接提示修改结果。

参数名称	修改方法	查看修改结果
公网访问	修改方法请参考 配置RabbitMQ实例的公网访问 。	系统跳转到“后台任务管理”页面，并显示当前任务的操作进度和结果。
ACL访问控制	单击  或  ，开启或关闭ACL访问控制。 仅RabbitMQ AMQP-0-9-1版本的实例支持此参数。	页面上方直接提示修改结果。

----结束

8.2 查看 RabbitMQ 客户端连接地址

客户端连接RabbitMQ实例生产和消费消息时，可通过RabbitMQ WebUI查看客户端的连接地址。

说明

- RabbitMQ AMQP-0-9-1版本不支持通过RabbitMQ WebUI查看客户端的连接地址。
- 客户端处于连接RabbitMQ实例时，才可以查看客户端连接地址。

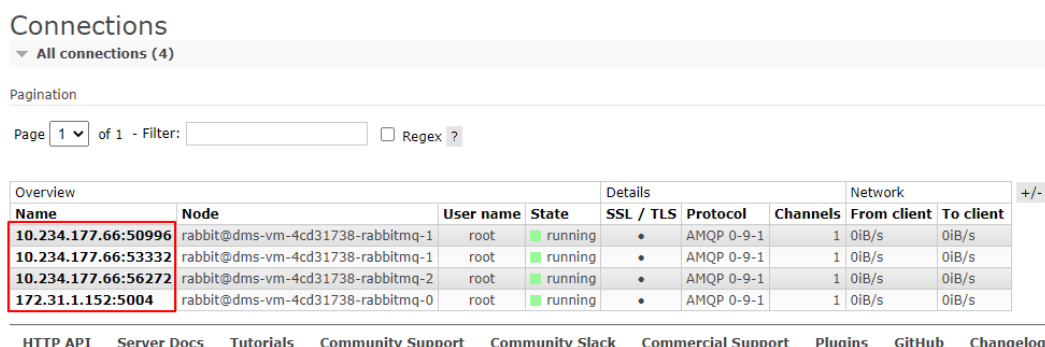
查看 RabbitMQ 客户端连接地址

步骤1 登录[RabbitMQ WebUI](#)。

步骤2 在导航栏单击“Connections”，进入“Connections”页面。

步骤3 查看客户端连接地址，如[图8-1](#)所示。

图 8-1 客户端连接地址



The screenshot shows the 'Connections' page in the RabbitMQ WebUI. It features a table with columns for Name, Node, User name, State, SSL / TLS, Protocol, Channels, From client, and To client. Four rows are visible, with the 'Name' column values highlighted in red: 10.234.177.66:50996, 10.234.177.66:53332, 10.234.177.66:56272, and 172.31.1.152:5004. The table also includes pagination controls and a filter input field.

Overview				Details			Network		
Name	Node	User name	State	SSL / TLS	Protocol	Channels	From client	To client	+/-
10.234.177.66:50996	rabbit@dms-vm-4cd31738-rabbitmq-1	root	running	•	AMQP 0-9-1	1	0iB/s	0iB/s	
10.234.177.66:53332	rabbit@dms-vm-4cd31738-rabbitmq-1	root	running	•	AMQP 0-9-1	1	0iB/s	0iB/s	
10.234.177.66:56272	rabbit@dms-vm-4cd31738-rabbitmq-2	root	running	•	AMQP 0-9-1	1	0iB/s	0iB/s	
172.31.1.152:5004	rabbit@dms-vm-4cd31738-rabbitmq-0	root	running	•	AMQP 0-9-1	1	0iB/s	0iB/s	

同一个客户端可以作为生产者生产消息，也可以作为消费者消费消息，连接IP地址是相同的，如[图8-1](#)所示，此时我们无法区分哪个是生产者IP地址，哪个是消费者IP地址。如果想要直观体现生产者/消费者IP地址，您可以在客户端中设置“clientProperties”参数，通过此参数来标明生产者/消费者IP地址，示例如下。

```
//配置客户端连接参数
HashMap<String, Object> clientProperties = new HashMap<>();
```



```
clientProperties.put("connection_name", "producer");
connectionFactory.setClientProperties(clientProperties);

//创建连接
Connection connection = connectionFactory.newConnection();
```

设置“clientProperties”参数后，连接地址显示如图8-2所示。

图 8-2 客户端连接地址（分区生产者/消费者 IP 地址）

Overview					Details		Network				+/-
Name	User name	State	SSL / TLS	Protocol	Channels	From client	To client	Heartbeat	Connected at		
10.234.177.66:65260 consumer	admin	running	○	AMQP 0-9-1	1	0iB/s	0iB/s	60s	10:53:21 2022-07-13		
10.234.177.66:58373 producer	admin	running	○	AMQP 0-9-1	1	0iB/s	0iB/s	60s	10:44:16 2022-07-13		

----结束

8.3 配置 RabbitMQ 实例标签

标签是RabbitMQ实例的标识，为RabbitMQ实例添加标签，方便您识别和管理拥有的RabbitMQ实例资源。

您可以在创建RabbitMQ实例时添加标签，也可以在RabbitMQ实例创建完成后，在“标签”页面添加标签，最多可以给实例添加20个标签。另外，您还可以删除标签。

如您的组织已经设定分布式消息服务RabbitMQ版的相关标签策略，则需按照标签策略规则为RabbitMQ实例添加标签。在“标签”页面新添加的标签不符合标签策略的规则，会添加失败。

标签共由两部分组成：“标签键”和“标签值”，其中，“标签键”和“标签值”的命名规则如表8-5所示。


表 8-5 标签命名规则

参数名称	规则
标签键	<ul style="list-style-type: none"> 不能为空。 对于同一个实例，Key值唯一。 长度为1~128个字符（中文也可以输入128个字符）。 由任意语种字母、数字、空格和字符组成，字符仅支持_ . : = + - @ 首尾字符不能为空格。 不能以_sys_开头。

参数名称	规则
标签值	<ul style="list-style-type: none"> 长度为0~255个字符（中文也可以输入255个字符）。 由任意语种字母、数字、空格和字符组成，字符仅支持_ . : = + - @ 创建实例时设置的标签值，首尾字符不能为空格。


配置 RabbitMQ 实例标签

步骤1 登录管理控制台。

步骤2 在管理控制台左上角单击 ，选择区域。

说明


此处请选择RabbitMQ实例所在的区域。

步骤3 在管理控制台左上角单击 ，选择“应用中间件 > 分布式消息服务RabbitMQ版”，进入分布式消息服务RabbitMQ专享版页面。

步骤4 单击待设置标签的实例名称，进入实例详情页面。

步骤5 单击“标签”页签，进入标签管理页面，页面显示该实例的标签列表。

步骤6 您可以根据实际需要，执行以下操作：

- 添加标签
 - 单击“创建/删除标签”，弹出“创建/删除标签”对话框。
 - 在“标签键”和“标签值”中，输入标签的键/值，单击“添加”。
如果您已经预定义了标签，在“标签键”和“标签值”中选择已经定义的标签键/值，单击“添加”。
 - 单击“确定”，成功为实例添加标签。
- 删除标签
通过以下任意一种方法，删除标签。
 - 在待删除的标签所在行，单击“删除”，弹出“删除标签”对话框。单击“是”，完成标签的删除。
 - 单击“创建/删除标签”，弹出“创建/删除标签”对话框。在待删除的标签后，单击 ，然后单击“确定”，完成标签的删除。

----结束

8.4 配置 RabbitMQ 实例回收站策略

开启回收站策略的实例在被删除后，会被临时存入回收站中，此时实例中的数据尚未被彻底删除，在保留天数内支持从回收站中恢复此实例。超过保留天数的实例会被彻底删除，无法恢复。

回收站策略默认是关闭状态。

📖 说明


回收站功能仅在“华北-北京四”、“华东-上海一”和“华南-广州”上线。

约束与限制

- 回收站中的按需实例不会收取实例的费用，但是会收取存储空间的费用。
- 包年/包月的实例退订后会存入回收站中，此时不会收取实例的费用，但是会按需收取存储空间的费用。


开启回收站策略

步骤1 登录管理控制台。

步骤2 在管理控制台左上角单击 ，选择区域。

📖 说明

请选择RabbitMQ实例所在的区域。

步骤3 在管理控制台左上角单击 ，选择“应用中间件 > 分布式消息服务RabbitMQ版”，进入分布式消息服务RabbitMQ专享版页面。

步骤4 在左侧导航栏选择“回收站”，进入“回收站”页面。

步骤5 单击“回收站策略”，弹出“回收站策略”对话框。

步骤6 开启启用回收站，并设置保留天数，单击“确定”。


📖 说明

- 保留天数只能设置为1~7天。
- 修改保留天数仅对修改后删除的实例生效，修改前已在回收站中的实例依然按照修改前的天数保留实例。

----结束


恢复 RabbitMQ 实例

步骤1 登录管理控制台。

步骤2 在管理控制台左上角单击 ，选择区域。

📖 说明

请选择RabbitMQ实例所在的区域。

步骤3 在管理控制台左上角单击 ，选择“应用中间件 > 分布式消息服务RabbitMQ版”，进入分布式消息服务RabbitMQ专享版页面。

步骤4 在左侧导航栏选择“回收站”，进入“回收站”页面。

步骤5 通过以下任意一种方法，恢复RabbitMQ实例。

- 勾选RabbitMQ实例名称左侧的方框，可选一个或多个，单击信息栏左上侧的“恢复”。

- 在待恢复RabbitMQ实例所在行，单击“恢复”。


步骤6 在弹出的“恢复实例”对话框中，单击“确定”。

恢复实例大约需要3~10分钟，在“RabbitMQ专享版”页面查看已恢复的实例。

----结束


修改保留天数

步骤1 登录管理控制台。

步骤2 在管理控制台左上角单击 ，选择区域。

说明

请选择RabbitMQ实例所在的区域。

步骤3 在管理控制台左上角单击 ，选择“应用中间件 > 分布式消息服务RabbitMQ版”，进入分布式消息服务RabbitMQ专享版页面。

步骤4 在左侧导航栏选择“回收站”，进入“回收站”页面。

步骤5 单击“回收站策略”，弹出“回收站策略”对话框。

步骤6 修改保留天数，单击“确定”。


说明

- 保留天数只能设置为1~7天。
- 修改保留天数仅对修改后删除的实例生效，修改前已在回收站中的实例依然按照修改前的天数保留实例。

----结束


导出回收站中的实例

步骤1 登录管理控制台。

步骤2 在管理控制台左上角单击 ，选择区域。

说明

请选择RabbitMQ实例所在的区域。

步骤3 在管理控制台左上角单击 ，选择“应用中间件 > 分布式消息服务RabbitMQ版”，进入分布式消息服务RabbitMQ专享版页面。


步骤4 在左侧导航栏选择“回收站”，进入“回收站”页面。

步骤5 单击“导出 > 导出全部数据到XLSX” / “导出 > 导出已选中数据到XLSX”，导出实例列表。

----结束


删除回收站中的实例

步骤1 登录管理控制台。

步骤2 在管理控制台左上角单击 ，选择区域。

说明

请选择RabbitMQ实例所在的区域。

步骤3 在管理控制台左上角单击 ，选择“应用中间件 > 分布式消息服务RabbitMQ版”，进入分布式消息服务RabbitMQ专享版页面。

步骤4 在左侧导航栏选择“回收站”，进入“回收站”页面。

步骤5 通过以下任意一种方法，删除实例。

- 勾选RabbitMQ实例名称左侧的方框，可选一个或多个，单击信息栏左上侧的“删除”。
- 在待删除RabbitMQ实例所在行，单击“删除”。

步骤6 在弹出的“删除实例”对话框中，输入“DELETE”，单击“确定”，完成删除RabbitMQ实例。


须知

回收站中的RabbitMQ实例删除后，实例中原有的数据将被删除，且没有备份，请谨慎操作。

----结束


关闭回收站策略

步骤1 登录管理控制台。

步骤2 在管理控制台左上角单击 ，选择区域。

说明

请选择RabbitMQ实例所在的区域。

步骤3 在管理控制台左上角单击 ，选择“应用中间件 > 分布式消息服务RabbitMQ版”，进入分布式消息服务RabbitMQ专享版页面。

步骤4 在左侧导航栏选择“回收站”，进入“回收站”页面。

步骤5 单击“回收站策略”，弹出“回收站策略”对话框。

步骤6 关闭启用回收站，单击“确定”。

----结束

8.5 重置 RabbitMQ 实例密码

如果您忘记了创建实例时设置的密码，可以通过重置密码功能，重新设置一个新的密码，使用新密码连接RabbitMQ实例。

📖 说明


RabbitMQ AMQP-0-9-1版本不支持重置实例密码。

前提条件

RabbitMQ实例处于“运行中”状态时，才能重置密码。


重置 RabbitMQ 实例密码

步骤1 登录管理控制台。

步骤2 在管理控制台左上角单击 ，选择区域。

📖 说明

此处请选择RabbitMQ实例所在的区域。

步骤3 在管理控制台左上角单击 ，选择“应用中间件 > 分布式消息服务RabbitMQ版”，进入分布式消息服务RabbitMQ专享版页面。

步骤4 通过以下任意一种方法，重置实例密码。

- 在待重置密码的实例所在行，单击“更多 > 重置密码”。
- 单击RabbitMQ实例名称，进入实例详情页面。单击右上角的“更多 > 重置密码”。

步骤5 输入“新密码”和“确认密码”，单击“确定”完成密码重置。

- 重置密码成功，界面提示重置实例的密码成功。
- 重置密码失败，界面提示重置实例的密码失败，请重新尝试重置密码操作。如果多次重置失败，请联系客服处理。

📖 说明

只有所有代理都重置密码成功，才会提示重置密码成功，否则会提示重置失败。

----结束

8.6 开启 RabbitMQ 实例插件

RabbitMQ实例创建后，支持通过插件的方式开启附加功能，实例创建后插件默认都是关闭状态。

RabbitMQ插件功能可用于测试和迁移业务等场景，不建议用于生产业务。因使用插件导致的可靠性问题，不在服务承诺的SLA范围内。详情请参考[产品介绍 > 约束与限制](#)。

📖 说明

RabbitMQ AMQP-0-9-1版本不支持插件功能。

RabbitMQ支持的插件如表8-6所示。其中，插件端口号不支持修改。

表 8-6 插件列表

插件名称	功能描述	端口号
rabbitmq_federation	表示实例是否开启消息同步功能。	-
rabbitmq_shovel	表示实例是否开启消息迁移功能。	-
rabbitmq_consistent_hash_exchange	表示实例是否支持x-consistent-hash。开启此插件后，可以创建x-consistent-hash类型的Exchange。	-

约束与限制


- 2020年7月1日之前购买的RabbitMQ实例，不支持在控制台开启插件，如果需要开启，请提交工单联系技术人员在后台开启插件。
- 2020年11月14日之前购买的RabbitMQ实例，不支持在控制台开启rabbitmq_consistent_hash_exchange插件，如果需要开启，请提交工单联系技术人员在后台开启插件。
- 开启插件过程中，不会重启实例。
- rabbitmq_shovel和rabbitmq_federation插件在特定场景下才支持开启，具体如表8-7所示。

表 8-7 支持开启插件的场景

实例	rabbitmq_shovel	rabbitmq_federation
未开启SSL的单机实例	支持	支持
已开启SSL的单机实例	不支持	不支持
未开启SSL的集群实例	不支持	支持
已开启SSL的集群实例	不支持	不支持


开启 RabbitMQ 实例插件

步骤1 登录管理控制台。

步骤2 在管理控制台左上角单击 ，选择区域。

📖 说明

此处请选择RabbitMQ实例所在的区域。

步骤3 在管理控制台左上角单击 ，选择“应用中间件 > 分布式消息服务RabbitMQ版”，进入分布式消息服务RabbitMQ专享版页面。

步骤4 单击待开启插件的实例名称，进入实例详情页面。

步骤5 在“插件管理”页签，单击待开启插件后的“开启”。

确认开启后，等待实例插件开启成功。


----结束

8.7 导出 RabbitMQ 实例列表

RabbitMQ支持通过控制台导出实例列表信息，供用户查看和使用。


导出 RabbitMQ 实例列表

步骤1 登录管理控制台。

步骤2 在管理控制台左上角单击 ，选择区域。

说明

请选择RabbitMQ实例所在的区域。

步骤3 在管理控制台左上角单击 ，选择“应用中间件 > 分布式消息服务RabbitMQ版”，进入分布式消息服务RabbitMQ专享版页面。

步骤4 通过以下任何一种方法，导出实例列表。

- 勾选待导出的实例，单击“导出 > 导出已选中数据到XLSX”，导出指定实例列表。
- 单击“导出 > 导出全部数据到XLSX”，导出全部实例列表。

----结束

8.8 删除 RabbitMQ 实例

对于按需计费的RabbitMQ实例，控制台支持删除单个实例，也可批量删除实例。对于包年/包月的RabbitMQ实例，如果不再使用，单击“操作”栏下的“更多 > 退订”，退订成功后，RabbitMQ实例会自动被删除。


删除的实例能否恢复取决于是否开启回收站策略，如果未开启回收站策略，实例删除后，实例中原有的数据将被删除，且没有备份。回收站策略默认关闭，如果需要开启，具体步骤请参见[开启回收站策略](#)。

前提条件

RabbitMQ实例状态为运行中、故障、已冻结、创建失败的按需付费实例才能执行删除操作。


删除 RabbitMQ 实例

步骤1 登录管理控制台。

步骤2 在管理控制台左上角单击 ，选择区域。

说明

此处请选择RabbitMQ实例所在的区域。

步骤3 在管理控制台左上角单击 ，选择“应用中间件 > 分布式消息服务RabbitMQ版”，进入分布式消息服务RabbitMQ专享版页面。

步骤4 通过以下任意一种方法，删除按需计费的RabbitMQ实例。

- 勾选RabbitMQ实例名称左侧的方框，可选一个或多个，单击信息栏左上侧的“更多 > 删除”。
- 在待删除RabbitMQ实例所在行，单击“更多 > 删除”。
- 单击RabbitMQ实例名称，进入实例详情页面。单击右上角的“更多 > 删除”。

说明

RabbitMQ实例状态为创建中、启动中、变更中、变更失败时不允许执行删除操作。

步骤5 在“删除实例”对话框中，输入“DELETE”，单击“确定”，完成删除RabbitMQ实例。

删除RabbitMQ实例大约需要1到60秒。

----结束

8.9 登录 RabbitMQ WebUI


RabbitMQ实例提供了开源RabbitMQ的集群管理工具，通过RabbitMQ管理地址可访问WebUI并对实例进行配置操作。

说明

RabbitMQ AMQP-0-9-1版本不支持RabbitMQ WebUI。


登录 RabbitMQ WebUI

步骤1 获取实例管理地址。

1. 登录管理控制台。
2. 在管理控制台左上角单击 ，选择区域。

说明

此处请选择与您的应用服务相同的区域。

3. 在管理控制台左上角单击 ，选择“应用中间件 > 分布式消息服务RabbitMQ版”，进入分布式消息服务RabbitMQ专享版页面。
4. 单击实例名称，进入实例详情页面，获取Web界面UI地址和用户名。

📖 说明

用户名和密码为创建RabbitMQ实例时自定义的内容。

图 8-3 获取实例 Web 界面 UI 地址（未开启公网访问）

连接信息

SSL	关闭 当前实例暂不支持动态开启/关闭 SSL
用户名	root 重置密码
内网连接地址	IPv4 192.168.0.101:5672 📄
Web界面UI地址	http://192.168.0.101:15672 📄

图 8-4 获取实例 Web 界面 UI 地址（开启公网访问）

公网访问 ?	<input checked="" type="checkbox"/> 已开启
公网连接地址	100.█.█.█:144:5672 📄
公网访问Web界面UI地址	http://100.█.█.█:144:15672 📄

步骤2 确认实例安全组规则是否配置正确。

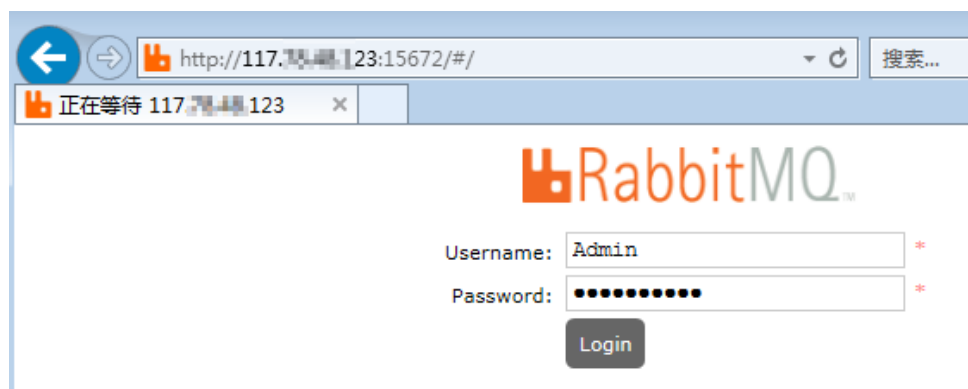
- 在实例详情页面的“基本信息 > 网络”，单击安全组名称，跳转到安全组页面。
- 选择“入方向规则”，查看安全组入方向规则。
 - 实例未开启SSL开关
 - 如果是VPC内访问，实例安全组入方向规则，需要允许端口5672的访问。
 - 如果是公网访问，需要允许端口15672的访问。
 - 实例已开启SSL开关
 - 如果是VPC内访问，实例安全组入方向规则，需要允许端口5671的访问。
 - 如果是公网访问，需要运行端口15671的访问。

步骤3 在浏览器中打开Web界面UI地址，进入WebUI登录页面。

📖 说明

- 如果RabbitMQ实例开启了公网访问，可直接在公网环境下的浏览器访问Web页面。
- 如果RabbitMQ实例未开启公网访问，您需要购买一台与RabbitMQ实例网络相通的Windows弹性云服务器，然后登录弹性云服务器访问Web页面。
购买弹性云服务器操作，请参考[购买并登录Windows弹性云服务器](#)。

图 8-5 登录实例 Web 页面



步骤4 输入用户名和密码，单击“Login”，登录完成。

----结束

9 变更 RabbitMQ 实例规格

RabbitMQ实例创建成功后，您可以根据业务需要，扩容或者缩容实例，RabbitMQ实例支持的变更配置如表9-1和表9-2所示。

表 9-1 变更配置列表（RabbitMQ 3.x.x 版本）

实例类型	变更配置类型	是否支持扩容	是否支持缩容
集群	代理个数	√	×
	存储空间	√	×
	代理规格	√	√
单机	代理个数	×	×
	存储空间	√	×
	代理规格	√	√

表 9-2 变更配置列表（RabbitMQ AMQP-0-9-1 版本）

实例类型	变更配置类型	是否支持扩容	是否支持缩容
集群	存储空间	√	×
	实例规格	√	×
单机	存储空间	√	×
	实例规格	√	×

约束与限制


- 为了实例运行正常，变更规格过程中，请勿对实例进行其他操作。
- 实例变更规格后，配置费用将发生变化。

前提条件

已创建RabbitMQ实例，且实例状态为“运行中”。


变更 RabbitMQ 实例规格

步骤1 登录管理控制台。

步骤2 在管理控制台左上角单击 ，选择区域。

说明

此处请选择RabbitMQ实例所在的区域。

步骤3 在管理控制台左上角单击 ，选择“应用中间件 > 分布式消息服务RabbitMQ版”，进入分布式消息服务RabbitMQ专享版页面。

步骤4 通过以下任意一种方法，变更实例规格。

- 在待变更规格的实例所在行，单击“更多 > 变更规格”。
- 单击RabbitMQ实例名称，进入实例详情页面。单击右上角的“更多 > 变更规格”。

步骤5 根据实际情况选择扩容存储空间、代理个数或者扩容/缩容代理规格。

- 扩容存储空间
 - RabbitMQ 3.x.x版本：在“变更类型”中，选择“存储空间”，在“单个代理存储空间”中，选择扩容后的单个代理的存储空间大小。
在实例列表页面的“可用存储空间”中查看扩容后的总存储空间大小（总存储空间大小=扩容后单个代理的存储空间*代理个数）。
 - RabbitMQ AMQP-0-9-1版本：在“变更类型”中，选择“存储空间”，在“存储空间”中，选择扩容后实例的存储空间大小。
在实例列表页面的“可用存储空间”中查看扩容后的总存储空间大小。

说明

- 可用存储空间=实际存储空间-用于存储日志的存储空间-格式化磁盘的损耗。例如，实际扩容存储空间到700GB，用于存储日志的数据的存储空间为100GB，格式化磁盘损耗7GB，那么扩容后的可用存储空间为593GB。
- 扩容存储空间不会影响业务。
- 扩容代理个数
在“变更类型”中，选择“代理数量”，在“代理数量”中，选择扩容后的代理个数。
在实例列表页面的“规格”中查看扩容后的代理个数。

说明

- 变更规格过程中会有秒级业务中断，客户端需要支持自动重连，建议在业务低谷时进行变更。
- RabbitMQ AMQP-0-9-1版本不支持扩容代理个数。
- 扩容/缩容代理规格
 - RabbitMQ 3.x.x版本：在“变更类型”中，选择“代理规格”，在“代理规格”中，选择扩容/缩容后的规格。
在实例列表页面的“规格”中查看扩容/缩容后的代理规格。

- RabbitMQ AMQP-0-9-1版本：在“变更类型”中，选择“实例规格”，在“实例规格”中，选择扩容后的规格。

在实例列表页面的“规格”中查看扩容后的实例规格。

说明

- RabbitMQ 3.x.x版本单机实例和没有配置镜像/仲裁队列的RabbitMQ 3.x.x版本集群实例在变更规格过程中会有分钟级业务中断，客户端需要支持自动重连，建议在业务低峰时进行变更。
- 配置了镜像/仲裁队列的RabbitMQ 3.x.x版本集群实例在变更规格过程中会有秒级业务中断，客户端需要支持自动重连，建议在业务低峰时进行变更。
- RabbitMQ AMQP-0-9-1版本单机实例和集群实例在变更规格过程中会有秒级连接闪断，客户端需要支持自动重连，建议在业务低峰时进行变更。

步骤6 单击“下一步”，确认目标规格信息无误后，单击“提交”。

---结束

10 迁移 RabbitMQ 业务

RabbitMQ业务迁移主要涉及到以下两个场景：

- 将线下单机或集群实例迁移到线上RabbitMQ实例。
- 将低版本的RabbitMQ实例迁移到高版本的RabbitMQ实例，例如将3.7.17版本的RabbitMQ实例迁移到3.8.35版本的RabbitMQ实例。

迁移原理

在实际业务场景中，RabbitMQ实例存在多个生产者 and 消费者，在业务迁移时，不会迁移数据，只是通过逐个增加/关闭消费者、生产者的方式迁移实例，这种迁移方式可以实现业务无感迁移。

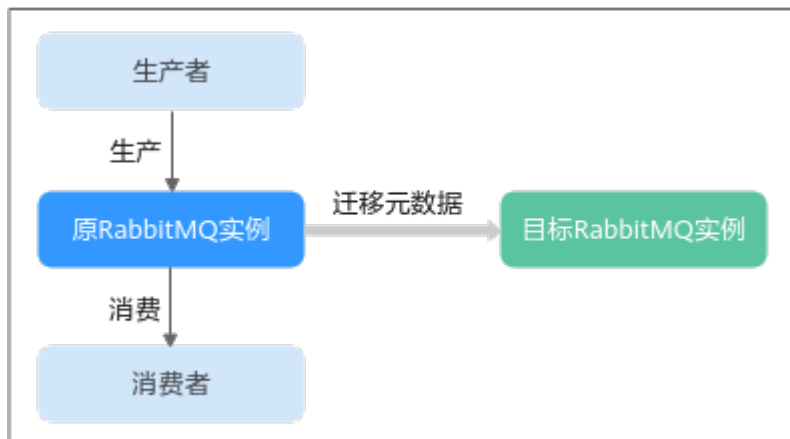
前提条件

已创建目标RabbitMQ实例，具体请参考[购买RabbitMQ实例](#)。

实施方案（双读模式迁移实例）

步骤1 将原RabbitMQ实例的元数据迁移到目标RabbitMQ实例。

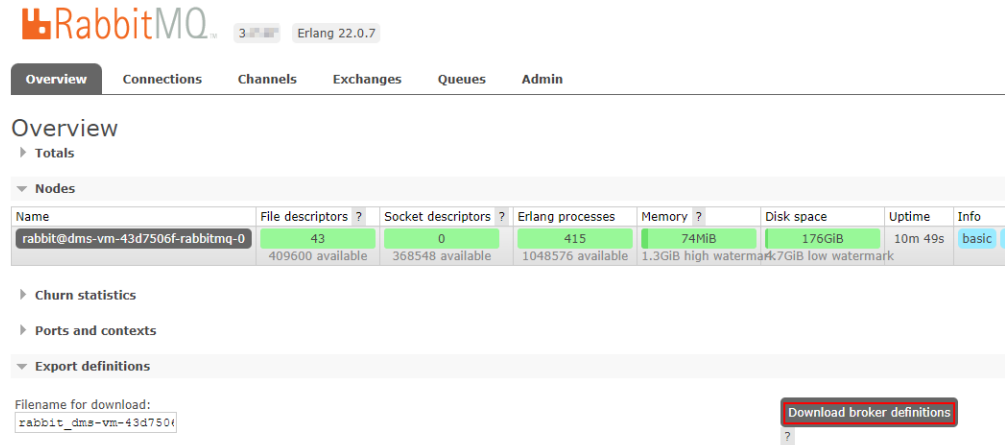
图 10-1 迁移元数据



迁移步骤如下：

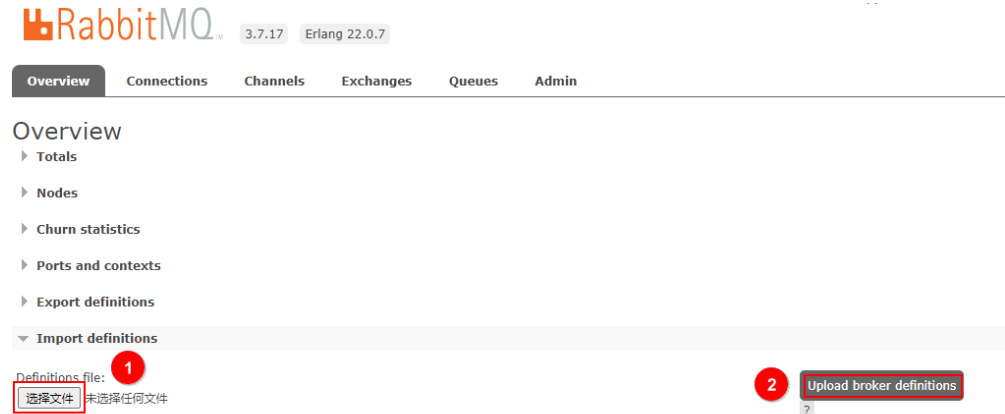
1. 登录原RabbitMQ的WebUI页面，在“Overview”页签中，单击“Download broker definitions”，导出元数据。

图 10-2 导出元数据



2. 登录目标RabbitMQ的WebUI页面，在“Overview”页签中，单击“选择文件”，选择步骤1.1中导出的元数据，单击“Upload broker definitions”，上传元数据。

图 10-3 导入元数据



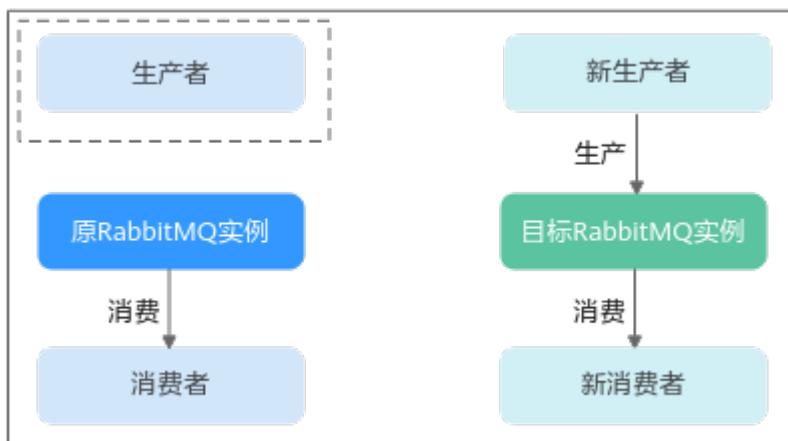
步骤2 为目标RabbitMQ实例添加新的消费者，准备消费目标实例的消息。

图 10-4 添加新消费者



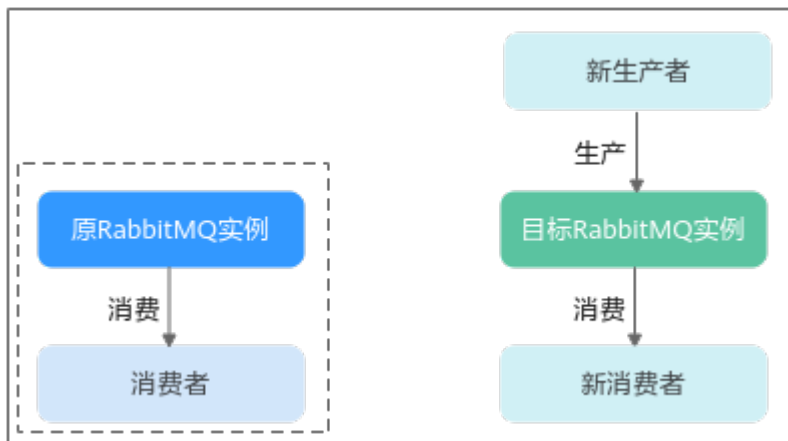
步骤3 为目标RabbitMQ实例添加新的生产者，下线原RabbitMQ实例的生产者，旧的消费者继续消费原RabbitMQ实例中的消息。

图 10-5 迁移生产者



步骤4 旧的消费者消费完原RabbitMQ实例的全部消息后，下线旧的消费者和原RabbitMQ实例。

图 10-6 下线旧的消费者和原 RabbitMQ 实例



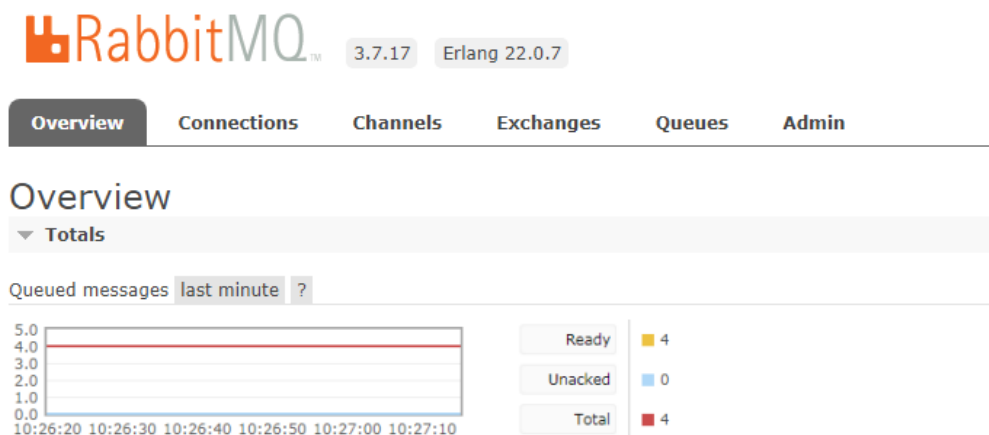
----结束

迁移后检查

通过以下方法，确认原实例是否消费完成：

- 在原RabbitMQ实例的WebUI页面查看，如图10-7所示。
Overview视图中，可消费消息数（Ready）以及未确认的消息数（Unacked）都为0时，说明消费完成。

图 10-7 RabbitMQ WebUI



- 调用API查看。
`curl -s -u username:password -XGET http://ip:port/api/overview`

参数说明：

- username：原实例登录RabbitMQ WebUI的账号
- password：原实例登录RabbitMQ WebUI的密码
- ip：原实例登录RabbitMQ WebUI的IP地址
- port：原实例登录RabbitMQ WebUI的端口号

回显信息中“messages_ready”和“messages_unacknowledged”都为0时，说明消费完成。

图 10-8 回显信息

```

"queue_totals":{
  "messages":0,
  "messages_details":{
    "rate":0
  },
  "messages_ready":0,
  "messages_ready_details":{
    "rate":0
  },
  "messages_unacknowledged":0,
  "messages_unacknowledged_details":{
    "rate":0
  }
},

```

11 测试实例性能

11.1 测试 RabbitMQ 生产速率和消费速率

本章节从实例规格、是否开启SSL、生产者/消费者数量、队列数量、队列类型、Exchange类型的维度进行分布式消息服务RabbitMQ版的性能测试，对比客户端消息生产速率和消费速率，得出性能测试结果。

- 测试场景一（实例规格）：相同Exchange、队列、生产者数量、消费者数量、不同的实例规格
- 测试场景二（是否开启SSL）：相同Exchange、队列、生产者数量、消费者数量、实例规格，开启SSL和未开启SSL的实例
- 测试场景三（生产者/消费者数量）：相同Exchange、队列、实例规格，不同数量的生产者和消费者
- 测试场景四（单队列和多队列）：相同Exchange、生产者数量、消费者数量、实例规格，不同的队列数量
- 测试场景五（队列类型）：相同Exchange、生产者数量、消费者数量、实例规格，不同的队列类型
- 测试场景六（Fanout Exchange）：相同实例规格、Fanout Exchange、队列数量，不同生产者数量、消费者数量

📖 说明

由于网络环境的影响和客户端的不同，测试结果可能会有差异。

测试环境

进行性能测试前，您需要先构建如下的测试环境：

1. 购买如表11-1所示RabbitMQ 3.8.35版本的集群实例，购买方法请参考[购买RabbitMQ实例](#)。

📖 说明

- 设置连接RabbitMQ密码时，建议不要使用特殊字符。如果使用特殊字符，在使用测试脚本时，需要对特殊字符进行转义处理，否则会报错。
- 购买“rabbitmq-2u4g”实例时，请开启公网访问，并在安全组中入方向规则中放通15672端口，以便在浏览器中访问WebUI界面。

表 11-1 实例参数

名称	代理数量	规格	是否开启SSL	磁盘类型
rabbitmq-ssl	3	rabbitmq.2u4g.cl uster	是	超高I/O
rabbitmq-2u 4g	3	rabbitmq.2u4g.cl uster	否	超高I/O
rabbitmq-4u 8g	3	rabbitmq.4u8g.cl uster	否	超高I/O
rabbitmq-8u 16g	3	rabbitmq.8u16g.c luster	否	超高I/O
rabbitmq-16 u32g	3	rabbitmq.16u32g. cluster	否	超高I/O

购买完成后，在实例详情页获取RabbitMQ实例的内网连接地址，并记录购买实例时设置的用户名和密码。如果是“rabbitmq-2u4g”实例，除了内网连接地址、用户名和密码外，还需要记录Web界面UI地址，此地址在后续登录WebUI界面设置镜像队列和惰性队列时需要使用。

连接信息

SSL 关闭 当前实例暂不支持动态开启/关闭 SSL

用户名 test [重置密码](#)

内网连接地址 IPv4 192.168.0.222:5672 [复制](#)

Web界面UI地址 http://192.168.0.222:15672 [复制](#)

- 在“rabbitmq-2u4g”实例中，[登录WebUI](#)，并设置[镜像队列](#)、[惰性队列](#)和[仲裁队列](#)。
- 在“rabbitmq-2u4g”实例的“/”Vhost下，创建“fanout”类型的Exchange，具体步骤请参考[创建RabbitMQ Exchange](#)。
- 获取测试工具[rabbitmq-perf-test-2.18.0-bin.tar.gz](#)。
- 购买客户端服务器。

购买区域、可用区、虚拟私有云、子网、安全组与RabbitMQ实例保持一致，规格为16U32G，Linux系统的ECS服务器，具体步骤请参考[购买弹性云服务器](#)。

购买完成ECS后，需要在ECS中完成以下配置：

- 安装[Java JDK](#)，并配置JAVA_HOME与PATH环境变量。

```
export JAVA_HOME=/root/jdk1.8.0_231
export PATH=$JAVA_HOME/bin:$PATH
```
- 下载[rabbitmq-perf-test-2.18.0-bin.tar.gz](#)，并解压。

```
tar -zxvf rabbitmq-perf-test-2.18.0-bin.tar.gz
```

测试脚本

须知

- 测试脚本自动创建的Exchange类型为“direct”，队列特性为非持久化、自动删除，在测试fanout类型Exchange和仲裁队列时，需要在脚本后增加“--predeclared”，表示使用自己定义的Exchange和队列参数。
- 测试开启SSL的实例时，需要将“amqp://”修改为“amqps://”，表示以加密形式传输数据。

单队列测试脚本：

```
./runjava com.rabbitmq.perf.PerfTest -h amqp://${实例用户名}:${实例密码}@${内网连接地址} -e ${Exchange名称} -s 1024 -u ${队列名称} -x ${生产者个数} -y ${消费者个数} -z ${运行时间}
```

参数说明如下：

- 实例用户名：[购买实例](#)时设置的用户名。
- 实例密码：[购买实例](#)时设置的密码。
- 内网连接地址：[购买实例](#)后获取的内网连接地址。
- 队列名称：队列的名称。
- 生产者个数：生产者的数量。
- 消费者个数：消费者的数量。
- 运行时间：脚本的运行时间，单位为秒。

多队列测试脚本：

```
./runjava com.rabbitmq.perf.PerfTest -h amqp://${实例用户名}:${实例密码}@${内网连接地址} -e ${Exchange名称} -s 1024 -x ${生产者个数} -y ${消费者个数} -z ${运行时间} --queue-pattern 'queue-%d' --queue-pattern-from 1 --queue-pattern-to x
```

参数说明如下：

- 实例用户名：[购买实例](#)时设置的用户名。
- 实例密码：[购买实例](#)时设置的密码。
- 内网连接地址：[购买实例](#)后获取的内网连接地址。
- 生产者个数：生产者的数量。
- 消费者个数：消费者的数量。
- 运行时间：脚本的运行时间，单位为秒。
- queue-%d：表示多个队列，队列名称前缀为queue-，%d表示变量，取值为从--queue-pattern-from数值到--queue-pattern-to数值的连续整数。例如--queue-pattern 'queue-%d' --queue-pattern-from 1 --queue-pattern-to 3，表示3个队列，队列名称为queue-1、queue-2、queue-3。

测试步骤

步骤1 登录客户端服务器，进入“rabbitmq-perf-test-2.18.0/bin”目录下。

步骤2 运行以下脚本，测试并记录不同实例规格下的生产速率和消费速率。

```
./runjava com.rabbitmq.perf.PerfTest -h amqp://${实例用户名}:${实例密码}@${内网连接地址} -e ${Exchange名称} -s 1024 -u ${队列名称} -x 3 -y 3 -z 300
```

示例如下：

```
./runjava com.rabbitmq.perf.PerfTest -h amqp://test:*****@192.168.0.150:5672 -e exchange-direct -s 1024 -u queue-1 -x 3 -y 3 -z 300
```

步骤3 运行以下脚本，测试并记录rabbitmq-ssl实例在不同生产者数量、消费者数量、队列数量的生产速率和消费速率。

单队列测试脚本：

```
./runjava com.rabbitmq.perf.PerfTest -h amqps://${实例用户名}:${实例密码}@${内网连接地址} -e ${Exchange名称} -s 1024 -u ${队列名称} -x ${生产者个数} -y ${消费者个数} -z 300
```

示例如下：

```
./runjava com.rabbitmq.perf.PerfTest -h amqps://test:*****@192.168.0.150:5671 -e exchange-direct -s 1024 -u queue-1 -x 1 -y 1 -z 300
```

多队列测试脚本：

```
./runjava com.rabbitmq.perf.PerfTest -h amqps://${实例用户名}:${实例密码}@${内网连接地址} -e ${Exchange名称} -s 1024 -x 3 -y 3 -z 300 --queue-pattern 'queue-%d' --queue-pattern-from 1 --queue-pattern-to x
```

示例如下：

```
./runjava com.rabbitmq.perf.PerfTest -h amqps://test:*****@192.168.0.150:5671 -e exchange-direct -s 1024 -x 3 -y 3 -z 300 --queue-pattern 'queue-%d' --queue-pattern-from 1 --queue-pattern-to 3
```

步骤4 运行以下脚本，测试并记录rabbitmq-2u4g实例在不同生产者数量、消费者数量、队列数量的生产速率和消费速率。

单队列测试脚本：

```
./runjava com.rabbitmq.perf.PerfTest -h amqp://${实例用户名}:${实例密码}@${内网连接地址} -e ${Exchange名称} -s 1024 -u ${队列名称} -x ${生产者个数} -y ${消费者个数} -z 300
```

示例如下：

```
./runjava com.rabbitmq.perf.PerfTest -h amqp://test:*****@192.168.0.150:5672 -e exchange-direct -s 1024 -u queue-1 -x 1 -y 1 -z 300
```

多队列测试脚本：

```
./runjava com.rabbitmq.perf.PerfTest -h amqp://${实例用户名}:${实例密码}@${内网连接地址} -e ${Exchange名称} -s 1024 -x ${生产者个数} -y ${消费者个数} -z 300 --queue-pattern 'queue-%d' --queue-pattern-from 1 --queue-pattern-to x
```

示例如下：

```
./runjava com.rabbitmq.perf.PerfTest -h amqp://test:*****@192.168.0.150:5672 -e exchange-direct -s 1024 -x 3 -y 3 -z 300 --queue-pattern 'queue-%d' --queue-pattern-from 1 --queue-pattern-to 3
```

步骤5 运行以下脚本，测试并记录rabbitmq-2u4g实例在不同生产者数量、消费者数量、队列数量、队列类型的生产速率和消费速率，其中队列类型分别为“惰性”、“镜像”和“仲裁”。

单队列测试脚本（不包括仲裁队列）：

```
./runjava com.rabbitmq.perf.PerfTest -h amqp://${实例用户名}:${实例密码}@${内网连接地址} -e ${Exchange名称} -s 1024 -u ${队列名称} -x ${生产者个数} -y ${消费者个数} -z 300
```

示例如下：

```
./runjava com.rabbitmq.perf.PerfTest -h amqp://test:*****@192.168.0.150:5672 -e exchange-direct -s 1024 -u queue-1 -x 1 -y 1 -z 300
```

多队列测试脚本（不包括仲裁队列）：

```
./runjava com.rabbitmq.perf.PerfTest -h amqp://${实例用户名}:${实例密码}@${内网连接地址} -e ${Exchange名称} -s 1024 -x ${生产者个数} -y ${消费者个数} -z 300 --queue-pattern 'queue-%d' --queue-pattern-from 1 --queue-pattern-to x
```

示例如下：

```
./runjava com.rabbitmq.perf.PerfTest -h amqp://test:*****@192.168.0.150:5672 -e exchange-direct -s 1024 -x 3 -y 3 -z 300 --queue-pattern 'queue-%d' --queue-pattern-from 1 --queue-pattern-to 3
```

单队列测试脚本（仲裁队列）：

```
./runjava com.rabbitmq.perf.PerfTest -h amqp://${实例用户名}:${实例密码}@${内网连接地址} -e ${Exchange名称} -s 1024 -u ${队列名称} -x ${生产者个数} -y ${消费者个数} -z 300 --predeclared
```

示例如下：

```
./runjava com.rabbitmq.perf.PerfTest -h amqp://test:*****@192.168.0.150:5672 -e exchange-direct -s 1024 -u queue-1 -x 1 -y 1 -z 300 --predeclared
```

多队列测试脚本（仲裁队列）：

```
./runjava com.rabbitmq.perf.PerfTest -h amqp://${实例用户名}:${实例密码}@${内网连接地址} -e ${Exchange名称} -s 1024 -x ${生产者个数} -y ${消费者个数} -z 300 --queue-pattern 'queue-%d' --queue-pattern-from 1 --queue-pattern-to x --predeclared
```

示例如下：

```
./runjava com.rabbitmq.perf.PerfTest -h amqp://test:*****@192.168.0.150:5672 -e exchange-direct -s 1024 -x 3 -y 3 -z 300 --queue-pattern 'queue-%d' --queue-pattern-from 1 --queue-pattern-to 3 --predeclared
```

步骤6 运行以下脚本，测试并记录rabbitmq-2u4g实例的Fanout Exchange在不同消费者数量、队列数量的生产速率和消费速率。

单队列测试脚本：

```
./runjava com.rabbitmq.perf.PerfTest -h amqp://${实例用户名}:${实例密码}@${内网连接地址} -e ${Exchange名称} -s 1024 -u ${队列名称} -x ${生产者个数} -y ${消费者个数} -z 300 --predeclared
```

示例如下：

```
./runjava com.rabbitmq.perf.PerfTest -h amqp://test:*****@192.168.0.150:5672 -e exchange-fanout -s 1024 -u queue-1 -x 1 -y 3 -z 300 --predeclared
```

----结束

测试结果

测试场景一（实例规格）：相同Exchange、队列、生产者数量、消费者数量、不同的实例规格

测试参数如下：

- Exchange：类型为“direct”，非持久化，不会自动删除。
- 队列：类型“经典队列”，数量为“3”，非持久化，会自动删除。
- 生产者：数量为“3”。
- 消费者：数量为“3”。

表 11-2 测试结果

实例规格	磁盘类型	代理数量	生产速率	消费速率
rabbitmq.2u4g.cluster	超高I/O	3	32052	25219
rabbitmq.4u8g.cluster	超高I/O	3	53774	47370

实例规格	磁盘类型	代理数量	生产速率	消费速率
rabbitmq.8u16g.cluster	超高I/O	3	54727	45730
rabbitmq.16u32g.cluster	超高I/O	3	66896	51061

通过上表的测试结果，得出以下结论，仅供参考：实例规格越大，实例性能越高。

测试场景二（是否开启SSL）：相同Exchange、队列、生产者数量、消费者数量、实例规格，开启SSL和未开启SSL的实例

测试参数如下：

- 实例规格：rabbitmq.2u4g.cluster*3。
- Exchange：类型为“direct”，非持久化，不会自动删除。
- 队列：类型“经典队列”，非持久化，会自动删除。

表 11-3 测试结果

是否开启SSL	生产者数量	消费者数量	队列数量	生产速率	消费速率
是	1	1	1	7631	6291
否	1	1	1	17104	17091
是	3	3	1	21819	21819
否	3	3	1	26050	26050
是	6	6	1	19617	19617
否	6	6	1	24720	24720
是	1	3	1	13909	13909
否	1	3	1	18732	18732
是	1	6	1	20006	20006
否	1	6	1	20371	20371
是	3	1	1	7537	5472
否	3	1	1	28743	28743
是	6	1	1	7813	5310
否	6	1	1	26663	26662
是	3	3	3	32052	25219
否	3	3	3	39951	37790

是否开启SSL	生产者数量	消费者数量	队列数量	生产速率	消费速率
是	3	3	6	32972	26440
否	3	3	6	38686	37464
是	3	3	10	31778	25375
否	3	3	10	39809	37912

通过上表的测试结果，得出以下结论，仅供参考：使用SSL连接RabbitMQ实例可以提供更高的安全性，但是生产速率和消费速率明显下降。建议使用更高规格的资源以弥补SSL加密解密带来的性能开销。合理的增加生产者数量、消费者数量和队列数，可以缓解SSL连接带来的性能下降。

测试场景三（生产者/消费者数量）：相同Exchange、队列、实例规格，不同数量的生产者和消费者

测试参数如下：

- 实例规格：rabbitmq.2u4g.cluster*3。
- Exchange：类型为“direct”，非持久化，不会自动删除。
- 队列：类型“经典队列”，数量“1”，非持久化，会自动删除。

表 11-4 测试结果

生产者数量	消费者数量	生产速率	消费速率
1	1	17104	17091
3	3	26050	26050
6	6	24720	24720
1	3	18732	18732
1	6	20371	20371
3	1	28743	28743
6	1	26663	26662

通过上表的测试结果，得出以下结论，仅供参考：适量增加生产者数量和消费者数量，可以提高生产速率和消费速率。在本测试中，1个消费者时，3个生产者的生产速率最高，当生产者增加到6个，生产速率略微下降，这是由于资源竞争和内存高水位会导致性能瓶颈。

测试场景四（单队列和多队列）：相同Exchange、生产者数量、消费者数量、实例规格，不同的队列数量

测试参数如下：

- 实例规格：rabbitmq.2u4g.cluster*3。
- Exchange：类型为“direct”，非持久化，不会自动删除。
- 队列：类型“经典队列”，非持久化，会自动删除。

表 11-5 测试结果

生产者数量	消费者数量	队列数量	生产速率	消费速率
3	3	1	26050	26050
3	3	3	39951	37790
3	3	6	38686	37464
3	3	10	39809	37912
6	6	1	24720	24720
6	6	10	54768	45847

通过上表的测试结果，得出以下结论，仅供参考：在生产者数量和消费者数量相同时，多队列的生产效率和消费效率高于单队列。这是由于队列之间的负载均衡减少了单个队列的压力，提高了消息处理效率。

测试场景五（队列类型）：相同Exchange、生产者数量、消费者数量、实例规格，不同的队列类型

测试参数如下：

- 实例规格：rabbitmq.2u4g.cluster*3。
- Exchange：类型为“direct”，非持久化，不会自动删除。
- 队列：经典队列、惰性队列和镜像队列设置为非持久化，会自动删除。仲裁队列设置为持久化、不会自动删除。

表 11-6 测试结果

生产者数量	消费者数量	队列数量	队列类型	生产速率	消费速率
3	3	1	经典	26050	26050
3	3	1	惰性	12291	12291
3	3	1	镜像	16513	16100
3	3	1	仲裁	6771	15568
3	3	10	经典	39809	37912
3	3	10	惰性	30543	29991
3	3	10	镜像	11130	8220
3	3	10	仲裁	52466	51772

生产者数量	消费者数量	队列数量	队列类型	生产速率	消费速率
6	6	1	经典	24720	24720
6	6	1	惰性	9934	8728
6	6	1	镜像	16000	14888
6	6	1	仲裁	6318	26841
6	6	10	经典	54768	45847
6	6	10	惰性	42556	33994
6	6	10	镜像	54625	45853
6	6	10	仲裁	47133	41011

通过上表的测试结果，得出以下结论，仅供参考：

- 惰性队列：惰性队列性能低于经典队列，增加队列数后，性能明显提高。
- 镜像队列：镜像队列的生产速率低于经典队列，因为在每次发布消息时，镜像队列需要将消息复制到所有镜像节点，过程中增加了网络传输和节点处理的负担。
- 仲裁队列：单队列时，消费速率高于生产速率，可能是因为仲裁队列需要在每次写入时进行一致性检查，从而影响了生产速率。多队列时，相比于单队列生产速率和消费速率明显提高，适合高一致性和高并发场景。

测试场景六（Fanout Exchange）：相同实例规格、Fanout Exchange、队列数量，不同生产者数量、消费者数量

测试参数如下：

- 实例规格：rabbitmq.2u4g.cluster*3。
- Exchange：类型为“fanout”，非持久化，不会自动删除。
- 队列：类型“经典队列”，数量“1”，非持久化，会自动删除。

表 11-7 测试结果

生产者数量	消费者数量	生产速率	消费速率
1	1	17233	17232
3	3	35539	35231
6	6	32635	31682
1	3	27864	27864
1	6	31518	31518

通过上表的测试结果，得出以下结论，仅供参考：适量增加生产者数量和消费者数量，可以提高生产速率和消费速率。

12 申请扩大 RabbitMQ 配额

什么是配额？

为防止资源滥用，平台限定了各服务资源的配额，对用户的资源数量和容量做了限制。如您最多可以创建多少个 RabbitMQ 实例。

如果当前资源配额限制无法满足使用需要，您可以申请扩大配额。

怎样查看我的配额？


1. 登录管理控制台。
2. 单击管理控制台左上角的 ，选择区域和项目。
3. 在页面右上角，选择“资源 > 我的配额”。
系统进入“服务配额”页面。

图 12-1 我的配额



4. 您可以在“服务配额”页面，查看各项资源的总配额及使用情况。
如果当前配额不能满足业务要求，请参考后续操作，申请扩大配额。

如何申请扩大配额？

1. 登录管理控制台。
2. 在页面右上角，选择“资源 > 我的配额”。
系统进入“服务配额”页面。

3. 单击“申请扩大配额”。
4. 在“新建工单”页面，根据您的需求，填写相关参数。
其中，“问题描述”项请填写需要调整的内容和申请原因。
5. 填写完毕后，勾选协议并单击“提交”。

13 查看监控指标与配置告警

13.1 查看 RabbitMQ 监控数据


云监控对分布式消息服务RabbitMQ版的运行状态进行日常监控，可以通过控制台直观的查看分布式消息服务RabbitMQ版各项监控指标。

前提条件

已创建RabbitMQ实例，且实例中有可消费的消息。


查看 RabbitMQ 监控数据

步骤1 登录管理控制台。

步骤2 在管理控制台左上角单击 ，选择区域。

说明

此处请选择RabbitMQ实例所在的区域。

步骤3 在管理控制台左上角单击 ，选择“应用中间件 > 分布式消息服务RabbitMQ版”，进入分布式消息服务RabbitMQ专享版页面。

步骤4 通过以下任意一种方法，查看监控数据。

- 在RabbitMQ实例名称后，单击“查看监控数据”。跳转到云监控页面，查看实例、节点和队列的监控数据，数据更新周期为1分钟。
- 单击RabbitMQ实例名称，进入实例详情页。在左侧导航栏单击“监控与告警 > 监控”，进入监控页面，查看实例、节点和队列的监控数据，数据更新周期为1分钟。

RabbitMQ 3.x.x版本实例在监控页面中，队列名称存在两种显示。如果队列在默认Vhost下，会直接显示队列名。如果队列不在默认Vhost下，队列名称显示为：队列所在的Vhost名称_队列名。例如：队列test01在Vhost-13142708中，此时监控中显示的队列为“Vhost-13142708_test01”。

图 13-1 队列的监控页面



----结束

13.2 RabbitMQ 支持的监控指标

功能说明

本章节定义了分布式消息服务RabbitMQ版上报云监控服务的监控指标的命名空间，监控指标列表和维度定义，用户可以通过云监控服务提供的管理控制台或API接口来检索RabbitMQ实例产生的监控指标和告警信息，也可以通过分布式消息服务RabbitMQ版控制台提供的“监控”页面来检索RabbitMQ实例产生的监控指标。

命名空间

SYS.DMS

实例监控指标

表 13-1 实例支持的监控指标（RabbitMQ 3.x.x 版本）

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期（原始指标）
connections	连接数	该指标用于统计RabbitMQ实例中的总连接数。 单位：个	≥ 0	RabbitMQ实例	1分钟
channels	通道数	该指标用于统计RabbitMQ实例中的总通道数。 单位：个	0~2047	RabbitMQ实例	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
queues	队列数	该指标用于统计 RabbitMQ 实例中的总队列数。 单位: 个	0~7000	RabbitMQ 实例	1分钟
consumers	消费者数	该指标用于统计 RabbitMQ 实例中的总消费者数。 单位: 个	0~28000	RabbitMQ 实例	1分钟
messages_ready	可消费消息数	该指标用于统计 RabbitMQ 实例中总可消费消息数量。 单位: 个	0~100000	RabbitMQ 实例	1分钟
messages_unacknowledged	未确认消息数	该指标用于统计 RabbitMQ 实例中总已经消费但还未确认的消息数量。 单位: 个	0~100000	RabbitMQ 实例	1分钟
publish	生产速率	统计 RabbitMQ 实例中实时消息生产速率。 单位: 个/秒	0~25000	RabbitMQ 实例	1分钟
deliver	消费速率(手工确认)	统计 RabbitMQ 实例中实时消息消费速率(手工确认)。 单位: 个/秒	0~25000	RabbitMQ 实例	1分钟
deliver_no_ack	消费速率(自动确认)	统计 RabbitMQ 实例中实时消息消费速率(自动确认)。 单位: 个/秒	0~50000	RabbitMQ 实例	1分钟
connections_states_running	运行状态的连接个数	该指标用于统计整个实例中的 connection, 状态是 starting/tuning/opening/running 状态的总数。 单位: 个 说明 2022年5月16号及以后购买的实例, 支持此监控项。	>= 0	RabbitMQ 实例	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
connections_states_flow	flow状态的连接数	该指标用于统计整个实例中的connection，状态是flow状态的总数。 单位：个 说明 2022年5月16号及以后购买的实例，支持此监控项。	>= 0	RabbitMQ实例	1分钟
connections_states_block	block状态的连接数	该指标用于统计整个实例中的connection，状态是blocking/blocked状态的总数。 单位：个 说明 2022年5月16号及以后购买的实例，支持此监控项。	>= 0	RabbitMQ实例	1分钟
connections_states_close	close状态的连接数	该指标用于统计整个实例中的connection，状态是closing/closed状态的总数。 单位：个 说明 2022年5月16号及以后购买的实例，支持此监控项。	>= 0	RabbitMQ实例	1分钟
channels_states_running	运行状态的通道数	该指标用于统计整个实例中的channel，状态是starting/tuning/opening/running状态的总数。 单位：个 说明 2022年5月16号及以后购买的实例，支持此监控项。	>= 0	RabbitMQ实例	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
channels_states_flow	flow状态的通道数	该指标用于统计整个实例中的channel，状态是flow状态的总数。 单位：个 说明 2022年5月16号及以后购买的实例，支持此监控项。	>= 0	RabbitMQ实例	1分钟
channels_states_block	block状态的通道数	该指标用于统计整个实例中的channel，状态是blocking/blocked状态的总数。 单位：个 说明 2022年5月16号及以后购买的实例，支持此监控项。	>= 0	RabbitMQ实例	1分钟
channels_states_close	close状态的通道数	该指标用于统计整个实例中的channel，状态是closing/closed状态的总数。 单位：个 说明 2022年5月16号及以后购买的实例，支持此监控项。	>= 0	RabbitMQ实例	1分钟
queues_states_running	运行状态的队列数	该指标用于统计整个实例中的queue，状态是running状态的总数。 单位：个 说明 2022年5月16号及以后购买的实例，支持此监控项。	>= 0	RabbitMQ实例	1分钟
queues_states_flow	flow状态的队列数	该指标用于统计整个实例中的queue，状态是flow状态的总数。 单位：个 说明 2022年5月16号及以后购买的实例，支持此监控项。	>= 0	RabbitMQ实例	1分钟

表 13-2 实例支持的监控指标（RabbitMQ AMQP-0-9-1 版本）

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
connections	连接数	该指标用于统计 RabbitMQ 实例中的总连接数。 单位：个	≥ 0	RabbitMQ 实例	1分钟
channels	通道数	该指标用于统计 RabbitMQ 实例中的总通道数。 单位：个	0~2000	RabbitMQ 实例	1分钟
queues	队列数	该指标用于统计 RabbitMQ 实例中的总队列数。 单位：个	0~1000	RabbitMQ 实例	1分钟
consumers	消费者数	该指标用于统计 RabbitMQ 实例中的总消费者数。 单位：个	≥ 0	RabbitMQ 实例	1分钟
message_s_ready	可消费消息数	该指标用于统计 RabbitMQ 实例中总可消费消息数量。 单位：个	0~1000000	RabbitMQ 实例	1分钟
publish	生产速率	统计 RabbitMQ 实例中实时消息生产速率。 单位：个/秒	≥ 0	RabbitMQ 实例	1分钟
instance_bytes_in_rate	生产流量	统计 RabbitMQ 实例中每秒生产的字节数。 单位：Byte/s、KB/s、MB/s、GB/s	≥ 0	RabbitMQ 实例	1分钟
instance_bytes_out_rate	消费流量	统计 RabbitMQ 实例中每秒消费的字节数。 单位：Byte/s、KB/s、MB/s、GB/s	≥ 0	RabbitMQ 实例	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
deliver_get	消费速率	统计RabbitMQ实例中实时消息消费速率。 单位: 个/秒	>= 0	RabbitMQ实例	1分钟
instance_disk_usage	实例磁盘容量使用率	实例磁盘容量使用率。 单位: %	>= 0	RabbitMQ实例	1分钟
instance_tps	实例每秒请求个数	统计RabbitMQ实例每秒处理的请求个数。 单位: 个	0~1000000	RabbitMQ实例	1分钟

节点监控指标

仅RabbitMQ 3.x.x版本支持节点监控指标。

表 13-3 节点支持的监控指标

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
fd_used	文件句柄数	该指标用于统计当前节点RabbitMQ所占用的文件句柄数。 单位: 个	0~65535	RabbitMQ实例节点	1分钟
socket_used	Socket连接数	该指标用于统计当前节点RabbitMQ所使用的Socket连接数。 单位: 个	0~50000	RabbitMQ实例节点	1分钟
proc_used	Erlang进程数	该指标用于统计当前节点RabbitMQ所使用的Erlang进程数。 单位: 个	0~1048576	RabbitMQ实例节点	1分钟
mem_used	内存占用	该指标用于统计当前节点RabbitMQ内存占用。 单位: Byte、KB、MB、GB、TB、PB	0~3200000000	RabbitMQ实例节点	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
disk_free	可用存储空间	该指标用于统计当前节点可使用的存储空间。 单位: Byte、KB、MB、GB、TB、PB	0~5000000000	RabbitMQ实例节点	1分钟
rabbitmq_alive	节点存活状态	表示Rabbitmq节点是否存活。 说明 2020年4月及以后购买的实例, 支持此监控项。	1: 存活 0: 离线	RabbitMQ实例节点	1分钟
rabbitmq_disk_usage	磁盘容量使用率	统计Rabbitmq节点虚拟机的磁盘容量使用率。 单位: % 说明 2020年4月及以后购买的实例, 支持此监控项。	0~100%	RabbitMQ实例节点	1分钟
rabbitmq_cpu_usage	CPU使用率	统计Rabbitmq节点虚拟机的CPU使用率。 单位: % 说明 2020年4月及以后购买的实例, 支持此监控项。	0~100%	RabbitMQ实例节点	1分钟
rabbitmq_cpu_core_load	CPU核均负载	统计Rabbitmq节点虚拟机CPU每个核的平均负载。 说明 2020年4月及以后购买的实例, 支持此监控项。	>0	RabbitMQ实例节点	1分钟
rabbitmq_memory_usage	内存使用率	统计Rabbitmq节点虚拟机的内存使用率。 单位: % 说明 2020年4月及以后购买的实例, 支持此监控项。	0~100%	RabbitMQ实例节点	1分钟
rabbitmq_disk_read_await	磁盘平均读操作耗时	该指标用于统计磁盘在测量周期内平均每个读IO的操作时长。 单位: ms 说明 2020年6月及以后购买的实例, 支持此监控项。	>0	RabbitMQ实例节点	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
rabbitmq_disk_writes_await	磁盘平均写操作耗时	该指标用于统计磁盘在测量周期内平均每个写IO的操作时长。 单位: ms 说明 2020年6月及以后购买的实例, 支持此监控项。	>0	RabbitMQ实例节点	1分钟
rabbitmq_node_bytes_in_rate	网络入流量	统计Rabbitmq节点每秒网络访问流入流量。 单位: Byte/s、KB/s、MB/s、GB/s 说明 2020年6月及以后购买的实例, 支持此监控项。	>0	RabbitMQ实例节点	1分钟
rabbitmq_node_bytes_out_rate	网络出流量	统计Rabbitmq节点每秒网络访问流出流量。 单位: Byte/s、KB/s、MB/s、GB/s 说明 2020年6月及以后购买的实例, 支持此监控项。	>0	RabbitMQ实例节点	1分钟
rabbitmq_node_queues	节点队列数	该指标用于统计Rabbitmq节点队列个数。 单位: 个 说明 2020年6月及以后购买的实例, 支持此监控项。	>0	RabbitMQ实例节点	1分钟
rabbitmq_memory_high_watermark	内存高水位状态	表示Rabbitmq节点是否触发内存高水位, 如果触发, 会阻塞集群的所有生产者。 说明 2020年6月及以后购买的实例, 支持此监控项。	1: 触发 0: 没有触发	RabbitMQ实例节点	1分钟
rabbitmq_disk_insufficient	磁盘高水位状态	表示Rabbitmq节点是否触发磁盘高水位, 如果触发, 会阻塞集群的所有生产者。 说明 2020年6月及以后购买的实例, 支持此监控项。	1: 触发 0: 没有触发	RabbitMQ实例节点	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
rabbitmq_disk_read_rate	磁盘读流量	统计节点磁盘每秒的读字节大小。 单位: Byte/s、KB/s、MB/s、GB/s 说明 2022年5月16号及以后购买的实例, 支持此监控项。	>= 0	RabbitMQ 实例节点	1分钟
rabbitmq_disk_write_rate	磁盘写流量	统计节点磁盘每秒的写字节大小。 单位: Byte/s、KB/s、MB/s、GB/s 说明 2022年5月16号及以后购买的实例, 支持此监控项。	>= 0	RabbitMQ 实例节点	1分钟
connections_usage	连接数使用率	当前节点实际连接数占最大连接数比率。 单位: % 说明 2024年2月18号及以后购买的实例, 支持此监控项。	>=0	RabbitMQ 实例节点	1分钟

队列监控指标

表 13-4 队列支持的监控指标 (RabbitMQ 3.x.x 版本)

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
queue_messages_unacknowledged	队列未确认消息数	该指标用于统计队列中已消费未确认消息数。 单位: 个	0~1000000	RabbitMQ 实例队列	1分钟
queue_messages_ready	队列可消费消息数	该指标用于统计队列中可消费的消息数。 单位: 个	0~1000000	RabbitMQ 实例队列	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
queue_consumer_s	消费者数量	该指标用于统计订阅该队列的消费者个数。 单位：个 说明 2022年5月16号及以后购买的实例，支持此监控项。	>= 0	RabbitMQ 实例队列	1分钟
queue_messages_publish_rate	生产速率	该指标用于统计每秒该队列的消息流入数。 单位：个/s 说明 2022年5月16号及以后购买的实例，支持此监控项。	>= 0	RabbitMQ 实例队列	1分钟
queue_messages_ack_rate	消费速率（手工确认）	该指标用于统计该队列每秒传递给客户端并确认的消息数。 单位：个/s 说明 2022年5月16号及以后购买的实例，支持此监控项。	>= 0	RabbitMQ 实例队列	1分钟
queue_messages_deliver_get_rate	消费速率	该指标用于统计该队列每秒的消息流出数。 单位：个/s 说明 2022年5月16号及以后购买的实例，支持此监控项。	>= 0	RabbitMQ 实例队列	1分钟
queue_messages_redeliver_rate	重传速率	该指标用于统计该队列每秒的重传消息数。 单位：个/s 说明 2022年5月16号及以后购买的实例，支持此监控项。	>= 0	RabbitMQ 实例队列	1分钟
queue_messages_persistent	消息总数（持久化）	该指标用来统计该队列中持久消息的总数（对于瞬时队列始终为0）。 单位：个 说明 2022年5月16号及以后购买的实例，支持此监控项。	>= 0	RabbitMQ 实例队列	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
queue_messages_ram	消息总数(内存)	该指标用于统计该队列中驻留在内存中的消息总数。 单位: 个 说明 2022年5月16号及以后购买的实例, 支持此监控项。	>= 0	RabbitMQ 实例队列	1分钟
queue_memory	Erlang 进程消耗字节数	该指标用于统计与队列关联的Erlang进程消耗的内存字节数, 包括堆栈、堆和内部结构。 单位: Byte、KB、MB、GB、TB、PB 说明 2022年5月16号及以后购买的实例, 支持此监控项。	>= 0	RabbitMQ 实例队列	1分钟
queue_message_bytes	消息大小总和	该指标用于统计该队列中所有消息的大小总和(字节)。 单位: Byte、KB、MB、GB、TB、PB 说明 2022年5月16号及以后购买的实例, 支持此监控项。	>= 0	RabbitMQ 实例队列	1分钟

表 13-5 Queue 支持的监控指标 (RabbitMQ AMQP-0-9-1 版本)

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
queue_messages_ready	队列可消费消息数	该指标用于统计队列中可消费的消息数。 单位: 个	>= 0	RabbitMQ 实例Queue	1分钟
queue_consumers	消费者数量	该指标用于统计订阅该队列的消费者个数。 单位: 个	>= 0	RabbitMQ 实例Queue	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
queue_messages_publish_rate	生产速率	该指标用于统计该队列每秒的消息流入数。 单位: 个/s	>= 0	RabbitMQ 实例Queue	1分钟
queue_messages_deliver_get_rate	消费速率	该指标用于统计该队列每秒的消息流出数。 单位: 个/s	>= 0	RabbitMQ 实例Queue	1分钟
queue_bytes_in_rate	生产流量	该指标用于统计该队列实时消息生产速率。 单位: Byte/s、KB/s、MB/s、GB/s	>= 0	RabbitMQ 实例Queue	1分钟
queue_bytes_out_rate	消费流量	该指标用于统计该队列实时消息消费速率。 单位: Byte/s、KB/s、MB/s、GB/s	>= 0	RabbitMQ 实例Queue	1分钟

Vhost 监控指标

仅RabbitMQ AMQP-0-9-1版本支持Vhost监控指标。

表 13-6 Vhost 支持的监控指标

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
vhost_connections	连接数	统计该Vhost中的总连接数。 单位: 个	>= 0	RabbitMQ 实例Vhost	1分钟
vhost_channels	通道数	统计该Vhost中的总通道数。 单位: 个	>= 0	RabbitMQ 实例Vhost	1分钟
vhost_queues	队列数	统计该Vhost中的总队列数。 单位: 个	>= 0	RabbitMQ 实例Vhost	1分钟
vhost_consumers	消费者数	统计该Vhost中的总消费者数。 单位: 个	>= 0	RabbitMQ 实例Vhost	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
vhost_messages_ready	可消费消息数	统计该Vhost中总可消费消息数量。 单位: 个	≥ 0	RabbitMQ实例Vhost	1分钟
vhost_messages_publish_rate	生产速率	统计该Vhost中实时消息生产速率。 单位: 个/秒	≥ 0	RabbitMQ实例Vhost	1分钟
vhost_messages_deliver_get_rate	消费速率	统计该Vhost中实时消息消费速率。 单位: 个/秒	≥ 0	RabbitMQ实例Vhost	1分钟
vhost_bytes_in_rate	生产流量	统计该Vhost中每秒生产的字节数。 单位: Byte/s、KB/s、MB/s、GB/s	≥ 0	RabbitMQ实例Vhost	1分钟
vhost_bytes_out_rate	消费流量	统计该Vhost中每秒消费的字节数。 单位: Byte/s、KB/s、MB/s、GB/s	≥ 0	RabbitMQ实例Vhost	1分钟
vhost_tps	vhost请求速率	统计当前Vhost每秒处理的请求个数。 单位: 个	≥ 0	RabbitMQ实例Vhost	1分钟

Exchange 监控指标

仅RabbitMQ AMQP-0-9-1版本支持Exchange监控指标。

表 13-7 Exchange 支持的监控指标

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
exchange_messages_publish_rate	生产速率	统计该Exchange中实时消息生产速率。 单位: 个/秒	≥ 0	RabbitMQ实例Exchange	1分钟
exchange_bytes_in_rate	生产流量	统计该Exchange中每秒生产的字节数。 单位: Byte/s、KB/s、MB/s、GB/s	≥ 0	RabbitMQ实例Exchange	1分钟

维度

Key	Value
rabbitmq_instance_id	RabbitMQ实例
rabbitmq_node	RabbitMQ实例节点
rabbitmq_queue	RabbitMQ实例队列
rabbitmq_vhost	RabbitMQ实例Vhost
rabbitmq_vhost_exchange	RabbitMQ实例Exchange
rabbitmq_vhost_queue	RabbitMQ实例Queue

13.3 配置 RabbitMQ 监控告警

本章节主要介绍部分监控指标的告警策略，以及配置操作。在实际业务中，建议按照以下告警策略，配置监控指标的告警规则。

表 13-8 RabbitMQ 实例配置告警的指标（RabbitMQ 3.x.x 版本）

指标名称	告警策略	指标说明	解决方案
内存高水位状态	告警阈值：原始值 ≥ 1 连续触发次数：1 告警级别：致命	告警阈值为1表示触发内存高水位，会阻塞消息生产	<ul style="list-style-type: none"> 加快消费 采用生产者确认的发送模式，并监控生产端消息生产速度和时长，当消息生产时长有明显增加时进行流控措施
磁盘高水位状态	告警阈值：原始值 ≥ 1 连续触发次数：1 告警级别：致命	告警阈值为1表示触发磁盘高水位，会阻塞消息生产	<ul style="list-style-type: none"> 减少惰性队列的消息堆积 减少持久化队列的消息堆积 删除队列
内存使用率	告警阈值：原始值 $>$ 业务预期使用率（推荐30%） 连续触发次数：连续3~5个周期 告警级别：重要	该指标需要分别为每个节点设置内存使用率告警，避免触发内存高水位阻塞生产	<ul style="list-style-type: none"> 加快消费 采用生产者确认的发送模式，并监控生产端消息生产速度和时长，当消息生产时长有明显增加时进行流控措施

指标名称	告警策略	指标说明	解决方案
CPU使用率	告警阈值：原始值 >业务预期使用率 (推荐70%) 连续触发次数：连续3~5个周期 告警级别：重要	该指标需要分别为每个节点设置CPU使用率告警，CPU使用率过高可能会影响生产速度	<ul style="list-style-type: none"> 减少镜像队列个数 对于集群实例，建议扩容节点个数，然后进行节点间重平衡
可消费消息数	告警阈值：原始值 >业务预期可消费消息数 连续触发次数：1 告警级别：重要	可消费消息数过多表示消息堆积	请参考 消息堆积的解决办法
未确认消息数	告警阈值：原始值 >业务预期未确认消息数 连续触发次数：1 告警级别：重要	未确认消息数过多可能会导致消息堆积	<ul style="list-style-type: none"> 检查消费者是否异常 检查消费者逻辑是否消耗时间过长
连接数	告警阈值：原始值 >业务预期连接数 连续触发次数：1 告警级别：重要	连接数突增可能是流量变大的预警	需检查业务是否正常，可参考其他告警
通道数	告警阈值：原始值 >业务预期通道数 连续触发次数：1 告警级别：重要	通道数突增可能是流量变大的预警	需检查业务是否正常，可参考其他告警
Erlang进程数	告警阈值：原始值 >业务预期进程数 连续触发次数：1 告警级别：重要	进程数突增可能是流量变大的预警	需检查业务是否正常，可参考其他告警

表 13-9 RabbitMQ 实例配置告警的指标（RabbitMQ AMQP-0-9-1 版本）

指标名称	告警策略	指标说明	解决方案
可消费消息数	告警阈值：原始值 >业务预期可消费消息数 连续触发次数：1 告警级别：重要	可消费消息数过多表示消息堆积	请参考 消息堆积的解决办法


指标名称	告警策略	指标说明	解决方案
连接数	告警阈值：原始值 >业务预期连接数 连续触发次数：1 告警级别：重要	连接数突增可能是 流量变大的预警	需检查业务是否正常， 可参考其他告警
通道数	告警阈值：原始值 >业务预期通道数 连续触发次数：1 告警级别：重要	通道数突增可能是 流量变大的预警	需检查业务是否正常， 可参考其他告警
实例磁盘容量使用率	告警阈值：原始值 >85% 连续触发次数：1 告警级别：紧急	磁盘容量使用率过 大可能是消息堆积	请参考 消息堆积的解决办法

说明

- 告警阈值请根据业务预期数设置。例如，业务预期使用率35%，则告警阈值设置35%。
- 连续触发次数和告警级别可根据业务逻辑自行调整。


配置 RabbitMQ 监控告警

步骤1 登录管理控制台。

步骤2 在管理控制台左上角单击 ，选择区域。

说明


此处请选择RabbitMQ实例所在的区域。

步骤3 在管理控制台左上角单击 ，选择“应用中间件 > 分布式消息服务RabbitMQ版”，进入分布式消息服务RabbitMQ专享版页面。

步骤4 通过以下任意一种方法，查看监控数据。

- 在RabbitMQ实例名称后，单击“查看监控数据”。跳转到云监控页面，查看实例、节点和队列的监控数据，数据更新周期为1分钟。
- 单击RabbitMQ实例名称，进入实例详情页。在左侧导航栏单击“监控与告警 > 监控”，进入监控页面，查看实例、节点和队列的监控数据，数据更新周期为1分钟。

步骤5 在实例监控指标页面中，找到需要创建告警的指标项，鼠标移动到指标区域，然后单

击指标右上角的 ，进入“创建告警规则”页面。

步骤6 在告警规则页面，设置告警信息。

创建告警规则操作，请查看[创建告警规则](#)。

1. 设置告警名称和告警的描述。
2. 设置告警策略和告警级别。
例如，在进行指标监控时，如果连续3个周期，连接数原始值超过设置的值，则产生告警，如果未及时处理，则每一天发送一次告警通知。
3. 设置“发送通知”开关。当开启时，设置告警生效时间、产生告警时通知的对象以及触发的条件。
4. 单击“立即创建”，等待创建告警规则成功。

---结束

14 查看 RabbitMQ 审计日志

通过云审计（Cloud Trace Service，CTS）服务，您可以记录与分布式消息服务 RabbitMQ 版相关的操作事件，便于日后的查询、审计和回溯。

前提条件

已开通CTS。

CTS 支持的 DMS for RabbitMQ 操作

表 14-1 CTS 支持的 DMS for RabbitMQ 操作列表

操作名称	资源类型	事件名称
删除后台任务成功	rabbitmq	deleteDMSBackendJobSuccess
删除后台任务失败	rabbitmq	deleteDMSBackendJobFailure
创建DMS实例订单成功	rabbitmq	createDMSInstanceOrderSuccess
创建DMS实例订单失败	rabbitmq	createDMSInstanceOrderFailure
修改DMS实例订单成功	rabbitmq	modifyDMSInstanceOrderSuccess
修改DMS实例订单失败	rabbitmq	modifyDMSInstanceOrderFailure
扩容实例成功	rabbitmq	extendDMSInstanceSuccess
扩容实例失败	rabbitmq	extendDMSInstanceFailure
重置DMS实例密码成功	rabbitmq	resetDMSInstancePasswordSuccess
重置DMS实例密码失败	rabbitmq	resetDMSInstancePasswordFailure
删除创建失败的DMS实例成功	rabbitmq	deleteDMSCreateFailureInstancesSuccess

操作名称	资源类型	事件名称
删除创建失败的DMS实例失败	rabbitmq	deleteDMSCreateFailureInstanceFailure
批量删除DMS实例成功	rabbitmq	batchDeleteDMSInstanceSuccess
批量删除DMS实例失败	rabbitmq	batchDeleteDMSInstanceFailure
修改DMS实例信息成功	rabbitmq	modifyDMSInstanceInfoSuccess
修改DMS实例信息失败	rabbitmq	modifyDMSInstanceInfoFailure
批量删除DMS实例任务	rabbitmq	batchDeleteDMSInstanceTask
解冻DMS实例任务成功	rabbitmq	unfreezeDMSInstanceTaskSuccess
解冻DMS实例任务失败	rabbitmq	unfreezeDMSInstanceTaskFailure
冻结DMS实例任务成功	rabbitmq	freezeDMSInstanceTaskSuccess
冻结DMS实例任务失败	rabbitmq	freezeDMSInstanceTaskFailure
删除DMS实例任务成功	rabbitmq	deleteDMSInstanceTaskSuccess
删除DMS实例任务失败	rabbitmq	deleteDMSInstanceTaskFailure
创建DMS实例任务成功	rabbitmq	createDMSInstanceTaskSuccess
创建DMS实例任务失败	rabbitmq	createDMSInstanceTaskFailure
扩容DMS实例任务成功	rabbitmq	extendDMSInstanceTaskSuccess
扩容DMS实例任务失败	rabbitmq	extendDMSInstanceTaskFailure
修改DMS实例信息任务成功	rabbitmq	modifyDMSInstanceInfoTaskSuccess
修改DMS实例信息任务失败	rabbitmq	modifyDMSInstanceInfoTaskFailure
回收站恢复实例成功	rabbitmq	out_recycleTaskSuccess
回收站恢复实例失败	rabbitmq	out_recycleTaskFailure

查看审计日志

查看DMS for RabbitMQ云审计日志，请参考[查看审计事件](#)。