

分布式消息服务 RabbitMQ

用户指南

文档版本 06
发布日期 2024-10-29



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 产品简介	1
1.1 什么是分布式消息服务 RabbitMQ	1
1.2 产品优势	1
1.3 典型应用场景	2
1.4 产品规格	4
1.5 与 Kafka、RocketMQ 的差异	6
1.6 与其他云服务的关系	8
1.7 约束与限制	8
1.8 RabbitMQ 相关概念	11
1.9 Exchange 类型介绍	12
1.10 权限管理	14
1.11 计费说明	16
2 快速入门	19
2.1 快速连接 RabbitMQ 并生产消费消息	19
3 RabbitMQ 业务使用流程	25
4 权限管理	27
4.1 创建用户并授权使用 DMS for RabbitMQ	27
5 购买 RabbitMQ 实例	31
6 配置 Vhost	36
6.1 创建 RabbitMQ Vhost	36
6.2 创建 RabbitMQ Exchange	38
6.3 绑定 RabbitMQ Exchange	40
6.4 创建 RabbitMQ Queue	42
6.5 绑定 RabbitMQ Queue	43
6.6 管理 RabbitMQ Vhost	45
6.6.1 查看 RabbitMQ Vhost 信息	45
6.6.2 删除 RabbitMQ Vhost	45
6.7 管理 RabbitMQ Exchange	47
6.7.1 解绑 RabbitMQ Exchange	47
6.7.2 删除 RabbitMQ Exchange	49
6.8 管理 RabbitMQ Queue	49

6.8.1 查看 RabbitMQ Queue 信息.....	49
6.8.2 清空 RabbitMQ Queue 消息.....	50
6.8.3 解绑 RabbitMQ Queue.....	52
6.8.4 配置 RabbitMQ 镜像队列.....	53
6.8.5 配置 RabbitMQ 惰性队列.....	55
6.8.6 配置 RabbitMQ 仲裁队列.....	56
6.8.7 配置 RabbitMQ 单一活跃消费者.....	60
6.8.8 删除 RabbitMQ Queue.....	62
7 连接实例.....	66
7.1 配置 RabbitMQ 网络连接.....	66
7.1.1 连接 RabbitMQ 网络要求.....	66
7.1.2 配置 RabbitMQ 实例的公网访问.....	67
7.2 配置 RabbitMQ 客户端心跳超时时间.....	69
7.3 使用客户端连接 RabbitMQ（关闭 SSL）.....	70
7.4 使用客户端连接 RabbitMQ（开启 SSL）.....	72
8 管理消息.....	76
8.1 配置 RabbitMQ 死信消息.....	76
8.2 配置 RabbitMQ 消息确认机制.....	77
8.3 配置 RabbitMQ 消息预取值.....	78
9 配置高级特性.....	80
9.1 配置 RabbitMQ 持久化.....	80
9.2 配置 RabbitMQ TTL.....	83
10 管理实例.....	85
10.1 查看和修改 RabbitMQ 实例基本信息.....	85
10.2 查看 RabbitMQ 客户端连接地址.....	87
10.3 配置 RabbitMQ 实例标签.....	88
10.4 重置 RabbitMQ 实例密码.....	90
10.5 开启 RabbitMQ 实例插件.....	90
10.6 使用 rabbitmq_tracing 插件.....	92
10.7 导出 RabbitMQ 实例列表.....	97
10.8 删除 RabbitMQ 实例.....	97
10.9 登录 RabbitMQ WebUI.....	98
11 变更 RabbitMQ 实例规格.....	100
12 迁移 RabbitMQ 业务.....	102
13 申请扩大 RabbitMQ 配额.....	106
14 查看监控指标与配置告警.....	107
14.1 查看 RabbitMQ 监控数据.....	107
14.2 RabbitMQ 支持的监控指标.....	108
14.3 配置 RabbitMQ 监控告警.....	110

15 查看 RabbitMQ 审计日志.....	113
16 常见问题.....	115
16.1 实例问题.....	115
16.1.1 RabbitMQ 使用的版本是多少?	115
16.1.2 RabbitMQ 实例 SSL 连接的协议版本号是多少?	115
16.1.3 创建实例时为什么无法查看子网和安全组等信息?	115
16.1.4 RabbitMQ 集群实例如何均衡分发请求到每个虚拟机?	115
16.1.5 RabbitMQ 实例集群内部的队列是否有冗余备份?	115
16.1.6 RabbitMQ 实例是否支持持久化, 如何定时备份数据?	115
16.1.7 RabbitMQ 实例开启 SSL 后, 证书怎么获取?	115
16.1.8 RabbitMQ 实例的 SSL 开关是否支持修改?	116
16.1.9 RabbitMQ 实例是否支持扩容?	116
16.1.10 RabbitMQ 支持双向认证吗?	116
16.1.11 RabbitMQ 支持升级 CPU 和内存吗?	116
16.1.12 如何关闭 RabbitMQ 的 WebUI?	116
16.1.13 实例是否支持修改可用区?	116
16.1.14 如何获取 region id?	117
16.1.15 为什么不能选择 2 个可用区?	117
16.1.16 单机 RabbitMQ 实例如何变更为集群实例?	117
16.1.17 RabbitMQ 实例创建后, 能修改 VPC 和子网吗?	117
16.2 连接问题.....	118
16.2.1 如何配置安全组?	118
16.2.2 RabbitMQ 客户端连接报错原因分析.....	120
16.2.3 RabbitMQ 实例是否支持公网访问?	121
16.2.4 RabbitMQ 是否支持跨 Region 部署?	121
16.2.5 RabbitMQ 实例是否支持跨 VPC 访问?	121
16.2.6 RabbitMQ 实例是否支持不同的子网?	121
16.2.7 SSL 方式连接 RabbitMQ 实例失败?	121
16.2.8 客户端是否可以通过 DNAT 方式访问 RabbitMQ 实例?	122
16.2.9 RabbitMQ 实例的 Web 管理页面无法打开.....	122
16.2.10 客户端是否可以连接同个 RabbitMQ 下多个 Vhost?	122
16.2.11 为什么 RabbitMQ 集群只有一个连接地址?	122
16.3 消息问题.....	123
16.3.1 RabbitMQ 实例支持延时消息队列么?	123
16.3.2 消息堆积对业务的影响及解决办法.....	123
16.3.3 消息的最长保留时间是多久?	125
16.3.4 消息创建时间在哪设置?	125
16.3.5 RabbitMQ 生产消息的最大长度是多少?	125
16.4 监控告警问题.....	125
16.4.1 云监控无法展示 RabbitMQ 监控数据.....	125
16.4.2 云监控显示通道数一直上升报警有影响吗?	125

A 修订记录.....	126
--------------------	------------

1 产品简介

1.1 什么是分布式消息服务 RabbitMQ

分布式消息服务RabbitMQ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。

- 即开即用
分布式消息服务RabbitMQ提供单机和集群的消息实例，拥有丰富内存规格，您可以通过控制台直接下单购买并创建，无需单独准备服务器资源。
- 消息特性丰富
支持AMQP协议，支持普通消息、广播消息、死信、延迟消息等特性。
- 灵活路由
在RabbitMQ中，生产者将消息发送到交换器，由交换器将消息路由到队列中。交换器支持Direct、Topic、Headers和Fanout四种路由方式，同时支持交换机组合和自定义。
- 高可用
RabbitMQ集群提供仲裁队列，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。
- 监控和告警
支持对RabbitMQ实例状态进行监控，支持对集群每个代理的内存、CPU、网络流量等进行监控。如果集群或节点状态异常，将触发告警。

1.2 产品优势

分布式消息服务RabbitMQ完全兼容开源社区版本，旨在为您提供便捷高效的消息队列。业务无需改动即可快速迁移上云，为您节省维护和使用成本。

- 一键式部署，免去集群搭建烦恼
只需要在实例管理界面选好规格配置，提交订单，后台将自动创建部署完成一整套RabbitMQ实例。
- 兼容开源，业务零改动迁移上云

兼容社区版RabbitMQ的API，具备原生RabbitMQ的所有消息处理特性。

业务系统基于开源的RabbitMQ进行开发，只需加入少量认证安全配置，即可使用分布式消息服务RabbitMQ，做到无缝迁移。

📖 说明

RabbitMQ实例兼容开源社区RabbitMQ 3.8.35版本。

- 独占式体验

RabbitMQ实例采用物理隔离的方式部署，租户独占RabbitMQ实例，每个RabbitMQ之间互不影响。

- 高性能

单队列性能最高可达10万TPS（默认配置），增加队列可获得更高性能。

- 数据安全

独有的安全加固体系，提供业务操作云端审计，消息存储加密等有效安全措施。在网络通信方面，除了提供SSL认证，还借助虚拟私有云（VPC）和安全组等加强网络访问控制。

- 无忧运维

云服务平台提供一整套完整的监控告警等运维服务，故障自动发现和告警，避免7*24小时人工值守。RabbitMQ实例自动上报相关监控指标，如分区数、主题数、堆积消息数等，并支持配置监控数据发送规则，您可以在第一时间通过短信、邮件等获得业务消息队列的运行使用和负载状态。

- 支持多语言客户端

RabbitMQ是一款基于AMQP协议的开源服务，用于在分布式系统中存储转发消息，服务器端用Erlang语言（支持高并发、分布式以及健壮的容错能力等特点）编写，支持多种语言的客户端，如：Python、Ruby、.NET、Java、JMS、C、PHP、ActionScript、XMPP、STOMP和AJAX等。

1.3 典型应用场景

RabbitMQ作为一款热门的消息队列中间件，具备高效可靠的消息异步传递机制，主要用于不同系统间的数据交流和传递，在企业解决方案、金融支付、电信、电子商务、社交、即时通信、视频、物联网、车联网等众多领域都有广泛应用。

异步通信

将业务中属于非核心或不重要的流程部分，使用消息异步通知的方式发给目标系统，这样主业务流程无需同步等待其他系统的处理结果，从而达到系统快速响应的目的。

如网站的用户注册场景，在用户注册成功后，还需要发送注册邮件与注册短信，这两个流程使用RabbitMQ消息服务通知邮件发送系统与短信发送系统，从而提升注册流程的响应速度。

图 1-1 串行发送注册邮件与短信流程

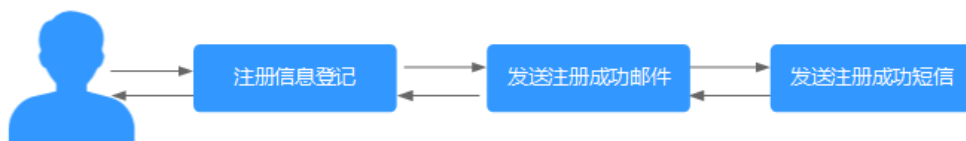
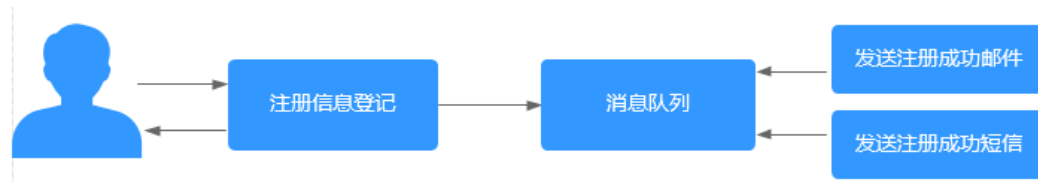


图 1-2 借助消息队列异步发送注册邮件与短信流程

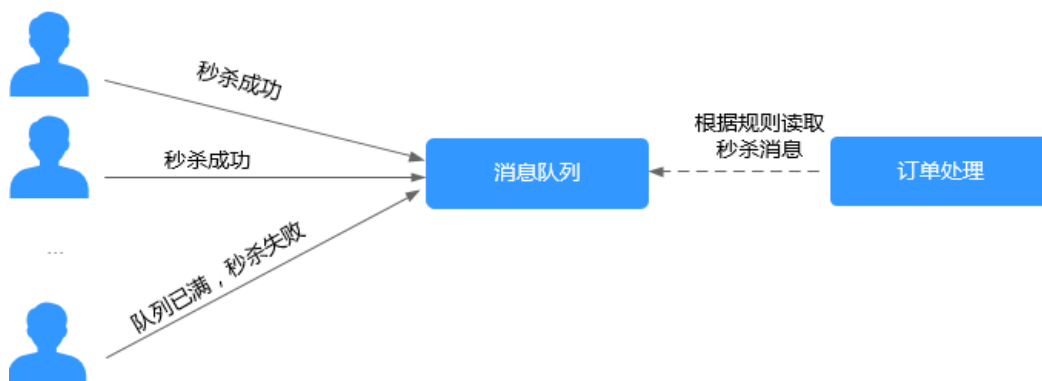


错峰流控与流量削峰

在电子商务系统或大型网站中，上下游系统处理能力存在差异，处理能力高的上游系统的突发流量可能会对处理能力低的某些下游系统造成冲击，需要提高系统的可用性的同时降低系统实现的复杂性。电商大促销等流量洪流突然来袭时，可以通过队列服务堆积缓存订单等信息，在下游系统有能力处理消息的时候再处理，避免下游订阅系统因突发流量崩溃。消息队列提供亿级消息堆积能力，3天的默认保留时长，消息消费系统可以错峰进行消息处理。

另外，在商品秒杀、抢购等流量短时间内暴增场景中，为了防止后端应用被压垮，可在前后端系统间使用RabbitMQ消息队列传递请求。

图 1-3 消息队列应对秒杀大流量场景



系统解耦

以电商秒杀、抢购等流量短时间内暴增场景为例，传统做法是，用户下单后，订单系统发送查询请求到库存系统，等待库存系统返回请求结果给订单系统。如果库存系统发生故障，订单系统获取不到数据，订单失败。这种情况下，订单系统和库存系统两个子系统高耦合。

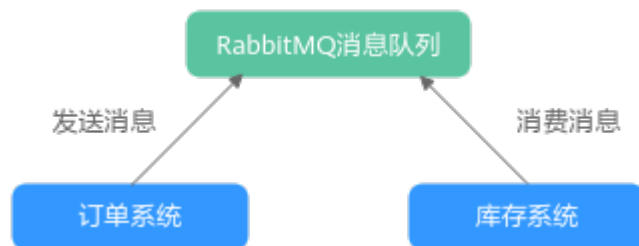
图 1-4 系统高耦合



引入RabbitMQ消息队列，当用户下单后，将消息写入到RabbitMQ消息队列中，然后返回用户下单成功。

库存系统订阅下单的消息，消费下单消息，然后进行库操作。即使库存系统出现故障，也不影响用户下单。

图 1-5 系统解耦



高可用

普通队列，由于队列以及队列内容仅存储在单代理上，当该代理故障后，对应的队列不可用。

RabbitMQ引入镜像队列机制，镜像队列是开源RabbitMQ 2.6.0版本新增的一个功能，允许集群将队列镜像到其他代理上，当集群某一代理宕机后，队列能自动切换到镜像中的其他代理，保证服务的可用性。

RabbitMQ引入仲裁队列机制，仲裁队列是开源RabbitMQ 3.8版本新增的一个功能，提供队列复制的能力，当集群某一代理宕机后，队列依旧可以正常运行，保证服务的可用性。

1.4 产品规格

RabbitMQ 实例规格

RabbitMQ实例兼容开源RabbitMQ 3.8.35，实例类型包括单机和集群，实例规格请参考表1-1。

表 1-1 RabbitMQ 集群实例规格

型号	代理数	存储空间范围 (GB)	TPS参考值	单个代理最大消费者数	单个代理建议队列数	单个代理最大连接数
rabbitmq.2u 4g.cluster	3	300~90000	3000	4000	100	1000
	5	500~150000	5000	4000	100	1000
	7	700~210000	7000	4000	100	1000
rabbitmq.4u 8g.cluster	3	300~90000	6000	8000	200	2000
	5	500~150000	10000	8000	200	2000
	7	700~210000	14000	8000	200	2000
rabbitmq.8u 16g.cluster	3	300~90000	12000	16000	400	4000
	5	500~150000	20000	16000	400	4000
	7	700~210000	28000	16000	400	4000

型号	代理数	存储空间范围 (GB)	TPS参考值	单个代理最大消费者数	单个代理建议队列数	单个代理最大连接数
rabbitmq.12 u24g.cluster	3	300~90000	24000	24000	600	6000
	5	500~150000	40000	24000	600	6000
	7	700~210000	56000	24000	600	6000
rabbitmq.16 u32g.cluster	3	300~90000	48000	32000	800	8000
	5	500~150000	80000	32000	800	8000
	7	700~210000	112000	32000	800	8000
rabbitmq.24 u48g.cluster	3	300~90000	60000	40000	1000	10000
	5	500~150000	100000	40000	1000	10000
	7	700~210000	140000	40000	1000	10000
rabbitmq.32 u64g.cluster	3	300~90000	72000	40000	1000	10000
	5	500~150000	120000	40000	1000	10000
	7	700~210000	168000	40000	1000	10000

📖 说明

- 为了保证稳定性，服务端限制了单条消息的最大长度为50MB，请勿发送大于此长度的消息。
- 上表中TPS（即生产+消费的TPS）是指以2K大小的消息为例的每秒处理消息条数，测试场景为不开启持久化的非镜像队列，实时生产实时消费，队列无积压。此数据仅供参考，生产使用需要以实际压测性能为准。
- 服务端的性能主要跟以下因素相关：队列数、消息堆积、连接数、channel、消费者数、镜像队列、优先级队列、消息持久化和exchange类型等，在选择实例规格时，请根据业务模型压测结果选择。
- 一条连接最多可以开启2047个channel。
- 单机版实例可用于测试场景，不建议用于生产业务，暂不提供单机版实例的产品规格。

新老规格对应关系

2种RabbitMQ实例规格对比，新老规格的对应关系如所示。

表 1-2 RabbitMQ 实例新老规格对应关系

老规格		对应的新规格	
规格类型	TPS参考值	规格类型	TPS参考值
4核 8GB * 3	3000	rabbitmq.4u8g.cluster * 3	6000

老规格		对应的新规格	
8核 16GB * 3	6000	rabbitmq.8u16g.cluster * 3	12000
16核 32GB * 3	24000	rabbitmq.16u32g.cluster * 3	48000

新老规格区别如下：

- 新规格性能更好，同等价格下性价比更优。
- 老规格使用的非独享资源，在高负载情况下容易出现资源抢占情况。新规格使用的独占资源，性能更优、稳定性更好。
- 新规格支持灵活的水平/垂直动态扩容，能更好的应对复杂的业务变化情况。
- 新规格支持更大规格的，最大可以支持rabbitmq.32u64g.cluster。

RabbitMQ 实例的存储空间估算参考

在集群模式中，RabbitMQ需要对消息持久化写入到磁盘中，因此，您在创建 RabbitMQ实例选择存储空间时，建议根据业务消息体积预估以及镜像队列副本数选择合适的存储空间。镜像队列副本数最大为集群的代理数。

例如：业务消息体积预估100GB，则磁盘容量最少应为100GB*镜像队列副本数+预留磁盘大小100GB。

如果是单机实例，则是计算业务消息体积+预留磁盘大小即可。

当前RabbitMQ实例支持修改集群实例的代理个数，您可以根据业务情况，随时更改集群代理个数。单机实例暂不支持变更规格。

1.5 与 Kafka、RocketMQ 的差异

表 1-3 功能差异

功能项	RocketMQ	Kafka	RabbitMQ
优先级队列	不支持	不支持	支持。建议优先级大小设置在0-10之间。
延迟队列	支持	不支持	支持
死信队列	支持	不支持	支持
消息重试	支持	不支持	不支持。
消费模式	支持客户端主动拉取和服务端推送两种方式。	客户端主动拉取。	支持客户端主动拉取和服务端推送两种模式。
广播消费	支持	支持	支持

功能项	RocketMQ	Kafka	RabbitMQ
消息回溯	支持	支持。Kafka支持按照offset和timestamp两种维度进行消息回溯。	不支持。RabbitMQ中消息一旦被确认消费就会被标记删除。
消息堆积	支持	支持。考虑吞吐因素，Kafka的堆积效率比RabbitMQ总体上要高。	支持
持久化	支持	支持	支持
消息追踪	支持	不支持	支持。 RabbitMQ中可以采用Firehose或者rabbitmq_tracing插件实现，但开启rabbitmq_tracing插件会影响性能，建议只在定位问题过程中开启。
消息过滤	支持	支持	不支持，但可以自行封装。
多租户	支持	支持	支持
多协议支持	兼容RocketMQ协议。	只支持Kafka自定义协议。	RabbitMQ基于AMQP协议实现。
跨语言支持	支持多语言的客户端。	采用Scala和Java编写，支持多种语言的客户端。	支持多种语言的客户端。
流量控制	RocketMQ 5.x支持基于实例规格的流量控制。	支持client、user和Topic级别，通过主动设置可将流控作用于生产者或消费者。	RabbitMQ的流控基于Credit-Based算法，是内部被动触发的保护机制，作用于生产者层面。
消息顺序性	单队列（queue）内有序。	支持单分区（partition）级别的顺序性。	单线程发送、单线程消费并且不采用延迟队列、优先级队列等一些高级功能时，才能实现消息有序。
安全机制	支持SSL认证。	支持SSL、SASL身份认证和读写权限控制。	支持SSL认证。
事务性消息	支持	支持	支持

1.6 与其他云服务的关系

- 弹性云服务器（Elastic Cloud Server）
弹性云服务器是由CPU、内存、操作系统、云硬盘组成的基础的计算组件。RabbitMQ实例运行在弹性云服务器上，一个代理对应一台弹性云服务器。
- 云硬盘（Elastic Volume Service）
云硬盘为云服务器提供块存储服务，RabbitMQ的所有数据（如消息和日志等）都保存在云硬盘中。
- 云审计（Cloud Trace Service）
云审计为您提供云服务资源的操作记录，记录内容包括您从云服务平台管理控制台或者开放API发起的云服务资源操作请求以及每次请求的结果，供您查询、审计和回溯使用。
- 虚拟私有云
RabbitMQ实例运行于虚拟私有云，需要使用虚拟私有云创建的IP和带宽。通过虚拟私有云安全组的功能可以增强访问RabbitMQ实例的安全性。
- 云监控（Cloud Eye）
云监控是一个开放性的监控平台，提供资源的实时监控、告警、通知等服务。

📖 说明

RabbitMQ实例向CloudEye上报监控数据的更新周期为1分钟。

- 弹性公网IP（Elastic IP）
弹性公网IP提供独立的公网IP资源，包括公网IP地址与公网出口带宽服务。RabbitMQ实例绑定弹性公网IP后，可以通过公网访问RabbitMQ实例。
- 数据加密服务（Data Encryption Workshop）
在创建RabbitMQ实例时，选择是否开启磁盘加密。开启磁盘加密，可以提升数据的安全性。磁盘加密依赖于数据加密服务提供的密钥。
- 标签管理服务（Tag Management Service）
标签管理服务是一种快速便捷将标签集中管理的可视化服务，提供跨区域、跨服务的集中标签管理和资源分类功能。
为RabbitMQ实例添加标签，可以方便用户识别和管理拥有的实例资源。

1.7 约束与限制

本章节介绍分布式消息服务RabbitMQ产品功能的约束和限制。

实例

表 1-4 实例约束与限制

限制项	约束和限制
版本	当前服务端版本为3.8.35。

限制项	约束和限制
连接数	RabbitMQ单机和集群实例，不同实例规格的连接数上限不一致，具体限制，请参考 产品规格 。
通道数	单条连接可以建立的通道数 ≤ 2047 。
内存高水位阈值	$\leq 40\%$ 内存使用率超过40%可能会触发内存高水位，内存高水位会导致生产者流程被阻塞。
磁盘高水位阈值	$\geq 5GB$ 磁盘剩余空间低于5GB会触发磁盘高水位，生产者流程被阻塞。
cluster_partition_handling	pause_minority 当集群发生网络分区时，代理会检查自己是否处于“少数派”（存储分区的代理数小于等于总代理数的一半称为少数派）。少数派中的代理将会自动关闭服务并定期检测网络状态，待分区恢复之后重新启动服务。如果未开启镜像队列，发生分区时少数派上的队列将无法生产消费。 此策略相当于放弃了可用性而选择了数据一致性。
rabbitmq_delayed_message_exchange	插件延迟时间存在1%左右的误差，可能提前或者推迟发送消息给消费者。
RabbitMQ插件	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。
修改VPC/子网/可用区	实例创建后，不支持修改VPC/子网/可用区。
存储空间	只支持扩大存储空间，不支持减小存储空间。
代理数量	<ul style="list-style-type: none"> 集群实例只支持增加代理数量，不支持减小代理数量。 增加代理数量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。 单机实例不支持修改代理数量。
代理规格	<ul style="list-style-type: none"> 支持扩容/缩容代理规格。 单机实例和没有配置镜像/仲裁队列的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。

Vhost

表 1-5 Vhost 约束与限制

限制项	约束和限制
删除Vhost	创建实例时创建的默认Vhost，不支持删除。

Exchange

表 1-6 Exchange 约束与限制

限制项	约束和限制
默认Exchange	RabbitMQ 3.8.35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。
绑定Exchange	<ul style="list-style-type: none"> RabbitMQ 3.8.35版本中，名为“(AMQP default)”的Exchange不能绑定任何Exchange。 “Internal”为“是”的Exchange只能绑定Exchange，不能绑定Queue。
删除Exchange	RabbitMQ 3.8.35版本中，默认Exchange不支持删除。

Queue

表 1-7 Queue 约束与限制

限制项	约束和限制
绑定Queue	<ul style="list-style-type: none"> RabbitMQ 3.8.35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。
惰性队列	仅RabbitMQ 3.8.35版本支持惰性队列。
仲裁队列	仅RabbitMQ 3.8.35版本支持仲裁队列。
单一活跃消费者	仅RabbitMQ 3.8.35版本支持单一活跃消费者特性。

消息

表 1-8 消息约束与限制

限制项	约束和限制
消息大小	单条消息的最大长度为50MB。 服务端限制了单条消息的最大长度，请勿发送大于此长度的消息，否则生产失败。

1.8 RabbitMQ 相关概念

云服务平台使用RabbitMQ作为消息引擎，RabbitMQ是一个生产者和消费者模型，主要负责接收、存储和转发消息。以下概念基于RabbitMQ进行描述。

消息

消息一般分为两部分，消息体和标签，标签主要用来描述这条消息，消息体是消息的内容，是一个JSON体或者数据等。

生产者发送消息，消费者消费消息，生产者与消费者彼此并无直接关系。

生产者 (Producer)

即向队列发送消息的一方。发布消息的最终目的在于将消息内容传递给其他系统/模块，使对方按照约定处理该消息。

消费者 (Consumer)

接收消息的一方。消费者订阅RabbitMQ的队列，当消费者消费一条消息时，只是消费消息的消息体。在消息路由的过程中，会丢弃标签，存入到队列中的只有消息体。

队列 (Queue)

队列是用于存储消息的，生产者将消息送到队列，消费者从队列中获取和消费消息。多个消费者可以同时订阅同一个队列，队列里的消息分配给不同的消费者。

代理 (Broker)

消息中间件的服务节点。

Vhost

Vhost是指虚拟主机，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。

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。
更多关于Exchange类型的描述请参见[Exchange类型介绍](#)。

1.9 Exchange 类型介绍

Exchange主要包含4种类型：Direct、Fanout、Topic和Header。

Direct Exchange

工作原理：

1. 将一个Queue绑定到某个Direct Exchange上，并赋予该绑定一个Routing Key。
2. 当一个携带着Routing Key的消息被发送给Direct Exchange时，Direct Exchange会将它路由给与绑定Queue时设置的Routing Key完全匹配的Queue。

路由规则：

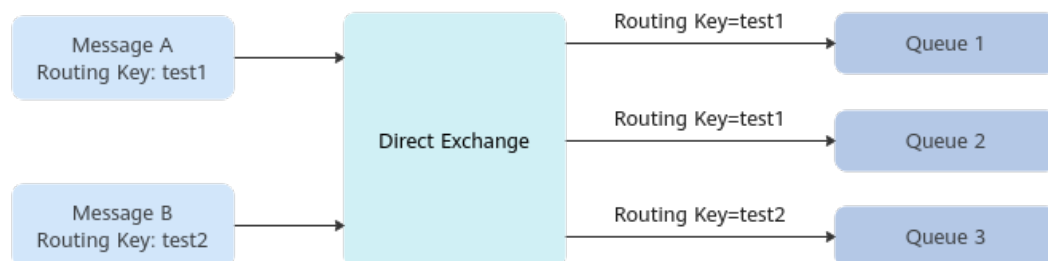
Direct Exchange会将消息路由到Routing Key完全匹配的Queue中。

应用场景：

Direct Exchange常用于单播路由。

应用示例：

图 1-6 Direct Exchange 示例



如上图所示，Message A将会发送到Queue 1和Queue 2中，Message B将会发送到Queue 3中。

Fanout Exchange

工作原理：

如果有N个Queue绑定了Fanout Exchange，当Fanout Exchange接收到一个消息时，会将它分别路由到这N个Queue中。Fanout Exchange是所有Exchange类型中转发消息最快的。

路由规则：

Fanout Exchange将消息路由到所有与其绑定的Queue中。

应用场景：

Fanout Exchange常用于广播消息。

应用示例：

图 1-7 Fanout Exchange 示例



如上图所示，Message A将会发送到Queue 1和Queue 2中。

Topic Exchange

工作原理：

1. 将一个Queue绑定到某个Topic Exchange上，并赋予该绑定一个Routing Key，此Routing Key包含通配符。
2. 当一个携带着Routing Key的消息被发送给Topic Exchange时，Topic Exchange会将Routing Key进行通配符匹配，如果匹配成功，则将它路由给对应的Queue。

Topic Exchange的通配符包括“*”和“#”，单词间通过“.”隔开，例如test.#。

- *表示一个单词。
- #表示零个、一个或者多个单词。

路由规则：

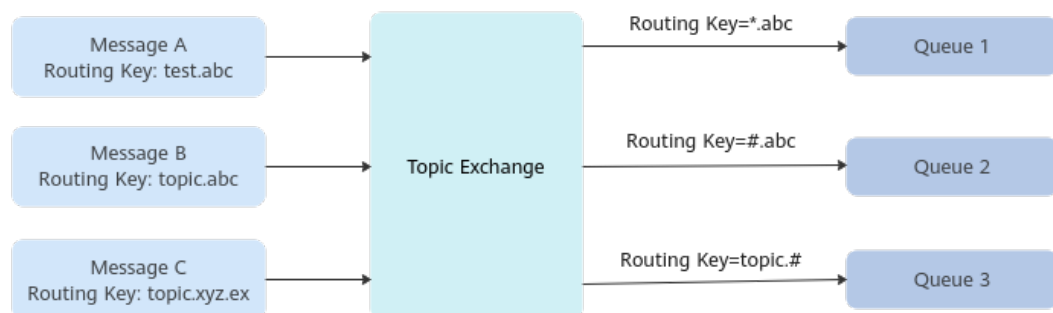
Topic Exchange会将Routing Key进行通配符匹配，然后将消息路由到匹配成功的Queue中。

应用场景：

Topic Exchange常用于多播路由。

应用示例：

图 1-8 Topic Exchange 示例



如上图所示，Message A将会发送到Queue 1和Queue 2，Message B将会发送到Queue 1、Queue 2和Queue 3，Message C将会发送到Queue 3。

Header Exchange

工作原理：

1. 将一个Queue绑定到某个Header Exchange上，并设置绑定属性键值对。
2. 将消息发送给Header Exchange时，需要设置消息的Header属性键值对，Header Exchange根据消息Headers属性键值对和绑定属性键值对的匹配情况路由消息。

匹配算法由一个特殊的绑定属性键值对控制。该属性为“x-match”，它的取值如下：

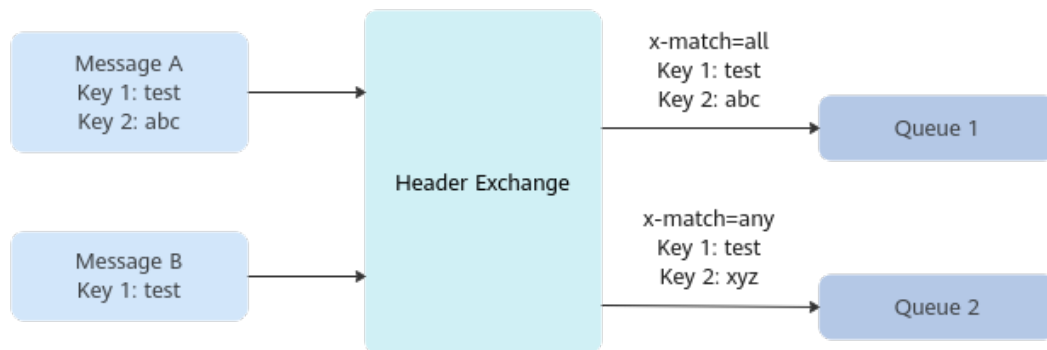
- x-match=all，表示所有的消息Headers属性键值对和绑定属性键值对都匹配才会路由消息。
- x-match=any，表示只要有一组消息Headers属性键值对和绑定属性键值对匹配，就会路由消息。

路由规则：

Header Exchange根据消息Headers属性键值对和绑定属性键值对的匹配情况路由消息。

应用示例：

图 1-9 Header Exchange 示例



如上图所示，Message A将会发送到Queue 1和Queue 2，Message B将会发送到Queue 2。

1.10 权限管理

如果您需要对云服务平台上购买的DMS for RabbitMQ资源，给企业中的员工设置不同的访问权限，以达到不同员工之间的权限隔离，您可以使用统一身份认证服务（Identity and Access Management，简称IAM）进行精细的权限管理。该服务提供用户身份认证、权限分配、访问控制等功能，可以帮助您安全的控制资源的访问。

通过IAM，您可以在账号中给员工创建IAM用户，并使用策略来控制他们对资源的访问范围。例如您的员工中有负责软件开发的人员，您希望他们拥有DMS for RabbitMQ的使用权限，但是不希望他们拥有删除RabbitMQ实例等高危操作的权限，那么您可以使用IAM为开发人员创建用户，通过授予仅能使用DMS for RabbitMQ，但是不允许删除RabbitMQ实例的权限策略，控制他们对DMS for RabbitMQ资源的使用范围。

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

IAM是云服务平台提供权限管理的基础服务，无需付费即可使用，您只需要为您账号中的资源进行付费。关于IAM的详细介绍，请参见[IAM产品简介](#)。

DMS for RabbitMQ 权限

默认情况下，管理员创建的IAM用户没有任何权限，需要将其加入用户组，并给用户组授予策略或角色，才能使得用户组中的用户获得对应的权限，这一过程称为授权。授权后，用户就可以基于被授予的权限对云服务进行操作。

DMS for RabbitMQ部署时通过物理区域划分，为项目级服务。授权时，“作用范围”需要选择“区域级项目”，然后在指定区域对应的项目中设置相关权限，并且该权限仅对此项目生效；如果在“所有项目”中设置权限，则该权限在所有区域项目中都生效。访问DMS for RabbitMQ时，需要先切换至授权区域。

权限根据授权精细程度分为角色和策略。

- **角色**：IAM最初提供的一种根据用户的工作职能定义权限的粗粒度授权机制。该机制以服务为粒度，提供有限的服务相关角色用于授权。由于云服务平台各服务之间存在业务依赖关系，因此给用户授予角色时，可能需要一并授予依赖的其他角色，才能正确完成业务。角色并不能满足用户对精细化授权的要求，无法完全达到企业对权限最小化的安全管控要求。
- **策略**：IAM最新提供的一种细粒度授权的能力，可以精确到具体服务的操作、资源以及请求条件等。基于策略的授权是一种更加灵活的授权方式，能够满足企业对权限最小化的安全管控要求。例如：针对DMS for RabbitMQ服务，管理员能够控制IAM用户仅能对实例进行指定的管理操作。多数细粒度策略以API接口为粒度进行权限拆分，DMS for RabbitMQ服务支持的API授权项请参见[细粒度策略支持的授权项](#)。

📖 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RabbitMQ分配用户与权限时，请选择并使用“DMS”的权限与策略。

如表1-9所示，包括了DMS for RabbitMQ的所有系统权限。

表 1-9 DMS for RabbitMQ 系统权限

系统角色/策略名称	描述	类别	依赖关系
DMS FullAccess	分布式消息服务管理员权限，拥有该权限的用户可以操作所有分布式消息服务的功能。	系统策略	无
DMS UserAccess	分布式消息服务普通用户权限（没有实例创建、修改、删除、扩容）。	系统策略	无
DMS ReadOnlyAccess	分布式消息服务的只读权限，拥有该权限的用户仅能查看分布式消息服务数据。	系统策略	无
DMS VPCAccess	分布式消息服务租户委托时需要授权的VPC操作权限。	系统策略	无
DMS KMSAccess	分布式消息服务租户委托时需要授权的KMS操作权限。	系统策略	无

系统角色/策略名称	描述	类别	依赖关系
DMS Administrator	分布式消息服务的管理员权限。	系统角色	依赖Tenant Guest和VPC Administrator。

表1-10列出了DMS for RabbitMQ常用操作与系统策略的授权关系，您可以参照该表选择合适的系统策略。

表 1-10 常用操作与系统策略的关系

操作	DMS FullAccess	DMS UserAccess	DMS ReadOnlyAccess	DMS VPCAccess	DMS KMSAccess
创建实例	√	×	×	×	×
修改实例	√	×	×	×	×
删除实例	√	×	×	×	×
变更实例规格	√	×	×	×	×
查询实例信息	√	√	√	×	×

相关链接

- [IAM产品简介](#)
- [创建用户组、用户并授予DMS for RabbitMQ权限](#)
- [细粒度策略支持的授权项](#)

1.11 计费说明

分布式消息服务RabbitMQ支持按需付费模式。

计费项

分布式消息服务RabbitMQ对您选择的RabbitMQ实例和RabbitMQ存储收费。

表 1-11 RabbitMQ 服务计费项

计费项	计费说明
实例	<ul style="list-style-type: none"> 对您选择的实例规格计费，具体请参见表1-12。 实例提供按需（小时）计费方式。
存储	<ul style="list-style-type: none"> 对您选择的实例存储空间计费（每个实例规格您都可以选择高IO和超高IO两种不同的云硬盘类型以满足您的业务需求）。实例存储可选择不同副本，如：实际队列容量为100GB，3副本，则应购买100GB*3=300GB存储容量。 存储空间范围见表1-12，步长100G。 实例存储提供按需（小时）计费方式。

表 1-12 RabbitMQ 集群实例规格

型号	代理数	存储空间范围（GB）	TPS参考值	单个代理最大消费者数	单个代理建议队列数	单个代理最大连接数
rabbitmq.2u 4g.cluster	3	300~90000	3000	4000	100	1000
	5	500~150000	5000	4000	100	1000
	7	700~210000	7000	4000	100	1000
rabbitmq.4u 8g.cluster	3	300~90000	6000	8000	200	2000
	5	500~150000	10000	8000	200	2000
	7	700~210000	14000	8000	200	2000
rabbitmq.8u 16g.cluster	3	300~90000	12000	16000	400	4000
	5	500~150000	20000	16000	400	4000
	7	700~210000	28000	16000	400	4000
rabbitmq.12 u24g.cluster	3	300~90000	24000	24000	600	6000
	5	500~150000	40000	24000	600	6000
	7	700~210000	56000	24000	600	6000
rabbitmq.16 u32g.cluster	3	300~90000	48000	32000	800	8000
	5	500~150000	80000	32000	800	8000
	7	700~210000	112000	32000	800	8000
rabbitmq.24 u48g.cluster	3	300~90000	60000	40000	1000	10000
	5	500~150000	100000	40000	1000	10000
	7	700~210000	140000	40000	1000	10000

型号	代理数	存储空间范围 (GB)	TPS参考值	单个代理最大消费者数	单个代理建议队列数	单个代理最大连接数
rabbitmq.32u64g.cluster	3	300~90000	72000	40000	1000	10000
	5	500~150000	120000	40000	1000	10000
	7	700~210000	168000	40000	1000	10000

计费模式

提供按需付费（小时）的计费方式：这种购买方式比较灵活，可以即开即停，按实际使用时长计费。以自然小时为单位整点计费，不足一小时按一小时计费。

变更配置

RabbitMQ集群实例支持扩容代理个数和磁盘大小，单机RabbitMQ实例支持扩容磁盘。

2 快速入门

2.1 快速连接 RabbitMQ 并生产消费消息

本文将为您介绍分布式消息服务RabbitMQ版入门的使用流程，以创建一个未开启SSL的RabbitMQ实例，客户端使用内网通过同一个VPC连接RabbitMQ实例生产消费消息为例，帮助您快速上手RabbitMQ。

图 2-1 RabbitMQ 使用流程



1. 准备工作

RabbitMQ实例运行于虚拟私有云（Virtual Private Cloud，以下简称VPC）中，在创建实例前需要确保有可用的虚拟私有云。

2. 创建RabbitMQ实例

在创建实例时，您可以根据需求选择需要的实例规格和数量。

3. 连接实例

客户端以RabbitMQ提供的demo，连接未开启SSL的实例。

步骤一：准备工作

步骤1 为用户添加RabbitMQ实例的操作权限。

如果您需要对云上的资源进行精细管理，请使用统一身份认证服务（Identity and Access Management，简称IAM）创建IAM用户及用户组，并授权，以使得IAM用户

获得RabbitMQ实例的操作权限。具体操作请参考[创建用户并授权使用DMS for RabbitMQ](#)。

步骤2 创建VPC和子网。

在创建RabbitMQ实例前，确保已存在可用的VPC和子网。创建VPC和子网的具体步骤，请参考[创建虚拟私有云和子网](#)。

须知

创建的VPC与RabbitMQ实例必须在相同的区域。

步骤3 创建安全组并添加安全组规则。

在创建RabbitMQ实例前，确保已存在可用的安全组。创建安全组的具体步骤，请参考[创建安全组](#)。

连接RabbitMQ实例前，请添加[表2-1](#)所示安全组规则，其他规则请根据实际需要添加。

表 2-1 安全组规则


方向	协议	端口	源地址	说明
入方向	TCP	5672	0.0.0.0/0	访问RabbitMQ实例（关闭SSL加密）。

说明

安全组创建后，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2-1](#)的规则。

步骤4 构建生产消费客户端。

本文以Linux系统的弹性云服务器（Elastic Cloud Server，简称ECS）作为生产消费客户端。在创建RabbitMQ实例前，请先创建开启弹性公网IP的ECS、安装JDK以及配置环境变量。

1. 登录管理控制台，在左上角单击，选择“计算 > 弹性云服务器”，创建一个ECS实例。
创建ECS的具体步骤，请参考[创建弹性云服务器](#)。如果您已有可用的ECS，可重复使用，不需要再次创建。
2. 使用root用户登录ECS。
3. 安装Java JDK，并配置JAVA_HOME与PATH环境变量。
 - a. 下载Java JDK。

说明

ECS默认自带的JDK可能不符合要求，例如OpenJDK，需要配置为Oracle的JDK，可至Oracle官方下载页面[下载Java Development Kit 1.8.111及以上版本](#)。

- b. 解压Java JDK。

```
tar -zxvf jdk-8u321-linux-x64.tar.gz
```

“jdk-8u321-linux-x64.tar.gz”为JDK的版本，请根据实际情况修改。
- c. 打开“.bash_profile”文件。

```
vim ~/.bash_profile
```
- d. 添加如下内容。

```
export JAVA_HOME=/opt/java/jdk1.8.0_321
export PATH=$JAVA_HOME/bin:$PATH
```

“/opt/java/jdk1.8.0_321”为JDK的安装路径，请根据实际情况修改。
- e. 按“Esc”，然后输入以下命令，按“Enter”，保存并退出“.bash_profile”文件。

```
:wq
```
- f. 执行如下命令使修改生效。

```
source .bash_profile
```
- g. 查看Java JDK是否安装成功。

```
java -version
```

回显信息中包含如下信息，表示Java JDK安装成功。

```
java version "1.8.0_321"
```

----结束

步骤二：创建 RabbitMQ 实例

步骤1 登录分布式消息服务RabbitMQ控制台，单击页面右上方的“购买RabbitMQ实例”。

步骤2 设置实例基础信息，配置详情请参考[表2-2](#)。

表 2-2 设置实例基础信息

参数	说明
计费模式	选择“按需计费”，即先使用再付费，按照RabbitMQ实例实际使用时长计费，秒级计费，按小时结算。
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。 选择“阿联酋-阿布扎比”。
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）。 选择“阿联酋-阿布扎比(默认)”。
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。 选择“可用区1”。
实例名称	实例名称支持自定义，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。 输入“rabbitmq-test”。

参数	说明
企业项目	该参数针对企业用户使用。企业项目是对企业不同项目间资源的分组和管理，属于逻辑隔离。 选择“default”。
版本	RabbitMQ的版本号。 选择“3.8.35”。
部署架构	选择“单机”，表示部署一个RabbitMQ代理。
CPU架构	支持“x86计算”。 保持默认值。
代理规格	根据业务需求选择相应的代理规格。 选择“rabbitmq.2u4g.single”。
代理数量	默认为“1”，不支持修改。
单个代理存储空间	根据实际需要选择存储RabbitMQ数据的磁盘类型和磁盘大小。 实例总存储空间 = 单个代理的存储空间 * 代理数量， RabbitMQ实例创建后，磁盘类型不支持修改。 磁盘类型选择“超高I/O”，磁盘大小设置为“100GB”。
磁盘加密	选择“不开启”。

步骤3 设置实例网络环境信息，配置详情请参考[表2-3](#)。

表 2-3 设置实例网络环境信息

参数	说明
虚拟私有云	虚拟私有云和子网在RabbitMQ实例创建完成后，不支持修改。 选择 准备工作 中设置好的虚拟私有云和子网。
安全组	选择 准备工作 中设置好的安全组。

步骤4 设置实例的访问方式，配置详情请参考[表2-4](#)。

表 2-4 设置实例的访问方式

参数	说明
SSL	选择“不开启”。

- host: 从[创建实例](#)中获取的连接地址。
- port: RabbitMQ实例的连接端口, 输入5672。
- user: [创建实例](#)时, 设置的用户名。
- password: [创建实例](#)时, 设置的密码。

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5 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[创建实例](#)中获取的连接地址。
- port: RabbitMQ实例的连接端口, 输入5672。
- user: [创建实例](#)时, 设置的用户名。
- password: [创建实例](#)时, 设置的密码。

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。

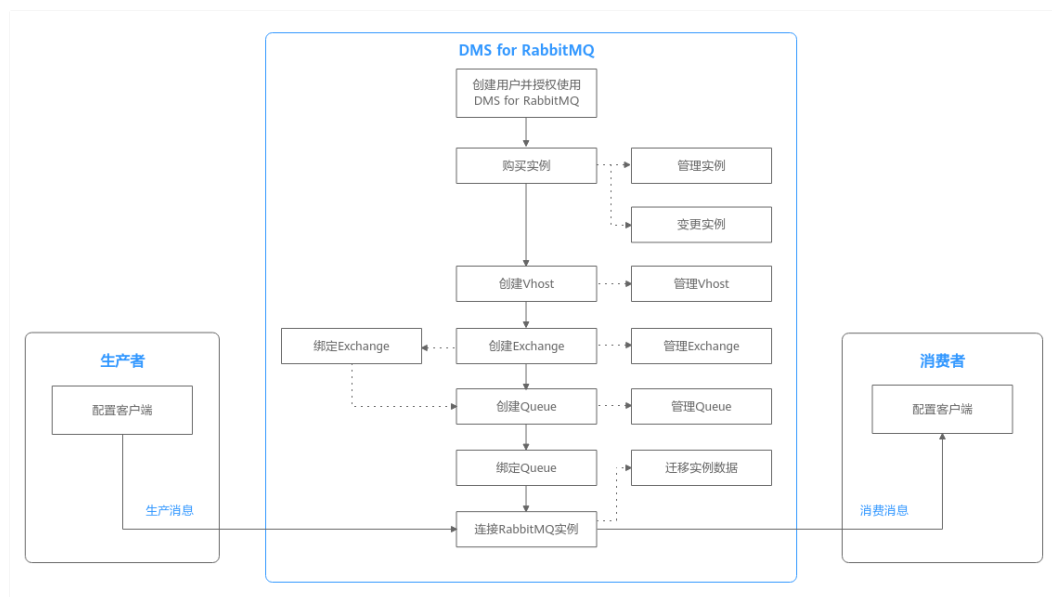
----**结束**

3 RabbitMQ 业务使用流程

分布式消息服务RabbitMQ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 3-1 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。

4 权限管理

4.1 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。

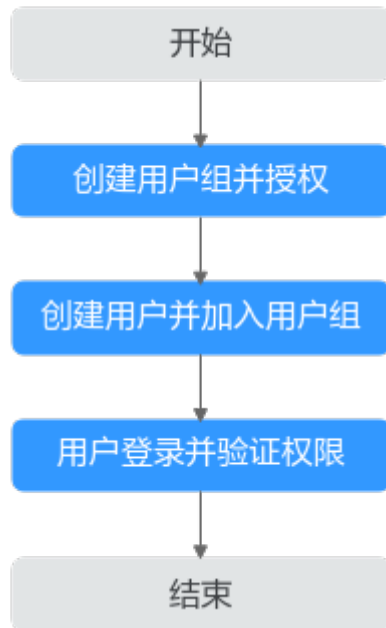
本章节为您介绍对用户授权的方法，操作流程如[图4-1](#)所示。

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 4-1 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，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（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](#)。

目前云服务平台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 4-1 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 4-2 DMS for RabbitMQ 请求条件

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

5 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5-1](#)所示资源。


表 5-1 RabbitMQ 实例依赖资源


资源名称	要求	创建方法
VPC和子网	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，请根据实际需要进行配置。 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。	创建VPC和子网的操作指导请参考《虚拟私有云 用户指南》的创建虚拟私有云和子网。
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7-2 。	创建安全组的操作指导请参考《虚拟私有云 用户指南》的创建安全组，为安全组添加规则的操作指导请参考《虚拟私有云 用户指南》的添加安全组规则。

资源名称	要求	创建方法
弹性IP地址	<p>如果客户端使用公网连接 RabbitMQ实例，请提前创建弹性IP地址。</p> <p>在创建弹性IP地址时，应注意如下要求：</p> <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。 	<p>创建弹性IP地址的操作指导请参考《弹性公网IP 用户指南》的“为弹性云服务器申请和绑定弹性公网IP”。</p>
密钥	<p>创建RabbitMQ实例时，如果需要对磁盘进行加密，请提前创建密钥。</p> <p>在创建密钥时，应注意如下要求：创建的密钥与RabbitMQ实例在相同的区域。</p>	<p>创建密钥的操作指导请参考《数据加密服务 用户指南》的“创建密钥”。</p>

购买 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。

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

步骤4 单击页面右上方的“购买RabbitMQ实例”。

步骤5 选择“计费模式”。

步骤6 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7 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8 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9 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10 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11 设置实例信息。

1. 版本：RabbitMQ的版本号，当前仅支持RabbitMQ 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. CPU架构：保持默认值即可。
4. 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。当内存占用率达到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。

5. 代理数量：根据实例情况选择代理个数。
6. 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[如何选择磁盘类型](#)。
 - 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）。
7. 在“磁盘加密”区域，选择是否开启磁盘加密。开启磁盘加密，可以提升数据的安全性，但是会降低磁盘读写的速度，请根据实际情况选择。磁盘加密依赖于数据加密服务的能力实现。开启磁盘加密后，需要选择提前创建好的密钥。如果没有密钥，单击“查看密钥列表”，跳转到数据加密服务页面，创建密钥。
RabbitMQ实例创建后，此参数不支持修改。
8. 虚拟私有云：选择已经创建好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。

9. 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。
10. 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 5-1 设置实例信息

部署架构 单机 集群

CPU架构 x86计算

代理规格

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

为了保证服务的稳定可靠，分布式消息服务RabbitMQ版采用了默认的40%高水位配置。当内存占用率达到40%高水位后，会触发流控，生产者流程会被阻塞。为了避免高水位的产生，请及时消费积压在队列中的消息。

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
实例创建后，代理数量不支持扩容。

单个代理存储空间 GB
实例总存储空间 100 GB
磁盘类型创建完成后不可修改，存储空间不支持扩容，所选磁盘性能和并发请参考 如何选择磁盘类型，并根据实际业务I/O要求进行选择。

磁盘加密

虚拟私有云
如需创建新的虚拟私有云，可前往 控制台 创建。实例创建完成后，虚拟私有云和子网都不能修改。

安全组

SSL
开启，数据将加密传输，安全性更高，但性能会下降。
注意：实例创建完成后，SSL开启/关闭不能修改。

步骤12 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13 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，创建弹性公网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14 填写完上述信息后，单击“立即购买”，进入“规格确认”页面。

步骤15 确认实例信息无误后，单击“提交”。

步骤16 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。

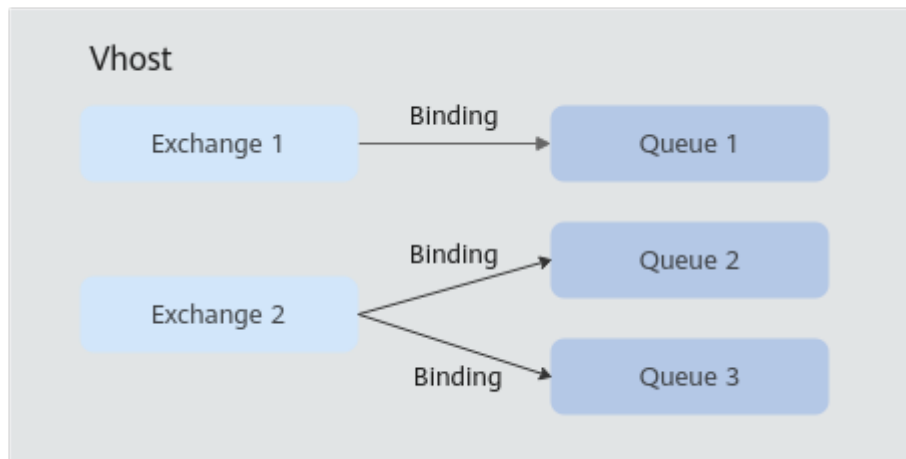
----结束

6 配置 Vhost

6.1 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 6-1 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。


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“应用服务 > 分布式消息服务 RabbitMQ”，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。开启消息追踪后，您可以跟踪消息的转发路径。

----结束

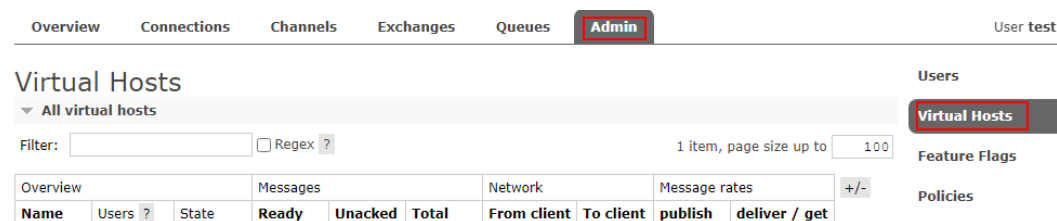
创建 RabbitMQ Vhost (RabbitMQ WebUI)

步骤1 登录[RabbitMQ WebUI](#)。

步骤2 在顶部导航栏选择“Admin”，进入Admin页面。

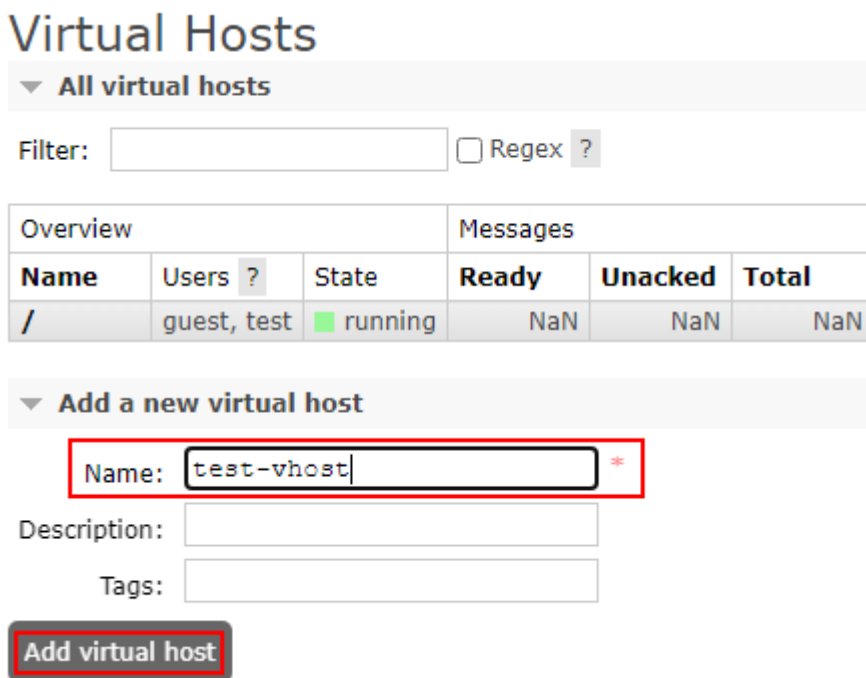
步骤3 在右侧导航栏选择“Virtual Hosts”，进入Virtual Hosts页面。

图 6-2 Virtual Hosts



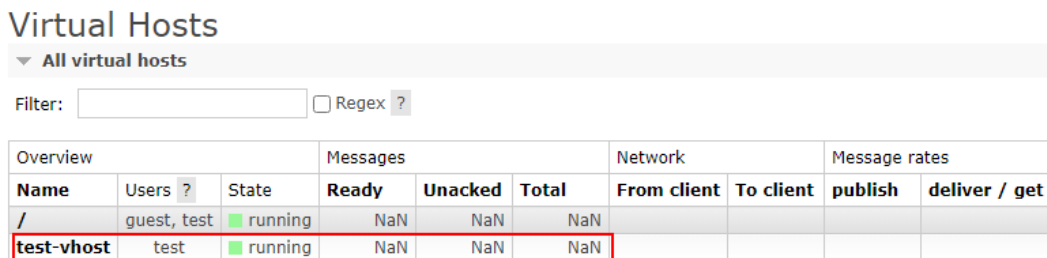
步骤4 在“Add a new virtual host”区域，输入Vhost名称，单击“Add virtual host”。

图 6-3 创建 Vhost (WebUI)



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

图 6-4 Vhost 列表 (WebUI)



----结束

6.2 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“应用服务 > 分布式消息服务 RabbitMQ”，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6-1，设置Exchange名称和配置信息。

表 6-1 Exchange 参数说明

参数	说明
名称	<p>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。</p> <p>创建Exchange后不能修改名称。</p>
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。
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。

参数	说明
持久化	是否开启Exchange持久化。 <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。
Internal	Exchange是否为RabbitMQ内部使用。 <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

6.3 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。
- “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“应用服务 > 分布式消息服务 RabbitMQ”，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6-2，设置绑定参数。



表 6-2 绑定参数说明

参数	说明
类型	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	<p>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。</p> <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。 • 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“应用服务 > 分布式消息服务 RabbitMQ”，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6-3，设置绑定参数。

表 6-3 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。• 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

6.4 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“应用服务 > 分布式消息服务 RabbitMQ”，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6-4，设置Queue的名称和配置信息。

表 6-4 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	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。
惰性队列模式	仅RabbitMQ 3.x.x版本包含此参数。 如果需要设置Queue为惰性队列，请输入“lazy”。 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。

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

6.5 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“应用服务 > 分布式消息服务 RabbitMQ”，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6-5](#)，设置绑定参数。

表 6-5 绑定参数说明

参数	说明
类型	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。

参数	说明
Routing Key	<p>设置Key值字符串，用于告知Exchange应该将消息投递到哪些Queue中。</p> <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。• 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


6.6 管理 RabbitMQ Vhost

6.6.1 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“应用服务 > 分布式消息服务 RabbitMQ”，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。

----结束

6.6.2 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）](#)


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“应用服务 > 分布式消息服务 RabbitMQ”，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

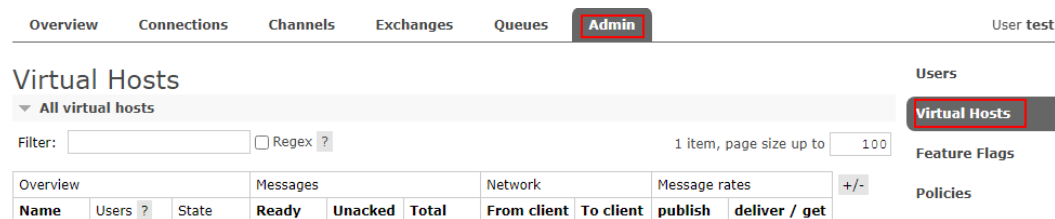
删除 Vhost（RabbitMQ WebUI）

步骤1 登录[RabbitMQ WebUI](#)。

步骤2 在顶部导航栏选择“Admin”，进入Admin页面。

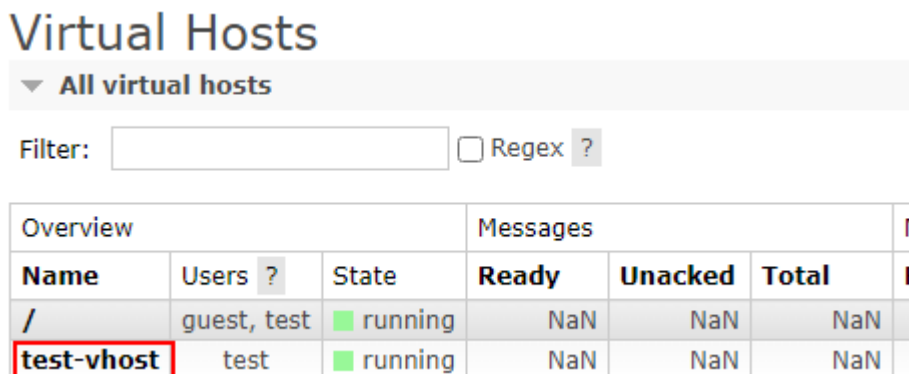
步骤3 在右侧导航栏选择“Virtual Hosts”，进入Virtual Hosts页面。

图 6-5 Virtual Hosts 页面



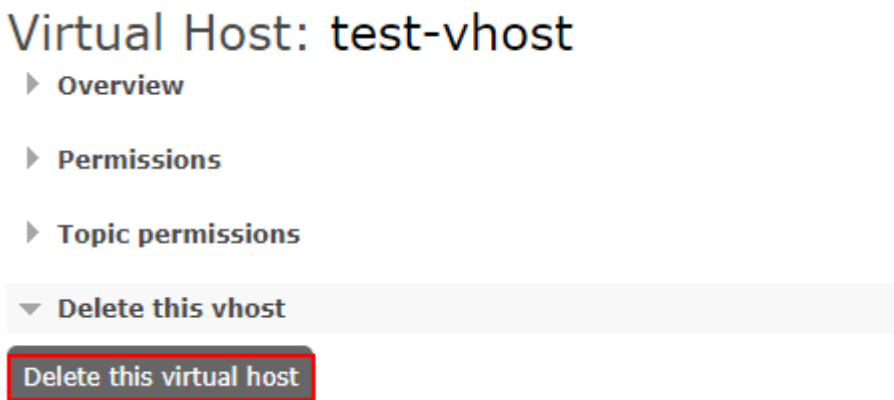
步骤4 单击待删除的Vhost名称，进入Vhost详情页。

图 6-6 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 6-7 删除 Vhost



步骤6 单击“确定”，完成Vhost的删除。

----结束

6.7 管理 RabbitMQ Exchange

6.7.1 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“应用服务 > 分布式消息服务 RabbitMQ”，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“应用服务 > 分布式消息服务 RabbitMQ”，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。

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

6.7.2 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“应用服务 > 分布式消息服务 RabbitMQ”，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，通过以下任何一种方法，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

6.8 管理 RabbitMQ Queue

6.8.1 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“应用服务 > 分布式消息服务 RabbitMQ”，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。

---结束

6.8.2 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）](#)

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。

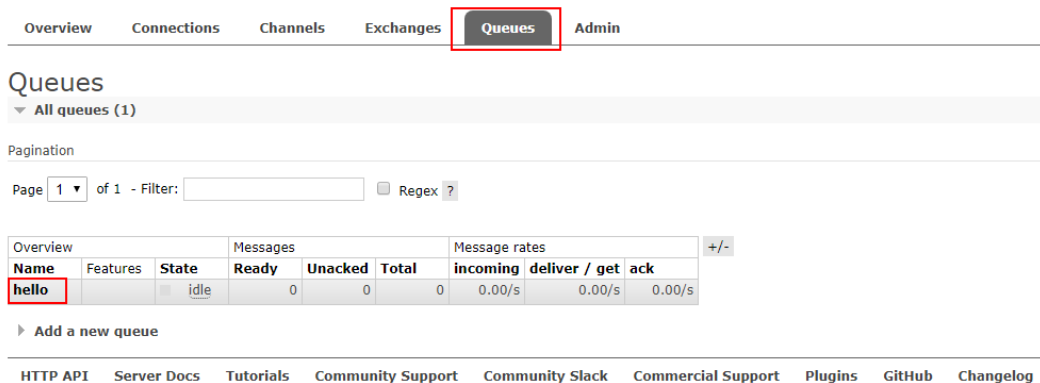
- 步骤3** 在管理控制台左上角单击 ，选择“应用服务 > 分布式消息服务 RabbitMQ”，进入分布式消息服务RabbitMQ专享版页面。
- 步骤4** 单击实例名称，进入实例详情页面。
- 步骤5** 在左侧导航栏选择“Vhost列表”，进入Vhost列表页面。
- 步骤6** 单击Vhost名称，进入Vhost详情页。
- 步骤7** 在“Queue”页签中，在待清空消息的Queue所在行，单击“清空消息”，弹出“清空消息”对话框。
- 步骤8** 单击“确定”，完成Queue的清空消息。

----结束

清空 Queue 消息 (RabbitMQ WebUI)

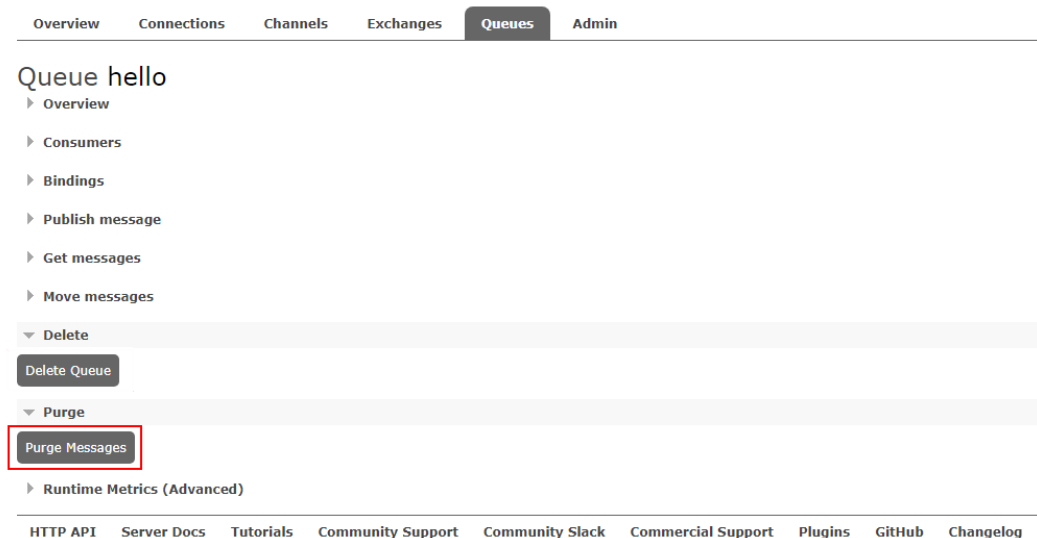
- 步骤1** 登录[RabbitMQ WebUI](#)。
- 步骤2** 在“Queues”页签，单击需要清空数据的队列名称，进入队列详情页面。

图 6-8 Queues



- 步骤3** 单击“Purge Messages”，清空队列数据。

图 6-9 清空队列数据



----结束

6.8.3 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“应用服务 > 分布式消息服务 RabbitMQ”，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

6.8.4 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。

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”。

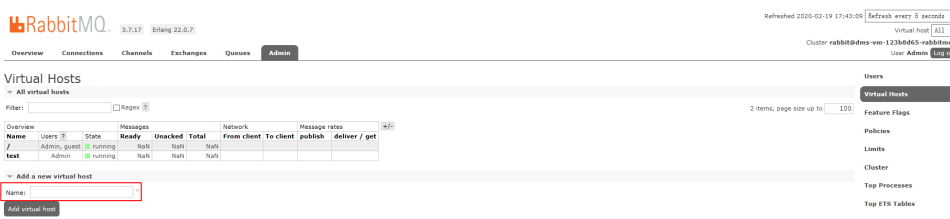
图 6-10 选择 Admin 菜单



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 6-11 创建 Vhost



步骤4 选择右侧导航栏“Policies”，为Vhost设置策略。

图 6-12 设置 Vhost 策略

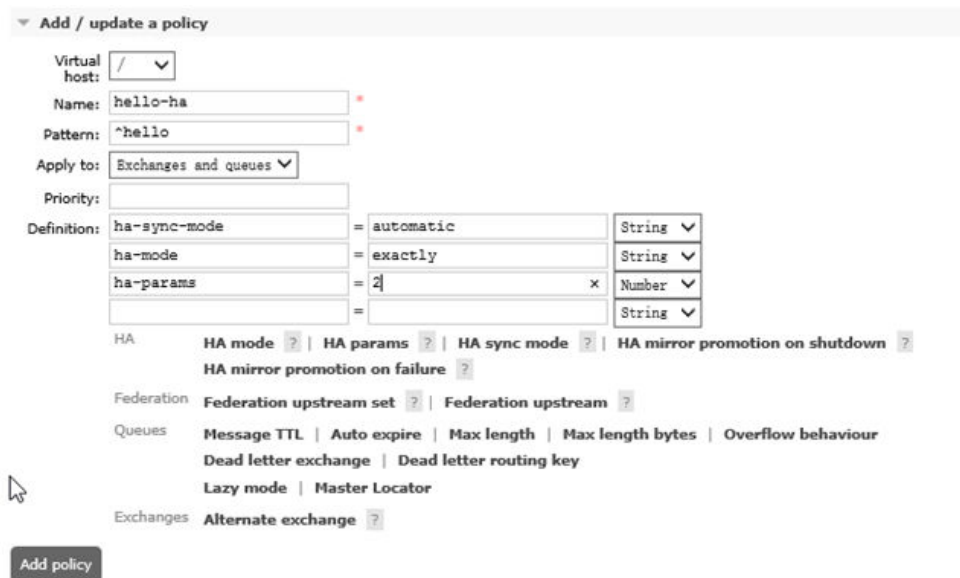


表 6-6 策略参数说明

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。
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 6-13 Vhost 策略



---结束

6.8.5 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

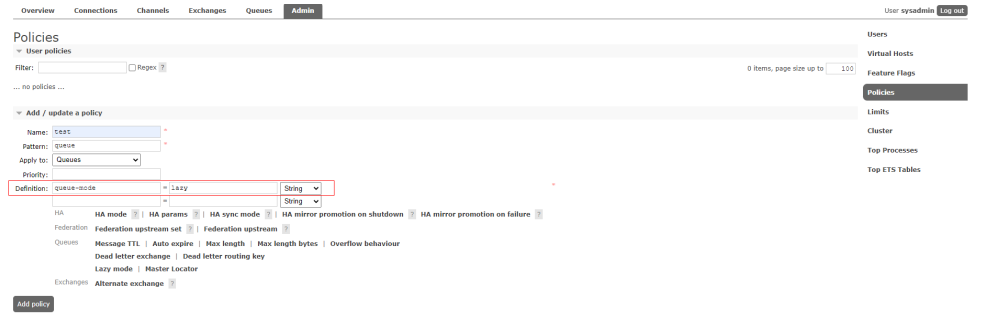
队列具备两种模式：default和lazy，默认模式为default。lazy模式即为惰性队列的模式，可以通过调用channel.queueDeclare方法的时候在参数中设置，也可以通过Policy的方式设置。如果一个队列同时使用这两种方式设置的话，Policy的方式具备更高的优先级。

- 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 6-14 通过 Policy 的方式设置惰性队列



6.8.6 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6-7所示，且消耗更多的内存和磁盘。

表 6-7 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
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 6-15 设置仲裁队列

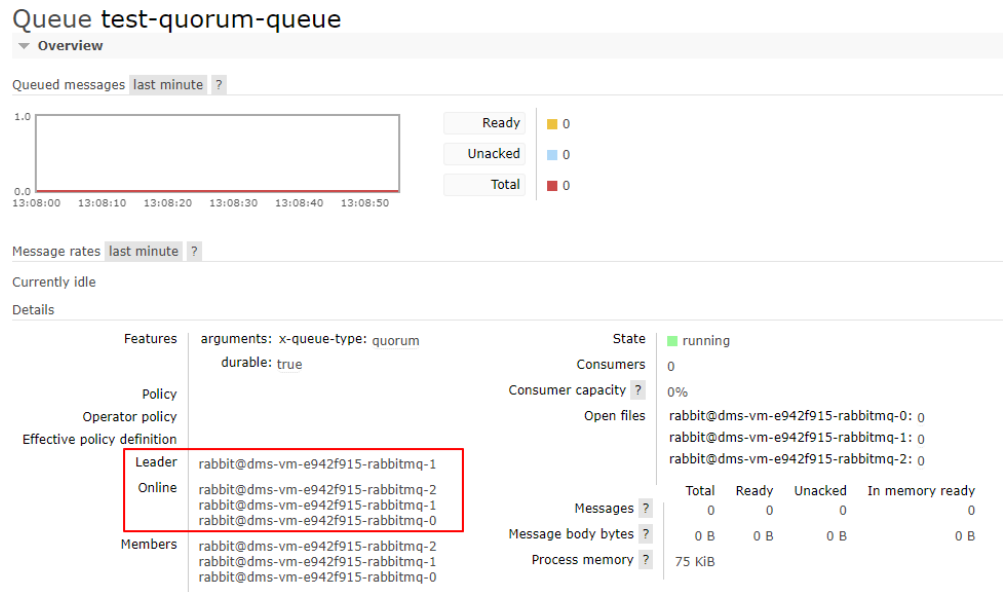
设置完成后，在“Queues”页面查看队列类型是否为“quorum”，如图6-16所示。“Node”中的“+2”表示该队列有2个副本，蓝色表示这两个副本消息同步已经完成，如果为红色则表示部分消息还未同步。

图 6-16 查看队列类型

Overview						Messages			Message rates			
Name	Node	Type	Features	State	Ready	Unacked	Total	incoming	deliver	get	ack	
test-quorum-queue	rabbit@dms-vm-e942f915-rabbitmq-1 +2	quorum	D Args	running	0	0	0	0	0	0	0	

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 6-17 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 6-18 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 6-19 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](#) ?

6.8.7 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。

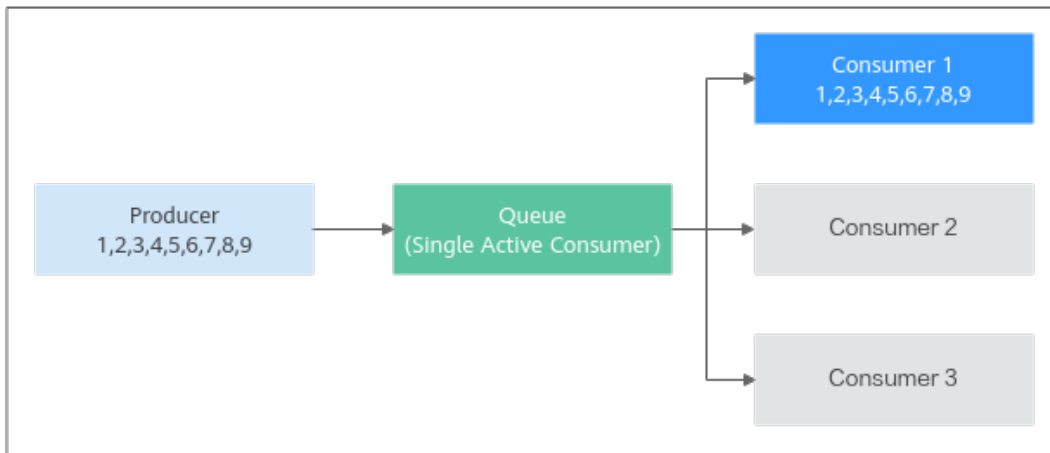
说明

仅RabbitMQ 3.8.35版本支持单一活跃消费者特性。

如图6-20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 6-20 单一活跃消费者消费流程



配置单一活跃消费者

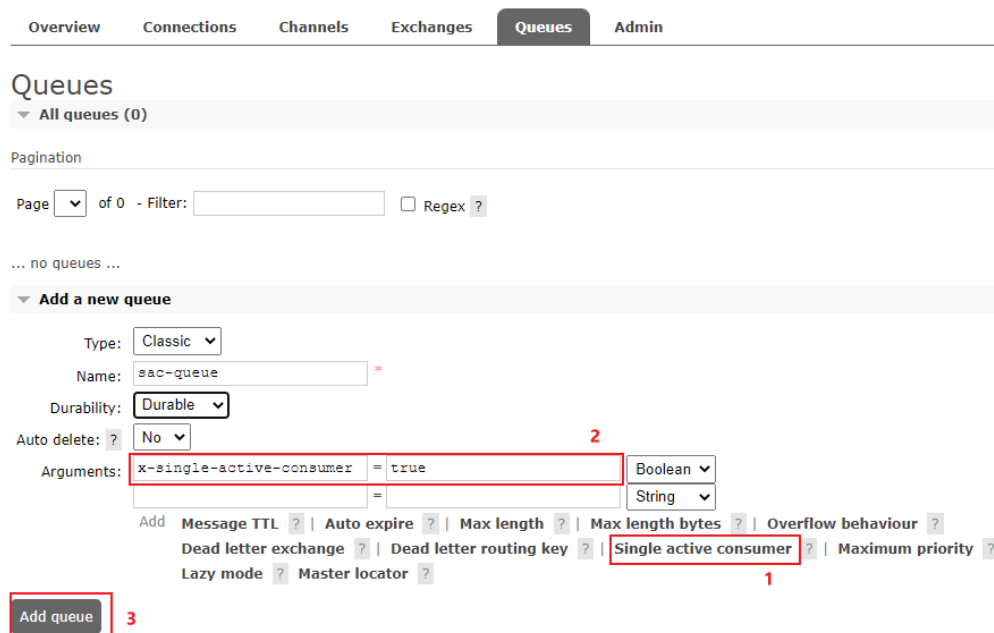
在声明队列时，可以配置单一活跃消费者，只需要将队列的“x-single-active-consumer”参数设置为“true”。

- 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 6-21 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 6-22 所示，“SAC”即表示队列已设置单一活跃消费者。

图 6-22 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 x-single-active-consumer: true

6.8.8 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）

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“应用服务 > 分布式消息服务 RabbitMQ”，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。

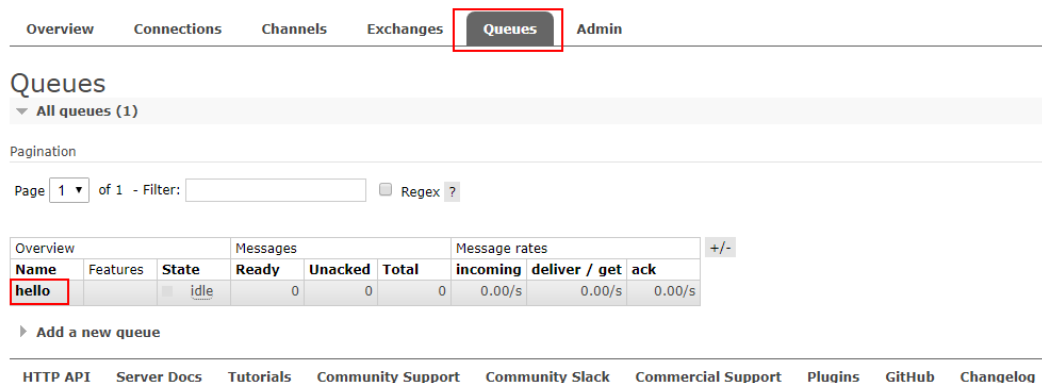
----结束

删除单个 Queue (RabbitMQ WebUI)

步骤1 [登录RabbitMQ WebUI页面](#)。

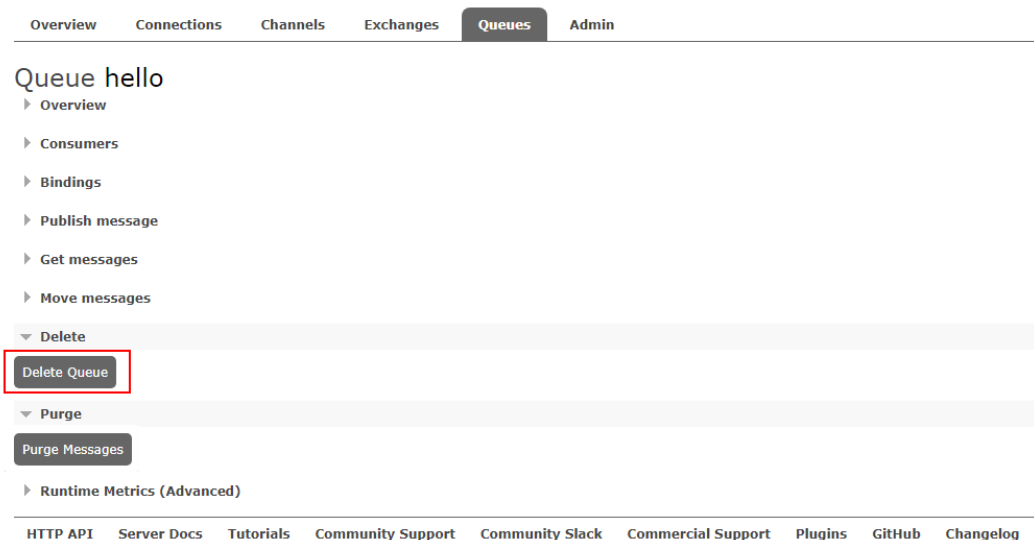
步骤2 在“Queues”页签，单击需要删除的队列名称，进入队列详情页面。

图 6-23 队列列表



步骤3 单击“Delete Queue”，删除单个队列。

图 6-24 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 6-25 通过 Policy 批量删除队列

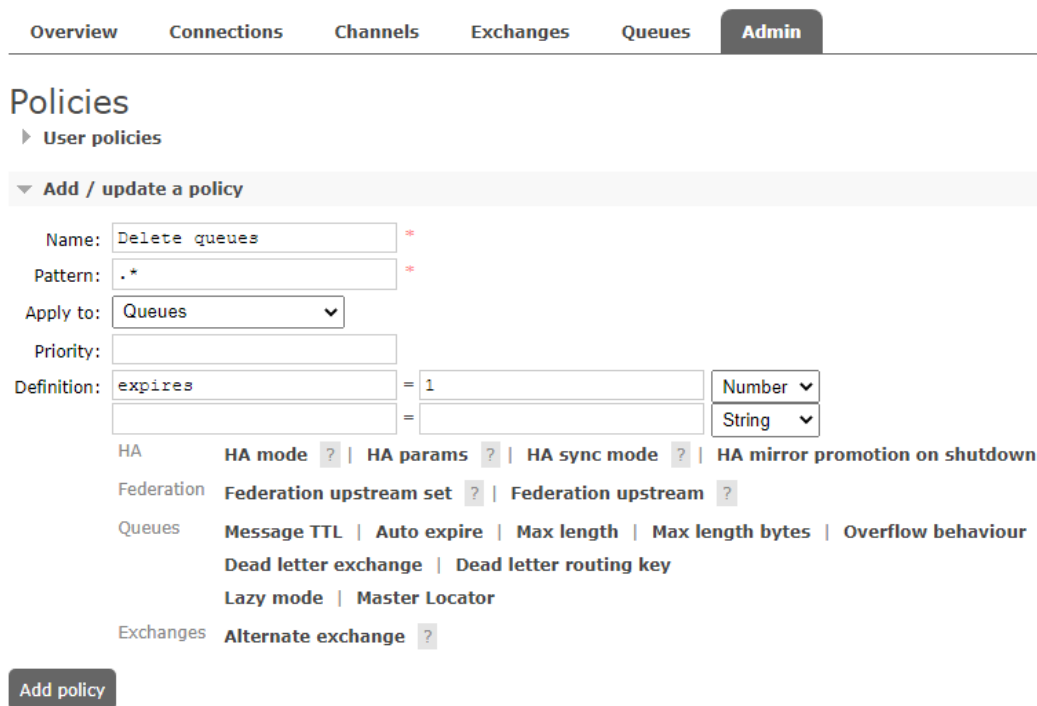


表 6-8 策略参数说明

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。

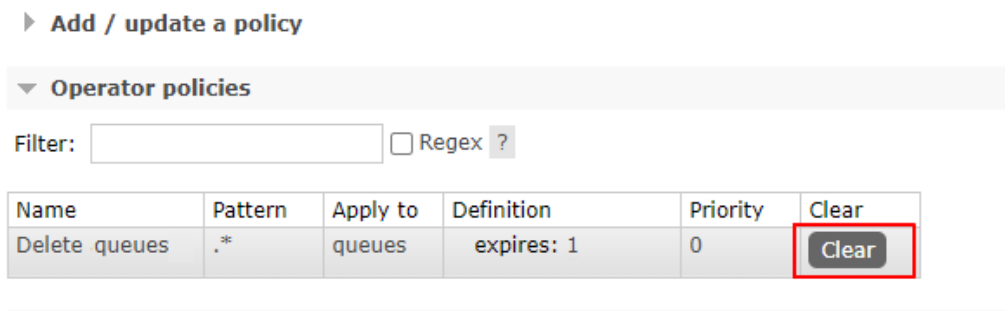
步骤3 单击“Add policy”。

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 6-26 删除策略



----结束

7 连接实例

7.1 配置 RabbitMQ 网络连接

7.1.1 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 7-1 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。	VPC对等连接说明

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7-2](#)的规则。

表 7-2 安全组规则

方向	类型	协议	端口	源地址	说明
入方向	IPv4	TCP	5672	RabbitMQ 客户端所在的IP地址或地址组	客户端使用IPv4地址访问 RabbitMQ实例（关闭SSL加密）
入方向	IPv4	TCP	5671	RabbitMQ 客户端所在的IP地址或地址组	客户端使用IPv4地址访问 RabbitMQ实例（开启SSL加密）
入方向	IPv4	TCP	15672	RabbitMQ 客户端所在的IP地址或地址组	访问Web界面UI地址（关闭SSL加密）
入方向	IPv4	TCP	15671	RabbitMQ 客户端所在的IP地址或地址组	访问Web界面UI地址（开启SSL加密）

7.1.2 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

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“应用服务 > 分布式消息服务 RabbitMQ”，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

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“应用服务 > 分布式消息服务 RabbitMQ”，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

7.2 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](#)。

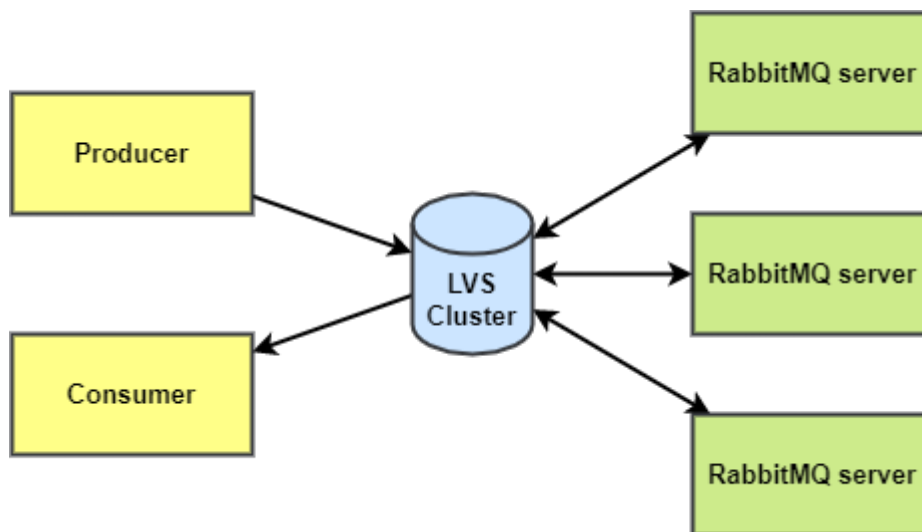
📖 说明

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7-1所示，单节点实例不涉及LVS。

图 7-1 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
$conn = new AMQPStreamConnection(RMQ_HOST, RMQ_PORT, RMQ_USER, RMQ_PASS,
RMQ_vhost, ['heartbeat'=> 60]);
```

7.3 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);
```

7.4 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
```

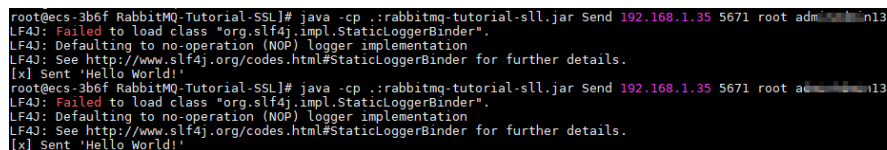
步骤5 运行生产消息示例。

```
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 7-2 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 7-3 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);
```

8 管理消息

8.1 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意一种情况，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);
```

8.2 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](#)。

8.3 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。

9 配置高级特性

9.1 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。

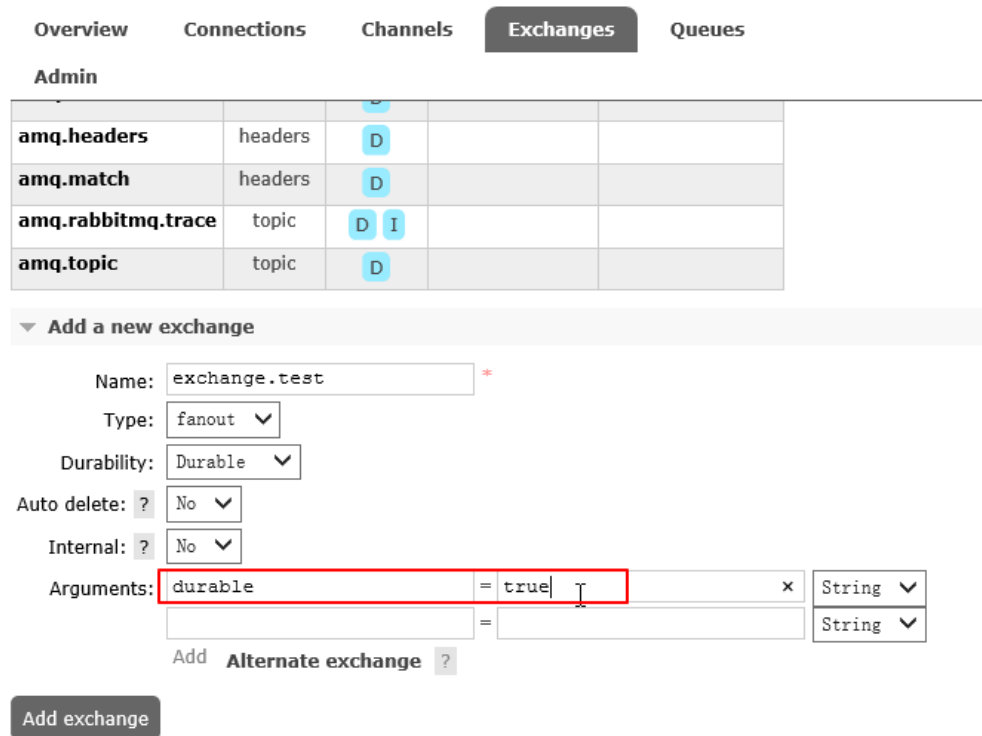
说明

- 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。

设置 Exchange 持久化

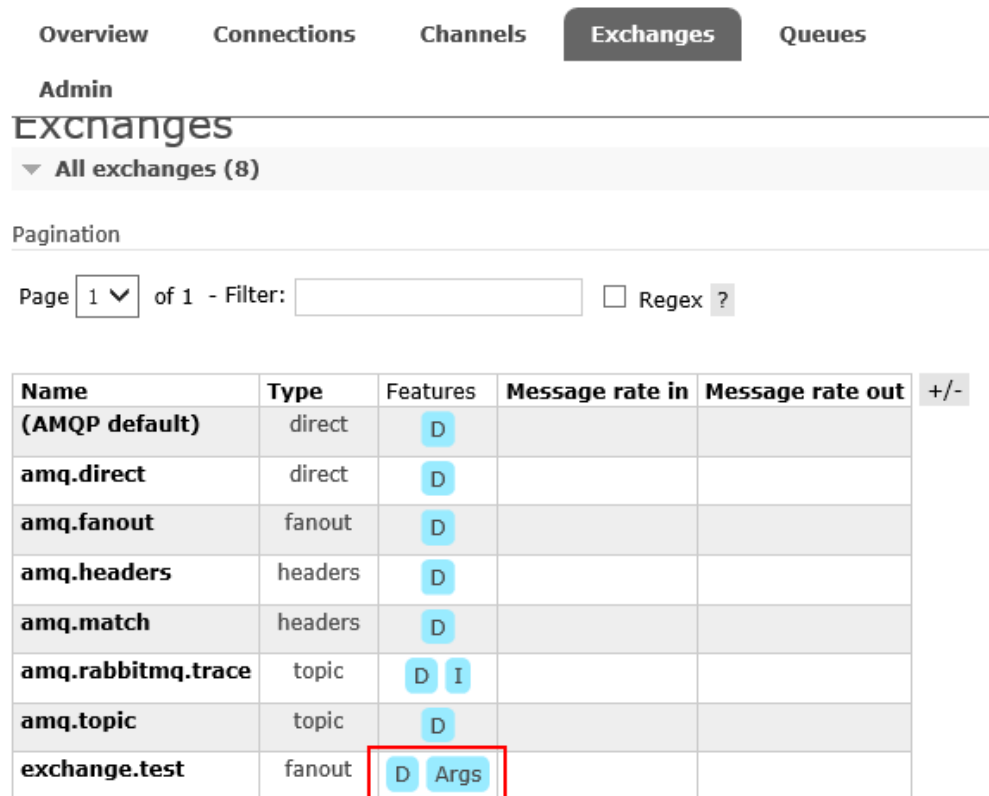
- 在[RabbitMQ WebUI页面](#)设置Exchange持久化。
创建Exchange时，设置“durable”为“true”，如[图9-1](#)所示。

图 9-1 设置 Exchange 持久化 (WebUI)



设置成功后如图9-2所示。

图 9-2 持久化的 Exchange (WebUI)

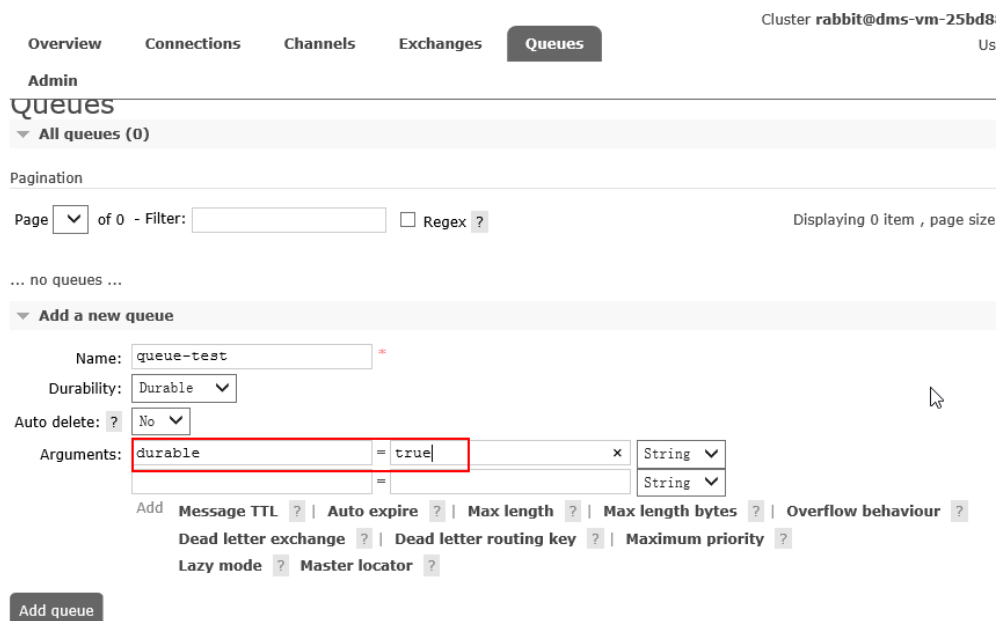


- 在RabbitMQ实例控制台设置Exchange持久化。
创建Exchange时，设置Exchange持久化，如所示。
设置成功后如所示。

设置 Queue 持久化

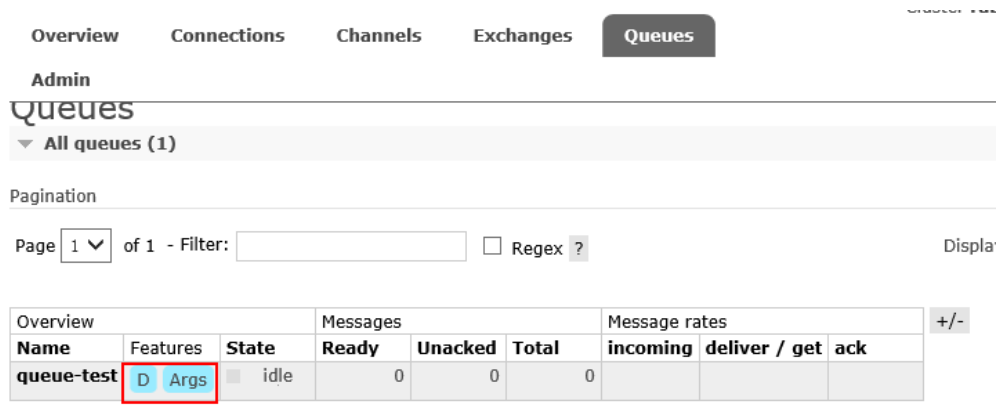
- 在RabbitMQ WebUI页面设置Queue持久化。
创建Queue时，设置“durable”为“true”，如图9-3所示。

图 9-3 设置 Queue 持久化 (WebUI)



设置成功后如图9-4所示。

图 9-4 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，如所示。
设置成功后如所示。

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());
```

9.2 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);
```

10 管理实例

10.1 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“应用服务 > 分布式消息服务 RabbitMQ”，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10-1](#)。

表 10-1 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。
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。

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10-2为连接实例的相关参数，其他参数，请查看页面显示。


表 10-2 连接参数说明

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。

---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“应用服务 > 分布式消息服务 RabbitMQ”，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 10-3 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。
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。

----结束

10.2 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

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10-1](#)所示。

图 10-1 客户端连接地址

The screenshot shows the 'Connections' page in the RabbitMQ management interface. It displays a table with columns for Overview, Details, and Network. The 'Overview' column includes Name, Node, User name, and State. The 'Details' column includes SSL / TLS, Protocol, and Channels. The 'Network' column includes From client, To client, and To client. Four rows are visible, with the first three rows having their IP addresses highlighted in red: 10.234.177.66:50996, 10.234.177.66:53332, and 10.234.177.66:56272. The fourth row has IP address 172.31.1.152:5004.

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10-1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10-2所示。

图 10-2 客户端连接地址（分区生产者/消费者 IP 地址）

The screenshot shows the 'Connections' page in the RabbitMQ management interface. It displays a table with columns for Overview, Details, and Network. The 'Overview' column includes Name, User name, and State. The 'Details' column includes SSL / TLS, Protocol, and Channels. The 'Network' column includes From client, To client, Heartbeat, and Connected at. Two rows are visible, with their IP addresses highlighted in red: 10.234.177.66:65260 (consumer) and 10.234.177.66:58373 (producer).

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

10.3 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。


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10-4所示。

表 10-4 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。
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“应用服务 > 分布式消息服务 RabbitMQ”，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

10.4 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。

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“应用服务 > 分布式消息服务 RabbitMQ”，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

10.5 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](#)。

RabbitMQ支持的插件如[表10-5](#)所示。其中，**插件端口号不支持修改**。

表 10-5 插件列表

插件名称	功能描述	端口号
rabbitmq_amqp1_0	表示实例是否支持AMQP1.0协议。	-
rabbitmq_delayed_message_exchange	表示实例是否开启消息延迟功能。插件延迟时间存在1%左右的误差，可能提前或者推迟发送消息给消费者。	-
rabbitmq_federation	表示实例是否开启消息同步功能。	-
rabbitmq_sharding	表示实例是否开启消息分片功能。	-
rabbitmq_shovel	表示实例是否开启消息迁移功能。	-
rabbitmq_tracing	表示实例是否开启消息追踪功能。	-
rabbitmq_mqtt	表示实例是否支持MQTT协议（TCP方式）。	1883
rabbitmq_web_mqtt	表示实例是否支持MQTT协议（WebSocket方式）。	15675
rabbitmq_stomp	表示实例是否支持STOMP协议（TCP方式）。	61613
rabbitmq_web_stomp	表示实例是否支持STOMP协议（WebSocket方式）。	15674
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

- 开启插件过程中，不会重启实例，但是以下插件（rabbitmq_mqtt、rabbitmq_web_mqtt、rabbitmq_stomp、rabbitmq_web_stomp）会重启keepalived，导致连接断开。连接断开后，是否会自动重连依赖于用户自身的业务逻辑。
- rabbitmq_shovel、rabbitmq_federation和rabbitmq_tracing插件在特定场景下才支持开启，具体如表10-6所示。


表 10-6 支持开启插件的场景

实例	rabbitmq_shovel	rabbitmq_federation	rabbitmq_tracing
未开启SSL的单机实例	支持	支持	支持
已开启SSL的单机实例	不支持	不支持	不支持

实例	rabbitmq_shovel	rabbitmq_federation	rabbitmq_tracing
未开启SSL的集群实例	不支持	支持	支持
已开启SSL的集群实例	不支持	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“应用服务 > 分布式消息服务 RabbitMQ”，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

10.6 使用 rabbitmq_tracing 插件

rabbitmq_tracing插件提供消息追踪功能，它能追踪流入流出RabbitMQ的消息，并对其封装，将封装后的消息日志存入相应的trace文件中。

未开启SSL的RabbitMQ单机实例，支持开启rabbitmq_shovel插件。RabbitMQ集群实例和开启SSL的RabbitMQ单机实例，不支持开启rabbitmq_shovel插件。

操作影响

- trace日志文件存在打满磁盘空间的风险，在高负载状态下建议不要开启rabbitmq_tracing插件。
- trace日志文件占用的磁盘空间在插件关闭后才会被释放，建议不要长期开启该插件，定位完成后关闭trace任务和插件。

前提条件

已购买实例。

使用 rabbitmq_tracing 插件

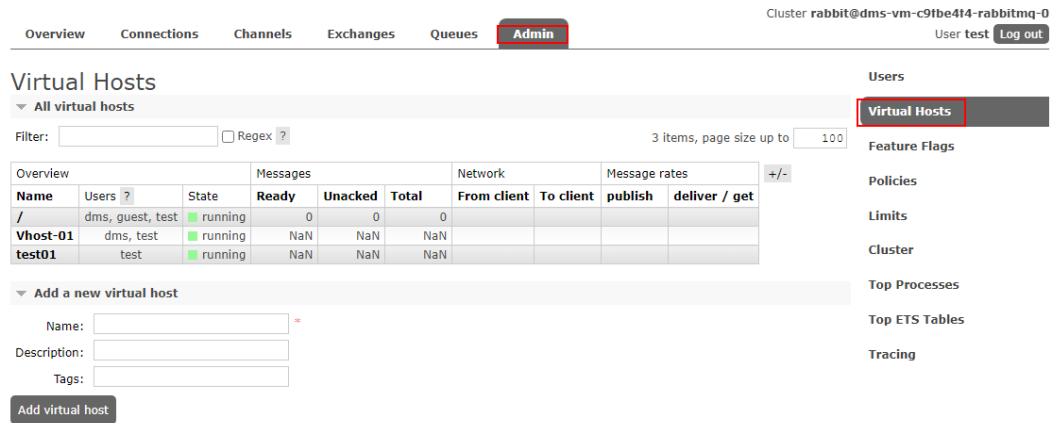
步骤1 开启rabbitmq_tracing插件，具体步骤请参考[开启RabbitMQ实例插件](#)。

步骤2 登录RabbitMQ WebUI页面。

步骤3 在顶部导航栏选择“Admin”，进入Admin页面。

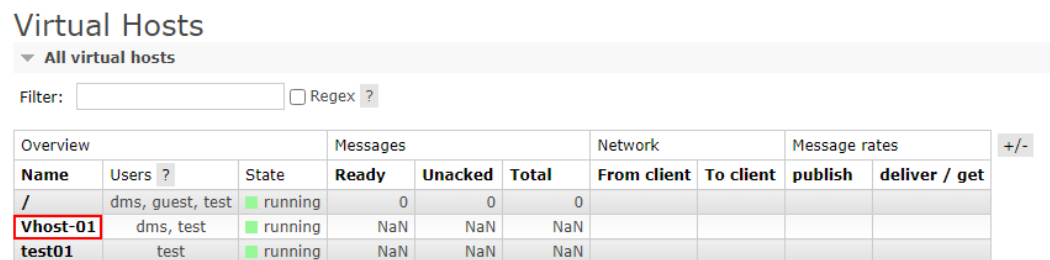
步骤4 在右侧导航栏选择“Virtual Hosts”，进入Virtual Hosts页面。

图 10-3 Virtual Hosts 页面



步骤5 单击待创建trace的Vhost名称，进入Vhost详情页面。

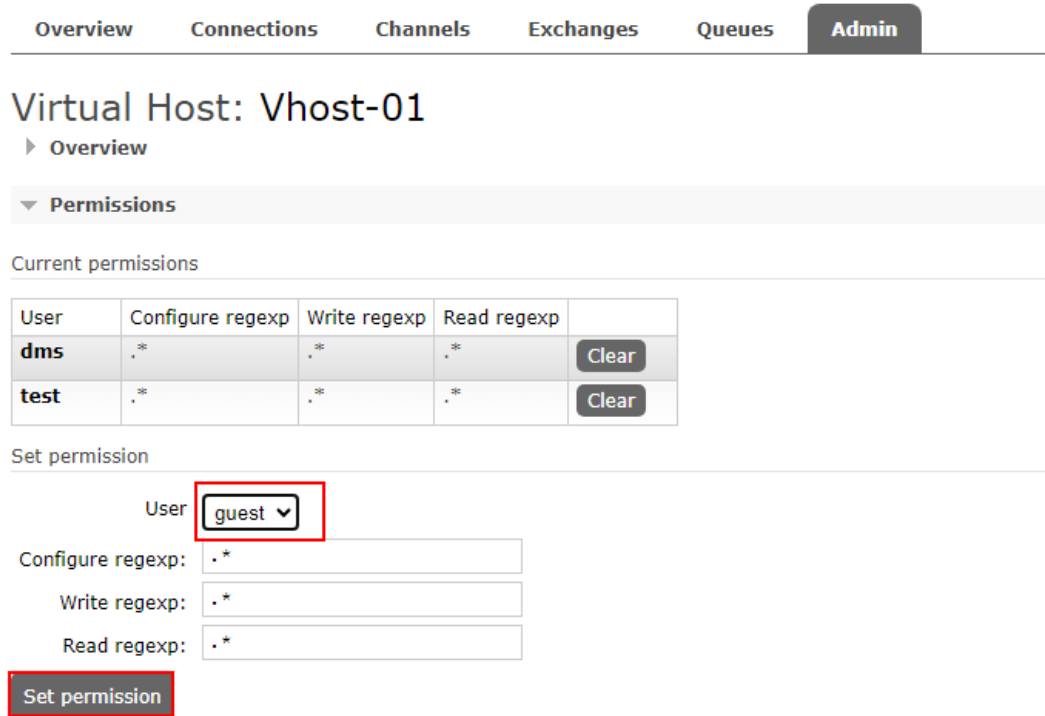
图 10-4 待创建 trace 的 Vhost 名称



步骤6 在“Permissions”区域，为Vhost添加guest权限。

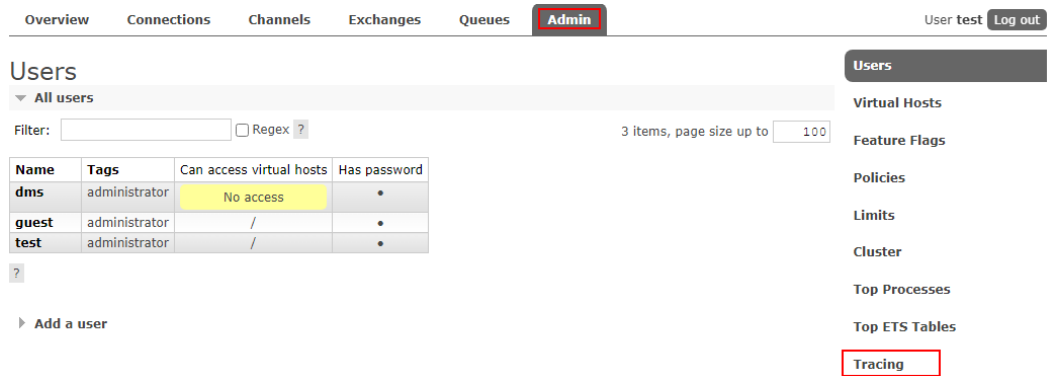
使用trace功能的Vhost必须配置guest权限，否则创建trace时会报错。

图 10-5 为 Vhost 添加 guest 权限



步骤7 在右侧导航栏选择“Tracing”，进入Tracing页面。

图 10-6 Admin 页面



步骤8 在“Add a new trace”区域，输入以下参数，单击“Add trace”，新增一个trace。

表 10-7 trace 参数说明

参数	说明
Virtual host	选择待创建trace的Vhost名称。
Name	自定义trace的名称，用于区分不同的trace。
Format	输出消息日志的格式。支持“Text”和“JSON”两种格式，“Text”格式方便阅读，“JSON”格式方便解析。

参数	说明
Tracer connection username	指定创建trace的用户名。
Tracer connection password	指定创建trace的密码。
Max payload bytes	每条消息的最大限制，单位为B。 假设“Max payload bytes”设置为“10”，当有超过10B的消息经过RabbitMQ流转时就会被载断，例如“trace test payload”会被载断成“trace test”。
Pattern	设置匹配的模式。取值示例如下： <ul style="list-style-type: none"> • #：追踪所有进入和离开RabbitMQ的消息 • publish.#：追踪所有进入RabbitMQ的消息 • deliver.#：追踪所有离开RabbitMQ的消息 • publish.delay_exchange：追踪进入指定交换机的信息，delay_exchange为交换机名称，请根据实际情况修改。 • deliver.delay_queue：追踪离开指定队列的消息，delay_queue为队列名称，请根据实际情况修改。

图 10-7 新增一个 trace

trace创建成功后，在“**All traces**”区域，显示已创建的trace列表。

图 10-8 trace 列表

Traces: rabbit@dms-vm-c9fbe4f4-rabbitmq-0

Node: rabbit@dms-vm-c9fbe4f4-rabbitmq-0

▼ All traces

Currently running traces								Trace log files		
Virtual host	Name	Pattern	Format	Payload limit	Rate	Queued	Tracer connection username	Name	Size	Delete
Vhost-01	delay_queue_trace	deliver.delay_queue	text	Unlimited		0 (queue)		delay_queue_trace.log	0 B	Delete
Vhost-01	delay_exchange_trace	publish.delay_exchange	text	Unlimited		0 (queue)		delay_exchange_trace.log	0 B	Delete

步骤9（可选）如果RabbitMQ实例为集群，在“**Node**”中切换到其他节点，重复**步骤8**，为其他所有节点创建trace。

图 10-9 切换节点

Traces: rabbit@dms-vm-c9fbe4f4-rabbitmq-0

Node: rabbit@dms-vm-c9fbe4f4-rabbitmq-0 ▼

步骤10 当trace日志文件中存入消息日志后，单击trace日志文件名称，查看日志内容。

图 10-10 trace 日志文件

Traces: rabbit@dms-vm-c9fbe4f4-rabbitmq-0

Node: rabbit@dms-vm-c9fbe4f4-rabbitmq-0 ▼

▼ All traces

Currently running traces

Virtual host	Name	Pattern	Format	Payload limit	Rate	Queued	Tracer connection username	
Vhost-01	delay_queue_trace	deliver.delay_queue	text	Unlimited		0 (queue)		Stop
Vhost-01	delay_exchange_trace	publish.delay_exchange	text	Unlimited		0 (queue)		Stop

Trace log files

Name	Size	
delay_queue_trace.log	0 B	Delete
delay_exchange_trace.log	0 B	Delete

“delay_exchange_trace.log” 的日志内容如图10-11所示。

图 10-11 delay_exchange_trace.log

```
=====
2022-07-20 3:22:32:837: Message published

Node:          rabbit@dms-vm-3492b4ba-rabbitmq-0
Connection:    <rabbit@dms-vm-3492b4ba-rabbitmq-0.1657790484.10274.7>
Virtual host:  /
User:         admin
Channel:      1
Exchange:     delay_exchange
Routing keys: [<<>>]
Routed queues: []
Properties:   [{<<"delivery_mode">>,signedint,2},{<<"headers">>,table,[]}]
Payload:      hello world
```

“delay_queue_trace.log” 的日志内容如图10-12所示。

图 10-12 delay_queue_trace.log

```
=====
2022-07-20 3:23:22:468: Message received

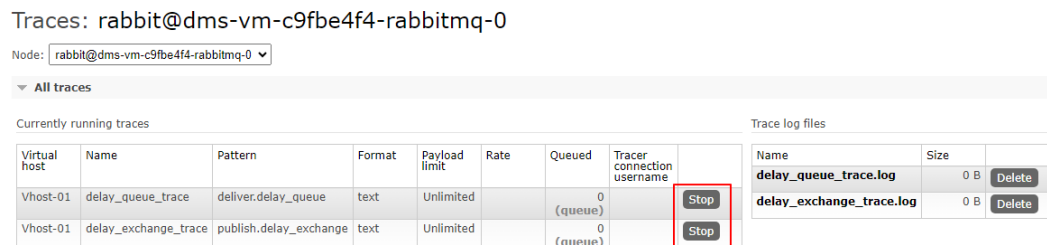
Node:          rabbit@dms-vm-3492b4ba-rabbitmq-0
Connection:    <rabbit@dms-vm-3492b4ba-rabbitmq-0.1657790484.10565.7>
Virtual host:  /
User:         admin
Channel:      1
Exchange:     delay_queue
Routing keys: [<<"delay_queue">>]
Queue:        delay_queue
Properties:   [{<<"delivery_mode">>,signedint,1},{<<"headers">>,table,[]}]
Payload:      hello world
```

----结束

关闭 trace 任务和插件

步骤1 在Tracing页面的“All traces”区域，单击“Stop”，关闭trace任务。

图 10-13 关闭 trace 任务



步骤2 在RabbitMQ实例控制台的“插件管理”页面，单击rabbitmq_tracing后的“关闭”，弹出“关闭插件”对话框。

步骤3 单击“是”，跳转到“后台任务管理”页面，任务状态变为“成功”时，表示成功关闭rabbitmq_tracing插件。


----结束

10.7 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“应用服务 > 分布式消息服务 RabbitMQ”，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，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

10.8 删除 RabbitMQ 实例

分布式消息服务RabbitMQ版管理控制台支持删除RabbitMQ实例，且可实现批量删除RabbitMQ实例。

须知


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。

前提条件

RabbitMQ实例状态为运行中、故障、创建失败时，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“应用服务 > 分布式消息服务 RabbitMQ”，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“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、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

10.9 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。


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“应用服务 > 分布式消息服务 RabbitMQ”，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。

步骤2 确认实例安全组规则是否配置正确。

1. 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。
2. 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。

图 10-14 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

11 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11-1所示。

表 11-1 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	√	√

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“应用服务 > 分布式消息服务 RabbitMQ”，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**
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）。

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，建议在业务低峰时进行变更。

- **扩容/缩容代理规格**
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。

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。

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

12 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和消费者，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。

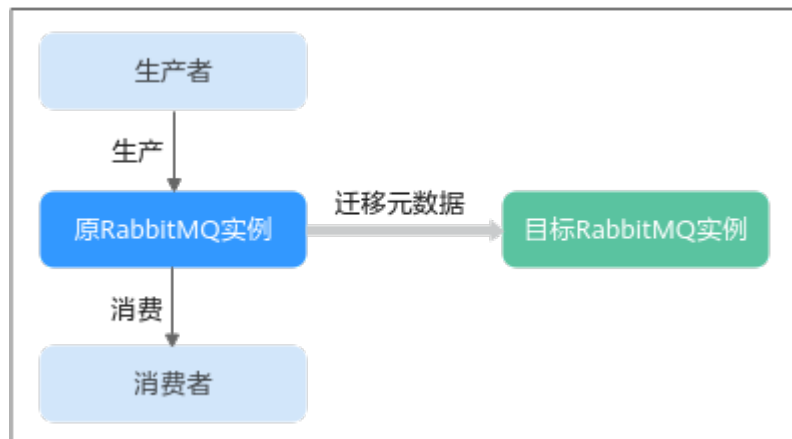
前提条件

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。

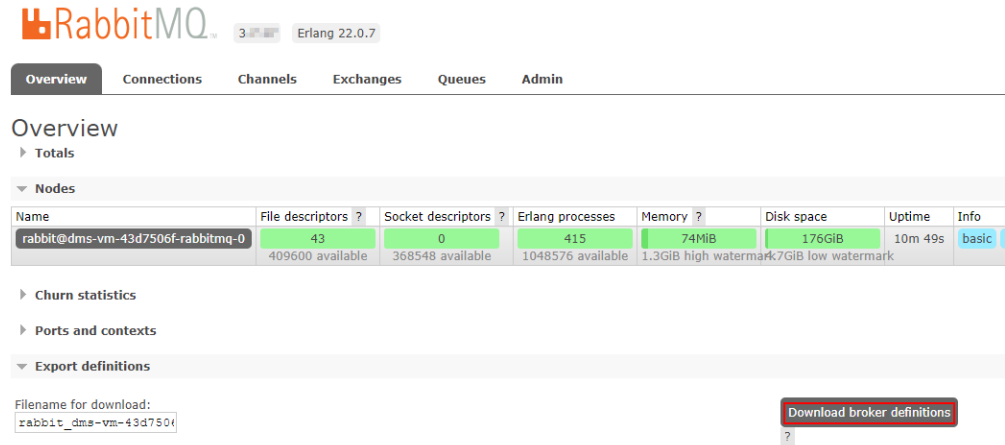
图 12-1 迁移元数据



迁移步骤如下：

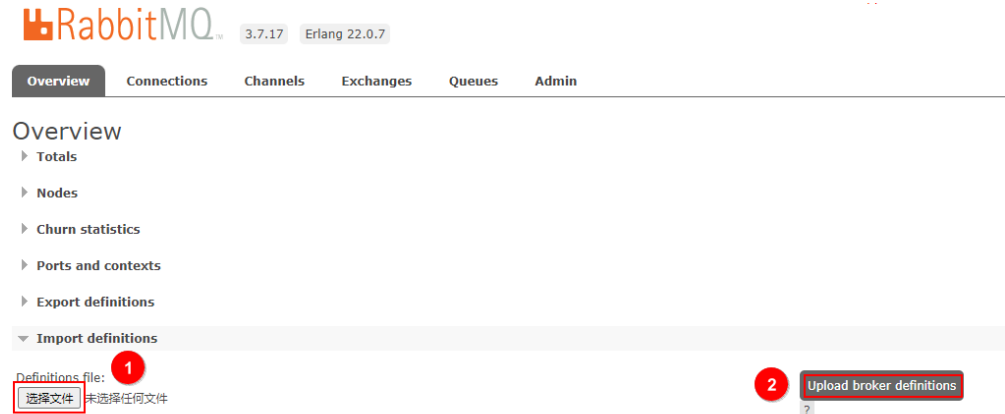
1. 登录原RabbitMQ的WebUI页面，在“Overview”页签中，单击“Download broker definitions”，导出元数据。

图 12-2 导出元数据



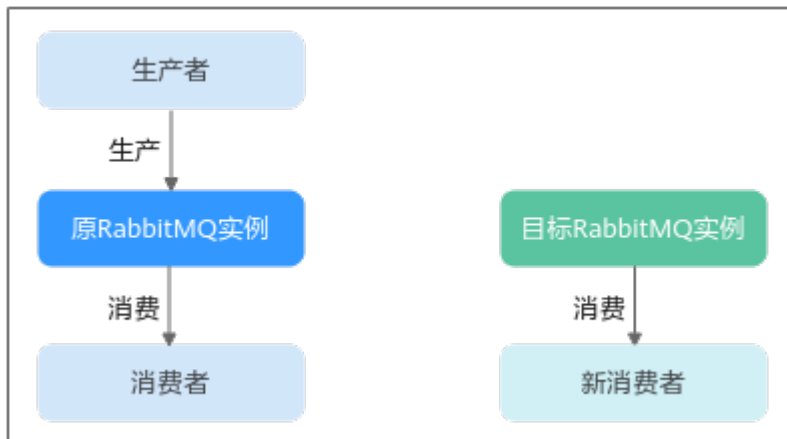
2. 登录目标RabbitMQ的WebUI页面，在“Overview”页签中，单击“选择文件”，选择步骤1.1中导出的元数据，单击“Upload broker definitions”，上传元数据。

图 12-3 导入元数据



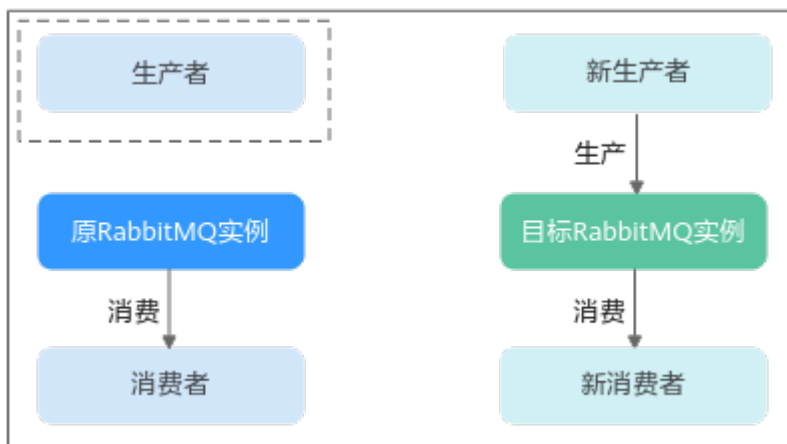
步骤2 为目标RabbitMQ实例添加新的消费者，准备消费目标实例的消息。

图 12-4 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 12-5 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 12-6 下线旧的消费者和原 RabbitMQ 实例



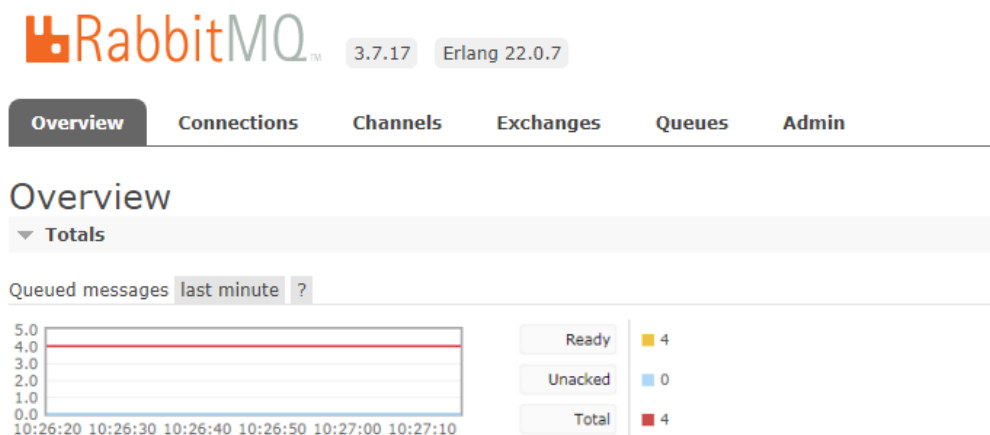
----结束

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12-7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 12-7 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 12-8 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
  }
},

```



13 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. 单击页面右上角的“My Quota”图标 。
系统进入“服务配额”页面。
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？

目前系统暂不支持在线调整配额大小。如您需要调整配额，请拨打热线或发送邮件至客服，客服会及时为您处理配额调整的需求，并以电话或邮件的形式告知您实时进展。

在拨打热线或发送邮件之前，请您准备好以下信息：

- Account Name、Project Name、Project ID。
获取方式如下：登录云账户管理控制台，在右上角单击账户名，选择“我的认证”，在“我的认证”页面获取相关信息。
- 配额信息，包括：
 - 服务名
 - 配额类别
 - 需要的配额值

如您需要调整配额，请联系管理员。

14 查看监控指标与配置告警

14.1 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“应用服务 > 分布式消息服务 RabbitMQ”，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”，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”。

----结束

14.2 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 14-1 实例支持的监控指标

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)
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
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
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	0~100000	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)
message_s_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

节点监控指标

表 14-2 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	0~5000	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

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)
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	0~320000000	RabbitMQ实例节点	1分钟
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	0~500000000	RabbitMQ实例节点	1分钟

队列监控指标

表 14-3 队列支持的监控指标

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)
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

维度

Key	Value
rabbitmq_instance_id	RabbitMQ实例
rabbitmq_node	RabbitMQ实例节点
rabbitmq_queue	RabbitMQ实例队列

14.3 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 14-4 RabbitMQ 实例配置告警的指标


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
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

📖 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 监控股警

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“应用服务 > 分布式消息服务 RabbitMQ”，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，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”，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

15 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 15-1 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
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	deleteDMSCreateFailureInstanceSuccess
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

操作名称	资源类型	事件名称
批量删除DMS实例任务	rabbitmq	batchDeleteDMSInstanceTask
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

查看审计日志

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

16 常见问题

16.1 实例问题

16.1.1 RabbitMQ 使用的版本是多少？

服务端RabbitMQ的版本是3.8.35。

16.1.2 RabbitMQ 实例 SSL 连接的协议版本号是多少？

TLS v1.2版本。

16.1.3 创建实例时为什么无法查看子网和安全组等信息？

创建实例时，如果无法查看虚拟私有云、子网、安全组、弹性IP，可能原因是该用户无Server Administrator和VPC Administrator权限，增加权限的详细步骤请参考[修改用户组权限](#)。

16.1.4 RabbitMQ 集群实例如何均衡分发请求到每个虚拟机？

集群内部使用LVS做负载均衡，由LVS将请求均衡分发到每个虚拟机节点。

16.1.5 RabbitMQ 实例集群内部的队列是否有冗余备份？

队列是否做镜像（即冗余备份）取决于用户的需要，如果用户设置了镜像，会在集群中多个代理上存储队列的副本，当某个代理故障，集群会从其他正常的代理中选择一个代理，用来同步队列数据。

16.1.6 RabbitMQ 实例是否支持持久化，如何定时备份数据？

RabbitMQ支持消息数据持久化，可从客户端连接RabbitMQ并设置消息持久化，也可在RabbitMQ集群管理工具界面创建队列时设置消息持久化。

不支持客户自定义定时备份数据，或从界面触发备份数据。

16.1.7 RabbitMQ 实例开启 SSL 后，证书怎么获取？

RabbitMQ 3.x.x版本实例开启SSL后只做单向认证，不需要证书。

16.1.8 RabbitMQ 实例的 SSL 开关是否支持修改？

不支持动态修改，即如果实例创建时没有选择开启，创建完成之后，不支持修改，建议在实例创建时将开关打开。

16.1.9 RabbitMQ 实例是否支持扩容？

单机版本的RabbitMQ实例支持扩大存储空间，以及扩容/缩容代理规格。

集群版本的RabbitMQ实例支持扩大存储空间和代理个数，以及扩容/缩容代理规格。

16.1.10 RabbitMQ 支持双向认证吗？

不支持。

16.1.11 RabbitMQ 支持升级 CPU 和内存吗？

RabbitMQ支持扩容/缩容代理规格，具体请参见[变更实例规格](#)。

16.1.12 如何关闭 RabbitMQ 的 WebUI？

创建RabbitMQ实例后，如果想要关闭RabbitMQ的WebUI，只要您在安全组入方向中不开放15672端口（实例未开启SSL时的端口）或者15671（实例开启SSL时的端口），此时就无法登录WebUI界面。

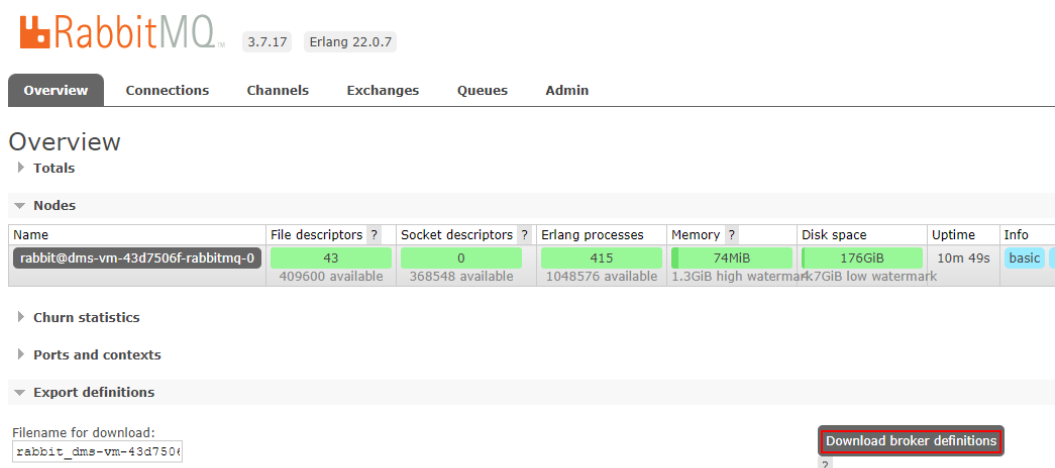
16.1.13 实例是否支持修改可用区？

不支持，您可以重新创建实例，以满足可用区要求，然后进行实例元数据的迁移。

RabbitMQ 3.x.x版本实例元数据的迁移步骤如下：

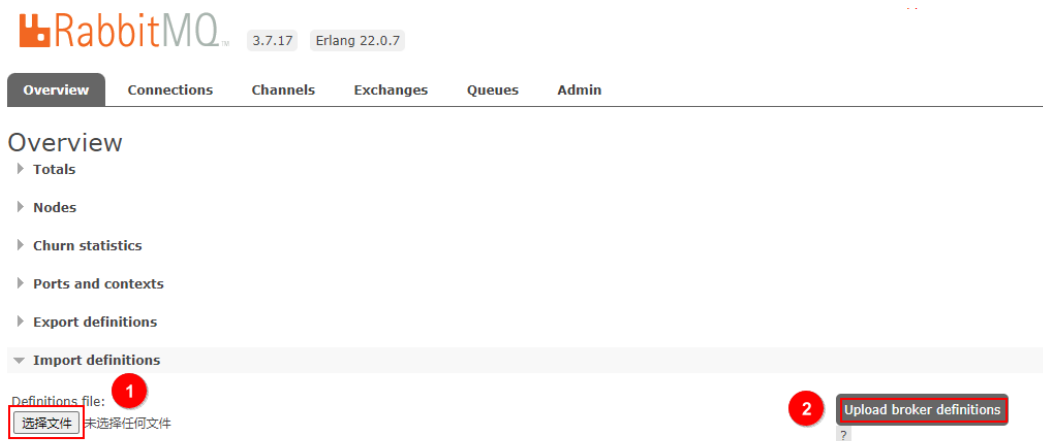
步骤1 登录重新创建前的RabbitMQ实例的WebUI页面。

步骤2 在“Overview”页签中，单击“Download broker definitions”，导出元数据。

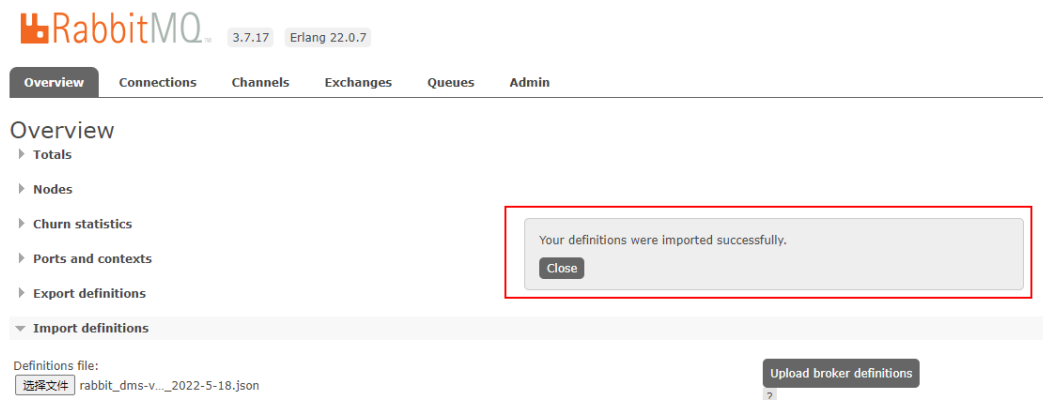


步骤3 登录重新创建的RabbitMQ实例的WebUI页面，在“Overview”页签中，单击“选择文件”，选择步骤2中导出的元数据。

步骤4 单击“Upload broker definitions”，上传元数据。



上传成功后，显示如下信息。



----结束

16.1.14 如何获取 region id?

获取region id的方法如下：

- 步骤1 进入[地区和终端节点](#)页面。
- 步骤2 页面表格中“区域”列，即为region id。

----结束

16.1.15 为什么不能选择 2 个可用区?

不支持选择2个可用区的原因如下：使用2个可用区的实例存在脑裂的风险。如果您需要提高RabbitMQ集群实例的可靠性，在创建实例时，建议选择3个或以上的可用区。

16.1.16 单机 RabbitMQ 实例如何变更为集群实例?

单机RabbitMQ实例不支持变更为集群实例。如果您需要使用集群RabbitMQ实例，建议重新创建实例，然后将业务迁移到新RabbitMQ实例中。

16.1.17 RabbitMQ 实例创建后，能修改 VPC 和子网吗?

不能修改VPC和子网。

16.2 连接问题

16.2.1 如何配置安全组？

RabbitMQ实例支持VPC内访问和公网访问，配置安全组的方式如下：

- VPC内访问实例

客户端只能部署在与RabbitMQ实例处于相同虚拟私有云（VPC）的弹性云服务器（ECS）上。

除了ECS、RabbitMQ实例必须处于相同VPC之外，还需要他们的安全组分别配置了正确的规则，客户端才能访问RabbitMQ实例。

- 建议ECS、RabbitMQ实例配置相同的安全组。安全组创建后，默认包含组内网络访问不受限制的规则。
- 如果配置了不同安全组，可参考如下配置方式：

📖 说明

- 假设ECS、RabbitMQ实例分别配置了安全组：sg-53d4、Default_All。
- 以下规则，远端可使用安全组，也可以使用具体的IP地址。

ECS所在安全组需要增加如下规则，以保证客户端能正常访问RabbitMQ实例。

图 16-1 配置 ECS 安全组



表 16-1 安全组规则

方向	协议端口	目的地址
出方向	全部放通	Default_All

RabbitMQ实例所在安全组需要增加如下规则，以保证能被客户端访问。

图 16-2 配置 RabbitMQ 实例安全组



表 16-2 安全组规则

方向	协议端口	源地址
入方向	全部放通	sg-53d4

- 通过公网访问实例
RabbitMQ实例所在安全组需要增加如下规则，以保证能被客户端访问。

说明

表16-3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 16-3 安全组规则

方向	协议端口	源地址
入方向	TCP:5672	RabbitMQ客户端所在的IP地址或地址组

具体如图16-3所示。

图 16-3 安全组规则 1



16.2.2 RabbitMQ 客户端连接报错原因分析

RabbitMQ客户端连接失败，可能原因包括地址填错、端口填错、用户名或者密码填错、超过最大连接数、未创建Vhost或者填错Vhost名称。

可能原因一：连接地址不正确

VPC内访问场景下，连接地址不正确时，报错如下：

```
[root@ecs-test RabbitMQ-Tutorial]# java -cp ./rabbitmq-tutorial.jar Send 192.168.125.110 5672 user *****
Exception in thread "main" java.net.NoRouteToHostException: No route to host (Host unreachable)
at java.net.PlainSocketImpl.socketConnect(Native Method)
at java.net.AbstractPlainSocketImpl.doConnect(AbstractPlainSocketImpl.java:350)
at java.net.AbstractPlainSocketImpl.connectToAddress(AbstractPlainSocketImpl.java:206)
```

公网访问场景下，连接地址不正确时，报错如下：

```
[root@ecs-test RabbitMQ-Tutorial]# java -cp ./rabbitmq-tutorial.jar Send 139.xxx.178 5672 user *****
Exception in thread "main" java.net.SocketTimeoutException: connect timed out
at java.net.PlainSocketImpl.socketConnect(Native Method)
at java.net.AbstractPlainSocketImpl.doConnect(AbstractPlainSocketImpl.java:350)
```

解决方法：在RabbitMQ控制台的“基本信息”页面，获取“内网连接地址”/“公网连接地址”，修改连接实例代码中的连接地址。

可能原因二：端口不正确

VPC内访问场景下，端口不正确时，报错如下：

```
[root@ecs-test RabbitMQ-Tutorial]# java -cp ./rabbitmq-tutorial.jar Send 192.168.125.111 5673 user *****
Exception in thread "main" java.net.ConnectException: Connection refused (Connection refused)
at java.net.PlainSocketImpl.socketConnect(Native Method)
at java.net.AbstractPlainSocketImpl.doConnect(AbstractPlainSocketImpl.java:350)
at java.net.AbstractPlainSocketImpl.connectToAddress(AbstractPlainSocketImpl.java:206)
```

公网访问场景下，端口不正确时，报错如下：

```
[root@ecs-test RabbitMQ-Tutorial]# java -cp ./rabbitmq-tutorial.jar Send 139.xxx.179 5673 user *****
Exception in thread "main" java.net.SocketTimeoutException: connect timed out
at java.net.PlainSocketImpl.socketConnect(Native Method)
at java.net.AbstractPlainSocketImpl.doConnect(AbstractPlainSocketImpl.java:350)
at java.net.AbstractPlainSocketImpl.connectToAddress(AbstractPlainSocketImpl.java:206)
```

解决方法：修改端口号。

可能原因三：用户名或密码错误

报错如下：

```
[root@ecs-test RabbitMQ-Tutorial]# java -cp ./rabbitmq-tutorial.jar Send 192.168.125.111 5672 user *****
Exception in thread "main" com.rabbitmq.client.AuthenticationFailureException: ACCESS_REFUSED - Login
was refused using authentication mechanism PLAIN. For details
see the broker logfile.
at com.rabbitmq.client.impl.AMQConnection.start(AMQConnection.java:351)
at
com.rabbitmq.client.impl.recovery.RecoveryAwareAMQConnectionFactory.newConnection(RecoveryAwareAMQ
ConnectionFactory.java:64)
```

解决方法：修改用户名或密码。如果忘记密码，参考[重置实例密码](#)，重置密码。

可能原因四：超过最大连接数

报错如下：

```
Producer Consumer (1)
D:\jdk1.8.0_272\bin\java.exe ...
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Exception in thread "main" java.util.concurrent.TimeoutException: Create breakpoint
    at com.rabbitmq.utility.BlockingCell.get(BlockingCell.java:77)
    at com.rabbitmq.utility.BlockingCell.uninterruptibleGet(BlockingCell.java:120)
    at com.rabbitmq.utility.BlockingValueOrException.uninterruptibleGetValue(BlockingValueOrException.java:36)
    at com.rabbitmq.client.impl.AMQChannel$BlockingRpcContinuation.getReply(AMQChannel.java:502)
    at com.rabbitmq.client.impl.AMQConnection.start(AMQConnection.java:326)
    at com.rabbitmq.client.impl.recovery.RecoveryAwareAMQConnectionFactory.newConnection(RecoveryAwareAMQConnectionFactory.java:64)
    at com.rabbitmq.client.impl.recovery.AutorecoveringConnection.init(AutorecoveringConnection.java:156)
    at com.rabbitmq.client.ConnectionFactory.newConnection(ConnectionFactory.java:1130)
    at com.rabbitmq.client.ConnectionFactory.newConnection(ConnectionFactory.java:1087)
    at com.rabbitmq.client.ConnectionFactory.newConnection(ConnectionFactory.java:1045)
    at com.rabbitmq.client.ConnectionFactory.newConnection(ConnectionFactory.java:1207)
    at com.yll.demo8.Producer.main(Producer.java:35)
```

解决方法：减少连接数，关闭不使用的连接。

可能原因五：未创建 Vhost 或者填错 Vhost 名称

报错如下：

```
Couldn't log in: server connection error 530, message: NOT_ALLOWED - vhost /localdev/ not found
```

解决方法：

- 如果未创建Vhost，进入RabbitMQ控制台的“Vhost列表”页面，创建Vhost。
- 如果是Vhost名称填错了，请参考RabbitMQ控制台的“Vhost列表”页面显示的Vhost名称，修改连接URL和配置文件。

16.2.3 RabbitMQ 实例是否支持公网访问？

支持。在创建RabbitMQ实例时开启“公网访问”，或创建完后，在实例详情页中将公网访问开关打开。

16.2.4 RabbitMQ 是否支持跨 Region 部署？

当前支持跨AZ（可用区），不支持跨Region部署。

16.2.5 RabbitMQ 实例是否支持跨 VPC 访问？

RabbitMQ实例支持跨VPC访问，可以通过创建VPC对等连接，将两个VPC的网络打通，实现跨VPC访问实例。

关于创建和使用VPC对等连接，请参考[VPC对等连接说明](#)。

16.2.6 RabbitMQ 实例是否支持不同的子网？

支持。

客户端与实例在相同VPC内，可以跨子网段访问。同一个VPC内的子网默认可以进行通信。

16.2.7 SSL 方式连接 RabbitMQ 实例失败？

首先排查安全组的入方向规则，是否放开了端口5671（SSL方式访问）或5672（非SSL访问）。

其次，参考如下内容配置SSL单向认证：

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

16.2.8 客户端是否可以通过 DNAT 方式访问 RabbitMQ 实例？

不可以。客户端可以使用代理、VPN、专线、FullNAT或者反向代理等方式访问 RabbitMQ实例。

16.2.9 RabbitMQ 实例的 Web 管理页面无法打开

可能原因：实例安全组配置不正确

解决方案：重新配置安全组，具体步骤如下。

1. 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。
2. 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。

16.2.10 客户端是否可以连接同个 RabbitMQ 下多个 Vhost？

客户端可以连接同个RabbitMQ下多个Vhost。

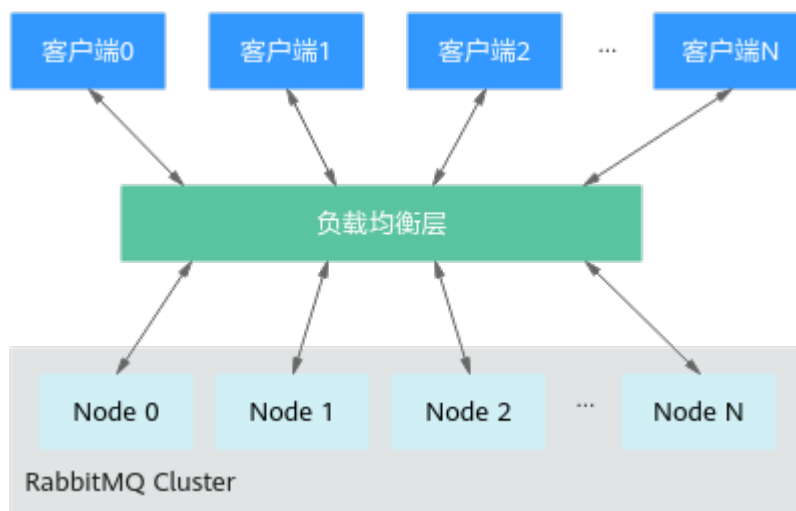
Vhost (Virtual Hosts) 是RabbitMQ的基本特性，每个Vhost相当于一个相对独立的RabbitMQ服务器，每个Vhost数据目录不同，共用一个进程。性能上，连接多个Vhost和单独使用一个Vhost差别不大，只是RabbitMQ进程多一些对象，建议使用业务模型实测。

Vhost的相关介绍，请参考官网文档[Virtual Hosts](#)。

16.2.11 为什么 RabbitMQ 集群只有一个连接地址？

RabbitMQ集群实例的连接地址，实际上是实例的LVS节点地址（负载均衡地址），客户端连接实例时，通过负载均衡器将客户端请求分发到集群实例的各个节点。

图 16-4 连接示意图



16.3 消息问题

16.3.1 RabbitMQ 实例支持延时消息队列么？

建议使用分布式消息服务RocketMQ版的定时/延迟消息来实现延时消息功能。

16.3.2 消息堆积对业务的影响及解决办法

消息堆积对业务的影响

过量的消息堆积可能会引起内存或磁盘告警，从而造成所有connection阻塞，进而影响到其他队列的使用，导致整体服务质量的下降。

消息堆积产生的原因

1. 一般来说消息堆积是由于生产消息的速率远大于消费消息的速率所导致的。比如某个时间段消费端处理消息异常缓慢，发送一条消息只要3秒钟，而消费一条消息需要1分钟，每分钟发送20个消息，只有一个消息被消费端处理，这样队列中就会产生大量的消息堆积。
2. 消费者出现异常，生产者一直在发送消息，但是消费者不能消费，造成消息积压。
3. 消费者没有出现异常，但是消费者与队列间的订阅可能出现了异常，也会导致消息无法被消费从而造成堆积的情况。
4. 消费者正常，与队列间的订阅也正常，但是消费端的代码本身逻辑耗费时间长导致了消费能力降低，这时候就会出现1中的情况从而导致消息堆积。

解决消息堆积的办法

1. **生产速率较快，消费速率较慢：**您可以通过以下方法解决。
 - 增加消费者数量提高消费速率。
 - 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。

2. **消费者异常**：建议排查消费者逻辑是不是有问题，优化程序。
3. **消费者与队列间的订阅异常**：建议排查队列和消费者之间的订阅是否正常。
4. **消费端的代码本身逻辑耗时间长**：建议给消息设置过期时间，设置方法如下：
 - 在生产消息时，设置消息过期时间。消息过期时间以expiration值体现。

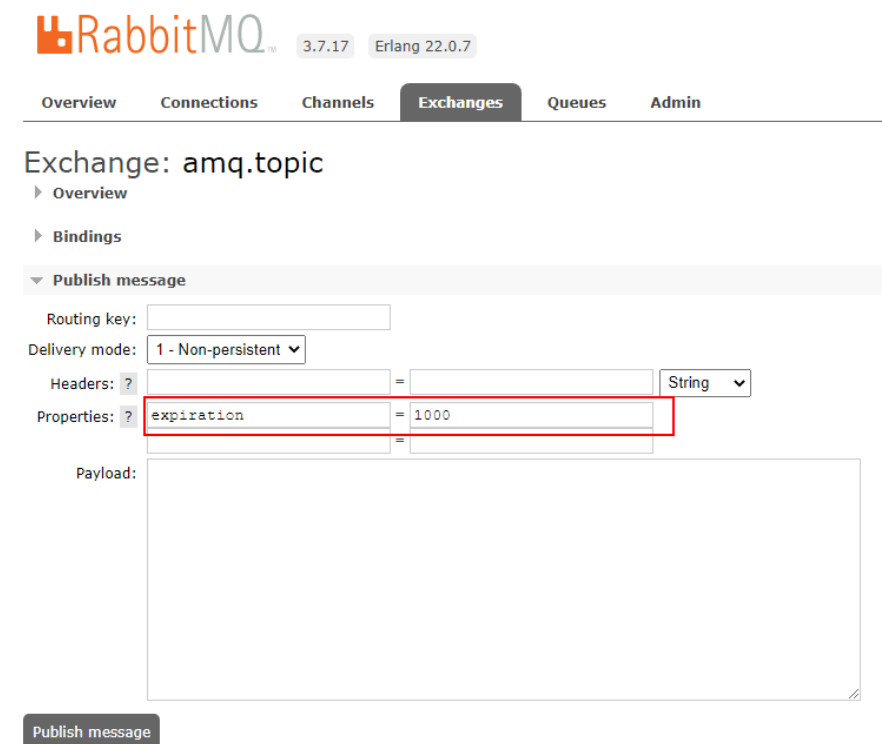
- 在properties中设置expiration值，单位为毫秒(ms)。

```
AMQP.BasicProperties properties = new AMQP.BasicProperties().builder()
    .deliveryMode(2)
    .contentEncoding("UTF-8")
    .expiration("10000")
    .build();
```

```
String message = "hello rabbitmq";
channel.basicPublish(exchange, routingKey, properties,
    message.getBytes(StandardCharsets.UTF_8));
```

- 在Web界面中设置expiration值，单位为毫秒(ms)。

[登录Web界面](#)，在“Exchanges”页签，单击Exchange名称，进入Exchange详情页。在“Publish message”区域，设置expiration值，如下图所示。



- 设置队列过期时间。队列过期时间以x-message-ttl值体现。从消息进入队列开始计算，超过了配置的队列过期时间，消息会自动被删除。

- 在客户端代码中设置x-message-ttl值，单位为毫秒(ms)。

```
Map<String, Object> arguments = new HashMap<String, Object>();
arguments.put("x-message-ttl", 10000);
channel.queueDeclare(queueName, true, false, false, arguments);
```

- 在Web界面新建队列时，设置x-message-ttl值，单位为毫秒(ms)。

[登录Web界面](#)，在“Exchanges”页签，新建队列时，设置x-message-ttl值，如下图所示。

The screenshot shows the RabbitMQ web interface for managing Exchanges. At the top, there are navigation tabs: Overview, Connections, Channels, Exchanges (selected), Queues, and Admin. Below the tabs, the page title is 'Exchanges' with a sub-link 'All exchanges (7)'. A table lists the following exchanges:

Name	Type	Features	Message rate in	Message rate out	+/-
(AMQP default)	direct	D			
amq.direct	direct	D			
amq.fanout	fanout	D			
amq.headers	headers	D			
amq.match	headers	D			
amq.rabbitmq.trace	topic	D I			
amq.topic	topic	D			

Below the table is a section 'Add a new exchange' with the following form fields:

- Name:
- Type:
- Durability:
- Auto delete:
- Internal:
- Arguments: (highlighted with a red box)
- Number:
- String:

At the bottom of the form, there is a button 'Add Alternate exchange ?'.

16.3.3 消息的最长保留时间是多久？

一般情况下消息如果未被消费会一直保留，只有被消费后，才会被删除。但是如果设置了过期时间（TTL），则以TTL时间为准。

16.3.4 消息创建时间在哪设置？

消息创建时间是由生产客户端在生产消息时设置的。

16.3.5 RabbitMQ 生产消息的最大长度是多少？

单条消息的最大长度为50MB，请勿发送大于此长度的消息，否则会导致生产失败。

16.4 监控告警问题

16.4.1 云监控无法展示 RabbitMQ 监控数据

监控数据无法展示，可能原因：队列名称开头包含特殊字符，例如点号“.”、下划线“_”，建议删除带特殊字符的队列。

16.4.2 云监控显示通道数一直上升报警有影响吗？

一个连接最大通道数是2047，超过后再创建通道数会失败，建议排查是否为资源没有释放导致的。

A 修订记录

发布日期	修订记录
2024-10-29	本次变更如下： <ul style="list-style-type: none">下线重启实例功能。新增查看Queue详情功能，主要涉及查看RabbitMQ Queue信息章节。
2024-01-24	本次变更如下： <ul style="list-style-type: none">下线3.7.17版本，主要涉及购买RabbitMQ实例和RabbitMQ使用的版本是多少? 章节。下线“升级版本”功能。
2023-03-09	本次变更如下： <ul style="list-style-type: none">新增磁盘加密功能，主要涉及购买RabbitMQ实例章节。新增快速入门。
2023-02-08	本次变更如下： <ul style="list-style-type: none">购买实例时，使用新版实例规格，主要涉及产品规格和购买RabbitMQ实例章节。新增升级版本和配置Vhost功能。
2021-11-26	本次变更如下： <ul style="list-style-type: none">新增计费说明章节。
2020-11-06	第一次正式发布。