

分布式消息服务 RabbitMQ

用户指南

文档版本 05
发布日期 2025-09-03



版权所有 © 华为云计算技术有限公司 2025。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	7
1.7 约束与限制	8
1.8 RabbitMQ 相关概念	11
1.9 Exchange 类型介绍	11
1.10 权限管理	14
2 快速入门	17
2.1 快速连接 RabbitMQ 并生产消费消息	17
3 RabbitMQ 业务使用流程	23
4 权限管理	25
4.1 创建用户并授权使用 DMS for RabbitMQ	25
5 购买 RabbitMQ 实例	29
6 配置 Vhost	39
6.1 创建 RabbitMQ Vhost	39
6.2 创建 RabbitMQ Exchange	41
6.3 绑定 RabbitMQ Exchange	43
6.4 创建 RabbitMQ Queue	45
6.5 绑定 RabbitMQ Queue	47
6.6 管理 RabbitMQ Vhost	48
6.6.1 查看 RabbitMQ Vhost 信息	48
6.6.2 删除 RabbitMQ Vhost	50
6.7 管理 RabbitMQ Exchange	52
6.7.1 解绑 RabbitMQ Exchange	52
6.7.2 删除 RabbitMQ Exchange	53
6.8 管理 RabbitMQ Queue	54
6.8.1 查看 RabbitMQ Queue 信息	54

6.8.2 清空 RabbitMQ Queue 消息.....	56
6.8.3 解绑 RabbitMQ Queue.....	58
6.8.4 配置 RabbitMQ 镜像队列.....	59
6.8.5 配置 RabbitMQ 惰性队列.....	61
6.8.6 配置 RabbitMQ 仲裁队列.....	64
6.8.7 配置 RabbitMQ 排他队列.....	70
6.8.8 配置 RabbitMQ 单一活跃消费者.....	70
6.8.9 删除 RabbitMQ Queue.....	72
7 连接实例.....	77
7.1 配置 RabbitMQ 网络连接.....	77
7.1.1 连接 RabbitMQ 网络要求.....	77
7.1.2 配置 RabbitMQ 实例的公网访问.....	78
7.2 配置 RabbitMQ 客户端心跳超时时间.....	80
7.3 使用客户端连接 RabbitMQ（关闭 SSL）.....	82
7.4 使用客户端连接 RabbitMQ（开启 SSL）.....	84
8 管理消息.....	88
8.1 配置 RabbitMQ 死信消息.....	88
8.2 配置 RabbitMQ 消息确认机制.....	90
8.3 配置 RabbitMQ 消息预取值.....	91
9 配置高级特性.....	93
9.1 配置 RabbitMQ 持久化.....	93
9.2 配置 RabbitMQ TTL.....	99
10 管理实例.....	101
10.1 查看和修改 RabbitMQ 实例基本信息.....	101
10.2 查看 RabbitMQ 客户端连接地址.....	103
10.3 查看 RabbitMQ 实例后台任务.....	104
10.4 配置 RabbitMQ 实例标签.....	105
10.5 配置 RabbitMQ 实例回收站策略.....	107
10.6 重置 RabbitMQ 实例密码.....	109
10.7 开启 RabbitMQ 实例插件.....	110
10.8 使用 rabbitmq_tracing 插件.....	112
10.9 导出 RabbitMQ 实例列表.....	116
10.10 删除 RabbitMQ 实例.....	117
10.11 登录 RabbitMQ WebUI.....	118
11 变更实例规格.....	120
11.1 变更 RabbitMQ 实例规格.....	120
12 迁移 RabbitMQ 业务.....	124
13 申请扩大 RabbitMQ 配额.....	129
14 查看监控指标与配置告警.....	130

14.1 查看 RabbitMQ 监控数据.....	130
14.2 RabbitMQ 支持的监控指标.....	131
14.3 配置 RabbitMQ 监控告警.....	138
15 查看 RabbitMQ 审计日志.....	142
16 常见问题.....	144
16.1 实例问题.....	144
16.1.1 RabbitMQ 使用的版本是多少?	144
16.1.2 RabbitMQ 实例 SSL 连接的协议版本号是多少?	144
16.1.3 创建实例时为什么无法查看子网和安全组等信息?	144
16.1.4 RabbitMQ 集群实例如何均衡分发请求到每个虚拟机?	144
16.1.5 RabbitMQ 实例集群内部的队列是否有冗余备份?	144
16.1.6 RabbitMQ 实例是否支持持久化, 如何定时备份数据?	144
16.1.7 RabbitMQ 实例开启 SSL 后, 证书怎么获取?	144
16.1.8 RabbitMQ 实例的 SSL 开关是否支持修改?	145
16.1.9 RabbitMQ 实例是否支持扩容?	145
16.1.10 RabbitMQ 支持双向认证吗?	145
16.1.11 RabbitMQ 支持升级 CPU 和内存吗?	145
16.1.12 如何关闭 RabbitMQ 的 WebUI?	145
16.1.13 实例是否支持修改可用区?	145
16.1.14 如何获取 region id?	146
16.1.15 为什么不能选择 2 个可用区?	146
16.1.16 单机 RabbitMQ 实例如何变更为集群实例?	146
16.1.17 RabbitMQ 实例创建后, 能修改 VPC 和子网吗?	146
16.1.18 RabbitMQ 扩容/缩容会影响业务吗?	147
16.2 连接问题.....	148
16.2.1 如何配置安全组?	148
16.2.2 RabbitMQ 客户端连接报错原因分析.....	149
16.2.3 RabbitMQ 实例是否支持公网访问?	150
16.2.4 RabbitMQ 是否支持跨 Region 部署?	150
16.2.5 RabbitMQ 实例是否支持跨 VPC 访问?	150
16.2.6 RabbitMQ 实例是否支持不同的子网?	151
16.2.7 SSL 方式连接 RabbitMQ 实例失败?	151
16.2.8 客户端是否可以通过 DNAT 方式访问 RabbitMQ 实例?	151
16.2.9 RabbitMQ 实例的 Web 管理页面无法打开.....	151
16.2.10 客户端是否可以连接同个 RabbitMQ 下多个 Vhost?	151
16.2.11 为什么 RabbitMQ 集群只有一个连接地址?	151
16.2.12 RabbitMQ 实例支持 ping 吗?	152
16.3 消息问题.....	152
16.3.1 RabbitMQ 实例支持延时消息队列么?	152
16.3.2 消息堆积对业务的影响及解决办法.....	152
16.3.3 消息的最长保留时间是多久?	154
16.3.4 消息创建时间在哪设置?	154

16.3.5 RabbitMQ 生产消息的最大长度是多少? 154

16.4 监控告警问题..... 154

16.4.1 云监控无法展示 RabbitMQ 监控数据..... 155

16.4.2 云监控显示通道数一直上升报警有影响吗? 155

A 修订记录..... 156

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，消息传输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。

对于同步链路，需要实时返回调用结果的场景，建议使用RPC（Remote Procedure Call）调用方案。

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

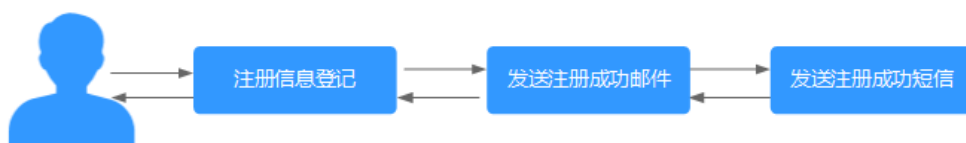
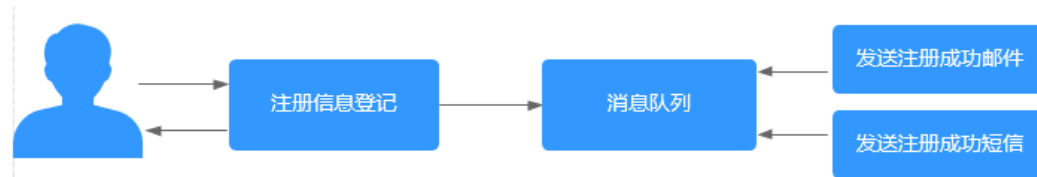


图 1-2 借助消息队列异步发送注册邮件与短信流程

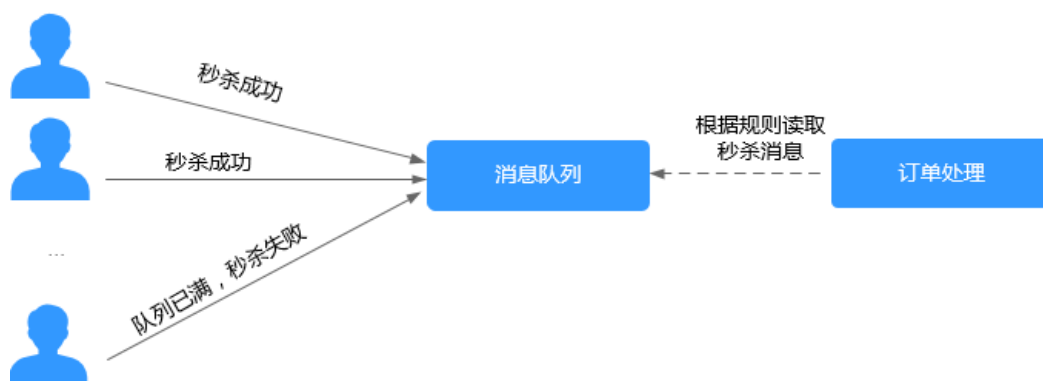


错峰流控与流量削峰

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。

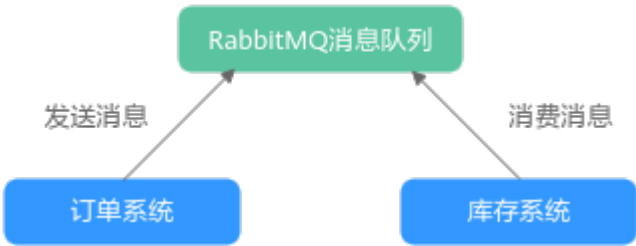
图 1-4 系统高耦合



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.12u24g.cluster	3	300~90000	24000	24000	600	6000
	5	500~150000	40000	24000	600	6000
	7	700~210000	56000	24000	600	6000
rabbitmq.16u32g.cluster	3	300~90000	48000	32000	800	8000
	5	500~150000	80000	32000	800	8000
	7	700~210000	112000	32000	800	8000
rabbitmq.24u48g.cluster	3	300~90000	60000	40000	1000	10000
	5	500~150000	100000	40000	1000	10000
	7	700~210000	140000	40000	1000	10000
rabbitmq.32u64g.cluster	3	300~90000	72000	40000	1000	10000
	5	500~150000	120000	40000	1000	10000
	7	700~210000	168000	40000	1000	10000

📖 说明

- 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。
- RabbitMQ 3.x.x版本实例单个代理支持的最大连接数如上表所示，整个集群支持的最大连接数不超过30000。

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量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 1-2 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
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。

功能项	RocketMQ	Kafka	RabbitMQ
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。
- 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，您在使用RabbitMQ实例时注意不要超过相应的约束和限制，以免程序出现异常。

须知

如果因为您的实例配置超过相应的约束和限制而导致的任何不稳定情况，不在SLA承诺和赔偿范围内。

实例

表 1-3 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。
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	<= 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	>= 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。

限制项	约束和限制
存储空间	- 只支持扩大存储空间，不支持减小存储空间。 - 扩容存储空间有次数限制，只能扩容20次。
代理数量	- 集群实例只支持增加代理数量，不支持减少代理数量。 - 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。
是否支持ping命令	- 单机实例：内网连接地址和公网连接地址都支持ping命令。 - 集群实例：内网连接地址支持ping命令，公网连接地址不支持ping命令。

Vhost

表 1-4 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。
Vhost名称	- RabbitMQ 3.x.x版本实例的Vhost名称以特殊字符开头，例如点号“.”，可能会导致监控数据无法显示。Vhost名称包含特殊字符，例如“%”、“ ”、“/”，会导致Vhost名称与监控页面不一致，在监控页面中特殊字符会以“_”显示。例如：Vhost名称为Vhost.1%1 2_3/，此时监控中显示的Vhost名为“Vhost.1_1_2_3_”。 - 在云监控中特殊字符“.”按5个字符计算长度。如果RabbitMQ 3.x.x版本实例的Vhost名称包含“.”，且经此规则计算后的总长度超过256个字符，可能会导致监控数据无法显示。

Exchange

表 1-5 Exchange 约束与限制

限制项	约束和限制
默认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。

限制项	约束和限制
绑定Exchange	- 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。
删除Exchange	RabbitMQ 3.x.x版本中，默认Exchange不支持删除。

Queue

表 1-6 Queue 约束与限制

限制项	约束和限制
绑定Queue	- 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。
惰性队列	RabbitMQ 3.8.35及以上版本支持惰性队列。
仲裁队列	RabbitMQ 3.8.35及以上版本支持仲裁队列。
单一活跃消费者	RabbitMQ 3.8.35及以上版本支持单一活跃消费者特性。
Queue名称	- RabbitMQ 3.x.x版本实例的Queue名称以特殊字符开头，例如点号“.”，可能会导致监控数据无法显示。Queue名称包含特殊字符，例如“%”、“ ”、“/”，会导致Queue名称与监控页面不一致，在监控页面中特殊字符会以“_”显示。例如：Queue名称为Queue.1%1 2_3/，此时监控中显示的Queue名为“Queue.1_1_2_3_”。 - 在云监控中特殊字符“.”按5个字符计算长度。如果RabbitMQ 3.x.x版本实例的Queue包含“.”，且经此规则计算后的总长度超过256个字符，可能会导致监控数据无法显示。

消息

表 1-7 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：

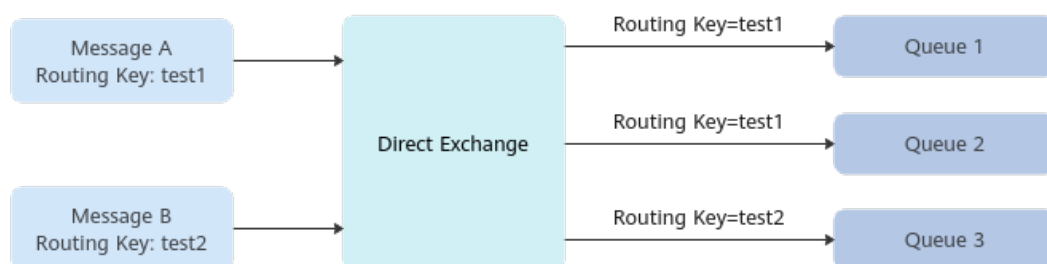
Direct Exchange会将消息路由到Routing Key完全匹配的Queue中。

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：

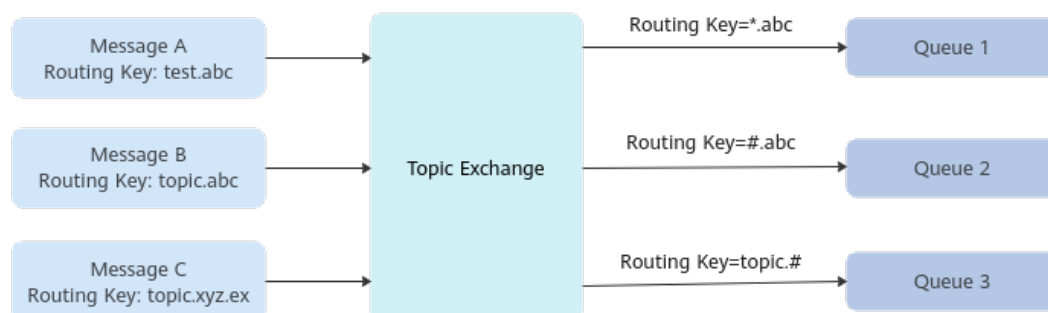
Topic Exchange会将Routing Key进行通配符匹配，然后将消息路由到匹配成功的Queue中。

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：

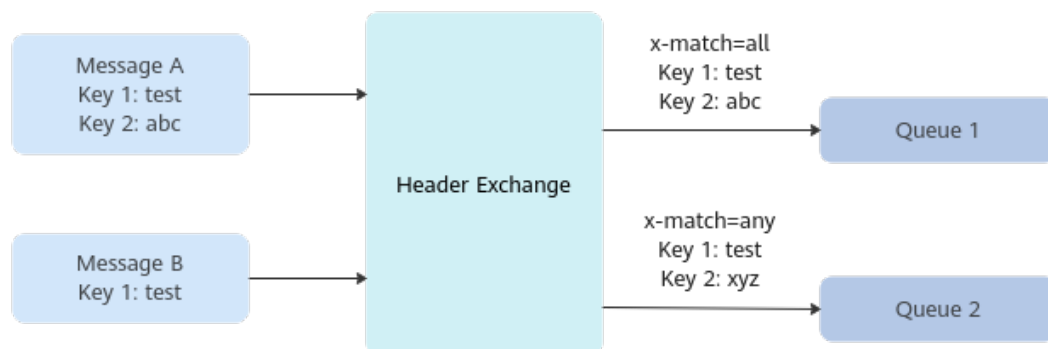
- x-match=all，表示所有的消息Headers属性键值对和绑定属性键值对都匹配才会路由消息。
- x-match=any，表示只要有一组消息Headers属性键值对和绑定属性键值对匹配，就会路由消息。

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1-8所示，包括了DMS for RabbitMQ的所有系统权限。

表 1-8 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
DMS ELBAccess	分布式消息服务租户委托时需要授权的ELB操作权限。	系统策略	无
DMS VPCEndpointAccess	分布式消息服务租户委托时需要授权的VPCEndpoint操作权限。	系统策略	无
DMSAgencyCheckAccessPolicy	分布式消息服务检查租户委托权限需要授权的IAM操作权限。	系统策略	无
DMS BSSAccess	分布式消息服务租户委托时需要授权的BSS操作权限。	系统策略	无
DMS OBSAccess	分布式消息服务租户委托时需要授权的OBS操作权限。	系统策略	无

表1-9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 1-9 常用操作与系统策略的关系

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

除**表1-9**以外的系统策略，没有任何DMS for RabbitMQ常用操作的权限。

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](#)

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](#)。

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](#)。

创建的安全组与RabbitMQ实例必须在相同的区域。

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 在“快速购买”页签，设置实例基础配置，配置详情请参考表2-2。

表 2-2 设置实例基础配置

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“马来西亚-吉隆坡”。
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”。

- 步骤3 设置实例规格和存储空间，配置详情请参考表2-3。

表 2-3 设置实例规格和存储空间

参数	说明
套餐规格	选择“测试规格”，该规格提供最小规格的单机实例，为您节约资源和使用成本，但由于仅为单机部署，无法支持高可用能力，适用测试或者调试业务。
类型	根据实际需要选择存储RabbitMQ数据的磁盘类型。 RabbitMQ实例创建后，磁盘类型不支持修改。 选择“超高I/O”。

参数	说明
容量	根据实际需要选择存储RabbitMQ数据的磁盘大小。 实例总存储空间 = 单个代理的存储空间 * 代理数量 选择“100GB”。

步骤4 设置实例网络环境信息，配置详情请参考[表2-4](#)。

表 2-4 设置实例网络环境信息

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

步骤5 设置实例的访问方式，保持默认即可。

步骤6 设置实例的认证方式，配置详情请参考[表2-5](#)。

表 2-5 设置实例的认证方式

参数	说明
RabbitMQ认证方式用户名	设置连接RabbitMQ实例的用户名。 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。 输入“test”。
密码	设置连接RabbitMQ实例的密码。 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7 设置实例高级配置，配置详情请参考[表2-6](#)，其他参数保持默认设置。

表 2-6 设置实例高级配置

参数	说明
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”。
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”。

步骤8 单击“确认订单”，进入规格确认页面。

步骤9 确认实例信息无误后，提交请求。

步骤10 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。

步骤11 实例创建成功后，单击实例名称，进入实例详情页。

步骤12 在“连接信息”区域，查看并记录连接地址。

----结束

步骤三：连接实例生产消费消息

步骤1 进入ECS的“root”目录下，下载[RabbitMQ-Tutorial.zip](#)示例工程代码。

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

说明

“/root”为示例工程代码存放路径，请根据实际情况修改。

步骤2 解压RabbitMQ-Tutorial.zip压缩包。

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

步骤3 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4 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[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。

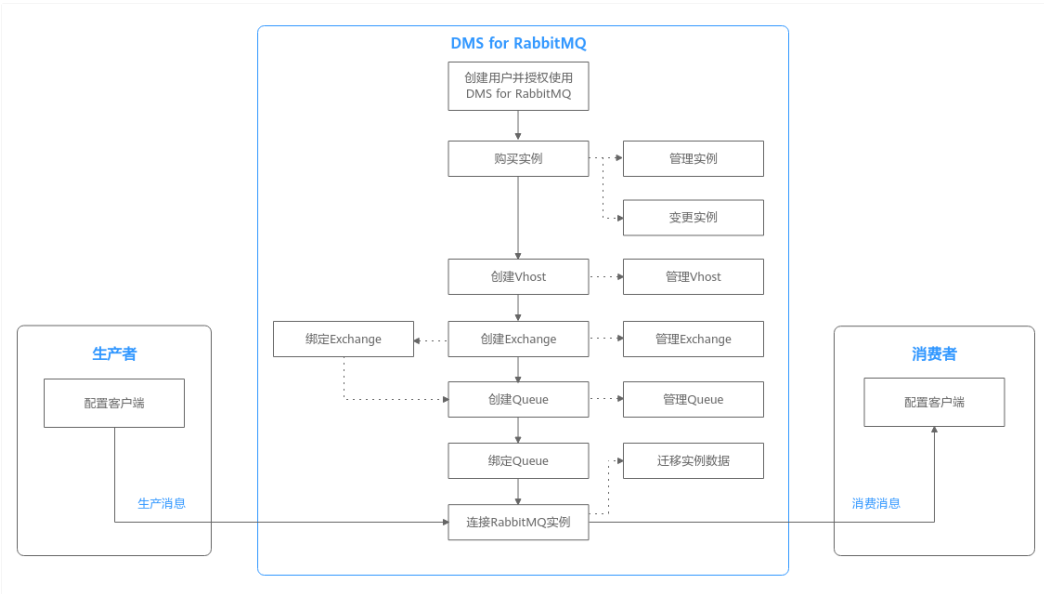
----结束

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



- 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。
- 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。
- 创建Vhost**
生产者和消费者连接RabbitMQ实例时，需要指定一个Vhost。
- 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。

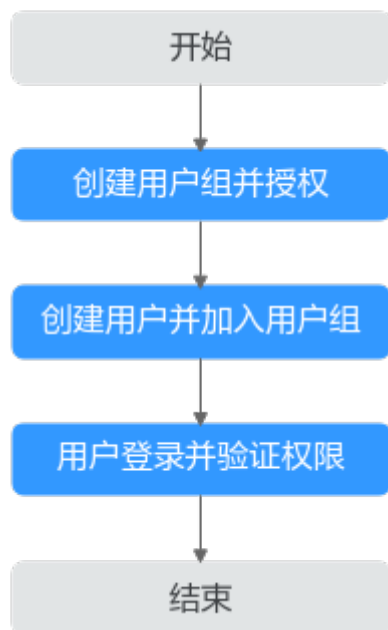
前提条件

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](#)。

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

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

新创建的用户登录到控制台，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。

- 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	【 格式 】 DMS:*:*rabbitmq:实例ID 【 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。

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。 安全组与RabbitMQ实例必须在相同的区域。 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	如果客户端使用公网连接 RabbitMQ实例，请提前创建弹性IP地址。 在创建弹性IP地址时，应注意如下要求： • 创建的弹性IP地址与RabbitMQ实例在相同的区域。 • RabbitMQ控制台无法识别开启IPv6转换功能的弹性IP地址。	创建弹性IP地址的操作指导请参考《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”。

快速购买 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RabbitMQ实例列表页面。
- 步骤4 单击页面右上方的“购买RabbitMQ实例”。
- 步骤5 在“快速购买”页签中，设置实例基础配置。

表 5-2 实例基础配置参数说明

参数名称	说明
计费模式	• 按需付费：一种后付费模式，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。
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，不支持修改可用区。

- 步骤6 设置套餐规格。
- 分布式消息服务RabbitMQ版已经预设了几种规格，请根据需要选择。磁盘类型和磁盘大小根据实际需求进行设置。**RabbitMQ实例创建后，磁盘类型不支持修改。**

图 5-1 套餐规格



步骤7 设置网络信息。

表 5-3 实例网络参数说明

参数名称	说明
虚拟私有云	选择已经创建好的或共享的虚拟私有云。 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。您可以单击右侧的“管理虚拟私有云”，跳转到网络控制台的“虚拟私有云”页面，查看或创建虚拟私有云。 虚拟私有云在RabbitMQ实例创建完成后，不支持修改。
子网	选择已经创建好的子网。 子网在RabbitMQ实例创建完成后，不支持修改。
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“创建安全组”，跳转到网络控制台的“安全组”页面，查看或创建安全组。 客户端连接RabbitMQ实例前，请根据不同的连接方式配置对应的安全组规则，否则会连接失败。具体的安全组规则请参考 表7-2 。

步骤8 设置实例访问方式。

表 5-4 实例访问方式参数说明

参数名称	说明
公网访问	是否开启公网访问。开启公网访问后，客户端可以通过弹性IP地址访问RabbitMQ实例。 开启公网访问时，需要设置弹性IP地址。如果弹性IP地址数量不足，请单击“创建弹性IP”跳转到弹性公网IP页面，创建公网IP地址。然后返回RabbitMQ控制台，在“公网IP地址”后单击，刷新弹性IP地址。 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或删除EIP，相应RabbitMQ实例的公网访问功能会自动关闭。
加密方式	是否开启SSL加密传输。开启SSL后，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。

步骤9 设置实例认证方式。

表 5-5 实例认证方式参数说明

参数名称	说明
RabbitMQ认证方式用户名	设置连接RabbitMQ实例的用户名。 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	设置连接RabbitMQ实例的密码。 密码需要符合以下命名规则： - 长度为8~32个字符。 - 至少包含以下字符中的3种：大写字母、小写字母、数字、特殊字符`~!@#\$\$%^&*()-_+=\ [{ } ; : " , < . > ?`和空格，并且不能以-开头。 - 不能与用户名或倒序的用户名相同。

步骤10 设置高级配置。

表 5-6 高级配置参数说明

参数名称	说明
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。

参数名称	说明
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。
标签	标签用于标识云资源，当您拥有相同类型的许多云资源时，可以使用标签按各种维度（例如用途、所有者或环境）对云资源进行分类。 如您的组织已经设定分布式消息服务RabbitMQ版的相关标签策略，则需按照标签策略规则为RabbitMQ实例添加标签。标签如果不符合标签策略的规则，则可能会导致RabbitMQ实例创建失败，请联系组织管理员了解标签策略详情。 - 如果您已经预定义了标签，在“标签键”和“标签值”中选择已经定义的标签键值对。另外，您可以单击“创建预定义标签”，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。
描述	设置实例的描述信息，长度为0~1024个字符。

步骤11 单击“确认订单”，进入“确认配置”页面。

步骤12 确认实例信息无误后，单击“提交”。

步骤13 在实例列表页面查看实例是否创建成功。

创建实例大约需要3到15分钟，此时实例的“状态”为“创建中”。

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

----结束

自定义购买 RabbitMQ 实例（3.x.x 版本）

步骤1 登录管理控制台。

步骤2 在管理控制台左上角单击 ，选择区域。

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RabbitMQ实例列表页面。

步骤4 单击页面右上方的“购买RabbitMQ实例”。

步骤5 在“自定义购买”页签中，设置实例基础配置。

表 5-7 实例基础配置参数说明

参数名称	说明
计费模式	• 按需付费：一种后付费模式，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。
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，不支持修改可用区。

步骤6 设置实例规格信息。

表 5-8 实例规格参数说明

参数名称	说明
版本	RabbitMQ的版本号，选择“3.8.35”。 实例创建后，版本号不支持修改。
部署架构	根据需求选择“单机”或“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。
代理规格	根据业务需求选择相应的代理规格。更多关于规格的介绍，请参见 产品规格 。 为了保证服务的稳定可靠，RabbitMQ采用了默认的40%高水位配置。当内存占用率到达40%后，可能会触发内存高水位，内存高水位会触发流控，导致生产者发送消息被阻塞。为了避免高水位的产生，请及时消费积压在队列中的消息。
代理数量	选择代理数量。
类型	选择存储RabbitMQ数据的磁盘类型。 RabbitMQ实例创建后，磁盘类型不支持修改。
容量	选择存储RabbitMQ数据的磁盘大小。

图 5-2 3.8.35 版本实例规格

Version

3.8.35

Software version of the instance. This setting is fixed once the instance is created. Ideally, use server and client with the same version.

Instance Type

Default

Architecture

Single-node

Cluster

Single-node instances cannot be changed to cluster ones upon creation.

Broker Flavor

Flavor Name	Maximum Connections per Broker	Recommended Queues per Broker
☒ rabbitmq.2u4g.single	1,000	100
☐ rabbitmq.4u8g.single	2,000	200
☐ rabbitmq.8u16g.single	4,000	400
☐ rabbitmq.16u32g.single	8,000	800
☐ rabbitmq.24u48g.single	10,000	1,000

Specifications Flavor Name: rabbitmq.2u4g.single | Maximum Connections per Broker: 1,000 | Recommended Queues per Broker: 100

Brokers

-

1

+

Storage Space per Broker

The system disk type and size cannot be changed. The storage space cannot be decreased. Select a disk type for your required service I/O. [Learn more](#)

Type

Ultra-high I/O

Capacity

-

100

+

Total storage space = 100 GB (1 brokers × 100 GB per broker)

步骤7 设置网络信息。

表 5-9 实例网络参数说明

参数名称	说明
虚拟私有云	选择已经创建好的或共享的虚拟私有云。 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。您可以单击右侧的“管理虚拟私有云”，跳转到网络控制台的“虚拟私有云”页面，查看或创建虚拟私有云。 虚拟私有云在RabbitMQ实例创建完成后，不支持修改。
子网	选择已经创建好的子网。 子网在RabbitMQ实例创建完成后，不支持修改。
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“创建安全组”，跳转到网络控制台的“安全组”页面，查看或创建安全组。 客户端连接RabbitMQ实例前，请根据不同的连接方式配置对应的安全组规则，否则会连接失败。具体的安全组规则请参考表7-2。

步骤8 设置实例访问方式。

表 5-10 实例访问方式参数说明

参数名称	说明
公网访问	是否开启公网访问。开启公网访问后，客户端可以通过弹性IP地址访问RabbitMQ实例。 开启公网访问时，需要设置弹性IP地址。如果弹性IP地址数量不足，请单击“创建弹性IP”跳转到弹性公网IP页面，创建公网IP地址。然后返回RabbitMQ控制台，在“公网IP地址”后单击，刷新弹性IP地址。 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或删除EIP，相应RabbitMQ实例的公网访问功能会自动关闭。
加密方式	是否开启SSL加密传输。开启SSL后，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。

步骤9 设置实例认证方式。

表 5-11 实例认证方式参数说明

参数名称	说明
RabbitMQ认证方式用户名	设置连接RabbitMQ实例的用户名。 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	设置连接RabbitMQ实例的密码。 密码需要符合以下命名规则： - 长度为8~32个字符。 - 至少包含以下字符中的3种：大写字母、小写字母、数字、特殊字符`~!@#\$\$%^&*()-_+=\ [{ }];:","<.>?`和空格，并且不能以-开头。 - 不能与用户名或倒序的用户名相同。

步骤10 设置高级配置。

表 5-12 高级配置参数说明

参数名称	说明
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。

参数名称	说明
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。
标签	标签用于标识云资源，当您拥有相同类型的许多云资源时，可以使用标签按各种维度（例如用途、所有者或环境）对云资源进行分类。 如您的组织已经设定分布式消息服务RabbitMQ版的相关标签策略，则需按照标签策略规则为RabbitMQ实例添加标签。标签如果不符合标签策略的规则，则可能会导致RabbitMQ实例创建失败，请联系组织管理员了解标签策略详情。 - 如果您已经预定义了标签，在“标签键”和“标签值”中选择已经定义的标签键值对。另外，您可以单击“创建预定义标签”，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。
描述	设置实例的描述信息，长度为0~1024个字符。

步骤11 在页面右侧“配置概要”中，查看已选择的实例配置。

步骤12 确认配置无误后，单击“确认订单”，进入“确认配置”页面。

步骤13 确认实例信息无误后，单击“提交”。

步骤14 在实例列表页面查看实例是否创建成功。

创建实例大约需要3到15分钟，此时实例的“状态”为“创建中”。

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

----结束

购买相同配置的 RabbitMQ 实例

若您需购买与当前RabbitMQ实例配置一致的新实例，可通过“购买相同配置”功能，一键复用当前实例配置，快速完成新实例购买，大幅节省配置时间。

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RabbitMQ实例列表页面。

步骤4 选择目标RabbitMQ实例，单击“操作”列下的“更多 > 购买相同配置”，进入“购买RabbitMQ实例”页面。

步骤5 系统已自动复制目标RabbitMQ实例的参数配置，根据需要调整待购买RabbitMQ实例的参数配置。参数配置的详细内容，请参见[自定义购买RabbitMQ实例（3.x.x版本）](#)。

为安全起见，当复制的RabbitMQ实例存在如下场景时，系统不会自动复制相应的参数配置，需要您手动配置：

- 客户端连接RabbitMQ实例的用户名和密码不会被复制，需要重新设置用户名和密码。
- 开启公网访问的RabbitMQ实例，需要重新设置公网IP地址。
- 目标RabbitMQ实例的名称不会被复制，需要重新设置RabbitMQ实例名称。

步骤6 在页面右侧“配置概要”中，查看已选择的实例配置。

步骤7 确认配置无误后，单击“确认订单”，进入“确认配置”页面。

步骤8 确认实例信息无误后，单击“提交”。

步骤9 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。

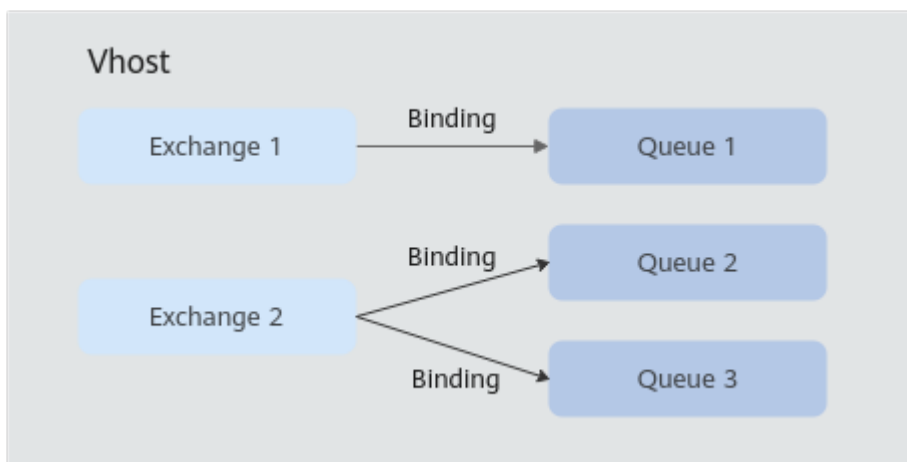
----结束

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。生产者与消费者连接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）](#)

约束与限制

- 实例创建后，会自动创建一个名为“/”的Vhost。
- Vhost名称以特殊字符开头，例如点号“.”，可能会导致监控数据无法显示。Vhost名称包含特殊字符，例如“%”、“|”、“/”，会导致Vhost名称与监控页

面不一致，在监控页面中特殊字符会以 “_” 显示。例如：Vhost名称为 Vhost.1%1|2_3/，此时监控中显示的Vhost名为 “Vhost.1_1_2_3_”。

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RabbitMQ实例列表页面。
- 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“实例管理 > 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名称须满足如下要求：只能由英文字母、数字、点、百分号、竖线、中划线、下划线、斜杆组成，长度2~128字符。

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。

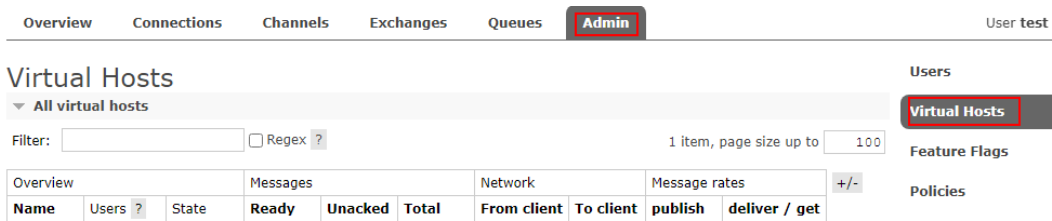
“Tracing” 表示是否开启消息追踪功能。开启消息追踪后，您可以跟踪消息的转发路径。

----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)

Virtual Hosts

All virtual hosts

Filter: ☐ Regex ?

Overview			Messages		
Name	Users ?	State	Ready	Unacked	Total
/	guest, test	■ running	NaN	NaN	NaN

Add a new virtual host

Name:

Description:

Tags:

Add virtual host

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)

Virtual Hosts

All virtual hosts

Filter: ☐ Regex ?

Overview			Messages			Network		Message rates	
Name	Users ?	State	Ready	Unacked	Total	From client	To client	publish	deliver / get
/	guest, test	■ running	NaN	NaN	NaN				
test-vhost	test	■ running	NaN	NaN	NaN				

----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。

约束与限制

- 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RabbitMQ实例列表页面。
- 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“实例管理 > 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	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、点、百分号、竖线、中划线、下划线、斜杠组成。 创建Exchange后不能修改名称。
类型	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 。 • 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	是否开启自动删除Exchange。 • 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。 - 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。 - 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RabbitMQ实例列表页面。

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“实例管理 > 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	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。 - 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，消息无法路由到目标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RabbitMQ实例列表页面。
- 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“实例管理 > 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。 - 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，消息无法路由到目标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。

约束与限制

- RabbitMQ 3.x.x版本实例的Queue名称以特殊字符开头，例如点号“.”，可能会导致监控数据无法显示。Queue名称包含特殊字符，例如“%”、“|”、“/”，会导致Queue名称与监控页面不一致，在监控页面中特殊字符会以“_”显示。例如：Queue名称为Queue.1%|2_3/，此时监控中显示的Queue名为“Queue.1_1_2_3_”。
- 在云监控中特殊字符“.”按5个字符计算长度。如果RabbitMQ 3.x.x版本实例的Queue包含“.”，且经此规则计算后的总长度超过256个字符，可能会导致监控数据无法显示。

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RabbitMQ实例列表页面。
- 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“实例管理 > 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、点、百分号、竖线、中划线、下划线、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。 - 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。 - 开启：当订阅该Queue消息的最后一个消费者取消订阅后，该Queue将被自动删除。 - 不开启：当订阅该Queue消息的最后一个消费者取消订阅后，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	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RabbitMQ实例列表页面。

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“实例管理 > 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	设置Key值字符串，用于告知Exchange应该将消息投递到哪些Queue中。 • 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，消息无法路由到目标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RabbitMQ实例列表页面。
- 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“实例管理 > 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。Exchange的详细信息请参见表6-6，Queue的详细信息请参见表6-7。

表 6-6 Exchange 详细信息说明

参数名称	说明
名称	Exchange的名称。

参数名称	说明
默认Exchange	该Exchange是否属于系统默认创建的。 - 是：此类Exchange不支持删除。 - 否：此类Exchange由您自行创建，支持删除。
绑定类型	Exchange的类型。 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。
持久化	该Exchange是否支持持久化。 - 是：该Exchange在服务器重启后仍然存在。 - 否：该Exchange在服务器重启后会被删除，需要重新创建。
Internal	该Exchange是否属于RabbitMQ内部使用。 - 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。
自动删除	该Exchange是否支持自动删除。 - 是：当最后一个绑定的Queue与Exchange解除绑定后，该Exchange会被自动删除。 - 否：当最后一个绑定的Queue与Exchange解除绑定后，不会删除该Exchange。

表 6-7 Queue 详细信息说明

参数名称	说明
名称	Queue的名称。
堆积消息	该Queue中堆积的消息数。
持久化	该Queue是否支持持久化。 - 是：该Queue在服务器重启后仍然存在。 - 否：该Queue在服务器重启后会被删除，需要重新创建。

参数名称	说明
自动删除	该Queue是否支持自动删除。 - 是：当订阅该Queue消息的最后一个消费者取消订阅后，该Queue将被自动删除。 - 否：当订阅该Queue消息的最后一个消费者取消订阅后，不会删除该Queue。
策略	该Queue已设置的策略的名称。

----结束

6.6.2 删除 RabbitMQ Vhost

本章节指导如何删除Vhost。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）

约束与限制

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

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RabbitMQ实例列表页面。
- 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“实例管理 > Vhost列表”，进入Vhost列表页面。
- 步骤6** 通过以下任意一种方法，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”。



警告

删除Vhost后，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”。

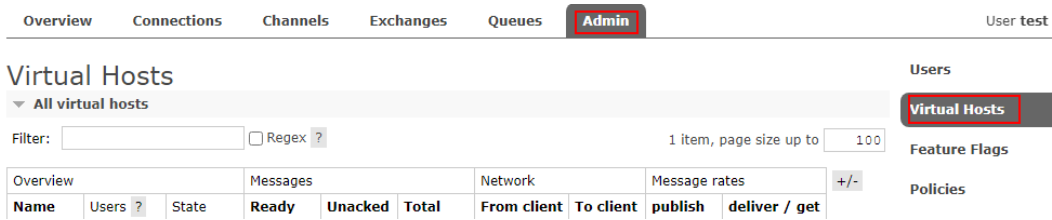
删除的Vhost已从Vhost列表中移除，表明Vhost已成功删除。

----结束

删除 Vhost（RabbitMQ WebUI）

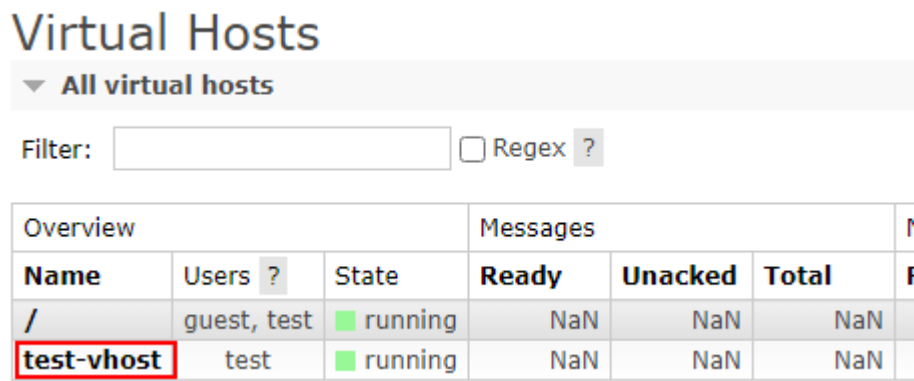
- 步骤1** 登录RabbitMQ WebUI。
- 步骤2** 在顶部导航栏选择“Admin”，进入Admin页面。
- 步骤3** 在右侧导航栏选择“Virtual Hosts”，进入Virtual Hosts页面。

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

Virtual Host: test-vhost

► Overview

► Permissions

► Topic permissions

▼ Delete this vhost

Delete this virtual host



警告

删除Vhost后，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6 单击“确定”，完成Vhost的删除。

在“Virtual Hosts”页面的“All virtual hosts”区域，已移除删除的Vhost，表明Vhost已成功删除。

----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Exchange。

前提条件

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

约束与限制

解绑Exchange后，对应的路由将不再提供服务，且无法恢复，请谨慎操作。

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RabbitMQ实例列表页面。
- 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“实例管理 > 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。

 警告

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RabbitMQ实例列表页面。
- 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“实例管理 > 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。

 警告

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。

约束与限制

- 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RabbitMQ实例列表页面。

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“实例管理 > Vhost列表”，进入Vhost列表页面。

步骤6 单击Vhost名称，进入Vhost详情页。

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

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

警告

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。

步骤8 单击“确定”，完成Exchange的删除。

删除的Exchange已从Exchange列表中移除，表明Exchange已成功删除。

----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RabbitMQ实例列表页面。
- 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“实例管理 > 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。

表 6-8 Queue 基本信息参数说明

参数名称	说明
名称	Queue的名称。
所属Vhost	该Queue所属的Vhost名称。
堆积消息	该Queue中堆积的消息数。
消费者数量	消费该Queue消息的消费者数量。
自动删除	该Queue是否支持自动删除。 - 是：当订阅该Queue消息的最后一个消费者取消订阅后，该Queue将被自动删除。 - 否：当订阅该Queue消息的最后一个消费者取消订阅后，不会删除该Queue。
死信交换器	该Queue如果绑定了死信交换器，显示死信Exchange的名称，否则显示“--”。
死信路由键	如果该Queue已绑定死信Exchange且设置了死信RoutingKey，则显示死信的RoutingKey，否则显示“--”。
消息过期时间（ms）	如果该Queue已设置消息过期时间，则显示此时间，否则显示“--”。 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	该Queue如果为惰性队列，显示“lazy”，否则显示“--”。 更多关于惰性队列的说明，请参见配置RabbitMQ惰性队列。

表 6-9 绑定信息参数说明

参数名称	说明
绑定类型	Exchange的绑定类型，仅会显示“queue”，表示Exchange绑定了Queue。
绑定源名称	与该Queue绑定的Exchange的名称。
Routing Key	消息路由到该Queue的Routing Key。

表 6-10 消费者信息参数说明

参数名称	说明
Consumer Tag	Consumer客户端的唯一标识符。
Channel	Consumer客户端和RabbitMQ实例连接的Channel。
Ack Required	消息是否自动确认。 • true：消息一旦发送给消费者，RabbitMQ就会立即将其标记为已确认，并从队列中删除。 • false：消息发送给消费者后，RabbitMQ会将消息标记为未确认状态，并保留在队列中，直到消费者发送确认信号（ack）给RabbitMQ，RabbitMQ才会删除消息。
Prefetch Count	消息的预取值。更多关于预取值的说明，请参见 配置RabbitMQ消息预取值 。
User	Consumer客户端连接RabbitMQ实例的用户名。

----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）](#)

约束与限制

- 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RabbitMQ实例列表页面。
- 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“实例管理 > 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。

 **注意**

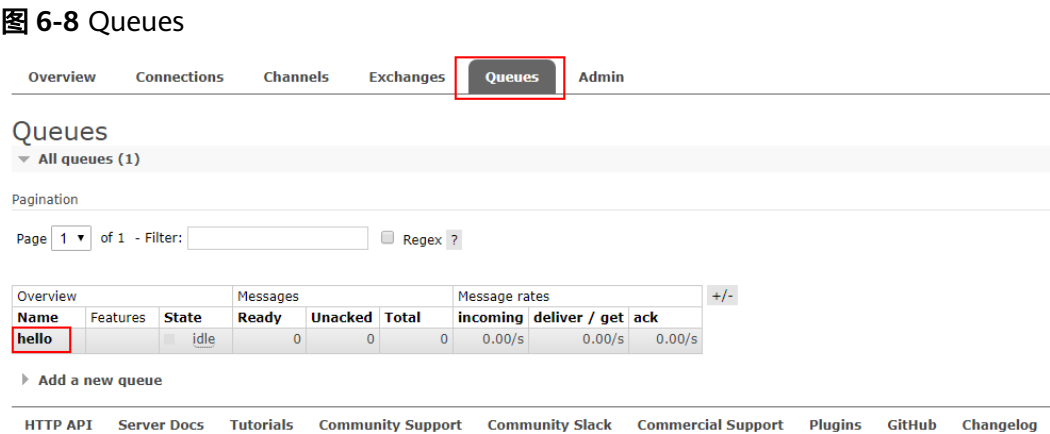
清空Queue消息后，Queue中的所有消息会被永久删除，且不可恢复，请谨慎操作。

步骤8 单击“确定”，完成Queue的清空消息。

----结束

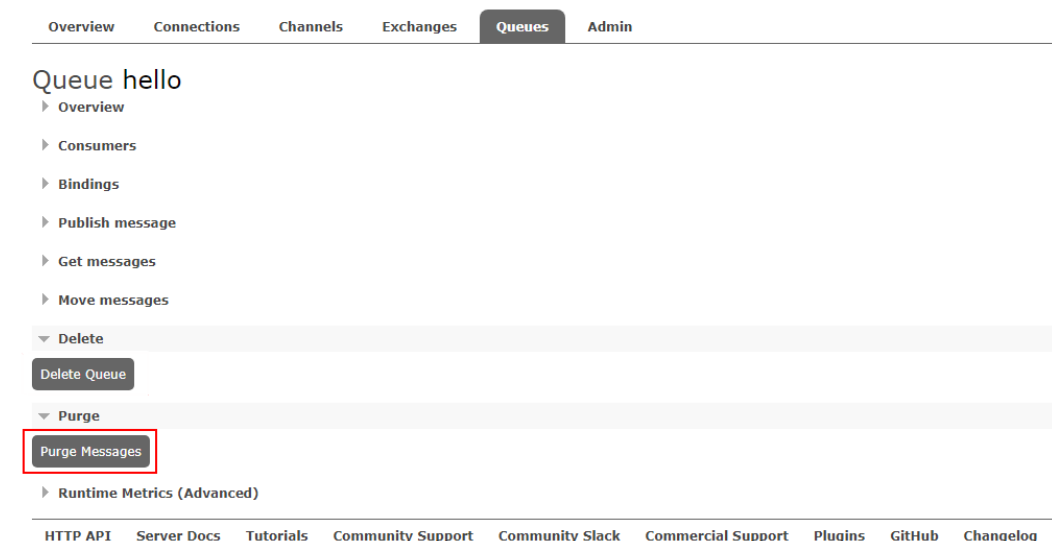
清空 Queue 消息（RabbitMQ WebUI）

- 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。



- 步骤3 单击“Purge Messages”，清空队列数据。

图 6-9 清空队列数据



⚠ 注意

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。

----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到该Queue。

约束与限制

解绑Queue后，对应的路由将不再提供服务，且无法恢复，请谨慎操作。

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RabbitMQ实例列表页面。

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“实例管理 > 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。



警告

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。

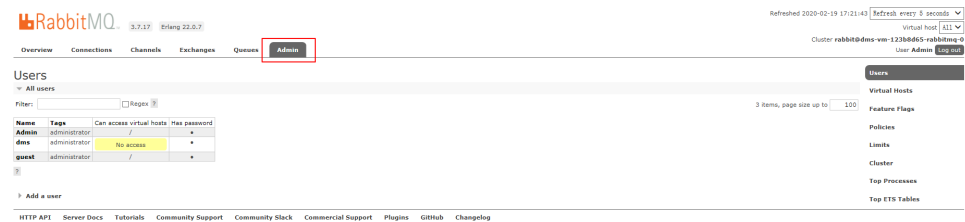
前提条件

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

Add / update a policy

Virtual host:

/

Name:

hello-ha

Pattern:

*hello

Apply to:

Exchanges and queues

Priority:

Definition:

ha-sync-mode

=

automatic

String

ha-mode

=

exactly

String

ha-params

=

2

Number

=

String

HA

HA mode

HA params

HA sync mode

HA mirror promotion on shutdown

HA mirror promotion on failure

Federation

Federation upstream set

Federation upstream

Queues

Message TTL

Auto expire

Max length

Max length bytes

Overflow behaviour

Dead letter exchange

Dead letter routing key

Lazy mode

Master Locator

Exchanges

Alternate exchange

Add policy

表 6-11 策略参数说明

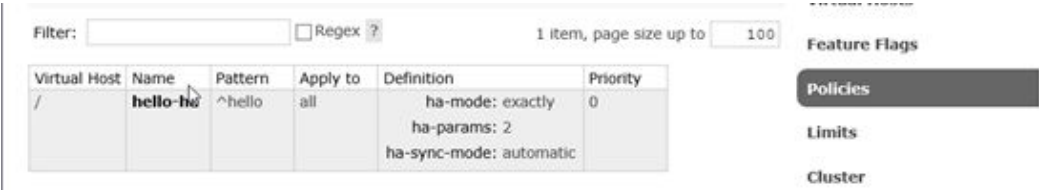
参数	说明
Virtual Host	选择策略所应用的Vhost。“/”表示默认Vhost。
Name	策略的名称，用户自定义。
Pattern	Queue的匹配模式（正则表达式）。
Apply to	策略所适用的目标。
Priority	策略的优先级，数字越大，优先级越高。

参数	说明
Definition	镜像定义，包括三个部分ha-sync-mode、ha-mode、ha-params。 ● 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。 - 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。 - 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。 说明 如果将队列镜像到集群所有节点上，可能会导致集群承受不必要的网络及磁盘I/O流量。因此，推荐您使用以下参数配置： ● ha-sync-mode: automatic ● ha-mode: exactly ● ha-params: $n/2+1$，其中n表示集群中总节点数量，$n/2$的值向下取整。 例如集群总节点数为3，ha-params设置为$3/2+1=2$，此时队列将镜像到一个主节点和一个从节点上。这样配置既保证了数据的高可用性，又避免了不必要的资源开销。

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。

处于以下情况时，**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](#)。

约束与限制

RabbitMQ 3.8.35及以上版本支持惰性队列。

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。

如果一个队列同时使用几种方式设置的话，Policy的方式具备更高的优先级。

- **Java客户端**：通过调用channel.queueDeclare方法的时候在参数中设置。
- **RabbitMQ WebUI**：通过Policy的方式设置。
- **RabbitMQ控制台**：创建队列时设置。

在 Java 客户端配置惰性队列

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);
```

在 RabbitMQ WebUI 配置惰性队列

步骤1 登录[RabbitMQ实例的Web UI](#)。

步骤2 在菜单栏，选择“Admin”。

图 6-14 选择 Admin 菜单



- 步骤3 在左侧导航栏选择“Policies”，进入“Policies”页面。
- 步骤4 新增一个策略。

图 6-15 新增一个策略

▼ Add / update a policy

Virtual host:

/ ▼

Name:

test *

Pattern:

Queue01 *

Apply to:

Queues ▼

Priority:

Definition:

queue-mode = lazy

String ▼

String ▼

Queues [All types]

Max length

Max length bytes

Overflow behaviour ?

Auto expire

Dead letter exchange

Dead letter routing key

Message TTL ?

表 6-12 策略参数说明

参数	说明
Virtual Host	设置策略所应用的Vhost。“/”表示默认Vhost。
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。选择“Queues”。
Priority	策略的优先级，数字越大，优先级越高。
Definition	通过“ queue-mode=lazy ”，配置惰性队列。

- 步骤5 单击“Add/update policy”。
- 策略添加成功后如下图所示。

图 6-16 查看惰性队列策略

Policies

▼ User policies

Filter:

☐ Regex ?

Virtual Host	Name	Pattern	Apply to	Definition	Priority
/	test	Queue01	queues	queue-mode: lazy	0

----结束

在 RabbitMQ 控制台配置惰性队列

- 步骤1 参考[创建RabbitMQ Queue](#)创建队列，并在“惰性队列模式”中输入“lazy”，完成惰性队列的创建。
- 步骤2 在Queue列表中，单击新创建的队列后的“查看详情”，进入“查看详情”页面。
当“惰性队列模式”显示“lazy”时，表示惰性队列创建成功。

图 6-17 查看队列详情



----结束

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](#)。

约束与限制

RabbitMQ 3.8.35及以上版本支持仲裁队列。

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6-13所示，且消耗更多的内存和磁盘。

表 6-13 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

在 Java 客户端配置仲裁队列

在声明队列时，将队列的“x-queue-type”参数设置为“quorum”。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);
```

在 RabbitMQ WebUI 配置仲裁队列

仲裁队列不能通过Policy设置，只能在创建队列时设置。

- 步骤1 登录RabbitMQ实例的Web UI。
- 步骤2 在菜单栏，选择“Queues”。
- 步骤3 创建一个仲裁队列。

图 6-18 创建仲裁队列

▼ Add a new queue

Virtual host:

/

▼

Type:

Quorum

▼

Name:

test

*

Node:

rabbit@dms-vm-c58192c2-rabbitmq-0

▼

Arguments:

=

String

▼

Add

Auto expire

?

|

Message TTL

?

|

Overflow behaviour

?

Single active consumer

?

|

Dead letter exchange

?

|

Dead letter routing key

?

Max length

?

|

Max length bytes

?

Delivery limit

?

|

Initial cluster size

?

Dead letter strategy

?

|

Leader locator

?

Add queue

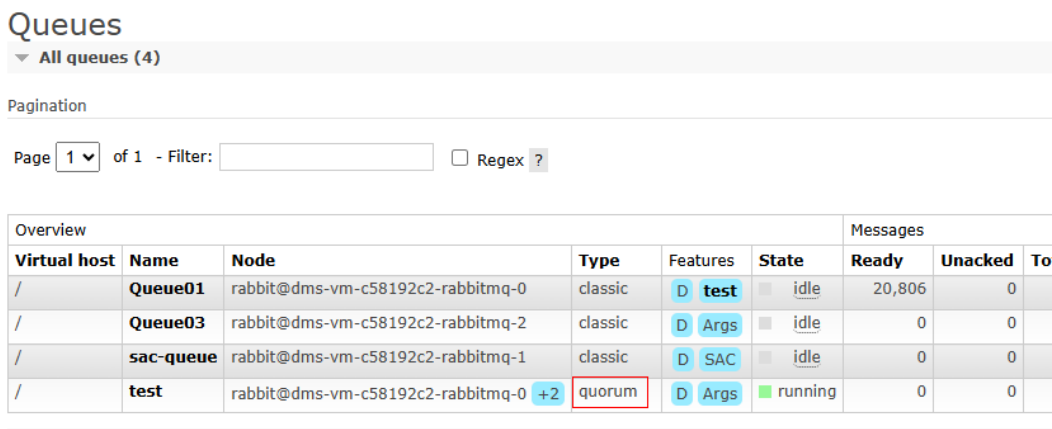
表 6-14 队列参数说明

参数名称	说明
Virtual Host	仲裁队列所属的Vhost。“/”表示默认Vhost。
Type	队列的类型，选择“Quorum”。

参数名称	说明
Name	仲裁队列的名称，用户自定义。
Node	仲裁队列部署的节点。
（可选）Arguments	扩展属性。无需设置。

- 步骤4 单击“Add queue”，完成仲裁队列的创建。
- 步骤5 在“Queues”页面查看“Type”为“quorum”，如图6-19所示，表示仲裁队列创建成功。

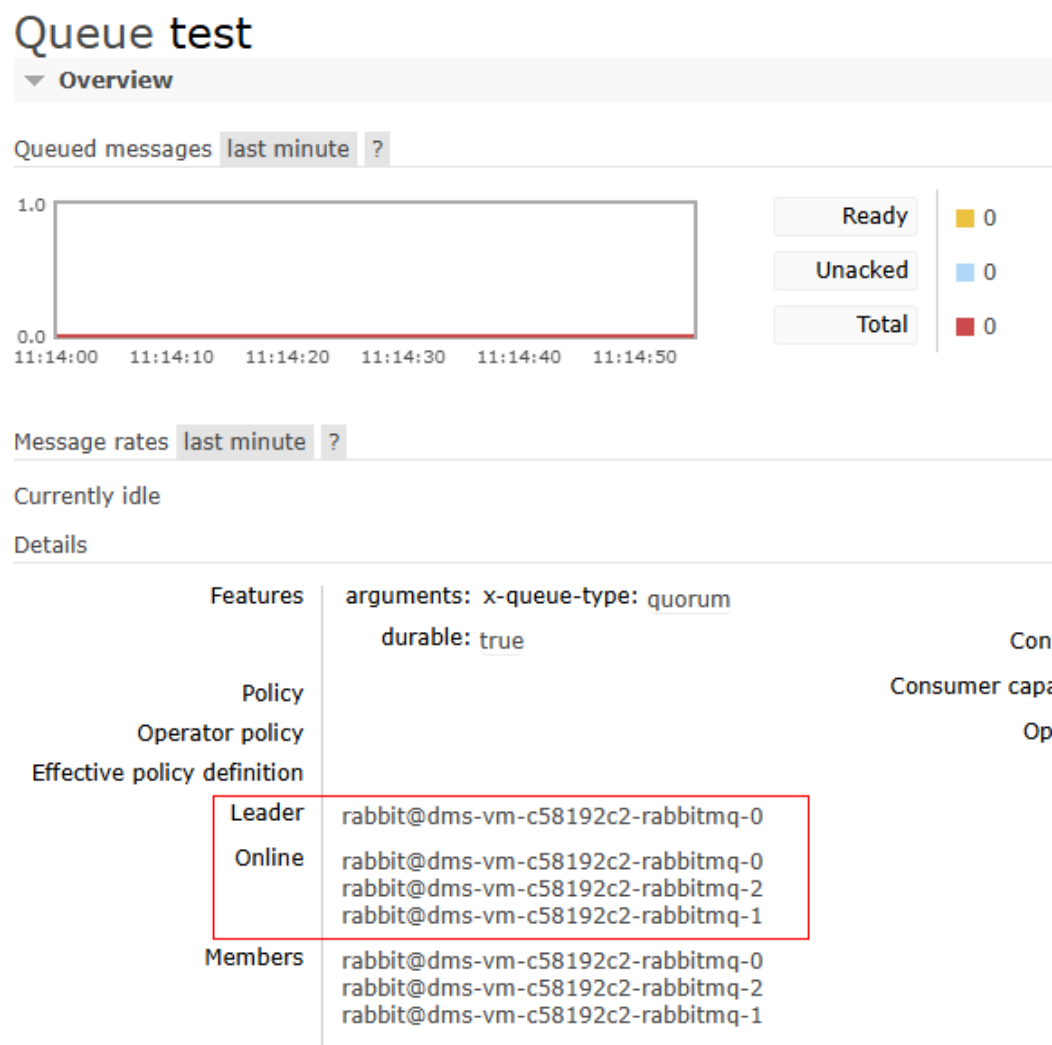
图 6-19 查看队列类型



“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。

- 步骤6 （可选）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 6-20 队列详情页



----结束

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。

表 6-15 配置仲裁队列的长度的参数说明

参数名称	说明
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。

参数名称	说明
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 6-21 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:

Exchanges and queues

Priority:

Definition:

max-in-memory-bytes

=

Number

=

String

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 ?

Add / update policy

- 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 6-22 使用队列属性设置 x-max-in-memory-length

Add a new queue

Type:

Quorum

Name:

Node:

rabbit@dms-vm-e942f915-rabbitmq-0

Arguments:

x-max-in-memory-length

=

Number

=

String

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 ?

Add queue

文档版本 05 (2025-09-03)

版权所有 © 华为云计算技术有限公司

69

6.8.7 配置 RabbitMQ 排他队列

排他队列只能被首次声明它的连接（Connection）可见和访问，包括生产消费消息、清空队列和删除队列等。其他连接无法访问该排他队列，也无法声明相同名称的排他队列。

排他队列无法持久化，一旦连接关闭，排他队列会被自动删除，队列中的数据也会被删除。

排他队列通常用于临时存储消息，且生产者和消费者使用相同连接的场景。

约束与限制

- 即使排他队列被声明为持久化队列，在连接断开时，该队列仍会被删除。
- 使用排他队列前，请谨慎评估队列被自动删除的风险，并考虑队列数据丢失场景。

配置 RabbitMQ 排他队列

以下示例演示在Java客户端设置排他队列。

```
boolean durable = false; //队列是否持久化
boolean exclusive = true; //队列是否排他
boolean autoDelete = true; //队列是否自动删除
Map<String, Object> arguments = new HashMap<>(); //队列的其他参数
channel.queueDeclare(EXCLUSIVE_QUEUE_NAME, durable, exclusive, autoDelete, arguments);
```

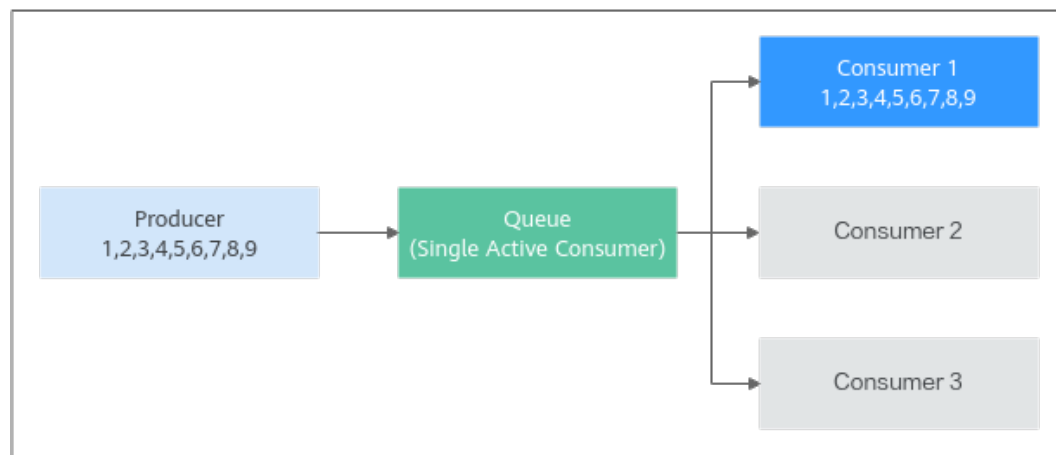
6.8.8 配置 RabbitMQ 单一活跃消费者

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。

如图6-23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 6-23 单一活跃消费者消费流程



约束与限制

RabbitMQ 3.8.35及以上版本支持单一活跃消费者特性。

在 Java 客户端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 = newHashMap<String, Object>();  
arguments.put("x-single-active-consumer", true);  
ch.queueDeclare("my-queue", false, false, false, arguments);
```

在 RabbitMQ WebUI 配置单一活跃消费者

- 步骤1 登录[RabbitMQ实例的Web UI](#)。
- 步骤2 在菜单栏，选择“Queues”。
- 步骤3 创建一个单一活跃消费者队列。

图 6-24 创建单一活跃消费者队列

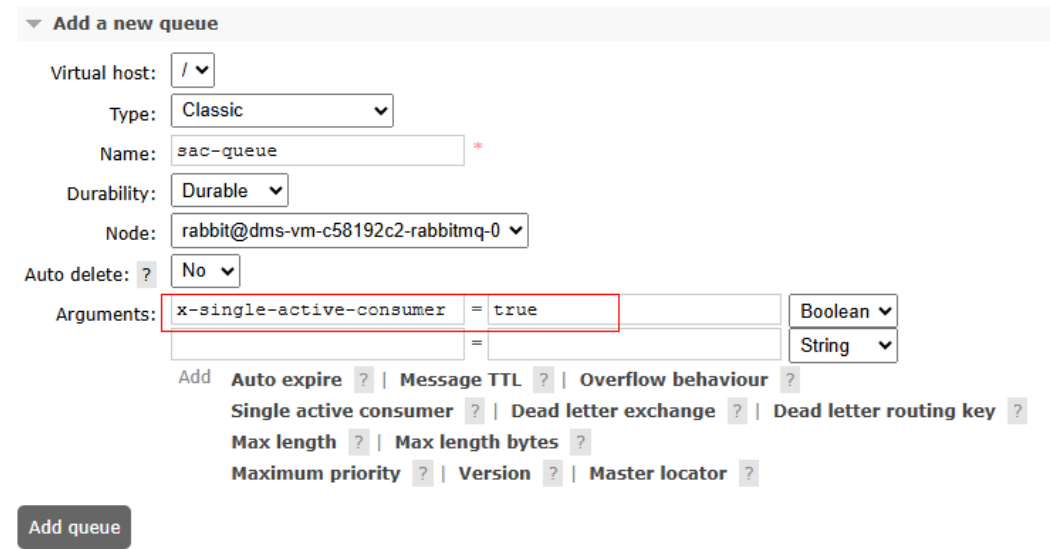


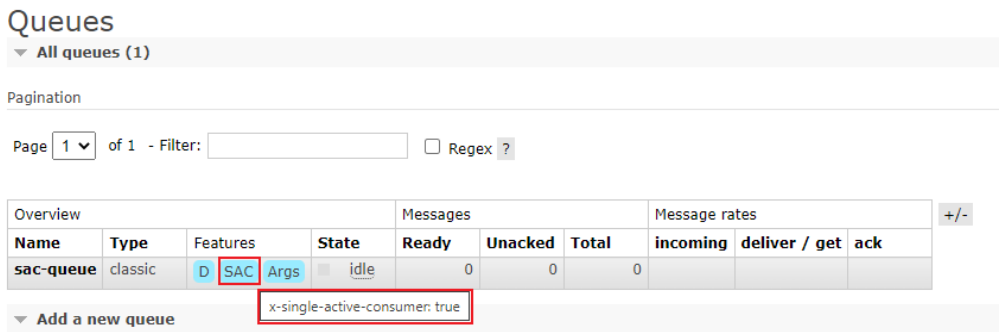
表 6-16 队列参数说明

参数名称	说明
Virtual Host	单一活跃消费者队列所属的Vhost。“/”表示默认Vhost。
Type	队列的类型。
Name	单一活跃消费者队列的名称，用户自定义。

参数名称	说明
Durability	是否开启持久化。 • Durable：该Queue在服务器重启后仍然存在。 • Transient：该Queue在服务器重启后会被删除，需要重新创建。
Node	单一活跃消费者队列部署的节点。
Auto delete	是否开启自动删除。 • Yes：当订阅该Queue消息的最后一个消费者取消订阅后，该Queue将被自动删除。 • No：当订阅该Queue消息的最后一个消费者取消订阅后，不会删除该Queue。
Arguments	通过“ x-single-active-consumer=true ”，配置单一活跃消费者属性。

- 步骤4 单击“Add queue”，完成单一活跃消费者队列的创建。
- 步骤5 在“Queues”页面查看队列特性是否包含单一活跃消费者。如图6-25所示，“SAC”即表示队列已设置单一活跃消费者。

图 6-25 查看队列特性



----结束

6.8.9 删除 RabbitMQ Queue

本章节指导如何删除Queue。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）](#)

约束与限制

- 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。

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RabbitMQ实例列表页面。
- 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“实例管理 > 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”。



警告

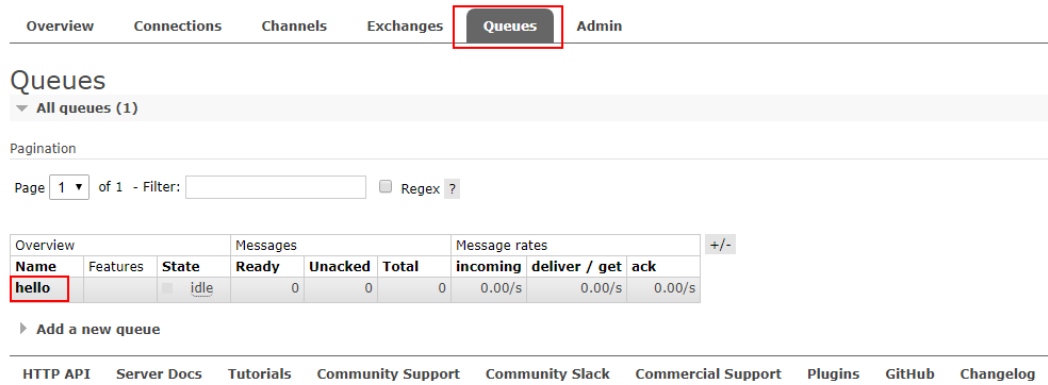
删除Queue后，该Queue下的所有配置（包括Exchange与Queue的绑定关系）也会被删除，且无法恢复。

- 步骤8 单击“确定”，完成Queue的删除。
- 删除的Queue已从Queue列表中移除，表明Queue已成功删除。
- 结束

删除单个 Queue（RabbitMQ WebUI）

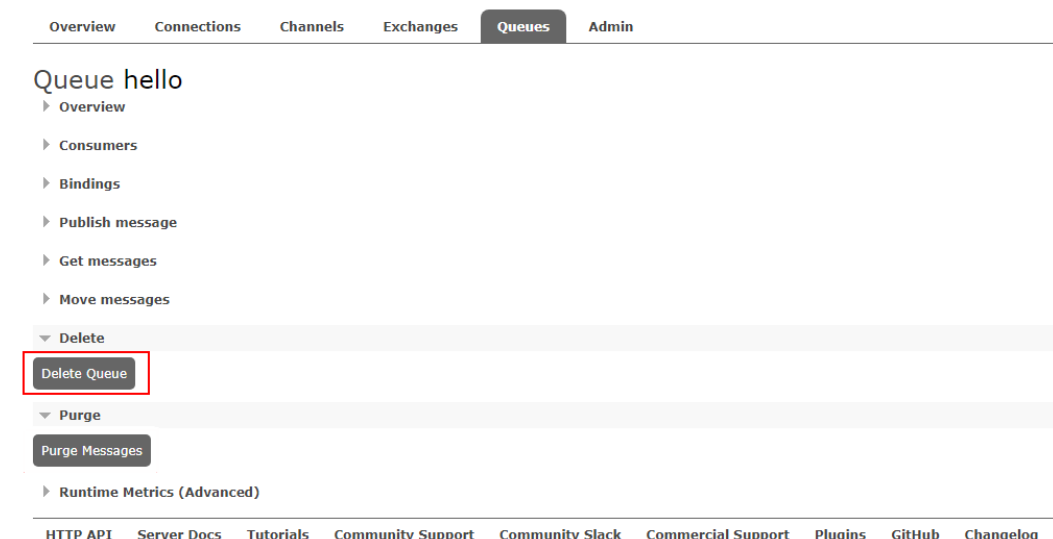
- 步骤1 登录RabbitMQ WebUI页面。
- 步骤2 在“Queues”页签，单击需要删除的队列名称，进入队列详情页面。

图 6-26 队列列表



步骤3 单击“Delete Queue”，删除单个队列。

图 6-27 删除单个队列



警告

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。

在“Overview”区域的Queue列表中，查看删除的Queue已从中移除，表明Queue已成功删除。

----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1ms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 6-28 通过 Policy 批量删除队列

OverviewConnectionsChannelsExchangesQueuesAdmin

Policies

▶ User policies

▼ Add / update a policy

Name:Delete queues*

Pattern:.**

Apply to:Queues▼

Priority:

Definition:expires=1Number▼

String▼

HA

HA mode? | HA params? | HA sync mode? | HA mirror promotion on shutdown

Federation

Federation upstream set? | Federation upstream?

Queues

Message TTL | Auto expire | Max length | Max length bytes | Overflow behaviour

Dead letter exchange | Dead letter routing key

Lazy mode | Master Locator

Exchanges

Alternate exchange?

Add policy

表 6-17 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ms。填写“expires”参数，值设置为“1”，表示队列过期时间为1ms。

警告

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。

步骤3 单击“Add policy”。

在“Queues”页签“Overview”区域的Queue列表中，查看删除的Queue已从中移除，表明Queue已成功删除。

文档版本 05 (2025-09-03)

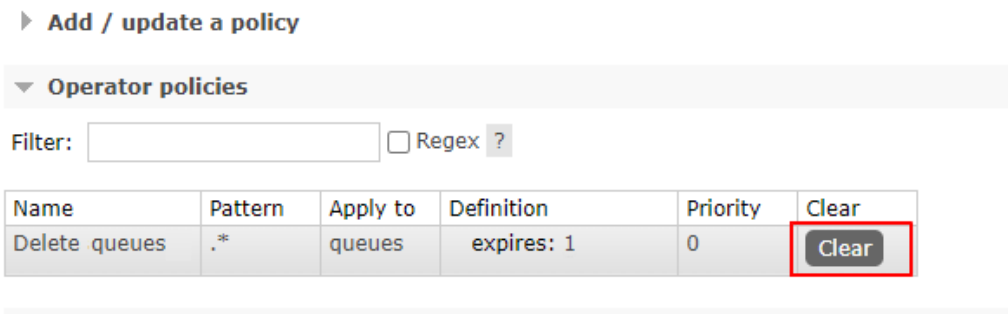
版权所有 © 华为云计算技术有限公司

75

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 6-29 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。	《虚拟私有云 用户指南》的“对等连接”章节

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。

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。

约束与限制

RabbitMQ控制台只支持绑定IPv4弹性IP地址，不支持绑定IPv6弹性IP地址。

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 IPv4 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RabbitMQ实例列表页面。
- 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10~30s，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。
- 步骤7** 在RabbitMQ实例的安全组中设置如表7-3所示规则后，才能成功通过IPv4弹性IP地址连接RabbitMQ。

表 7-3 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）

----结束

关闭 IPv4 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RabbitMQ实例列表页面。
- 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10~30s，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90s的场景，请在客户端开启心跳并设置小于90s的心跳超时时间，防止客户端断连。如果心跳超时时间设置过低，在短暂网络抖动情况下可能会导致连接重建。如果心跳超时时间设置过长，可能导致集群进行主备切换时感知时间过长。**推荐设置心跳超时时间为10s。**

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](#)。

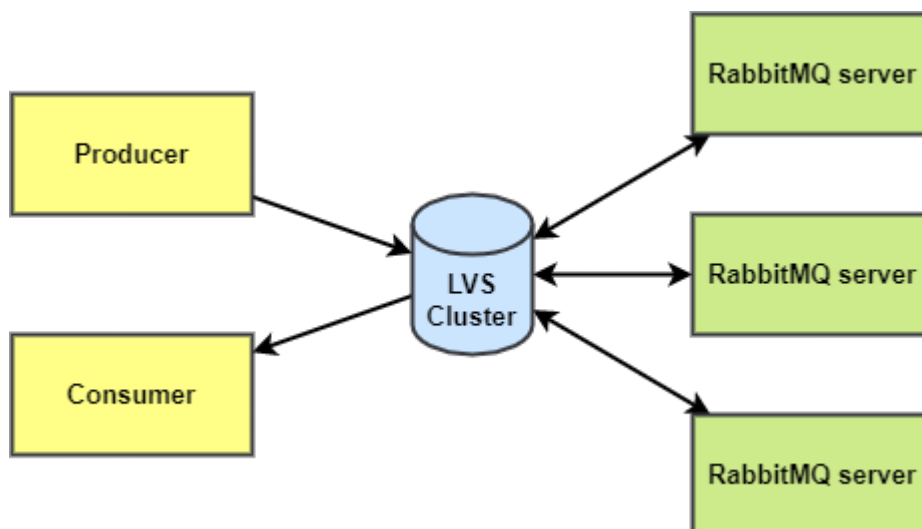
注意

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90s。如果客户端在90s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90s的场景，请在客户端开启心跳并设置小于90s的心跳超时时间。推荐设置心跳超时时间为10s。

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10s
cf.setRequestedHeartbeat(10);
```

- 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10s
cf.RequestedHeartbeat = TimeSpan.FromSeconds(10);
```

- 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10s
params = pika.ConnectionParameters(host='host', heartbeat=10,
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10s
$connection = new AMQPStreamConnection(RMQ_HOST, RMQ_PORT, RMQ_USER, RMQ_PASS,
RMQ_vhost, ['heartbeat'=> 10]);
```

- 使用Spring-amqp设置心跳超时时间，示例如下。

```
// 设置心跳时间为10s
@Bean
CachingConnectionFactory connectionFactory(ConnectionFactory rabbitConnectionFactory) {
    CachingConnectionFactory ccf = new CachingConnectionFactory(rabbitConnectionFactory);
```

```
ccf.setHost("...");
ccf.setRequestedHeartBeat(10);
// ...
return ccf;
}
```

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

以下操作命令以Linux系统为例进行说明：

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}
```

表 7-4 生产消息参数说明

参数名称	说明
host	RabbitMQ实例的连接地址，从 前提条件 中获取。
port	RabbitMQ实例的连接端口，输入5672。
user	连接RabbitMQ实例的用户名，从 前提条件 中获取。

参数名称	说明
password	连接RabbitMQ实例的密码，从前提条件中获取。

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}
```

表 7-5 消费消息参数说明

参数名称	说明
host	RabbitMQ实例的连接地址，从前提条件中获取。
port	RabbitMQ实例的连接端口，输入5672。
user	连接RabbitMQ实例的用户名，从前提条件中获取。
password	连接RabbitMQ实例的密码，从前提条件中获取。

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如下，需要修改的参数如表7-6所示。

```
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();
```

表 7-6 参数说明

参数名称	说明
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!	待发送的消息，根据实际需要修改。

- 连接实例并消费消息示例代码如下，需要修改的参数如表7-7所示。

```
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);
```

表 7-7 参数说明

参数名称	说明
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。

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 home 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

以下操作命令以Linux系统为例进行说明：

- 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。

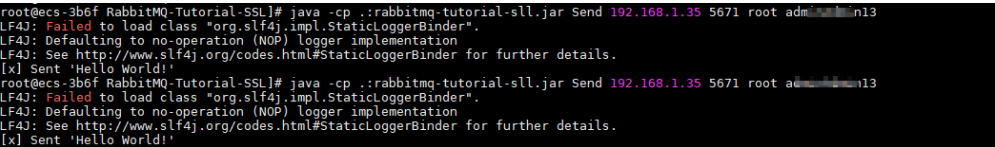
```
cd RabbitMQ-Tutorial-SSL
```
- 步骤5 运行生产消息示例。

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

表 7-8 生产消息参数说明

参数名称	说明
host	RabbitMQ实例的连接地址，从 前提条件 中获取。
port	RabbitMQ实例的连接端口，输入5671。
user	连接RabbitMQ实例的用户名，从 前提条件 中获取。
password	连接RabbitMQ实例的密码，从 前提条件 中获取。

图 7-2 生产消息示例



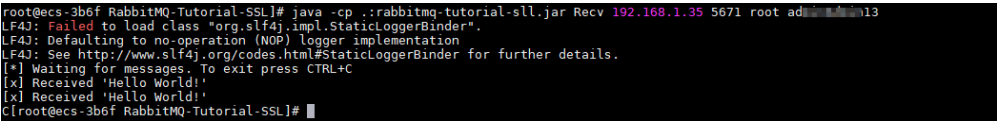
- 步骤6 运行消费消息示例。

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

表 7-9 消费消息参数说明

参数名称	说明
host	RabbitMQ实例的连接地址，从前提条件中获取。
port	RabbitMQ实例的连接端口，输入5671。
user	连接RabbitMQ实例的用户名，从前提条件中获取。
password	连接RabbitMQ实例的密码，从前提条件中获取。

图 7-3 消费消息示例



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如下，需要修改的参数如表7-10所示。

```
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();
```

表 7-10 参数说明

参数名称	说明
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!	待发送的消息，根据实际需要修改。

- 连接实例并消费消息示例代码如下，需要修改的参数如表7-11所示。

```
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( QUEUE_NAME, true, consumer);
```

表 7-11 参数说明

参数名称	说明
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。

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何一种情况，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](#)。

在RabbitMQ中，使用死信可能会对性能产生负面影响，请慎用。

在 Java 客户端配置 Exchange 和 Routing Key

为队列配置死信Exchange，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Exchange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);
```

在 RabbitMQ 控制台配置死信 Exchange 和 Routing Key

步骤1 参考[创建RabbitMQ Queue](#)创建队列，并在“死信交换器”中选择死信Exchange，“死信路由器”中输入死信Routing Key。

图 8-1 配置死信 Exchange 和 Routing Key

Create Queue

Queues are a buffer that stores messages. Each message is delivered to one or more queues.

Basic Settings

Name

Queue05

Fixed once created.

Persistence

☒

Auto-Delete

☐

More Settings ^

Dead Letter Exchange

DLQ01

Rejected and expired messages are resent to this exchange.

Dead Letter Routing Key

test

The dead letter exchange sends dead letter messages to the queue with a binding key that corresponds to this routing key.

Time to Live

Enter a time to live.

ms

How long messages published to queues can remain before being discarded.

Lazy Queue

Enter lazy to make the queue lazy.

Make the queue lazy to store more messages on disk and save memory. Otherwise, messages are stored in memory and delivered quickly.

步骤2 在Queue列表中，单击新创建的队列后的“查看详情”，进入“查看详情”页面。

当“死信交换器”和“死信路由键”为**步骤1**中设置的信息时，表示死信Exchange和Routing Key配置成功。

图 8-2 查看死信交换器和路由键

查看详情

名称	Queue05	所属Vhost	/
堆积消息	0	消费者数量	0
自动删除	否	死信交换器	DLQ01
死信路由键	test	消息过期时间 (ms)	--
惰性队列模式	--		

----结束

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。

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

约束与限制

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。

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。

约束与限制

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。

约束与限制

- 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。

在 RabbitMQ WebUI 设置 Exchange 持久化

步骤1 登录[RabbitMQ实例的Web UI](#)。

步骤2 在菜单栏，选择“Exchanges”。

步骤3 创建一个Exchange，并将“durable”设置为“true”，如[图9-1](#)所示。

图 9-1 设置 Exchange 持久化（WebUI）

▼ Add a new exchange

Name:

durable-exchange

*

Type:

direct

▼

Durability:

Durable

▼

Auto delete:

?

No

▼

Internal:

?

No

▼

Arguments:

durable

=

true

String

▼

=

String

▼

Add

Alternate exchange

?

Add exchange

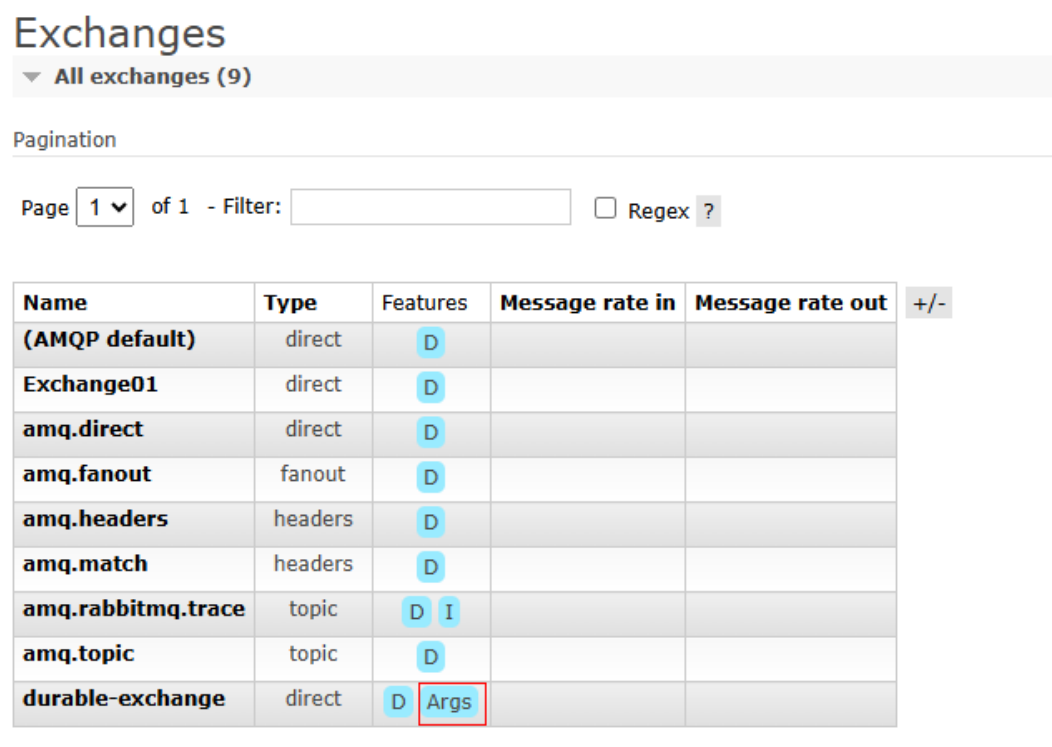
表 9-1 创建 Exchange 参数说明

参数名称	说明
Name	Exchange名称，用户自定义。
Type	Exchange类型。 在下拉框中选择。 • 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。
Durability	是否开启持久化。 • Durable: 该Exchange在服务器重启后仍然存在。 • Transient: 该Exchange在服务器重启后会被删除，需要重新创建。
Auto delete	是否开启自动删除。 • Yes: 当最后一个绑定的Queue与Exchange解除绑定后，该Exchange会被自动删除。 • No: 当最后一个绑定的Queue与Exchange解除绑定后，不会删除该Exchange。
Internal	Exchange是否为RabbitMQ内部使用。 • Yes: 该Exchange只能绑定其他Exchange，不能绑定Queue。 • No: 该Exchange可以绑定其他Exchange和Queue。

参数名称	说明
Arguments	通过“ durable=true ”，配置Exchange持久化。

步骤4 在Exchange列表中，将鼠标放在新创建Exchange的“Args”上，如图9-2所示，如果悬浮窗显示“durable: true”，表示配置Exchange持久化成功。

图 9-2 持久化的 Exchange（WebUI）



----结束

在 RabbitMQ 控制台设置 Exchange 持久化

步骤1 参考[创建RabbitMQ Exchange](#)创建Exchange，并设置Exchange持久化，如图9-3所示。

图 9-3 设置 Exchange 持久化（控制台）

Create Exchange

1.Exchanges route RabbitMQ messages.

2.Exchanges route messages based on the binding keys, routing keys, and headers.

3.Producers send messages to exchanges first, rather than directly to queues. Exchange route messages to one or more queues.

Name

Exchange-36202605

Fixed once created.

Target Type

direct

fanout

topic

headers

Automatically delete

Persistence

Internal

步骤2 在Exchange列表中，查看新创建Exchange是否已开启持久化。如果“持久化”显示“是”，表示开启成功，如图9-4所示。

图 9-4 持久化的 Exchange（控制台）

☐ Name	Default	Target Type	Persistence	Internal	Automatica...	Operation
☐ (AMQP default)	Yes	direct	Yes	No	No	Bind Exchange Delete
☐ Exchange-36202605	No	direct	Yes	No	No	Bind Exchange Delete

----结束

在 RabbitMQ WebUI 设置 Queue 持久化

- 步骤1 登录RabbitMQ实例的Web UI。
- 步骤2 在菜单栏，选择“Queues”。
- 步骤3 创建一个Queue，并设置“Durability”为“Durable”。

图 9-5 创建 Queues

▼ Add a new queue

Type: Classic ▼

Name: test

Durability: Durable ▼

Node: rabbit@dms-vm-76c3569c-rabbitmq-0 ▼

Auto delete: ? No ▼

Arguments: = String ▼

Add

Message TTL ?

|

Auto expire ?

|

Max length ?

|

Max length bytes

|

Dead letter exchange ?

|

Dead letter routing key ?

|

Single active co

|

Lazy mode ?

|

Master locator ?

Add queue

表 9-2 Queue 参数说明

参数名称	说明
Type	队列的类型。
Name	单一活跃消费者队列的名称，用户自定义。
Durability	是否开启持久化。设置“Durability”为“Durable”。 - Durable：该Queue在服务器重启后仍然存在。 - Transient：该Queue在服务器重启后会被删除，需要重新创建。
Node	单一活跃消费者队列部署的节点。
Auto delete	是否开启自动删除。 - Yes：当订阅该Queue消息的最后一个消费者取消订阅后，该Queue将被自动删除。 - No：当订阅该Queue消息的最后一个消费者取消订阅后，不会删除该Queue。
（可选）Arguments	Queue属性。无需设置。

步骤4 单击“Add queue”，完成持久化队列的创建。

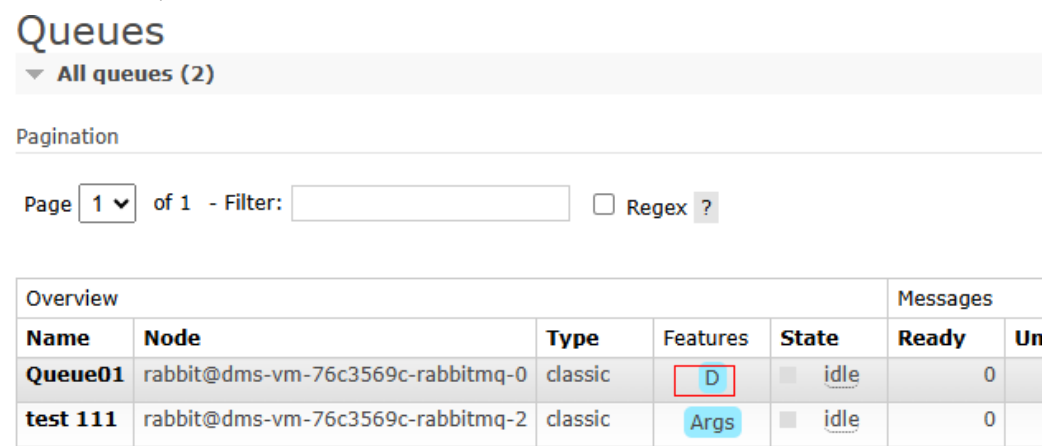
步骤5 在“Queues”页面查看新创建的Queue是否为持久化Queue。“Features”显示“D”时，表示此Queue为持久化Queue。

文档版本 05 (2025-09-03)

版权所有 © 华为云计算技术有限公司

97

图 9-6 查看 Queue 属性

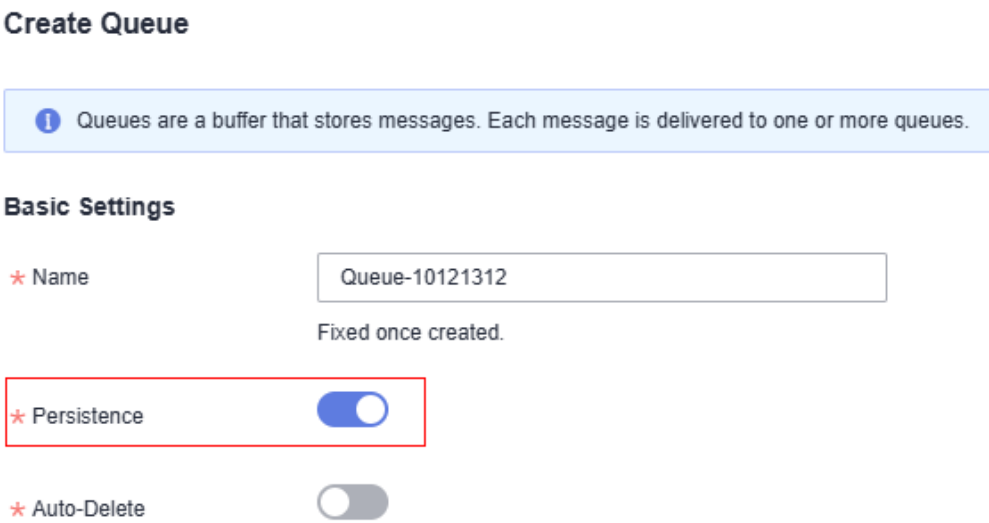


----结束

在 RabbitMQ 实例控制台设置 Queue 持久化

步骤1 参考[创建RabbitMQ Queue](#)创建Queue，并设置Queue持久化，如[图9-7](#)所示。

图 9-7 设置 Queue 持久化（控制台）



步骤2 在Queue列表中，查看新创建Queue是否已开启持久化。如果“持久化”显示“是”，表示开启成功，如[图9-8](#)所示。

图 9-8 持久化的 Queue（控制台）

☐	Name	Accumulated Messages	Persistence	Auto-Delete	Policy	Operation
☐	Queue-25718751	0	Yes	No		Clear Message Delete

----结束

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。

TTL是RabbitMQ中需要慎用的特性，它可能会对性能产生负面影响。

约束与限制

同时设置队列TTL和消息TTL时，以较小的TTL为准。

在 RabbitMQ 控制台设置队列 TTL

步骤1 参考[创建RabbitMQ Queue](#)创建队列，并设置“消息过期时间”。

图 9-9 设置消息过期时间

Create Queue

Queues are a buffer that stores messages. Each message is delivered to one or more queues.

Basic Settings

* Name
Fixed once created.

* Persistence ☒

* Auto-Delete ☐

More Settings ^

Dead Letter Exchange
Rejected and expired messages are resent to this exchange.

Dead Letter Routing Key
The dead letter exchange sends dead letter messages to the queue with a binding key that corresponds to this routing key.

Time to Live ms
How long messages published to queues can remain before being discarded.

Lazy Queue
Make the queue lazy to store more messages on disk and save memory. Otherwise, messages are stored in memory and delivered quickly.

步骤2 在Queue列表中，单击新创建的队列后的“查看详情”，进入“查看详情”页面。

当“消息过期时间”显示**步骤1**中设置的时间时，表示队列TTL设置成功。

图 9-10 查看消息过期时间

查看详情

名称	Queue06	所属Vhost	/
堆积消息	0	消费者数量	0
自动删除	否	死信交换器	--
死信路由键	--	消息过期时间 (ms)	300
惰性队列模式	--		

----结束

在 Java 客户端设置队列 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ms。

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1800000ms
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ms。

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6000ms
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ms。

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60000ms
    .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RabbitMQ实例列表页面。
- 步骤4** RabbitMQ实例支持通过筛选功能查询目标实例。当前支持的筛选条件为名称、状态、版本、实例类型、规格、已用/可用存储空间、计费模式、可用区、实例ID、内网连接地址、公网连接地址、企业项目和资源标签。
- 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。
创建失败	RabbitMQ实例处于创建实例失败的状态。

状态	说明
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。
回收中	RabbitMQ实例正在进行临时删除操作。
回收站中	RabbitMQ实例处于回收站中。
恢复中	RabbitMQ实例正在从回收站中恢复。

步骤5 单击RabbitMQ实例的名称，进入该RabbitMQ实例的概览页面，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。
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。

----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RabbitMQ实例列表页面。
- 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。	页面上方直接提示修改结果。
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。	页面上方直接提示修改结果。
描述	单击  ，输入修改后的描述，单击  。 长度为 0~1024 个字符。	页面上方直接提示修改结果。
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。	页面上方直接提示修改结果。
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。

约束与限制

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

Connections

All connections (4)

Page 1 of 1 - Filter: ☐ Regex ?

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	

HTTP API Server Docs Tutorials Community Support Community Slack Commercial Support Plugins GitHub Changelog

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）

Connections

All connections (2)

Page 1 of 1 - Filter: ☐ Regex ?

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	

HTTP API Server Docs Tutorials Community Support Community Slack Commercial Support Plugins GitHub

----结束

10.3 查看 RabbitMQ 实例后台任务

对实例的一些操作，如表10-4所示，会启动一个后台任务，您可以在后台任务页，查看该操作的状态等信息，同时可通过删除操作，清理任务信息。

表 10-4 后台任务列表

任务名	说明
创建实例	创建一个RabbitMQ实例。

任务名	说明
插件变更	- 开启插件 - 关闭插件
开启公网访问	开启公网访问。
关闭公网访问	关闭公网访问。
变更规格	- 扩容存储空间 - 扩容代理数量 - 扩容代理规格 - 缩容代理规格

查看 RabbitMQ 实例后台任务

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 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 RabbitMQ 实例列表页面。

步骤4 单击 RabbitMQ 实例的名称，进入实例的“概览”页面。

步骤5 在左侧导航栏单击“实例管理 > 后台任务管理”，进入后台任务管理页面。

步骤6 在“即时任务”页签，单击时间段下拉框，选择搜索时间，在搜索框中输入关键字，按“Enter”，界面显示相应时间段内启动的任务。

在后台任务管理页面，您还可以执行以下操作：

- 单击 ，刷新任务状态。
- 单击“删除”，弹出“删除任务记录”对话框，单击“确定”，清理任务信息。
您只能在任务已经执行完成，即任务状态为成功或者失败时，才能执行删除操作。

----结束

10.4 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。

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

标签共由两部分组成：“标签键”和“标签值”，其中，“标签键”和“标签值”的命名规则如表 10-5 所示。

表 10-5 标签命名规则

参数名称	规则
标签键	- 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	- 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。

约束限制

一个RabbitMQ实例最多添加20个标签。

配置单个 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RabbitMQ实例列表页面。
- 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 在左侧导航栏单击“实例管理 > 标签”，进入标签列表页面。
- 步骤6 单击“编辑标签”，弹出“编辑标签”对话框。
- 步骤7 您可以根据实际需要，添加/删除/编辑标签。

表 10-6 标签操作说明

操作类型	操作步骤
添加标签	1. 单击“添加标签”，在“标签键”和“标签值”中，输入标签的键/值。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。最多可以添加20个标签。 2. 单击“确定”，成功为实例添加标签。在标签列表页面，查看新添加的标签。

操作类型	操作步骤
编辑标签	修改标签键/值，单击“确定”，完成标签的修改。 在标签列表页面，查看修改后的标签键/值。
删除标签	在待删除的标签所在行，单击“删除”。然后单击“确定”，完成标签的删除。 已删除的标签不再显示在标签列表中，表明标签已成功删除。

----结束

10.5 配置 RabbitMQ 实例回收站策略

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

回收站策略默认是关闭状态。

约束与限制

- 回收站中的按需实例不会收取实例的费用，但是会收取存储空间的费用。

开启回收站策略

步骤1 登录管理控制台。

步骤2 在管理控制台左上角单击 ，选择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RabbitMQ实例列表页面。

步骤4 在左侧导航栏选择“回收站”，进入“回收站”页面。

步骤5 单击“回收站策略”，弹出“回收站策略”对话框。

步骤6 开启启用回收站，并设置保留天数（只能设置为1~7天），单击“确定”。

----结束

恢复 RabbitMQ 实例

步骤1 登录管理控制台。

步骤2 在管理控制台左上角单击 ，选择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RabbitMQ实例列表页面。

步骤4 在左侧导航栏选择“回收站”，进入“回收站”页面。

步骤5 通过以下任意一种方法，恢复RabbitMQ实例。

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

步骤6 在弹出的“恢复实例”对话框中，单击“确定”。

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

----结束

修改保留天数

步骤1 登录管理控制台。

步骤2 在管理控制台左上角单击 ，选择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RabbitMQ实例列表页面。

步骤4 在左侧导航栏选择“回收站”，进入“回收站”页面。

步骤5 单击“回收站策略”，弹出“回收站策略”对话框。

步骤6 修改保留天数（只能设置为1~7天），单击“确定”。

修改保留天数仅对修改后删除的实例生效，修改前已在回收站中的实例依然按照修改前的天数保留实例。

----结束

导出回收站中的实例

步骤1 登录管理控制台。

步骤2 在管理控制台左上角单击 ，选择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RabbitMQ实例列表页面。

步骤4 在左侧导航栏选择“回收站”，进入“回收站”页面。

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

----结束

删除回收站中的实例

步骤1 登录管理控制台。

步骤2 在管理控制台左上角单击 ，选择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RabbitMQ实例列表页面。

步骤4 在左侧导航栏选择“回收站”，进入“回收站”页面。

步骤5 通过以下任意一种方法，删除实例。

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

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

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

----结束

关闭回收站策略

步骤1 登录管理控制台。

步骤2 在管理控制台左上角单击 ，选择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RabbitMQ实例列表页面。

步骤4 在左侧导航栏选择“回收站”，进入“回收站”页面。

步骤5 单击“回收站策略”，弹出“回收站策略”对话框。

步骤6 关闭启用回收站，单击“确定”。

----结束

10.6 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RabbitMQ实例列表页面。

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“... > 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。

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

10.7 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10-7](#)所示。其中，**插件端口号不支持修改**。

表 10-7 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

插件名称	功能描述	端口号
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10-8所示。

表 10-8 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
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RabbitMQ实例列表页面。
- 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 在左侧导航栏选择“实例管理 > 插件管理”，然后在待开启的插件所在行单击“开启”。
- 确认开启后，等待实例插件开启成功。
- 结束

10.8 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。

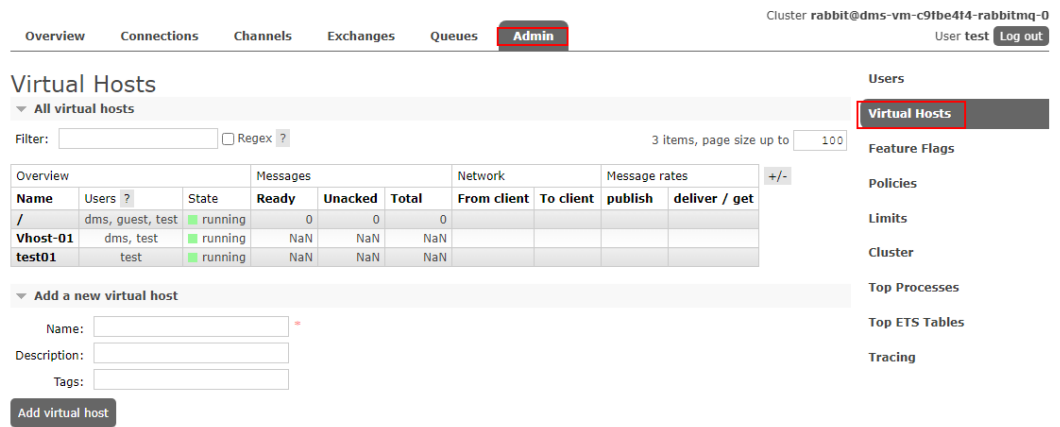
前提条件

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 进入 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

Virtual Hosts

All virtual hosts

Filter: ☐ Regexp ?

Overview			Messages			Network		Message rates		+/-
Name	Users ?	State	Ready	Unacked	Total	From client	To client	publish	deliver / get	
/	dms, guest, test	running	0	0	0					
Vhost-01	dms, test	running	NaN	NaN	NaN					
test01	test	running	NaN	NaN	NaN					

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

OverviewConnectionsChannelsExchangesQueuesAdmin

Virtual Host: Vhost-01

Overview

Permissions

Current permissions

User	Configure regexp	Write regexp	Read regexp	
dms	.*	.*	.*	Clear
test	.*	.*	.*	Clear

Set permission

User

guest

Configure regexp:.*

Write regexp:.*

Read regexp:.*

Set permission

步骤7 在右侧导航栏选择“Tracing”，进入Tracing页面。

图 10-6 Admin 页面

OverviewConnectionsChannelsExchangesQueuesAdmin

User testLog out

Users

All users

Filter: ☐ Regexp ?

3 items, page size up to 100

Name	Tags	Can access virtual hosts	Has password
dms	administrator	No access	•
guest	administrator	/	•
test	administrator	/	•

?

Add a user

Users

Virtual Hosts

Feature Flags

Policies

Limits

Cluster

Top Processes

Top ETS Tables

Tracing

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 10-9 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。
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： • #：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

▼ Add a new trace

Virtual host:

Vhost-01

Name:

Format:

Text

Tracer connection username:

Tracer connection password:

Max payload bytes: ?

Pattern:

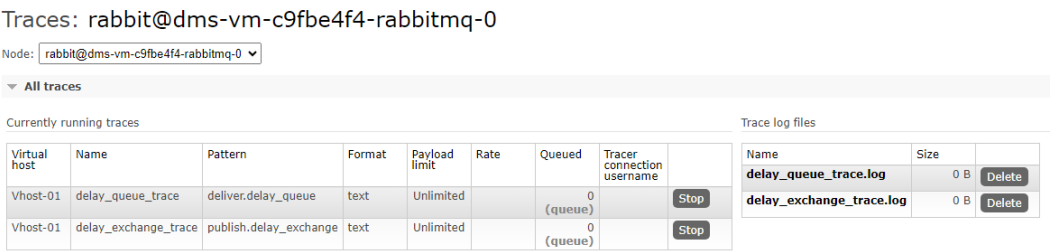
#

Examples: #, publish.#, deliver.# #.amq.direct, #.myqueue

Add trace

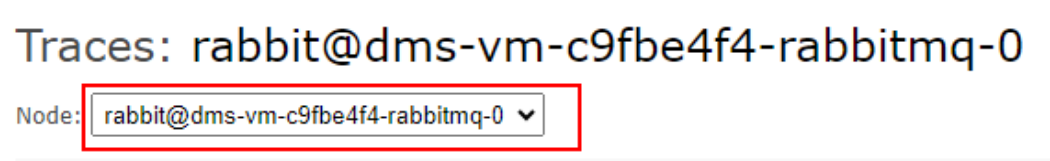
trace创建成功后，在“All traces”区域，显示已创建的trace列表。

图 10-8 trace 列表



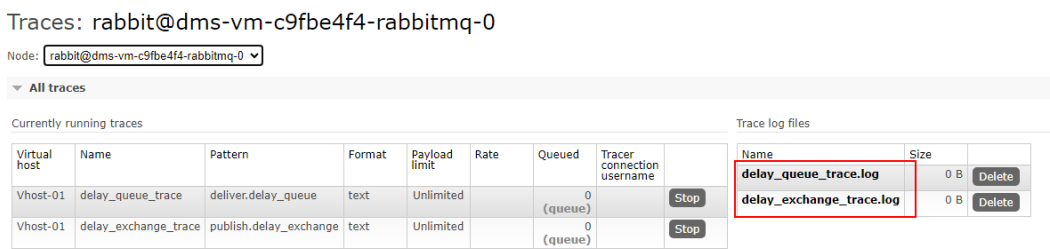
步骤9 （可选）如果RabbitMQ实例为集群，在“Node”中切换到其他节点，重复**步骤8**，为其他所有节点创建trace。

图 10-9 切换节点



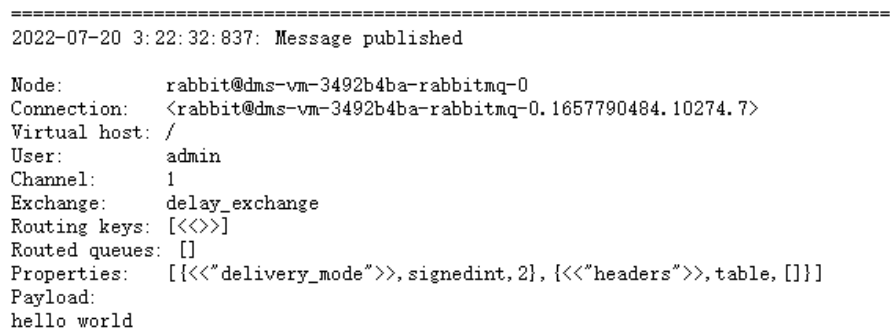
步骤10 当trace日志文件中存入消息日志后，单击trace日志文件名称，查看日志内容。

图 10-10 trace 日志文件



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

图 10-11 delay_exchange_trace.log



“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

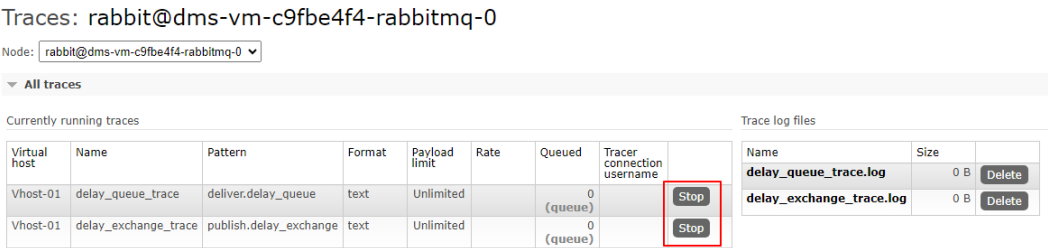
Node:      rabbit@dms-vm-3492b4ba-rabbitmq-0
Connection: <rabbit@dms-vm-3492b4ba-rabbitmq-0.1657790484.10565.7>
Virtual host: /
User:      admin
Channel:   1
Exchange:
Routing keys: [<<"delay_queue">>]
Queue:     delay_queue
Properties: [{<<"delivery_mode">>,signedint,1},{<<"headers">>,table,[]}]
Payload: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

10.9 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RabbitMQ实例列表页面。

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

10.10 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。

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

前提条件

RabbitMQ实例状态为运行中、故障、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RabbitMQ实例列表页面。

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“... > 删除”。



警告

如果未开启回收站策略，实例删除后，实例中原有的数据将被删除，且没有备份。

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60s。

已删除的RabbitMQ实例不再显示在实例列表中，表明RabbitMQ实例已成功删除。

----结束

10.11 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。

前提条件

已获取创建RabbitMQ实例时设置的用户名和密码。如果您忘记了密码，请参考[重置RabbitMQ实例密码](#)重置密码。

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RabbitMQ实例所在的区域。
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RabbitMQ实例列表页面。
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。

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“概览 > 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，需要允许端口15672的访问。
 - 实例已开启SSL开关，需要允许端口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。

- 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 变更实例规格

11.1 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	代理数量	√	×
	存储空间	√	×
	代理规格	√	√
单机	代理数量	×	×
	存储空间	√	×
	代理规格	√	√

变更实例规格的影响

表 11-2 变更实例规格的影响

变更配置类型	影响
代理数量	- 扩容代理数量过程中会重启RabbitMQ的负载均衡进程，导致节点主备倒换，造成秒级业务中断，客户端需要支持自动重连，建议在业务低峰时进行变更。 - 变更时长和新增代理数量有关，单个代理的变更时长一般在5~10分钟，总变更时长为（5~10分钟）*新增代理数量。

变更配置类型	影响
存储空间	- 扩容存储空间不会影响业务。 - 扩容存储空间有次数限制，每个实例只能扩容20次。 - 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。 - 预计总变更时长为5分钟内。
代理规格	- RabbitMQ 3.x.x版本单机实例在变更规格过程中，节点会重启，造成分钟级业务中断，客户端需要支持自动重连，建议在业务低峰时进行变更。 - RabbitMQ 3.x.x版本没有配置镜像/仲裁队列的集群实例在变更规格过程中，节点会逐个重启，造成分钟级业务中断，客户端需要支持自动重连，建议在业务低峰时进行变更。 - RabbitMQ 3.x.x版本配置了镜像/仲裁队列的集群实例在变更规格过程中，节点会逐个重启，造成秒级业务中断，客户端需要支持自动重连，建议在业务低峰时进行变更。 - RabbitMQ 3.x.x版本集群实例在变更规格过程中，变更中的节点连接会切换至其余节点，导致其余节点存在过载风险（比如超过连接数、内存高水位等），建议您在实例规格范围内使用，具体实例规格请参考产品规格。 Exchange、Queue和消息需要配置持久化，否则节点重启后，消息可能丢失。设置持久化的方法，请参考配置RabbitMQ持久化。 - 代理采用滚动重启的方式进行实例变更，变更时长和代理数量有关，单个代理的变更时长一般在5~10分钟，总变更时长为（5~10分钟）*代理数量。

以下示例演示在Java客户端设置消息重试。

```
ConnectionFactory connectionFactory = new ConnectionFactory();
//设置服务地址
connectionFactory.setHost("localhost");
//设定端口
connectionFactory.setPort(5672);
//自动重试:
connectionFactory.setAutomaticRecoveryEnabled(true);
connectionFactory.setNetworkRecoveryInterval(5);
connectionFactory.setTopologyRecoveryEnabled(true);
```

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”。

扩容存储空间（3.x.x 版本）

步骤1 登录管理控制台。

步骤2 在管理控制台左上角单击 ，选择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RabbitMQ实例列表页面。

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“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“变更规格”。

步骤5 根据实际情况扩容存储空间。

1. 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。不同实例规格对应的存储空间范围不同，具体请参考[产品规格](#)，单击“下一步”。
2. 确认扩容信息无误后，单击“提交”。
3. 在实例列表页面，查看变更是否成功？
 - 实例状态从“变更中”变为“运行中”时，表示变更成功，在“已用/可用存储空间”中查看扩容后的存储空间大小（即扩容后的单个代理的存储空间*代理数量）。
 - 实例状态从“变更中”变为“变更失败”时，表示变更失败，把鼠标放在“变更失败”上，显示变更失败的原因。

实例状态为“变更失败”时，无法对实例进行修改和删除等操作。等一段时间实例状态会自动从“变更失败”变为“运行中”，此时可以继续对实例进行操作。如果未变为“运行中”，请联系客服处理。

----结束

扩容代理数量（3.x.x 版本）

步骤1 登录管理控制台。

步骤2 在管理控制台左上角单击 ，选择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RabbitMQ实例列表页面。

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“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“变更规格”。

步骤5 根据实际情况扩容代理数量。

1. 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。不同实例规格对应的代理数量范围不同，具体请参考[产品规格](#)。如果已开启公网访问，在“弹性IP地址”中，选择新扩容代理的公网IP地址，单击“下一步”。
2. 确认扩容信息无误后，单击“提交”。

3. 在实例列表页面，查看变更是否成功？

- 实例状态从“变更中”变为“运行中”时，表示变更成功，在“规格”中查看扩容后的代理数量。
- 实例状态从“变更中”变为“变更失败”时，表示变更失败，把鼠标放在“变更失败”上，显示变更失败的原因。

实例状态为“变更失败”时，无法对实例进行修改和删除等操作。等一段时间实例状态会自动从“变更失败”变为“运行中”，此时可以继续对实例进行操作。如果未变为“运行中”，请联系客服处理。

----结束

扩容/缩容代理规格（3.x.x 版本）

步骤1 登录管理控制台。

步骤2 在管理控制台左上角单击 ，选择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RabbitMQ实例列表页面。

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“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“变更规格”。

步骤5 根据实际情况扩容/缩容代理规格。

1. 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，单击“下一步”。
2. 确认扩容/缩容信息无误后，单击“提交”。
3. 在实例列表页面，查看变更是否成功？
 - 实例状态从“变更中”变为“运行中”时，表示变更成功，在“规格”中查看扩容/缩容后的代理规格。
 - 实例状态从“变更中”变为“变更失败”时，表示变更失败，把鼠标放在“变更失败”上，显示变更失败的原因。

实例状态为“变更失败”时，无法对实例进行修改和删除等操作。等一段时间实例状态会自动从“变更失败”变为“运行中”，此时可以继续对实例进行操作。如果未变为“运行中”，请联系客服处理。

----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。

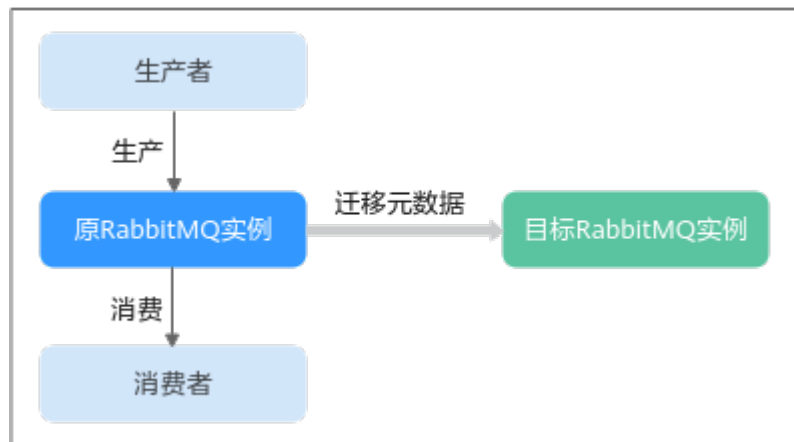
前提条件

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。

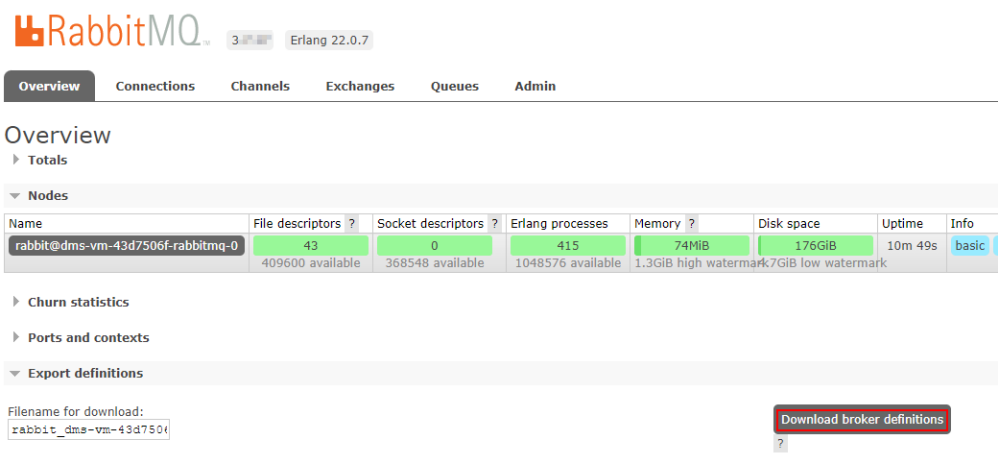
图 12-1 迁移元数据



迁移步骤如下：

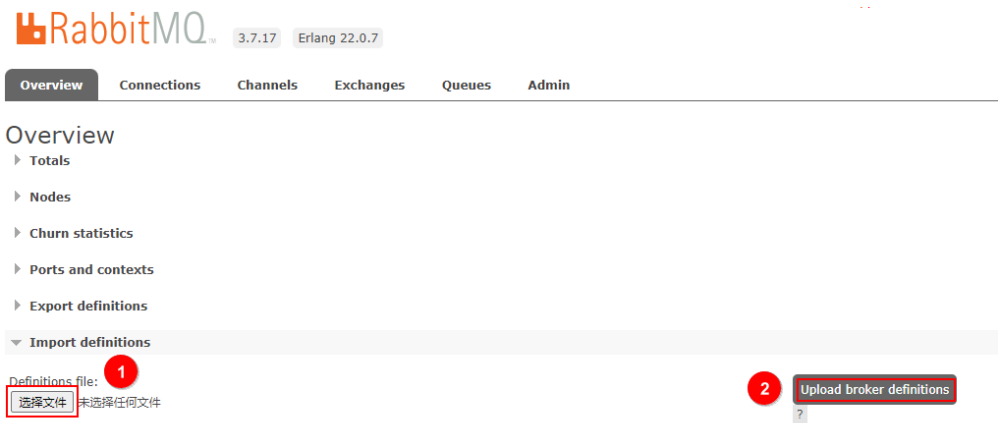
1. 登录原RabbitMQ的WebUI页面，在“Overview”页签中，单击“Download broker definitions”，导出元数据。

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



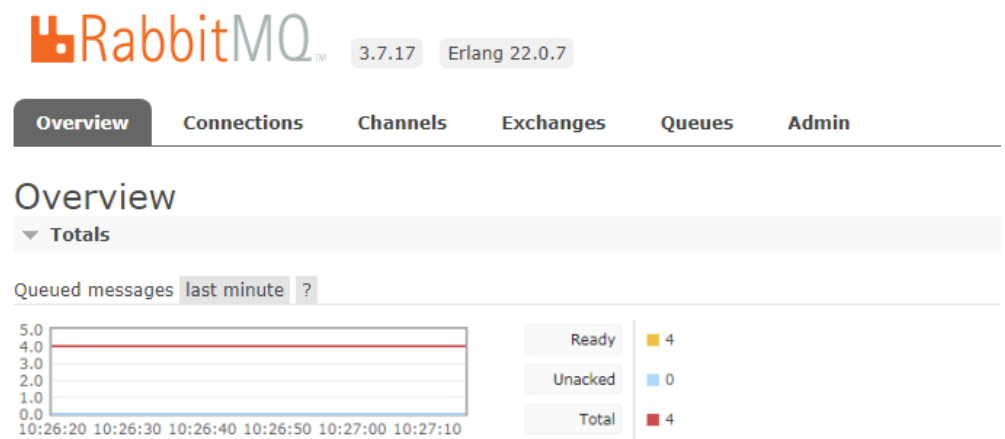
----结束

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`

表 12-1 参数说明

参数名称	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RabbitMQ实例列表页面。

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 > 监控详情”，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或API接口来检索RabbitMQ实例产生的监控指标和告警信息，也可以通过分布式消息服务RabbitMQ版控制台提供的“监控详情”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	单位	进制	测量对象（维度）	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。	>= 0	Count	不涉及	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。	0~2047	Count	不涉及	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。	0~7000	Count	不涉及	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。	0~280000	Count	不涉及	RabbitMQ实例	1分钟
messages_ready	可消费消息数	该指标用于统计RabbitMQ实例中总可消费消息数量。	0~10000000	Count	不涉及	RabbitMQ实例	1分钟
messages_unacknowledged	未确认消息数	该指标用于统计RabbitMQ实例中总已经消费但还未确认的消息数量。	0~10000000	Count	不涉及	RabbitMQ实例	1分钟

指标ID	指标名称	指标含义	取值范围	单位	进制	测量对象（维度）	监控周期（原始指标）
publish	生产速率	统计RabbitMQ实例中实时消息生产速率。	0~25000	Count/s	不涉及	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）。	0~25000	Count/s	不涉及	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）。	0~50000	Count/s	不涉及	RabbitMQ实例	1分钟
connections_state_running	运行状态的连接个数	该指标用于统计整个实例中的connection，状态是starting/tuning/opening/running状态的总数。	≥ 0	Count	不涉及	RabbitMQ实例	1分钟
connections_state_flow	flow状态的连接数	该指标用于统计整个实例中的connection，状态是flow状态的总数。	≥ 0	Count	不涉及	RabbitMQ实例	1分钟
connections_state_block	block状态的连接数	该指标用于统计整个实例中的connection，状态是blocking/blocked状态的总数。	≥ 0	Count	不涉及	RabbitMQ实例	1分钟
connections_state_close	close状态的连接数	该指标用于统计整个实例中的connection，状态是closing/closed状态的总数。	≥ 0	Count	不涉及	RabbitMQ实例	1分钟

指标ID	指标名称	指标含义	取值范围	单位	进制	测量对象（维度）	监控周期（原始指标）
channels_states_running	运行状态的通道数	该指标用于统计整个实例中的channel，状态是starting/tuning/opening/running状态的总数。	≥ 0	Count	不涉及	RabbitMQ实例	1分钟
channels_states_flow	flow状态的通道数	该指标用于统计整个实例中的channel，状态是flow状态的总数。	≥ 0	Count	不涉及	RabbitMQ实例	1分钟
channels_states_block	block状态的通道数	该指标用于统计整个实例中的channel，状态是blocking/blocked状态的总数。	≥ 0	Count	不涉及	RabbitMQ实例	1分钟
channels_states_close	close状态的通道数	该指标用于统计整个实例中的channel，状态是closing/closed状态的总数。	≥ 0	Count	不涉及	RabbitMQ实例	1分钟
queue_s_states_running	运行状态的队列数	该指标用于统计整个实例中的queue，状态是running状态的总数。	≥ 0	Count	不涉及	RabbitMQ实例	1分钟
queue_s_states_flow	flow状态的队列数	该指标用于统计整个实例中的queue，状态是flow状态的总数。	≥ 0	Count	不涉及	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	单位	进制	测量对象（维度）	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。	0~65535	Count	不涉及	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。	0~50000	Count	不涉及	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。	0~1048576	Count	不涉及	RabbitMQ 实例节点	1分钟
mem_used	内存占用	该指标用于统计当前节点 RabbitMQ 内存占用。	0~3200000000	Byte	1024(IEC)	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。	0~5000000000	Byte	1024(IEC)	RabbitMQ 实例节点	1分钟
rabbitmq_alive	节点存活状态	表示 RabbitMQ 节点是否存活。	1: 存活 0: 离线	不涉及	不涉及	RabbitMQ 实例节点	1分钟
rabbitmq_disk_usage	磁盘容量使用率	统计 RabbitMQ 节点虚拟机的磁盘容量使用率。	0~100	%	不涉及	RabbitMQ 实例节点	1分钟
rabbitmq_cpu_usage	CPU 使用率	统计 RabbitMQ 节点虚拟机的 CPU 使用率。	0~100	%	不涉及	RabbitMQ 实例节点	1分钟

指标ID	指标名称	指标含义	取值范围	单位	进制	测量对象（维度）	监控周期（原始指标）
rabbitmq_cpu_load	CPU核均负载	统计RabbitMQ节点虚拟机CPU每个核的平均负载。	>0	不涉及	不涉及	RabbitMQ实例节点	1分钟
rabbitmq_memory_usage	内存使用率	统计RabbitMQ节点虚拟机的内存使用率。 2025年9月前，内存使用率=虚拟机已用内存/虚拟机总内存 2025年9月及以后，内存使用率=RabbitMQ进程占用内存/虚拟机总内存	0~100	%	不涉及	RabbitMQ实例节点	1分钟
rabbitmq_disk_read_wait	磁盘平均读操作耗时	该指标用于统计磁盘在测量周期内平均每个读IO的操作时长。	>0	ms	不涉及	RabbitMQ实例节点	1分钟
rabbitmq_disk_write_wait	磁盘平均写操作耗时	该指标用于统计磁盘在测量周期内平均每个写IO的操作时长。	>0	ms	不涉及	RabbitMQ实例节点	1分钟
rabbitmq_node_bytes_in_rate	网络流入流量	统计RabbitMQ节点每秒网络访问流入流量。	>0	Byte/s	1024(IEC)	RabbitMQ实例节点	1分钟
rabbitmq_node_bytes_out_rate	网络流出流量	统计RabbitMQ节点每秒网络访问流出流量。	>0	Byte/s	1024(IEC)	RabbitMQ实例节点	1分钟

指标ID	指标名称	指标含义	取值范围	单位	进制	测量对象（维度）	监控周期（原始指标）
rabbitmq_node_queues	节点队列数	该指标用于统计RabbitMQ节点队列个数。	>0	Count	不涉及	RabbitMQ实例节点	1分钟
rabbitmq_memory_high_watermark	内存高水位状态	表示RabbitMQ节点是否触发内存高水位，如果触发，会阻塞集群的所有生产者。	1：触发 0：没有触发	不涉及	不涉及	RabbitMQ实例节点	1分钟
rabbitmq_disk_insufficient	磁盘高水位状态	表示RabbitMQ节点是否触发磁盘高水位，如果触发，会阻塞集群的所有生产者。	1：触发 0：没有触发	不涉及	不涉及	RabbitMQ实例节点	1分钟
rabbitmq_disk_read_rate	磁盘读流量	统计节点磁盘每秒的读字节大小。	>= 0	Byte/s	1024(IEC)	RabbitMQ实例节点	1分钟
rabbitmq_disk_write_rate	磁盘写流量	统计节点磁盘每秒的写字节大小。	>= 0	Byte/s	1024(IEC)	RabbitMQ实例节点	1分钟
connections_usage	连接数使用率	当前节点实际连接数占最大连接数比率。	>=0	%	不涉及	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	单位	进制	测量对象（维度）	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。	0~10000000	Count	不涉及	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。	0~10000000	Count	不涉及	RabbitMQ实例队列	1分钟
queue_consumers	消费者数量	该指标用于统计订阅该队列的消费者个数。	>= 0	Count	不涉及	RabbitMQ实例队列	1分钟
queue_messages_publish_rate	生产速率	该指标用于统计每秒该队列的消息流入数。	>= 0	Count/s	不涉及	RabbitMQ实例队列	1分钟
queue_messages_ack_rate	消费速率（手工确认）	该指标用于统计该队列每秒传递给客户端并确认的消息数。	>= 0	Count/s	不涉及	RabbitMQ实例队列	1分钟
queue_messages_deliver_get_rate	消费速率	该指标用于统计该队列每秒的消息流出数。	>= 0	Count/s	不涉及	RabbitMQ实例队列	1分钟
queue_messages_redeliver_rate	重传速率	该指标用于统计该队列每秒的重传消息数。	>= 0	Count/s	不涉及	RabbitMQ实例队列	1分钟

指标ID	指标名称	指标含义	取值范围	单位	进制	测量对象（维度）	监控周期（原始指标）
queue_messages_persistent	消息总数（持久化）	该指标用来统计该队列中持久消息的总数（对于瞬时队列始终为0）。	>= 0	Count	不涉及	RabbitMQ实例队列	1分钟
queue_messages_ram	消息总数（内存）	该指标用于统计该队列中驻留在内存中的消息总数。	>= 0	Count	不涉及	RabbitMQ实例队列	1分钟
queue_memory	Erlang进程消耗字节数	该指标用于统计与队列关联的Erlang进程消耗的内存字节数，包括堆栈、堆和内部结构。	>= 0	Byte	1024(IEC)	RabbitMQ实例队列	1分钟
queue_message_bytes	消息大小总和	该指标用于统计该队列中所有消息的大小总和（字节）。	>= 0	Byte	1024(IEC)	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
内存高水位状态	告警阈值：原始值 ≥1 连续触发次数：1 告警级别：致命	告警阈值为1表示触发内存高水位，会阻塞消息生产	- 加快消费 - 采用生产者确认的发送模式，并监控生产端消息生产速度和时长，当消息生产时长有明显增加时进行流控措施
磁盘高水位状态	告警阈值：原始值 ≥1 连续触发次数：1 告警级别：致命	告警阈值为1表示触发磁盘高水位，会阻塞消息生产	- 减少惰性队列的消息堆积 - 减少持久化队列的消息堆积 - 删除队列
内存使用率	告警阈值：原始值 >业务预期使用率（推荐30%） 连续触发次数：连续3~5个周期 告警级别：重要	该指标需要分别为每个节点设置内存使用率告警，避免触发内存高水位阻塞生产	- 加快消费 - 采用生产者确认的发送模式，并监控生产端消息生产速度和时长，当消息生产时长有明显增加时进行流控措施
CPU使用率	告警阈值：原始值 >业务预期使用率（推荐70%） 连续触发次数：连续3~5个周期 告警级别：重要	该指标需要分别为每个节点设置CPU使用率告警，CPU使用率过高可能会影响生产速度	- 减少镜像队列个数 - 对于集群实例，建议扩容节点个数，然后进行节点间重平衡
可消费消息数	告警阈值：原始值 >业务预期可消费消息数 连续触发次数：1 告警级别：重要	可消费消息数过多表示消息堆积	请参考 消息堆积的解决办法
未确认消息数	告警阈值：原始值 >业务预期未确认消息数 连续触发次数：1 告警级别：重要	未确认消息数过多可能会导致消息堆积	- 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

指标名称	告警策略	指标说明	解决方案
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 监控告警

步骤1 登录管理控制台。

步骤2 在管理控制台左上角单击 ，选择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RabbitMQ实例列表页面。

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 > 监控详情”，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](#)。

- 设置告警名称和告警的描述。
- 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。

- 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	deleteDMSCreateFailureInstancesSuccess
删除创建失败的DMS实例失败	rabbitmq	deleteDMSCreateFailureInstancesFailure
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
回收站恢复实例成功	rabbitmq	out_recycleTaskSuccess
回收站恢复实例失败	rabbitmq	out_recycleTaskFailure

查看审计日志

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

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。

集群版本的RabbitMQ实例支持扩大存储空间和代理个数。

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不支持升级CPU和内存。

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。

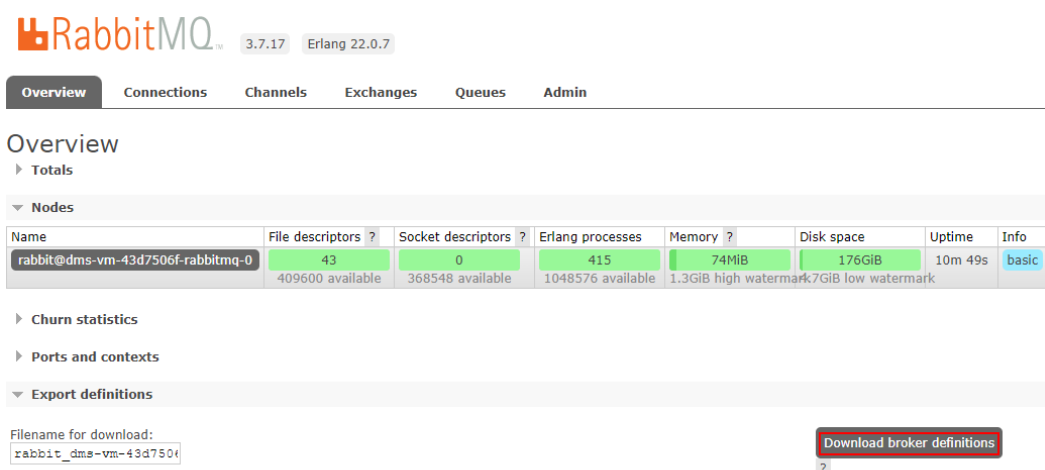
16.1.13 实例是否支持修改可用区？

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。

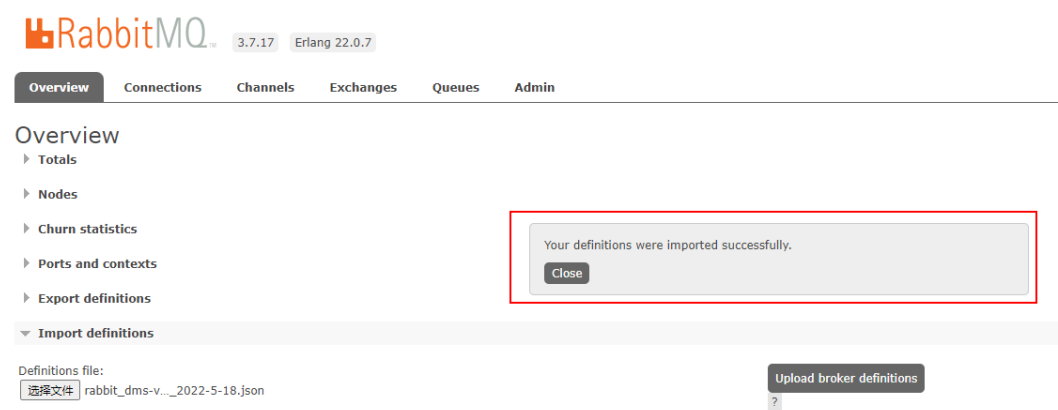


步骤3 登录重新创建的RabbitMQ实例的WebUI页面，在“Overview”页签中，单击“选择文件”，选择**步骤2**中导出的元数据。

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。

16.1.18 RabbitMQ 扩容/缩容会影响业务吗？

扩容/缩容RabbitMQ实例的影响如表16-1所示。

表 16-1 变更实例规格的影响

变更配置类型	影响
代理数量	- 扩容代理数量过程中会重启RabbitMQ的负载均衡进程，导致节点主备倒换，造成秒级业务中断，客户端需要支持自动重连，建议在业务低峰时进行变更。 - 变更时长和新增代理数量有关，单个代理的变更时长一般在5~10分钟，总变更时长为（5~10分钟）*新增代理数量。
存储空间	- 扩容存储空间不会影响业务。 - 扩容存储空间有次数限制，每个实例只能扩容20次。 - 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。 - 预计总变更时长为5分钟内。
代理规格	- RabbitMQ 3.x.x版本单机实例在变更规格过程中，节点会重启，造成分钟级业务中断，客户端需要支持自动重连，建议在业务低峰时进行变更。 - RabbitMQ 3.x.x版本没有配置镜像/仲裁队列的集群实例在变更规格过程中，节点会逐个重启，造成分钟级业务中断，客户端需要支持自动重连，建议在业务低峰时进行变更。 - RabbitMQ 3.x.x版本配置了镜像/仲裁队列的集群实例在变更规格过程中，节点会逐个重启，造成秒级业务中断，客户端需要支持自动重连，建议在业务低峰时进行变更。 - RabbitMQ 3.x.x版本集群实例在变更规格过程中，变更中的节点连接会切换至其余节点，导致其余节点存在过载风险（比如超过连接数、内存高水位等），建议您在实例规格范围内使用，具体实例规格请参考产品规格。 Exchange、Queue和消息需要配置持久化，否则节点重启后，消息可能丢失。设置持久化的方法，请参考配置RabbitMQ持久化。 - 代理采用滚动重启的方式进行实例变更，变更时长和代理数量有关，单个代理的变更时长一般在5~10分钟，总变更时长为（5~10分钟）*代理数量。

以下示例演示在Java客户端设置消息重试。

```
ConnectionFactory connectionFactory = new ConnectionFactory();
//设置服务地址
connectionFactory.setHost("localhost");
//设定端口
connectionFactory.setPort(5672);
//自动重试:
```

```
connectionFactory.setAutomaticRecoveryEnabled(true);
connectionFactory.setNetworkRecoveryInterval(5);
connectionFactory.setTopologyRecoveryEnabled(true);
```

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。
 - a. 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。
 - b. 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。

表 16-2 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。

表 16-3 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。

表 16-4 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

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“概览”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：

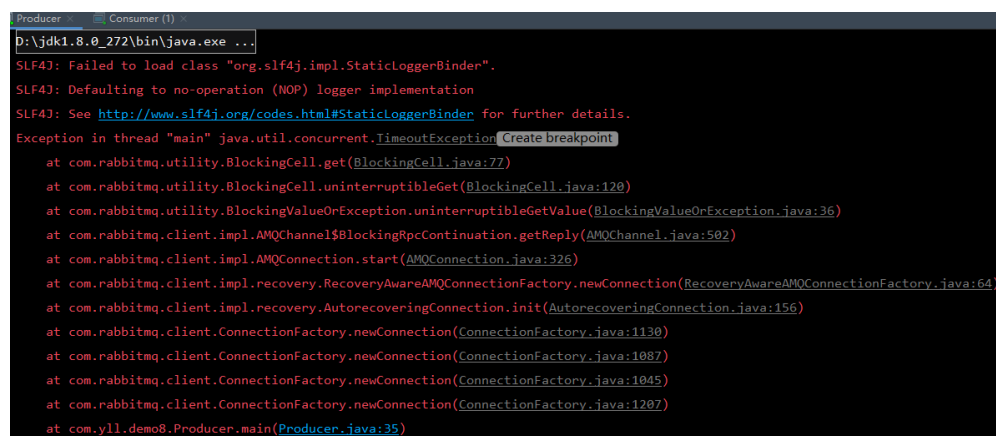
```
[root@ecs-test RabbitMQ-Tutorial]# java -cp ./rabbitmq-tutorial.jar Send 192.168.125.111 5672 user *****
Exception in thread "main" com.rabbitmq.client.AuthenticationFailureException: ACCESS_REFUSED - Login
was refused using authentication mechanism PLAIN. For details
see the broker logfile.
at com.rabbitmq.client.impl.AMQConnection.start(AMQConnection.java: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

```
at
com.rabbitmq.client.impl.recovery.RecoveryAwareAMQConnectionFactory.newConnection(RecoveryAwareAMQConnectionFactory.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“实例管理 > 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“实例管理 > 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RabbitMQ实例控制台的“概览 > 网络”页面中，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，需要允许端口15672的访问。
 - 实例已开启SSL开关，需要允许端口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。

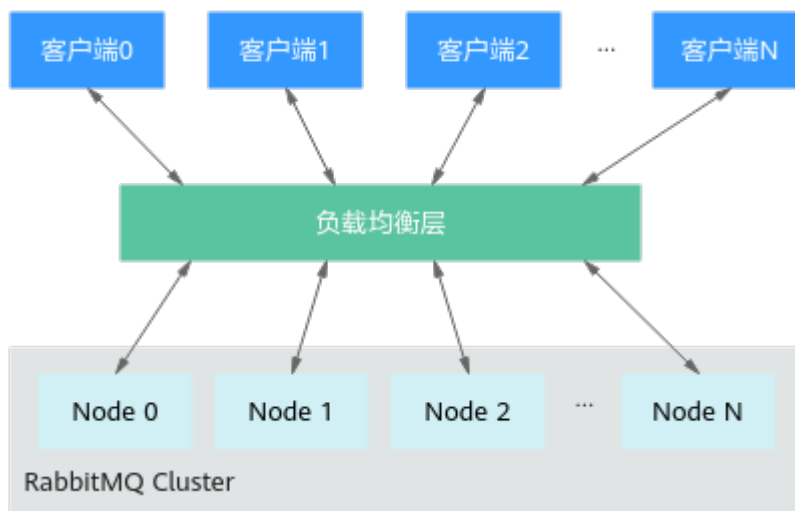
Vhost（Virtual Hosts）是RabbitMQ的基本特性，每个Vhost相当于一个相对独立的RabbitMQ服务器，每个Vhost数据目录不同，共用一个进程。性能上，连接多个Vhost和单独使用一个Vhost差别不大，只是RabbitMQ进程多一些对象，建议使用业务模型实测。

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 16-1 连接示意图



16.2.12 RabbitMQ 实例支持 ping 吗？

RabbitMQ部分实例支持ping命令，部分不支持，具体如下：

- 单机实例：内网连接地址和公网连接地址都支持ping命令。
- 集群实例：内网连接地址支持ping命令，公网连接地址不支持ping命令。

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费时间长**：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。

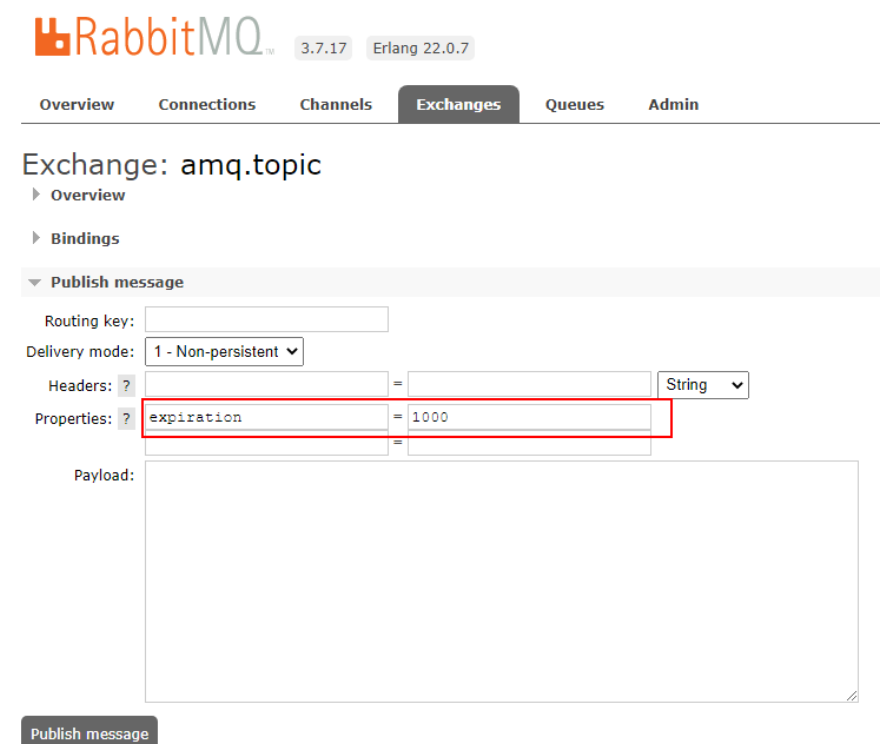
- 在properties中设置expiration值，单位为毫秒(ms)。

```
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。



The screenshot shows the RabbitMQ Web UI for the 'Exchange: amq.topic'. The 'Publish message' section is expanded, and the 'expiration' property is set to '1000' in the 'Properties' field. The 'Routing key' is empty, 'Delivery mode' is '1 - Non-persistent', and 'Headers' are empty. The 'Payload' field is a large text area. The 'Publish message' button is at the bottom.

- 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。



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 监报告警问题

16.4.1 云监控无法展示 RabbitMQ 监控数据

RabbitMQ 3.x.x 版本实例

RabbitMQ 3.x.x版本实例的队列名称或Vhost名称属于以下任一情况时，可能会导致监控数据无法展示。

- RabbitMQ的队列名称以特殊字符开头，例如点号“.”。建议删除带特殊字符的队列。
- RabbitMQ的Vhost名称以特殊字符开头，例如点号“.”。建议删除带特殊字符的Vhost。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名为“Vhost-13142708__test01”。
- 在云监控中特殊字符“.”按5个字符计算长度。如果队列名称或Vhost名称包含“.”，且经此规则计算后的总长度超过256个字符，会导致监控数据无法展示。建议删除带“.”的队列或Vhost。

RabbitMQ 3.x.x版本实例的队列名称或Vhost名称包含特殊字符，例如“%”、“|”、“/”，会导致队列/Vhost名称与监控页面不一致，在监控页面中特殊字符会以“_”显示。例如：队列Queue.1%1|2_3/在默认Vhost中，此时监控中显示的队列名为“Queue.1_1_2_3_”。

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
2025-08-29	本次变更如下： - 新增一键购买相同配置实例功能，主要涉及购买RabbitMQ实例章节。 - 控制台导航栏优化，主要涉及创建RabbitMQ Vhost章节。
2025-04-30	本次变更如下： - 新增回收站功能，主要涉及配置RabbitMQ实例回收站策略章节。 - 购买页面改版，主要涉及购买RabbitMQ实例章节。 - 下线重启RabbitMQ功能。
2024-06-26	本次变更如下： 新增标签策略功能，修改购买RabbitMQ实例和配置RabbitMQ实例标签章节。
2023-05-09	本次变更如下： - 新增快速入门。 - 购买实例时，使用新版实例规格，并且新增3.8.35版本，主要涉及购买RabbitMQ实例和产品规格章节。 - 实例详情页改版，主要涉及重启RabbitMQ实例、删除RabbitMQ实例和重置RabbitMQ实例密码章节。 - 新增配置Vhost和配置高级特性章节。
2022-08-12	第一次正式发布。