

阿里云

消息队列RabbitMQ版
产品简介

文档版本：20220712

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
 危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 危险 重置操作将丢失用户配置数据。
 警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告 重启操作将导致业务中断，恢复业务时间约十分钟。
 注意	用于警示信息、补充说明等，是用户必须了解的内容。	 注意 权重设置为0，该服务器不会再接受新请求。
 说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击设置> 网络> 设置网络类型。
粗体	表示按键、菜单、页面名称等UI元素。	在结果确认页面，单击确定。
Courier字体	命令或代码。	执行 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	<code>bae log list --instanceid</code> <code>Instance_ID</code>
[] 或者 [a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ } 或者 {a b}	表示必选项，至多选择一个。	<code>switch {active stand}</code>

目录

1.什么是消息队列RabbitMQ版？	05
2.名词解释	07
3.基本概念	10
3.1. Connection和Channel	10
3.2. Exchange	11
4.产品优势	18
5.产品功能	20
6.应用场景	21
7.实例类型	22
8.使用限制	25
9.开服地域	28
10.开源对比	30

1.什么是消息队列RabbitMQ版？

消息队列RabbitMQ版是一款基于高可用分布式存储架构实现的AMQP 0-9-1协议的消息产品。消息队列RabbitMQ版兼容开源RabbitMQ客户端，解决开源各种稳定性痛点（例如消息堆积、脑裂等问题），同时具备高并发、分布式、灵活扩缩容等云消息服务优势。

核心概念

消息队列RabbitMQ版的核心概念如下：

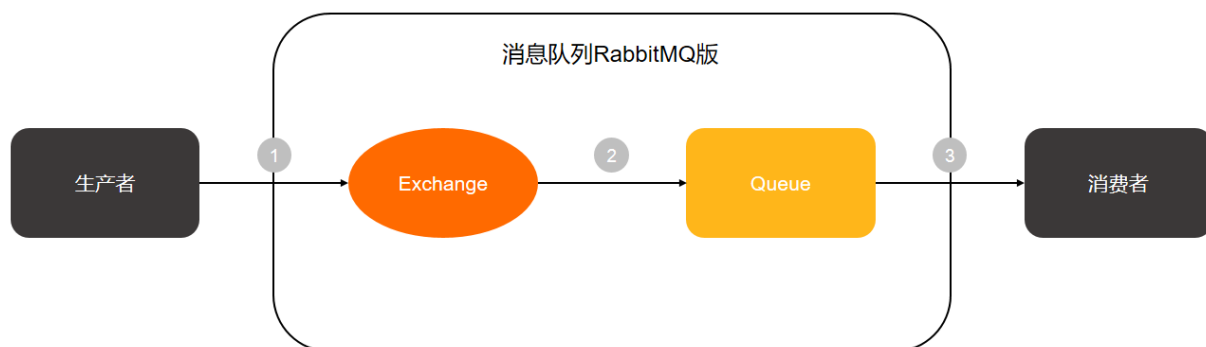
- 生产者：发送消息的应用。
- 消费者：接收消息的应用。
- Exchange：将消息路由到Queue的组件。
- Queue：存储消息的缓冲区。

更多消息队列RabbitMQ版相关概念，请参见[名词解释](#)。

消息队列RabbitMQ版消息流

消息在消息队列RabbitMQ版的组件之间流动的过程如下：

1. 生产者向Exchange发送消息。
2. Exchange根据消息属性将消息路由到Queue进行存储。
3. 消费者从Queue拉取消息进行消费。



开源对比

从性能、稳定性、功能三方面对比消息队列RabbitMQ版与开源RabbitMQ，消息队列RabbitMQ版更具优势。具体信息，请参见[开源对比](#)。

产品优势

相较于开源RabbitMQ，消息队列RabbitMQ版具备灵活易用、功能增强、超强性能、安全可靠的优势。更多信息，请参见[产品优势](#)。

产品功能

消息队列RabbitMQ版兼容开源RabbitMQ，提供完善的运维配套，支持多种消息类型。更多信息，请参见[产品功能](#)。

应用场景

消息队列RabbitMQ版广泛用于金融、保险、政企、电商、新零售、物流、视频互动、能源等行业的业务消息通讯。消息队列RabbitMQ版通常用于业务的异步解耦、削峰填谷和分布式缓存同步。更多信息，请参见[应用场景](#)。

使用限制

消息队列RabbitMQ版对集群、接口调用以及字符进行了限制，您在消息队列RabbitMQ版时注意不要超过相应的限制值，以免程序出现异常。更多信息，请参见[使用限制](#)。

2. 名词解释

本文主要对消息队列RabbitMQ版涉及的专有名词及术语进行定义和解析，方便您更好地理解相关概念并使用消息队列RabbitMQ版。

A

Alternate Exchange

备份Exchange，简称AE，用于接收配置了备份Exchange的Exchange路由失败的消息。

Arguments

Queue的参数，可用于设置死信Exchange、消息过期时间、死信Routing key等。

Auto Delete

自动删除属性，对于设置了自动删除的Exchange，如果绑定到该Exchange的最后一个Queue解除绑定，那么该Exchange将会自动删除。对于设置了自动删除的Queue，如果订阅该Queue的最后一个消费端取消订阅后，那么该Queue将会自动删除。

B

Binding

一套绑定规则，用于告诉Exchange消息应该被存储到哪个Queue。它的作用是把Exchange和Queue按照路由规则绑定起来。

Binding Key

用于告知Exchange应该将消息投递到哪些Queue中。

C

Channel

在客户端的每个物理TCP连接里，可建立多个Channel，每个Channel代表一个会话任务。具体信息，请参见[Connection](#)和[Channel](#)。

Connection

TCP连接，生产者或消费者与消息队列RabbitMQ版的物理TCP连接。具体信息，请参见[Connection](#)和[Channel](#)。

E

Exchange

生产者将消息发送到Exchange，由Exchange将消息路由到一个或多个Queue中。Exchange根据消息的属性或内容路由消息。具体信息，请参见[Exchange](#)。

I

Internal

内建类型，该类型的Exchange用于Exchange之间的绑定。

M

Message ID

Message ID（消息标识符）是消息的可选属性，类型为short string。Message ID在业务上通常被设置为唯一，适用于追踪和识别销售单、工单等需要保证消息唯一的场景。消息队列RabbitMQ版服务端不会对消息进行幂等处理。如需实现消息幂等，即如果消息重试多次，消费端对该重复消息消费多次与消费一次的结果是相同的，并且多次消费没有对系统产生副作用，在为每条消息设置唯一Message ID的基础上，您还需要在消息队列RabbitMQ版的Consumer客户端对消息进行幂等处理，具体信息，请参见[消息幂等](#)。

Q

Queue

消息队列，每个消息都会被投入到一个或多个Queue里。

R

Routing Key

生产者在向Exchange发送消息时，需要指定一个Routing Key来设定该消息的路由规则。Routing Key需要与Exchange类型及Binding Key联合使用才能生效。一般情况下，生产者在向Exchange发送消息时，可以通过指定Routing Key来决定消息被路由到哪个或哪些Queue。

S

实例

一个独立的消息队列RabbitMQ版资源实体，包含Vhost、Exchange、Queue等基本的资源要素。

实例限流

当消息队列RabbitMQ版实例的TPS流量峰值超过您所购买实例的TPS规格上限时，消息队列RabbitMQ版实例会被限流。

限流后的行为如下：

具体信息，请参见[实例限流最佳实践](#)。

生产者

消息生产者，即投递消息的程序。

死信Exchange

用于路由死信消息的Exchange。死信Exchange会根据Binding Key、死信Routing Key、Header属性将死信消息投递至死信Queue。死信Exchange可以是任何一种常见类型的Exchange，例如Direct Exchange，具体信息，请参见[死信Exchange](#)。

死信Routing Key

死信消息的路由规则。如果不设置死信消息的Routing Key，则死信消息的Routing Key默认为消息本身的Routing Key。

死信消息

被重新发送到死信Exchange的消息。消息变成死信消息的可能原因如下：

- `requeue`参数被设置为 `false`，消费者使用 `basic.reject` 或 `basic.nack` 否定应答（NACK）消息。
- 消息重试次数超过16次，消息重试失败。具体信息，请参见[消息重试](#)。



注意 过期消息或超过队列最大长度被丢弃的消息不会成为死信消息。

死信Queue

死信Exchange绑定的Queue，用于存储死信消息。

V**Vhost**

虚拟主机（Virtual Host），用作逻辑隔离，分别管理各自的Exchange、Queue和Binding，使得应用安全的运行在不同的Vhost实例上，相互之间不会干扰。一个实例下可以有多个Vhost，一个Vhost里面可以有若干个Exchange和Queue。生产者和消费者连接消息队列RabbitMQ版需要指定一个Vhost。

X**消费者**

消息消费者，即接收消息的程序。

消息存活时间

消息在Queue中的有效期。某条消息在Queue中的留存时间超过配置的消息存活时间时，则该消息过期。消息存活时间的值必须为非负整型数，单位为毫秒。例如，某条消息的存活时间的值是1000，则代表该消息最多会在Queue中存活1秒。具体信息，请参见[消息存活时间](#)。

Y**延时消息**

生产者将消息发送到消息队列RabbitMQ版服务端，但并不期望这条消息立马投递，而是延迟一定时间后才投递到消费者进行消费，该消息即延时消息。具体信息，请参见[延时消息](#)。

永久性

在服务器重启之后Queue、Exchange以及相应Binding仍然存在的现象。

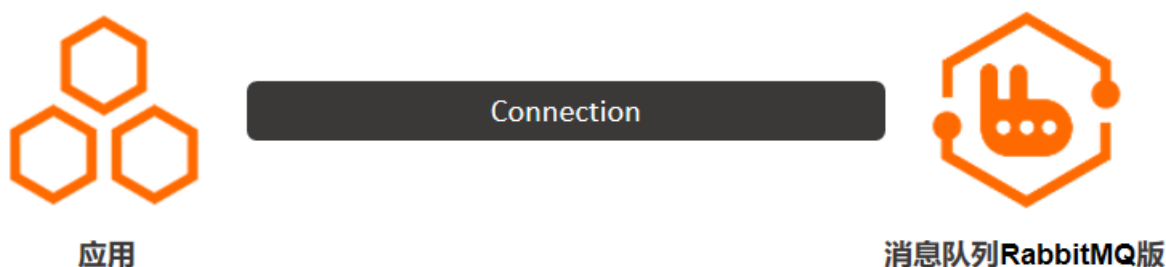
3. 基本概念

3.1. Connection和Channel

本文介绍消息队列RabbitMQ版中Connection和Channel的概念及其使用建议。

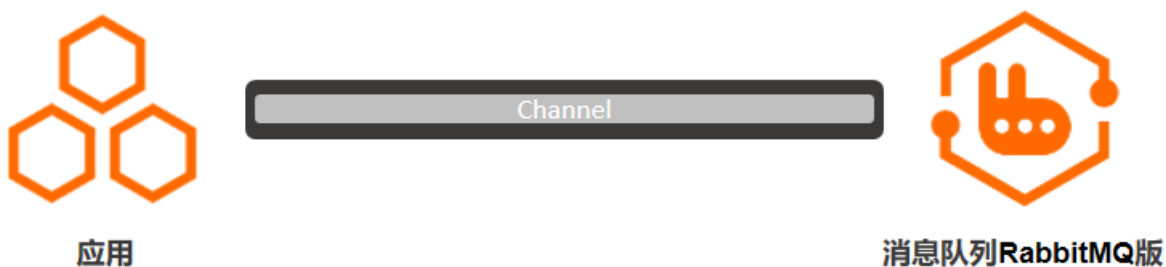
Connection

Connection是物理TCP连接。Connection将应用与消息队列RabbitMQ版连接在一起。Connection会执行认证、IP解析、路由等底层网络任务。应用与消息队列RabbitMQ版完成Connection建立大约需要15个TCP报文交互，因而会消耗大量的网络资源和消息队列RabbitMQ版资源。大量的Connection会对消息队列RabbitMQ版造成巨大压力，甚至触发消息队列RabbitMQ版SYN洪水攻击防护，导致消息队列RabbitMQ版无响应，进而影响您的业务。



Channel

Channel是物理TCP连接中的虚拟连接。当应用通过Connection与消息队列RabbitMQ版建立连接后，所有的AMQP协议操作（例如创建队列、发送消息、接收消息等）都会通过Connection中的Channel完成。Channel可以复用Connection，即一个Connection下可以建立多个Channel。Channel不能脱离Connection独立存在，而必须存活在Connection中。当某个Connection断开时，该Connection下的所有Channel都会断开。当大量应用需要与消息队列RabbitMQ版建立多个连接时，建议您使用Channel来复用Connection，从而减少网络资源和消息队列RabbitMQ版资源消耗。



使用建议

Connection和Channel的使用建议如下：

- 保持Connection长连接，请勿频繁开启或关闭Connection。如果确实需要频繁开启或关闭连接，请使用Channel。单实例开启Connection或Channel的接口限制，请参见[使用限制](#)。
- 一个进程对应一个Connection，一个进程中的多个线程则分别对应一个Connection中的多个Channel。
- Producer和Consumer分别使用不同的Connection进行消息发送和消费。

3.2. Exchange

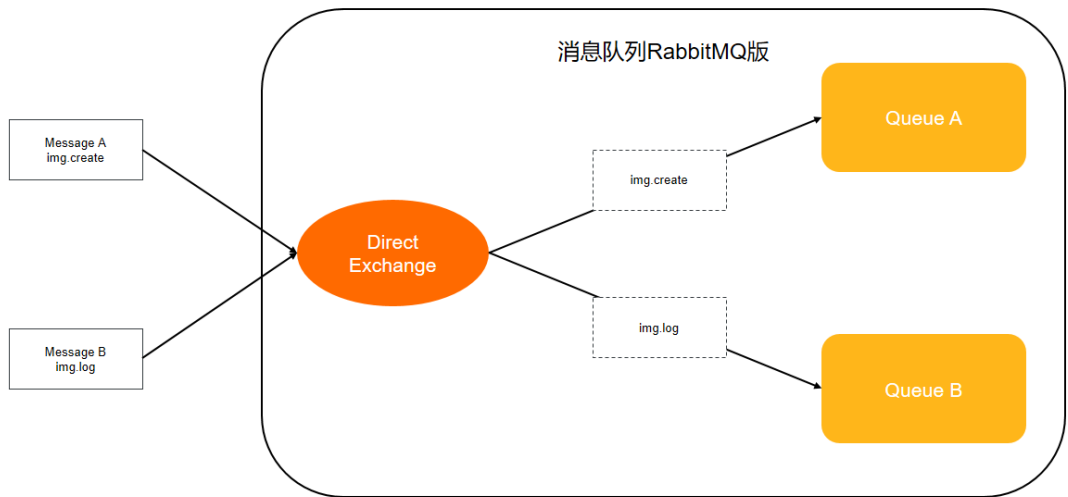
本文介绍消息队列Rabbit MQ版的Exchange的概念和类型，帮助您理解和选择Exchange。

什么是Exchange

Exchange是消息队列Rabbit MQ版的消息路由代理。生产者向消息队列Rabbit MQ版发送消息时，不会直接将消息发送到Queue，而是先将消息发送到Exchange，由Exchange将消息路由到一个或多个Queue。
Exchange根据Binding Key、Routing Key以及Headers属性路由消息。

Direct Exchange

- 路由规则
Direct Exchange根据Binding Key和Routing Key完全匹配的规则路由消息。
- 使用场景
Direct Exchange适用于通过简单字符标识符区分消息的场景。Direct Exchange常用于单播路由。
- 匹配示例
Direct Exchange根据Binding Key和Routing Key完全匹配的规则路由消息的示例如下：



Message	Routing Key	Binding Key	Queue
Message A	img.create	img.create	Queue A
Message B	img.log	img.log	Queue B

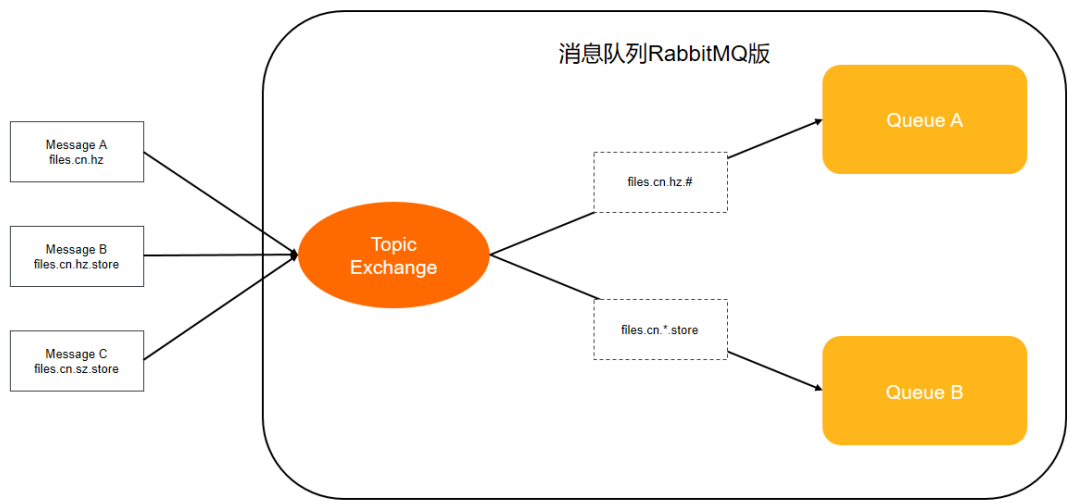
Topic Exchange

- 路由规则
Topic Exchange根据Binding Key和Routing Key通配符匹配的规则路由消息。Topic Exchange支持的通配符包括星号 (*) 和井号 (#)。星号 (*) 代表一个英文单词 (例如cn)。井号 (#) 代表零个、一个或多个英文单词，英文单词间通过英文句号 (.) 分隔，例如cn.zj.hz。
- 使用场景

Topic Exchange适用于通过通配符区分消息的场景。Topic Exchange常用于多播路由。例如，使用Topic Exchange分发有关于特定地理位置的数据。

● 路由示例

Topic Exchange根据Binding Key和Routing Key通配符匹配的规则路由消息的示例如下：



Message	Routing Key	Binding Key	Queue
Message A	files.cn.hz	files.cn.hz.#	Queue A
Message B	files.cn.hz.store	- files.cn.hz.# - files.cn.*.store	- Queue A - Queue B
Message C	files.cn.sz.store	files.cn.*.store	Queue B

Fanout Exchange

● 路由规则

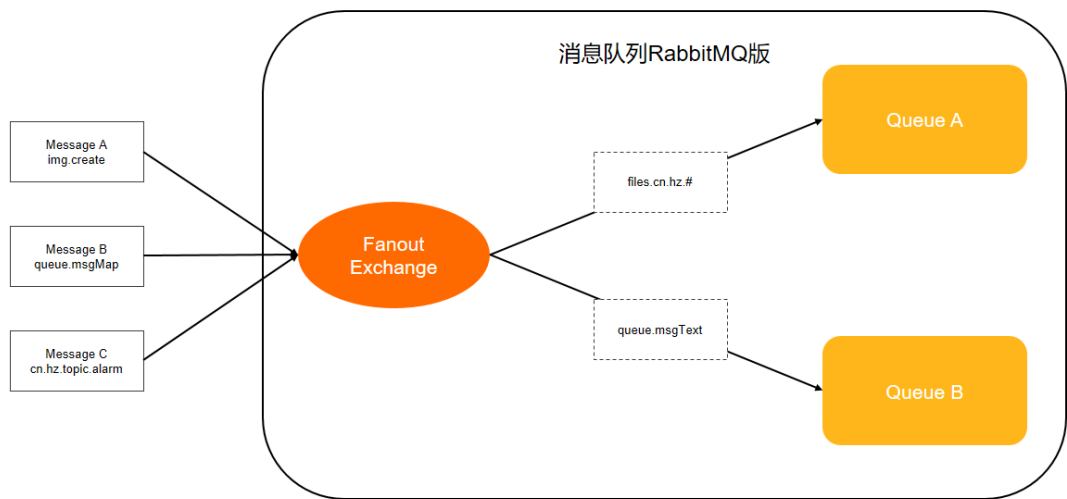
Fanout Exchange忽略Routing Key和Binding Key的匹配规则将消息路由到所有绑定的Queue。

● 使用场景

Fanout Exchange适用于广播消息的场景。例如，分发系统使用Fanout Exchange来广播各种状态和配置更新。

● 路由示例

Fanout Exchange忽略Routing Key和Binding Key的匹配规则将消息路由到所有绑定的Queue的示例如下：



Message	Routing Key	Binding Key	Queue
Message A	img.create	- files.cn.hz.# - queue.msgText	- Queue A - Queue B
Message B	queue.msgMap	- files.cn.hz.# - queue.msgText	- Queue A - Queue B
Message C	cn.hz.topic.alarm	- files.cn.hz.# - queue.msgText	- Queue A - Queue B

Headers Exchange

● 路由规则

Headers Exchange可以被视为Direct Exchange的另一种表现形式。Headers Exchange可以像Direct Exchange一样工作，不同之处在于Headers Exchange使用Headers属性代替Routing Key进行路由匹配。您在绑定Headers Exchange和Queue时，可以设置绑定属性的键值对。然后，在向Headers Exchange发送消息时，设置消息的Headers属性键值对。Headers Exchange将根据消息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，只有以下两种取值：

- all: 所有除x-match以外的绑定属性键值对必须和消息Headers属性键值对匹配才会路由消息。
- any: 只要有一组除x-match以外的绑定属性键值对和消息Headers属性键值对匹配就会路由消息。

以下两种情况下，认为消息Headers属性键值对和绑定属性键值对匹配：

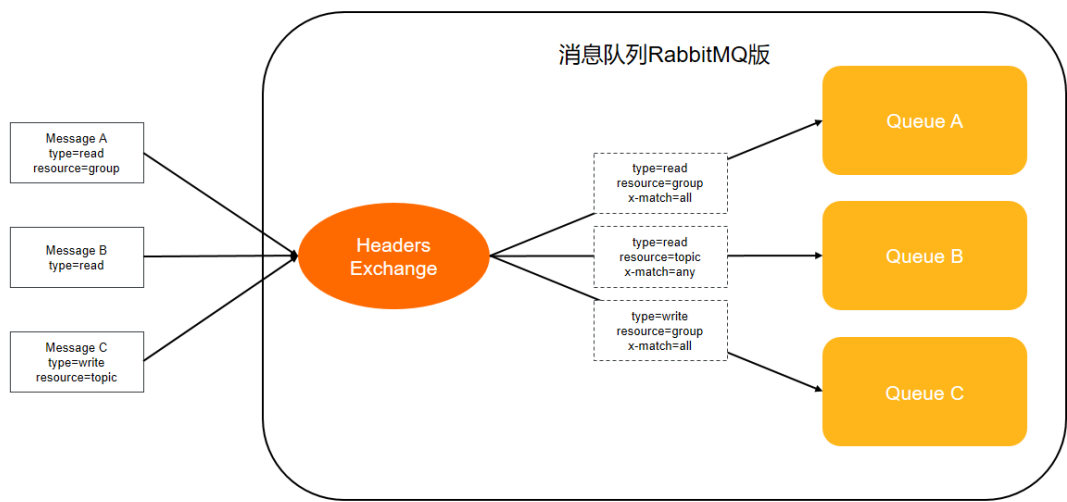
- 消息Headers属性的键和值与绑定属性的键和值完全相同。
- 消息Headers属性的键和绑定属性的键完全相同，但绑定属性的值为空。

● 使用场景

Headers Exchange适用于通过多组Headers属性区分消息的场景。Headers Exchange常用于多播路由。例如，涉及到分类或者标签的新闻更新。

● 使用示例

Headers Exchange根据消息Headers属性和Binding Headers属性的匹配规则路由消息的示例如下：



Message	消息Headers属性	Binding Headers属性	Queue
Message A	<code>type=read resource=group</code>	<code>type=read resource=group x-match=all</code> <code>type=read resource=topic x-match=any</code>	◦ Queue A ◦ Queue B
Message B	<code>type=read</code>	<code>type=read resource=topic x-match=any</code>	Queue B
Message C	<code>type=write resource=topic</code>	<code>type=read resource=topic x-match=any</code>	Queue B

JMS Queue Exchange

除AMQP协议约定的Exchange外，消息队列RabbitMQ版还支持JMS Queue Exchange。JMS Queue Exchange本质上就是Direct Exchange。消息队列RabbitMQ版通过JMS Queue Exchange来实现JMS规范约定的Queue。

❓ 说明 该类Exchange只能通过SDK创建。

● 路由规则

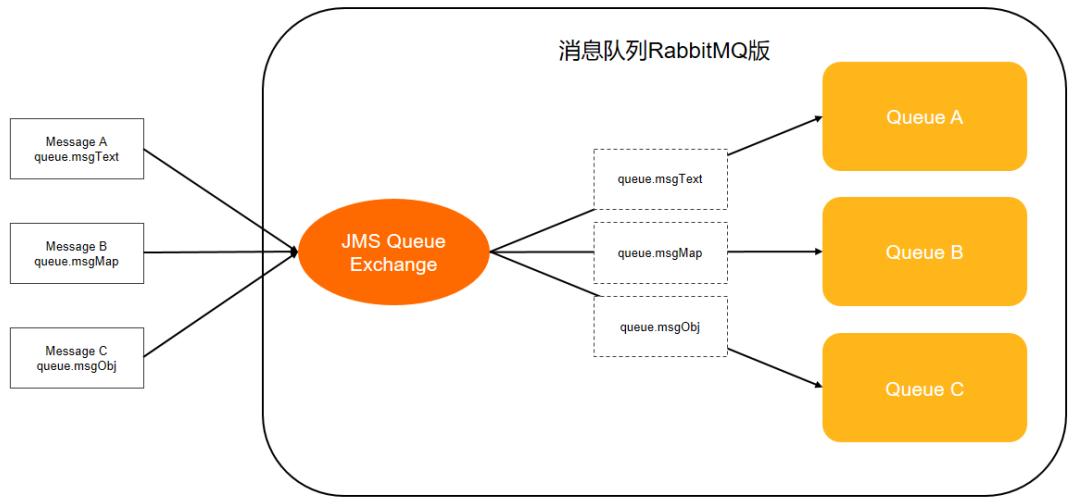
JMS Queue Exchange根据Binding Key和Routing Key完全匹配的规则路由消息。

● 使用场景

JMS Queue Exchange适用于通过简单字符标识符区分消息的场景。JMS Queue Exchange常用于单播路由。

● 使用示例

JMS Queue Exchange根据Binding Key和Routing Key完全匹配的规则路由消息的示例如下：



Message	Routing Key	Binding Key	Queue
Message A	queue.msgText	queue.msgText	Queue A
Message B	queue.msgMap	queue.msgMap	Queue B
Message C	queue.msgObj	queue.msgObj	Queue C

JMS Topic Exchange

除AMQP协议约定的Exchange外，消息队列RabbitMQ版还支持JMS Topic Exchange。JMS Topic Exchange本质上就是Topic Exchange。消息队列RabbitMQ版通过JMS Topic Exchange来实现JMS规范约定的Topic。

● 路由规则

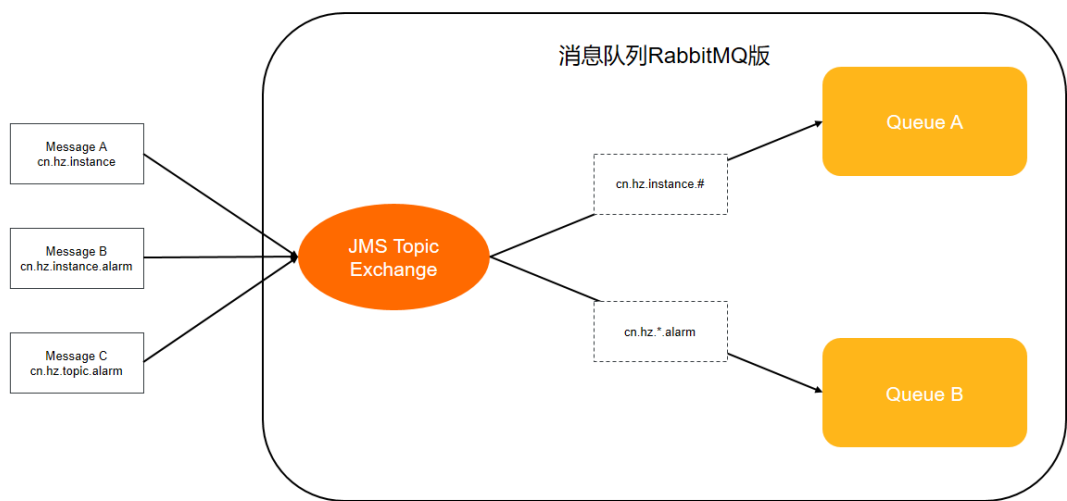
JMS Topic Exchange根据Binding Key和Routing Key通配符匹配的规则路由消息。JMS Topic Exchange支持星号（*）和井号（#）。星号（*）代表一个英文单词（例如cn）。井号（#）代表零个、一个或多个英文单词，英文单词间通过英文句号（.）分隔（例如cn.zj.hz）。

● 使用场景

JMS Topic Exchange适用于通过通配符区分消息的场景。JMS Topic Exchange常用于多播路由。例如，使用JMS Topic Exchange分发有关于特定地理位置的数据。

● 使用示例

JMS Topic Exchange根据Binding Key和Routing Key通配符匹配的规则路由消息的示例如下：



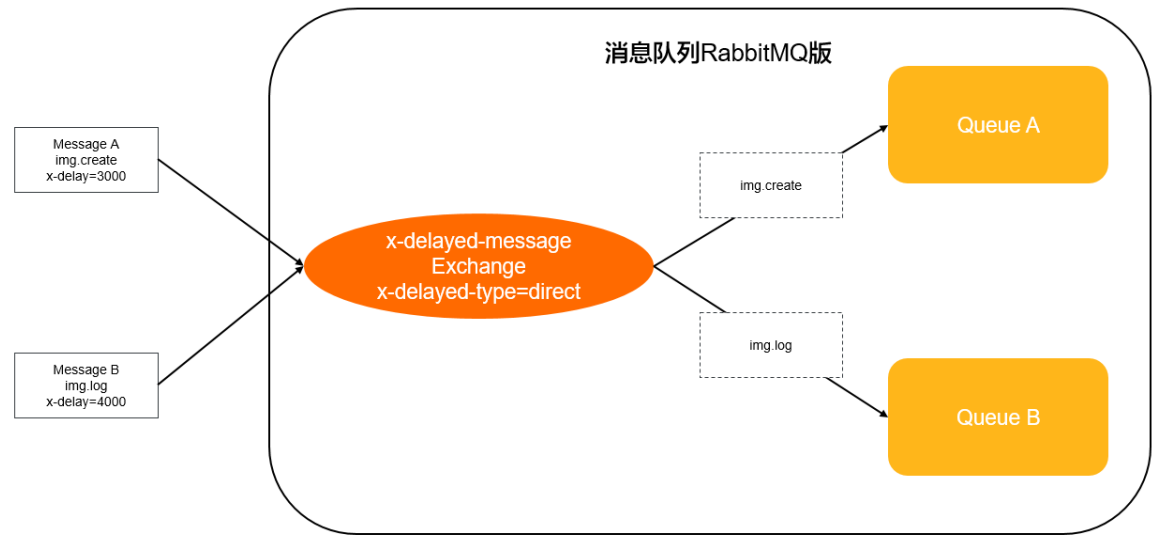
Message	Binding Key	Routing Key	Queue
Message A	cn.hz.instance	cn.hz.instance.#	Queue A
Message B	cn.hz.instance.alarm	- cn.hz.instance.# - cn.hz.*.alarm	- Queue A - Queue B
Message C	cn.hz.topic.alarm	cn.hz.*.alarm	Queue B

x-delayed-message Exchange

消息队列RabbitMQ版还兼容开源RabbitMQ以插件形式支持的x-delayed-message Exchange。您无需安装该插件，只需声明该类Exchange，并自定义消息的Header属性x-delay来指定消息延时投递的时间段，单位为毫秒。消息将在x-delay定义的时间段后被投递到对应的Queue。

- 路由规则
 - x-delayed-message Exchange根据扩展字段x-delayed-type指定的Exchange类型确定路由规则。支持x-delayed-message的Exchange类型如下：
 - Direct Exchange
 - Topic Exchange
 - Fanout Exchange
 - Headers Exchange
 - JMS Queue Exchange
 - JMS Topic Exchange
- 使用场景
 - x-delayed-message Exchange适用于需要延时投递消息的场景。更多信息，请参见[延时消息](#)。
- 使用示例

x-delayed-message Exchange根据x-delayed-type指定的Exchange类型的路由规则路由消息。以x-delayed-type指定为Direct类型为例，Direct Exchange根据Binding Key和Routing Key完全匹配的规则路由消息的示例如下：



Message	x-delay	Routing Key	Binding Key	Queue
Message A	3000	img.create	img.create	Queue A
Message B	4000	img.log	img.log	Queue B

消息路由说明如下：

- Message A到达Exchange后，x-delayed-message Exchange会在3000毫秒后将Message A投递至Queue A。
- Message B到达Exchange后，x-delayed-message Exchange会在4000毫秒后将Message B投递至Queue B。

4. 产品优势

本文主要介绍消息队列Rabbit MQ版相比于自建开源Rabbit MQ所具备的优势，以便您更进一步了解消息队列Rabbit MQ版。

灵活易用

- 开箱即用

完全兼容AMQP 0-9-1标准协议，完全兼容Rabbit MQ开源社区，快速迁移上云。

- 全托管服务

一键开通消息队列Rabbit MQ版服务，即享有免部署免运维的云消息服务。依托于阿里云专业的自动化运维团队，核心链路秒级、分钟级进行多维度巡检，全面的健康诊断报告，针对不健康的状态进行告警与及时恢复，保证服务的可用性与业务的连续性。

高可用

- 服务地域化、多可用区以及分布式集群化部署，确保服务高可用，即便整个机房不可用仍可正常提供消息服务。
- 集群分布式部署，服务节点无状态、无单点。即使单节点不可用，请求将Failover至其他节点保障消息服务正常提供。

功能增强

- 延时消息

- 支持延时消息，实现秒级精准，无先入先出限制。
- 简单易用，在代码上只需要设置一个参数即可完成，解决开源Rabbit MQ无延时队列的痛点。
- 高精度支持，最短可精确到秒级，最长可持续1天。
- 高性能，性能与普通消息一致。
- 多副本数据冗余，确保消息可恢复。

- 消息重试

- 重试一次一分钟，最多16次重试。
- 超过最大重试次数后的消息进入死信队列，死信消息可查询、可导出。

- 死信Exchange

增强型死信Exchange，不仅支持被否定应答的消息，还支持重试失败的消息自动进入死信Exchange，确保消息不丢失。

超强性能

- 平台性能上限

支持百万级队列，横向扩容带来性能的线性增长，性能上无并发限制。

- 单队列扩容

解决Rabbit MQ单队列性能瓶颈，支持单队列的横向扩展，性能上无并发限制。

- 弹性伸缩

可扩展能力强、性能线性增强，可根据业务需求弹性伸缩，对用户透明。

- 海量堆积
 - 在海量消息堆积的情况下，始终保持高性能，不影响集群的正常服务。
 - 消息的生产者与消费者实现隔离，满足生产者大量的并发，消费者稳定消费。

安全可靠

- 权限粒度

实例、Vhost、Queue以及Exchange细粒度权限控制，对每次消息请求进行安全访问控制。
- 主子账号

全面对接阿里云RAM主子账号、黑白名单、STS等功能。
- 加密传输

支持TLS传输加密协议。

5. 产品功能

消息队列RabbitMQ版兼容开源RabbitMQ并提供一套完善的运维工具。

资源支持

消息队列RabbitMQ版兼容开源RabbitMQ的各个组件：

- Queue：支持消息存储，支持集群消费模式。

 **说明** 消息队列RabbitMQ版内部暂未实现消息优先级的处理逻辑，您的客户端通过可选队列参数x-max-priority声明的优先级队列实际是无效的。

- Exchange：支持将消息按照规则路由到一个或多个Queue，同时对不符合路由规则的消息进行丢弃。
- Vhost：支持开源的虚拟主机能力，解决生产环境、多业务拆分等隔离需求。

运维配套

消息队列RabbitMQ版提供一整套完善、便捷、易用的产品运维工具：

- 监报告警：实时监控在线连接、消息堆积、投递延迟，及时发现问题。
- 资源报表：设备查询、分组在线设备管理、消息收发统计等。
- OpenAPI（RESTful）：提供自助资源管理和运维功能，RESTful标准，灵活便捷。

消息类型

消息队列RabbitMQ版支持多种消息类型：

- 普通消息：无特性消息，区别于延时消息。
- 延时消息：在指定时间段之后才可被消费者消费的消息，适用于对消息消费有时间窗口要求的场景。

 **说明** 消息队列RabbitMQ版不支持事务消息，即不支持AMQP 0-9-1协议中的事务操作（包括txSelect、txCommit、txRollback等命令）。为兼容以上事务操作，消息队列RabbitMQ版服务端在接收命令后会直接返回成功状态，但并不具备实际作用。

6. 应用场景

本文为您介绍消息队列Rabbit MQ版的应用场景，以便您更好地判断如何在业务中使用消息队列Rabbit MQ版。

异步解耦

消息队列Rabbit MQ版可用于单体应用被拆解为微服务后不同微服务间的通信。应用解耦的好处是不同应用的迭代不再相互依赖，而异步通信的好处是数据不再需要被立即处理。异步解耦能有效缩短数据链路长度，提高数据处理效率。

削峰填谷

大型活动带来较高流量脉冲时，没有做好相应保护容易导致系统超负荷甚至崩溃，限制太过则会导致请求大量失败而影响用户体验。消息队列Rabbit MQ版能做到削峰填谷。其高性能的消息处理能力可以承接流量脉冲而不被击垮，在确保系统可用性的同时，通过快速有效的请求响应技术提升用户体验。其海量消息堆积能力确保下游业务在安全水位内平滑稳定的运行，避免流量高峰的冲击。

分布式缓存同步

大量并发访问数据库会导致页面响应时间长。通过消息队列Rabbit MQ版构建分布式缓存，支持实时通知数据变化，有效降低页面响应时间，满足对变更的大量访问需求。

7.实例类型

本文介绍消息队列Rabbit MQ版提供的实例类型。

实例类型说明

实例类型说明

项目类别	项目	专业版	企业版	铂金版
高可用和高可靠	部署架构	共享实例（逻辑隔离），集群服务高可用，数据持久化三副本。	共享实例（逻辑隔离），集群服务高可用，数据持久化三副本。	专享实例（独占物理集群），集群服务高可用，数据持久化三副本。
	SLA	99.95%	99.95%	99.99%
功能特性	可观测和监控告警	控制台实时数据观测。 更多信息，请参见 监控指标 。	Prometheus+Grafana支持的Dashboard大盘，精细化指标数据图标展示。 更多信息，请参见 Dashboard 。	Prometheus+Grafana支持的Dashboard大盘，精细化指标数据图标展示。 更多信息，请参见 Dashboard 。
	全球消息路由	不支持。	支持。	支持。
使用限制	集群限制	有限制。 更多信息，请参见 集群限制 。	支持更大的消息等。 更多信息，请参见 集群限制 。	几乎无限制。 更多信息，请参见 集群限制 。
	接口调用限制	有限制。 更多信息，请参见 接口调用限制 。	有限制。 更多信息，请参见 接口调用限制 。	无限制。 更多信息，请参见 接口调用限制 。

实例类型使用限制

集群限制

限制项	专业版	企业版	铂金版
单实例的Vhost数量	单实例的Vhost数量取值范围为[Queue数量 ,200]。	200	4096
单实例的Connection数量	等于Queue数量×4。 单实例的Connection数量取值范围为[1000,5000]。	根据实例规格购买。 单实例至少有2000的免费额度，最大连接数为1万。	根据实例规格购买。 单实例有5万的免费额度，最大连接数为10万。
单Vhost的Exchange数量	等于Queue数量。	等于Queue数量。	无

限制项	专业版	企业版	铂金版
单Vhost的Binding数量	等于Queue数量×10。 单Vhost的Binding数量取值范围为[4096,20000]。	等于Queue数量×10。 单Vhost的Binding数量取值范围为[4096,20000]。	无
单Exchange绑定的Queue数量	256	256	256 如需定制，请 提交工单 。
单Queue的Consumer数量	等于Queue数量×4。 单Queue的Consumer数量取值范围为[50,5000]。	等于Queue数量×4。 单Queue的Consumer数量取值范围为[800,5000]。	无
单Connection的Channel数量	单Connection的Channel数量取值为64或Queue数量。 - 如果Queue数量小于或等于64，则单Connection的Channel数量等于64。 - 如果Queue数量大于64，则单Connection的Channel数量等于Queue数量。	等于Queue数量。 单Connection的Channel数量取值范围为[Queue数量,2048]。	3000
消息大小	64 KB	1 MB ? 说明 当消息小于64 KB时，算一次请求；当消息大于64 KB时，每4 KB算一次请求。	1 MB
延时消息最大延时时间	24小时	3天	7天
消息最大保留时间	3天	3天	3天
消息最大重入Queue次数	16	16	64
消息超时重试间隔	60秒	5分钟	30分钟

接口调用限制

限制项	限制项接口	专业版	企业版	铂金版
单实例发送消息	<code>basicPublish</code>	25000 TPS	25000 TPS	无

限制项	限制项接口	专业版	企业版	铂金版
单实例同步获取消息	<code>basicGet</code>	500 TPS	500 TPS	无
单实例清Queue	<code>purgeQueue</code>	500 TPS	500 TPS	无
单实例创建Exchange	<code>exchangeDeclare</code>	500 TPS	500 TPS	无
单实例删除Exchange	<code>exchangeDelete</code>	500 TPS	500 TPS	无
单实例创建Queue	<code>queueDeclare</code>	500 TPS	500 TPS	无
单实例删除Queue	<code>queueDelete</code>	500 TPS	500 TPS	无
单实例创建Binding	<code>queueBind</code>	500 TPS	500 TPS	无
单实例删除Binding	<code>queueUnbind</code>	500 TPS	500 TPS	无
单实例恢复消息	<code>basicRecover</code>	500 TPS	500 TPS	无
单实例重入Queue消息	<code>basicReject(requeue=true)</code> <code>basicNack(requeue=true)</code>	20 TPS	20 TPS	无

8.使用限制

消息队列RabbitMQ版对集群、接口调用以及字符进行了限制，您在使用消息队列RabbitMQ版时注意不要超过相应的限制值，以免程序出现异常。

集群限制

限制项	专业版	企业版	铂金版
单实例的Vhost数量	单实例的Vhost数量取值范围为[<i>Queue</i> 数量,200]。	200	4096
单实例的Connection数量	等于Queue数量×4。 单实例的Connection数量取值范围为[1000,5000]。	根据实例规格购买。 单实例至少有2000的免费额度，最大连接数为1万。	根据实例规格购买。 单实例有5万的免费额度，最大连接数为10万。
单Vhost的Exchange数量	等于Queue数量。	等于Queue数量。	无
单Vhost的Binding数量	等于Queue数量×10。 单Vhost的Binding数量取值范围为[4096,20000]。	等于Queue数量×10。 单Vhost的Binding数量取值范围为[4096,20000]。	无
单Exchange绑定的Queue数量	256	256	256 如需定制，请 提交工单 。
单Queue的Consumer数量	等于Queue数量×4。 单Queue的Consumer数量取值范围为[50,5000]。	等于Queue数量×4。 单Queue的Consumer数量取值范围为[800,5000]。	无
单Connection的Channel数量	单Connection的Channel数量取值为64或 <i>Queue</i> 数量。 - 如果<i>Queue</i>数量小于或等于64，则单Connection的Channel数量等于64。 - 如果<i>Queue</i>数量大于64，则单Connection的Channel数量等于Queue数量。	等于Queue数量。 单Connection的Channel数量取值范围为[<i>Queue</i> 数量,2048]。	3000

限制项	专业版	企业版	铂金版
消息大小	64 KB	1 MB ❓ 说明 当消息小于64 KB时，算一次请求；当消息大于64 KB时，每4 KB算一次请求。	1 MB
延时消息最大延时时间	24小时	3天	7天
消息最大保留时间	3天	3天	3天
消息最大重入Queue次数	16	16	64
消息超时重试间隔	60秒	5分钟	30分钟

接口调用限制

 注意

- 预付费专业版和企业版：单个接口的调用量受限制，且所有接口的调用总量受限于购买的TPS流量峰值。
- 预付费铂金版：单个接口的调用量无限制，但所有接口的调用总量受限于购买的TPS流量峰值。

限制项	限制项接口	专业版	企业版	铂金版
单实例发送消息	<code>basicPublish</code>	25000 TPS	25000 TPS	无
单实例同步获取消息	<code>basicGet</code>	500 TPS	500 TPS	无
单实例清Queue	<code>purgeQueue</code>	500 TPS	500 TPS	无
单实例创建Exchange	<code>exchangeDeclare</code>	500 TPS	500 TPS	无
单实例删除Exchange	<code>exchangeDelete</code>	500 TPS	500 TPS	无
单实例创建Queue	<code>queueDeclare</code>	500 TPS	500 TPS	无
单实例删除Queue	<code>queueDelete</code>	500 TPS	500 TPS	无
单实例创建Binding	<code>queueBind</code>	500 TPS	500 TPS	无
单实例删除Binding	<code>queueUnbind</code>	500 TPS	500 TPS	无
单实例恢复消息	<code>basicRecover</code>	500 TPS	500 TPS	无

限制项	限制项接口	专业版	企业版	铂金版
单实例重入Queue消息	<code>basicReject(requeue=true)</code> <code>basicNack(requeue=true)</code>	20 TPS	20 TPS	无

字符限制

限制项	限制值
Queue名称	只能包含字母、数字、短划线 (-)、下划线 (_)、半角句号 (.)、井号 (#)、正斜线 (/)、at符号 (@)，长度限制1~255字符。
Exchange名称	只能包含字母、数字、短划线 (-)、下划线 (_)、半角句号 (.)、井号 (#)、正斜线 (/)、at符号 (@)，长度限制1~255字符。
Vhost名称	只能包含字母、数字、短划线 (-)、下划线 (_)、半角句号 (.)、井号 (#)、正斜线 (/)、at符号 (@)，长度限制1~255字符。
Binding Key名称	- 普通类型：只能包含字母、数字、短划线 (-)、下划线 (_)、半角句号 (.)、正斜线 (/)、at符号 (@)，长度限制1~255字符。 - Topic类型：只能包含字母、数字、短划线 (-)、下划线 (_)、星号 (*)、半角句号 (.)、井号 (#)、正斜线 (/)、at符号 (@)，长度限制1~255字符。 不能以半角句号 (.) 开头或结尾。对于井号 (#) 或星号 (*)，如果以其开头，则其后需有半角句号 (.)，如果以其结尾，则其前需有半角句号 (.)，如果既不是开头也不是结尾，则其前后均需有半角句号 (.)。
Consumer Tag名称	只能包含字母、数字、短划线 (-)、下划线 (_)、半角句号 (.)、井号 (#)、正斜线 (/)、at符号 (@)，长度限制1~255字符。

9.开服地域

本文介绍消息队列Rabbit MQ版专业版实例、企业版实例和铂金版实例支持的地域。

- 下表中的✓表示该地域已支持此类型的实例。
- 下表中的表示该地域暂不支持此类型的实例。若您有开通指定地域的需求，请[提交工单](#)反馈给技术支持人员。

公共云

地域	地域ID (RegionId)	专业版	企业版	铂金版
华东1（杭州）	cn-hangzhou	✓	✓	✓
华东2（上海）	cn-shanghai	✓	✓	✓
华北1（青岛）	cn-qingdao	✓		✓
华北2（北京）	cn-beijing	✓	✓	✓
华北3（张家口）	cn-zhangjiakou	✓	✓	✓
华北5（呼和浩特）	cn-huhehaote	✓		✓
华南1（深圳）	cn-shenzhen	✓	✓	✓
华南3（广州）	cn-guangzhou	✓	✓	✓
西南1（成都）	cn-chengdu	✓		✓
中国香港	cn-hongkong	✓	✓	✓
印度尼西亚（雅加达）	ap-southeast-5	✓		✓

地域	地域ID (RegionId)	专业版	企业版	铂金版
日本（东京）	ap-northeast-1	✓		✓
新加坡（新加坡）	ap-southeast-1	✓	✓	✓
澳大利亚（悉尼）	ap-southeast-2	✓		✓
马来西亚（吉隆坡）	ap-southeast-3	✓		✓
德国（法兰克福）	eu-central-1	✓		✓
美国（硅谷）	us-west-1	✓		✓
美国（弗吉尼亚）	us-east-1	✓		✓

金融云

地域名称	地域ID (RegionId)	专业版	企业版	铂金版
华东2金融云	cn-shanghai-finance-1	✓	✓	✓
华北2金融云	cn-beijing-finance-1	✓		✓

10. 开源对比

本文从功能、稳定性、性能、Exchange和Queue等几个方面对比阿里云消息队列Rabbit MQ版与开源Rabbit MQ。

功能

项目	消息队列Rabbit MQ版	开源Rabbit MQ
客户端SDK支持情况	支持开源所有语言和所有版本的SDK。	开源SDK。
定时消息	秒级精确度。兼容插件x-delayed-message用法和TTL两种使用方式。更多信息，请参见 延时消息 。	通过插件或使用消息存活时间过期转移方式实现。
事务消息	不支持。	支持。
消息重试机制	消息消费超过一定时间未响应会重新投递。超时时间和重试次数的详细解释，请参见 消息重试 。	无消息重试机制。无法跳过有问题消息的消费处理，无法处理新生产的消息，从而会导致消息大量堆积引起内存问题，导致消息服务宕机。
可观测能力：Dashboard	- 指标丰富，维度可精确到Vhost、Exchange和Queue，便于您快速发现和定位问题。 - 基于Prometheus+Grafana实现指标的采集和展示，开箱即用，开发成本低。 详细信息，请参见 Dashboard 。	支持以下两种方案： - 方案一：通过Management UI能够获取丰富的指标，但需要自建指标存储及展示的系统。 - 方案二：通过Prometheus+Grafana实现，该方案获取的指标较简单，维度不够精确，不利于快速定位业务问题。
可观测能力：消息轨迹	- 轨迹数据白屏化展示，消息完整的生命周期清晰可见，一目了然。 - 提供强大的索引能力，可根据Queue、消息ID、消息处理耗时等完成不同维度的查询。 详细信息，请参见 消息轨迹 。	消息轨迹信息以文本格式存储在服务器的log文件中，查询和定位问题效率较低。

稳定性

项目	消息队列Rabbit MQ版	开源Rabbit MQ
海量消息堆积能力	海量消息堆积能力，高性能不受消息堆积影响。	抗堆积能力差，容易引发内存问题而导致宕机。
弹性能力	集群分布式无主架构，能够横向快速地扩缩容集群规模。	通过变更机器规格来扩容、缩容。

项目	消息队列Rabbit MQ版	开源Rabbit MQ
服务可用性	99.95%集群分布式高可用架构，多可用区高可用。	Erlang语言，运维靠经验摸索，且无法避免开源架构的稳定性痛点。
数据可靠性	数据三副本，不会影响TPS性能。	配置副本数变多会导致TPS下降。
巡检系统	自动发现并修复死锁、宕机等问题。	无。

性能

项目	消息队列Rabbit MQ版	开源Rabbit MQ
集群TPS上限	无上限。消息队列Rabbit MQ版采用集群分布式部署，无主架构，能够横向扩容、缩容集群规模。	有上限。受机器性能限制，扩容依赖升级机器硬件规格，且有上限。
单个Queue的TPS上限	无上限。消息队列Rabbit MQ版支持单队列的横向扩展，性能上无并发限制，容量无上限。	有上限，单队列性能上限为单节点上限。
连接数	无上限。消息队列Rabbit MQ版实例能承载的连接数可随集群规模扩大而增加，不受连接数增加的影响。	有上限。受机器性能限制，单机的连接数有上限，且无法扩容。
定时消息	秒级精确度、高性能、开箱即用。	使用复杂。

Exchange和Queue

Exchange

项目	消息队列Rabbit MQ版	开源Rabbit MQ
Exchange类型	支持direct、fanout、headers、topic、x-delayed-message五种类型。	支持direct、fanout、headers、topic、x-delayed-message五种类型。
持久化	支持配置持久化与非持久化。	支持配置持久化与非持久化。
Auto Delete	支持。	支持。
Internal	支持。	支持。
Alternate exchange	不支持。	支持。

Queue

项目	消息队列Rabbit MQ版	Rabbit MQ
队列类型	无需配置，分布式高可用集群。	需配置。 • Classic：经典镜像队列。 • Quorum：仲裁队列。

项目	消息队列Rabbit MQ版	Rabbit MQ
节点	无需配置，服务免运维。	需配置，可选择节点。
是否持久化	支持持久化和非持久化。	支持持久化和非持久化。
Max length	无需配置，支持海量消息堆积。	需配置，防止消息堆积过多而引起的内存问题导致宕机。
Max length bytes		
Max in memory length		
Max in memory bytes		
Delivery limit	无需配置，固定值，默认16次。消息重试机制，请参见 消息重试 。	需配置。
Dead letter exchange	支持。	支持。
Dead letter routing key	支持。	支持。
Single active consumer	不支持。	支持。