

# 消息队列 MQTT 版

## 数据集成

### 产品文档



**【版权声明】**

©2013–2026 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

**【商标声明】**

及其他腾讯云服务相关的商标均为腾讯集团下的相关公司主体所有。另外，本文档涉及的第三方主体的商标，依法由权利人所有。

**【服务声明】**

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

# 文档目录

## 数据集成

集成数据到云函数 SCF

集成数据到 CKafka

集成数据到 RocketMQ

# 数据集成

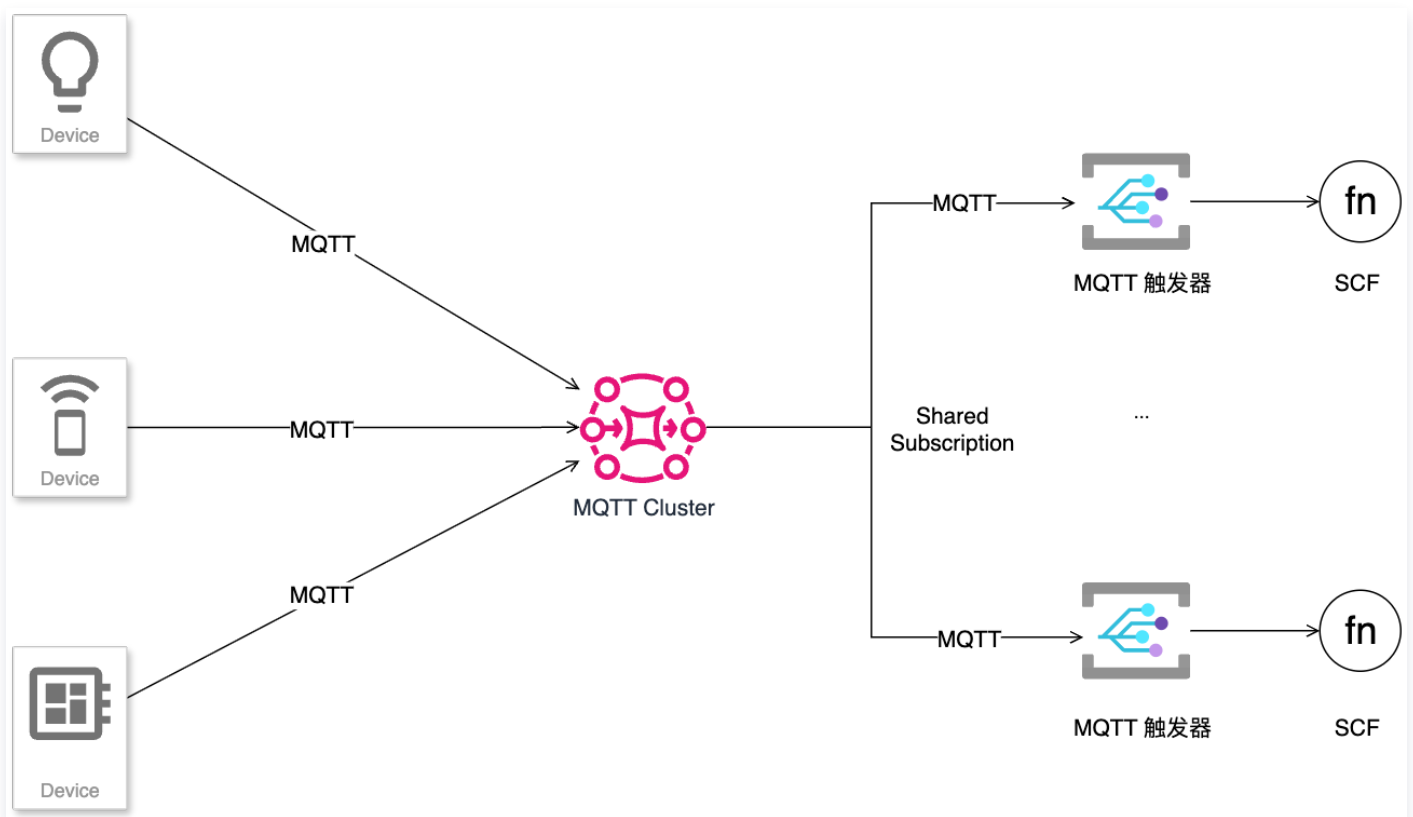
## 集成数据到云函数 SCF

最近更新时间：2026-01-30 15:10:03

[查看网页](#)

### 实现原理

云函数支持 [MQTT 消息队列触发器](#)。触发器基于 MQTT 共享订阅（Shared Subscription）机制，能够根据业务中定义的 Topic Filter 订阅并消费消息，继而自动触发对应的云函数执行。通过这一机制，云函数可与物联网设备建立高效、异步的通信链路。消息经由 MQTT 集群进行可靠路由与分发，通过共享订阅实现多个触发器的负载均衡，从而支撑高并发、需实时响应的物联网业务场景，如车辆状态监控与远程智能控制。



### 业务应用场景

#### 新能源车电量管理

汽车的TCU模块通过MQTT消息触发云函数，根据汽车的状态调用后续的不同服务

- 低电量预警与响应：车辆 TCU 检测到电量低于阈值时，通过 MQTT 主题发布状态消息，触发云函数执行通知用户、导航至充电站等操作。
- 远程设备控制：云函数通过 MQTT 向车辆发送指令（如限制空调功率），实现节能调控。
- 实时状态同步：通过 HTML/小程序页面订阅 MQTT 主题，实时查看车辆电量变化。

## 特性与优势

- **事件驱动与实时响应**：云函数由 MQTT 消息自动触发，无需轮询。设备状态变化（如电量低）可立即作为事件传递至云端，触发后续业务链。

实现了近乎实时的业务处理，极大降低了从事件发生到系统响应的延迟，极佳地满足物联网应用对实时性的要求。

- **无服务器架构，极致弹性与成本优化**：云函数以 Serverless 模式运行，根据 MQTT 消息的到达频率自动伸缩，无需管理服务器。零闲置成本，自动扩容可轻松应对突发流量，运维简化。
- **松耦合的集成方式**：MQTT 主题（Topic）作为消息总线，将设备上报与业务处理完全解耦。不同的业务功能（如通知、导航、节能控制）由不同的云函数独立处理，有较高的灵活性和可维护性。

## 操作步骤

### 策略和权限

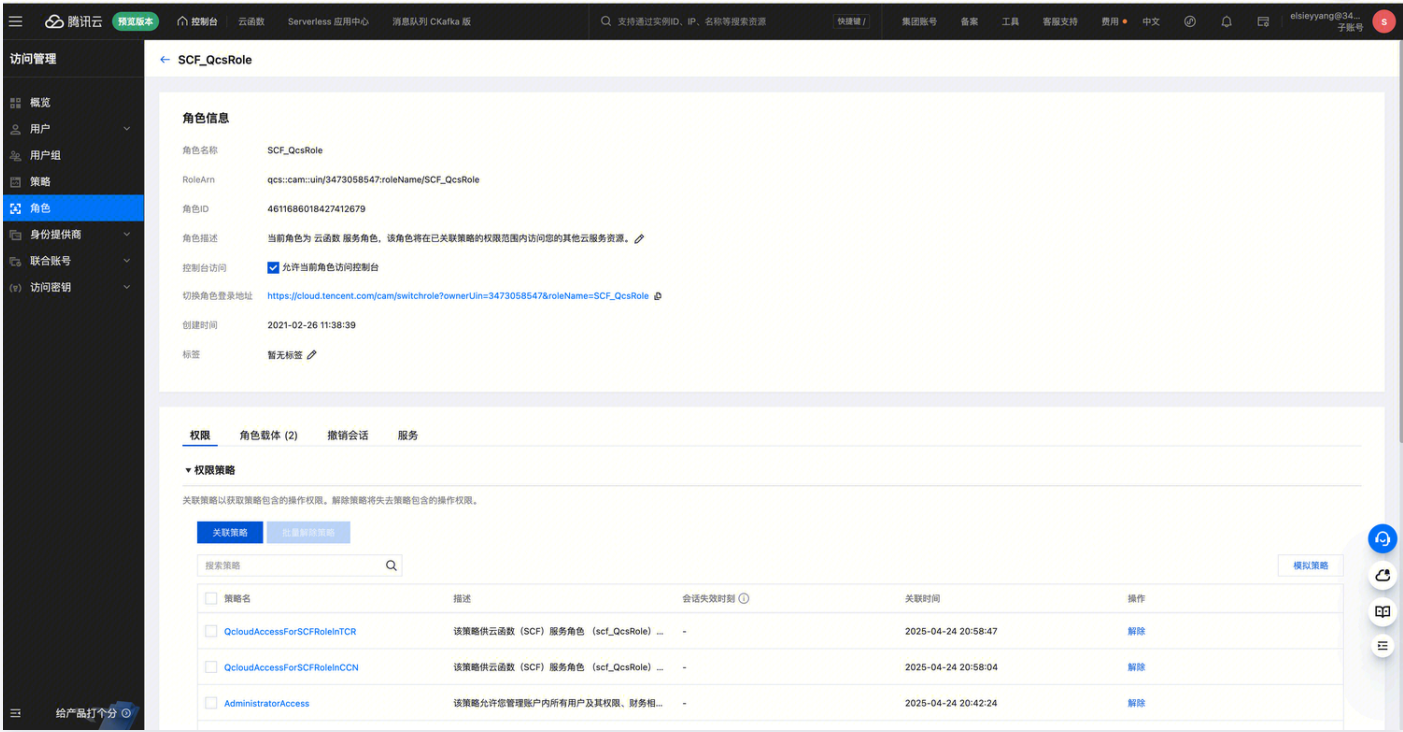
1. 检查 scf\_QcsRole 中是否有以下策略：[QcloudMQTTReadOnlyAccess](#)、[QcloudAccessForSCFRoleInMQTT](#)。

1.1 登录 [访问管理控制台](#)，跳转至【角色】菜单。

1.2 搜索 scf\_qcsRole



1.3 单击角色详情，查看策略。

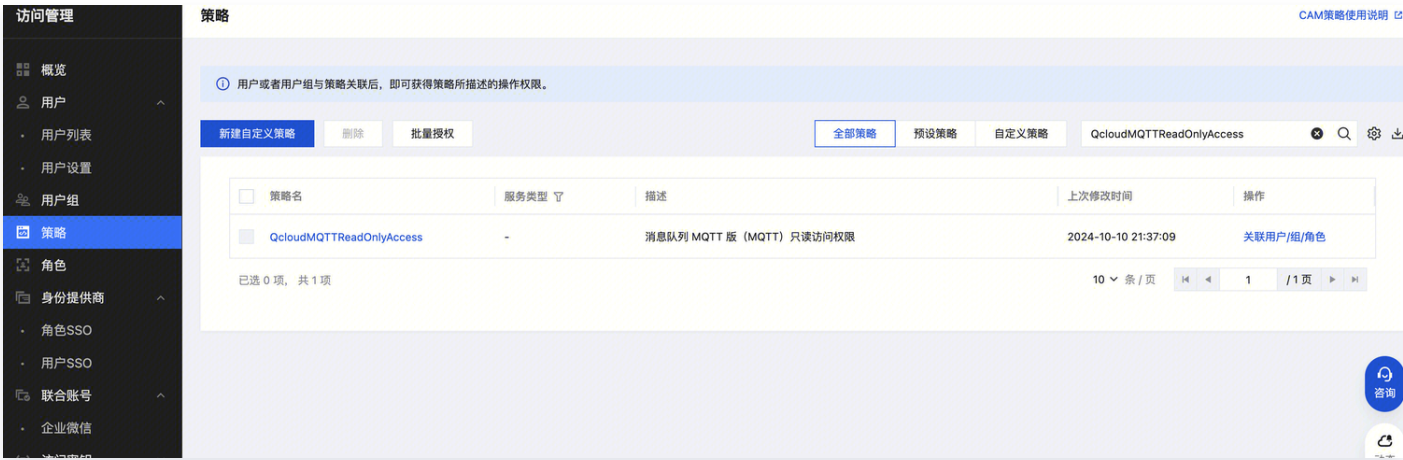


1.4 搜索 `QcloudMQTTReadOnlyAccess`、`QcloudAccessForSCFRoleInMQTT`，若能搜到对应的结果，则代表已关联成功。



1.5 策略缺失的情况下，可按照下列步骤为角色添加策略：

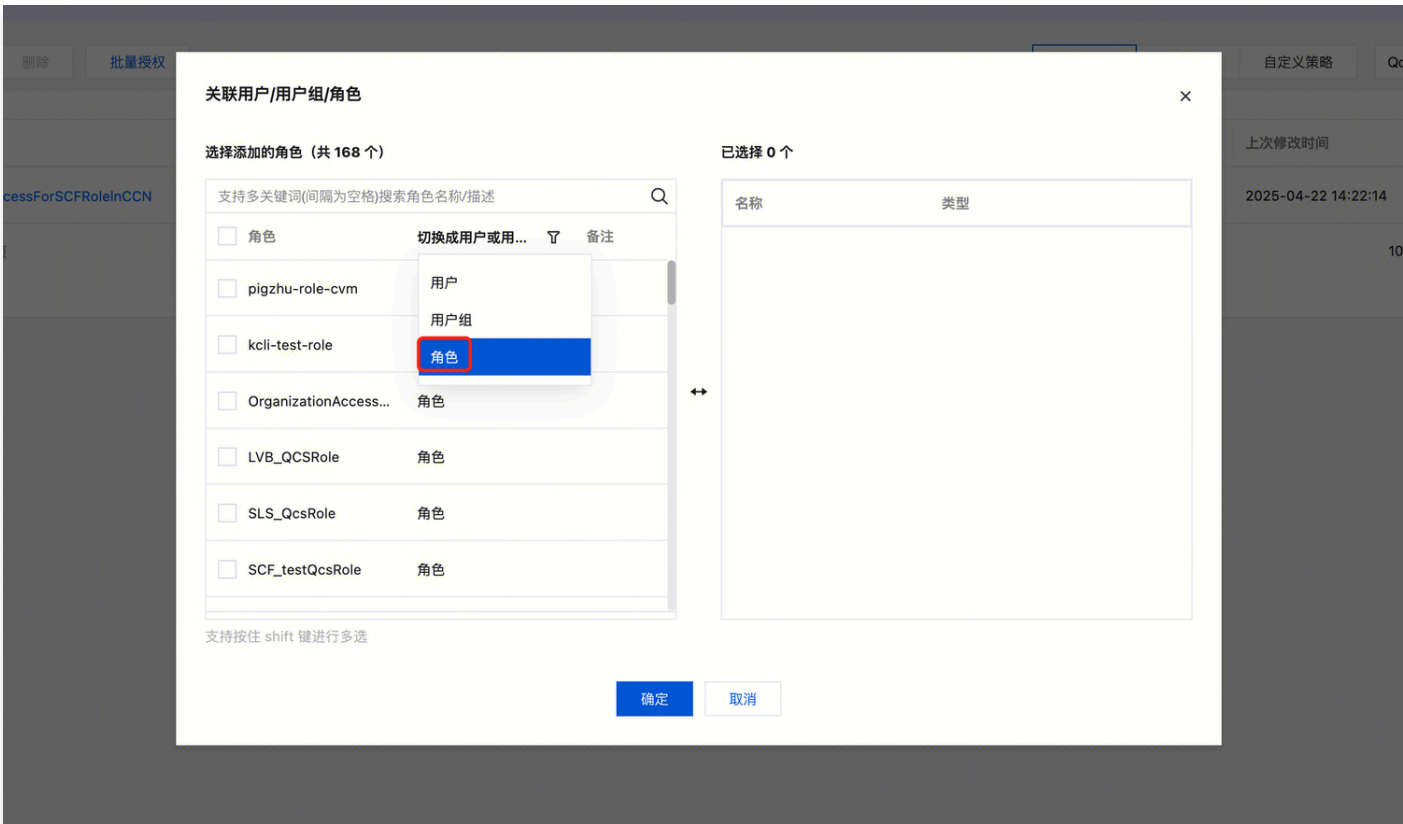
- 1.5.1 使用主账号登录腾讯云控制台
- 1.5.2 单击访问管理 > 策略
- 1.5.3 搜索缺失的策略，如：`QcloudMQTTReadOnlyAccess`



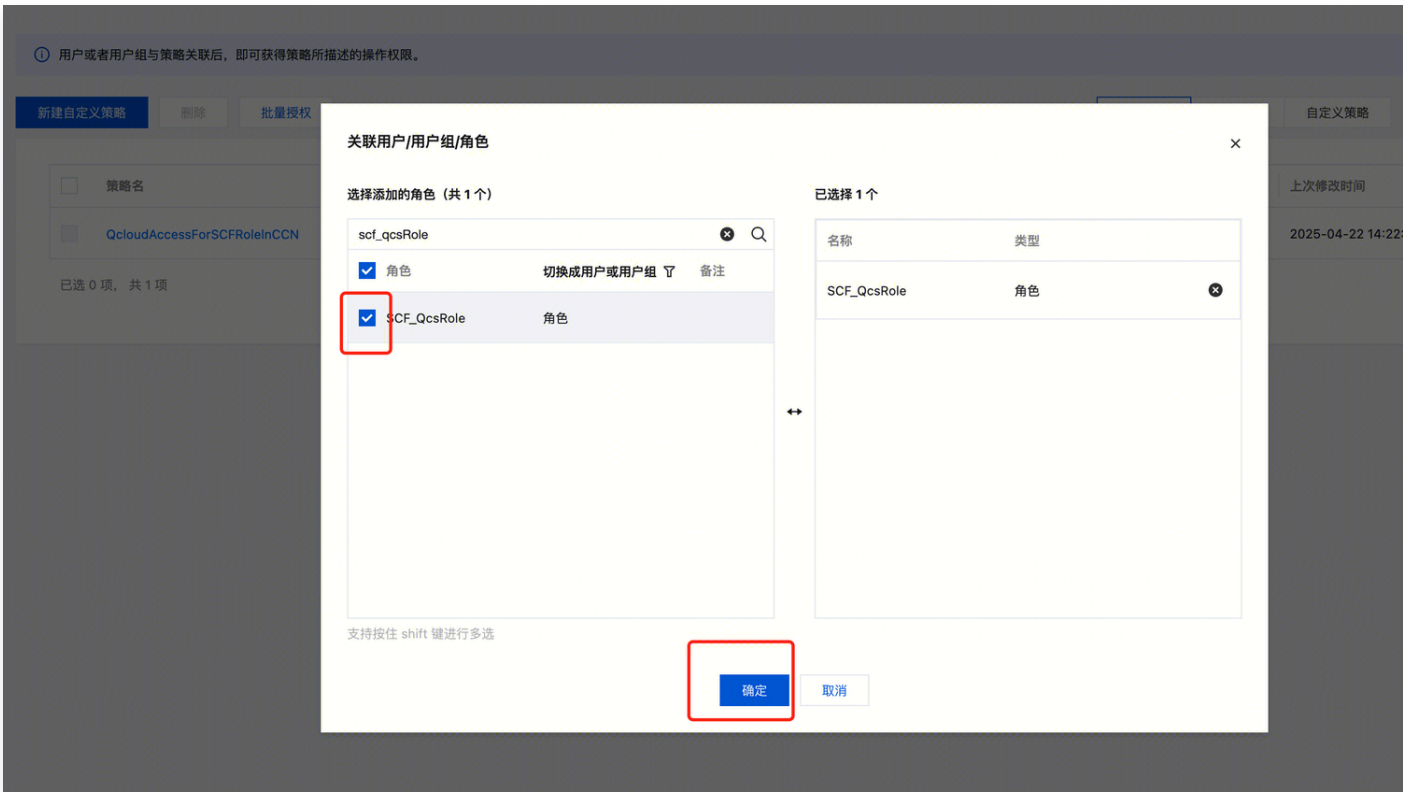
1.5.4 单击关联用户/组/角色。



1.5.5 在弹窗中，选择切换成角色。



1.5.6 搜索 scf\_qcsRole，勾选并确定。



1.5.7 页面提示已关联成功。



## 配置 MQTT 触发器

1. 登录 [Serverless 控制台](#)，跳转到函数服务页面。
2. 单击函数名，进入函数管理页面。



3. 选择函数配置 TAB，单击右上角编辑。



4. 如下图配置触发器。



5. 配置日志



6. 根据业务需要，填入需要的 Topic Filter，符合 Topic Filter 的消息触发执行 云函数

触发别名/版本

别名：默认流量

若采用灰度发布来控制版本流量，建议选择别名

触发方式

MQTT 触发

触发器名称

SCF-mqtt-1750312217347

MQTT 实例

mqtt-32jogd59 (lizhanhui-test)

MQTT 实例

订阅主题

选择 Topic自定义 Topic Filter

Topic Filter

#

支持订阅自建Topic及 MQTT 消息队列内置Topic，可使用“+”、“#”通配符表达更多语义

消费方式

顺序消费

非顺序消费

加密信息

账号root

密码.....

开启 Base64 标准编码

启用

启用后，会自动将您的消息内容进行 Base64 编码处理

最大批量消息数

1

重试次数

3

最长等待时间

60

立即启用

启用

我已阅读并同意《腾讯云函数网络服务协议》

完成

取消

7. 云函数创建完成后，观察MQTT 控制台 > 客户端管理。  
可以看到触发器订阅者列表。

消息队列 MQTT 版

集群管理 / lizhanhui-test

调整配置编辑销毁/退还

lizhanhui-test

运行中

ID mqtt-32jogd59

地域 华东地区(南京)

创建时间 2025-06-19 13:41:18

展开更多

基本信息

监控

Topic

客户端管理

认证管理

授权策略管理

云函数触发器

请输入客户端 ID

客户端 ID	连接状态	客户端地址	MQTT 协议版本	客户端创建时间	操作
lam-fvb5n8mk-7wtN6rJ2q	在线	21.72.12.126:50408	MQTT V3.1.1	2025-06-19 14:55:13	查看详情 断开连接
lam-fvb5n8mk-YNUnwZnDs	在线	30.182.59.113:59254	MQTT V3.1.1	2025-06-19 14:55:13	查看详情 断开连接
lam-fvb5n8mk-1T0jcpVcNp	在线	30.182.59.113:59244	MQTT V3.1.1	2025-06-19 14:55:13	查看详情 断开连接
lam-fvb5n8mk-rjoON3zZe7	在线	30.182.59.113:59228	MQTT V3.1.1	2025-06-19 14:55:13	查看详情 断开连接
lam-fvb5n8mk-IXAA2kk1ex	在线	21.72.12.126:50428	MQTT V3.1.1	2025-06-19 14:55:13	查看详情 断开连接
lam-fvb5n8mk-2hN3EuxfOh	在线	21.72.12.126:50406	MQTT V3.1.1	2025-06-19 14:55:13	查看详情 断开连接
lam-fvb5n8mk-6hcfxONBAd	在线	21.72.12.126:50416	MQTT V3.1.1	2025-06-19 14:55:13	查看详情 断开连接

查看客户端详情， 可以观察到订阅表达式与配置一致。

订阅关系

客户端事件

客户端消息轨迹

Topic 订阅	服务质量等级	未确认消息数量	堆积消息数量
\$share/lam-fvb5n8mkSCF-mqtt-1750316026075/sensor/#	1	0	0

共 1 条

10 条 / 页

1 / 1 页

动态

文档

反馈

验证

发送消息到 MQTT 集群, 查看云函数日志。

Serverless

概览

函数服务

Serverless 应用

高级能力

层

插件

资源购买

自定义域名

拓展能力

EB事件总线

函数管理

版本管理

别名管理

触发管理

函数 URL

监控信息

日志查询

并发配额

部署日志

helloworld-1750311997 正常

函数服务帮助文档

日志查询

日志查询功能由腾讯云日志服务CLS提供，[腾讯云日志服务免费额度](#) 已于2022年9月5日0点起调整。如果您暂时不需要日志投递功能，可以在「函数配置」中选择关闭，并前往 [CLS控制台](#) 删除不必要的存量函数日志，避免产生费用。

调用日志

高级检索

版本: \$LATEST

全部日志

近15分钟

2025-06-19 20:38:11 ~ 2025-06-19 20:53:11

刷新

请输入requestID

2025-06-19 20:53:03 调用成功

2025-06-19 20:53:02 调用成功

请求id: 4fd4ca55-fec3-40e5-93ed-81bc368b7109

常见错误说明及解决方案

时间: 2025-06-19 20:53:02 运行时间:1608ms 运行内存:82.12261199951172MB

日志:  
START RequestId: 4fd4ca55-fec3-40e5-93ed-81bc368b7109  
12:53:03,128 |INFO in ch.qos.logback.classic.util.ContextInitializer@747edf66 - Trying to configure with ch.qos.logback.classic.joran.SerializedModelConfigurator  
12:53:03,128 |INFO in ch.qos.logback.classic.util.ContextInitializer@747edf66 - No custom configurators were discovered as a service.  
12:53:03,125 |INFO in ch.qos.logback.classic.LoggerContext[default] - This is logback-classic version ?  
12:53:03,141 |INFO in ch.qos.logback.classic.LoggerContext[default] - Could NOT find resource [logback.scmo]  
12:53:03,140 |INFO in ch.qos.logback.classic.LoggerContext[default] - Could NOT find resource [logback-test.scmo]  
12:53:03,131 |INFO in ch.qos.logback.classic.util.ContextInitializer@747edf66 - Constructed configurator of type class ch.qos.logback.classic.joran.SerializedModelConfigurator  
12:53:03,171 |INFO in ch.qos.logback.classic.util.ContextInitializer@747edf66 - Constructed configurator of type class ch.qos.logback.classic.util.DefaultJoranConfigurator  
12:53:03,170 |INFO in ch.qos.logback.classic.util.ContextInitializer@747edf66 - Trying to configure with ch.qos.logback.classic.util.DefaultJoranConfigurator  
12:53:03,170 |INFO in ch.qos.logback.classic.util.ContextInitializer@747edf66 - ch.qos.logback.classic.joran.SerializedModelConfigurator.configure() call lasted 12 milliseconds. ExecutionStatus=INVOKE\_NEXT\_IF\_ANY  
12:53:03,381 |INFO in ch.qos.logback.core.model.processor.AppenderModelHandler - About to instantiate appender of type [ch.qos.logback.core.ConsoleAppender]

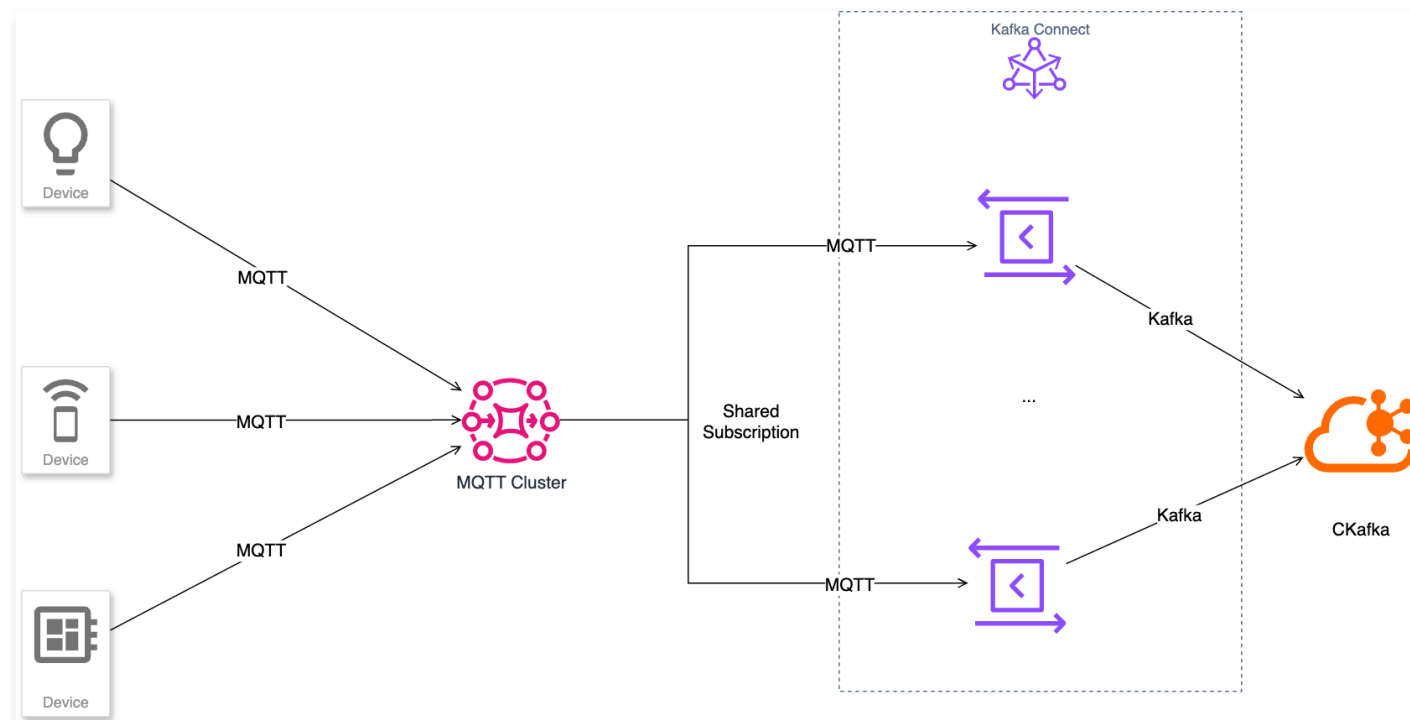
# 集成数据到 CKafka

最近更新时间：2026-01-30 15:10:03

[查看网页](#)

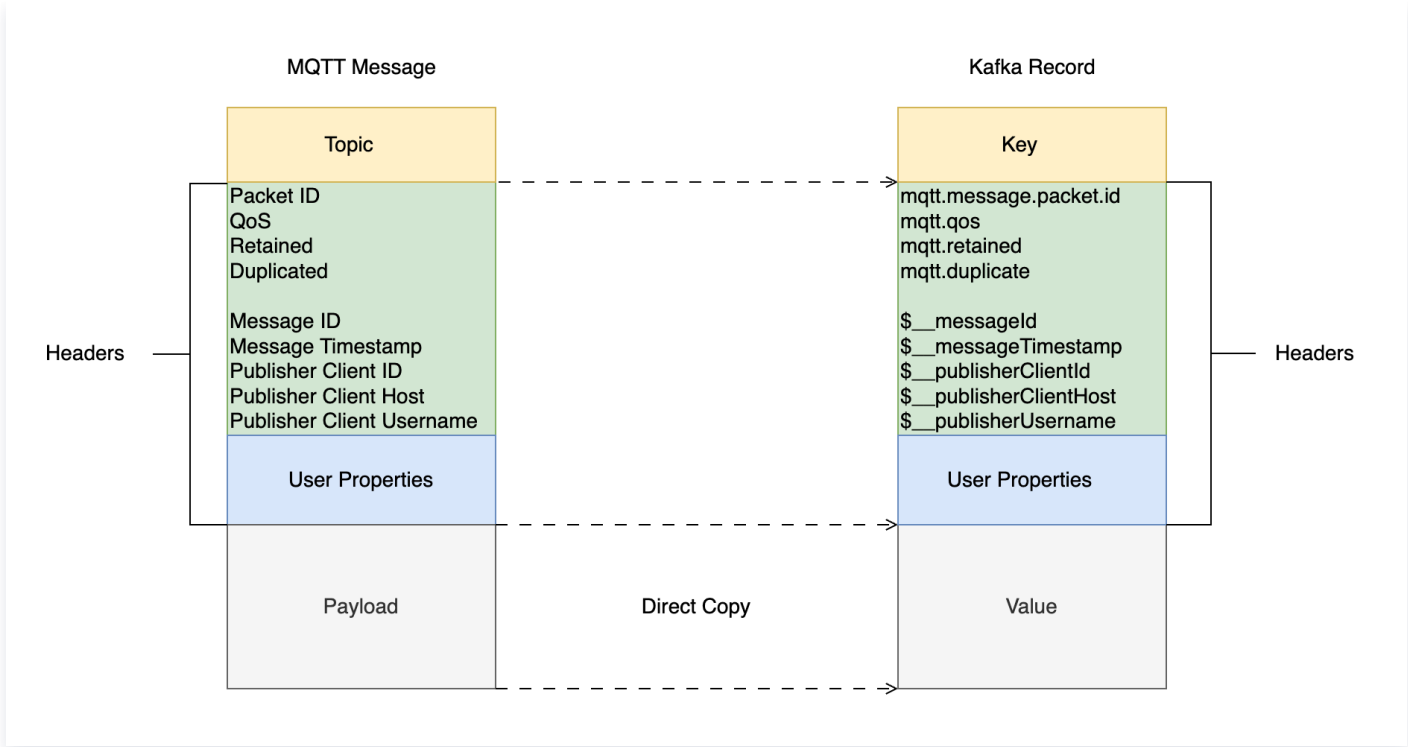
## 实现原理

CKafka 连接器内置 MQTT Source Plugin，通过 MQTT 共享订阅机制，实时接入 MQTT 消息并转发至 CKafka 集群。该共享订阅模式支持高并发配置，可有效保障数据传输吞吐量，充分满足 Kafka 与大数据生态集成时对高流量接入和处理能力的需求。



## 数据映射

MQTT 消息在转换为 Kafka Record 时, 映射关系如下:



## MQTT Message

一条 MQTT 消息由三部分组成: 系统字段、用户属性、Payload 参考: [MQTT Control Packet format](#)。

### 系统字段

字段名称	语义
Packet ID	控制指令 ID, 不唯一, 快速复用详见 <a href="#">Spec 2.2.1</a>
Duplicated	详见 <a href="#">Spec 3.3.1.1</a>
QoS	详见 <a href="#">Spec 3.3.1.2</a>
Retained	详见 <a href="#">Spec 3.3.1.3</a>
Message ID	扩展字段, 唯一消息编号
Message Timestamp	扩展字段, 服务端存储消息时间
Publisher Client ID	扩展字段, 发布消息的客户端标识符
Publisher Client Host	扩展字段, 发布消息的客户端 IP
Publisher Username	扩展字段, 发布消息的客户端用户名

### User Properties

用户指定的键值对列表, 参见 [Spec 3.3.2.3.7](#)。

## Kafka Record

字段	语义
Key	记录的键值, 可选
Headers	记录关联键值对, 常用来存储元数据, 比如 Content Type、事件时间等, 可选
Payload	记录的真正负载数据, 消息体

## Headers 使用场景

- Message 路由
- 元数据存储描述
- 链路追踪和日志
- 定制化业务处理
- 安全认证
- 消息优先级
- 互操作性/兼容性指令
- 流处理

## 业务应用场景

### 智慧城市与交通数字孪生

- 实时采集城市多源交通数据（如车辆车牌、速度、行驶轨迹），通过 MQTT 主题上报，并借助 Kafka 连接器接入大数据生态。
- 支持基于车牌号等属性进行高效检索与分析（如车辆轨迹还原），为交通监控、调度和仿真提供数据支撑。

## 特性与优势

消息队列 CKafka 版是一个分布式、高吞吐量、高可扩展性的消息系统，然其本身并非专为边缘物联网通信场景设计，其客户端通常需要稳定的网络环境和较高的硬件资源，而物联网领域中海量的设备和应用产生的数据往往通过轻量级的 MQTT 协议进行传输。通过 CKafka MQTT 连接器实现 MQTT 协议与 CKafka 生态的无缝集成，将设备端发布的 MQTT 消息实时流入 CKafka 主题，确保数据能够被实时处理、存储和进一步分析。该集成不仅保留了 MQTT 在弱网与低资源环境下的通信优势，还充分发挥 CKafka 在高吞吐、高可靠及生态兼容性方面的能力，真正实现了物联网数据与大数据系统间的灵活、稳定和高效整合。

## 操作步骤

### 策略与权限

1. 登录 [消息队列 MQTT 版控制台](#)，进入[集群详情](#)页面，确认当前 MQTT 集群是否开启授权策略管理。

- 1.1 若未开启权限策略，数据面资源暂无权限管理，可以使用任意“用户名+密码”进行连接、生产和消费等操作，详情见 [配置数据面授权](#)。在此情况下，进行数据集成到CKafka的操作时无需其他额外配置，但是由于缺少权限管控，会存在一定的数据安全风险。
  - 1.2 若已开启权限策略，请按照下文所述步骤进行操作。
2. 进入 **认证管理** > **用户名和密码**，单击**新建用户**，为数据集成任务创建专用账号和密码，用户名为 `ckafka_connector`，并在说明中注明此用户为仅用于 MQTT 与 CKafka 数据集成任务，如图所示。

添加角色

×

用户名 \*

ckafka\_connector

不能为空，支持数字、大小写字母和符号（空格除外），不能超过 32 个字符

密码 \*

自动生成密码

自定义密码

说明

仅用于MQTT与CKafka数据集成任务。

不能超过 128 个字符

保存

取消

3. 进入 **授权策略管理** 页面，单击**新建授权策略**，强烈建议在本策略中明确授权上一步所创建的 CKafka 数据集成专用账号，以实现精确的权限控制。具体配置方式可参考下图，其余字段请结合实际需求填写，详情参考 [配置数据面授权](#)。

## ← 新建授权策略

策略名称 \*

不能为空，3-64个字符，支持中文、字母、数字、“-”及“\_”

描述

仅供MQTT与CKafka数据集成任务使用。

创建方式

策略生成器

JSON

效果(Effect)



允许



拒绝

操作 \*

全部



Topic \*

新增一行

支持使用通配符和策略变量，最多 10 条，详见 [表达式说明](#)。

用户名

ckafka\_connector

填写单个用户名或单条资源表达式，支持使用通配符和策略变量，详见 [表达式说明](#)，为空表示支持所有用户名。

客户端 ID

请输入客户端 ID

填写单条资源表达式，支持使用通配符和策略变量，详见 [表达式说明](#)，为空表示支持所有客户端。

IP 地址

请输入IP 地址

来源支持以下格式: 单个IP: 192.168.0.1 或 CIDR: 192.168.1.0/24

QoS \*

全部



消息类型 \*



保留消息



普通消息

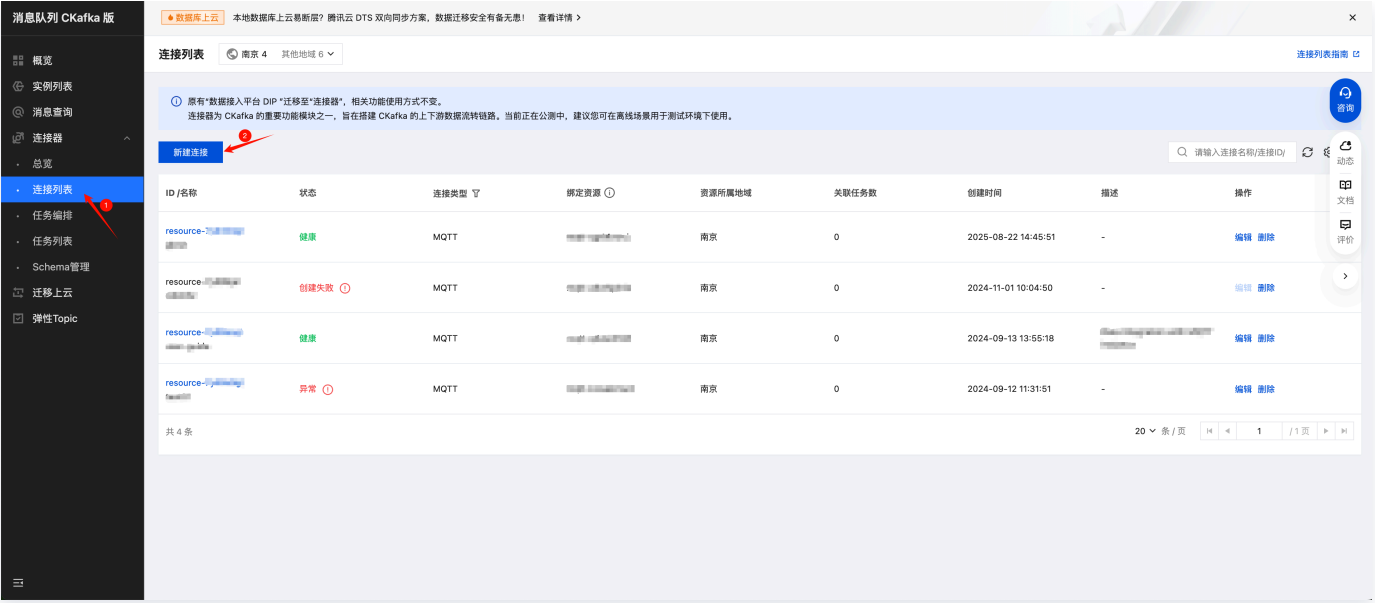
全部勾选表示支持所有消息

创建策略

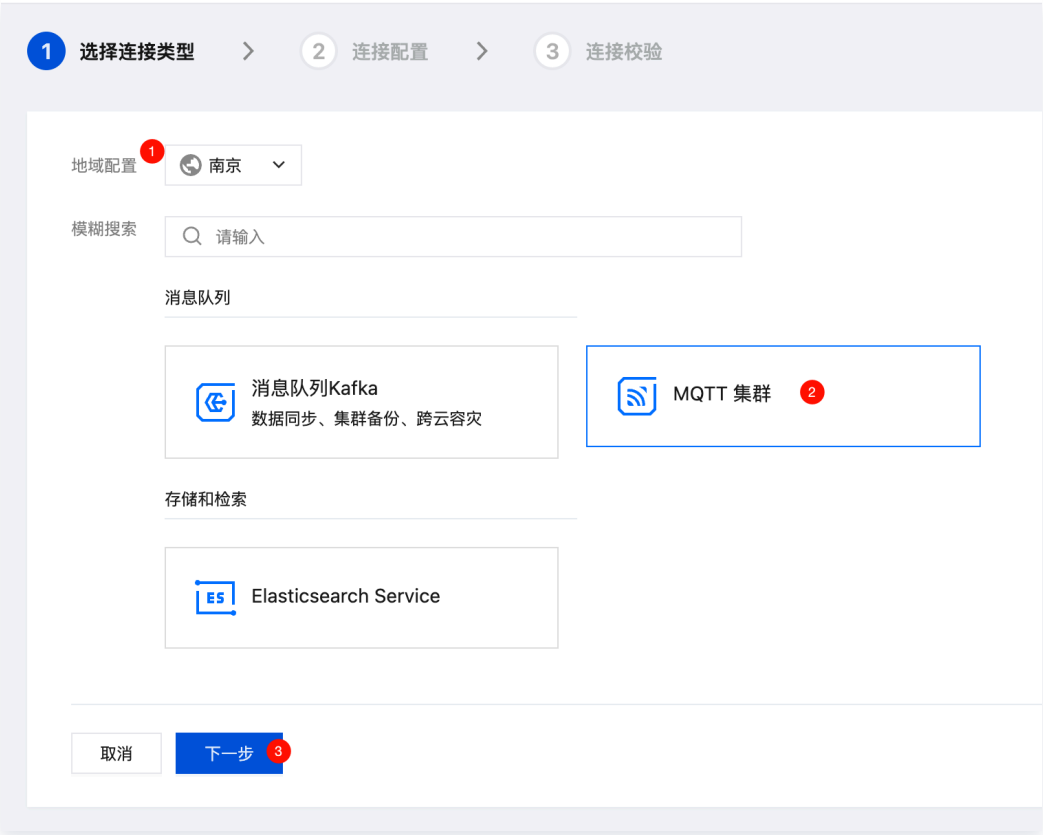
关闭

## 配置 CKafka 连接器

1. 登录 [消息队列 CKafka 版控制台](#)，进入[连接列表](#)页面，首先在页面最上方确认连接的所属地域。
2. 单击[新建连接](#)，进行连接器的创建



3. 按下图步骤进行连接信息的选择，连接类型选择 **MQTT 集群**，单击下一步进入连接配置页面。



4. 输入连接名称、描述等基本信息，并在下拉框中选择目标 MQTT 集群。此处的用户名和密码用作连接认证，是在 MQTT 集群中创建的数据集成专用账号密码，详见 [策略和权限](#) 小节。单击下一步进入连接校验过程。

← 新建连接

✓ 选择连接类型

>

2 连接配置

>

3 连接校验

连接名称

描述

MQTT 集群

mqtt-3/

用户名

ckafka\_connector

密码

.....

上一步

下一步

5. 当校验均通过后，连接即创建成功。您可在 **CKafka 控制台 > 连接器 > 连接列表** 中查看新增的连接。
- 对于已创建的连接，连接列表中展示其基本信息，包括 ID、名称、状态、连接类型、绑定资源、资源所属地域、关联任务数、创建时间、描述等。
  - 单击操作列中的**编辑**按钮，可修改连接配置。更新连接后，系统会默认开启“更新并重启所有关联任务”的开关，请在操作时根据实际业务需求谨慎选择；
  - 单击操作列中的**删除**按钮将删除此连接。

消息队列 CKafka 版

性能提升

腾讯云新一代服务器 SA9 性能提升 30%+，成本直降 10%！专为中小企业减负，每一分投入都能看见算力回报！

查看详情 >

连接列表

南京 5 其他地域 6

连接列表

原有“数据接入平台 DIP”迁移至“连接器”，相关功能使用方式不变。  
连接器为 CKafka 的重要功能模块之一，旨在搭建 CKafka 的上下游数据流转链路。当前正在公测中，建议您可在高线场景用于测试环境下使用。

新建连接

请输入连接名称/连接ID

ID / 名称	状态	连接类型	绑定资源	资源所属地域	关联任务数	创建时间	描述	操作
resource-...	健康	MQTT	mqtt-...	南京	0	2025-08-22 14:45:51	-	编辑 删除
resource-...	创建失败	MQTT	mqtt-...	南京	0	2024-11-01 10:04:50	-	编辑 删除
resource-...	异常	MQTT	mqtt-...	南京	0	2024-09-13 13:55:18	-	编辑 删除
resource-...	异常	MQTT	mqtt-...	南京	0	2024-09-12 11:31:51	-	编辑 删除

共 4 条

20 / 页

创建数据集成任务

前提条件

在 MQTT 集群的同一地域下，已有创建好的 CKafka 实例，具体操作详情参阅 [消息队列 CKafka 快速入门](#)。

## 任务创建

1. 进入 [CKafka 控制台](#) > [连接器](#) > [任务列表](#)，单击左上角新建任务，进行任务相关信息填写，任务类型选择 **数据接入** > **MQTT集群**，单击下一步进入数据源配置。

新建任务

1 基础设置 > 2 数据源配置 > 4 数据目标配置

地域配置 \* 南京

任务名称

只能包含字母、数字、下划线、“-”、“.”

任务类型 **数据接入** 数据流出

模糊搜索

其他

MQTT 集群

取消 下一步

2. 在下拉框中选择合适的连接，若没有合适选项，可单击下方跳转按钮，进入**新建连接**步骤；输入订阅的 Topic，若订阅多个主题请用“，”隔开。

新建任务

基础设置 > 2 数据源配置 > 4 数据目标配置

数据源

没有合适的连接，[前往新建](#)

订阅的 Topic

是否对数据进行处理或校验 ☐

上一步 下一步

3. 进行数据目标配置，确定分发策略以及目标CKafka实例。单击提交完成任务创建。

← 新建任务

✓ 基础设置

>

✓ 数据源配置

>

4 数据目标配置

分发策略

自动创建 Topic

选择已有 Topic

Topic类型

CKafka实例内Topic

CKafka实例

ckafka-xxxxxx-xxxxxx-xxxxxx

目标 Topic

若实例设置了ACL策略，请确保选中的topic有读写权限

数据压缩算法

不开启

数据压缩可以减少网络 IO 传输量，减少磁盘存储空间，[数据压缩说明](#)

上一步

提交

4. 当任务创建成功后，MQTT 集群下会自动新建一个共享订阅组，用于执行数据集成。

共享订阅组管理

① 在使用共享订阅前，需要在页面先创建完成共享订阅组（ShareName），详见[共享订阅组](#)。

新建 (1/20)

请输入关键字进行搜索

共享订阅组名称	负载均衡策略	均衡生效时延	描述	创建时间	更新时间	操作
group-resource-7j-xxxxxx-task-xxxxxx	随机	-	-	2025-10-16 16:32:36	2025-10-16 16:32:36	<a href="#">编辑</a> <a href="#">删除</a>

也可前往 客户端管理 页面查看执行该任务的连接器客户端详情。

客户端管理

南京

当前集群

mqtt-xxxxxx-xxxxxx

产品体验，您说了算

客户端管理

请输入客户端 ID

客户端 ID	连接状态	客户端地址	MQTT 协议版本	证书序列号(SN)	客户端创建时间	操作
xxxxxx-xxxxxx-xxxxxx-xxxxxx-xxxxxx	未连接	xxxxxx-xxxxxx-xxxxxx-xxxxxx-xxxxxx	MQTT V3.1.1	-	xxxxxx-xxxxxx-xxxxxx-xxxxxx-xxxxxx	<a href="#">查看详情</a> <a href="#">断开连接</a>
xxxxxx-xxxxxx-xxxxxx-xxxxxx-xxxxxx	未连接	xxxxxx-xxxxxx-xxxxxx-xxxxxx-xxxxxx	MQTT V3.1.1	-	xxxxxx-xxxxxx-xxxxxx-xxxxxx-xxxxxx	<a href="#">查看详情</a> <a href="#">断开连接</a>
resource-7j-xxxxxx-task-xxxxxx-xxxxxx	在线	xxxxxx-xxxxxx-xxxxxx-xxxxxx-xxxxxx	MQTT V5	-	2025-10-16 16:32:36	<a href="#">查看详情</a> <a href="#">断开连接</a>

共 3 条

20 条 / 页

1 / 1 页

# 集成数据到 RocketMQ

最近更新时间：2026-01-30 15:10:03

❗ 说明：当前功能灰度开放中，若您需要使用可以通过工单 [联系我们](#)。

[查看网页](#)

## 实现原理

腾讯云消息队列 MQTT 版支持与消息队列 RocketMQ 版之间的消息路由转发：MQTT 消息经由 RocketMQ 提供的跨集群复制能力，根据预配置的路由规则，被转发至 RocketMQ 的相应主题中；同理 RocketMQ 的消息也可以投递到 MQTT 中，进而进一步投递到设备端。

RocketMQ 作为分布式消息队列，负责对消息进行持久化和分发，供下游多个业务服务消费与处理，从而形成从物联网终端状态感知到云端业务响应的闭环机制。整个过程实现了 MQTT 协议与 RocketMQ 协议之间的透明转换与可靠消息同步，确保了设备状态变更能够实时、准确地触达后端业务系统。

## 业务应用场景

### 充电桩设备的智能运维与状态管理

- 设备连接与遗嘱设置：充电桩设备在通过 MQTT 协议连接至云端服务端时，会预先在其连接请求中设置一条遗嘱消息，并指定其将被发布的 MQTT 主题。
- 异常离线与遗嘱触发：当充电桩因突发停电、网络中断或自身故障等原因异常断开连接时，MQTT 服务端会立即自动将该设备预设的遗嘱消息发布到指定的主题中，从而向云端宣告该设备已非正常离线。
- 跨协议消息路由：通过预先在 TDMQ 控制台配置的消息路由规则，MQTT 服务端指定主题上的遗嘱消息会被自动、可靠地转发至 TDMQ RocketMQ 的一个特定主题中。
- 运维服务消费与状态感知：后端的云端检修服务持续订阅 RocketMQ 的主题。一旦有新的遗嘱消息（即故障消息）到达，该服务便能立刻消费并解析该消息，准确获取故障设备的身份 ID、位置信息及可能的错误代码，在识别到设备异常状态后，随即自动触发后续的业务流程。

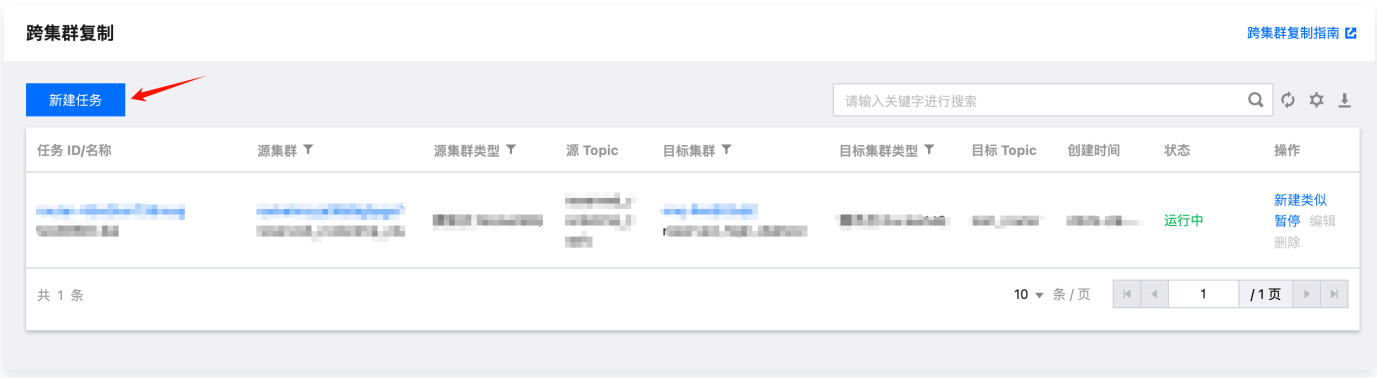
## 特性与优势

- 高可靠与自动化：可靠地通过批处理并发送 MQTT 消息到 RocketMQ，实现物联网设备与 RocketMQ 及应用系统的集成。
- 灵活的主题映射：RocketMQ 数据集成支持将 MQTT 主题灵活映射到 RocketMQ 主题。
- 业务可扩展性：RocketMQ 作为消息中间件，允许多个业务服务同时消费设备状态消息，支持系统平滑扩展和业务灵活迭代。

## 操作步骤

### 创建跨集群复制任务

- 1. 登录 消息队列 RocketMQ 版控制台，进入 跨集群复制 页面。
- 2. 单击新建任务，进行任务的创建。



- 3. 选定消息来源和消息复制目标，按要求填写字段：

← 新建跨集群复制任务

任务类型

Topic 跨集群复制

任务名称

请输入任务名称

消息来源

腾讯云 RocketMQ

腾讯云 MQTT

源 Topic

地域

北京

集群

暂无数据

Topic

请选择

/ 不填写认为复制到一级 Topic

用户

请选择

消息复制目标

腾讯云 RocketMQ

目标 Topic

地域

北京

集群

请选择

Topic

是否立即开启任务

☒

创建任务

关闭

- 任务名称：200字符以内，只能包含 中文、数字、字母、“-”和“\_”；
- 源 Topic：通过下拉依次选择地域、集群、和 Topic，如果没有合适的集群或 Topic 可以在集群列表页进行新建。
- 目标 Topic：通过下拉依次选择地域、集群、和 Topic，如果没有合适的集群或 Topic 可以在集群列表页进行新建。

- 是否立即开启任务：如果打开开关，在任务创建完成后就按照当前任务的配置进行复制。

3.1 单击**创建任务**后，会跳转到任务列表页，在任务初始化后即创建完成。

## 查看任务详情

- 在任务创建完成后，您可以在任务的列表页看到新增的复制任务，同时可以快速查看任务的状态。单击操作列的**启动/暂停**可以快速的开启和暂停任务。
- 您可以单击任务名称，进入任务详情页查看任务的详细配置。运行中的任务不能修改配置信息，如果要修改复制任务的配置，请先暂停任务后，单击操作栏的**编辑**，或者进入任务详情页，单击“基本信息”右上角的**编辑**，修改任务的信息。



- 在监控部分，您可以查看当前消息复制任务的实时监控，例如源消息消费总条数、消息复制失败条数、消息同步延迟等。

