消息队列 MQTT 版

控制台指南

腾讯云

【版权声明】

【商标声明】

【服务声明】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。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。

【联系我们】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或95716。

文档目录

控制台指南
集群管理

最近更新时间：2025-01-02 17:05:32

集群是 MQTT 中的一个资源维度，不同集群的 Topic、 客户端等完全隔离。每个集群会有集群的资源限制。例如：Topic 总数、连接数、订阅数等。常见的使用方式例如有：开发测试环境使用一个专门集群，生产环境使用一个专门的集群。

创建集群

1. 登录 MQTT 控制台 。
2. 在左侧导航栏选择**资源管理** > **集群管理**，选择好地域后，单击**新建集群**进入集群购买页面。

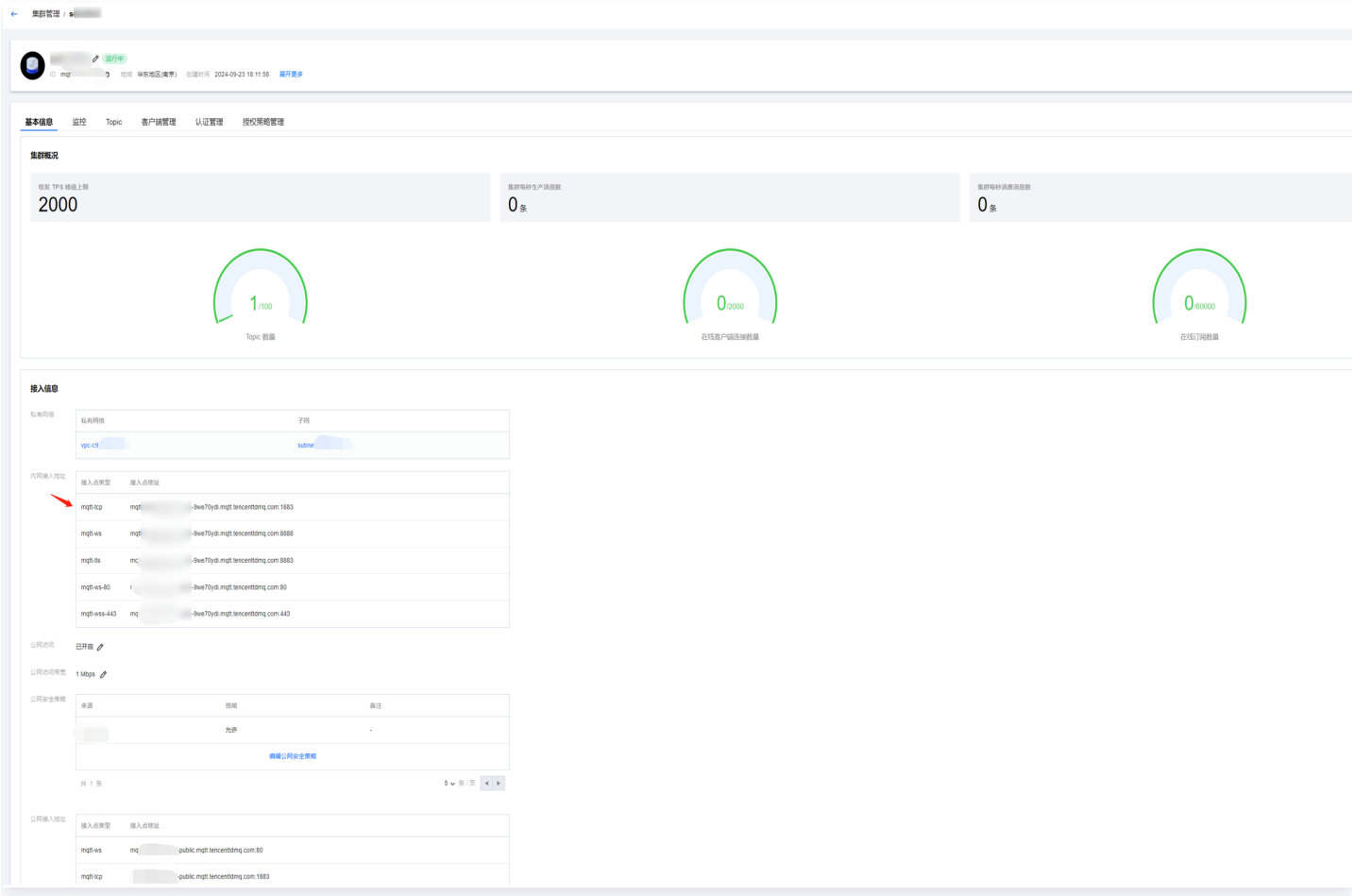
配置	说明
计费模式	当前支持包年包月和按量计费两种模式。
地域	选择与您的业务最靠近的地域，处于不同地域的云产品内网不通，购买后不能更换，请您谨慎选择。例如，广州地域的云服务器无法通过内网访问上海地域的集群。若需要跨地域内网通信，请查阅 云联网 。
集群规格	当前提供了基础版和专业版两种版本规格，两个版本之间的差异可以查看 产品系列 。
TPS 规格	包含生产消息和消费消息的总和，按照设置不同对单条消息进行折算，详细规则见 购买说明 。
客户端连接数	基础版最高支持 2000 个客户端连接数，不支持额外购买连接数；专业版根据 TPS 规格不同赠送的免费连接数不同，可单独额外购买连接数。
私有网络	授权将新购集群接入点域名绑定至私有网络（VPC）。
公网	开启公网带宽后会新增单独的费用，开通后可在集群管理页关闭。计费价格参见 公网计费说明 。
集群名称	填写集群名称， 长度不能超过64个字符，只能包含数字、字母、“-”和“_”。
标签	标签用于从不同维度对资源分类管理。

3. 选好配置单击**立即购买**，等待3～5分钟，即可在控制台看到完成创建的集群。

查看集群详情

在 集群管理 列表页，单击集群的“ID”，进入集群详情页面。在详情页中，您可以查询到：

- 集群的基础信息：集群名称/ID、地域、创建时间、说明等。
- 集群概况：展示所选时间范围内当前集群的收发 TPS 峰值上限、集群每秒生产消息数、集群每秒消费消息数。
- 接入信息：展示私有网络、内网接入地址和公网接入点信息。



编辑集群信息

1. 在**集群管理**列表页，单击操作列的**编辑**。

2. 在弹窗中填写集群名称和集群说明，单击**提交**即可完成修改。

销毁集群

1. 在**集群管理**列表页，单击操作列的**更多>销毁/退还**。
2. 在销毁实例的确认弹框中，确认**销毁**，即可删除集群。

> ⚠ **注意：**
> 销毁后，该集群下的所有配置都会被清空，且无法恢复，请谨慎操作。

调整集群配置

如当前的集群规格不满足您的业务需求，您可以在控制台上调整您的集群规格，包括 集群规格、TPS 规格、客户端连接数。

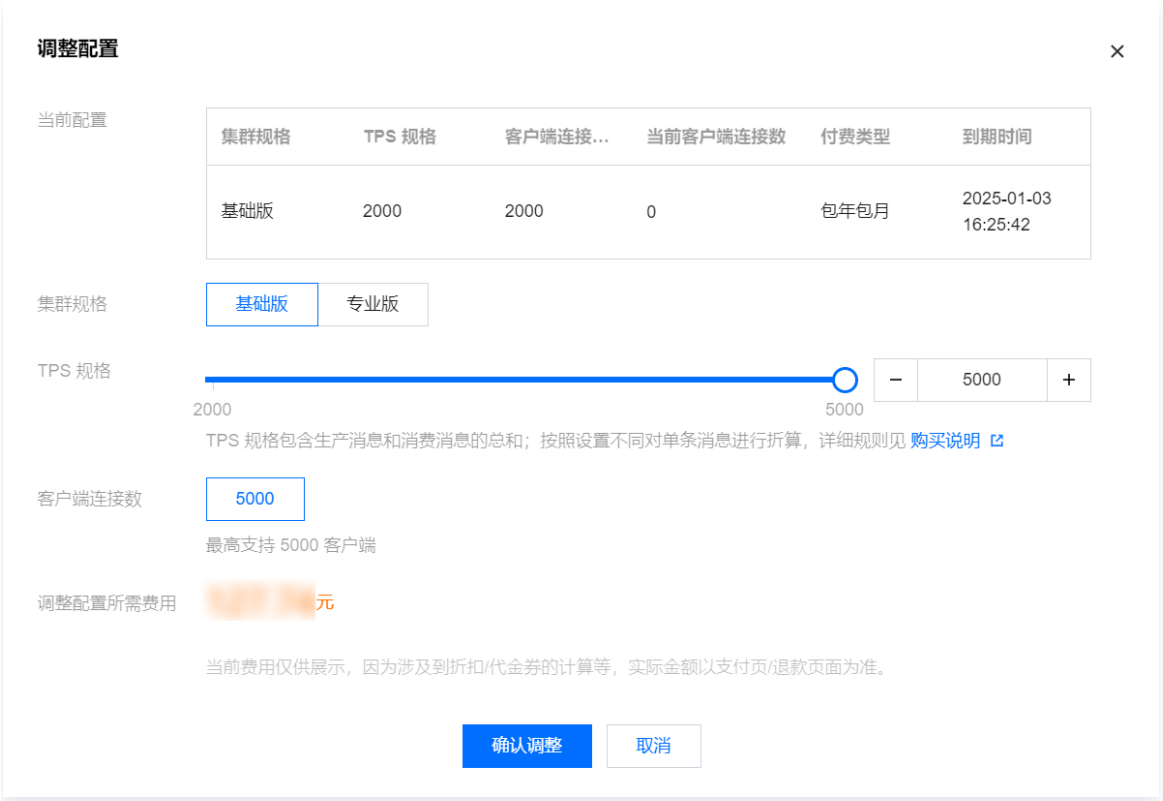
调整集群规格有两个入口：

- 入口一：登录 MQTT 控制台，在集群管理列表页，单击需要调整规格的集群其操作列的**更多** > **调整配置**。
- 入口二：登录 MQTT 控制台，在**集群基本信息**页面，右上角单击**调整配置**。

调整配置信息如下：

- 目标集群规格：当前支持基础版和专业版，两个版本之间的差异可以查看 产品系列 。
- 目标 TPS 规格：在集群规定的范围内，提升或者降低特定的售卖 TPS 规格。
- 客户端连接数：仅专业版支持调整。

> ⚠ **注意：**
> - 在进行集群规格的调整，尤其是降低配置时，需要格外注意集群的 TPS 和连接数。如果当前的客户端连接数和 TPS 超出了目标集群规格，则在降低配置后，会出现限流情况。
> - 在进行跨类型的降配时，如从专业版降配为基础版时，需要检查是否有高级功能正在使用中（功能差异可以查看 产品系列 ），避免出现升级后功能不可用带来的业务中断。


调整配置
当前配置
集群规格
TPS 规格
客户端连接...
当前客户端连接数
付费类型
到期时间
基础版
2000
2000
0
包年包月
2025-01-03 16:25:42
集群规格
基础版
专业版
TPS 规格
2000
5000
5000
TPS 规格包含生产消息和消费消息的总和；按照设置不同对单条消息进行折算，详细规则见 购买说明
客户端连接数
5000
最高支持 5000 客户端
调整配置所需费用
元
当前费用仅供展示，因为涉及到折扣/代金券的计算等，实际金额以支付页/退款页面为准。
确认调整
取消

Topic 管理

最近更新时间：2024-12-27 14:25:13

操作场景

MQTT 协议基于 Pub/Sub 模型，Topic 通常用来对系统生产的各类消息做一个集中的分类和管理。客户端可以订阅一个或多个 Topic，以便于接收和这些主题相关的消息。同样客户端也可以发布消息到某个 Topic，以便其他订阅该 Topic 的客户端接收到这些消息。
根据标准 MQTT 协议，Topic 存在多级，且拥有动态的特性，中间用“/”分隔，定义第一级 Topic 为父级 Topic。使用前，需要在 MQTT 控制台创建父 Topic，二级 Topic 无需创建，直接在代码中设置。

使用限制

单集群中最多可创建300个 Topic。

前提条件

已创建好对应的集群，详情参见 集群管理 。

操作步骤

创建 Topic

1. 登录 MQTT 控制台 。
2. 在左侧导航栏单击**资源管理** > **集群管理**，选择好地域后，单击目标集群的“ID”，进入集群基本信息页面。
3. 在顶部页签选择**Topic**，单击**新建**，填写好 Topic 名称和说明。

4. 单击**提交**，在 Topic 列表中即可看见创建好的 Topic。

编辑 Topic

1. 在 Topic 列表中，找到需要编辑的 Topic ，单击操作栏中的**编辑**。
2. 在弹出的对话框中可以对 Topic 说明进行编辑。
3. 单击**提交**即完成对 Topic 的编辑。

删除 Topic

- **批量删除：**在 Topic 列表中，勾选所有需要删除的 Topic，单击左上角的**批量删除**，在弹出的提示框中，单击**删除**，完成删除。
- **单个删除：**在 Topic 列表中，找到需要删除的 Topic，单击操作列的**删除**，在弹出的提示框中，单击**删除**，完成删除。

> **注意：**
> Topic 删除后，该 Topic 下的所有配置将会被清空，且无法恢复，请谨慎执行。

Topic 详情页

单击一级 Topic 名称，进入 Topic 详情页。在 Topic 详情页可以查看 Topic 的基本信息，同时可以输入二级 Topic（子 Topic）的全名查找具体的二级 Topic，如果存在此二级 Topic，则会展示二级 Topic 的具体信息，如在线订阅关系和在线客户端等等。

集群管理 / Topic 管理 / **test1**

基本信息

Topic 名称	test1	在线订阅数量	30
描述	-	在线订阅客户端数量	30
创建时间	2024-04-15 14:19:09		

子 Topic

请输入子级 Topic 名称，例如 "topic/subtopic"

子 Topic	在线订阅数量	在线订阅客户端数量
10/	1	1
20/	1	1
30/	1	1
9/	1	1
19/	1	1
3/	1	1

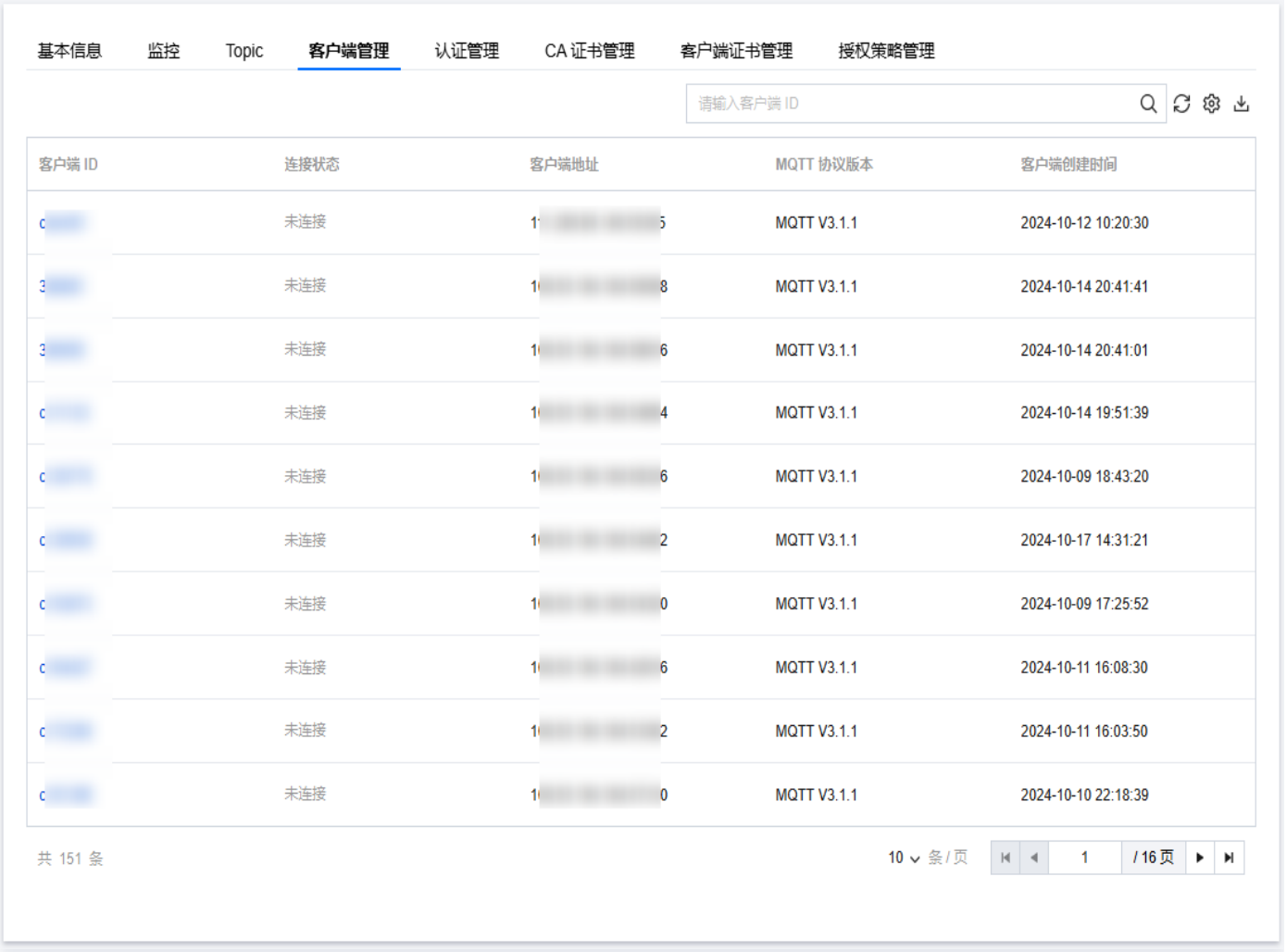
客户端管理

最近更新时间：2025-01-02 17:05:32

本文介绍如何 MQTT 控制台上查看客户端信息包含基本信息、连接信息和订阅关系等，帮助您实时掌握客户端状态，针对相关问题进及时进行处理。

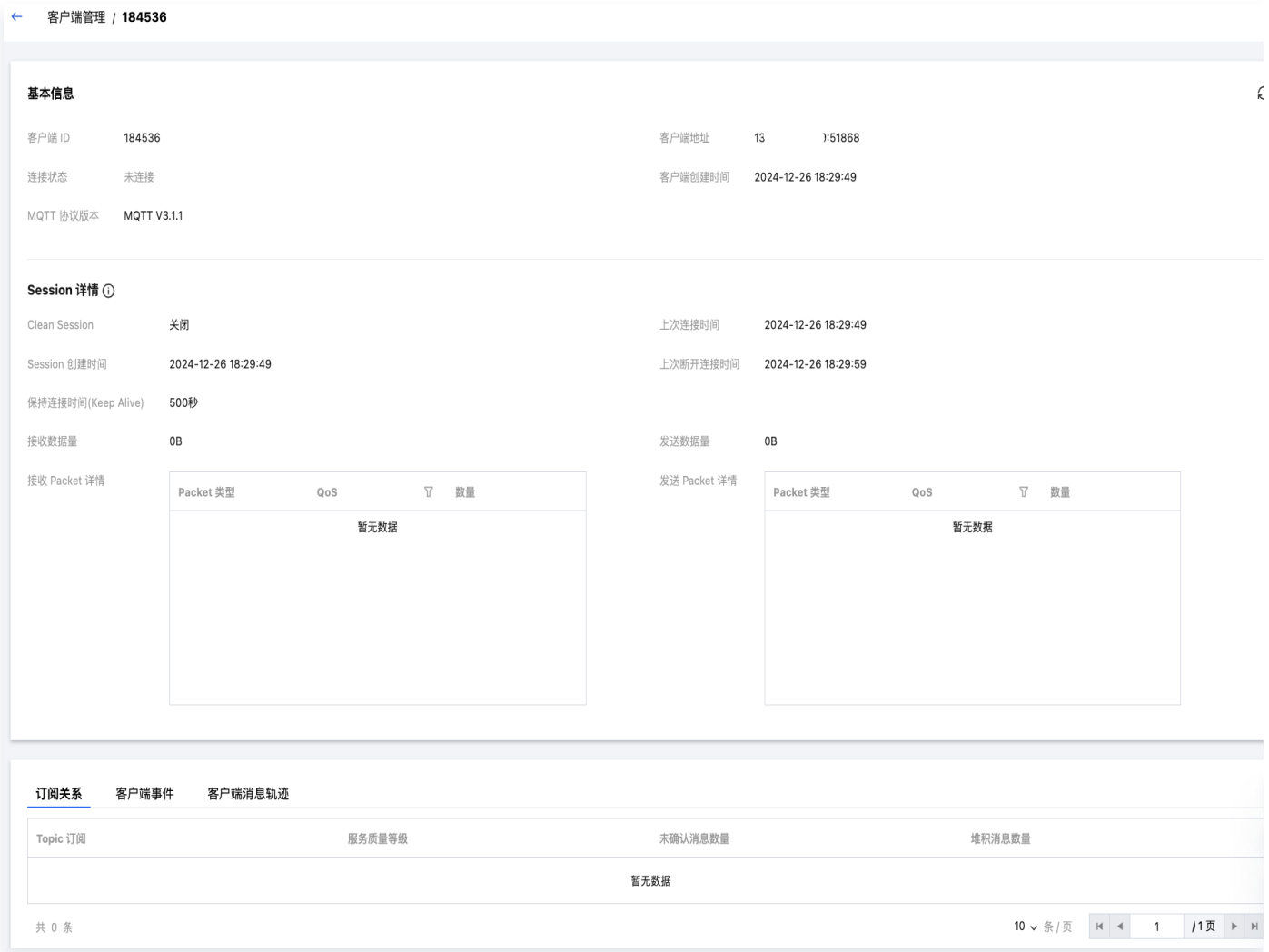
操作步骤

1. 登录 MQTT 控制台 。
2. 在左侧导航栏单击**资源管理** > **集群管理**，选择好地域后，单击目标集群的“ID”，进入集群基本信息页面。
3. 在顶部页签栏选择**客户端管理**，即可查看客户端列表。



4. 单击客户端 ID，可以进一步查看该客户端相关信息。
   - 基本信息：客户端基本信息，包含客户端的 ID、地址、创建时间和 MQTT 协议版本。
   - Session：客户端在当前会话（Session）期间的数据，断开重连后数据会实时更新。

- 订阅关系：客户端订阅的 Topic 及服务质量等级。
- 客户端事件：支持指定时间内的客户端的事件查询，例如上下线和连接等。
- 客户端消息轨迹：支持查询客户端具体某条消息的详情，例如：消息 ID、请求时间、QoS 等。



认证管理
认证管理概述

最近更新时间：2024-12-27 14:25:13

消息队列 MQTT 版提供了多种认证方式以保证服务端与客户端之间通信的安全性。当前支持四种模式：用户名+密码认证、X.509 证书认证、对接自定义第三方服务认证和外部 HTTP 认证。

- **用户名+密码认证：**用户名+密码” 认证是消息队列 MQTT 版最基础的认证方式，默认所有集群的客户端在连接服务端时都需要传输用户名（username）和密码（password）。
- **X.509 证书认证：**消息队列 MQTT 版提供了默认的服务端证书进行单向认证，即 wss（WebSockets）和 TLS 接入点。如需要使用自有证书进行认证，可以在控制台开启单向/双向认证，使用维护在腾讯云的自有证书。
- **对接自定义第三方服务认证：**腾讯云消息队列 MQTT 版支持客户对接外部 JWT 服务进行认证和授权，客户端在连接 MQTT 服务端时（发送 Connect Packet 时），如果验证通过，MQTT 服务端会进一步检查 Payload 中的声明部分(Claims)，例如 iss(Issuer)，exp(Expiration Time)，nbf(Not Before)，iat(Issued At)，aud(Audience) 等， 判断 JWT 的合法性。当 JWT 通过签名签证和 Claims 检查后，MQTT 服务端接受客户端连接请求。
- **外部 HTTP 认证：**客户端在进行连接（发送 Connect Packet）时，MQTT 使用客户端的信息（如用户名，密码等）构造 HTTP 请求，请求到达指定的 HTTP 认证服务后，MQTT 会根据该 HTTP 请求的返回结果来判断认证是否通过。如果认证通过，则允许该客户端连接服务端；如果认证不通过，则拒绝该客户端的连接。

> **说明：**
> - X.509 证书认证包含单向认证和双向认证。其中，双向认证中的 “一机一证” 场景因为其所需的额外底层算力资源及相对较高的开发要求，当前**仅专业版集群支持**。
> - 当前**仅专业版集群支持**对接自定义第三方服务认证，如 JWT 等。

用户名和密码认证

最近更新时间：2024-12-27 14:25:13

“用户名+密码” 认证是消息队列 MQTT 版最基础的认证方式，默认所有集群的客户端在连接服务端时都需要传输用户名（username）和密码（password）。

操作步骤

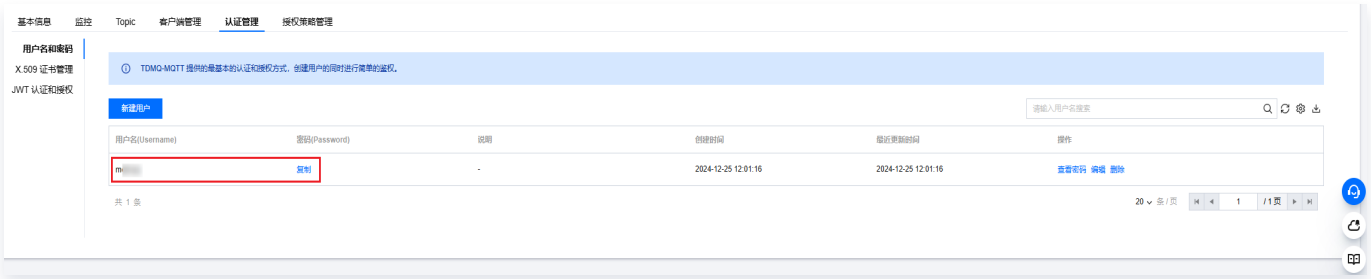
1. 登录 MQTT 控制台 。
2. 在左侧导航栏单击**资源管理** > **集群管理**，选择好地域后，单击目标集群的“ID”，进入集群基本信息页面。
3. 在集群详情页，选择**认证管理**，进入**用户名和密码**页签。
4. 单击**新建用户**，填写用户名和说明，设置密码。用户名（username）和密码（password）是 MQTT 提供的最基本的认证方式，后续使用客户端收发消息时需要填写。
   - 用户名：最长为32个字符，支持数字、大小写字母和分隔符（"_","-"）。
   - 设置密码：支持系统自动生成密码或者自定义设置密码。
   - 说明（选填）：不得超过128个字符。

5. 单击**保存**，完成用户创建，后续在权限管理列表中，可以通过以下任一种方式复制用户名和密码。

> **注意：**
> 密码泄露很可能导致您的数据泄露，请妥善保管您的密码。

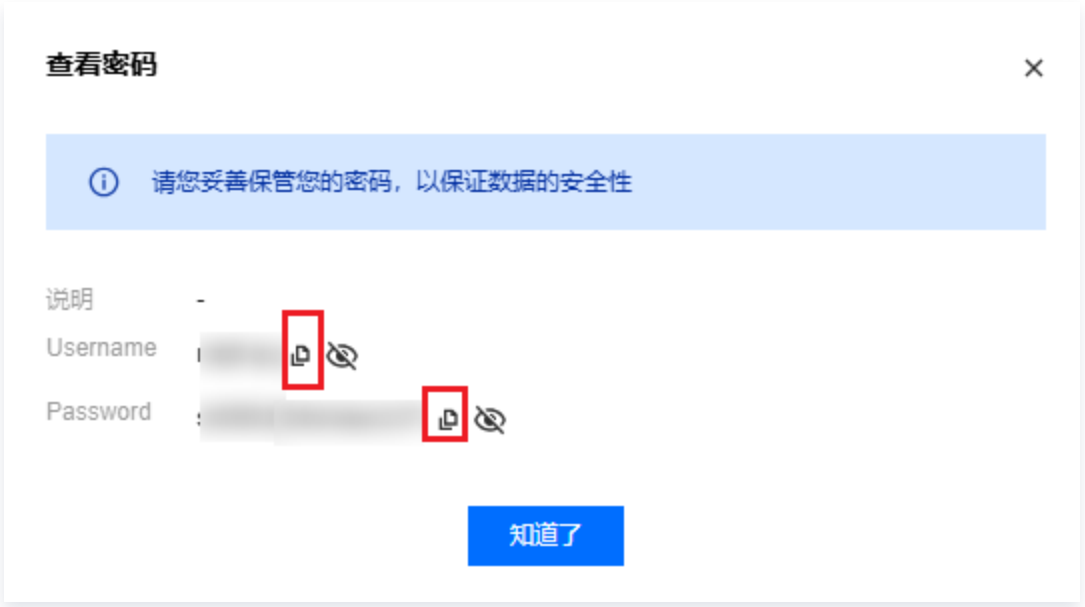
方式一：密钥列复制

在用户名(Username)、密码(Password)列复制。



方式二：操作列查看并复制

单击操作列的**查看密码**，在查看密码弹框中单击复制图标。



X.509 证书认证
单向或双向认证

最近更新时间：2024-12-27 14:25:13

为了保证集群的安全，消息队列 MQTT 版提供了默认的服务端证书进行单向认证，即 wss（WebSockets）和 TLS 接入点。如需要使用自有证书进行认证，可以按照以下指引在页面开启单向/双向认证，使用维护在腾讯云的自有证书。

单向认证

单向认证由**客户端认证服务端**，客户端对服务端的认证通过服务端证书完成。服务端会使用您选择的证书和客户端建联。服务器证书需要您自行购买或自行签发后，托管到 SSL 证书，再在 MQTT 控制台完成配置。

双向认证

双向认证指客户端与服务端之间相互认证。MQTT 通过服务端证书和客户端 CA 证书完成服务端和客户端的认证，以保证客户端和服务端通信链路的安全及可靠。MQTT 支持单向认证和双向认证两种认证方式。

- **客户端对服务端的认证** 通过 **服务端证书** 完成。
- **服务端对客户端的认证**通过 **CA 证书** 完成。客户端发起连接请求时，会将设备证书传递到服务端，服务端将根据客户端提前注册的 CA 证书验证该设备证书的正确性，验证通过则允许客户端连接服务端。

配置证书

1. 登录 MQTT 控制台 。
2. 在左侧导航栏单击**资源管理** > **集群管理**，选择好地域后，单击要配置证书的集群的“ID”，进入集群基本信息页面。
3. 在认证管理页面，进入 **X.509 证书管理**页签，点击右侧的编辑图标 ，在弹窗中完成证书配置。
    - 证书来源：消息队列 MQTT 版提供了默认的服务端证书进行单向认证，即 wss（WebSockets）和 TLS 接入点，同时也支持客户使用自定义证书进行认证。
    - 认证方式：默认支持单向认证和双向认证两种认证方式，专业版集群额外支持 “一机一证” 的模式（实际也是一种特殊的双向认证方式，具体使用详见 一机一证使用指南 ）。根据您的业务需求选择合适的认证方式，并勾选您要配置的 SSL 证书，如果现有的证书不合适，您可以参考 SSL 快速入门 申请证书。
        - 单向认证：选择单向认证后，您需要配置服务端证书。
        - 双向认证：选择双向认证后，您需要配置服务端证书和 CA 证书。
4. 单击**提交**，即可完成证书配置。

编辑证书

如果当前的证书不符合您的需求，您可以单击**认证管理 > X.509 证书管理**模块右上角的编辑图标，修改认证方式和证书配置。

注意：

- 在初次使用或者更换来自 SSL 证书的CA证书或者服务端证书时，需要主账号授权服务角色 MQTT_QCSLinkedRoleInSendSSLcertificate 以获取下载并应用 SSL 证书的功能。
- 如果在控制台更换了服务端证书和 CA 证书后，对应的客户端需要更新证书，以免因为服务端和客户端证书更新不同步导致认证报错。为了保证证书切换过程平滑，消息队列 MQTT 版在您修改完成证书后，会提供十分钟左右的过渡时间，此时间内，新旧证书的认证均会通过认证，请您抓紧时间完成老证书在客户端代码的替换。

其他认证方式

- 专业版集群额外支持 “一机一证” 的模式，具体使用详见 一机一证使用指南 。

- 专业版集群额外支持对接外部自定义认证功能，如果您还需要对接外部自定义 JWT 认证方式，请查看 使用 JWT 认证 。

自定义 X.509 证书实现 “一机一证”

最近更新时间：2025-01-08 18:06:02

使用场景

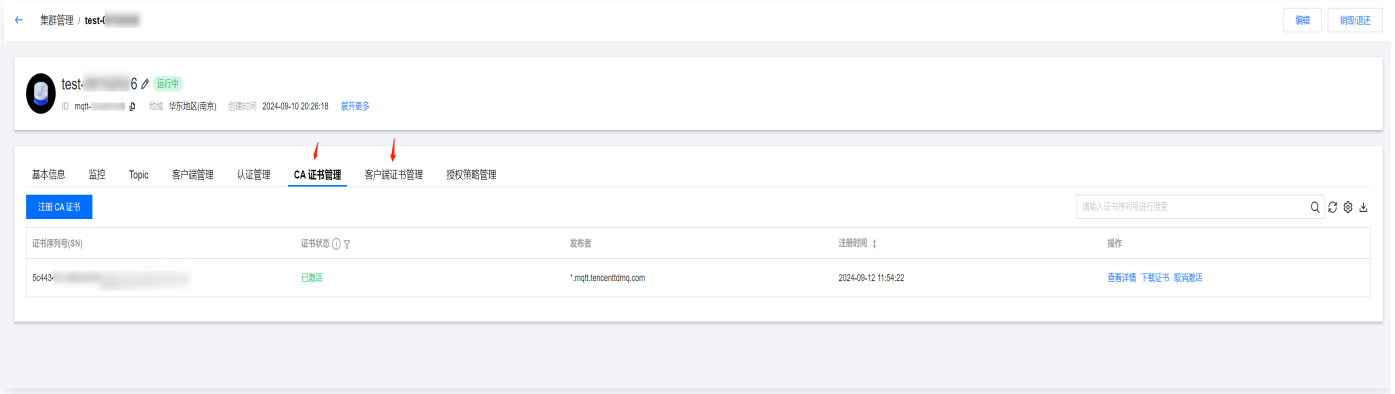
“一机一证” 实际是双向认证的一种特殊情况，每个客户端（每台设备）使用自行签发的 CA 证书及 CA 证书签发的不同的客户端证书（设备证书）进行认证。
消息队列 MQTT 版专业集群额外支持了 ”一机一证“ 的功能，您在产品的控制台上自由注册和管理设备的 CA 证书和客户端证书（设备证书），在设备出厂前，通过给每台设备烧录独特的设备证书，这样极大程度上降低了单个设备证书泄漏的影响半径。

操作步骤

配置证书注册方式

1. 登录 MQTT 控制台 。
2. 在左侧导航栏单击**资源管理** > **集群管理**，选择好地域后，单击要配置证书的集群的“ID”，进入集群基本信息页面。
3. 在**认证管理**页面，进入 **X.509 证书管理**页签，单击右侧的编辑图标 ，在弹窗中完成证书配置。
   - 证书来源：选择 “**自定义证书**” 选项。
   - 认证方式：选择 “**一机一证**” 选项。
   - 服务端证书配置：
     - 证书来源：根据您的业务需求选择合适的认证方式，当前仅支持选择来自腾讯云 **SSL证书**。
     - 授权 MQTT 服务可以下载并应用 SSL 证书能力：必选，不勾选 MQTT 服务将无法应用 SSL 证书能力。
     - 服务端证书：勾选您要配置的 SSL 证书，如果当前没有维护在腾讯云的 SSL 证书或者现有的证书不合适，您可以参见 SSL 快速入门 申请证书。
   - CA 证书来源：当前仅支持手动上传 CA 证书并注册。开启 “一机一证” 功能后，在集群详情页面 CA 证书管理页面添加 CA 证书。下文将重点介绍。
   - 客户端证书来源：支持**自动注册**和**手动注册**。
   - 默认激活状态：支持注册后自动激活和手动激活。

4. 单击**提交**，完成证书配置。
5. 完成 “一机一证” 的配置后，集群详情页会新增 “CA 证书管理” 和 “客户端证书管理” 页面，如下图所示。



手动注册 CA 证书

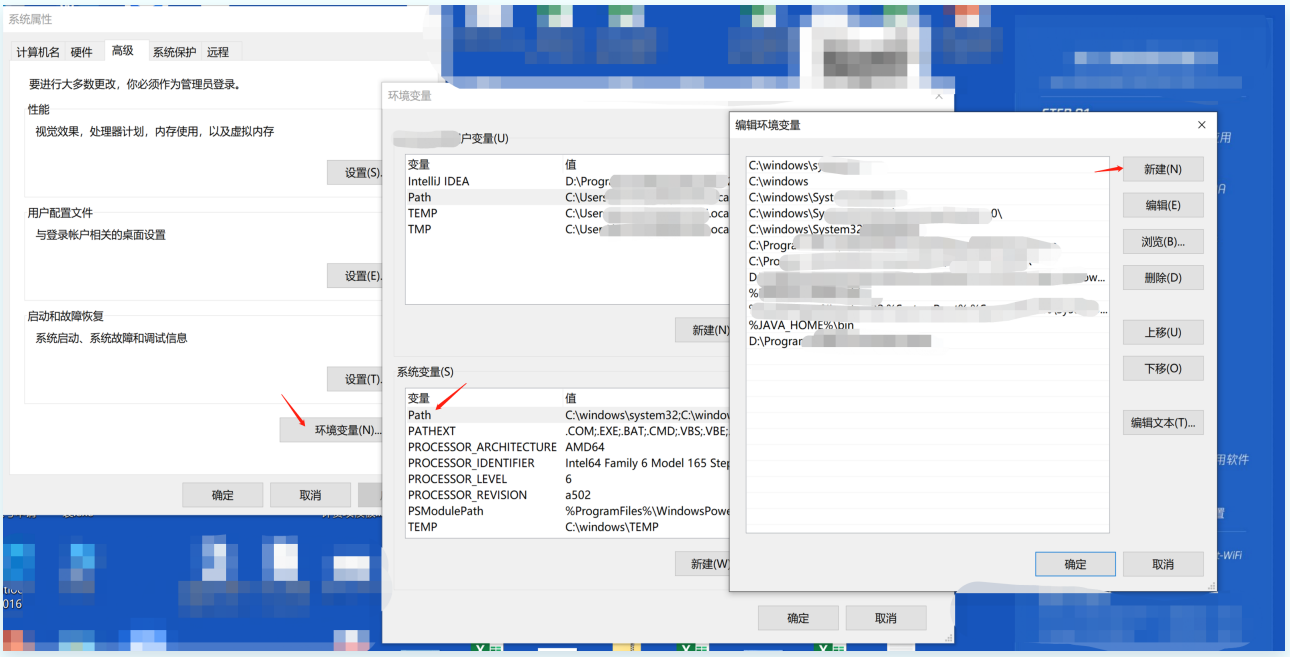
开启 “一机一证” 认证模式后，客户需要先在 “CA 证书管理” 页面手动注册 CA 证书。

签发 CA 证书

在正式注册 CA 证书之前，需要自行签发 CA 证书，以下简要介绍签发 CA 证书的步骤，如您已有完成签发的 CA 证书可以跳过当前的“签发 CA 证书”步骤。

1. 下载并安装 OpenSSL 。

> **注意：**
> 如您使用的是 Windows 系统，需要将 OpenSSL 的安装目录的 bin 子目录添加到您的系统 PATH 环境变量中，如下图所示。



2. 以下均以 RSA 算法证书为例，使用以下命令生成一个密钥对。

```
openssl genrsa -out CA.key 2048
```

3. 使用上一步生成的密钥对中的私钥（private key）生成证书签发请求文件（csr）。

```
openssl req -new -key CA.key -out CA.csr
```

4. 页面会返回以下示例，根据提示输入对应的参数。

```
Country Name (2 letter code) [AU]:
State or Province Name (full name) []:
Locality Name (for example, city) []:
Organization Name (for example, company) []:
```

```
Organizational Unit Name (for example, section) []:
Common Name (e.g. server FQDN or YOUR name) []:
Email Address []:
```

> ⚠ **注意：**
> - Common Name 字段填写的域名需要和集群的接入点域名保持一致，推荐使用 `*.mqtt.tencenttdmq.com` 。
> - 如果需要保证该证书的配置仅对当前集群生效，可以根据使用场景（公网或 VPC 网络）写入对应的域名，例如公网可以填写 `mqtt-xxxxxx-gz-public.mqtt.tencenttdmq.com` ，内网可以填写 `mqtt-xxxxxx-gz-vpce-xxxxxx.mqtt.tencenttdmq.com` 。

5. 生成自签名的 RSA 算法的 CA 证书 `CA.crt` 。

Mac

```
openssl x509 -req -extfile /System/Library/OpenSSL/openssl.cnf -
extensions v3_ca -in CA.csr -out CA.crt -signkey CA.key -CAcreateserial
-days 3650
```

Windows

```
openssl x509 -req -extfile C:\Progra~1\OpenSSL-Win64\bin\cnf\openssl.cnf
-extensions v3_ca -in CA.csr -out CA.crt -signkey CA.key -CAcreateserial
-days 3650
```

CentOS

```
openssl x509 -req -extfile /etc/pki/tls/openssl.cnf -extensions v3_ca -
in CA.csr -out CA.crt -signkey CA.key -CAcreateserial -days 3650
```

验证证书

在本地完成 CA 证书的签发后，您可以在控制台通过注册码后，生成对应的验证证书，验证证书用于验证 CA 证书本身有效性，包括证书是否由可信任的 CA 机构颁发。参考上述签发 CA 证书的流程，生成验证验证的流程如下：

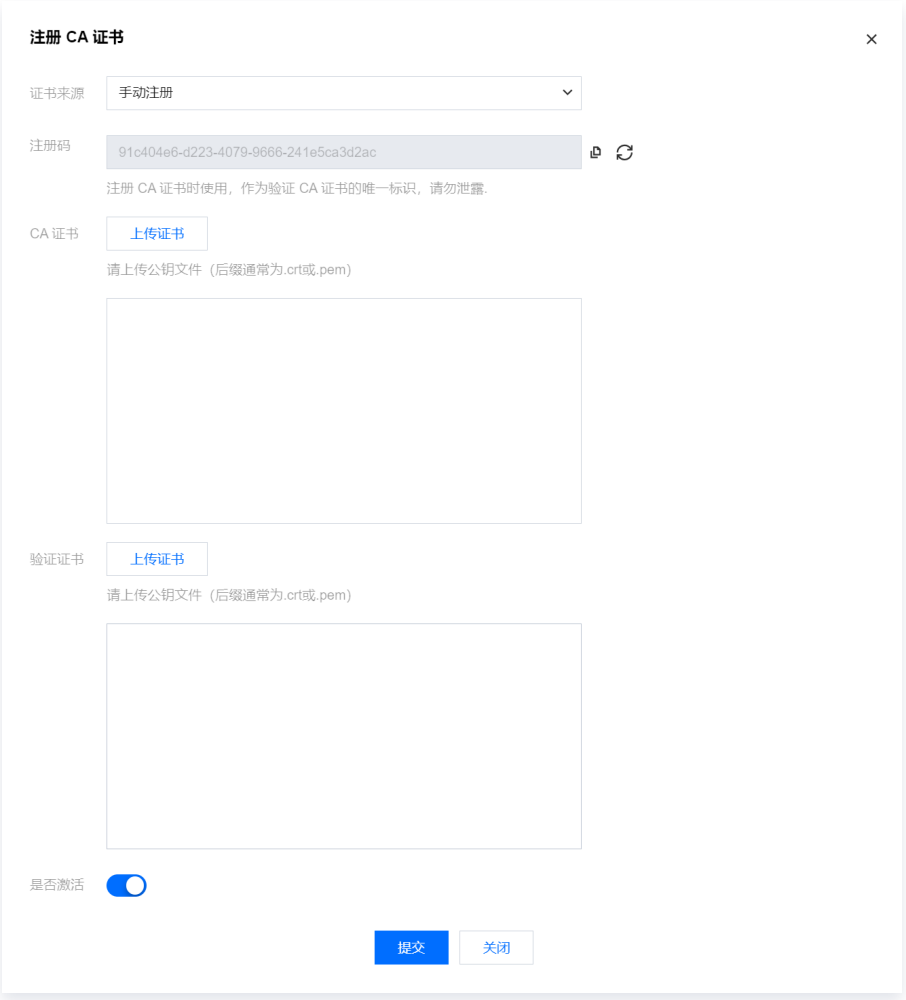
1. 以下均以 RSA 算法证书为例，使用以下命令生成验证证书的密钥对。

```
openssl genrsa -out VerificationCert.key 2048
```

2. 使用上一步生成的密钥对中的私钥（private key）生成验证证书的签发请求文件（csr）。

```
openssl req -new -key verificationCert.key -out VerificationCert.csr
```

3. 登录 MQTT 控制台 在，左侧导航栏单击 **MQTT > 集群管理**，选择好地域后，单击要配置证书的集群的“ID”，进入集群基本信息页面。在 **CA 证书管理** 页面，点击 **注册 CA 证书** 按钮，进入以下页面。复制页面上方生成的注册码，填写到下一步的 Common Name 字段。



> **注意：**
> 在本地生成并上传验证验证前，请不要关闭当前弹窗。弹窗关闭后会导致注册码重新生成，导致验证证书校验不通过。

4. 页面会返回以下示例，根据提示输入对应的参数。将上一步复制的注册码填写到 Common Name 字段中，其他字段视实际情况填写，可以为空。

```
You are about to be asked to enter information that will be
incorporated
```

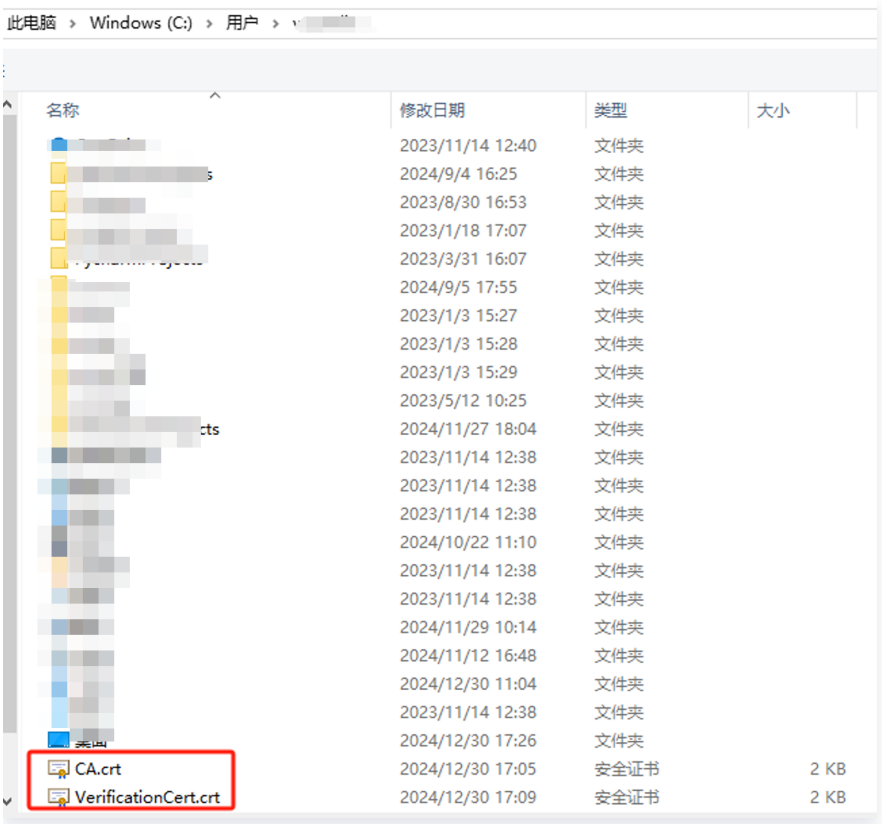
```
into your certificate request.
What you are about to enter is what is called a Distinguished Name or
a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:
    State or Province Name (full name) []:
    Locality Name (for example, city) []:
    Organization Name (for example, company) []:
    Organizational Unit Name (for example, section) []:
    Common Name (e.g. server FQDN or YOUR name)
[]:your_registration_code
    Email Address []:

    Please enter the following 'extra' attributes
    to be sent with your certificate request
    A challenge password []:
    An optional company name []:
```

5. 执行如下命令，生成CA证书的验证证书 VerificationCert.crt。

```
openssl x509 -req -in verificationCert.csr -CA CA.crt -CAkey CA.key -
CAcreateserial -out VerificationCert.crt -days 600 -sha512
```

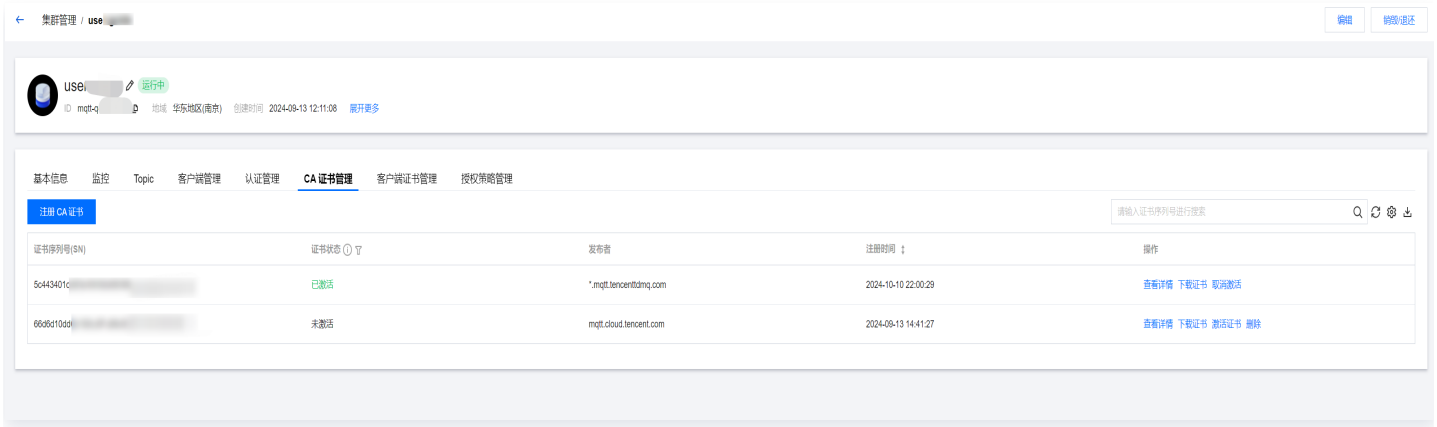
上传证书

返回到控制台弹窗，上传 CA 证书和验证证书，单击确定后提交。
上传证书后，服务端会使用验证证书中的公钥对 CA 证书的签名进行解密，并与验证证书中内容的哈希值进行比较。如果两者匹配，则表明签名有效，进而证明 CA 证书是可信的。



管理 CA 证书

CA 证书完成注册后，您可以随时在页面管理已注册的 CA 证书。
在 CA 证书的列表页，可以查看已经注册的 CA 证书的状态，CA 证书有两种状态：已激活和未激活，未激活的证书可以被删除。

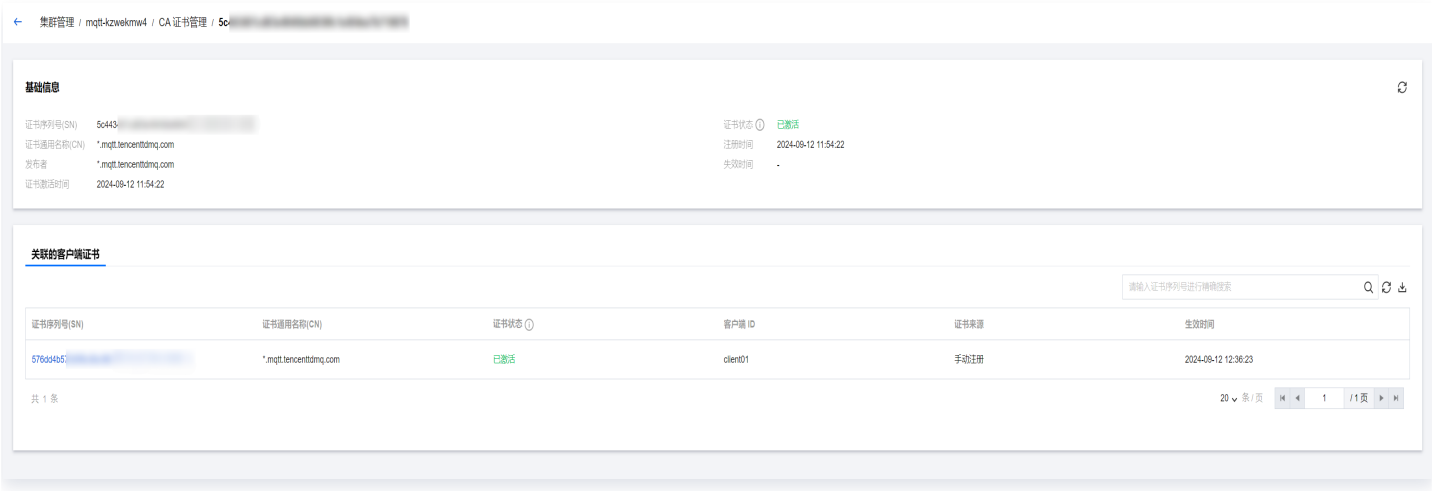


注意：

- CA 证书取消激活后，使用当前 CA 证书的客户端证书在连接时会被拒绝，因此在进行 CA 证书的状态流转时需要特别注意对于客户端连接的影响
- CA 证书在删除时，会校验当前 CA 证书下是否有处于“已激活”状态下的客户端证书（设备证书），如果有的话会禁止当前的删除 CA 证书操作；如当前 CA 证书下的客户端证书（设备证书）处于“未激

> 活”或者“已吊销”状态，则CA证书可以正常删除。

单击列表操作栏的**查看详情**，进入 CA 证书详情页，可以查看 CA 证书的相关信息。
在基础信息部分，页面展示 CA 证书的状态，Common Name，Serial Number 等信息。基础信息里的 “失效时间” 为在控制台，CA 证书取消激活的时间，并非证书失效的时间。如果证书为激活状态，则不展示“失效时间”，如下图所示。
同时，详情页还会展示关联了当前 CA 证书的客户端证书，点击客户端证书的序列号可以进入客户端证书的详情。



使用 CA 证书生成服务端/客户端证书

以下以 RSA 算法为例，简单介绍如何使用 CA 证书生成服务端或客户端证书。
如果您使用自签的 CA 证书来生成服务端和客户端证书，您可以按照以下指引来生成，服务端证书生成完成后，您可以上传到 SSL 证书 。

1. 使用如下命令创建一个名称为 `client.csr.cfg` 的文件。

```
[req]
default_bits = 2048
prompt = no
default_md = sha256
distinguished_name = dn

[dn]
C=CN
ST=SHXXX
L=SH
O=TX
OU=MQTT
```

```
emailAddress=xxx@xxx
CN=client-test
```

2. 使用如下命令创建一个名称为 `client.crt.cfg` 的文件。

```
authorityKeyIdentifier=keyid,
issuerbasicConstraints=CA:FALSE
keyUsage = digitalSignature, nonRepudiation, keyEncipherment,
dataEncipherment, keyCertSign
```

3. 生成客户端证书私钥和验证证书的签发请求文件（csr）。

Mac

```
openssl req -new -sha256 -nodes -out client.csr -newkey rsa:2048 -keyout
client.key -config <(cat client.csr.cfg)
```

Windows

```
openssl req -new -sha256 -nodes -out client.csr -newkey rsa:2048 -keyout
client.key -config client.csr.cfg
```

4. 生成客户端证书。

```
openssl x509 -req -in client.csr -CA CA.crt -CAkey CA.key -
CAcreateserial -out client.crt -days 500 -sha256 -extfile
client.crt.cfg
```

注册客户端证书

客户端证书支持 **手动注册** 和 **自动注册**（在上文提到的 “配置证书注册方式” 时指定）。

自动注册客户端证书

如果选择自动注册客户端，则客户端在和服务端连接时，服务端会校验客户端证书关联的 CA 证书是否已经注册，
如果对应的 CA 证书未注册，则认证不通过，客户端连接会被拒绝。
如果对应的 CA 证书已注册，则认证通过，客户端证书会自动出现在 “客户端证书管理” 的列表页。

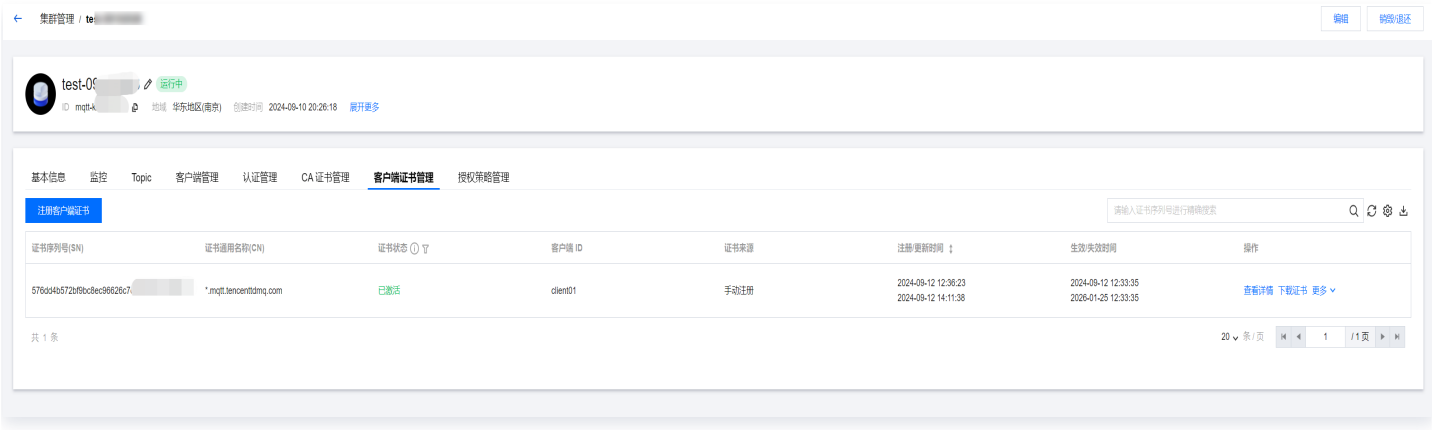
手动注册客户端证书

1. 登录 MQTT 控制台 。
2. 在左侧导航栏单击**资源管理** > **集群管理**，选择好地域后，单击要配置证书的集群的“ID”，进入集群基本信息页面。
3. 进入**客户端证书管理**页面，单击**注册客户端证书**。在以下弹窗填写客户端证书的相关信息：
   - 证书来源：手动注册。
   - CA 证书：选择已经注册了的 CA 证书。
   - 客户端证书：按照文件格式要求上传证书，见上文的 “使用 CA 证书生成客户端证书” 部分。
   - 客户端 ID：“一机一证” 场景下的补充字段，不必填，您可以根据实际场景填写。如果客户端连接时传的 client id 为空，该字段会被认为表示 client id；如果cllient id 和该字段均为空，则服务端会以客户端证书的 Common Name 字段为准。因此假设客户端没有传 client id 的情况下，请注意保证该字段不要有重复的情况。
   - 是否激活：客户端证书注册后的生效状态，默认开启。您也可以在注册完成后，在控制台手动开启。

管理客户端证书

客户端证书完成注册后，您可以随时在页面管理已注册的客户端证书。

在客户端证书的列表页，可以查看已经注册的证书的状态，客户端证书有三种状态：已激活、未激活、已吊销。已吊销的证书可以被删除。



单击列表操作栏的**查看详情**，进入客户端证书详情页，可以查看客户端证书的相关信息。

在基础信息部分，页面展示证书的状态，Common Name，Serial Number，关联的 CA 证书等信息。基础信息里的 “失效时间” 为在控制台，CA 证书取消激活的时间，并非证书失效的时间。如果证书为激活状态，则不展示“失效时间”，如下图所示。

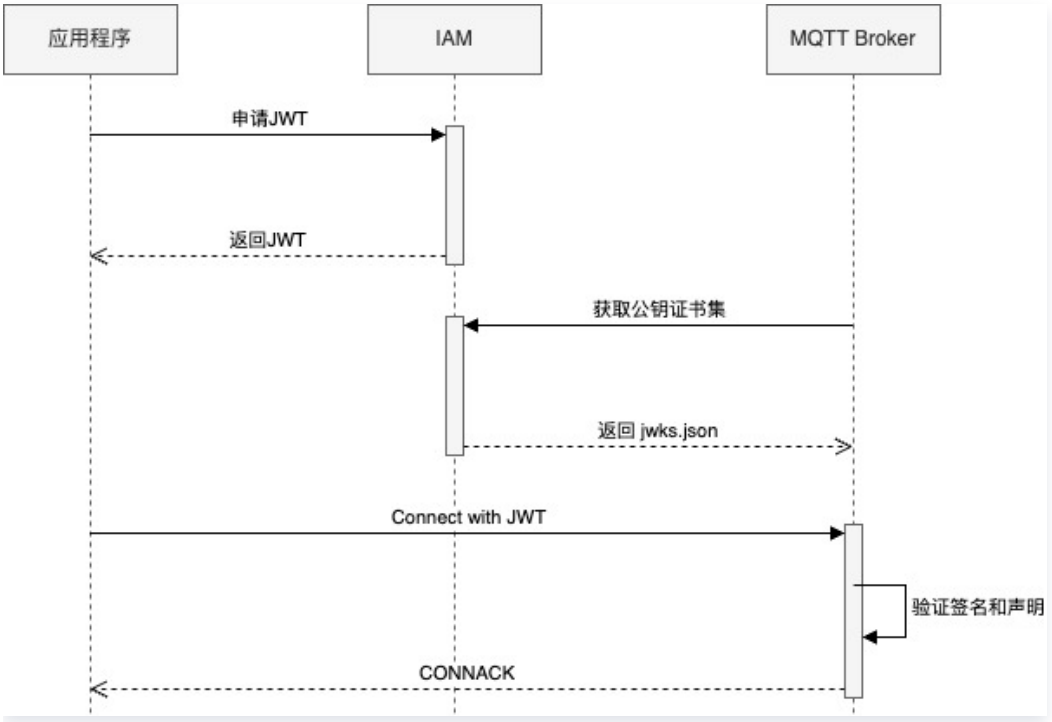
同时，详情页还会展示关联了当前 CA 证书的客户端证书，点击客户端证书的序列号可以进入客户端证书的详情。

使用 JWT 进行认证

最近更新时间：2025-01-08 18:06:02

使用场景

JSON Web Token（详细参见 RFC 7519）作为一种用于定义网络传输 JSON 对象的标准，经常被运用到基于 Token 的认证解决方案中。
整体的认证原理如下图所示：应用程序从身份识别和访问管理服务（简称 IAM）获取 JWT，客户端在连接请求中携带 JWT，MQTT Broker 将使用预先配置的签名方式（下图的示例为公钥证书集的方式，也可以使用其他签名方式）对收到的 JWT 进行验证，如果内容一致，说明在传输过程中 Token 没有被篡改，认证成功；反之认证失败。

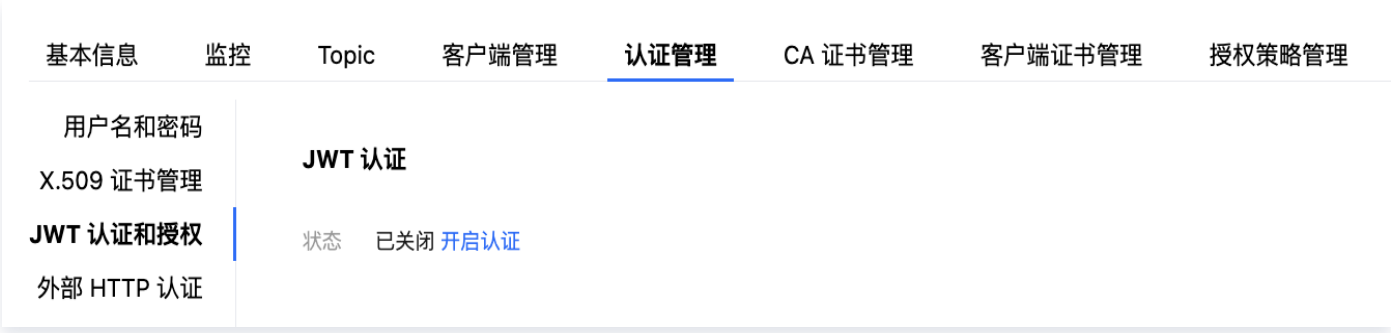


腾讯云消息队列 MQTT 版支持客户对接外部 JWT 服务进行认证和授权，客户端在连接 MQTT 服务端时（发送 Connect Packet 时），如果验证通过，MQTT 服务端会进一步检查 Payload 中的声明部分(Claims)，例如 iss(Issuer)，exp(Expiration Time)，nbf(Not Before)，iat(Issued At)，aud(Audience) 等， 判断 JWT 的合法性。
当 JWT 通过签名签证和 Claims 检查后， MQTT 服务端接受客户端连接请求。本文将指引您在控制台配置和使用 JWT 进行认证和鉴权。

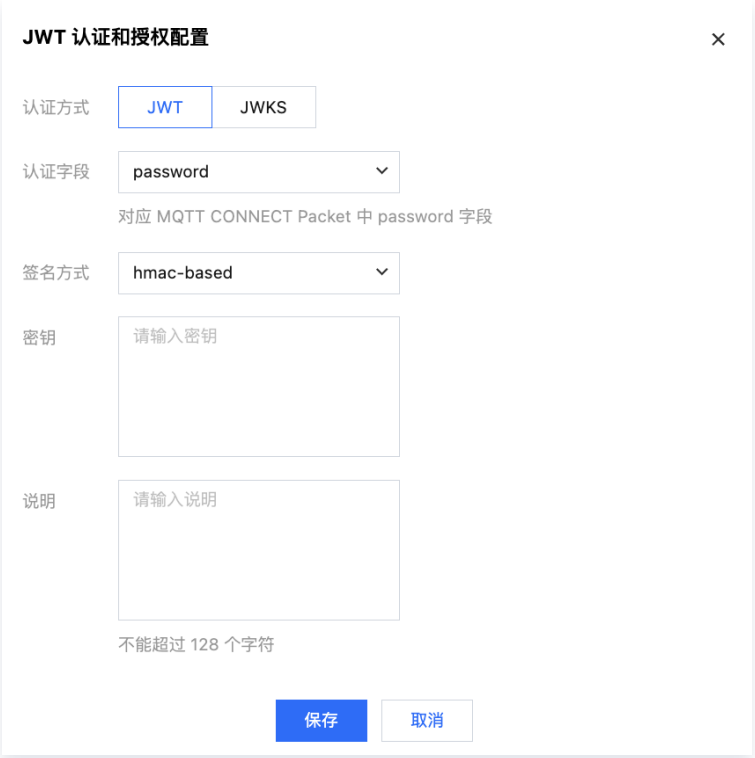
操作步骤

1. 登录 MQTT 控制台 。
2. 在左侧导航栏单击**资源管理** > **集群管理**，选择好地域后，单击要配置证书的集群的“ID”，进入集群基本信息页面。

3. 单击**认证管理**，切换至 **JWT 认证和授权**。单击**开启认证**。



4. 在弹窗内完成 JWT 认证和授权的相关配置，认证方式支持 **JWT** 和 **JWKS** 两种。



JWT 认证

根据下面说明完成配置：

- 认证字段：指定 MQTT 客户端在连接时使用 CONNECT Packet 的 User Name 或者 Password 字段携带 JWT Token。
- 签名方式：JWT Token 的签名算法，当前支持 hmac-based 和 公钥（public-key）两种方式的各类主要算法。
- 密钥：根据不同签名方式的选择，填写对应的密钥，密钥内容由客户根据加密规范自行定义。
- 说明：非必填，按需填写。

填写完成后，单击**保存**。

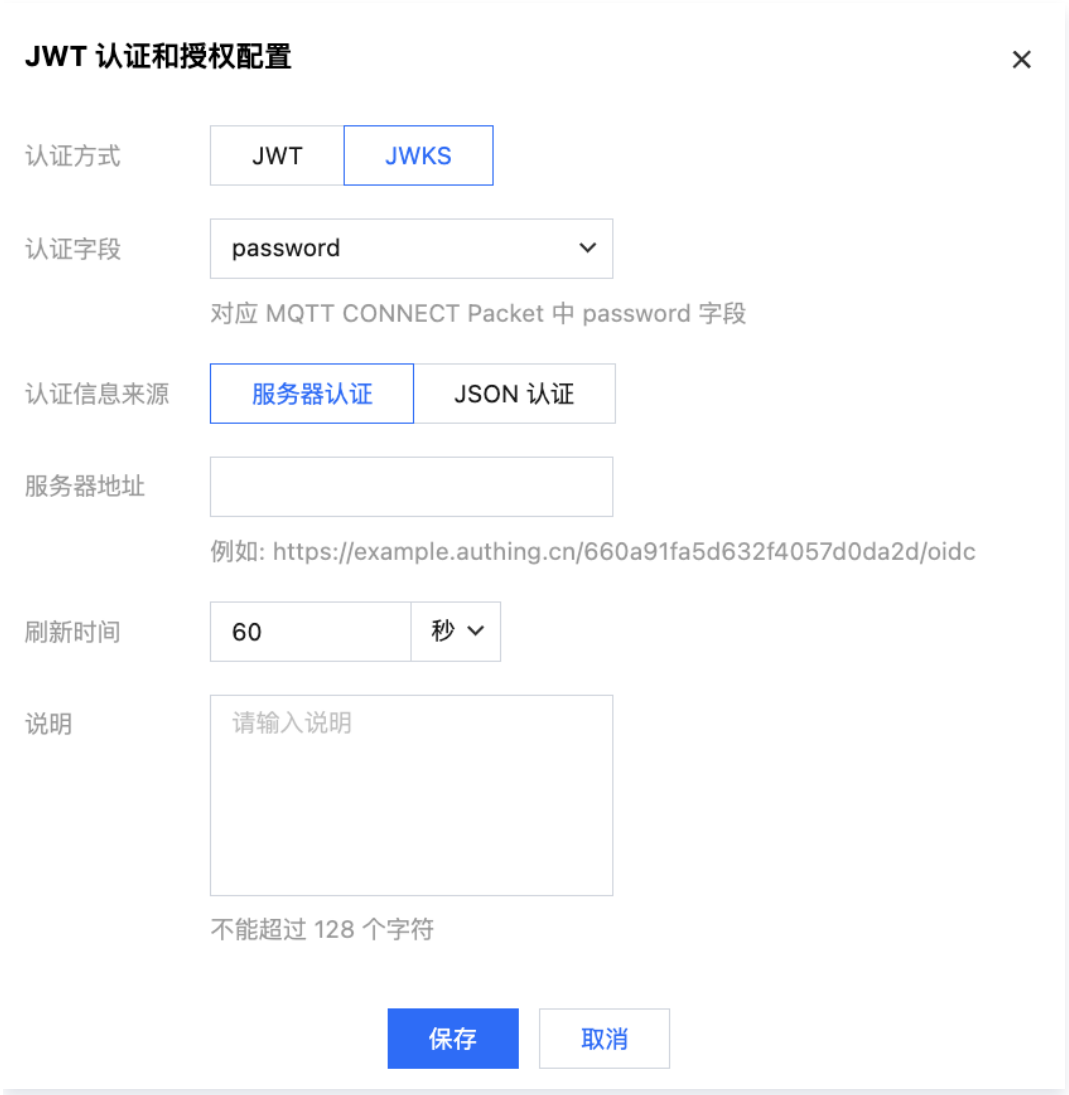
JWKS 认证

腾讯云消息队列 MQTT 版也支持客户使用 JWKS（Json Web Key Set）的形式来使用 JWT 进行认证，根据下面说明完成配置：

- 认证字段：指定 MQTT 客户端在连接时使用 CONNECT Packet 的 User Name 或者 Password 字段携带 JWT Token。
- 认证信息来源：支持 JWKS 服务器和 JSON 认证两种方式。
  - **服务器认证**：JWKS 服务器指 MQTT 服务端动态从指定的地址获取公钥集合，因此您除了填写 JWKS 服务器地址之外，还需要填写定期从服务器获取公钥集合的时间。

    当您的 IAM 提供的 JWKS 端点公网可以访问时，请选择**服务器认证**。
    - 服务器地址：IAM 服务提供 JWKS 端点地址。端点应符合 OIDC 规范，不需要包含/.well-known/jwks.json 部分。请参见您 IAM 服务提供方文档或者 Auth0 指引。
    - 刷新时间：MQTT Broker 从 IAM 服务刷新公钥集的间隔。默认值为60s。
  - **JSON 认证**：JSON 认证方式指您维护固定的公钥集合，MQTT 服务端每次根据请求时的规则选用特定的公钥集里的公钥进行加签。认证器将使用从 JWKS 端点查询到的公钥列表 exp(Expiration Time)，nbf(Not Before)，iat(Issued At)。

    当 IAM 提供方的 JWKs 端点服务网络与 MQTT Broker 网络不通时，选择 **JSON 认证**。请从 IAM 提供方下载 jwks.json 文件并将内容复制到文本框中。



注意事项

1. JWT 的 Payload 部分请勿携带敏感信息。
2. JWT 本身包含认证信息，一旦泄露，JWT 持有者将获得该令牌的所有权限。为了减少风险，JWT 有效期不宜过长。
3. 为了安全考量，JWT 的按业务场景分别授予。
4. 传输链路应该安全，应避免明文公网传输。

使用外部 HTTP 服务认证

最近更新时间：2025-01-10 11:59:32

消息队列 MQTT 版支持通过对接外部第三方的 HTTP 服务进行客户端认证和简单的鉴权。
客户端在进行连接（发送 Connect Packet）时，MQTT 使用客户端的信息（例如用户名，密码等）构造 HTTP 请求，请求到达指定的 HTTP 认证服务后，MQTT 会根据该 HTTP 请求的返回结果来判断认证是否通过。如果认证通过，则允许该客户端连接服务端；如果认证不通过，则拒绝该客户端的连接。

认证原理

当 MQTT 客户端连接到 MQTT 时，MQTT 作为请求客户端需要按照 "API" 要求的格式构造并向 HTTP 服务发起请求，而 HTTP 服务需要按照 "客户端" 的要求返回结果，HTTP 响应状态码 (HTTP Status Code) 被用于判断认证请求是否成功。HTTP 认证服务需要满足以下条件：

- HTTP 响应的编码格式 `content-type` 必须是 `application` 或者 `json` 。
- 认证结果由 `body` 中的 `result` 标示，可选 `allow` 、 `deny` 、 `ignore` 。
- 是否为超级用户由 `body` 中的 `is_superuser` 标示， 可选 `true` 、 `false` 。
- 认证结果应通过 `Status Code` `200` 或 `204` 进行返回。
- 其他响应码将被认为 HTTP 认证请求执行失败，例如 `4xx` 、 `5xx` 等。此时认证结果使用缺省值 `ignore` ，继续执行认证链。如果当前的 HTTP 认证器是链上的最后一个认证器，则认证失败，客户端将被拒绝连接。

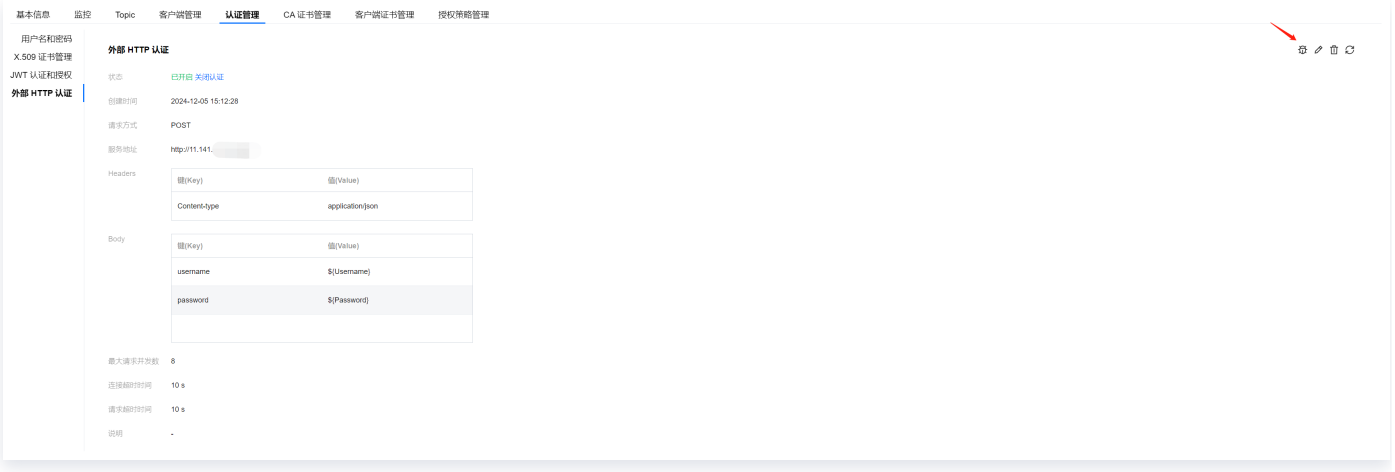
响应示例如下：

```
HTTP/1.1 200 OK
Headers: Content-Type: application/json
...
Body:
{
        "result": "allow", // "allow" | "deny" | "ignore"
        "client_attrs": {
            "role": "admin",
            "sn": "10c61f1a1f47"
        }
        "expire_at": 1654254601, // 可选，表示当前认证策略的过期时间
        "acl": // 可选，用于使用既有的HTTP认证服务支持简单的授权，授权策略将会被缓
存到MQTT服务端，再次触发时更新
        [
            {
                "effect": "allow",
                "action": ["publish","subscribe","connect"],
                "topic": "topic/AA/#",
                "qos": [1]
```

```
            },
            {
                "effect": "deny",
                "action": ["publish","connect"],
                "topic": "topic/BB"
            }
        ]
}
```

操作步骤

1. 登录 MQTT 控制台 。
2. 在左侧导航栏单击**资源管理** > **集群管理**，选择好地域后，单击目标集群的“ID”，进入集群基本信息页面。
3. 在集群详情页，选择**认证管理**，进入**外部 HTTP 认证**页签，单击**创建认证**。
   - 请求方式：选择 HTTP 请求方式，可选值：`get` ，`post` 。
   - 服务地址：请输入 HTTP(S) 服务的 URL。
   - Headers：HTTP 请求头配置，可以添加多个请求头，在填写值（value）时支持和模板变量，格式为"${变量名称}"，通过输入"${"可以触发提示。
   - Body：请求模板，在填写值（value）时支持和模板变量，格式为"${变量名称}"，通过输入"${"可以触发提示。
   - 最大请求并发数：设置最大请求并发数，范围为1-10。
   - 连接超时时间：设置连接超时等待时长，范围为1-30秒。
   - 请求超时时间：设置请求超时等待时长，范围为1-30秒。
   - 说明：填写说明信息，不能超过 128 个字符。
4. 单击**确认**完成创建。
5. 创建完成后，可以单击右上方的 进行认证参数的调试。



权限管理
数据面授权策略说明

最近更新时间：2025-01-02 17:05:32

基本概念

授权是指对 MQTT 客户端的连接（CONNECT）、发布（PUBLISH）和订阅（SUBSCRIBE）操作进行权限控制。当 MQTT 客户端连接，发布或者订阅时，MQTT 服务端查询授权数据源，将查询到的访问控制规则与要执行的操作进行匹配，根据匹配结果确定允许或者拒绝本次操作。
访问控制语句逻辑上由 Access、Actions,、Topics 和 Condition 4 部分组成。

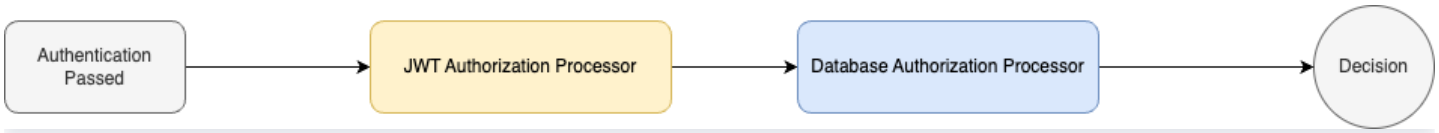
领域	字段名称	示例值	必填
决策	effect	allow/deny	是
操作	actions	["connect", "pub", "sub"]	是
资源	topics	["home/room1/*", "sensor/temperature/0"]	否
条件	clientId	"sensor*"	否
	username	"user*"	否
	qos	[0, 1, 2]	否
	retain	true/false	否
	ip	客户端IP地址: 10.0.0.1 or CIDR 10.0.0.0/16	否

MQTT 实例默认提供基于数据库的策略存储源，可通过控制台或者云 API 定义、更新、排序、删除策略。策略修变更定期同步到 MQTT 节点后生效。

工作原理

授权链

除了内置的基于数据库的访问控制源，也支持基于JWT claim定义ACL。多个授权器共同组成一个授权器链。当客户端执行操作时，按照授权器链先后进行匹配，直到获取决策结果。



每一个授权器内可定义0或多条授权策略，授权器按照定义的先后顺序，依次根据当前客户端操作(Actions)、操作资源(Topics)、以及客户端本身信息(Client ID、Username、证书信息)与策略规则进行匹配。

- 当授权器的某一条策略规则匹配时，根据规则的决策，允许或者拒绝客户端操作，授权器链匹配结束。
- 当策略规则不匹配时，尝试匹配该授权器内定义的下一条规则。
- 当该授权器所有规则都不匹配时，交由下一个授权器进行匹配。
- 当所有授权器都匹配结束仍未配到规则，默认拒绝客户端请求。

策略变量

Topic、ClientId、Username 支持以下策略变量，当授权器执行匹配操作时，会将 `${PolicyVariable}` 替换成真实值后再进行匹配。

变量名/PolicyVariable	语义
Username	MQTT 客户端连接 Username。
ClientId	MQTT Client ID。
Certificate.Subject.Country	一机一证场景下证书国家信息，详细参见 RFC4519 。
Certificate.Subject.Organization	一机一证场景下证书组织信息，详细参见 RFC4519 。
Certificate.Subject.OrganizationalUnit	一机一证场景下证书组织单元信息，详细参见 RFC4519 。
Certificate.Subject.State	一机一证场景下证书省、直辖市信息，详细参见 RFC4519 。
Certificate.Subject.CommonName	一机一证场景下证书CommonName，详细参见 RFC4519 。
Certificate.Subject.SerialNumber	一机一证场景下证书序列号，详细参见 RFC4519 。

策略通配符

Topics 字段

Topics字段支持下列通配符

通配符/Wildcard	语义
+	与 MQTT 协议 Topic Filter Wildcard 一致。
#	与 MQTT 协议 Topic Filter Wildcard 一致。
?	任何一个字符。
*	任意个字符。

ClientId、Username 字段

通配符/Wildcard	语义
?	任何一个字符。
*	任意个字符。

策略缓存

根据授权器的特性，MQTT 服务器可能会对策略规则进行缓存以加快获取策略速度。 因此，通过控制台或者云 API 变更的策略需要等待缓存更新后才能生效。

策略顺序

策略顺序会影响授权器链的最终结果，当定义多条策略规则时，请确认授权顺序符合业务要求。

策略示例

允许所有客户端操作

```
{
    "effect" : "allow",
    "actions" : [
        "connect",
        "pub",
        "sub"
    ],
    "topics" : [
        "*"
    ],
    "condition" : {
        "ip" : "0.0.0.0/0",
```

```
        "clientId" :  "" ,
        "username" :  "" ,
        "qos" :  [
            0 ,
            1 ,
            2
        ] ,
        "retain" :  [
            "true" ,
            "false"
        ]
    }
}
```

拒绝所有客户端操作

```
{
    "effect": "deny",
    "actions": [
        "connect",
        "pub",
        "sub"
    ],
    "topics": [
        "*"
    ],
    "condition": {
        "ip": "0.0.0.0/0",
        "clientId": "",
        "username": "",
        "qos": [
            0,
            1,
            2
        ],
        "retain": [
            "true",
            "false"
        ]
    }
```

```
}
```

允许所有客户端发布消息到一个 Topic

```
{
    "effect" :  "allow",
    "actions" :  [
        "connect",
        "pub"
    ],
    "topics" :  [
        "topicA/test"
    ],
    "condition" :  {
        "ip" :  "0.0.0.0/0",
        "clientId" :  "",
        "username" :  "",
        "qos" :  [
            0,
            1,
            2
        ],
        "retain" :  [
            "true",
            "false"
        ]
    }
}
```

允许所有客户端发布消息到多个 Topic

```
{
    "effect": "allow",
    "actions": [
        "connect",
        "pub"
    ],
    "topics": [
        "home/sensor", "device/1"
    ],
```

```
    "condition": {
        "ip": "0.0.0.0/0",
        "clientId": "",
        "username": "",
        "qos": [
            0,
            1,
            2
        ],
        "retain": [
            "true",
            "false"
        ]
    }
}
```

允许客户端发送到任意子 Topic

```
{
    "effect": "allow",
    "actions": [
        "connect",
        "pub"
    ],
    "topics": [
        "home/#", "device/+"
    ],
    "condition": {
        "ip": "0.0.0.0/0",
        "clientId": "",
        "username": "",
        "qos": [
            0,
            1,
            2
        ],
        "retain": [
            "true",
            "false"
        ]
    }
}
```

允许客户端发送到指定前缀、后缀 Topic

```
{
    "effect": "allow",
    "actions": [
        "connect",
        "pub"
    ],
    "topics": [
        "prefix*", "*suffix"
    ],
    "condition": {
        "ip": "0.0.0.0/0",
        "clientId": "",
        "username": "",
        "qos": [
            0,
            1,
            2
        ],
        "retain": [
            "true",
            "false"
        ]
    }
}
```

允许 client-Id 包含 username 的 client 连接

```
{
    "effect": "allow",
    "actions": [
        "connect"
    ],
    "topics": [
        "*"
    ],
    "condition": {
        "ip": "0.0.0.0/0",
        "clientId": "*${Username}*",
        "username": "",
        "qos": [
```

```
            0,
            1,
            2
        ],
        "retain": [
            "true",
            "false"
        ]
    }
}
```

拒绝 username 中包含 root 的客户端连接

```
{
    "effect": "deny",
    "actions": [
        "connect"
    ],
    "topics": [
        "*"
    ],
    "condition": {
        "ip": "0.0.0.0/0",
        "clientId": "*",
        "username": "*root*",
        "qos": [
            0,
            1,
            2
        ],
        "retain": [
            "true",
            "false"
        ]
    }
}
```

仅允许客户端 IP 在指定网段订阅消息

```
{
    "effect": "allow",
    "actions": [
```

```
            "connect", "sub"
        ],
        "topics": [
            "*"
        ],
        "condition": {
            "ip": "192.168.0.0/16",
            "clientId": "*",
            "username": "*",
            "qos": [
                0,
                1,
                2
            ],
            "retain": [
                "true",
                "false"
            ]
        }
}
```

客户端ClientId与BYOC证书CommonName匹配的客户端允许发送订阅特定消息

```
{
    "effect": "allow",
    "actions": [
        "connect", "pub", "sub"
    ],
    "topics": [
        "home/${Username}/+", "sensor/${ClientId}/#"
    ],
    "condition": {
        "ip": "192.168.0.0/16",
        "clientId": "*${Certificate.Subject.CommonName}*",
        "username": "*",
        "qos": [
            0,
            1,
            2
        ],
        "retain": [
            "true",
            "false"
```

```
            ]
        }
    }
```

操作步骤

1. 登录 MQTT 控制台 。
2. 在左侧导航栏单击**资源管理** > **集群管理**，选择好地域后，单击目标集群的“ID”，进入集群基本信息页面。
3. 在集群详情页，选择**授权策略管理**页签，单击**新建策略**，填写策略信息。
   - 策略名称：3-64个字符，支持中文、字母、数字、“-”及“_”。
   - 描述：选填，不得超过128个字符。
   - 创建方式：同时支持可视化的策略配置和 JSON 文件配置。
   - 效果(Effect)：“允许” 或者 “拒绝” 二者选一，若选择 “允许”，则表示满足配置的以下条件时，客户端的操作可以进行，若选择 “拒绝”，即满足配置的以下条件时，客户端的操作将被拒绝。
   - 操作：授权策略针对的不同的请求，包含连接（CONNECT），发送消息（PUBLISH）和 订阅消息（SUBSCRIBE），支持多选。
   - Topic：支持使用通配符和策略变量。
   - 用户名：选填，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，为空表示支持所有用户名。
   - 客户端 ID：选填，填写单条资源表达式，支持使用通配符和策略变量，为空或者 * 表示支持所有客户端。
   - IP 地址：选填，仅支持填写单个 IP（如 192.168.0.1）或 CIDR 格式（如 192.168.1.0/24）。
   - QoS：选择授权策略支持的 QoS 等级。
   - 保留消息：选择授权策略是否支持保留消息（retain message）和非保留消息。

4. 单击**创建策略**，完成创建，返回授权策略列表页。
5. 在策略列表页，可以手动调整策略鉴权顺序，服务端会按照当前的策略顺序进行鉴权，如果策略配置冲突，以排序更高的为准。

控制台权限访问管理 CAM 主账号获取访问授权

最近更新时间：2024-12-27 14:25:14

操作背景

由于 MQTT 需要访问其他云产品的 API，所以需要授权 MQTT 创建服务角色。

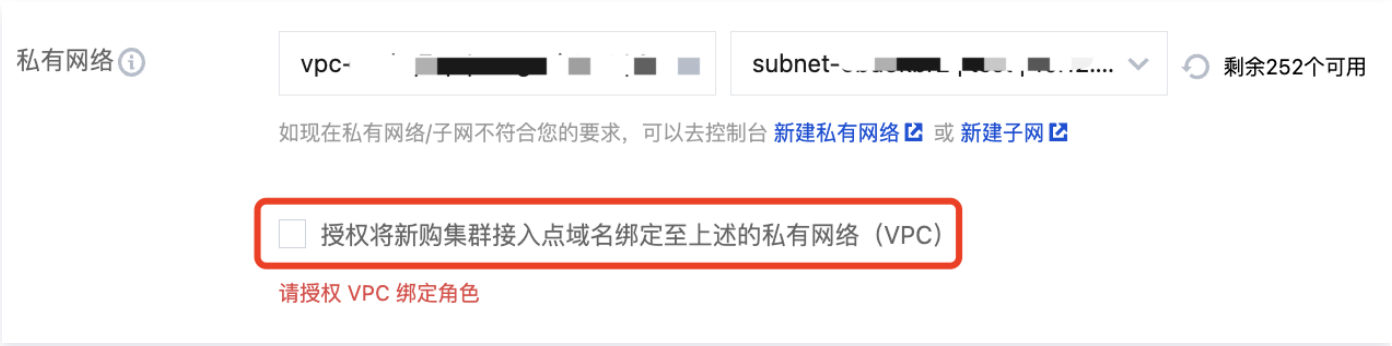
前提条件

您已 注册腾讯云账号 。

> **说明：**
> 当您注册腾讯云账号后，系统默认为您创建了一个主账号，用于快捷访问腾讯云资源。

操作步骤

1. 登录 消息队列 MQTT 版控制台 ，在左侧导航栏选择**资源管理** > **集群管理**，单击**新建集群**。
2. 在购买集群页面进行网络配置时，选择好私有网络后，勾选**授权将新购集群接入点域名绑定至上述的私有网络（VPC）**时，将会弹出需要您授权的提示弹窗。



3. 单击**前往授权**，进入访问管理控制台，单击**同意授权**，则为 MQTT 授权服务角色访问您的其他云服务资源。

4. 授权完成后，您可以继续创建 MQTT 集群并使用相关服务。

子账号获取访问授权
授予子账号访问权限

最近更新时间：2024-12-27 14:25:14

CAM 基本概念

主账号通过给子账号绑定策略实现授权，策略设置可精确到 **[API，资源，用户/用户组，允许/拒绝，条件]** 维度。

账号体系

- **主账号**：拥有腾讯云所有资源，可以任意访问其任何资源。
- **子账号**：包括子用户和协作者。
  - **子用户**： 由主账号创建，完全归属于创建该子用户的主账号。
  - **协作者**：本身拥有主账号身份，被添加作为当前主账号的协作者，则为当前主账号的子账号之一，可切换回主账号身份。
- **身份凭证**：包括登录凭证和访问证书两种，**登录凭证**指用户登录名和密码，**访问证书**指云 API 密钥（SecretId 和 SecretKey）。

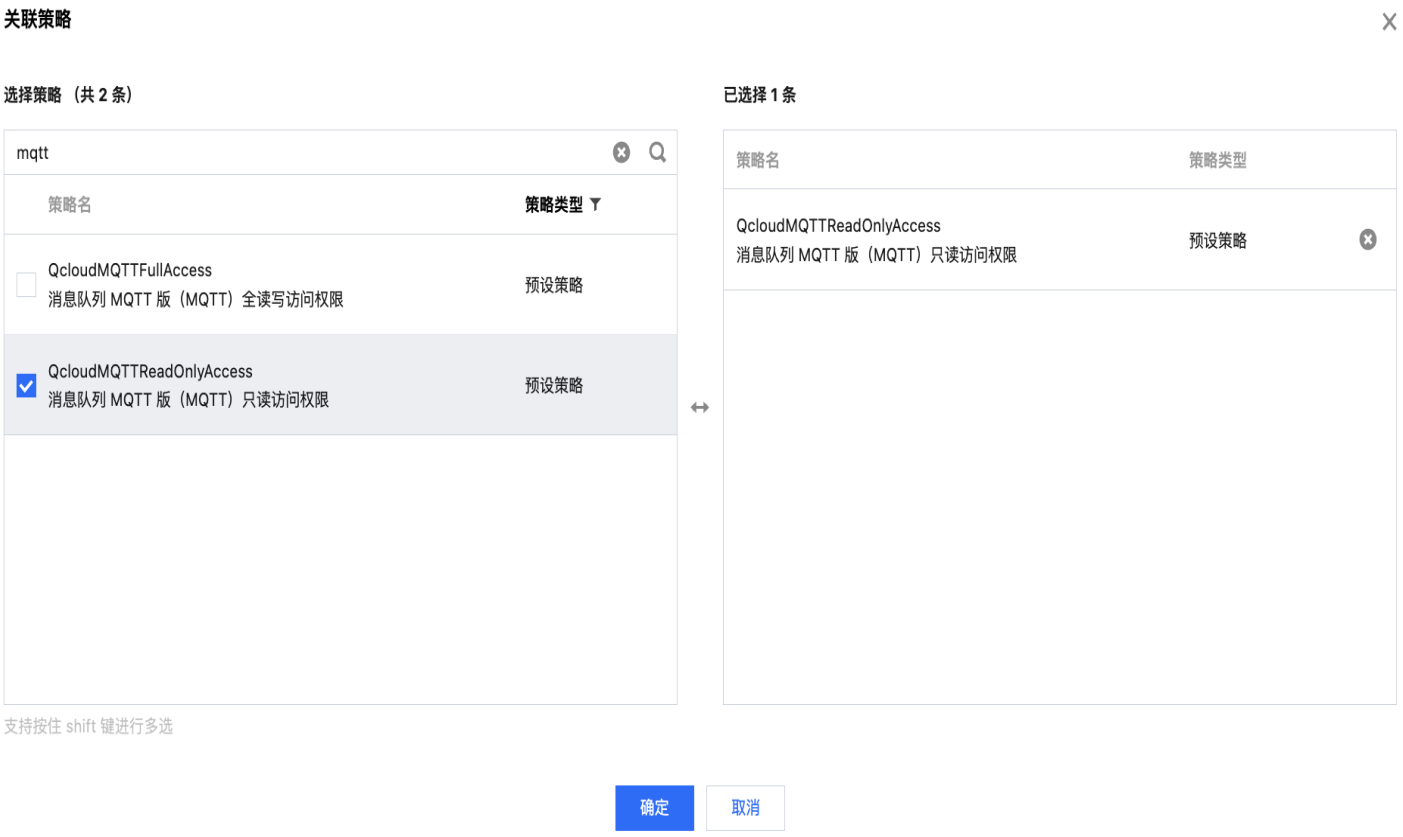
资源与权限

- **资源**：资源是云服务中被操作的对象，如一个云服务器实例、COS 存储桶、VPC 实例等。
- **权限**：权限是指允许或拒绝某些用户执行某些操作。默认情况下，**主账号拥有其名下所有资源的访问权限**，而**子账号没有主账号下任何资源的访问权限**。
- **策略**：策略是定义和描述一条或多条权限的语法规范。**主账号**通过将**策略关联**到用户/用户组完成授权。

子账号使用 MQTT

为了保证子账号能够顺利使用 MQTT，主账号需要对子账户进行授权。
主账号登录 访问管理控制台，在子账号列表中找到对应的子账号，单击操作列的**授权**。MQTT 为子账号提供了两种预设策略：

- QcloudMQTTReadOnlyAccess：仅能查看控制台的相关信息。
- QcloudMQTTFullAccess：可以在产品控制台进行读写等相关操作。



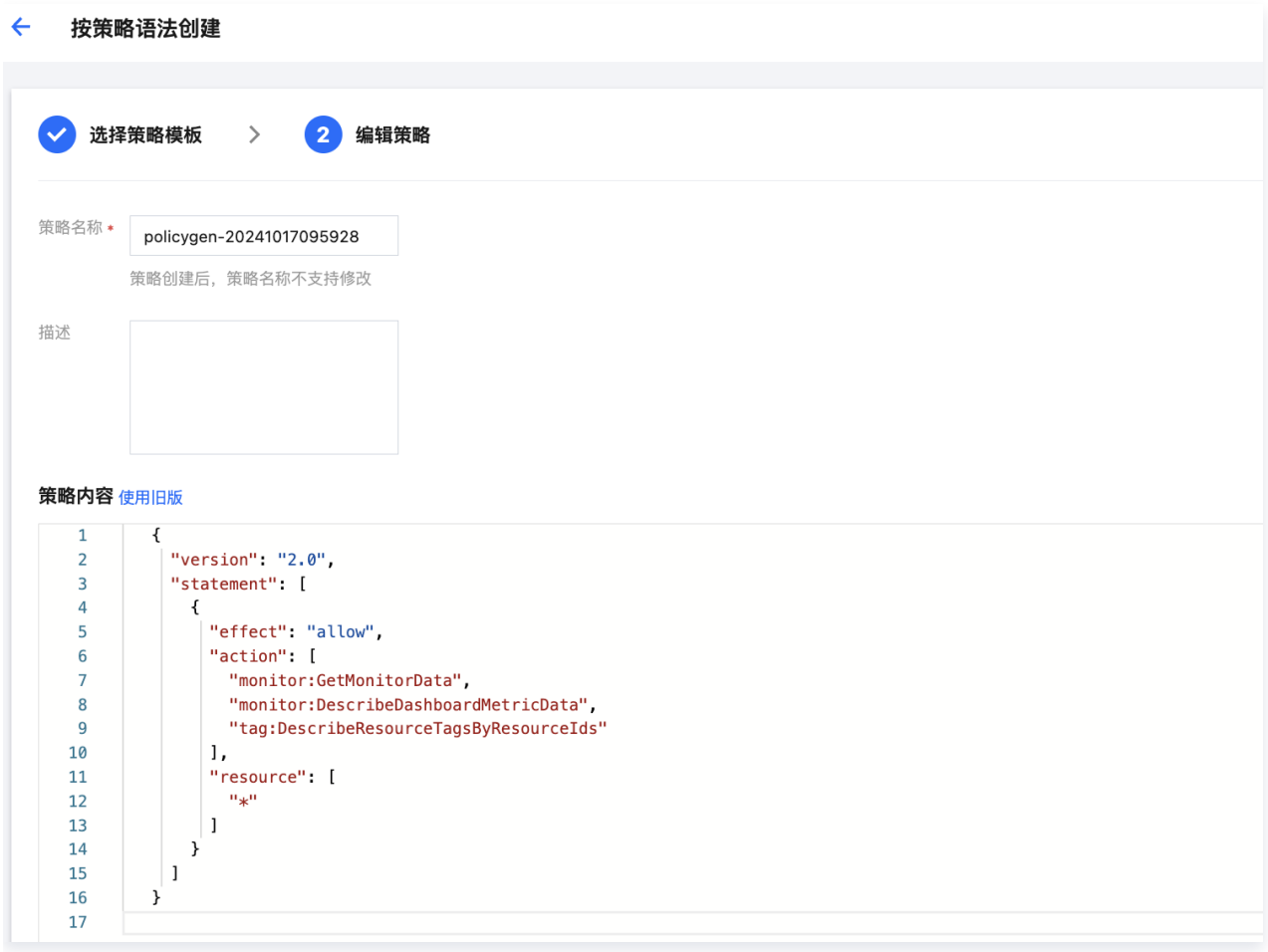
除了以上的预设策略外，为了方便使用，主账号还需要根据实际需要，授予子账号合适的其他云产品调用权限。MQTT 使用中涉及到以下云产品的相应接口权限：

云产品	接口名	接口作用	对应在 MQTT 中的作用
腾讯云可观测平台（Monitor）	GetMonitorData	查询指标监控数据	查看控制台展示的相应监控指标
腾讯云可观测平台（Monitor）	DescribeDashboardMetricData	查询指标监控数据	查看控制台展示的相应监控指标
资源标签（Tags）	DescribeResourceTagsByResourceIds	查询资源标签	查看集群的资源标签

为了给子账号增加上述权限，主账号还需要在 访问管理控制台 的策略页面，进行新建自定义策略操作。单击按策略语法创建后，选择空白模板，输入以下策略语法：

```
{
  "version": "2.0",
  "statement": [
    {
      "effect": "allow",
      "action": [
```

```
            "monitor:GetMonitorData",
            "monitor:DescribeDashboardMetricData",
            "tag:DescribeResourceTagsByResourceIds"
        ],
        "resource": [
            "*"
        ]
    }
  ]
}
```



创建完成策略后，在操作列，将创建好的策略关联给子账号即可，如下图所示：

授予子账号操作级权限

最近更新时间：2025-01-02 17:05:32

操作场景

本文指导您使用腾讯云主账号为子账号进行操作级授权，您可以根据实际需要，为子账号授予不同的读写权限。

操作步骤

授予全量读写权限

> **说明：**
> 授予子账号全量读写权限后，子账号将拥有对主账号下**所有资源**的**全读写能力**。

1. 使用主账号登录 访问管理控制台 。
2. 在左侧导航栏，单击**策略**，进入策略管理列表页。
3. 在右侧搜索栏中，输入 QcloudMQTTFullAccess 进行搜索。

4. 在搜索结果中，单击 QcloudMQTTFullAccess 的**关联用户/组/角色**，选择需要授权的子账号。

5. 单击**确定**完成授权。该策略会显示在用户的策略列表中。

授予只读权限

说明：

> 授予子账号只读权限后，子账号将拥有对主账号下**所有资源**的**只读能力**。

1. 使用主账号登录 访问管理控制台 。
2. 在左侧导航栏，单击**策略**，进入策略管理列表页。
3. 在右侧搜索栏中，输入 QcloudMQTTReadOnlyAccess 进行搜索。

4. 在搜索结果中，单击 QcloudMQTTReadOnlyAccess 的**关联用户/组**，选择需要授权的子账号。

5. 单击**确定**完成授权。该策略会显示在用户的策略列表中。


用户详情
子用户
账号ID
备注
访问方式
控制台访问、编程访问
标签
暂无标签
安全手机
更换中
安全邮箱
微信
快捷操作
消息管理
删除用户
禁用用户
快捷登录
权限
服务
组 (0)
安全
API 密钥
小程序
标签策略
权限策略
关联策略以获取策略包含的操作权限。解除策略将失去策略包含的操作权限。特别的，解除随组关联类型的策略是通过将用户从关联该策略的用户组中移出。
关联策略
解除策略
搜索策略
模拟策略
策略名
描述
关联类型
策略类型
关联时间
操作
QcloudMQTTReadOnlyAccess
消息队列 MQTT 版（MQTT）只读访问权限
直接关联
预设策略
2024-10-17 10:08:38
解除

授予子账号资源级权限

最近更新时间：2024-12-27 14:25:14

操作场景

该任务指导您使用主账号给子账号进行资源级授权，得到权限的子账号可以获得对某个资源的控制能力。

操作前提

- 拥有腾讯云主账号，且已经开通腾讯云访问管理服务。
- 主账号下至少有一个子账号，且已根据 子账号获取访问授权 完成授权。
- 至少拥有一个 MQTT 实例。

操作步骤

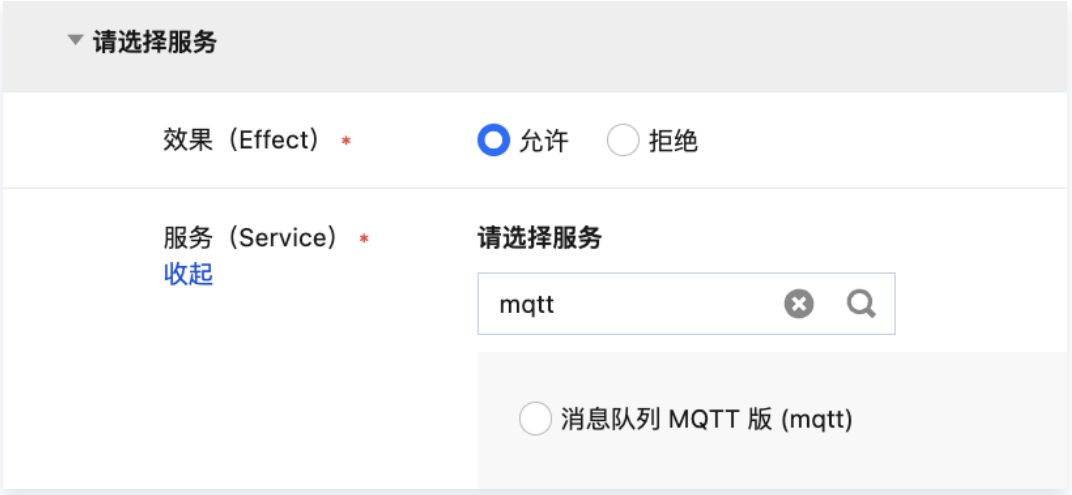
您可通过访问管理控制台的策略功能，将主账号拥有的 MQTT 资源授权给子账号，详细 **MQTT 资源授权给子账号**操作如下。本示例以授权一个集群资源给子账号为例，其他类型资源操作步骤类似。

步骤1：获取 MQTT 集群的资源 ID

使用主账号登录到 消息队列 MQTT 版控制台，在集群管理页面获取并复制集群的“ID”。

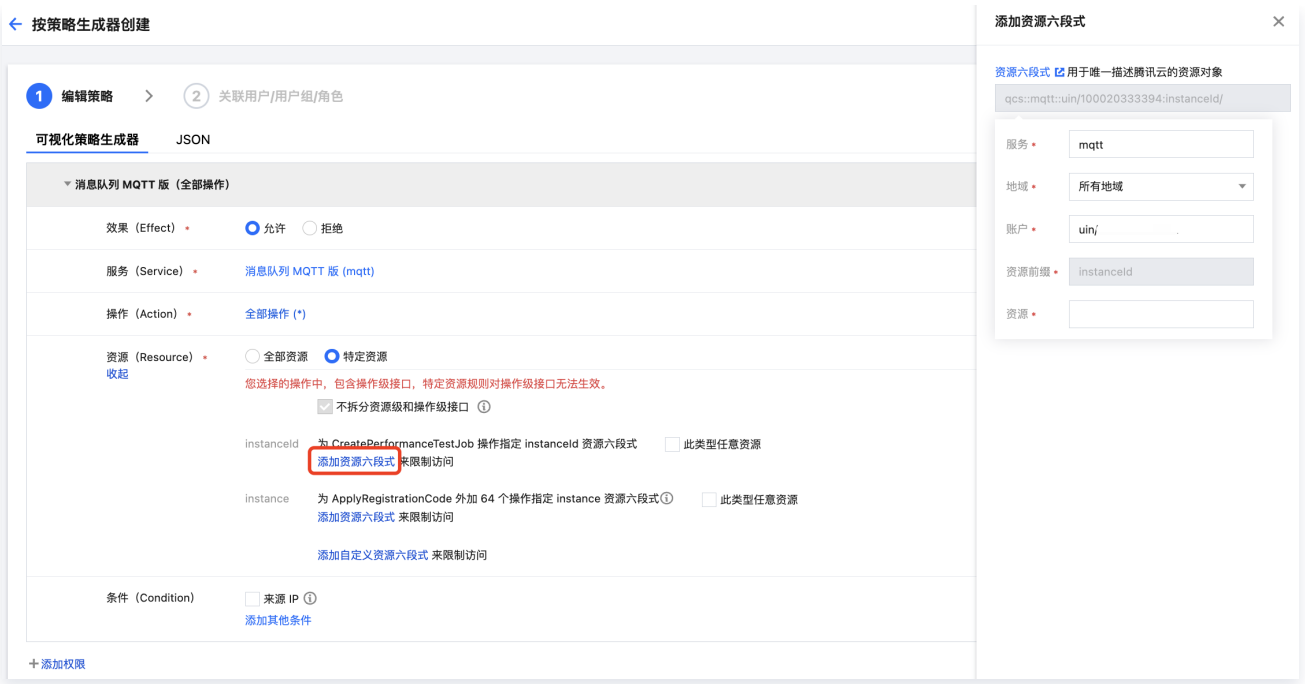
步骤2：新建授权策略

1. 进入 访问管理控制台，单击左侧导航栏的 **策略**。
2. 单击**新建自定义策略**，选择**策略生成器创建**。
3. 在可视化策略生成器中，保持**效果**为**允许**，在**服务**中输入 mqtt 进行筛选，在结果中选择**消息队列 MQTT 版 (mqtt)**。



4. 在**操作**中选择**全部操作**，您也可以根据自己的需要选择操作类型。

5. 在**资源**中选择**特定资源**，您可以勾选右侧**此类型任意资源（授权所有该类资源）**，或者并单击**添加资源六段式（授权特定资源）**。
6. 在弹出的侧边对话框中的**资源**中，填入要授权的资源的 **ID**，获取流程可参见 步骤1 。



7. 单击**下一步**，按需填写策略名称。
8. 单击**选择用户**或**选择用户组**，可选择需要授予资源权限的用户或用户组。

9. 单击**完成**，授予资源权限的子账号就拥有了访问相关资源的能力。

附录

支持资源级授权的 API 列表

MQTT 支持资源级授权，您可以指定子账号拥有特定资源的接口权限。
支持资源级授权的接口列表如下：

API 名	API 描述	资源类型	资源六段式
DescribeEvent	事件查询	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
KickOutClient	踢出客户端	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
CreateAuthorizationPolicy	创建授权策略	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
DeleteAuthorizationPolicy	删除授权策略优先级	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
DescribeConsumerClientLag	查询客户端堆积	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
DescribeTrace	轨迹查询	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
ModifyInstance	修改 MQTT 实例属性	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
DescribeDeviceCertificate	查询设备证书详情	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
DescribeCaCertificate	查询证书详情	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}

ModifyAuthorizationPolicy	修改授权策略	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
DeleteCaCertificate	删除 Ca 证书	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
DeleteInstance	删除 MQTT 实例	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
DescribeAuthenticator	查询 MQTT 认证器	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
DeleteTopic	删除 MQTT 主题	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
ModifyInstanceCertBinding	更新 MQTT 集群证书证书	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
DeleteAuthenticator	删除一个 MQTT 认证器	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
DescribeInstance	查询 MQTT 实例详情信息	instance	qcs::mqtt:${region}:uin/${uin}:instance/${InstanceId}
DescribeDeviceCertificates	查询设备证书	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
DescribeInstanceCert	查询 MQTT 集群证书列表	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
CreateUser	添加 MQTT 角色	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
RegisterDeviceCertificate	注册设备证书	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}

ModifyUser	修改 MQTT 角色	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
UpdateAuthorizationPolicyPriority	更新授权策略优先级	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
DescribeAuthorizationPolicies	查询授权策略	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
RevokedDeviceCertificate	吊销设备证书	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
DeactivateDeviceCertificate	失效设备证书证书	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
DeleteDeviceCertificate	删除设备证书	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
ActivateDeviceCertificate	生效设备证书	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
CreatePerformanceTestJob	创建 MQTT 性能测试任务	instanceId	qcs::mqtt::uin/${uin}:instanceId/${InstanceId}
DescribePerformanceTestJobMetric	查询 MQTT 性能测试结果指标数据	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
DescribePerformanceTestJobList	获取 MQTT 性能测试任务列表	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
DescribePerformanceTestJobNodes	查询 MQTT 性能测试任务执行节点信息	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
DescribePerformanceTestJob	查看 MQTT 性能测试任务	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}

TerminatePerformanceTestJob	终止 MQTT 性能测试任务	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
DescribeSubscription	查询 MQTT 首 Topic 下的客户端列表	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
DescribeMessage	查询 MQTT 消息	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
PublishMessage	发布 MQTT 消息	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
DescribeMessageList	查询 MQTT 消息列表	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
DeleteClientSubscription	为 MQTT 客户端删除一条订阅	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
AddClientSubscription	为 MQTT 客户端增加一条订阅	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
DescribeClient	查询 MQTT 客户端详情	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
CheckJWKSEndpointConnection	检查 JWKS 端点连通性	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
ModifyJWKSAuthenticator	修改 MQTTJWKS 认证器	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
CreateJWKSAuthenticator	创建一个 MQTTJWKS 认证器	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
ModifyJWTAuthenticator	修改 MQTTJWT 认证器	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}

ApplyRegistrationCode	申请注册 CA 用的注册码	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
RegisterCaCertificate	注册 CA 证书	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
DeactivateCaCertificate	失效 CA 证书	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
ActivateCaCertificate	激活 CA 证书	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
DescribeCaCertificates	查询集群 CA 证书列表	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
DescribeClientList	查询 MQTT 客户端列表	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
DescribeUserList	查询 MQTT 用户列表	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
DeleteUser	删除 MQTT 角色	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
DescribeInstanceRealtimeStats	查询 MQTT 集群实时指标	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
CreateTopic	创建 MQTT 主题	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
DescribeSubTopic	查询 MQTT 订阅子 Topic 列表	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
ModifyTopic	修改 MQTT 主题属性	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}

DescribeTopicList	查询 MQTT 主题列表	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
DescribeTopic	查询 MQTT 主题详情	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
CreateInsInternalEndpoint	为 MQTT 实例创建内网接入点	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
DeleteInsInternalEndpoint	删除 MQTT 实例的内网接入点	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
DescribeInsInternalEndpoint	查询 MQTT 实例内网接入点	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
ModifyInsPublicEndpoint	更新 MQTT 实例公网接入点	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
CreateInsPublicEndpoint	为 MQTT 实例创建公网接入点	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
DescribeInsVPCEndpoints	查询 MQTT 实例 VPC 接入点	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
DescribeInsPublicEndpoints	查询 MQTT 实例公网接入点	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
DeleteInsPublicEndpoint	删除 MQTT 实例的公网接入点	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
CreateJWTAuthenticator	创建一个 MQTTJWT 认证器	instance	qcs::mqtt::uin/${uin}:instance/${InstanceId}
DescribeInstanceList	获取 MQTT 实例列表	instance	qcs::mqtt::uin/${uin}:instance/${instanceId}

不支持资源级授权的 API 列表：

API 名	API 描述
DescribeProductSKUList	获取 MQTT 产品售卖规格
CreateInstance	购买 MQTT 实例

授予子账号标签级权限

最近更新时间：2025-01-02 17:05:32

操作场景

该任务指导您通过标签的鉴权方式，使用主账号给子账号进行某标签下资源的授权。得到权限的子账号可以获得具有相应标签下资源的控制能力。

操作前提

- 拥有腾讯云主账号，且已经开通腾讯云访问管理服务。
- 主账号下至少有一个子账号，且已根据 子账号获取访问授权 完成授权。
- 至少拥有一个 MQTT 集群资源实例。
- 至少拥有一个**标签**，若您没有，可以前往 标签控制台 > **标签列表**进行新建。

操作步骤

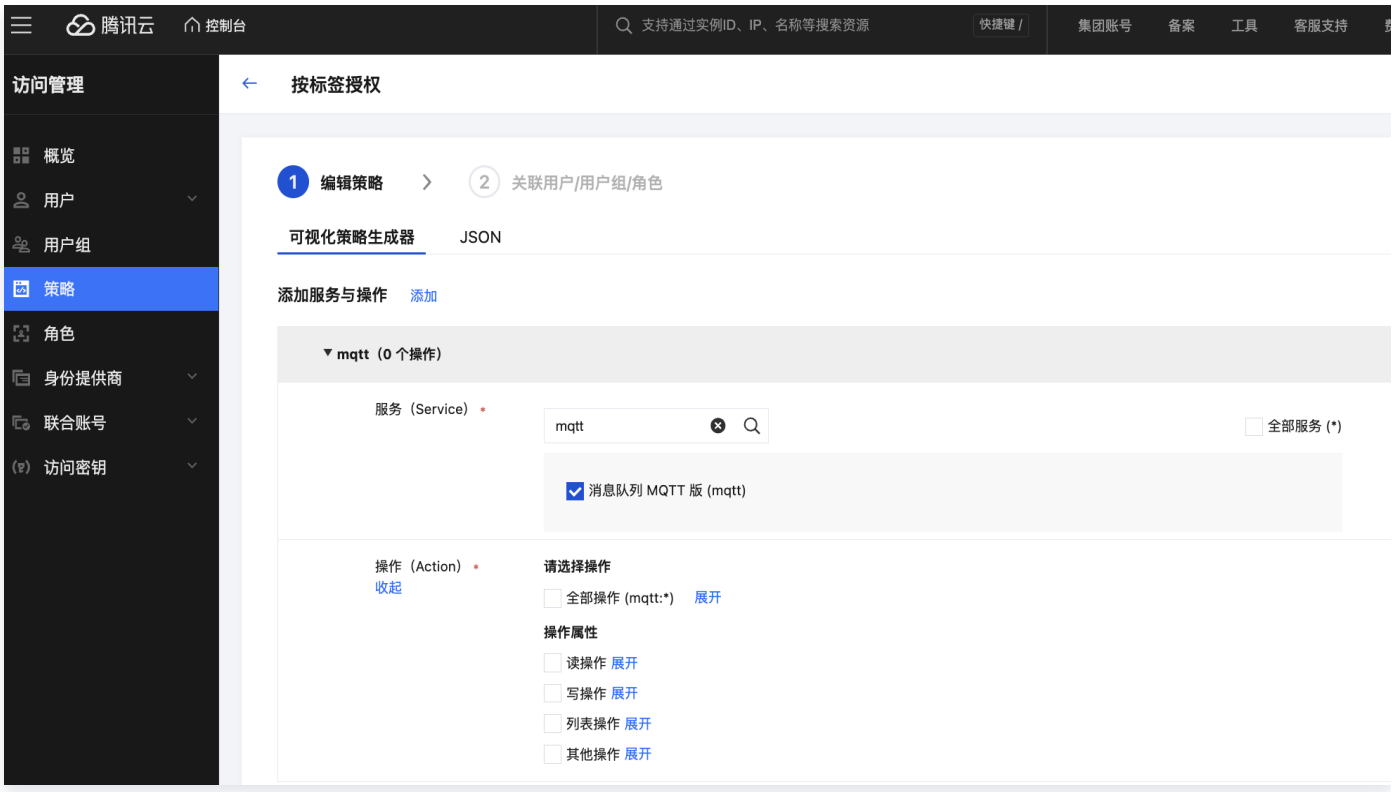
您可通过访问管理控制台的策略功能，将主账号拥有的、已经绑定标签的 MQTT 资源，通过**按标签授权**的方式授予子账号这些资源的读写权限，详细**按标签授予资源权限给子账号**的操作如下。

步骤 1：为资源绑定标签

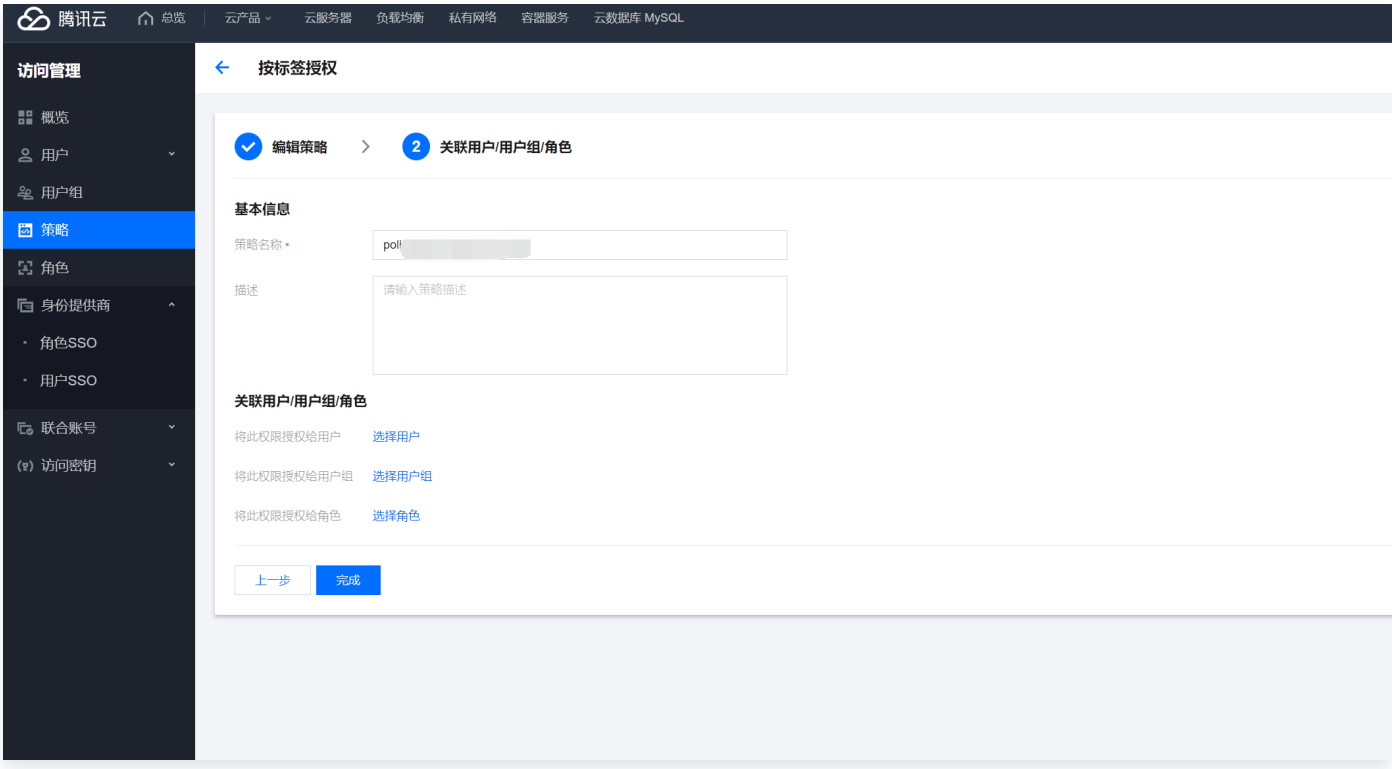
1. 使用**主账号**登录到 消息队列 MQTT 版控制台，进入集群管理页面。
2. 勾选目标集群，单击左上角的**编辑资源标签**，为集群绑定好资源标签。

步骤 2：按标签授权

1. 进入 访问管理控制台，单击左侧导航栏的 策略。
2. 单击**新建自定义策略**，选择**按标签授权**。
3. 在可视化策略生成器中，在**服务**中输入 mqtt 进行筛选，在结果中选择**消息队列 MQTT 版（mqtt）**，在**操作**中选择**全部操作**，您也可以根据需要选择相应的操作。



4. 单击**下一步**，按需填写策略名称。
5. 单击**选择用户**或**选择用户组**，可选择需要授予资源权限的用户或用户组。

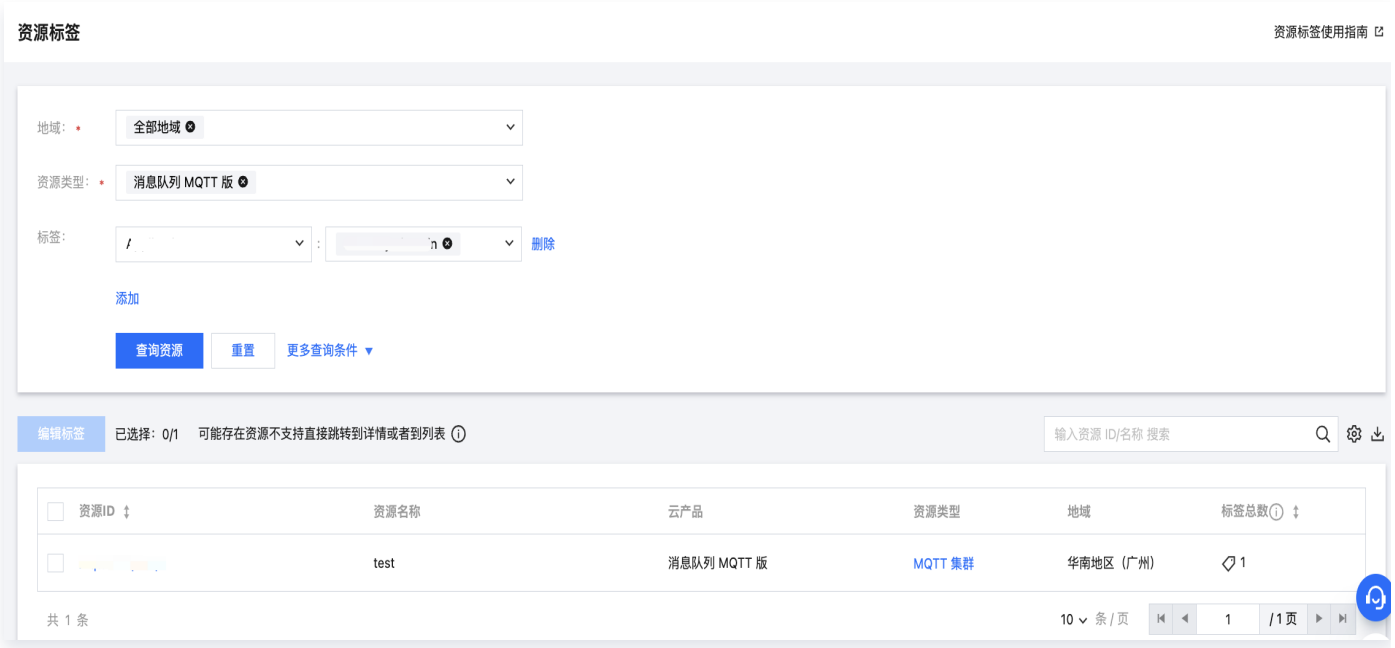


6. 单击**完成**，相关子账号就能够根据策略控制指定标签下的资源。

统一管理资源标签

您也可以在 标签控制台 统一管理资源标签，详细操作如下：

1. 登录腾讯云 标签控制台 。
2. 在左侧导航栏选择资源标签，根据需要选择查询条件，并在**资源类型**中选择 **消息队列 MQTT 版** > **MQTT 集群**。
3. 单击**查询资源**。
4. 在结果中勾选需要的资源，单击**编辑标签**，即可批量进行标签的绑定或解绑操作。



查看监控

最近更新时间：2024-12-27 14:25:13

操作场景

MQTT 支持监控您账户下创建的资源，包括集群、Topic 等，您可以根据这些监控数据，分析集群的使用情况，针对可能存在的风险及时处理。

监控指标

MQTT 当前支持的监控指标如下：

分类	监控指标	单位	指标含义	说明
集群相关指标	在线连接数量	Count	当前集群建立的客户端连接数量。	/
	在线订阅数量	Count	当前集群建立的客户端订阅数量。	/
生产消费相关指标	生产消息数量	Count	当前主题在一个统计周期内的发送消息数量。	生产消费指标支持按照 Qos 等级进行筛选。 QoS 包括以下级别： • QoS0：代表最多分发一次。 • QoS1：代表至少达到一次。 • QoS2：代表仅分发一次。
	消费消息数量	Count	当前主题在一个统计周期内的消费消息数量。	

查看监控数据

1. 登录 MQTT 控制台 。
2. 在左侧导航栏单击 **MQTT > 集群管理**，选择好地域后，单击要配置权限集群的“ID”，进入集群基本信息页面。
3. 在顶部页签选择**监控**，设置好时间范围后，可查看对应的监控数据。


基本信息
监控
Topic
客户端管理
认证管理
授权策略管理
近1小时
近24小时
近3天
近7天
2024-11-18 14:47:17 ~ 2024-11-18 15:47:17
时间粒度:
1 分钟
实时刷新:
关闭
集群相关指标
在线连接数量(Count)
在线订阅数量(Count)
1.0
0.8
0.6
0.4
0.2
15:00
15:15
15:30
mqtt-
生产消费相关指标
Topic
全部
指定 Topic
QoS
全部
生产消息数量(Count)
消费消息数量(Count)

消息查询

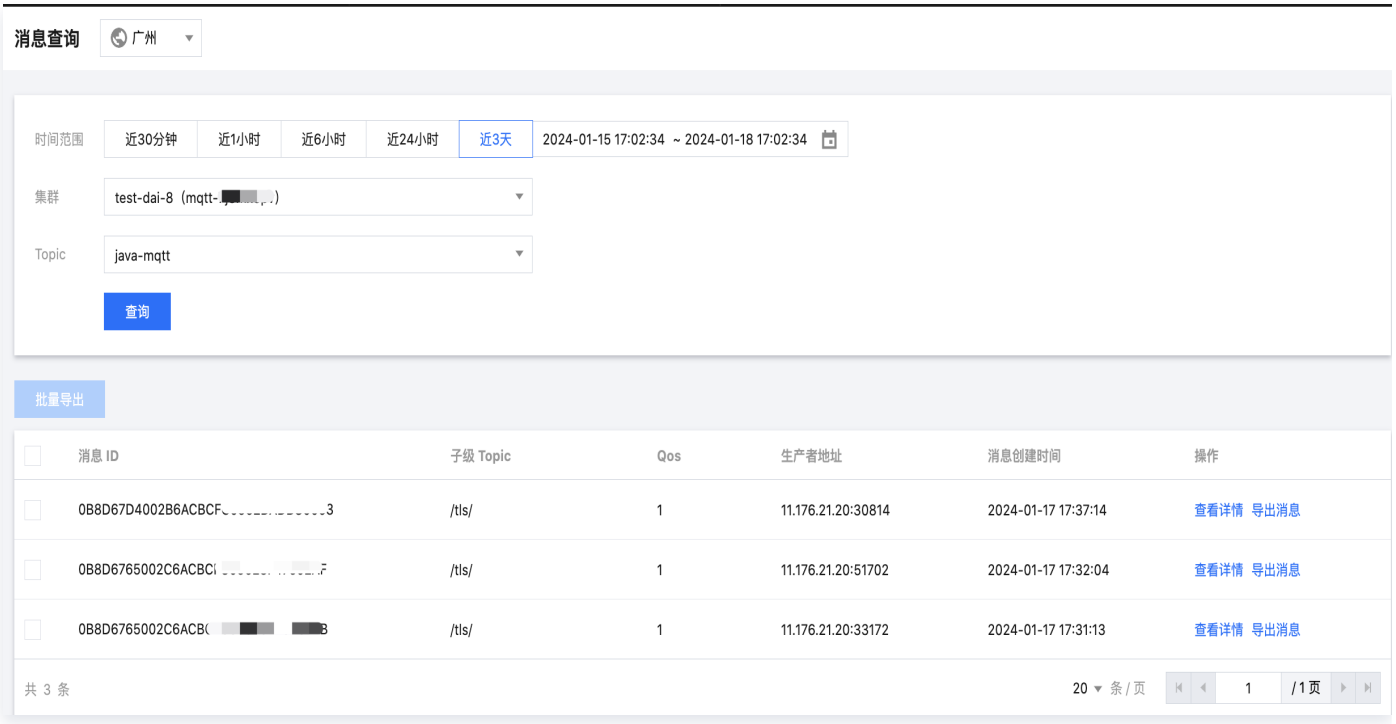
最近更新时间：2024-12-27 14:25:13

操作场景

MQTT 控制台提供消息查询功能，当您需要排查问题时，您可以按照时间维来查询具体某条消息的详情，例如：消息内容、消息参数、消息由哪个生产 IP 发送等。

查询消息

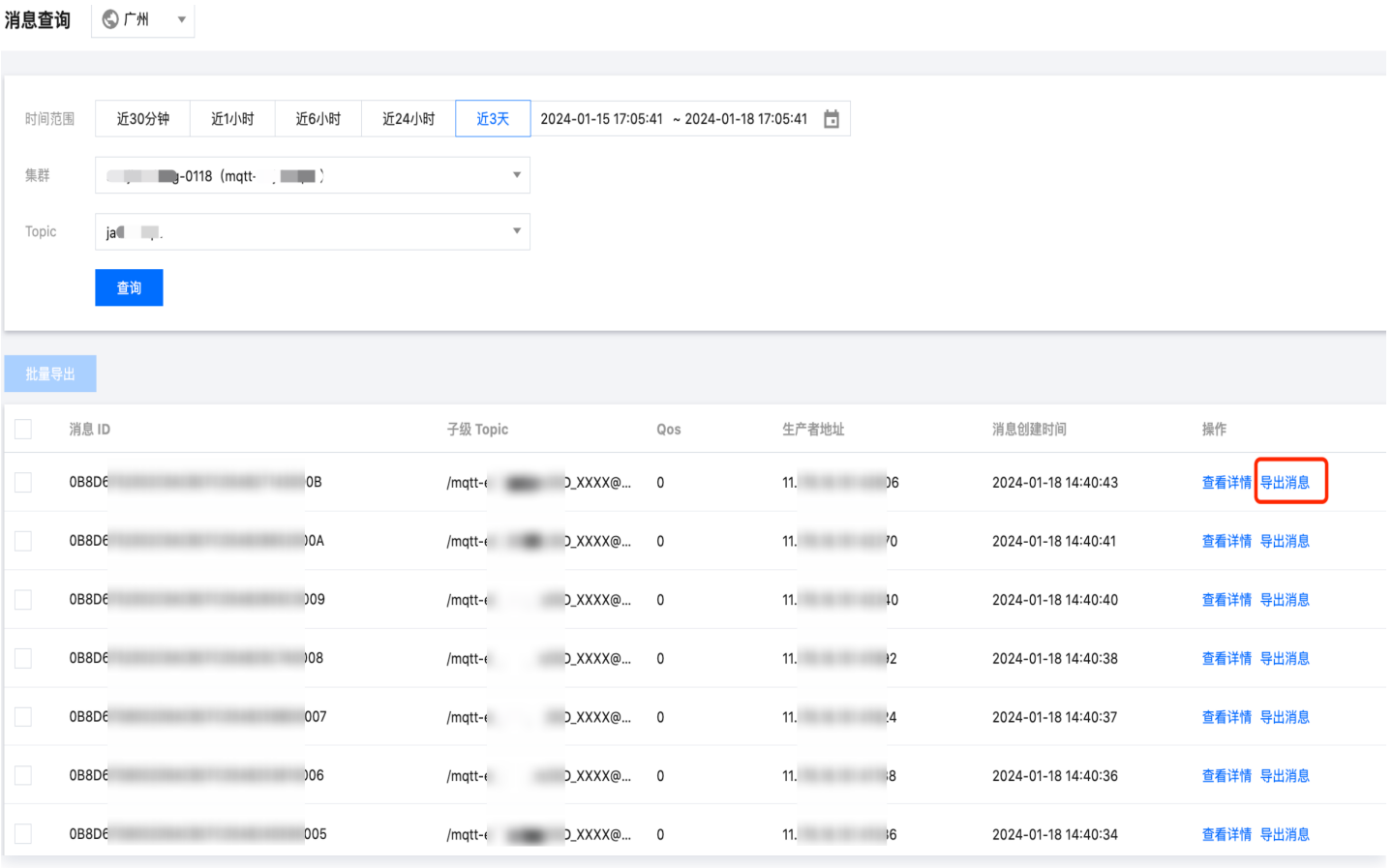
1. 登录 MQTT 控制台 。
2. 在左侧导航栏单击 **MQTT > 消息查询**，选择好地域后根据页面提示输入查询条件。
   - 时间范围：选择需要查询的时间范围，支持近30分钟，近1小时，近6小时，近24小时，近3天和自定义时间范围。
   - 当前集群：选择需要查询的 Topic 所在的集群。
   - Topic：选择需要查询的 Topic。
3. 单击**查询**，下方列表会展示所有查询到的结果并分页展示。



4. 找到您希望查看内容或参数的消息，单击操作列的**查看详情**，即可查看消息的基本信息、内容（消息体）、详情参数和消费状态。

导出消息

在查询到某条消息后，您可以单击操作列的**导出消息**将该条消息的 ID、子级 Topic、Qos、时间属性等信息导出。同时，您可以批量选择当前页面内的消息，批量导出多条消息。将消息导出到本地后，用户可以根据需要对消息进行一些处理，如复制消息体或者进行时间排序等操作。

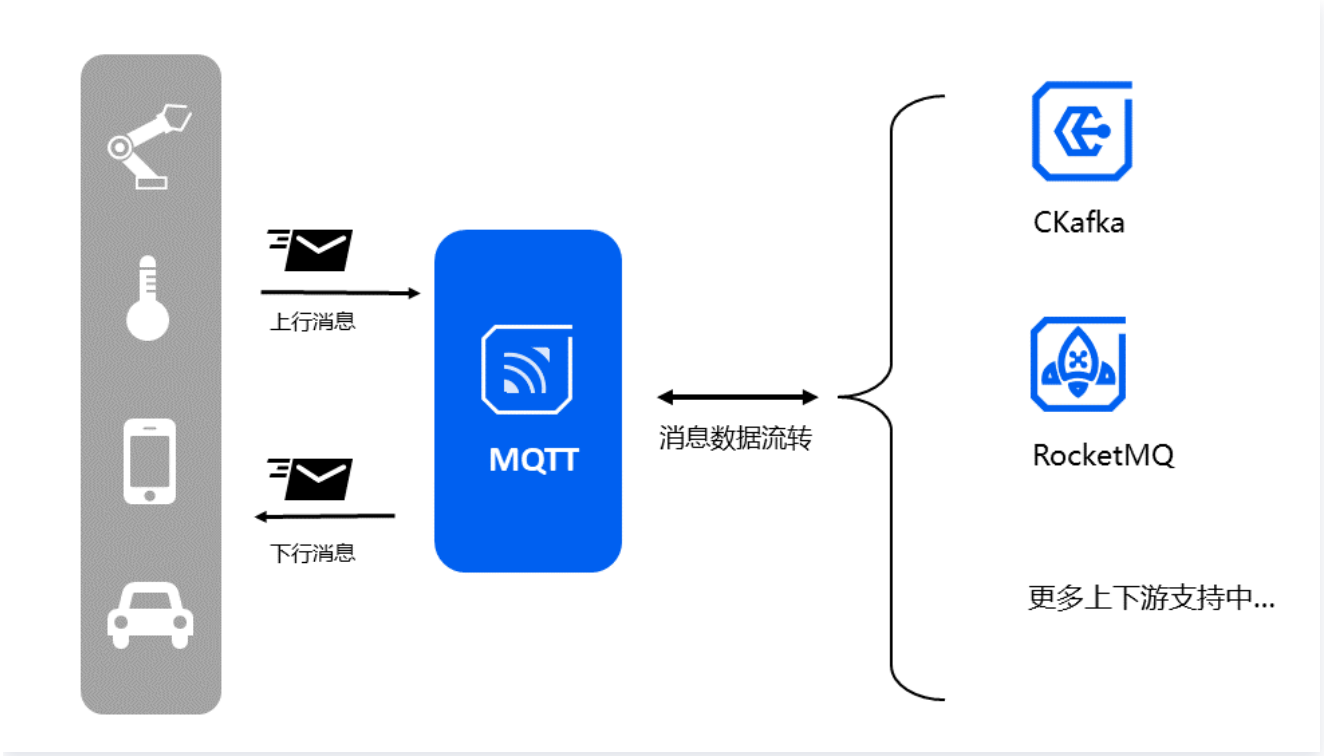

消息查询
广州
时间范围
近30分钟
近1小时
近6小时
近24小时
近3天
2024-01-15 17:05:41 ~ 2024-01-18 17:05:41
集群
Topic
查询
批量导出
消息 ID
子级 Topic
Qos
生产者地址
消息创建时间
操作
2024-01-18 14:40:43
2024-01-18 14:40:41
2024-01-18 14:40:40
2024-01-18 14:40:38
2024-01-18 14:40:37
2024-01-18 14:40:36
2024-01-18 14:40:34
查看详情
导出消息

数据互通

最近更新时间：2025-01-02 17:05:32

操作场景

TDMQ MQTT 支持将消息数据从 MQTT 集群投递到腾讯云的其他产品，从而实现数据的流通和集成，便于客户将后续的数据进行进一步存储和处理。
通过配置消息复制任务，数据在不同产品之间同步（同个地域或不同地域间）；当前已经支持 MQTT 和 消息队列 CKafka/RocketMQ 之间的消息集成和互通，后续会打通 MQTT 和更多云产品之间的数据集成通道。
您可以按照 Topic 维度，把消息队列集群 A 的某个 Topic 的消息复制到其他消息队列产品集群 B 的某个 Topic。在进行某个 Topic 的消息复制时，您可以任意启停消息复制任务，并且支持通过监控查看复制任务的进度和健康程度。



计费规则

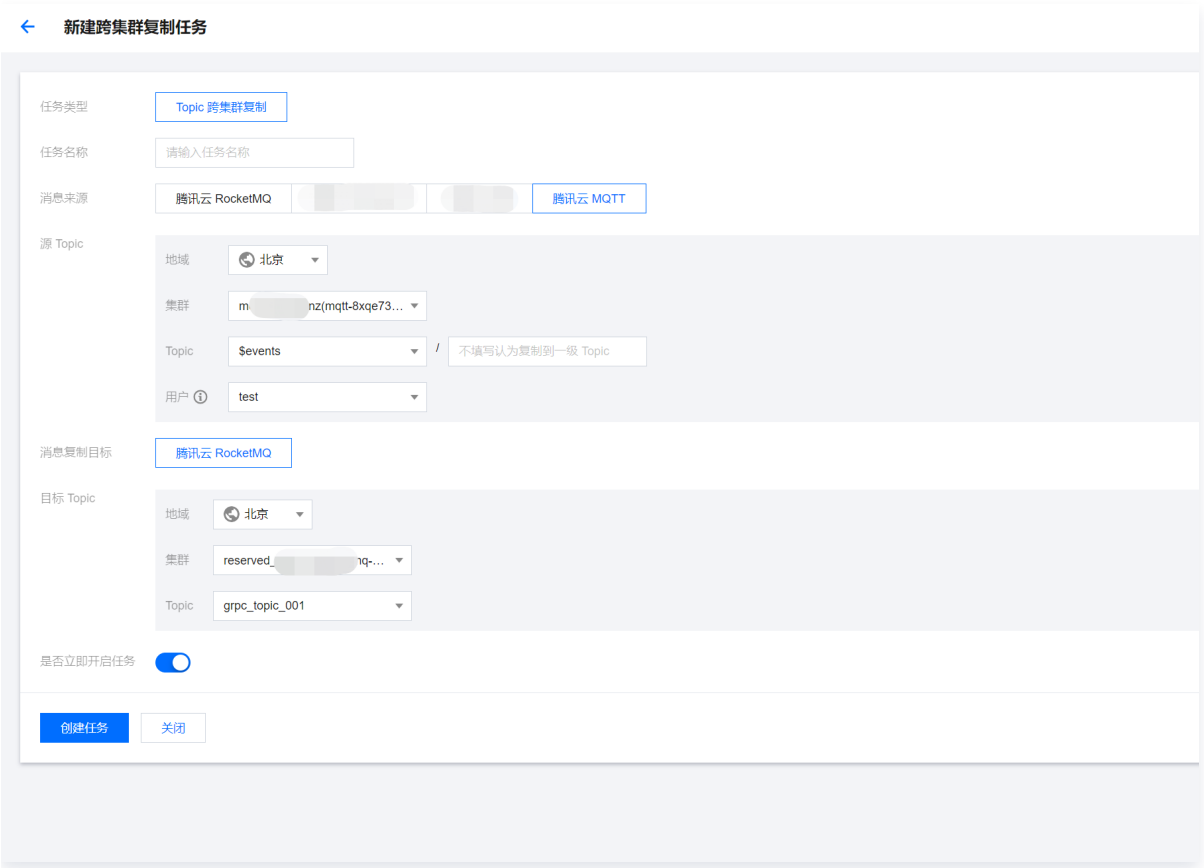
消息跨集群复制功能当前免费；在开始收费前，腾讯云会提前一个月多次通过站内信、短信和邮件等形式通知客户。复制消息产生的流量将会占用集群的 TPS 规格，因此如果有大批量的消息复制任务时，需要提前扩容集群以保证消息正常的生产和消费。

操作步骤

以下以消息从 MQTT 流转到 RocketMQ 集群为例说明操作过程，MQTT 和 CKafka 之间的消息流转可以参见 CKafka 连接器管理 部分。

创建任务

1. 登录 RocketMQ 控制台，在左侧导航栏单击**跨集群复制**。
2. 进入跨集群复制页面，单击页面上方的 **新建任务**， 按照要求填写以下字段：
   - 任务名称：3-64个字符，只能包含数字、字母、“-”和“_”。
   - 消息来源：选择腾讯云 MQTT。
   - 源 Topic：通过下拉依次选择地域、集群、 Topic 和用户，如果找不到需要的集群或 Topic 可以在集群列表页进行新建。MQTT 集群的 Topic 还支持系统 Topic（如默认事件的主题 $events/+ 等）。用户为需要使用具备消费消息的角色进行复制任务，为保证任务顺利进行，**请保证选择的用户角色有权限进行消息的消费和生产**（已配置了对应的策略）。
   - 目标 Topic：通过下拉依次选择地域、集群和 Topic，如果找不到需要的集群或 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。

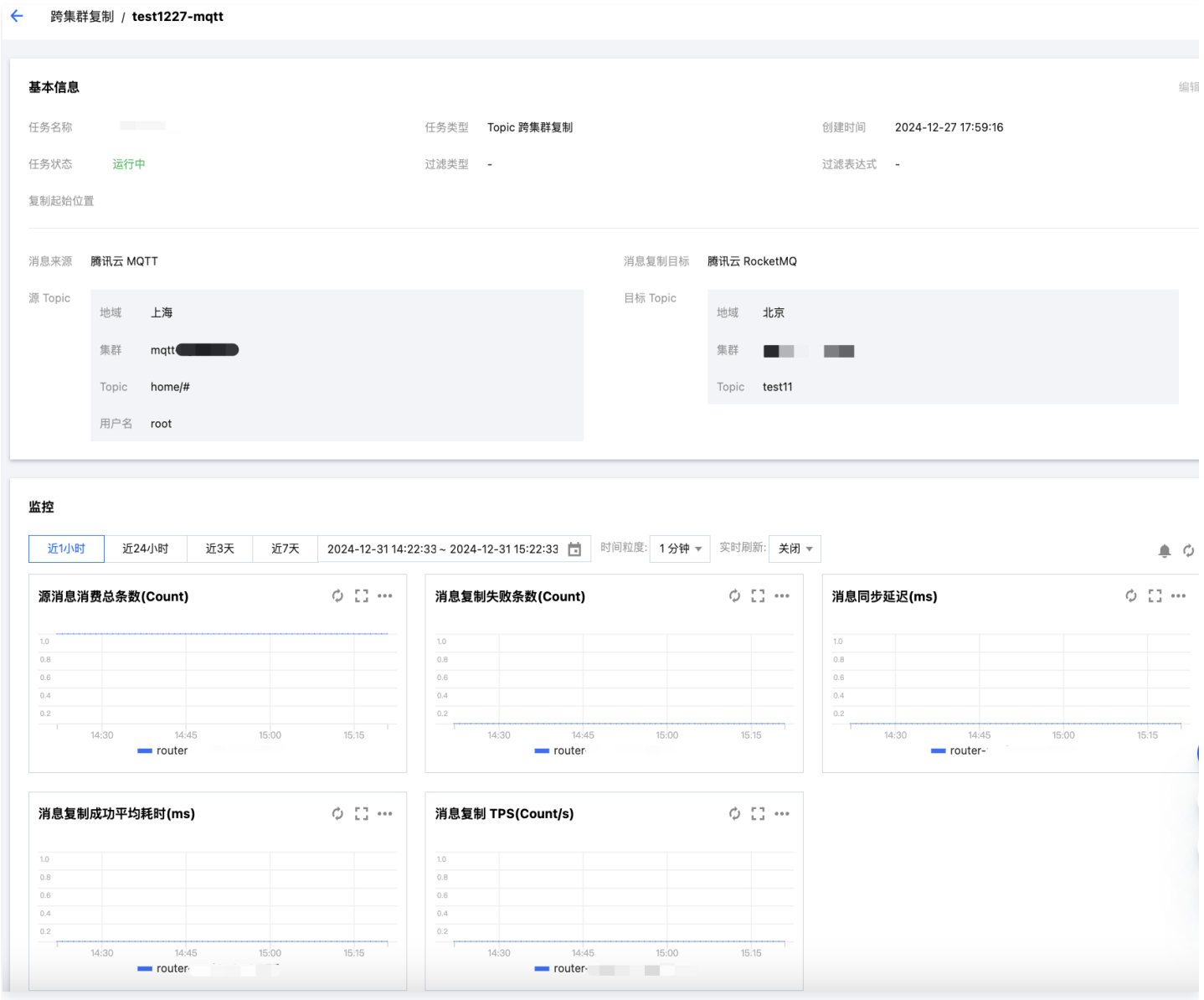


单击**创建任务**后，会跳转到任务列表页，在任务初始化后即创建完成。
您创建的复制任务是单向的，即如果您创建一个 Topic A 到 Topic B 的复制任务，Topic A 的消息会自动复制到 Topic B；如果您需要双向的复制任务，您需要再次新建一个从 Topic B 到 Topic A 的复制任务。

查看任务详情

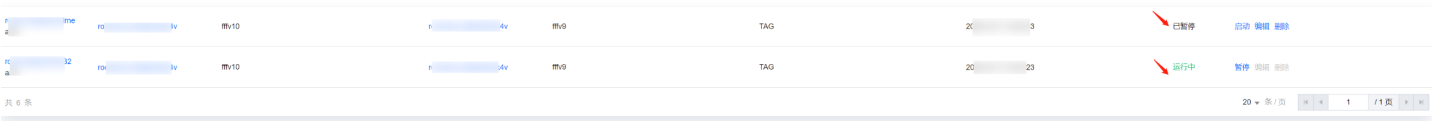
在任务创建完成后，您可以在任务的列表页看到新增的复制任务，同时可以快速查看任务的状态。单击操作列的**启动/暂停**可以快速的开启和暂停任务。

进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。
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，；例如过滤规则和起始时间等等。 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。



异常处理

正常情况下，状态栏会展示 “运行中” 或者 “已暂停” 的状态；如果状态为 “启动失败”，您需要检查任务运行状态和任务详细配置是否正确，例如 SQL 表达式是否正确等；鼠标悬浮在失败状态上会有具体的失败原因。



如果任务状态失败，您可以单击操作栏的**编辑**，或者进入任务详情页，单击“基本信息”的**编辑**，重新更正任务的信息。