消息队列 MQTT 版

规则引擎（Beta）



腾讯云

版权所有：腾讯云计算（北京）有限责任公司

【版权声明】

©2013-2026 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【商标声明】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【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

规则引擎（Beta）
- 规则引擎概述
- 管理连接
- 管理规则
- SQL 参考
  - 数据来源与字段
  - SQL 语法与示例
- 数据结构
  - JSON Schema
- 输出动作
  - 输出到 CKafka
  - 输出到 MQTT
  - 输出到云函数 SCF
  - 动态字段
- 监控与排障
- 使用限制

版权所有：腾讯云计算（北京）有限责任公司

规则引擎（Beta）
规则引擎概述

最近更新时间：2026-11-18 15:35:00

> **说明：**
> 规则引擎目前处于 Beta 公测阶段，公测期间免费使用。详情请参见下文 Beta 公测说明 。

规则引擎是消息队列 MQTT 版内置的数据处理组件。客户端发布的消息和客户端产生的事件进入集群后，规则引擎使用 SQL 对数据进行提取、过滤和转换，并将结果输出到 CKafka、MQTT 主题或云函数 SCF。全程无需编写代码，也无需自行部署和运维转发服务。

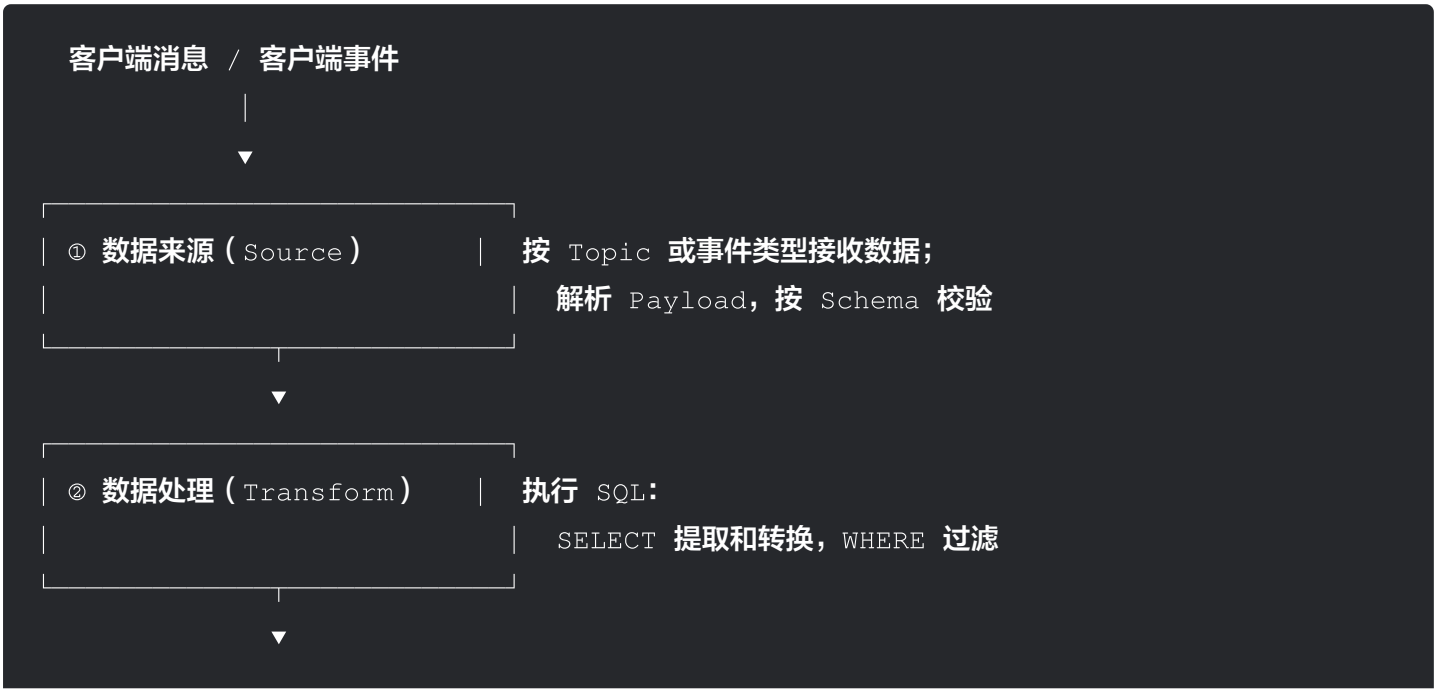
功能说明

借助规则引擎，您可以：

- **按条件筛选数据**：例如只转发温度超过阈值的上报，其余消息不进入下游。
- **按需裁剪和重组字段**：例如从设备上报的 JSON 中提取几个字段，重命名后输出。
- **把一条消息拆成多条**：例如将设备批量上报的数组按元素拆分，逐条投递。
- **响应客户端事件**：例如设备上下线、订阅变化、证书注册或拒绝时，把事件转发给业务系统。
- **把数据路由到不同目的地**：例如写入 CKafka 供大数据处理，重发布到另一个 MQTT 主题供应用订阅，或触发云函数执行业务逻辑。

工作原理

每条规则是一条由三个环节组成的处理流水线：



```
┌──────────────────────────┐
│ ③ 输出动作（Sink）        │ 把 SQL 结果写入目标系统：
│                          │   CKafka / MQTT / SCF
└──────────────────────────┘
```

1. **数据来源**：规则只接收与其数据来源匹配的数据，可以是指定 Topic（支持通配符）上的消息，也可以是某一类客户端事件。每条 JSON 消息会按您定义的 Schema 解析 Payload，不符合 Schema 的消息在这一环节将被丢弃。
2. **数据处理**：规则对每条数据执行 SQL。不满足 `WHERE` 条件的数据被过滤，满足条件的数据按 `SELECT` 生成输出结果。一条数据通常产生一条结果，使用 `UNNEST` 展开数组时可以产生多条。
3. **输出动作**：每条 SQL 结果按动作配置渲染为目标消息（例如 Kafka 的 Topic、Key、Value），然后投递到目标系统。

这三个环节也是监控的三个观测点，每个环节都有对应的数量指标，例如规则触发数、SQL 成功数、动作成功数。详情请参见 监控与排障 。

基本概念

概念	说明
规则	规则引擎的管理对象，由一个数据来源、一段 SQL 和一个输出动作组成。规则可以独立启用、停用、编辑和删除。
数据来源	规则的输入，分为两类：**消息**（按 Topic 匹配客户端发布的消息）和**客户端事件**（连接、断开、订阅、取消订阅、证书注册、证书拒绝）。数据来源的**名称**就是 SQL 中 `FROM` 后面的表名。
Schema	描述消息 Payload 结构的 JSON Schema，用于说明 Payload 包含哪些字段、各字段的类型以及哪些字段必填。在 SQL 中读取 JSON 嵌套字段前，必须先定义 Schema。客户端事件的 Schema 由系统内置，无需定义。
SQL	对每条数据进行提取、转换和过滤的语句，形如 `SELECT … FROM <数据来源名称> WHERE …` 。SQL 的输出列可以在输出动作中引用。
输出动作	规则的输出，决定处理结果写什么、写到哪里。当前支持 CKafka、MQTT、SCF 三种，每条规则配置一个输出动作。
连接	输出动作访问目标系统所需的连接信息，例如目标实例和鉴权信息。连接单独管理，可被多个输出动作复用。
动态字段	输出动作中可以引用 SQL 结果的配置项，例如 Kafka 的 Topic 和 Value、MQTT 的目标 Topic 和 Payload。填写方式有**输出字段**（原样使用某一列的值）和**输出模板**（把固定文本和多个字段拼接为字符串）两种。

这些概念的关系如下：

```
规则
├── 数据来源（名称 + Topic / 事件类型 + Schema）
├── SQL（FROM 数据来源名称）
└── 输出动作 ──引用──▶ 连接
```

应用场景

场景	数据来源	处理逻辑	输出动作
高温告警写入大数据平台	消息：`sensors/#`	`SELECT client_id, payload.temp AS temp FROM sensors WHERE payload.temp > 30`	CKafka
按 Topic 分流到不同 Kafka Topic	消息：`factory/+/telemetry`	用 `topic LIKE 'factory/line1/%'` 过滤，每条规则写入一个 Kafka Topic，多条规则实现分流	CKafka
设备离线通知业务应用	客户端事件：客户端断开	`SELECT clientid, reason, disconnected_at FROM offline_events`	MQTT（重发布到 `notify/offline`）
批量上报拆分为单条	消息：`batch/#`	用 `UNNEST` 展开 `payload.records` 数组，每个元素生成一条结果	CKafka
没有 Kafka，直接触发业务逻辑	消息：`orders/#`	`SELECT client_id, payload.order_id AS order_id FROM orders`	SCF
数据推送到另一个 MQTT 实例	消息：`device/+/status`	`SELECT topic, payload FROM device_status`	MQTT（投递到同地域其他实例）

投递语义

使用规则引擎前，请了解以下投递语义，它们会影响下游系统的设计：

- **至少一次（at-least-once）**：下游暂时不可用、网络抖动、限流等可恢复的失败，规则引擎会持续重试，直到投递成功。下游可能收到重复数据，请在下游做幂等处理。
- **无法处理的数据直接丢弃，不重试**：Payload 不符合 Schema、SQL 执行出错、渲染后的消息不合法（例如超出大小限制）时，数据被丢弃并计入对应环节的丢弃数。当前不提供死信队列，丢弃的数据无法找回或重放。

版权所有：腾讯云计算（北京）有限责任公司

- **配置错误需要手动修复**：目标资源不存在或无权限时（例如 Kafka Topic 被删除），规则持续失败并自动重试恢复，直到您修正配置。
- **从最新位置开始处理**：规则创建、重新启用或更新后，从最新位置开始接收数据，此前的消息不会补处理。

各类失败的具体处理方式，请参见各输出动作文档中的**失败处理**，以及 监控与排障 。

公测说明

规则引擎目前处于 Beta 公测阶段，说明如下：

项目	说明
计费	公测期间免费，正式版的计费项、单价和开始时间另行公告。正式收费前，腾讯云会通过站内信、短信、邮件等方式提前通知。
支持地域	消息队列 MQTT 版已开放的所有地域
支持的集群版本与规格	专业版和铂金版集群均支持
配额	每个集群最多创建10条规则、10个连接；每条规则包含 1 个数据来源、1 段 SQL、1 个输出动作。详情请参见 使用限制 。

> **说明：**
> 规则引擎本身公测免费，但相关资源按各自规则计费：MQTT 动作重发布到本实例的消息会占用集群 TPS 规格；写入 CKafka 和调用云函数产生的费用，按 CKafka 和云函数的计费规则收取。

管理连接

最近更新时间：2026-11-18 15:11:00

连接保存输出动作访问目标系统所需的信息，例如目标实例、函数和鉴权信息。连接独立于规则管理，一个连接可以被多个输出动作引用。本文介绍连接的类型，以及新建、测试、更换和删除连接的方法。

连接概述

连接与输出动作的分工

对象	负责的内容	示例
连接	连接到哪里、使用什么身份连接	CKafka 实例；MQTT 实例及用户名、密码；SCF 函数
输出动作	写什么、怎么写	Kafka Topic、Key、Value；MQTT 目标 Topic、QoS、Payload；SCF 触发内容等

例如，两条规则可以共用一个 CKafka 连接，分别写入不同的 Kafka Topic。

连接类型

连接类型与输出动作类型一一对应。配置输出动作时，连接下拉框中只列出同类型的连接。

连接类型	对应的输出动作	连接对象
CKafka	CKafka 动作	同地域的 CKafka 实例
MQTT	MQTT 动作（重发布到本实例，或投递到其他实例）	同地域的 MQTT 实例
SCF	SCF 动作	云函数

连接列表

在左侧导航栏选择**规则引擎** > **连接管理**，在页面顶部选择地域和集群，查看当前集群下的全部连接。连接属于集群维度，不同集群的连接互不可见。

列	说明
连接 ID/名称	连接 ID 由系统生成，格式为 `connector_xxxxxxxx`
描述	创建连接时填写的描述
连接类型	CKafka、MQTT 或 SCF

版权所有：腾讯云计算（北京）有限责任公司

规则数量	引用该连接的规则数量
创建时间	连接的创建时间

新建连接

新建 CKafka 连接

1. 登录 消息队列 MQTT 版控制台，在左侧导航栏选择**规则引擎** > **连接管理**，在页面顶部选择地域和目标集群。
2. 单击**新建连接**，填写以下参数：

参数	是否必填	说明
连接类型	是	选择 CKafka
连接名称	是	系统默认生成 `connection-xxxxxxxx`，可修改。建议使用能识别用途的名称，便于在输出动作中选择。
描述	否	不超过128个字符
CKafka 实例	是	只列出与当前 MQTT 集群**同地域**的 CKafka 实例

3. 单击**测试连接**，测试通过后，单击**确定**保存。

> ⚠ **注意：**
> 当前 CKafka 连接不涉及认证配置，不支持开启 SASL 鉴权的 CKafka 实例。

新建 MQTT 连接

MQTT 连接有两种用途：

- **重发布到本实例**：目标实例选择当前集群。
- **投递到其他实例**：目标实例选择同地域的其他 MQTT 集群。

两种用途都需要创建连接，并填写目标实例的用户名和密码。

1. 在**规则引擎** > **连接管理**页面，单击**新建连接**。
2. 填写以下参数：

参数	是否必填	说明
连接类型	是	选择 MQTT
连接名称	是	系统默认生成，可修改
描述	否	不超过128个字符
MQTT 实例	是	可选择当前集群，或同地域的其他 MQTT 集群
用户名	是	规则引擎使用该用户名连接目标实例并发布消息
密码	是	用户名对应的密码，保存后不回显

3. 单击**测试连接**，测试通过后，单击**确定**保存。

> **说明：**
> - 用户名和密码可以使用目标实例上已有的账号，也可以为规则引擎单独新建一个。单独新建便于区分规则引擎发布的消息，但不是必须的。
> - 如果目标实例开启了授权策略，需确保该用户名具有目标 Topic 的**发布**权限，否则输出动作会因无权限而持续失败。授权方法请参见 授权策略 。

新建 SCF 连接

SCF 连接指向一个具体的云函数版本。

1. 在**规则引擎** > **连接管理**页面，单击**新建连接**。
2. 填写以下参数：

参数	是否必填	说明
连接类型	是	选择 SCF
连接名称	是	系统默认生成，可修改
描述	否	不超过128个字符
命名空间	是	云函数所在的命名空间
函数名	是	要触发的云函数
版本	是	要触发的函数版本

3. 单击**测试连接**，测试通过后，单击**确定**保存。

传给函数的内容在 SCF 输出动作的**触发内容**中配置，详情请参见 输出到云函数 SCF 。

测试连接

新建连接时必须先测试连接，**测试通过后才能保存**。测试连接只验证规则引擎到目标系统的链路是否连通。
测试失败时，请参考下表排查：

连接类型	常见原因	处理建议
CKafka	CKafka 实例状态异常	在 CKafka 控制台确认实例运行正常
MQTT	用户名或密码错误	在目标实例的**认证管理**中核对用户名和密码
MQTT	目标实例状态异常	确认目标集群运行正常
SCF	命名空间、函数或版本不存在	在云函数控制台确认函数及版本存在
通用	网络临时异常	稍后重试

说明：
测试通过只代表链路连通，不检查输出动作中的配置。例如，CKafka 连接测试通过，不代表输出动作中填写的 Kafka Topic 一定存在；MQTT 连接测试通过，也不代表该用户名具有目标 Topic 的发布权限。

更换连接

当前不支持修改已创建的连接。如需更换目标实例、更新密码或切换函数版本，请按以下步骤操作：

1. **新建连接**：按新的配置新建一个连接，测试通过后保存。
2. **改绑输出动作**：进入规则详情，编辑输出动作，把**连接**改为新连接，然后保存。
3. **更新规则**：回到规则列表，在该规则的操作列单击**更新**，规则在 1 分钟内按新配置生效。
4. **删除旧连接**：确认规则运行正常后，删除旧连接。

注意：
更新规则后，规则从最新位置开始处理数据，此前的处理进度和监控数据不会保留，更新过程中到达的消息可能不被处理。建议在业务低峰期操作。

删除连接

在连接列表的操作列单击**删除**，确认后即可删除。如果连接仍被输出动作引用（连接列表中**规则数量**大于0），则无法删除，系统会提示引用该连接的动作数量。请先把这些输出动作改绑到其他连接（请参见上文**更换连接**），或删除对应规则，再删除连接。

管理规则

最近更新时间：2026-11-18 15:35:00

规则是规则引擎的管理对象，由一个数据来源、一段 SQL 和一个输出动作组成。本文介绍规则的新建、测试、启停、编辑和删除方法。

查看规则列表

在左侧导航栏选择**规则引擎** > **规则管理**，在页面顶部选择地域和目标集群，查看当前集群下的全部规则。

列	说明
规则 ID/名称	规则 ID 由系统生成，格式为 `pipeline_xxxxxxxx`
状态	规则当前的运行状态，例如运行中
监控	单击图标查看该规则的监控数据，详情请参见 监控与排障
数据源	规则的数据来源：消息来源显示 Topic（例如 `sensors/#` ），事件来源显示事件类型
动作	输出动作的类型和名称
启停	启用或停用规则的开关
创建时间	规则的创建时间
操作	**编辑**、**删除**；规则修改后待生效时显示**更新**

新建规则按钮上显示当前集群已创建和可创建的规则数，例如**新建规则（2/10）**。达到上限后无法继续新建。

新建规则

单击**新建规则**，按下文向导完成5步配置。

步骤 1：配置规则信息

参数	是否必填	说明
规则名称	是	用于识别规则，不超过 128 字节
备注	否	规则用途说明
启用状态	是	选择**启用**时，规则创建成功后立即开始处理数据

步骤 2：配置数据来源

数据来源决定规则处理哪些数据。

参数	是否必填	说明
名称	是	数据来源的名称，同时作为 SQL 中 `FROM` 后面的表名。
来源类型	是	**消息**或**客户端事件**
Topic	来源类型为消息时必填	要处理的消息 Topic，支持通配符 `+`（单层）和 `#`（多层），例如 `sensors/#`、`factory/+/telemetry`
事件类型	来源类型为客户端事件时必填	从下拉列表中选择一类事件
Payload 类型	来源类型为消息时必填	可选 JSON、String 或 Binary，详情见 下文
Schema	Payload 类型为 JSON 且需要读取内部字段时必填	描述 Payload 的字段结构，详情见下文 定义 Schema

数据来源名称的命名规则

名称必须同时满足以下条件：

- 以小写英文字母开头。
- 长度不超过128个字符。
- 只包含小写英文字母、数字和下划线。
- 不以下划线结尾，不包含连续的两个下划线。

> ⚠ **注意：**
> SQL 的 `FROM` 后面写的是**数据来源名称**，不是 Topic。例如数据来源名称为 `sensors`、Topic 为 `sensors/#`，SQL 应写成 `FROM sensors`，而不是 `FROM "sensors/#"`。

客户端事件

来源类型选择**客户端事件**时，可以选择以下6类事件：

事件类型	事件主题（v3，默认版本）	事件主题（v1/v2）	备注
连接事件	$events/client/connected	$events/client_connected	客户端连接成功

断开事件	$events/client/disconnected	$events/client_disconnected	客户端连接断开，详细原因可参考下表 客户端断开原因 。
订阅事件	$events/session/subscribed	$events/session_subscribed	客户端会话新增了订阅。
取消订阅事件	$events/session/unsubscribed	$events/session_unsubscribed	客户端会话取消了订阅。
证书注册事件	$events/certificate/registered	$events/certificate_registered	客户端注册了设备证书。
证书拒绝事件	$events/certificate/rejected	$events/certificate_rejected	客户端设备证书被拒绝，证书状态为非激活状态（PENDING_ACTIVATION），详细原因可参考下表 客户端证书拒绝原因 。

事件的字段结构由集群的事件方言决定（v1、v2、v3，默认 v3）。控制台根据当前集群的方言自动匹配事件和字段，您只需选择事件类型，无需定义 Schema。事件字段在 SQL 中直接作为列引用，例如 `SELECT clientid, reason FROM offline_events` **。**

- **各事件的字段说明，请参见 数据来源与字段 。**
- **查看和切换集群事件方言的方法，请参见 查询客户端事件 。**

Payload 类型

Payload 类型	适用场景	SQL 中的能力
JSON	Payload 是 JSON 文本	定义 Schema 后，可读取任意嵌套字段，并使用 `UNNEST` 展开数组
String	Payload 是普通文本	只能读取整个 `payload`
Binary	Payload 是二进制数据，例如 Protobuf	只能读取整个 `payload`

定义 Schema

Schema 描述 JSON Payload 的字段结构，作用如下：

- **字段解析**：SQL 根据 Schema 读取 `payload.xxx` 等嵌套字段。
- **输入校验**：不符合 Schema 的消息在进入 SQL 前被丢弃。
- **测试和提示**：测试 SQL 根据 Schema 预填模拟消息。

操作步骤

1. 粘贴一条真实或样例消息，单击**解析**，系统自动生成字段列表。
2. 检查每个字段的类型，按需修正。例如样例中的 `"temp": 30` 被识别为整数，但实际上报可能带小数，此时应改为 number。
3. 勾选**必填**字段。

必填字段与校验规则如下：

字段定义	字段存在且类型正确	缺少该字段	字段类型不匹配
必填	正常取值	整条消息丢弃	整条消息丢弃
非必填	正常取值	放行，SQL 中该字段为 NULL	放行，SQL 中该字段为 NULL

被丢弃的消息计入规则引擎监控中的来源丢弃数。

> **说明：**
> - 建议只把业务上不可缺少的字段设为必填。必填字段越多，因个别设备上报格式不规范而被丢弃的消息越多
> - Schema 的完整语法（支持的类型、`format` 用法、数值转换规则等），请参见 JSON Schema 参考。

步骤:3：配置数据处理 SQL

1. 在 SQL 编辑器中编写 SQL，基本格式如下：

```
SELECT <字段或表达式> FROM <数据来源名称> WHERE <条件>
```

   语法详情请参见 SQL 语法。
2. 单击**测试 SQL**，验证 SQL 是否符合预期。SQL 的**输出列名**就是输出动作中可以引用的字段名。

步骤 4：配置输出动作

选择动作类型和对应类型的连接，并配置目标参数。每条规则配置一个输出动作。

动作类型	用途	需要的连接	参数说明
CKafka	写入同地域 CKafka 实例的 Topic	CKafka 连接	输出到 CKafka

MQTT	重发布到本实例的其他 Topic，或投递到同地域其他 MQTT 实例	MQTT 连接	输出到 MQTT
SCF	异步触发云函数	SCF 连接	输出到云函数 SCF

所有动作的表单都分为三部分：

- **基本配置**：动作名称、备注和动作类型。
- **目标配置**：连接和目标参数。
- **输出内容**：写入目标的消息内容。

连接下拉框中只列出与动作类型一致的连接。没有可用的连接时，单击**前往新建**创建连接，创建后返回并单击刷新。
输出动作中的 Topic、Key、Value、Payload 等配置项可以引用 SQL 结果，详情请参见 动态字段 。

步骤 5：完成创建

确认配置后，系统按以下顺序依次创建，并逐项显示状态（待执行、执行中、成功、失败）：

1. 规则
2. 数据来源
3. 数据处理
4. 输出动作

全部显示**成功**后，规则创建完成。如果创建时选择了启用，规则开始处理数据。
某一项失败时，创建停在该项。失败原因通常有以下两类：

- **参数不合法**：例如 SQL 引用了不存在的字段，或输出动作引用了 SQL 中没有的列。请根据提示修改参数，单击该项的**重试**。
- **网络暂时异常**：直接重试。

测试 SQL

测试 SQL 用于在规则上线前验证 SQL 和 Schema 是否符合预期。

1. 在**数据处理**步骤中，单击**测试 SQL**。
2. 系统根据 Schema 预填一条模拟消息，按需修改模拟消息中的字段值，包括 `topic` 、`client_id` 等系统字段和 Payload。
3. 单击**执行**，查看测试结果。

测试结果说明如下：

结果	含义	处理建议
输出一行或多行数据	消息满足 `WHERE` 条件，输出结果即实际投递时可引用的字段	检查输出列名和值是否符合预期
无输出	消息不满足 `WHERE` 条件，被过滤	与预期不符时，检查 `WHERE` 条件

报错	SQL 语法错误，或模拟消息不符合 Schema	根据错误信息修改 SQL、Schema 或模拟消息

> ⚠ **注意：**
> - 测试 SQL **不会触发输出动作**，不会向 CKafka、MQTT 或 SCF 写入任何数据。

启用和停用规则

在规则列表的**启停**列，单击开关启用或停用规则。

操作	效果
停用	规则停止接收和处理数据，不再匹配消息，也不再触发输出动作
启用	规则从**最新位置**开始接收数据

> ⚠ **注意：**
> 停用期间到达的消息，在重新启用后不会补处理。如需在维护期间保留数据，请在停用前评估影响。

编辑和更新规则

编辑规则

在规则列表的操作列单击**编辑**，可以修改规则的数据来源、Schema、SQL 和输出动作。

更新生效

编辑并保存后，修改**不会立即生效**，线上仍按原配置运行。此时规则列表的操作列出现**更新**，单击**更新**后，新配置在**1分钟内**生效。

> ⚠ **注意：**
> 规则更新后：
> - 规则从**最新位置**开始处理数据，更新过程中到达的少量消息可能不被处理。
> - 此前的处理进度和监控数据不会保留，监控数据从更新时刻重新开始统计。
>
> 建议在业务低峰期更新规则。如需对比更新前后的效果，请在更新前记录监控数据。

删除规则

在规则列表的操作列单击**删除**，确认后删除规则。

- 删除规则会同时删除其数据来源、SQL 和输出动作，**不会删除**输出动作引用的连接。
- 删除操作不可恢复。

SQL 参考
数据来源与字段

最近更新时间：2026-11-18 15:35:00

本文介绍规则 SQL 中可引用的字段：消息来源的字段、客户端事件来源的字段，以及不同 Payload 类型在 SQL 中的能力。

消息来源的字段

数据来源类型为**消息**时，SQL 中可以引用以下字段：

字段	说明
`topic`	消息的 Topic
`qos`	消息的 QoS
`client_id`	发布消息的客户端 ID
`username`	发布消息的客户端用户名
`timestamp`	消息时间戳
`user_properties`	MQTT 5 用户属性，可使用 `user_property_get` 、`user_property_get_all` 函数按 key 读取
`peerhost`	发布端地址。TCP 连接为客户端 IP，QUIC 连接为 stream ID
`message_id`	消息 ID
`payload`	消息内容。JSON 类型可通过 `payload.xxx` 读取内部字段，请参见 SQL 语法

使用说明：

- `SELECT *` 按上表顺序输出全部字段，包括 `peerhost` 和 `message_id` 。
- 除 `payload` 外，其他字段不能继续访问子字段，例如 `topic.value` 不合法。
- `WHERE` 中引用的字段可以不出现在 `SELECT` 中。

> ⚠ **注意：**
> 消息来源元数据中的客户端 ID 字段名是 `client_id` ；客户端事件来源中，发生连接或断开的客户端 ID 位于 `payload.clientid` （v2 方言为 `payload.clientId` ）。两者含义和写法都不同，请勿混用。

客户端事件来源的字段

数据来源类型为**客户端事件**时，事件字段与消息来源一样，通过 `payload.` 前缀引用。字段名由集群的事件方言决定，例如 v1、v3 为 `payload.clientid`，v2 为 `payload.clientId`：

```
-- 数据来源：名称 offline_events，事件类型为客户端断开
SELECT payload.clientid AS clientid,
       payload.reason_code AS reason_code,
       payload.disconnected_at AS disconnected_at
FROM offline_events
WHERE payload.reason_code <> 'normal_disconnect'
```

各事件包含的字段，请参见下文**客户端事件字段参考**。
部分事件字段的值是**数字字符串**，例如客户端连接事件的 `keepalive`、`proto_ver`，取消订阅事件的 `qos`。做数值比较前，请先用 `CAST` 或 `TRY_CAST` 转换：

```
SELECT payload.clientid AS clientid FROM connect_events
WHERE TRY_CAST(payload.keepalive AS INT) > 300
```

Payload 类型与 SQL 能力

Payload 类型	`SELECT payload`	`payload.xxx` 嵌套字段	`UNNEST`
JSON（已定义 Schema）	支持	支持	支持
JSON（未定义 Schema）	支持	不支持	不支持
String	支持	不支持	不支持
Binary	支持	不支持	不支持

- String 和 Binary 类型即使配置了 Schema，也不使用。
- Binary 类型的 `payload` 需要原样转发时，在输出动作中使用**输出字段**模式，请参见 动态字段 。

客户端事件字段参考

事件的字段结构由集群的**事件方言**决定。控制台根据当前集群的方言自动匹配，您只需选择事件类型。消息队列 MQTT 版支持 v1、v2、v3 三种事件方言，**默认为 v3**。集群版本为1.4.0及以上时，可在**集群管理**的集群基本信息页切换。详情请参见 查询客户端事件 。
字段名区分大小写，例如 v2 方言为 `clientId`，v1、v3 方言为 `clientid`。下表列出的是 Schema 中的属性名，在 SQL 中需加 `payload.` 前缀引用，例如 `payload.clientid`。类型为**数字字符串**的字段，值是字符串形式的数字（例如 `"60"`），不是 JSON 数字，做数值比较前请先用 `TRY_CAST` 转换。

客户端连接事件

v1、v3 方言

字段	类型	说明
`event`	字符串	事件类型
`clientid`	字符串	客户端 ID
`username`	字符串	用户名
`clean_start`	字符串	是否清除会话，值为 `"true"` 或 `"false"`
`connected_at`	整数	连接时间（毫秒时间戳）
`keepalive`	数字字符串	保活时间（秒）
`expiry_interval`	数字字符串	会话过期时间
`proto_ver`	数字字符串	MQTT 协议版本
`peername`	字符串	客户端地址
`sockname`	字符串	服务端地址
`node`	字符串	服务节点
`instance`	字符串	实例 ID
`timestamp`	整数	事件时间（毫秒时间戳）
`dialect`	字符串	事件方言
`certificateChainSn`	字符串	证书链序列号

v2 方言

字段	类型	说明
`eventType`	字符串	事件类型
`eventIndex`	整数	事件序号
`clientId`	字符串	客户端 ID

time	整数	事件时间（毫秒时间戳）
channelId	整数	连接通道 ID
clientIp	字符串	客户端 IP
dialect	字符串	事件方言
certificateChainSn	字符串	证书链序列号

客户端断开事件

v1、v3 方言

字段	类型	说明
event	字符串	事件类型
clientid	字符串	客户端 ID
username	字符串	用户名
reason_code	字符串	断开原因码，例如 normal_disconnect
reason	字符串	断开原因，例如 normal 。取值请参见 客户端断开原因
disconnected_at	整数	断开时间（毫秒时间戳）
expiry_interval	数字字符串	会话过期时间
peername	字符串	客户端地址
sockname	字符串	服务端地址
node	字符串	服务节点
instance	字符串	实例 ID
timestamp	整数	事件时间（毫秒时间戳）
dialect	字符串	事件方言
certificateChainSn	字符串	证书链序列号

v2 方言

字段	类型	说明
eventType	字符串	事件类型
eventIndex	整数	事件序号
clientId	字符串	客户端 ID
time	整数	事件时间（毫秒时间戳）
channelId	整数	连接通道 ID
clientIp	字符串	客户端 IP
reasonCode	字符串	断开原因码
reason	字符串	断开原因
dialect	字符串	事件方言
certificateChainSn	字符串	证书链序列号

客户端订阅事件

所有方言的字段相同。

字段	类型	说明
event	字符串	事件类型
clientid	字符串	客户端 ID
username	字符串	用户名
topic	字符串	订阅的 Topic
qos	整数	订阅的 QoS
result	字符串	订阅结果
reason_code	字符串	原因码
peerhost	字符串	客户端地址
node	字符串	服务节点

版权所有：腾讯云计算（北京）有限责任公司

instance	字符串	实例 ID
timestamp	整数	事件时间（毫秒时间戳）
certificateChainSn	字符串	证书链序列号

客户端取消订阅事件

所有方言的字段相同。

字段	类型	说明
event	字符串	事件类型
clientid	字符串	客户端 ID
username	字符串	用户名
topic	字符串	取消订阅的 Topic
qos	数字字符串	QoS。与订阅事件不同，此处为字符串
peerhost	字符串	客户端地址
node	字符串	服务节点
instance	字符串	实例 ID
timestamp	整数	事件时间（毫秒时间戳）
certificateChainSn	字符串	证书链序列号

证书注册事件

所有方言的字段相同。

字段	类型	说明
event	字符串	事件类型
sn	字符串	证书序列号
caSn	字符串	CA 证书序列号
commonName	字符串	证书通用名称

status	字符串	证书状态
certificateChainSn	字符串	证书链序列号
peerName	字符串	客户端地址
node	字符串	服务节点
instance	字符串	实例 ID
timestamp	整数	事件时间（毫秒时间戳）

证书拒绝事件

所有方言的字段相同。

字段	类型	说明
event	字符串	事件类型
reason	字符串	拒绝原因。取值请参见 查询客户端事件
sn	字符串	证书序列号
caSn	字符串	CA 证书序列号
commonName	字符串	证书通用名称
status	字符串	证书状态
certificateChainSn	字符串	证书链序列号
peerName	字符串	客户端地址
node	字符串	服务节点
instance	字符串	实例 ID
timestamp	整数	事件时间（毫秒时间戳）

版权所有：腾讯云计算（北京）有限责任公司

SQL 语法与示例

最近更新时间：2026-11-18 15:11:01

规则引擎使用标准 SQL 对每条消息逐条进行字段提取、计算和过滤。

SQL 语法

语句结构

只能写一条 SQL 语句，且必须是 `SELECT` 查询。

```
SELECT <字段或表达式> [AS <别名>], ...
FROM <数据来源名称>
[WHERE <条件>]
```

- `SELECT`：选择和计算要输出的字段。输出列就是输出动作中可以引用的字段。
- `FROM`：数据来源名称。
- `WHERE`：过滤条件。不满足条件的消息不产生输出。

FROM 与数据来源名称

`FROM` 后面写的是规则中**数据来源的名称**。

数据来源配置	正确写法
名称 `sensors`，Topic `sensors/#`	`FROM sensors`
名称 `offline_events`，事件类型为客户端断开	`FROM offline_events`

- 名称必须与数据来源名称**完全一致**，且不加引号。命名规则请参见 管理规则 。

读取 JSON 字段

Payload 类型为 JSON 且已定义 Schema 时，可以用以下写法读取内部字段：

```
SELECT
    payload.device.id,          -- 点号访问嵌套字段
    payload."orderId",          -- 带引号，等价于 payload.orderId
    payload['orderId'],         -- 字符串下标，等价于 payload.orderId
    payload.items[0].price      -- 数组下标，从 0 开始
FROM orders
```

读取规则：

- **字段名区分大小写**，必须与 Schema 中定义的名称完全一致。
- 路径中的每一层都必须能在 Schema 中找到：对象才能用点号取字段，数组才能用下标。

输出列名

SQL 的输出列名，就是输出动作中 `${列名}` 可以引用的名字，区分大小写。支持使用 AS 指定输出列名。

```
SELECT payload.deviceId AS DeviceId FROM sensors
```

不写别名时的自动命名规则如下：

表达式	自动生成的列名
`client_id`	`client_id`
`payload.device.Id`	`Id`（取路径最后一段）
`payload.items[0]`	`0`
`payload.temp * 1.8 + 32` 等复杂表达式	`expr_1`、`expr_2` ……
自动名称重复，例如 `payload.orderId, payload.orderid`	`orderId`、`orderid_1`

> **说明：**
> 建议为每个输出列显式写别名，例如 `payload.deviceId AS device_id`。这样列名可预期，输出动作也不易引用错。不确定列名时，可使用 测试 SQL 查看实际输出。

运算符与表达式

规则引擎支持以下逐条消息的表达式：

类别	示例
字面量	`'abc'`、`123`、`1.5`、`true`
算术运算	`payload.temp * 1.8 + 32`、`payload.a % 10`
比较运算	`=`、`<>`、`!=`、`>`、`>=`、`<`、`<=`

逻辑运算	`AND`、`OR`、`NOT`
空值判断	`payload.temp IS NULL`、`payload.temp IS NOT NULL`
类型转换	`CAST(payload.code AS INT)`、`TRY_CAST(payload.code AS INT)`
模式匹配	`topic LIKE 'factory/line1/%'`、`ILIKE`（不区分大小写）、`SIMILAR TO`
字符串拼接	`payload.a || payload.b`
范围判断	`payload.temp BETWEEN 30 AND 50`
固定列表	`payload.level IN ('warn', 'error')`
条件表达式	`CASE WHEN payload.temp > 50 THEN 'high' WHEN payload.temp > 30 THEN 'mid' ELSE 'low' END`

使用说明：

- 字符串常量使用单引号，例如 `'warn'`。
- `CAST` 转换失败会导致 SQL 运行出错，`TRY_CAST` 转换失败时返回 NULL，不会出错。处理设备上报数据时，建议优先使用 `TRY_CAST`，原因请参见下文 编写健壮的 SQL。
- `IN` 只支持固定值列表，不支持 `IN (SELECT ...)`。

函数

规则引擎支持**确定性的标量函数**，涵盖字符串、正则表达式、数学、日期和时间、条件与空值、数组、编解码与哈希、结构体等类别，以及读取 MQTT 5 用户属性的专用函数。

子查询、CTE 与 UNNEST

派生子查询

`FROM` 中可以使用子查询，并且可以多层嵌套。外层查询可以引用内层的输出列，整个查询只能有一个最底层数据来源，子查询的每一层都只能沿着这一个数据来源展开。如下所示：

```
SELECT device_id, temp_f
FROM (
    SELECT payload.device_id AS device_id,
```

```
            payload.temp * 1.8 + 32 AS temp_f
    FROM sensors
) AS t
WHERE temp_f > 100
```

CTE

一个查询最多声明 1 个 CTE（`WITH` 子句）：

```
WITH base AS (
    SELECT payload.device_id AS device_id, payload.temp AS temp
    FROM sensors
)
SELECT device_id, temp
FROM base
WHERE temp > 30
```

UNNEST

`UNNEST` **把 Payload 中的一个数组按元素展开，一条消息产生多条输出，每个元素一条。**
例如设备批量上报多条读数：

```
{"device_id": "d001", "records": [{"ts": 1, "temp": 30}, {"ts": 2,
"temp": 31}]}
```

```
SELECT r.record.ts AS ts, r.record.temp AS temp
FROM (
    SELECT UNNEST(payload.records) AS record
    FROM batch_upload
) AS r
```

输出两条结果： `{ts: 1, temp: 30}` **和** `{ts: 2, temp: 31}` **。输出动作对每条结果各执行一次，例如在 CKafka 中写入两条 Kafka 消息。数组为 NULL 或空数组** `[]` **时，该消息不产生任何输出。**
使用规则：

- **整个查询（含所有子查询）最多使用一次** `UNNEST` **。**
- `UNNEST` **只能作为** `SELECT` **列表中的一个完整项，并且必须写别名，例如** `UNNEST(payload.records) AS record` **。**
- `UNNEST` **所在的** `SELECT` **必须直接读取数据来源，不能读取子查询的结果。**

- 参数必须是 Payload 中的数组路径（例如 `payload.records`、`payload.data.items`），不能是系统字段，也不能是表达式（例如 `UNNEST(COALESCE(...))`）。
- 必须为 Payload 定义 Schema，并在 Schema 中将该字段声明为数组。
- 不支持 `FROM UNNEST(...)` 和 `CROSS JOIN UNNEST(...)` 的写法。
- 需要按元素字段过滤时，把 `UNNEST` 放在子查询中，在外层写 `WHERE`。

> **说明：**
> 使用 `UNNEST` 后，监控中的 SQL 成功数大于处理消息数，这是正常现象。

编写健壮的 SQL

> **注意：**
> SQL 在运行时出错（例如除数为 0、`CAST` 转换失败）时，**同一批次的所有消息都会被丢弃**，包括其中格式正常的消息，并且不会重试。
> 规则引擎把短时间内到达的多条消息合并为一批处理。一条异常消息引发的 SQL 错误会让整批失败，被丢弃的数量计入监控中的 **SQL 失败数**，因此 SQL 失败数可能大于异常消息的数量。

在 Schema 校验阶段拦截的问题只会丢弃该条消息，不影响其他消息。因此，应尽量让异常数据在数据来源环节被拦截，避免进入 SQL 后才出错。

风险	不推荐的写法	推荐的写法
类型转换失败	`CAST(payload.code AS INT)`	`TRY_CAST(payload.code AS INT)`，失败时返回 NULL。
除数为0	`payload.a / payload.b`	`payload.a / NULLIF(payload.b, 0)`，除数为0时结果为 NULL。
关键字段缺失或类型错误	依赖 SQL 处理	在 Schema 中把关键字段设为必填，在数据来源环节逐条拦截。
字符串形式的数字	在 SQL 中使用 `CAST`	在 Schema 中用 `"type": "string", "format": "double"` 等声明，在数据来源环节转换。

上线前，建议使用 测试 SQL 测试边界值：0、负数、空字符串、超长字符串和缺失字段。

SQL 示例

按客户端过滤

```
SELECT client_id, payload FROM sensors
```

```
WHERE client_id LIKE 'factory-%'
```

按 Topic 层级过滤

```
SELECT topic, payload FROM telemetry
WHERE topic LIKE 'factory/line1/%'
```

提取并重命名嵌套字段

```
SELECT
    client_id,
    payload.device.id AS device_id,
    payload.metrics.temp AS temp,
    timestamp AS ts
FROM sensors
```

计算与单位换算

```
SELECT
    payload.device_id AS device_id,
    payload.temp * 1.8 + 32 AS temp_f
FROM sensors
```

按区间打标签

```
SELECT
    payload.device_id AS device_id,
    CASE
        WHEN payload.temp >= 50 THEN 'critical'
        WHEN payload.temp >= 30 THEN 'warning'
        ELSE 'normal'
    END AS level
FROM sensors
WHERE payload.temp >= 30
```

展开数组并按元素过滤

```
SELECT r.item.sku AS sku, r.item.qty AS qty
FROM (
    SELECT UNNEST(payload.items) AS item
    FROM orders
) AS r
WHERE r.item.qty > 0
```

处理设备离线事件

```
SELECT clientid, reason, disconnected_at
FROM offline_events
WHERE reason <> 'normal'
```

数据结构
JSON Schema

最近更新时间：2026-11-18 15:11:01

规则引擎使用 JSON Schema 描述 Payload 的结构。控制台可以通过粘贴样例消息自动生成 Schema，一般无需手写。本文介绍 Schema 的解析规则，供您调整字段类型时参考。

根节点

根节点必须是 `object` 或 `array`：

```
{ "type": "object", "properties": { } }
```

```
{ "type": "array", "items": { } }
```

根节点不能是标量类型或 `null`，也不能使用类型数组，例如 `["object", "null"]`。

支持的类型

支持 `object`、`array`、`boolean`、`integer`、`number`、`string`。不支持 `null` 类型。字段可能为空时，请不要把它设为必填。

对象与数组

- 对象的字段只从 `properties` 中查找，字段名区分大小写。`additionalProperties`、`patternProperties` 等关键字不生效。
- 数组用一个 `items` 描述所有元素，元素可以是对象或基本类型，也可以嵌套数组。不支持元组形式（`items` 为数组）和 `prefixItems`。

字段类型与 format

format 决定字段在 SQL 中的数据类型，取值区分大小写。

Schema 类型	format	SQL 中的类型
integer	`int32`	32 位整数
integer	`int64`、不填或其他自定义值	64 位整数
number	`float`	32 位浮点数

number	`double` 、不填或其他自定义值	64 位浮点数
string	`int32`	32 位整数
string	`int64` 、 `integer`	64 位整数
string	`float`	32 位浮点数
string	`double`	64 位浮点数
string	`date`	日期
string	`time`	时间（毫秒精度）
string	`date-time`	时间戳（毫秒精度，统一转为 UTC）
string	`duration`	时长（毫秒精度）
string	不填或其他自定义值	字符串
boolean	不填或其他自定义值	布尔

说明：

- “其他自定义值”指上表之外的 format（例如 `email` 、 `uint8` ），会被忽略。format 区分大小写， `Int32` 也算自定义值。
- 上表中的 format 只能用在对应的类型上： `integer` 只能用 `int32` 、 `int64` ； `number` 只能用 `float` 、 `double` ； `boolean` 、 `object` 、 `array` 不能用上表中的任何 format。否则 Schema 校验失败，例如 `{"type": "boolean", "format": "date"}` 、 `{"type": "integer", "format": "integer"}` 。

必填（required）字段

- 在对象的 `required` 数组中列出必填字段，例如 `"required": ["device_id", "temp"]` 。
- 通过数组下标访问的字段（例如 `payload.items[0].price` ）始终按非必填处理。
- 数组中的对象元素，按元素对象自身的 `required` 判断。

示例

示例一：数值字段

下面的 Schema 对应样例消息 `{"device_id": "d001", "temp": 35.2, "humidity": 60}` ，其中 `temp` 为必填：

```
{
  "type": "object",
  "properties": {
    "device_id": { "type": "string" },
    "temp": { "type": "number" },
    "humidity": { "type": "integer" }
  },
  "required": ["temp"]
}
```

按此 Schema：

- 消息中缺少 `temp`，或 `temp` 为 `null`：整条消息丢弃。
- 消息中缺少 `humidity`：`humidity` 取 NULL，消息正常处理。

示例二：数字以字符串形式上报

设备把数字作为字符串上报时，例如 `{"device_id": "d001", "temp": "36.5"}`，把该字段定义为 `{"type": "string", "format": "double"}`：

```
{
  "type": "object",
  "properties": {
    "device_id": { "type": "string" },
    "temp": { "type": "string", "format": "double" }
  },
  "required": ["temp"]
}
```

SQL 中即可直接对 `temp` 做数值比较，例如 `WHERE payload.temp > 30`。

> ⚠ **注意：**
> 消息中字段的 JSON 类型必须与 Schema 声明一致。按此 Schema，设备必须始终带引号上报 `temp`；如果收到不带引号的数字 `36.5`，类型不匹配，整条消息会被丢弃。

输出动作
输出到 CKafka

最近更新时间：2026-11-18 15:11:01

CKafka 动作把每一条 SQL 输出结果写入目标 CKafka 实例的 Topic。可以动态生成 Topic、Key、Value 和 Headers，适用于数据转储、告警输出与后续流式处理。

前提条件

- 已创建与 MQTT 集群**同地域**的 CKafka 实例。
- 已提前创建目标 Topic，不依赖自动创建 Topic。
- 已创建 CKafka 连接，并确认规则执行节点到目标 Broker 的网络可达，请参见 管理连接 。
- 若目标实例启用了认证或授权，请确认所用连接具备目标 Topic 的写入权限。

参数说明

目标配置

参数	是否必填	填写方式	说明
连接	是	选择连接	选择已创建的 CKafka 连接
Kafka Topic	是	输出模板	固定值，例如 `temp-alerts` ；或引用 SQL 输出列，例如 `alerts-${level}`
Kafka Headers	否	输出模板	键值对列表，名称和值均使用模板；按配置顺序发送，允许重复的键；不配置则不携带 Headers
压缩算法	否	固定值	可选不压缩（默认）、Snappy、LZ4、GZIP，具体选项以控制台提供的为准

输出内容

参数	是否必填	填写方式	说明
消息的键 Key	否	输出模板	参与分区路由，结果为文本；当前控制台仅支持输出模板，不支持输出字段或二进制取值

消息的值 Value	否	输出字段或输出模板	消息正文；输出字段支持文本或二进制列，输出模板始终生成文本

> **说明：**
> Key 模板即使只写 `${device_id}`，仍属于输出模板，不能配置成输出字段模式。不配置 Value 时发送的是非 null 的空字节值，不是 Kafka null tombstone（删除标记）。

Key 与 Value 的边界情况

配置或结果	写入 Kafka 的内容
不配置 Key	Key 为 `null`，由 Producer 的无 Key 分区逻辑选择分区
配置 Key，模板结果为空字符串	Key 为非 null 的空文本，与不配置 Key 不同
不配置 Value	Value 为非 null 的空字节，不是 tombstone
Value 模板结果为空，或输出字段的文本、二进制列为 NULL	Value 为非 null 的空内容，不是 tombstone

动态字段的写法请参见 动态字段 。

示例一：按设备写入高温告警

配置数据来源

数据来源名称为 `sensors`，Payload 类型为 JSON。示例消息：

```
{ "device_id": "d001", "temp": 35.2 }
```

对应 JSON Schema：

```
{
  "type": "object",
  "properties": {
    "device_id": { "type": "string" },
    "temp": { "type": "number" }
  },
  "required": ["device_id", "temp"]
}
```

配置 SQL

```
SELECT client_id, topic,
       payload.device_id AS device_id,
       payload.temp AS temp,
       timestamp AS ts
FROM sensors
WHERE payload.temp > 30
```

配置输出动作

参数	填写方式	值
Kafka 主题名称	输出模板	`temp-alerts`
Header 名称	输出模板	`source`
Header 值	输出模板	`mqtt-${topic}`
消息的键 Key	输出模板	`${device_id}`
消息的值 Value	输出模板	`{"device":"${device_id}","temp":${temp:0},"ts":"${ts}"}`

Key 使用设备字段 `device_id`，与按设备分区的目的保持一致。SQL 顶层 `timestamp` 是时间戳类型，模板将其格式化为时间文本，因此 JSON 中的 `ts` **需要加引号**，不能直接当数值拼接；`temp` 使用 `${temp:0}` 兜底，避免 NULL 造成非法 JSON。

> ⚠ **注意：**
> 模板不自动做 JSON 转义。本示例假定 `device_id` 是可安全放入 JSON 字符串的受控标识符，`temp` 是合法有限数值。若字段值可能包含双引号、反斜杠或控制字符，请在上游正确编码，或透传已编码的完整 JSON 文本。NULL 数值的默认值也不能处理 `NaN`、`inf` 或字符串转义。

示例二：原样写入二进制数据

将数据来源 `raw_messages` 的 Payload 类型设置为 BINARY。

```
SELECT client_id, payload AS binary_value
FROM raw_messages
```

参数	填写方式	值
Kafka 主题名称	输出模板	`raw-data`
消息的键 Key	输出模板	`${client_id}`
消息的值 Value	输出字段	`${binary_value}`

Value 保留原始二进制内容，不会自动编码为 JSON 或 Base64。若 `client_id` 为 NULL，Key 模板生成的是非 null 空文本；需要无 Key 路由时应省略整个 Key 配置。

失败处理

情况	处理方式
Producer 队列暂满	保留尚未发送的记录并施加背压，等待容量；不是丢弃，也不是已发送消息的重新投递
网络异常、请求超时、Broker 暂不可用、临时限流或副本不足	客户端持续重试并退避，默认初始退避 100ms、最大 1000ms，不是固定每秒一次
消息超过目标大小限制、记录不合法、目标 Topic 无权限等记录或目标永久拒绝	丢弃受影响的未确认记录，计入动作失败数；已确认结果保留
Topic、Key、Value 或 Header 单字段模板渲染超限，或渲染后的 Topic 不合法	只丢弃该条记录，其他记录继续处理
认证失败、集群权限拒绝或永久共享客户端拒绝	丢弃该客户端已开始投递的批次中受影响的未确认记录；后续投递可能重建客户端，已丢弃记录不等待自动补发

版权所有：腾讯云计算（北京）有限责任公司

输出到 MQTT

最近更新时间：2026-11-18 15:11:01

MQTT 动作把每一条 SQL 输出结果发布为一条 MQTT 消息。连接指向当前实例时，可将处理结果重发布到业务 Topic；连接指向其他实例时，可将结果投递到其他 MQTT 集群。两种用法的动作参数相同，区别在于所选连接的目标。

前提条件

- 已创建 MQTT 连接，填写目标实例的连接地址、用户名和密码；当前控制台要求用户名和密码非空，请参见 管理连接 。
- 已确认规则执行节点能够访问目标实例；跨实例使用时，请确认产品支持的地域范围与网络条件。
- 若目标实例启用了授权策略，请为连接账号授予目标 Topic 的**发布**权限，请参见 授权策略 。

参数说明

目标配置

参数	是否必填	填写方式	说明
连接	是	选择连接	选择已创建的 MQTT 连接，目标实例由连接决定
目标 Topic	是	输出模板	固定文本或引用 SQL 输出列，例如 `notify/offline/${clientid}`；不支持 Topic Filter 通配符 `+` 、 `#`
QoS	是	固定值	`0` 、 `1` 或 `2` ；表示动作向目标实例发布时的 QoS，**不是订阅方最终收到消息的 QoS**

MQTT 5.0 消息属性

MQTT 动作以 MQTT 5 协议发布消息，以下属性均为选填。使用 MQTT 3.1.1 的订阅方不能接收这些属性，但属性在目标服务端的处理不因此取消。

参数	填写方式	说明
Content Type	固定值	Payload 的 MIME 类型，例如 `application/json` ；不会执行内容转换、JSON 编码或 JSON 校验

Payload Format Indicator	固定值	`0` 表示未指定格式的字节流，`1` 表示 UTF-8 文本；不会把二进制转换为文本
Message Expiry Interval	固定值	消息过期时间，单位为秒，范围为 `0` ~ `4294967295`，最终受目标服务端策略约束
Response Topic	输出模板	请求响应场景中接收回复的 Topic，必须满足目标 Topic 的合法性规则
Correlation Data	输出模板	关联请求与响应的标识；当前控制台仅支持输出模板，结果按 UTF-8 字节处理，不支持输出字段或二进制取值

Correlation Data 最长为 65,535 字节，超过协议上限的记录会被丢弃。Content Type 不能包含 NUL，UTF-8 编码后不能超过 65,535 字节。

输出内容

参数	是否必填	填写方式	说明
Payload	否	输出字段或输出模板	输出字段可原样使用 SQL 输出的文本或二进制列，输出模板始终生成文本；不配置时发送空 Payload

发布消息的 `retain` 标志固定为 `false`，不提供将结果发布为保留消息的选项。空 Payload 与不发送消息不同，仍会向目标 Topic 发布一条消息。动态字段的写法请参见 动态字段 。

防环路机制

MQTT 动作在每条发布消息的 MQTT 5.0 用户属性（User Property）中添加系统保留属性 `$__hop=1` 。规则的数据来源收到仍带有该属性的消息时，跳过 SQL 处理，并计入监控中的**来源丢弃数**。判断只检查属性名 `$__hop` 是否存在，不检查属性值。

- 系统输出消息在属性被保留时，不会再次触发接收它的规则。
- 业务客户端自行携带该属性的消息也会被跳过，请不要在业务消息中使用这个属性名。
- MQTT 5.0 订阅方可能收到该属性，业务逻辑忽略即可。
- 桥接、MQTT 3.x 转发或应用重新发布若移除了该属性，不在防环路保证范围内；不能据此承诺任意跨系统链路都不会循环。

失败处理

情况	处理方式

临时网络异常、服务端繁忙、限流或临时连接故障	持续重试并等待退避；规则关闭、取消或系统故障可能中止重试
消息过大、目标 Topic 不合法、服务端判定 Payload 格式不合法，或所选 QoS 不受支持	丢弃受影响记录，计入动作失败数
单条字段模板渲染超限、Correlation Data 超过协议上限，或渲染后的 Topic 不合法	只丢弃该条记录，其他可投递记录继续处理
PUBLISH 返回 `NotAuthorized`	按记录级永久拒绝丢弃该条，修正权限后可处理后续记录；已丢弃记录不自动补发
连接认证失败、永久连接拒绝或永久断开原因	丢弃当前受影响的未确认记录，保留已确认结果；不保留原消息直至账号或权限修复

示例一：设备异常离线通知

配置数据来源

选择客户端断开事件，数据来源名称为 `offline_events`。本示例适用于使用 EMQ/V3 JSON 渲染、包含下列字段的断开事件；其他事件渲染格式需相应调整 Schema 和 SQL。
JSON Schema 示例：

```
{
  "type": "object",
  "properties": {
    "clientid": { "type": "string" },
    "reason": { "type": "string" },
    "reason_code": { "type": "string" },
    "disconnected_at": { "type": "integer" }
  },
  "required": ["clientid", "reason_code", "disconnected_at"]
}
```

事件中的 `clientid` 表示断开的客户端，SQL 顶层 `client_id` 表示外层消息的发布者，两者不能互相替代。`reason` 是说明文本，判断异常断开应使用稳定的 `reason_code`。

配置 SQL

```
SELECT payload.clientid AS clientid,
       payload.reason AS reason,
       payload.reason_code AS reason_code,
```

```
       payload.disconnected_at AS disconnected_at
FROM offline_events
WHERE payload.reason_code <> 'normal_disconnect'
```

配置输出动作

参数	填写方式	值
目标 Topic	输出模板	`notify/offline/${clientid}`
QoS	固定值	`1`
Content Type	固定值	`application/json`
Payload Format Indicator	固定值	`1`
Payload	输出模板	`{"client":"${clientid}","reason_code":"${reason_code}","at":${disconnected_at:0}}`

当 `clientid` 为 `d001`、`reason_code` 为 `keep_alive_timeout` 时，通知发布到 `notify/offline/d001`。

示例二：原样转发二进制 Payload

将数据来源 `raw_messages` 的 Payload 类型设置为 BINARY，不使用 JSON 路径读取。

```
SELECT payload AS forwarded_payload
FROM raw_messages
```

参数	填写方式	值
目标 Topic	输出模板	`forward/raw`
QoS	固定值	`1`
Payload Format Indicator	固定值	`0`
Payload	输出字段	`${forwarded_payload}`

该方式保留二进制内容，不会自动转换为 UTF-8、Base64 或 JSON。请勿把二进制列放入输出模板。

版权所有：腾讯云计算（北京）有限责任公司

输出到云函数 SCF

最近更新时间：2026-11-18 15:11:01

SCF 动作把 SQL 输出结果封装为事件，**异步**触发目标云函数。函数按 `Records` 数组处理多条结果，适用于业务逻辑触发和下游系统集成。

前提条件

- 已创建目标云函数，确认所用命名空间、函数和版本或别名有效。
- 已创建 SCF 连接，确认连接指向预期的函数及版本，请参见 管理连接 。
- 已确认调用身份具有目标函数的调用权限，规则执行环境的调用链路可用。

> ⚠ **注意：**
> 连接测试当前仅支持 CKafka 和 MQTT 连接，不支持 SCF 连接。请勿将其他类型连接的测试通过当作 SCF 可调用的证明，SCF 调用链路请通过规则运行验证。

参数说明

目标配置

参数	是否必填	填写方式	说明
连接	是	选择连接	选择已创建的 SCF 连接，函数命名空间、函数名称和版本由连接决定
环境变量	否	输出模板	随一次调用传递的请求级键值对，名称和值均可引用 SQL 输出列；不是逐条 Records 独立配置

输出内容

参数	是否必填	填写方式	说明
触发内容	是	输出模板	每条 SQL 输出结果对应的 JSON 元素；当前控制台要求该项存在且为输出模板，不支持输出字段模式

若 SQL 已输出完整、合法的 JSON 文本列，可使用模板 `${json_text}` 引用该列。只包含一个占位符的模板仍是输出模板，受模板的类型与长度规则约束，不能配置成输出字段。动态字段的写法请参见 动态字段 。

函数收到的事件格式

SCF 动作按**一个处理批次的全部 SQL 输出结果**构造一次请求。单个批次可能产生多组 SQL 输出数据，这些输出组中的有效记录共同组成该请求的 `Records` 数组：

```
{
  "Records": [
    { "device": "d001", "temp": 35.2 },
    { "device": "d002", "temp": 41.0 }
  ]
}
```

这不是动作按时间窗口独立收集任意数量的记录，也不会跨多个处理批次自动聚合。SQL 输出为零行时不调用函数；单条触发内容渲染失败的记录从本次有效 `Records` 中排除。函数代码需要遍历 `Records` 逐条处理。

JSON 编码要求

触发内容会直接作为 JSON 元素放入 `Records`，系统不会额外加引号、转义或进行 JSON 序列化。

- 输出 `{ "device": "d001" }` 可作为对象元素；普通文本 `abc` 不能直接作为 JSON 元素。
- 模板中的字符串外层加引号，不等于正确转义字符串内部的双引号、反斜杠和控制字符。
- NULL 数值可用 `${amount:0}` 等默认值，或在 SQL 中排除；`NaN`、`inf` 等非有限值不能作为 JSON 数值。
- 不可控文本应由上游正确序列化成完整 JSON 文本，再通过一个模板占位符整体引用，不要在已有 JSON 文本外再次添加引号。

失败处理

下表中的 HTTP 状态指云函数**调用接口**返回的状态，不是函数业务代码内部的返回字段。

情况	处理方式
HTTP 429、5xx 或临时网络异常	保留当前批次并持续重试整个调用；后续批次不能越过当前重试批次；规则关闭、取消或系统故障可中止重试
HTTP 401、403、404	当前规则执行实例失败；仍启用且配置可用的规则由后台清理并尝试重建，不是当前调用原地持续重试
其他非 2xx 状态，包括 HTTP 413 等请求拒绝	本次调用对应批次的未成功记录按策略丢弃，计入动作失败数，不自动拆分或补发
单条触发内容模板渲染超限	只丢弃该条，其他有效记录组成请求；若没有有效记录，则不发起调用
环境变量名称或值模板渲染超限	跳过该环境变量键值项，其他可渲染项继续传递

> ⚠ **注意：**
> 401、403、404 发生后，请检查调用身份、命名空间、函数名称和版本。保持启用不等于问题已经修复，配置未修正时重建后可能再次失败。被允许的策略丢弃的记录，不承诺在配置修复后自动补发；有补偿要求的业务应独立保留来源与业务标识。

成功与重复边界

- HTTP 2xx 只表示调用平台接受请求，不代表异步函数业务执行成功；函数执行结果请在云函数控制台查看日志与监控。
- 网络异常时平台可能已接受调用，但规则尚未收到确认，重试可能重复调用整个批次。
- 规则执行实例失败后，尚未可靠确认的来源数据可能重放，恢复不是业务 exactly-once 保证，函数必须支持重复记录。

示例一：订单事件触发函数

配置数据来源与 SQL

数据来源名称为 `orders`，Payload 类型为 JSON。示例消息：

```
{ "order_id": "o001", "amount": 99.5 }
```

对应 JSON Schema：

```
{
  "type": "object",
  "properties": {
    "order_id": { "type": "string" },
    "amount": { "type": "number" }
  },
  "required": ["order_id", "amount"]
}
```

```
SELECT client_id,
       payload.order_id AS order_id,
       payload.amount AS amount
FROM orders
WHERE payload.amount > 0
```

配置输出动作

参数	填写方式	值
环境变量名称	输出模板	`SOURCE`
环境变量值	输出模板	`mqtt-rule`
触发内容	输出模板	`{"client":"${client_id}","order":"${order_id}","amount":${amount:0}}`

示例中的客户端 ID 和订单号必须是可直接放入 JSON 字符串的受控文本，金额必须为合法有限数值。任意文本不能直接套用此模板，不可控字符串请采用下方「透传已编码 JSON 文本」的方式。

函数处理示例

```
def main_handler(event, context):
    processed = 0
    for record in event["Records"]:
        order_id = record["order"]
        amount = record["amount"]
        # 演示读取记录；实际业务应在这里执行持久化幂等处理
        print({"order": order_id, "amount": amount})
        processed += 1
    return {"processed": processed}
```

此代码仅演示遍历结构，不提供持久化幂等或业务事务。实际幂等键可使用订单号结合事件类型或版本，避免把同一订单的不同合法事件误判为重复。

示例二：透传已编码的 JSON 文本

上游将每条待处理对象序列化为完整合法 JSON 文本。将数据来源 `json_events` 的 Payload 类型设置为 STRING，使 SQL 输出文本列，不进行 JSON 字段访问。
示例 Payload 的实际字节内容：

```
{ "order": "o001", "amount": 99.5 }
```

```
SELECT payload AS json_text
FROM json_events
```

触发内容选择**输出模板**，填写 `${json_text}` 。对上述一条记录，请求体为：

```
{
  "Records": [
    { "order": "o001", "amount": 99.5 }
  ]
}
```

STRING 来源不替上游校验 JSON，每条 Payload 必须预先保证是完整合法的 JSON 元素。不能传入已包含多个待批量记录的 `Records` 包装并期待动作自动展开，动作只把每行模板结果作为一个元素。

动态字段

最近更新时间：2026-11-18 15:11:01

输出动作中可以引用 SQL 结果的配置项称为**动态字段**，例如 CKafka 的 Topic、Key 和消息的值，MQTT 的目标 Topic 和 Payload，云函数 SCF 的触发内容。本文介绍动态字段的填写方式、引用规则、各类型的输出格式以及运行时处理规则。

填写方式

动态字段有两种填写方式。

填写方式	写法	结果	适用场景
输出字段	只能填一个完整的字段引用，例如 `${payload}`	原样使用该列的值和类型，**可以保留二进制内容**	原样透传某一列，例如转发原始 Payload
输出模板	固定文本和字段引用任意组合，例如 `alarm/${device_id}`，也可以只填固定文本	结果**始终是文本**	拼接 Topic、拼接 JSON 等

> **说明：**
> 输出模板即使只包含一个占位符（例如 `${device_id}`），仍属于输出模板，不会自动转换为输出字段

生成 SQL 输出列

动态字段读取的是 **SQL 转换结果中一行的顶层列**，不是输入消息本身。请先在 SQL 中选出需要的字段并指定输出列名，再在动作中引用这些列名。
假设数据来源名称为 `devices`，Payload 为 JSON，已确认的 JSON Schema 包含字符串 `deviceId` 和数值 `temperature`。示例消息如下：

```
{"deviceId":"sensor_01","temperature":26.5}
```

```
SELECT payload.deviceId AS device_id,
       payload.temperature AS temp,
       topic AS source_topic
FROM devices
```

动作可以引用 `${device_id}`、`${temp}`、`${source_topic}`。不能使用 `${payload.deviceId}`、`${payload[0]}`、`${a.b}` 访问嵌套数据；嵌套路径、数组访问、计算和条件判断都应放在 SQL 中，并为结

果设置明确的别名。

各动作字段支持的填写方式

动作	动作字段	输出字段	输出模板
CKafka	Kafka 主题名称	不支持	支持
CKafka	消息的键 Key	不支持	支持
CKafka	消息的值 Value	支持	支持
CKafka	Header 名称、Header 值	不支持	支持
MQTT	目标 Topic	不支持	支持
MQTT	Payload	支持	支持
MQTT	Response Topic	不支持	支持
MQTT	Correlation Data	不支持	支持
云函数 SCF	触发内容	不支持	支持
云函数 SCF	环境变量名称、环境变量值	不支持	支持

QoS、Payload Format Indicator、Message Expiry Interval、Content Type、压缩算法、连接与认证参数均为固定值，不是动态字段。CKafka 的 Key 和 Value、MQTT 的 Payload 可以不配置，不配置时的行为请参见对应动作文档。

字段引用写法

写法	含义
`$name`	引用列名 `name`，以首个非名称字符结束
`${name}`	引用列名 `name`，边界明确，**推荐使用**
`${name:默认值}`	该行列值为 NULL 时使用字面默认文本，仅输出模板支持

输出字段

输出字段只允许一个完整列引用，以下两种写法等价：

```
$body
```

```
${body}
```

不能加前后缀、多个引用、默认值或转义：`prefix-${body}`、`${body:empty}` 都不是合法的输出字段。

SQL 输出类型	输出字段行为
文本	原样发送文本
二进制	原样发送字节，仅限接受二进制的字段
数值、布尔、日期等标量	不接受；请先在 SQL 中转换为文本，或改用输出模板
对象、数组等嵌套类型	不接受；请先在 SQL 中提取支持的列
文本或二进制列中的 NULL	分别变为空字符串或空字节，不保留 NULL 标记

数值需要以文本原样输出时，请先在 SQL 中转换：

```
SELECT CAST(payload.temperature AS VARCHAR) AS temp_text FROM devices
```

原样输出完整 JSON 文本

上游已保证消息内容是完整合法 JSON 时，可将数据来源的 Payload 类型设置为 STRING，在 SQL 中原样选择文本列：

```
SELECT payload AS json_text FROM json_messages
```

MQTT Payload 或 CKafka Value 可选择输出字段 `${json_text}`；SCF 的触发内容只支持输出模板，应填写 `${json_text}`，**不要加额外引号**。该方式不做 JSON 解析、重新编码或补引号，要求上游发送的是完整合法的 JSON 元素，而不是未编码的普通字符串。
若数据来源为 JSON 类型且需要输出完整对象或数组，可在 SQL 中选择 `payload AS json_text`，但请先使用**测试 SQL** 确认该列的实际输出类型，再决定填写方式。

原样输出二进制数据

将数据来源的 Payload 类型设置为 BINARY，在 SQL 中把 `payload` 选为文本或二进制列，再把 MQTT Payload 或 CKafka Value 设为输出字段 `${body}`：

```
SELECT payload AS body FROM raw_messages
```

该方式保留原始字节。输出模板不支持二进制列，也不会隐式进行 UTF-8 解码或 Base64 编码，请勿把二进制列放入输出模板。

输出模板

模板由固定文本和占位符拼接，结果始终是文本：

```
telemetry/${device_id}
设备 ${device_id} 当前温度 ${temp} ℃
${source_topic}/${device_id}
```

各类型的输出格式

SQL 输出列类型	在模板中的输出格式
文本	原样输出，不自动转义
布尔	`true` 或 `false`
整数	十进制，例如 `42` 、 `-7`
浮点数	保留浮点形式，例如 `1.0` ；非有限值可输出 `NaN` 、 `inf` ，**不是合法 JSON 数值**
日期	`YYYY-MM-DD` ，例如 `1970-01-01`
时间（毫秒精度）	`HH:MM:SS[.mmm]` ，例如 `00:00:00.120`
时间戳（无时区、毫秒）	`YYYY-MM-DDTHH:MM:SS[.mmm]`
时间戳（UTC、毫秒）	RFC 3339 格式，末尾带 `Z` ，例如 `1970-01-01T00:00:00.123Z`
时长（毫秒）	ISO 8601 格式，例如 `PT0.001S` ；零时长为 `P0D`
二进制	**不支持**。请改用允许二进制的输出字段，或由上游显式编码为文本
对象、数组	**不支持**。请在 SQL 中取出需要的标量字段

NULL 值与默认值

默认值**仅在列存在且该行值为 NULL 时生效**，不替代空字符串、 `0` 、 `false` ，也不补救不存在的列。

模板	`name` 为 NULL	`name` 为 `sensor_01`
`${name}`	空字符串	`sensor_01`

`${name:unknown}`	`unknown`	`sensor_01`
`${name:}`	空字符串	`sensor_01`
`device/${name:unknown}`	`device/unknown`	`device/sensor_01`

`${temp:0}` 的默认值是字符 `0`，最终结果仍为文本。默认值不能嵌套占位符，`${name:${fallback}}` 、`${name:$fallback}` 均不支持。按另一列补值请在 SQL 中处理：

```
SELECT COALESCE(payload.deviceId, client_id) AS device_id,
       COALESCE(payload.temperature, 0) AS temp
FROM devices
```

转义字符

下表是**实际填写到模板中的文本**，不是接口层 JSON 字符串的编码形式。

模板中填写	输出字面字符
`\$`	`$`
`\\`	一个反斜杠 `\`
`\{`	`{`
`\}`	`}`
`\:`	`:`

输入模板： `price=\$10`
输出文本： `price=$10`

输入模板： `${name:x\:y}`
`name` **为** `NULL` **时：** `x:y`

输入模板： `\${name}`
输出文本： `${name}`

只有以上五种转义有效，`\n` 、`\t` 、`\"` 不是模板转义；末尾单个反斜杠也不合法。需要真实换行时，请在输入框中直接换行。

使用模板拼接 JSON

模板只拼接文本，不会自动给字符串加引号，也不会转义字段值里的引号、反斜杠或换行。展开后的字段内容不会被再次当成模板解析。
以下模板只适用于 `device_id` 为受控的字母、数字、下划线字符串，且 `temp` 为合法有限数值的输入：

```
SELECT payload.deviceId AS device_id,
       payload.temperature AS temp
FROM devices
WHERE payload.deviceId IS NOT NULL
```

```
{"device_id":"${device_id}","temperature":${temp:0}}
```

示例输出：

```
{"device_id":"sensor_01","temperature":26.5}
```

⚠ **注意：**

- 字符串字段需要自行加引号；数值字段为 NULL 时会留下缺失值，请使用默认值兜底，或在 SQL 的 `WHERE` 中排除 NULL。
- 字段内容包含双引号、反斜杠或换行时，外层加引号不能替代 JSON 转义，拼出的 JSON 可能不合法。`NaN`、`inf` 也不是合法 JSON 数值。
- 字段内容不可控时，优先透传已经合法编码的完整 JSON 文本：MQTT Payload 与 CKafka Value 可使用输出字段 `${json_text}`；SCF 触发内容只支持输出模板，应填写 `${json_text}`，不要加额外引号。

长度与数量限制

限制对象	上限	说明
单个模板原文	128 KB（131,072 字节）	按 UTF-8 字节计算，不是字符数
单个模板中的占位符出现次数	64 次	同一列重复引用也逐次计数
单行、单个模板字段展开结果	1 MB（1,048,576 字节）	包含固定文本、所有字段值和默认值

版权所有：腾讯云计算（北京）有限责任公司

模板原文或占位符次数超限，在保存配置时即被拒绝；展开结果超限发生在运行时，不会静默截断为有效内容，受影响记录按对应动作的错误策略处理。
上述限制针对**模板字段**，不是整个 MQTT 或 Kafka 消息、整个 SCF 请求体的总容量承诺。输出字段不经过同一套模板长度检查，但仍受目标服务、协议、批次和内存等限制。

配置检查与运行时处理

保存时检查：保存输出动作时，系统检查以下内容，不通过则无法保存：

- 引用的字段在 SQL 输出中是否存在，大小写是否一致。
- 输出字段是否只引用了一个字段，引用的字段类型是否支持。
- 模板语法是否正确，包括默认值中的字段引用。
- 结果类型是否符合目标字段的要求，例如 Kafka Topic 必须是文本。

运行时处理：每条 SQL 结果独立生成一条目标消息。某条结果渲染后不合法（例如生成的 Topic 不合法、超过长度上限）时，**只丢弃这一条**，不影响同批的其他结果，也不重试。丢弃数计入监控中的**动作失败数**。

监控与排障

最近更新时间：2026-11-18 15:11:01

规则引擎按数据来源、数据处理和输出动作三个环节提供监控指标。数据在哪个环节减少，就从哪个环节开始排查。本文介绍监控指标、失败处理机制和常见问题的排查方法。

查看监控

在**规则管理**列表中，单击规则**监控**列的图标，进入规则详情页的**运行监控**页签，查看该规则的监控数据。

运行监控页签包含以下内容：

- 页面顶部展示规则触发数、处理消息数、动作执行数、动作在途数四项概览，单击右上角的刷新图标可刷新数据。
- 下方监控大盘按**数据来源相关指标**、**数据处理相关指标**、**输出动作相关指标**三组展示全部指标的趋势图。
- 大盘上方可以选择时间范围（例如近1小时）、时间粒度（例如1分钟）和实时刷新频率，右侧的**指标目录**支持按名称搜索和快速定位指标。

环节	指标	含义
数据来源（source）	规则触发数（条）	进入规则引擎的消息总数。一条消息匹配多条规则时，每条规则分别计数
数据来源（source）	来源丢弃数（条）	未通过 Schema 校验而被丢弃、未进入 SQL 处理的消息数。MQTT 动作发出的消息（带防环路标记）被跳过时，也计入此项
数据处理（transform）	处理消息数（条）	通过 Schema 校验、进入 SQL 处理的消息数
数据处理（transform）	SQL 成功数（条）	SQL 执行成功且满足 `WHERE` 条件后输出的结果数。使用 `UNNEST` 展开数组时，一条消息可能输出多条结果
数据处理（transform）	SQL 失败数（条）	SQL 执行阶段被丢弃的消息数，可按丢弃原因的错误码查看，默认展示 Top 5。不满足 `WHERE` 条件属于正常过滤，**不计入**此项
输出动作（sink）	动作执行数（条）	SQL 产生结果后，提交给输出动作、准备写入目标系统的数量
输出动作（sink）	动作成功数（条）	目标系统确认写入成功的结果数。SCF 异步调用时，只表示函数平台已接收，不代表函数执行成功
输出动作（sink）	动作失败数（条）	未能写入目标系统并被丢弃的结果数

版权所有：腾讯云计算（北京）有限责任公司

输出动作（sink）	动作在途数（条）	已发往目标系统、尚未收到确认的结果数（包括正在重试的结果），可用于判断目标系统是否处理缓慢

> **说明：**
> - 规则**更新后**，监控数据从更新时刻重新开始统计，此前的数据不保留。

失败处理机制

规则引擎把运行中的失败分为三类，处理方式不同：

类型	典型原因	处理方式	数据结果	监控表现
暂时失败	下游繁忙、限流、网络抖动、服务暂时不可用、Producer 队列暂满	持续重试并退避；队列暂满时保留未发送记录并背压	不丢失，恢复后继续投递，**可能重复**	动作在途数持续不降，动作失败数不增加
数据错误	消息不符合 Schema、SQL 运行出错、消息过大、渲染结果不合法或超限	直接丢弃，不重试	**丢失**，无法找回	来源丢弃数、SQL 失败数或动作失败数增加
永久拒绝	目标 Topic 无权限、认证失败、记录被目标拒绝、HTTP 401/403/404 等	按各输出动作的错误分类处理：丢弃受影响的未确认记录，或使当前规则执行实例失败并由后台清理、尝试重建	已丢弃记录不补发；修正配置后仅新记录继续处理	动作失败数增加，动作成功数停滞

各输出动作的具体失败场景，请参见 输出到 CKafka 、输出到 MQTT 和 输出到云函数 SCF 中的**失败处理**。

常见现象

现象	含义
SQL 成功数明显少于处理消息数，但 SQL 失败数为0	正常现象：大部分消息不满足 `WHERE` 条件，被过滤。过滤不计入 SQL 失败数
SQL 成功数大于处理消息数	正常现象：SQL 使用了 `UNNEST`，一条消息展开成多条结果
SQL 失败数大于异常消息的数量	SQL 运行出错时，同批次的所有消息一起被丢弃，包括正常消息。请参见 SQL 语法 中的**编写健壮的 SQL**
动作在途数持续不降，动作失败数不增加	下游暂时不可用，正在重试。数据**没有丢失**，下游恢复后继续投递

动作在途数持续积压，动作成功数长时间不增长	可能是配置错误（Topic 不存在、无权限等），规则在反复重试恢复。请按下文**常见问题排查**处理

> ⚠ **注意：**
> 来源丢弃数、SQL 失败数、动作失败数的统计对象不同，**不能直接相加**作为总丢失数。例如，SQL 整批出错时，SQL 失败数按批次内的消息数统计。

常见问题排查

排查时，按数据在哪个环节减少逐段定位：

```
规则触发数是否 > 0?
  ├─ 否 → 规则没有收到数据，见「问题1」
  └─ 是 → 来源丢弃数是否明显增加?
            ├─ 是 → 见「问题2」
            └─ 否 → SQL 成功数是否 > 0?
                      ├─ 否 → 见「问题3」
                      └─ 是 → 动作成功数是否增长?
                                ├─ 否 → 见「问题4」
                                └─ 是 → 规则正常；下游没收到数据见「问题5」
```

问题1：规则触发数为0

规则没有收到数据。可能原因和处理方法如下：

可能原因	处理方法
规则已停用	在规则列表中确认启停开关已开启
修改后未单击**更新**	编辑规则后，需在列表中单击**更新**才生效
Topic 不匹配	核对数据来源的 Topic 和通配符，例如 `sensors/+` 不匹配 `sensors/a/b`
客户端事件来源选错事件类型	核对事件类型
刚创建、启用或更新后不足1分钟	稍等后重试

问题2：来源丢弃数增加

可能原因	处理方法

Payload 不是合法的 JSON	检查设备上报格式
缺少必填字段，或必填字段类型不匹配	检查 Schema 中的必填字段和类型；用实际消息在**测试 SQL** 中验证
数值转换失败，例如必填整数字段收到 `"1.5"`	调整字段类型（例如改为 number），或取消必填
消息由 MQTT 动作发出（带防环路标记）	属于预期行为，请参见 输出到 MQTT 中的**防环路机制**

问题3：SQL 成功数为0

可能原因	处理方法
所有消息都不满足 `WHERE` 条件（SQL 失败数为0）	用实际消息在**测试 SQL** 中验证条件
字段为 NULL 导致比较结果不成立	非必填字段缺失时为 NULL，`NULL > 30` 不成立
SQL 失败数增加	SQL 运行出错，例如除数为0、`CAST` 失败。可在监控中按错误码查看失败原因，并改用 `TRY_CAST`、`NULLIF`

问题4：动作成功数不增长

现象	可能原因	处理方法
动作在途数持续存在，下游有故障	下游暂时不可用	等待下游恢复，规则会自动继续投递
动作失败数增加，动作成功数停滞	目标 Topic 无权限、认证失败、函数或版本不存在、记录被目标永久拒绝	按对应输出动作的**失败处理**确认错误分类，修正连接、权限或目标资源配置；已丢弃的记录不会自动补发。连接配置有误时，请参见 管理连接 中的**更换连接**
动作失败数增加	消息过大；渲染后的 Topic 不合法（例如包含通配符、为空）；SCF 返回非2xx 状态码	检查输出动作中的模板和字段值；检查函数日志

问题5：动作成功数正常，但下游没有收到数据

可能原因	处理方法

查看的 Topic 或实例不对	动态 Topic 可能写入了与预期不同的 Topic。确认连接指向的实例
MQTT 动作使用 QoS 0	QoS 0 的成功只表示已发出，订阅方离线时消息不保留
期望 MQTT 动作发出的消息继续触发其他规则	MQTT 动作发出的消息不会再触发规则
SCF 调用成功但业务未执行	异步调用成功只表示函数平台已接受事件，请在云函数控制台查看函数日志
SCF 函数没有遍历 `Records`	函数收到的事件是 `{"Records":[...]}`，请参见 **输出到云函数 SCF** 处理

使用限制

最近更新时间：2026-11-18 15:11:01

本文介绍规则引擎在 Beta 公测阶段的配额、范围限制、数值限制。

配额

项目	限制
每个集群的规则数	10 条
每个集群的连接数	10 个
每条规则的数据来源	1 个
每条规则的 SQL	1 段
每条规则的输出动作	1 个

范围限制

项目	限制
CKafka 目标	仅支持同地域的 CKafka 实例，不支持开启 SASL 鉴权的实例
MQTT 目标	仅支持同地域的 MQTT 实例
连接修改	不支持修改已创建的连接
发布协议	MQTT 动作以 MQTT 5 发布，retain 固定为 false

数值限制

项目	限制
规则描述长度	128 个字符
规则名称长度	128 字节
数据来源名称长度	128 个字符
单条消息大小	不单独限制，受集群消息大小限制

版权所有：腾讯云计算（北京）有限责任公司

单个模板长度	128 KB
单个模板中的字段引用数	64 个

版权所有：腾讯云计算（北京）有限责任公司