事件总线

事件源

产品文档

腾讯云

文档目录

事件源
事件源概述

最近更新时间：2024-01-22 20:52:28

事件源指事件的来源，负责将生产的事件发布到事件总线 EventBridge。发布到事件总线 EventBridge 的事件格式均遵循 CloudEvents 1.0 规范，具体可参见 事件结构 。
腾讯云事件总线 EventBridge 支持以下方式接入事件源：

腾讯云服务

当希望使用腾讯云服务产生的**监控事件**与**审计事件**作为事件源接入时，只需开通相应的腾讯云服务，即可自动接入事件总线 EventBridge。通过配置预定义的事件模式和事件目标，轻松完成将事件源发布到云服务总线，经过事件模式过滤后将事件路由到事件目标。目前支持告警监控事件与审计运维事件的投递管理。详情见 官方云服务事件。

自定义应用

当希望自定义业务作为事件源接入时，您需要配置自己的应用使用 API/SDK 接入事件总线 EventBridge。通过创建自定义总线、配置自定义事件模式和事件目标，将您自己的应用产生的事件发布到自定义总线，经过自定义事件模式过滤后将事件路由到事件目标。详情见 自定义事件。

连接器

连接器用于从**消息队列 TDMQ** 等事件源中主动拉取事件，并将事件以**标准化的格式**推送到自定义事件集中。您无需关心底层的消费投递逻辑，通过在自定义事件集中绑定一个或多个连接器，即可以实现自动拉取消息队列、网关的事件内容，并推送至指定事件集。详情见 连接器 文档。

官方云服务事件源

最近更新时间：2024-01-22 20:52:28

注意：

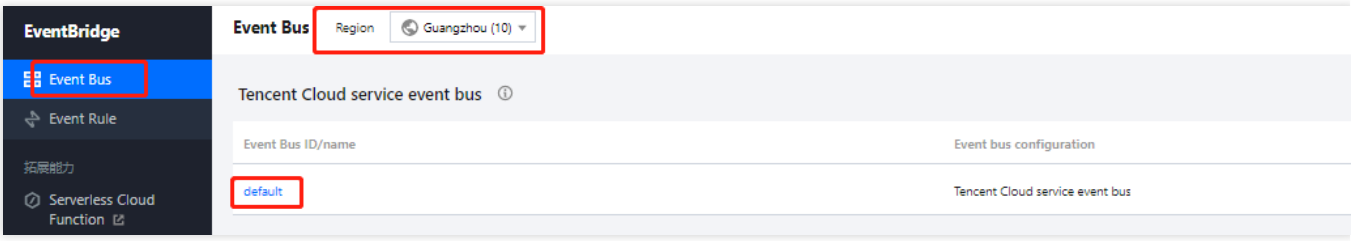
对于云产品产生的告警、审计等运维事件，将全部投递至云服务事件集，该投递为默认投递，不支持更改或编辑。您可以在事件总线控制台，为云服务事件集绑定相应的规则和目标，完成云服务事件的分发处理。

当希望使用官方云服务产生的**监控事件**（如云服务器的内核故障、内存oom等）与**审计事件**（Coming Soon）作为事件源接入时，只需开通相应的腾讯云服务，即可自动接入事件总线 EventBridge。官方云服务事件将被默认投递到**云服务事件集**。具体请参见 云服务事件。

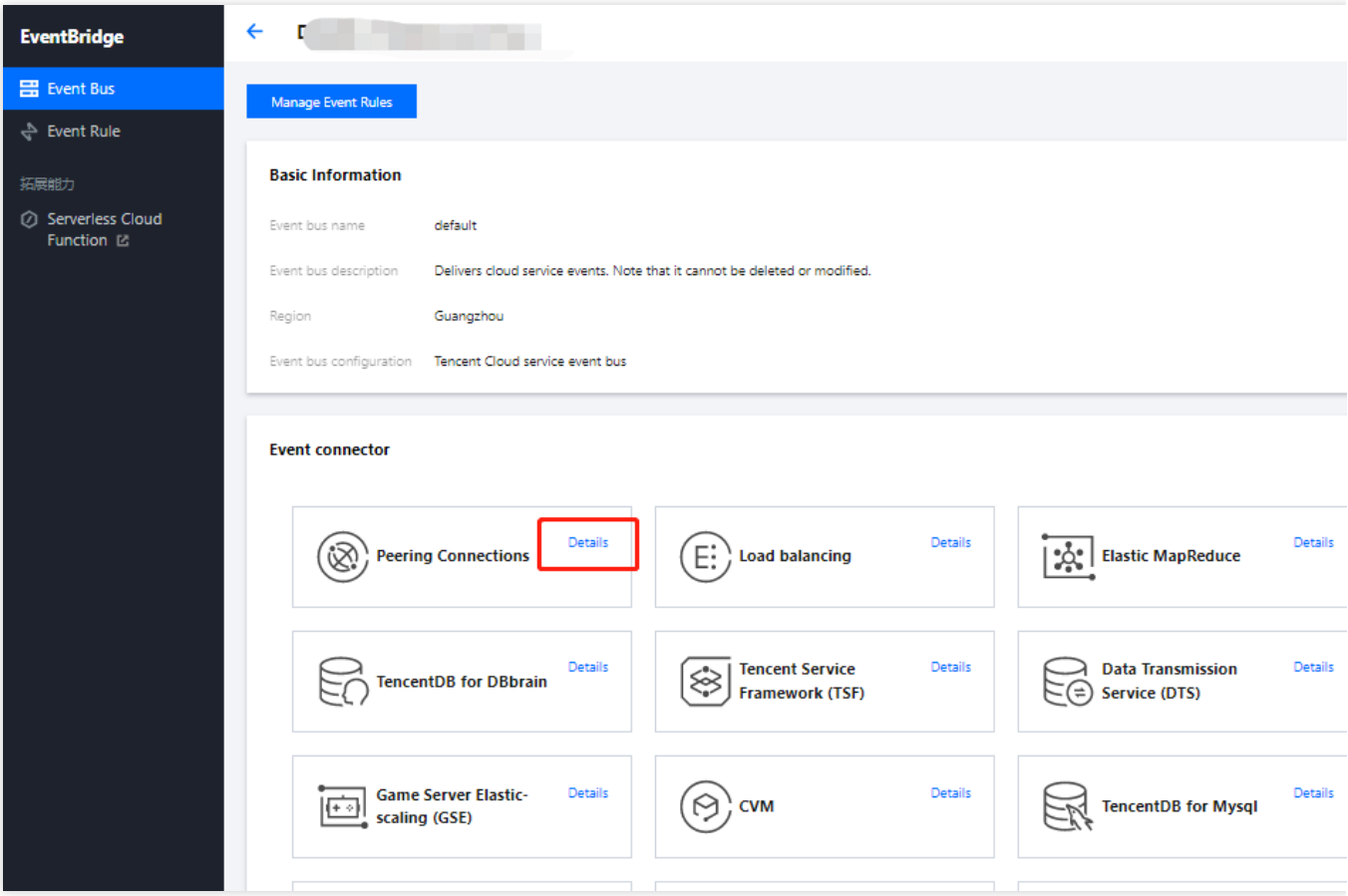
操作步骤

您可以通过以下步骤查看当前支持的所有云服务 (云监控) 数据源。

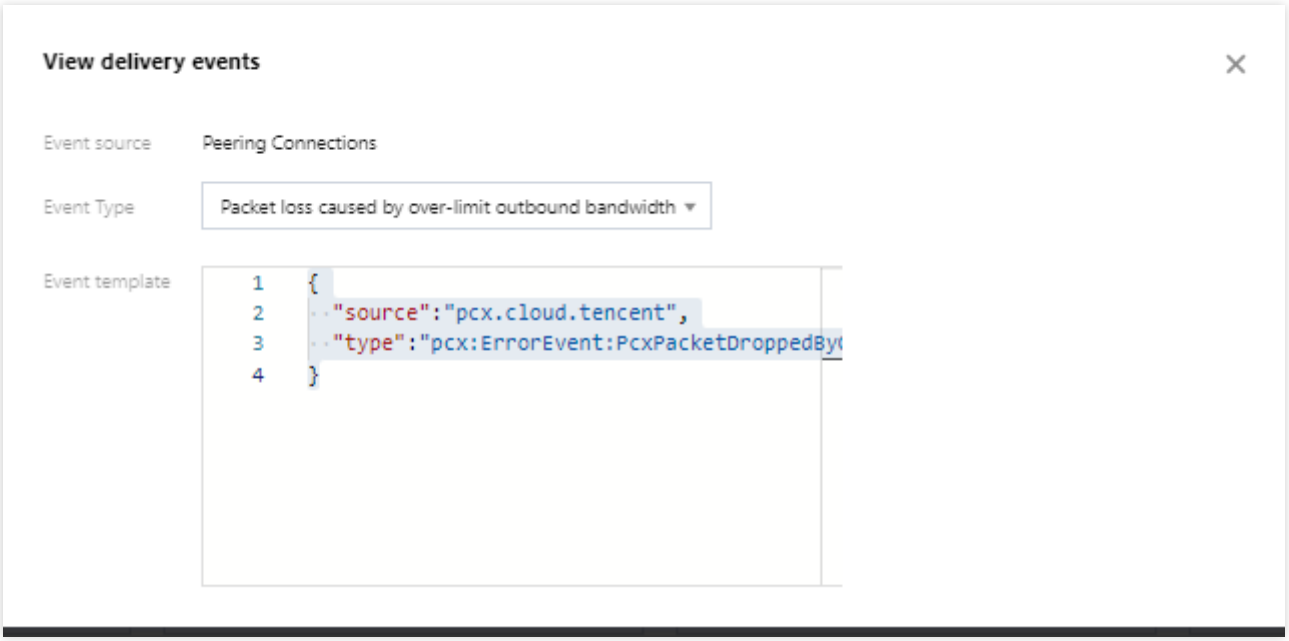
1. 登录 事件总线控制台，进入**云服务事件集default**，查看目前已经接入云服务事件集的云服务事件。



2. 在**事件连接器**中可以查看目前所有支持告警事件推送的云服务：



3. 单击详情后可以看到目前支持的所有告警事件类型：



支持事件源列表

云监控告警事件

云审计服务事件

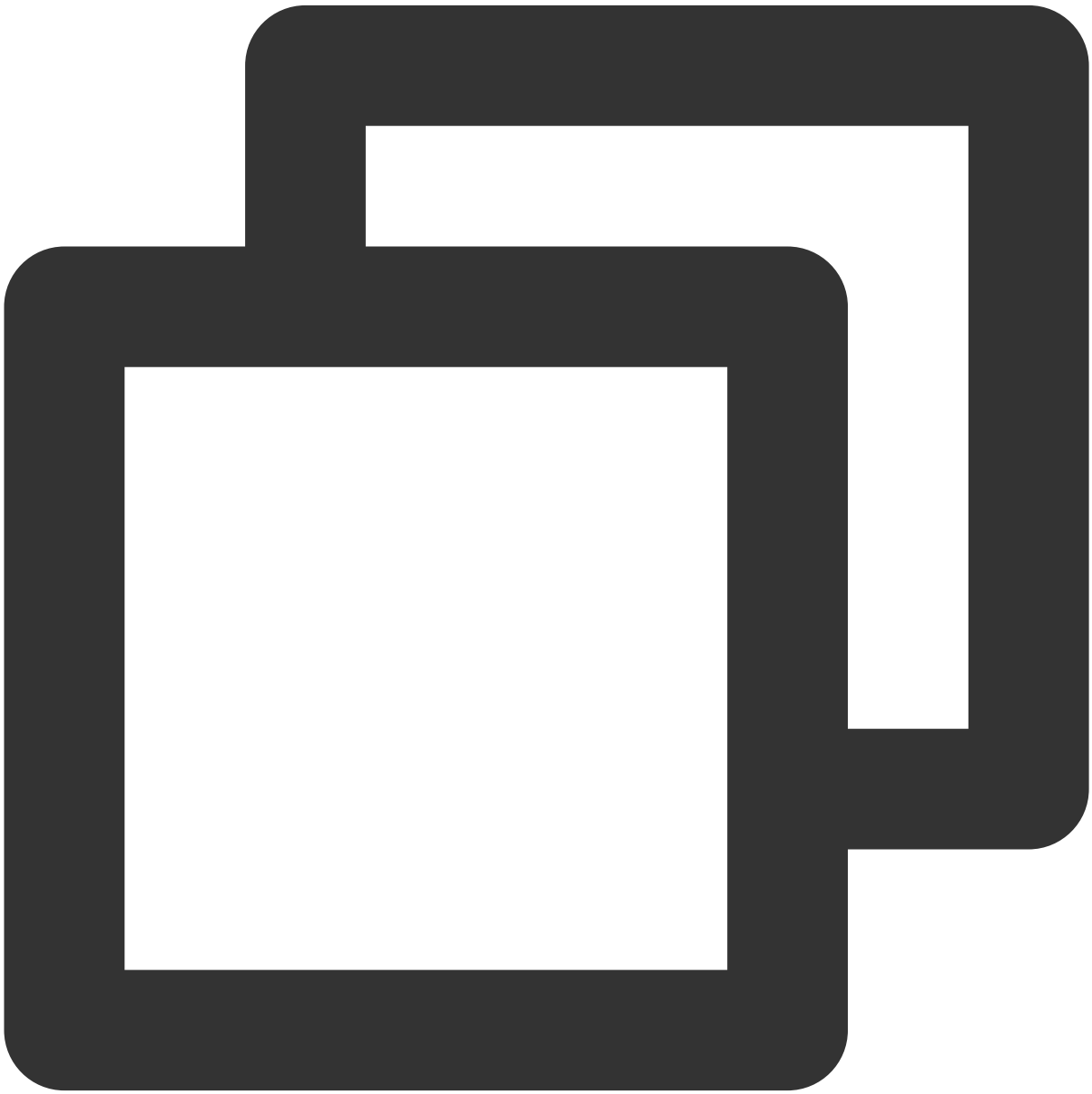
事件结构

最近更新时间：2024-01-22 20:52:28

事件是状态变化的数据记录。本文介绍事件总线 EventBridge 的事件参数详情。

事件源发布事件到事件总线 EventBridge 需要按照 CloudEvents 规范。了解 CloudEvents 规范的更多信息，请参见 CloudEvents 1.0。

以下是事件源发布到事件总线 EventBridge 的示例结构：



```
{
```

```
    "specversion":"0",
    "id":"13a3f42d-7258-4ada-da6d-023a333b4662",
    "type":"cos:created:object",
    "source":"cos.cloud.tencent",
    "subjuect":"qcs::cos:ap-guangzhou:uid1250000000:bucketname",
    "time":"1615430559146",
    "region":"ap-guangzhou",
    "datacontenttype": "application/json;charset=utf-8",
    "data":{
       $data_value
    }
}
```

事件涉及的参数如下说明：

字段	描述	字符串类型
specversion	事件结构体版本（cloudevents 遵循版本，目前为1.0.2）。	String
id	PUT Event 返回的 ID 信息。	String
type	PUT Event 输入的事件类型。云服务默认写 COS:Created:PostObject，用“:” 分割类型字段。	String
source	事件来源（云服务事件必传此参数，为 subjuect 的缩写 ）。云服务默认为 `xxx.cloud.tencent` 。	String
subjuect	事件来源详情可自定义，云服务默认使用 QCS 描述，例如 `qcs::dts:ap-guangzhou:appid/uin:xxx` 。	String
timer	发生事件的时间，0时区毫秒时间戳，例如1615430559146。	Timestamp
datacontenttype	数据类型申明。	String
region	地域信息。	String
data	PUT Event 输入的事件详情。	String

事件源发布到事件总线 EventBridge 的事件有以下两种类型：

腾讯云服务事件

腾讯云服务作为事件源自动接入事件总线 EventBridge。

自定义应用事件

您的应用作为事件源接入时，需要配置应用使用 API/SDK 接入事件总线 EventBridge。

连接器
连接器概述

最近更新时间：2024-01-22 20:52:28

连接器用于从事件源中主动拉取事件，并将事件以**标准化的格式**推送到自定义事件集中。您无需关心底层的消费投递逻辑，通过在事件集中绑定一个或多个连接器，即可以实现自动拉取消息队列、网关的事件内容，并推送至指定的自定义事件集。

事件总线 EventBridge 的连接器支持以下事件源：

事件源	配置方法
API 网关（APIGW）HTTP	APIGW 连接器
消息队列 Ckafka	Ckafka 连接器
DTS 数据订阅	DTS 连接器

事件总线 EventBridge 提供以下连接器管理能力：

创建连接器

查看连接器

编辑连接器

启动连接器

暂停连接器

删除连接器

配置 APIGW 连接器

最近更新时间：2024-01-22 20:52:28

操作场景

您可以通过配置 APIGW 来处理 WebHook 的投递事件，通过第三方 WebHook 接收其他系统产生的事件，APIGW 连接器是 HTTP 场景下理想的跨端接收事件的连接器。
APIGW 连接器实现方式为 **Push 模型**，APIGW 会监控请求并生成调用事件投递至事件集，并将相关事件通过事件规则路由到更多服务。

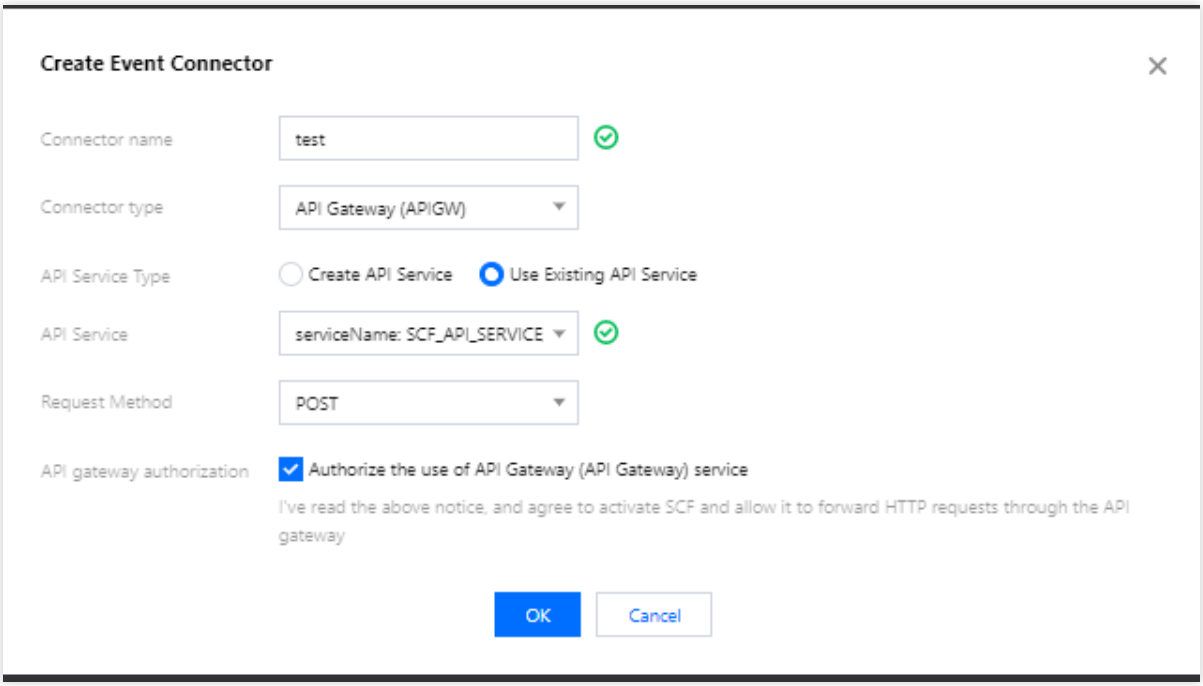
前提条件

已 创建事件集。

操作步骤

1. 登录 事件总线控制台，选择左侧导航栏中的**事件集**。
2. 在“事件集”列表，选择期望配置 APIGW 连接器的事件集。
3. 在“事件集详情”页事件连接器配置项中单击**添加**，如下图所示：

4. 根据页面提示填写相关信息，如下图所示：

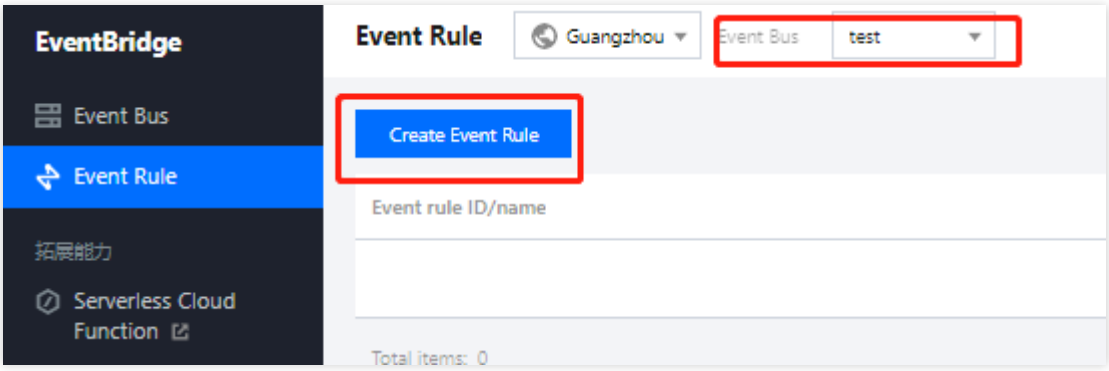


其中**连接器类型**选择 **API网关(APIGW)**连接器类型。

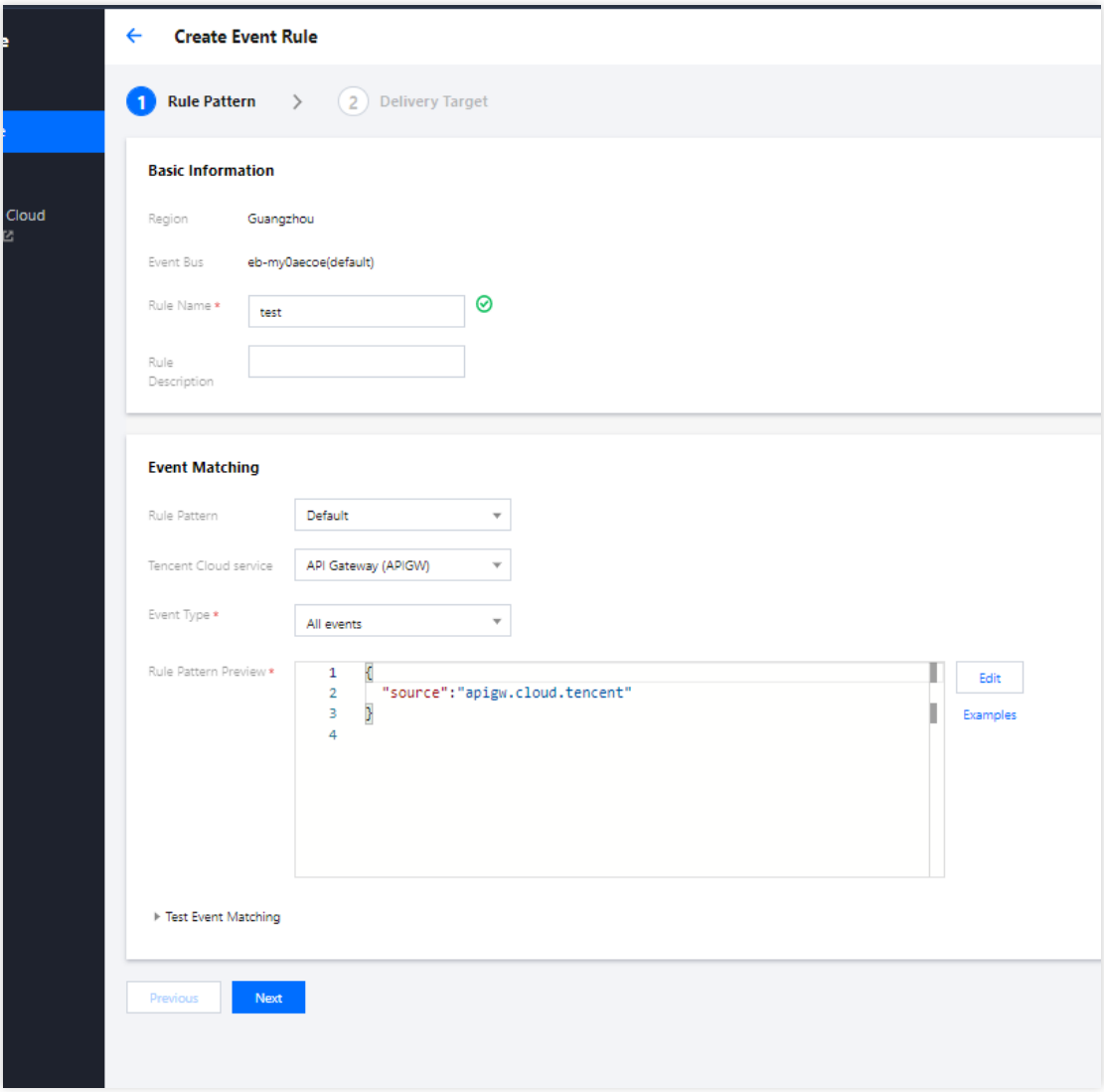
5. 单击**确定**完成创建。

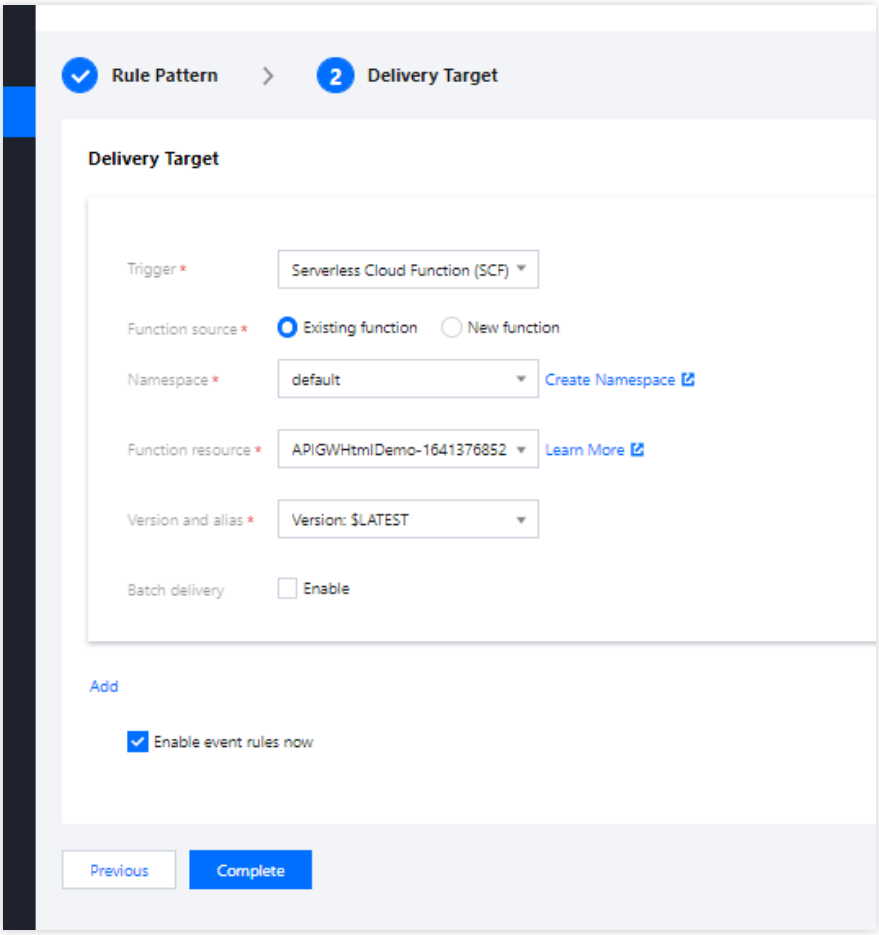
6. 选择左侧导航栏中的**事件规则**。

7. 在“事件规则”顶部选框，选择与之前创建一致的事件集信息，并单击**新建事件规则**，如下图所示：



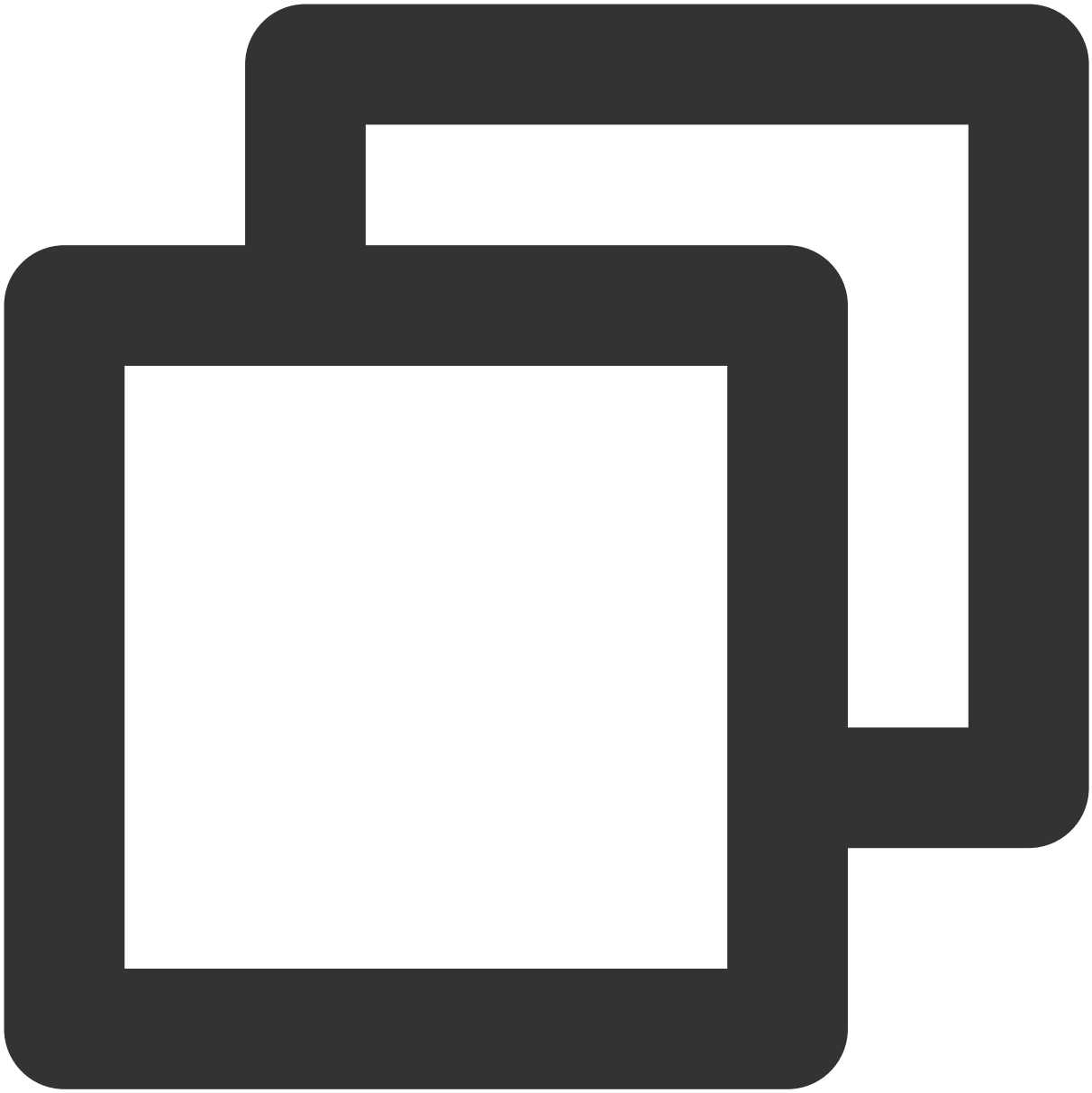
8. 根据页面提示填写相关信息，如下图所示


Create Event Rule
1 Rule Pattern
2 Delivery Target
Basic Information
Region
Guangzhou
Event Bus
eb-my0aecoe(default)
Rule Name *
test
Rule Description
Event Matching
Rule Pattern
Default
Tencent Cloud service
API Gateway (APIGW)
Event Type *
All events
Rule Pattern Preview *
{
"source":"apigw.cloud.tencent"
}
Edit
Examples
Test Event Matching
Previous
Next
Cloud



9. 其中**云服务类型**选择 **API网关(APIGW)**，并配置触发目标端。
10. 单击**确定**即可创建 APIGW 连接器。

APIGW 连接器数据结构说明



```
{
    "specversion": "1.0",
    "id": "13a3f42d-7258-4ada-da6d-023a333b4662",
    "type": "connector:apigw",
    "source": "apigw.cloud.tencent",
    "subjuect": "qcs::apigw:ap-guangzhou:uid1250000000/appidxxx:Serverid/Appid",
    "time": "1615430559146",
    "region": "ap-guangzhou",
    "datacontenttype": "application/json;charset=utf-8",
    "data": {
        "headers": {
```

```
            "Accept-Language": "en-US,en,cn",
            "Accept": "text/html,application/xml,application/json",
            "Host": "service-3ei3tii4-251000691.ap-guangzhou.apigateway.myqloud.com
            "User-Agent": "User Agent String"
        },
        "body": "{\\"test\\":\\"body\\"}",
        "stageVariables": {
            "stage": "release"
        },
        "path": "/test/value",
        "queryString": {
            "foo": "bar",
            "bob": "alice"
        },
        "httpMethod": "POST"
    }
}
```

参数说明如下：

参数	描述
path	记录实际请求的完整 Path 信息。
httpMethod	记录实际请求的 HTTP 方法。
queryString	记录实际请求的完整 Query 内容。
body	记录实际请求转换为 String 字符串后的内容。
headers	记录实际请求的完整 Header 内容。

配置 Ckafka 连接器

最近更新时间：2024-01-22 20:52:28

操作场景

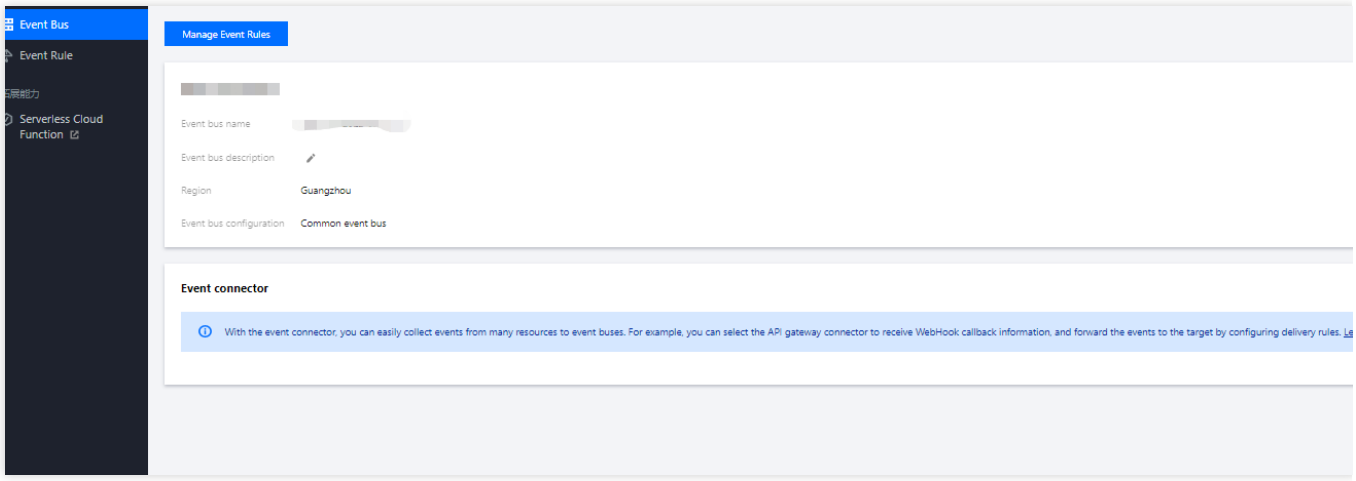
您可以通过配置 Ckafka 连接器来消费 Ckafka 消息队列的内容。Ckafka 连接器实现方式为 **Pull 模型**，事件连接器会主动拉取 Ckafka 内容，并将相关事件通过事件规则路由到更多服务。本文为您介绍如何创建 Ckafka 连接器及 Ckafka 连接器生成的事件结构。

前提条件

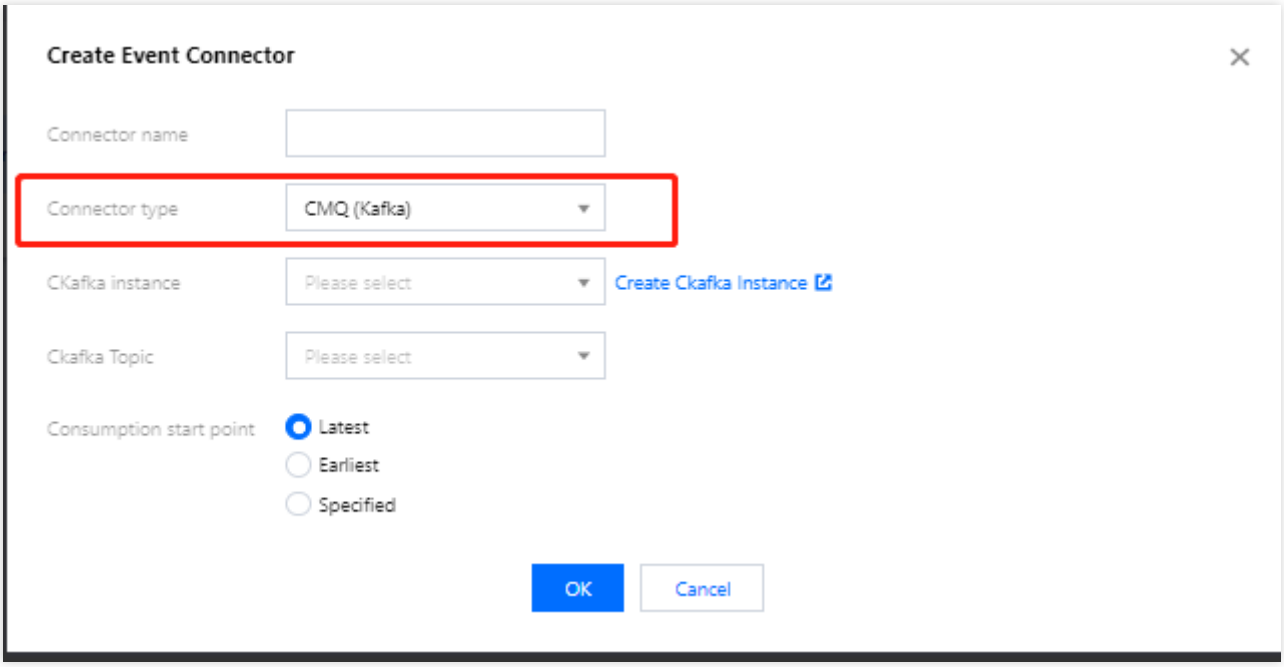
已 创建事件集。

操作步骤

1. 登录 事件总线控制台，选择左侧导航栏中的**事件集**。
2. 在“事件集”列表，选择期望配置 Ckafka 连接器的事件集。
3. 在“事件集详情”页事件连接器配置项中单击**添加**，如下图所示：



4. 根据页面提示填写相关信息，如下图所示：

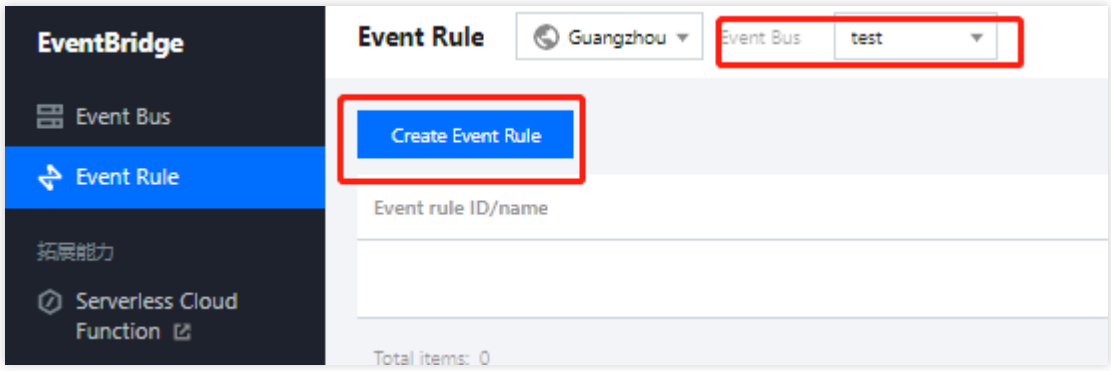


其中**连接器类型**选择 **消息队列（Kafka）** 连接器，其余配置项按照提示填写。

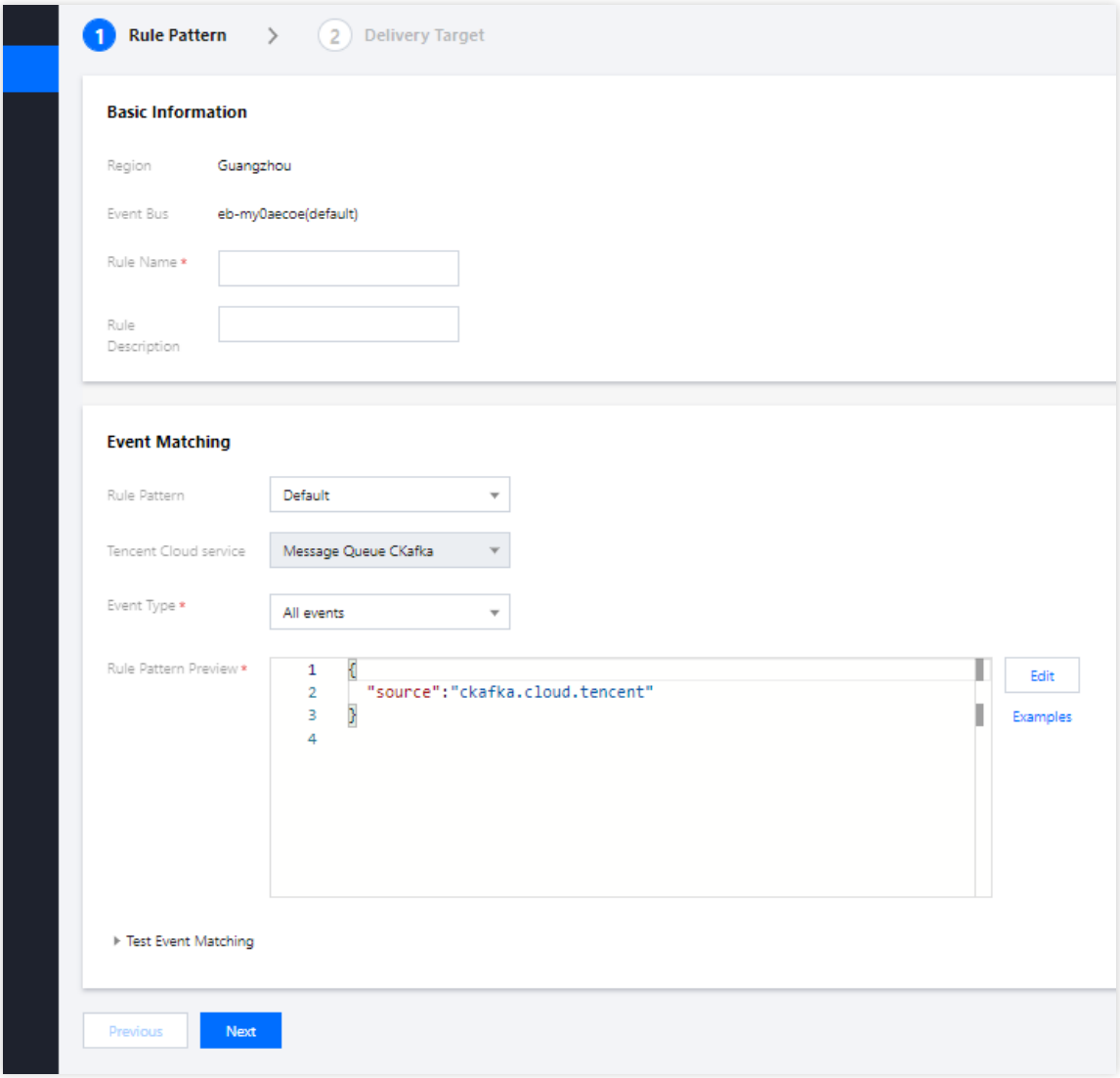
注意：

目前只支持云上 Ckafka 实例投递，请确认您的 Ckafka 实例没有配置用户名密码等信息，否则连接器可能无法成功获取消息。

5. 单击**确定**完成创建。
6. 选择左侧导航栏中的**事件规则**。
7. 在“事件规则”顶部选框，选择与之前创建一致的事件集信息，并单击**新建事件规则**，如下图所示：



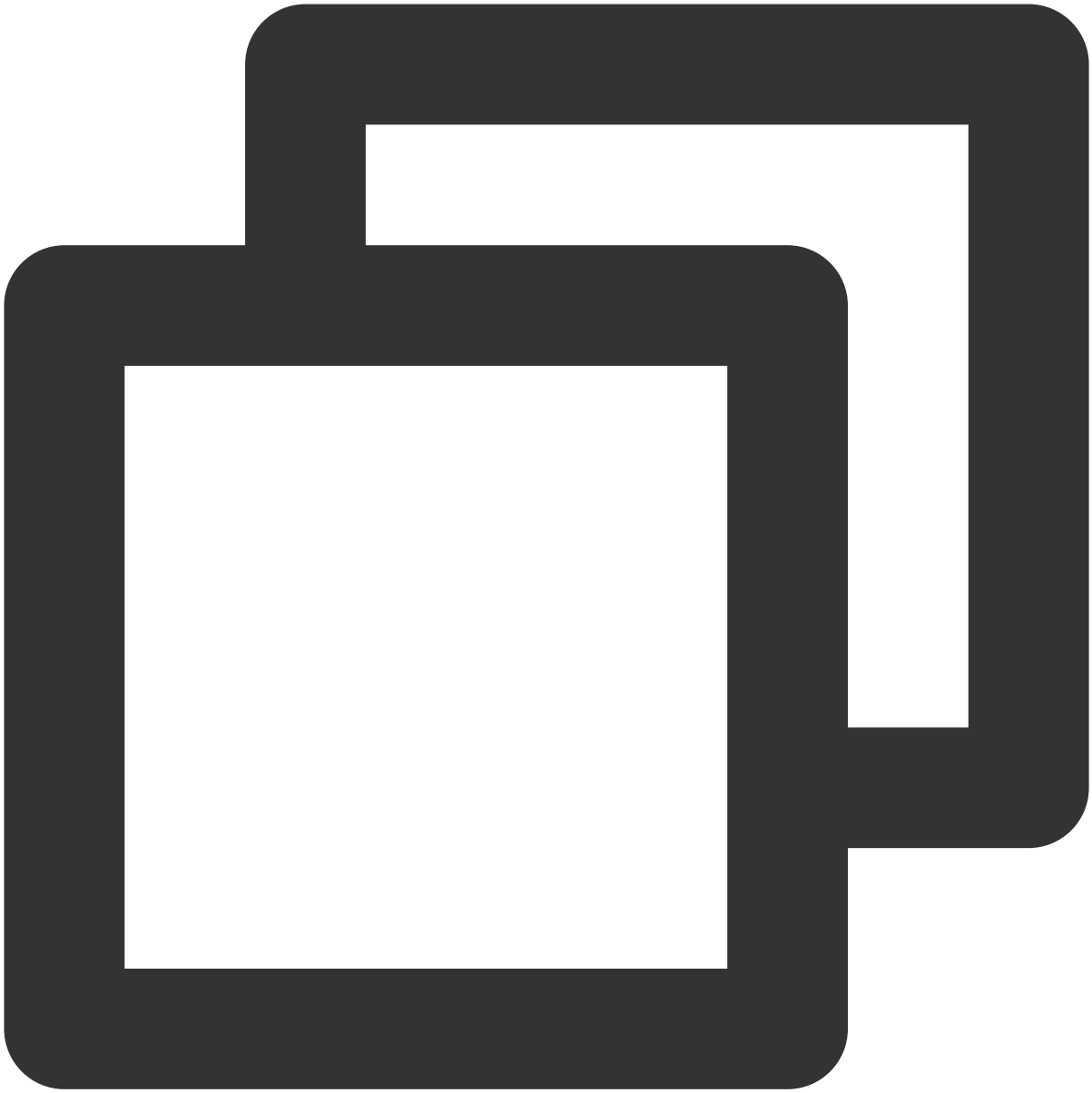
8. 根据页面提示填写相关信息，如下图所示：



其中**云服务类型**选择 **消息队列（CKafka）**，并配置触发目标端。

9. 单击**确定**即可创建 Ckafka 连接器。

Ckafka 连接器生成的事件结构说明



```
{
        "specversion":"1.0",
        "id":"13a3f42d-7258-4ada-da6d-023a333b4662",
        "type":"connector:kafka",
        "source":"ckafka.cloud.tencent",
        "subjuect": "qcs::ckafka:ap-guangzhou:uin/1250000000:ckafkaId/uin/125000000
        "time":"1615430559146",
        "region":"ap-guangzhou",
        "datacontenttype":"application/json;charset=utf-8",
        "data":{
             "topic":"test-topic",
```

```
            "Partition":1,
            "offset":37,
            "msgKey":"test",
            "msgBody":"Hello from Ckafka again!"
        }
}
```

参数说明如下：

参数	描述
topic	Ckafka 投递 Topic。
Partition	事件源所在分区，一个 Topic 可以包含一个或者多个 Partition，CKafka 以 Partition 作为分配单位。
offset	消费分组，指定消费区域。
msgKey	Ckafka 消息 Key。
msgBody	Ckafka 消息体。

配置 DTS 连接器

最近更新时间：2024-01-22 20:52:28

操作场景

数据传输服务（Data Transmission Service，DTS）支持 MySQL、MariaDB、PostgreSQL、Redis、MongoDB 等多种关系型数据库及 NoSQL 数据库数据订阅，数据库中关键业务的数据变化信息，并提供给下游进行业务订阅、获取和消费，方便用户搭建云数据库和异构系统之间的数据同步，如缓存更新，ETL（数据仓库技术）实时同步，业务异步解耦等。
在事件总线（EventBridge）中，您可以通过配置 DTS 连接器，基于 DTS 数据订阅实时拉取源实例的 Binlog 增量日志，完成日志的消费与处理，并实现下游不同目标的分发。本文为您介绍如何创建 DTS 连接器及 DTS 连接器生成的事件结构。

功能介绍

更多详情，请参考 DTS 数据订阅产品文档。

支持数据库

源数据库类型	源数据库版本	支持订阅的数据类型
MySQL	MySQL 5.5.x、5.6.x、5.7.x、8.0.x	数据更新 结构更新 全实例
MariaDB	MariaDB 10.0.10、MariaDB 10.1.9	数据更新 结构更新 全实例
MariaDB（Percona）	MariaDB（Percona）5.6.x、5.7.x	数据更新 结构更新 全实例
TDSQL MySQL	TDSQL MySQL 5.6.x、5.7.x、8.0.18	数据更新 结构更新 全实例
TDSQL-C MySQL	TDSQL-C MySQL 5.7.x、8.0.x	数据更新 结构更新 全实例

TDSQL PostgreSQL 版	PostgreSQL 版	数据更新

支持订阅操作

DTS 支持订阅对象选择的粒度为库、表，具体支持如下三种订阅类型：
数据更新：指订阅 DML 操作。
结构更新：指订阅 DDL 操作。
全实例：指订阅所有库表的 DML 以及 DDL 操作。

限制说明

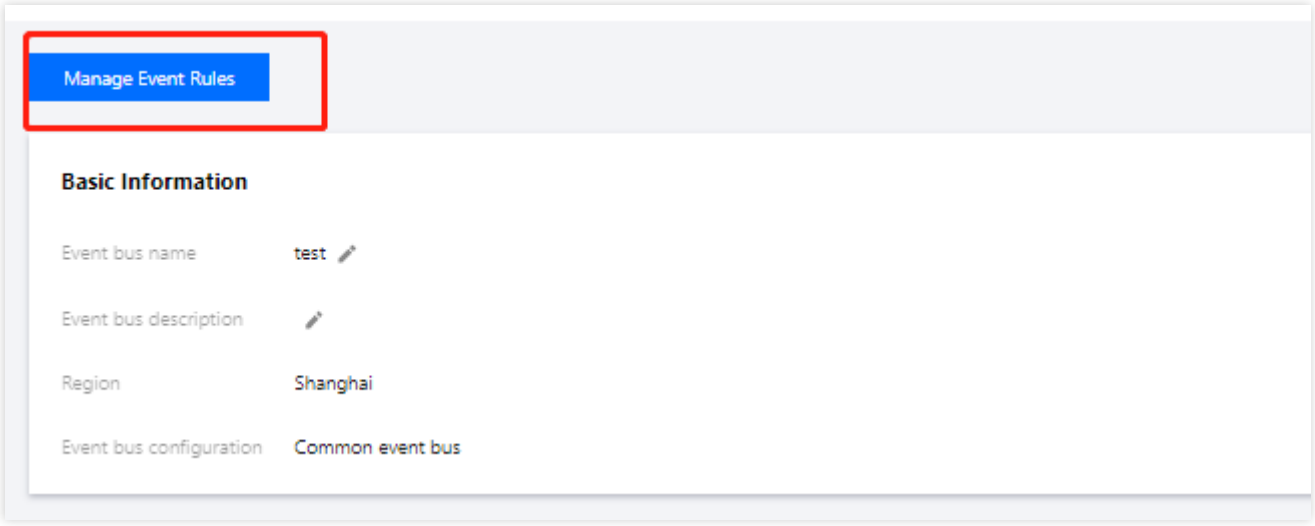
EB 侧限制事件大小，如果上游日志大小超过 **1 MB**，将无法成功投递至 EB 完成消费，请注意控制您的日志大小。
当前方案下，EB 对于多 Partition 会并发消费，无法保证顺序性。
为保证您的数据可以正常消费，创建成功后，请谨慎更新消费组账号密码信息，如果更新后，请重新绑定连接器，否则可能无法正常完成消费。
DTS 支持批量投递，对于批量变更的操作，将会在同一条事件中以数组格式完成投递。

前提条件

1. 已 开通 DTS 数据订阅服务 并创建实例。
2. **子账号**需要通过主账号获取 EventBridge 和 DTS 的相关操作权限。

操作步骤

1. 登录 事件总线控制台，选择左侧导航栏中的**事件集**。
2. 在“事件集”列表，选择期望配置 DTS 连接器的事件集。
3. 在“事件集详情”页事件连接器配置项中单击**添加**，连接器类型选择**数据订阅(DTS)**。
4. 根据页面提示选择需要消费的数据订阅实例，并填入消费组名称、账号、密码等信息。如果还没有完成消费对象绑定，请在 DTS 控制台完成相关配置。
5. 单击**确定**完成创建，在事件集页面查看您绑定的连接器信息。
6. 在事件集详情页中，单击**管理事件规则**。如下图所示：

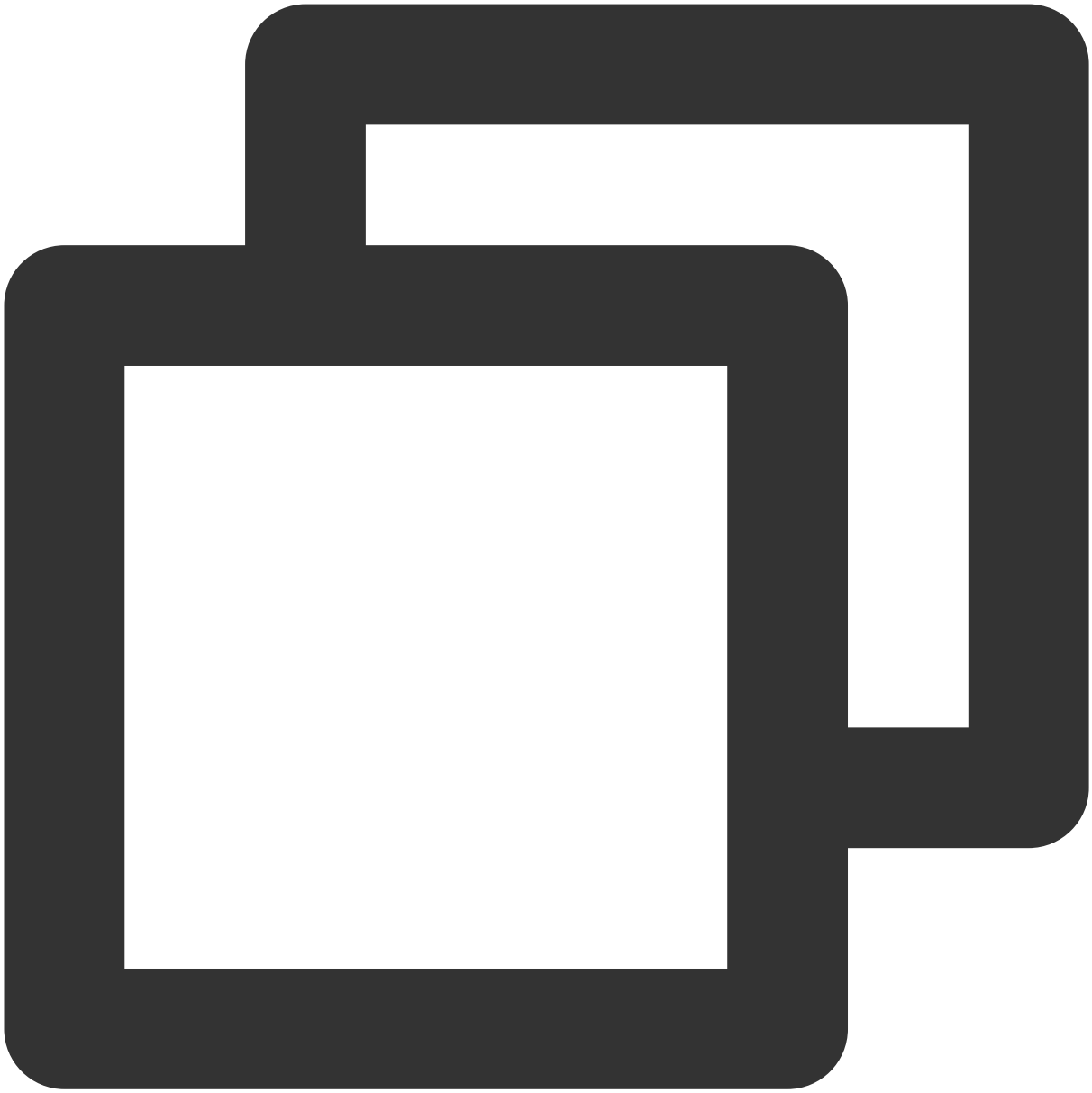


7. 单击**新建事件规则**，根据页面提示填写相关信息。
其中**云服务类型**选择**数据订阅（DTS）**，并根据实际需求，配置 数据转换 和绑定触发目标。
8. 单击**确定**即可完成创建。

事件格式

DTS 连接器接收处理后的事件格式如下：

DDL 操作示例




```
{
  "id":"38cecd93-a9c2-11ec-b952-******d8da53:16",
  "type":"dts:MYSQL:INSERT",
  "specversion":"1.0",
  "source":"dts.cloud.tencent",
  "subject":"cdb-xxxx",
  "time":1648109734,
  "region":"ap-guangzhou",
  "datacontenttype":"application/json;charset=utf-8",
  "tags":null,
  "data":{
```

```
        "topic":"topic-subs-xxxx-cdb-xxxx",
        "partition":3,
        "offset":16005,
        "partition_seq":16006,
        "event":{
           "dmlEvent":{
              "columns":[
                 {
                    "name":"string",
                    "originalType":"text"
                 },
                 {
                    "name":"int",
                    "originalType":"tinyint(4)"
                 },
                 {
                    "name":"time",
                    "originalType":"time"
                 },
                 {
                    "name":"double",
                    "originalType":"double"
                 },
                 {
                    "name":"id",
                    "originalType":"int(11)",
                    "isKey":true
                 },
                 {
                    "name":"float",
                    "originalType":"float"
                 },
                 {
                    "name":"longtext",
                    "originalType":"longtext"
                 }
              ],
              "rows":[
                 {
                    "newColumns":[
                       {
                          "dataType":13,
                          "charset":"utf8",
                          "bv":"dG1w"
                       },
                       {
```

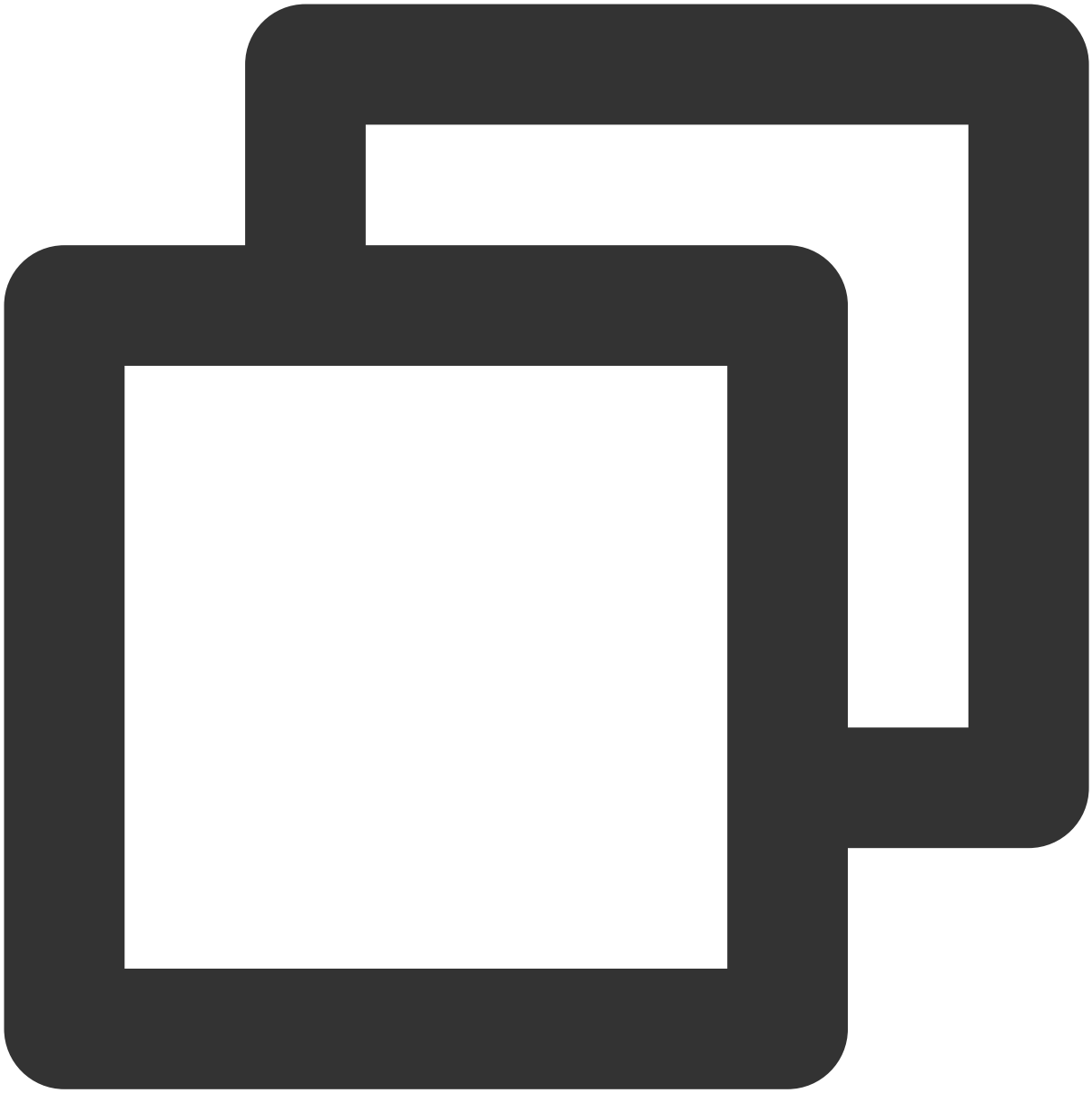
```
                    },
                    {

                    },
                    {
                        "dataType":10,
                        "sv":"1"
                    },
                    {
                        "dataType":3,
                        "sv":"3"
                    },
                    {

                    },
                    {

                    }
                ]
            }
        ]
    }
},
"header":{
    "sourceType":1,
    "messageType":2,
    "timestamp":1648109734,
    "serverId":109741,
    "fileName":"mysql-bin.000005",
    "position":2234587,
    "gtid":"38cecd93-a9c2-11ec-b952-******d8da53:16",
    "schemaName":"dts",
    "tableName":"dts_mysql",
    "seqId":16017,
    "isLast":true
},
"eb_consumer_time":"2022-03-24 16:15:34.287359965 +0800 CST m=+1120.357657669
"eb_connector":""
}
}
```


DML 操作示例




```
{
    "id":"38cecd93-a9c2-11ec-b952-******8da53:19",
    "type":"dts:MYSQL:DDL",
    "specversion":"1.0",
    "source":"dts.cloud.tencent",
    "subject":"cdb-xxxx",
    "time":1648110060,
    "region":"ap-guangzhou",
    "datacontenttype":"application/json;charset=utf-8",
    "tags":null,
    "data":{
```

```
        "topic":"topic-subs-aniwxeewm4-cdb-xxxx",
        "partition":0,
        "offset":16065,
        "partition_seq":16066,
        "event":{
            "ddlEvent":{
                "schemaName":"dts",
                "sql":"ALTER TABLE `dts_mysql` ADD COLUMN `t` tinyint (0) NULL , ADD UN
            }
        },
        "header":{
            "sourceType":1,
            "messageType":3,
            "timestamp":1648110060,
            "serverId":109741,
            "fileName":"mysql-bin.000005",
            "position":2235430,
            "gtid":"38cecd93-a9c2-11ec-b952-******d8da53:19",
            "seqId":16087,
            "isLast":true
        },
        "eb_consumer_time":"2022-03-24 16:21:01.19682088 +0800 CST m=+1447.267118604"
        "eb_connector":""
    }
}
```


参数说明

参数	描述
id	事件 ID，EB 自动生成，每条事件在 EB 内的唯一标识。
type	事件类型，三段式形式，对 dts 连接器，格式为 `dts:${数据库类型}:${操作类型}` 。
specversion	Cloudevents 版本，默认 1.0，EB 自动生成。
source	事件来源，对 dts 连接器，统一为 `dts.cloud.tencent` 。
subject	事件产生具体实例，对 dts 连接器，为数据订阅绑定的数据库实例 ID。
time	事件投递到 EB 的时间。
region	事件产生地域。
tags	资源标签。
data	数据库实际 binlog 日志内容。

data.topic	数据订阅实例信息。
data.partition	消费 partition。
data.offset	消费位点。
data.event	分为 dmlEvent 和 ddlEvent，dmlEvent 介绍数据表的 schema 格式和更改内容，ddlEvent 介绍具体 sql 操作。
data.header	操作日志头部信息，包含数据库名称、表名称、变更时间戳等具体信息。

配置 TDMQ 连接器

最近更新时间：2024-01-22 20:52:28

操作场景

您可以通过配置 TDMQ 连接器来消费 TDMQ 消息队列的内容。TDMQ 连接器实现方式为 **Pull 模型**，事件连接器会主动拉取 TDMQ 内容，并将相关事件通过事件规则路由到更多服务。本文为您介绍如何创建 TDMQ 连接器和 TDMQ 连接器生成的事件结构。

注意

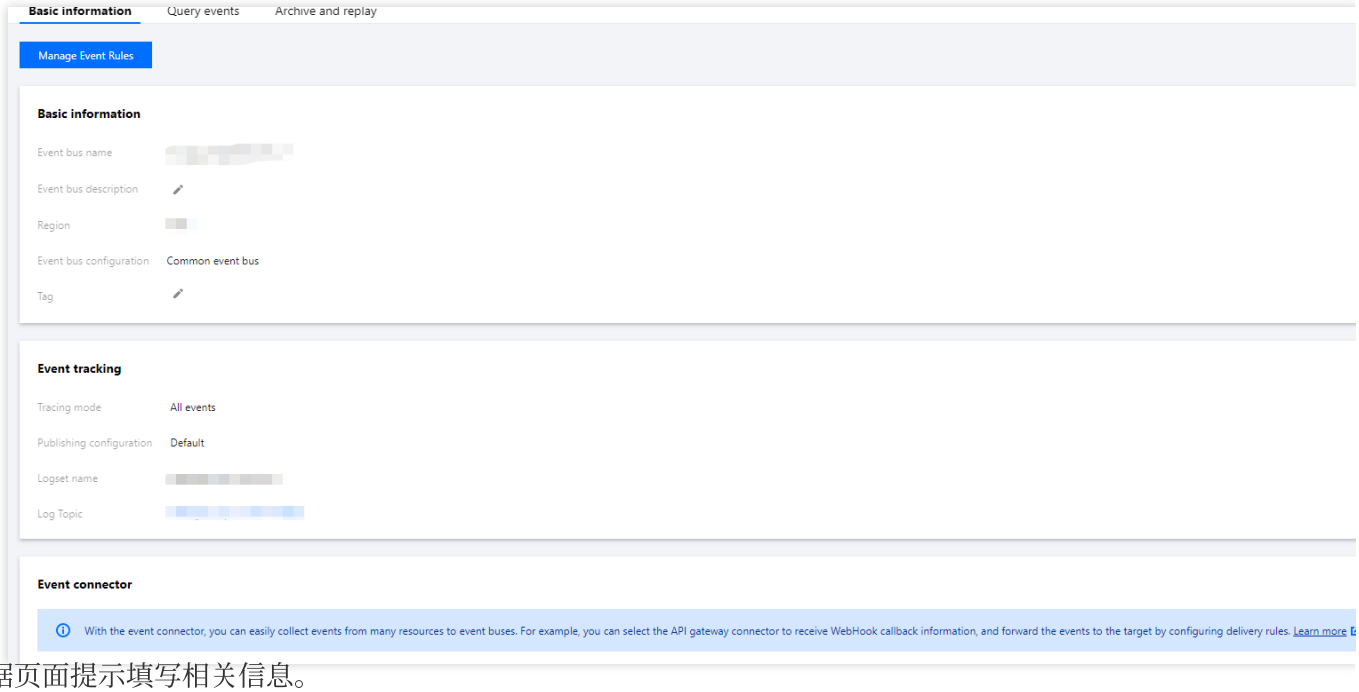
目前 TDMQ 连接器只支持消费 TDMQ-Pulsar 版本队列信息。

前提条件

已 创建事件集。

操作步骤

1. 登录 事件总线控制台，选择左侧导航栏中的**事件集**。
2. 在“事件集”列表，选择期望配置 TDMQ 连接器的事件集。
3. 在“事件集详情”页事件连接器配置项中单击**添加**，如下图所示：



4. 根据页面提示填写相关信息。

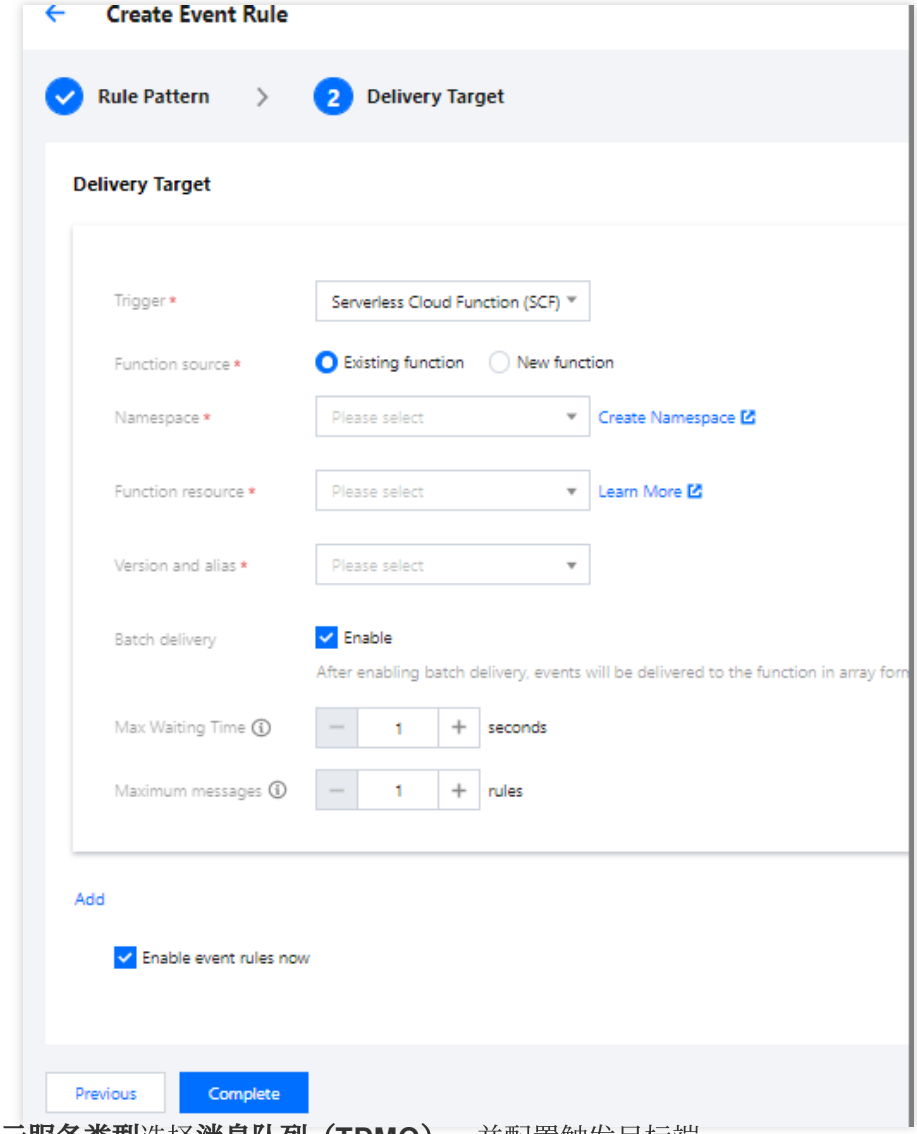
   其中**连接器类型**选择**消息队列（TDMQ）**连接器，其余配置项按照提示填写。

5. 单击**确定**完成创建。

6. 选择左侧导航栏中的**事件规则**。

7. 在“事件规则”顶部选框，选择与之前创建一致的事件集信息，并单击**新建事件规则**，如下图所示：

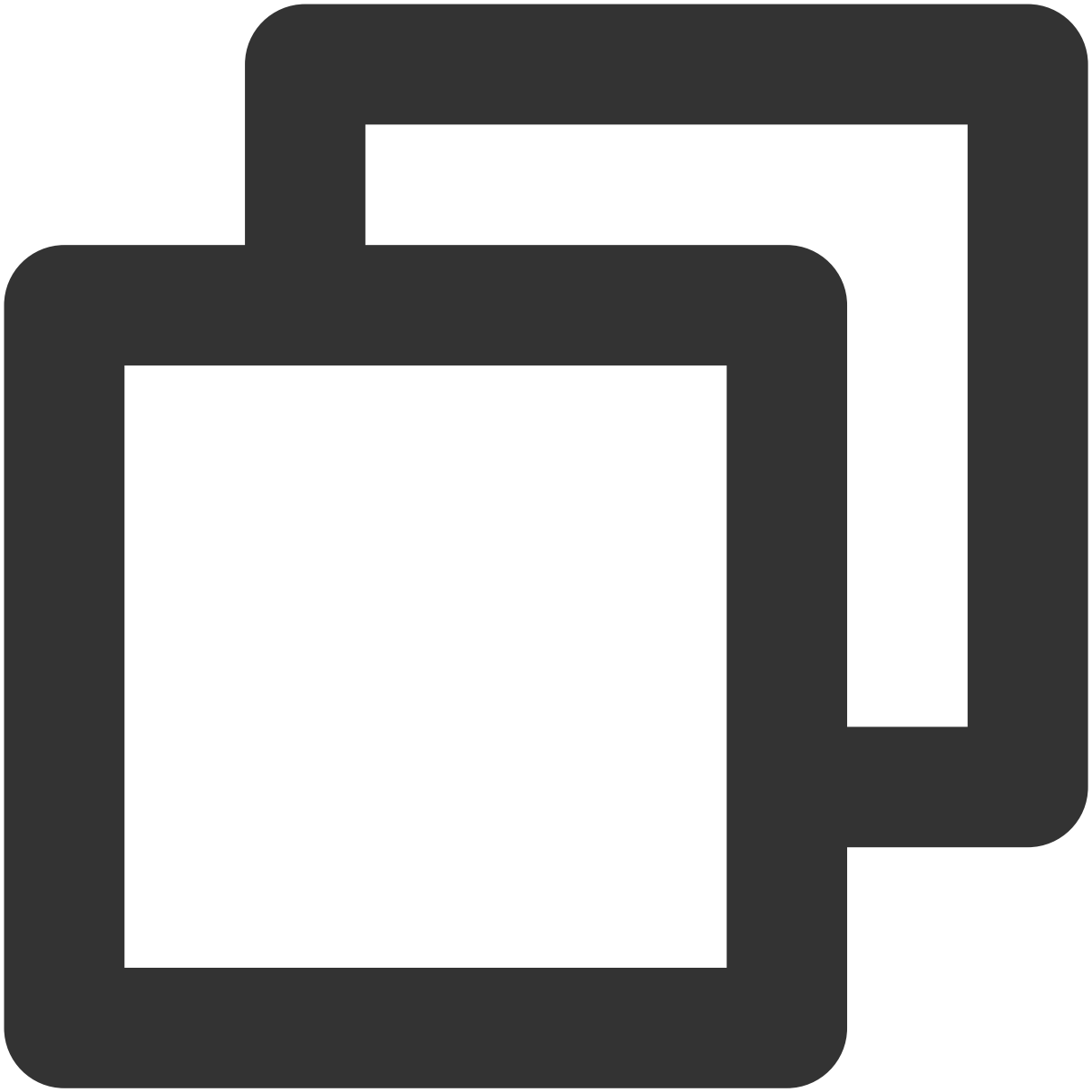
8. 根据页面提示填写相关信息，如下图所示：



其中**云服务类型**选择**消息队列（TDMQ）**，并配置触发目标端。

9. 单击**确定**即可创建 TDMQ 连接器。

TDMQ 连接器的数据结构说明



```
{
    "specversion": "1.0",
    "id": "13a3f42d-7258-4ada-da6d-023a333b4662",
    "type": "connector:tdmq",
    "source": "tdmq.cloud.tencent",
    "subjuect": "qcs::tdmq:$region:$account:topicName/$topicSets.clusterId/$topicSe
    "time": "1615430559146",
    "region": "ap-guangzhou",
    "datacontenttype": "application/json;charset=utf-8",
    "data": {
                    "topic":  "persistent://appid/namespace/topic-1",
```

```
            "tags": "testtopic",
                    "TopicType": "0",
                    "subscriptionName": "xxxxxx",
                    "toTimestamp": "1603352765001",
            "partitions": "0",
                    "msgId": "123345346",
                    "msgBody": "Hello from TDMQ!"
    }
}
```

参数说明如下：

参数	描述
topic	Topic 完整路径 `persistent://appid/namespace/topic-1` 。
tags	TDMQ 标签。
topictype	topic 类型描述： 0：普通消息。 1：全局顺序消息。 2：局部顺序消息。 3：重试队列。 4：死信队列。
subscriptionName	订阅名称。
timestamp	时间戳，精确到毫秒。
partitions	TDMQ 队列消费的 partition。
msgId	TDMQ 消息 ID。
msgBody	TDMQ 消息体。