

函数工作流 FunctionGraph

# 用户指南

文档版本 01  
发布日期 2025-09-08



版权所有 © 华为云计算技术有限公司 2025。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

## 商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## 注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

## 华为云计算技术有限公司

地址：贵州省贵安新区黔中大道交兴功路华为云数据中心 邮编：550029

网址：<https://www.huaweicloud.com/>

# 目录

- 1 FunctionGraph 业务使用流程.....1
- 2 创建用户并授权使用 FunctionGraph..... 5
- 3 FunctionGraph 支持的运行时语言..... 9
- 4 创建函数.....10
  - 4.1 创建空白函数.....10
    - 4.1.1 创建事件函数..... 10
    - 4.1.2 创建 HTTP 函数..... 14
  - 4.2 使用函数模板创建函数..... 25
  - 4.3 使用容器镜像创建函数..... 27
  - 4.4 使用 Terraform 创建函数..... 32
  - 4.5 创建 GPU 函数..... 34
    - 4.5.1 GPU 函数概述..... 34
    - 4.5.2 自定义镜像方式创建 GPU 函数..... 37
    - 4.5.3 定制运行时方式创建 GPU 函数..... 39
- 5 配置函数.....41
  - 5.1 函数配置概述.....41
  - 5.2 配置函数代码.....42
    - 5.2.1 在线编辑函数代码..... 42
    - 5.2.2 直接上传函数代码..... 45
    - 5.2.3 通过 OBS 上传函数代码..... 48
  - 5.3 配置函数的依赖包..... 50
    - 5.3.1 函数依赖包概述..... 50
    - 5.3.2 制作函数的私有依赖包..... 52
    - 5.3.3 为函数配置依赖包..... 55
  - 5.4 配置函数的委托权限..... 58
  - 5.5 配置函数的网络环境..... 65
  - 5.6 配置函数的触发器..... 69
    - 5.6.1 函数触发器概述..... 70
    - 5.6.2 使用定时触发器（TIMER）..... 89
    - 5.6.3 使用 API 网关服务（APIG 专享版）触发器..... 95
    - 5.6.4 使用 API Connect（APIC）触发器..... 100
    - 5.6.5 使用云审计服务（CTS）触发器..... 102

|                                          |            |
|------------------------------------------|------------|
| 5.6.6 使用文档数据库服务（DDS）触发器（即将下线不建议使用）       | 107        |
| 5.6.7 使用数据接入服务（DIS）触发器                   | 109        |
| 5.6.8 使用分布式消息服务 Kafka 版（KAFKA）触发器        | 113        |
| 5.6.9 使用开源 Kafka（OPENSOURCEKAFKA）触发器     | 116        |
| 5.6.10 使用分布式消息服务 RabbitMQ 版（RABBITMQ）触发器 | 118        |
| 5.6.11 使用云数据库 GeminiDB MongoDB 触发器       | 120        |
| 5.6.12 使用云数据库 GeminiDB DynamoDB 触发器      | 122        |
| 5.6.13 使用设备接入（IoTDA）触发器                  | 127        |
| 5.6.14 使用云日志服务（LTS）触发器                   | 129        |
| 5.6.15 使用消息通知服务（SMN）触发器                  | 131        |
| 5.6.16 使用对象存储服务（OBS）触发器                  | 132        |
| 5.6.17 使用 EventGrid 触发器（OBS 应用事件源）       | 135        |
| 5.6.18 使用 EventGrid 触发器（RocketMQ 自定义事件源） | 137        |
| 5.6.19 使用 EventGrid 触发器（RabbitMQ 自定义事件源） | 140        |
| 5.6.20 管理函数的触发器                          | 142        |
| 5.7 在线调试函数                               | 143        |
| <b>6 调用函数</b>                            | <b>148</b> |
| <b>7 管理函数</b>                            | <b>153</b> |
| 7.1 配置函数初始化                              | 153        |
| 7.2 配置函数的常规信息                            | 155        |
| 7.3 配置函数的磁盘挂载                            | 156        |
| 7.4 配置函数的环境变量                            | 165        |
| 7.5 配置函数的异步调用策略                          | 170        |
| 7.6 配置函数的并发处理                            | 177        |
| 7.7 配置函数的版本                              | 179        |
| 7.8 配置函数的别名                              | 181        |
| 7.9 配置函数的标签                              | 184        |
| 7.10 配置函数的动态内存                           | 185        |
| 7.11 配置函数的心跳检测                           | 186        |
| 7.12 配置函数的快照式冷启动                         | 187        |
| 7.13 配置函数的 WebSocket 连接                  | 190        |
| 7.14 配置函数的流式返回                           | 200        |
| 7.15 配置 Java 函数类隔离和预停止                   | 202        |
| 7.16 配置 HTTP 函数的请求头传入密钥                  | 203        |
| 7.17 导入导出函数                              | 204        |
| 7.18 启用禁用函数                              | 206        |
| 7.19 为函数配置预留实例                           | 207        |
| 7.20 基于 RAM 共享函数资源                       | 212        |
| <b>8 配置函数流</b>                           | <b>214</b> |
| 8.1 创建函数流                                | 214        |
| 8.2 启动函数流                                | 217        |

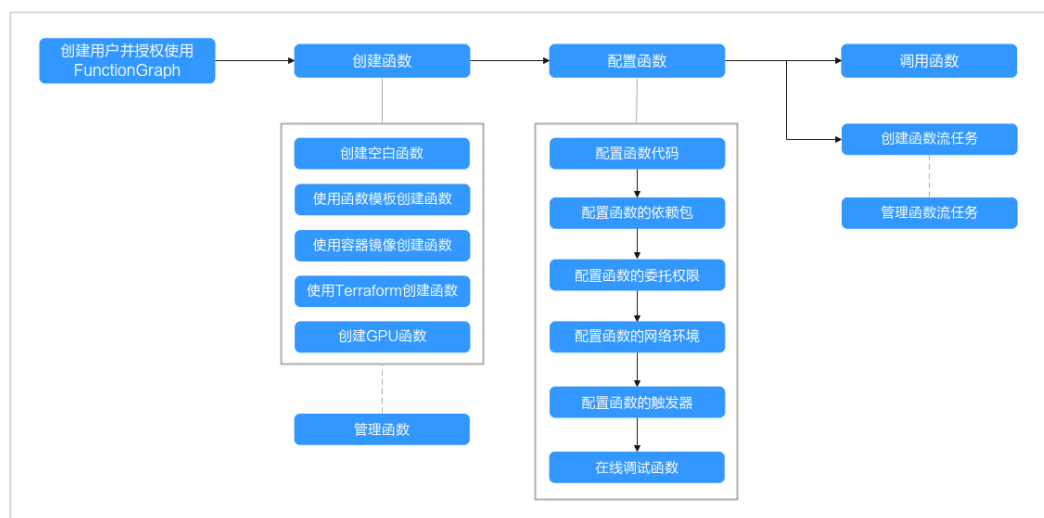
|                                      |            |
|--------------------------------------|------------|
| 8.3 函数流组件配置说明.....                   | 218        |
| 8.3.1 配置函数流的 EG 服务组件.....            | 218        |
| 8.3.2 配置函数流的函数服务组件.....              | 219        |
| 8.3.3 配置函数流的子流程控制器组件.....            | 221        |
| 8.3.4 配置函数流的并行分支控制器组件.....           | 222        |
| 8.3.5 配置函数流的开始控制器组件.....             | 223        |
| 8.3.6 配置函数流的异常处理控制器组件.....           | 227        |
| 8.3.7 配置函数流的循环控制器组件.....             | 228        |
| 8.3.8 配置函数流的时间等待控制器组件.....           | 229        |
| 8.3.9 配置函数流的服务控制器组件.....             | 230        |
| 8.3.10 配置函数流的条件分支控制器组件.....          | 231        |
| 8.4 管理函数流.....                       | 232        |
| <b>9 通过应用中心部署函数应用.....</b>           | <b>237</b> |
| <b>10 通过 KooCLI 管理函数.....</b>        | <b>242</b> |
| <b>11 申请扩大 FunctionGraph 配额.....</b> | <b>248</b> |
| <b>12 查看监控指标与配置告警.....</b>           | <b>250</b> |
| 12.1 FunctionGraph 监控概述.....         | 250        |
| 12.2 FunctionGraph 支持的监控指标.....      | 251        |
| 12.3 查看 FunctionGraph 的监控数据.....     | 255        |
| 12.4 配置 FunctionGraph 的监控告警.....     | 257        |
| 12.5 配置和查看函数的调用日志.....               | 258        |
| 12.6 配置和查看函数的调用链信息.....              | 264        |
| <b>13 查看审计日志.....</b>                | <b>269</b> |
| 13.1 CTS 支持的 FunctionGraph 操作列表..... | 269        |
| 13.2 在 CTS 事件列表查看云审计事件.....          | 269        |

# 1 FunctionGraph 业务使用流程

函数工作流FunctionGraph是一项基于事件驱动的功能托管计算服务。使用FunctionGraph函数，只需编写业务函数代码并设置运行的条件，无需配置和管理服务器等基础设施，函数以弹性、免运维、高可靠的方式运行。此外，按函数实际执行资源计费，不执行不产生费用。

使用FunctionGraph快速创建函数的流程如下：

图 1-1 FunctionGraph 业务流程全景图

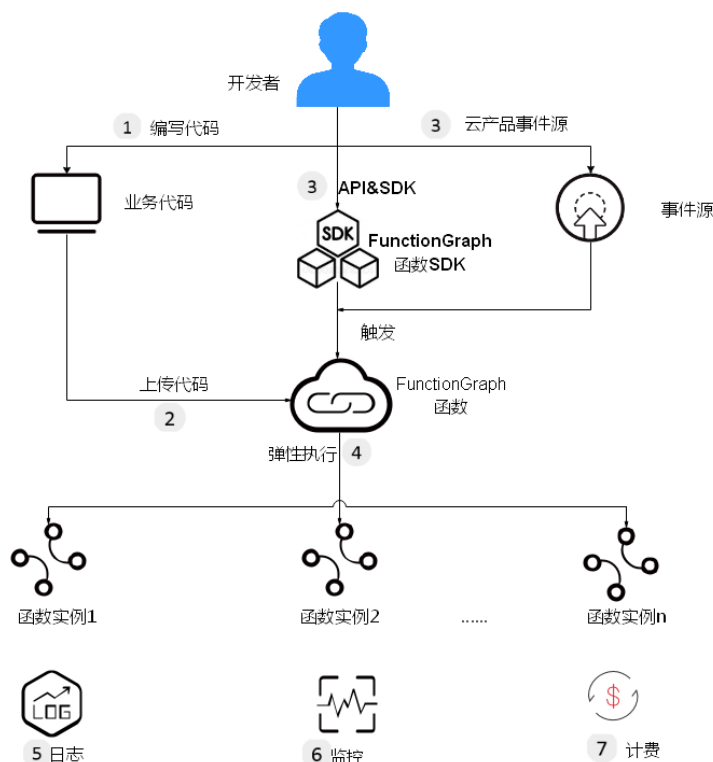


1. **创建用户并授权使用FunctionGraph**：确保登录的用户已有“FunctionGraph Administrator”权限。
2. **创建函数**：支持“创建空白函数”、“使用函数模板创建函数”、“使用容器镜像创建函数”、“使用Terraform创建函数”及“创建GPU函数”。
3. **配置函数**：支持“配置函数代码”、“配置函数的依赖”、“配置函数访问公网或VPC网络”、“配置函数的委托权限”、“配置函数的触发器”及“在线调试函数”。
4. **调用函数**：可通过RESTful API或云服务事件源触发函数执行。
5. **函数流**：支持创建和管理函数流任务。

## 函数使用流程

图1-2介绍使用FunctionGraph函数实现业务需求的流程。

图 1-2 FunctionGraph 使用流程



涉及的配置项说明如下：

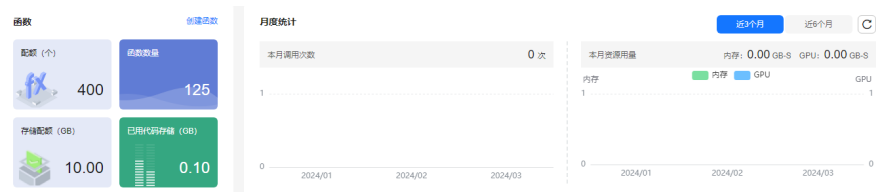
- 1. 编写代码**  
用户编写业务代码，目前支持Node.js、Python、Java、Go、C#、PHP、Cangjie和定制运行时语言，详情请参见[开发指南](#)。
- 2. 上传代码**  
上传业务代码，目前支持在线编辑代码和上传开发好的代码文件部署，详情请参见[配置函数代码](#)和[配置函数的依赖包](#)。
- 3. API和云产品事件源触发函数执行**  
通过API和云产品事件源触发函数执行，触发方法请参见[配置函数的触发器](#)。
- 4. 弹性执行**  
函数在执行过程中，会根据请求量弹性扩容，支持请求峰值的执行，此过程无需配置，由FunctionGraph完成。并发数限制请参见[使用限制](#)。
- 5. 查看日志**  
FunctionGraph函数实现了与云日志服务的对接，可以查看函数运行日志信息，请参见[配置和查看函数的调用日志](#)。
- 6. 查看监控**  
FunctionGraph函数实现了与云监控服务的对接，无需手动配置，即可查看图形化监控信息，请参见[FunctionGraph支持的监控指标](#)。

7. 计费方式
- 函数执行结束后，根据函数请求执行次数和执行时间计费，查看费用详情请参见[费用账单](#)。

总览页面介绍

- 登录函数工作流控制台，在左侧导航栏选择“总览”，进入“总览”页面。
- 可以查看函数数量/配额信息、代码存储/存储配额、函数月度调用次数/月度资源用量。

图 1-3 月度统计



- 可以查看租户层面的监控信息：调用次数、调用数TOP10、错误次数、错误数TOP10、运行时间、被拒绝次数。  
运行监控指标说明如[表1-1](#)所示。

表 1-1 监控指标说明表

| 指标       | 单位 | 说明                                                                                                                                  |
|----------|----|-------------------------------------------------------------------------------------------------------------------------------------|
| 调用次数     | 次  | 函数总的调用请求数，包含了错误和被拒绝的调用。异步调用在该请求实际被系统执行时才开始计数。                                                                                       |
| 调用数TOP10 | -  | 展示指定时间范围内（最近1天/最近3天/自定义）的函数调用数TOP10。                                                                                                |
| 运行时间     | 毫秒 | 最大运行时间为某统计粒度（周期）下，即某一时间段内所有函数单次执行最大的运行时间。<br>最小运行时间为某统计粒度（周期）下，即某一时间段内所有函数单次执行最小的运行时间。<br>平均运行时间为某统计粒度（周期）下，即某一时间段内所有函数单次执行平均的运行时间。 |
| 错误次数     | 次  | 指发生异常请求的函数不能正确执行完并且返回200，都计入错误次数。函数自身的语法错误或自身执行错误也会计入该指标。                                                                           |
| 错误数TOP10 | -  | 展示指定时间范围内（最近1天/最近3天/自定义）的函数错误数TOP10。                                                                                                |
| 被拒绝次数    | 次  | 由于并发请求太多，系统流控而被拒绝的请求次数。                                                                                                             |

- 可以查看函数流指标：调用次数、运行时间、错误次数、运行中

| 指标   | 单位 | 说明                                              |
|------|----|-------------------------------------------------|
| 调用次数 | 次  | 函数流总的调用请求数，包含了正确、错误和运行中的调用。异步函数流在请求被系统执行时才开始计数。 |
| 运行时间 | 毫秒 | 时间段内单次函数流执行平均的运行时间。                             |
| 错误次数 | 次  | 指发生异常请求的函数流不能正确执行完，会计入错误次数。                     |
| 运行中  | 个  | 正在运行中的函数流的数量。                                   |

# 2 创建用户并授权使用 FunctionGraph

如果您需要对您所拥有的FunctionGraph进行精细的权限管理，您可以使用[统一身份认证服务](#)（Identity and Access Management，简称IAM），通过IAM，您可以：

- 根据企业的业务组织，在您的华为云账号中，给企业中不同职能部门的员工创建IAM用户，让员工拥有唯一安全凭证，并使用FunctionGraph资源。
- 根据企业用户的职能，设置不同的访问权限，以达到用户之间的权限隔离。
- 将FunctionGraph资源委托给更专业、高效的其他华为云账号或者云服务，这些账号或者云服务可以根据权限进行代运维。

如果华为云账号已经能满足您的要求，不需要创建独立的IAM用户，您可以跳过本章节，不影响您使用FunctionGraph服务的其它功能。

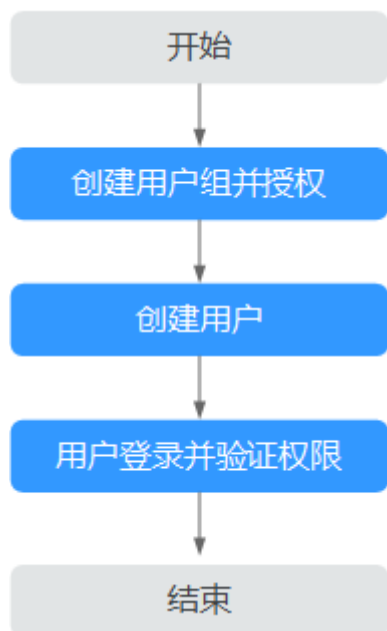
本章节为您介绍对用户授权的方法，操作流程如[图2-1](#)所示。

## 前提条件

给用户组授权之前，请您了解用户组可以添加的FunctionGraph权限，并结合实际需求进行选择，FunctionGraph支持的系统权限，请参见[FunctionGraph系统策略](#)。若您需要对除FunctionGraph之外的其它服务授权，IAM支持服务的所有权限请参见[系统权限](#)。

## 示例流程

图 2-1 给用户授权使用 FunctionGraph 权限的流程



### 1. 创建用户组并授权

使用IAM账号登录IAM控制台创建用户组，并授予FunctionGraph查询及调用权限“FunctionGraph Invoker”。

### 2. 创建用户并加入用户组

使用IAM账号在IAM控制台创建用户，并将其加入1中创建的用户组。

### 3. 用户登录验证权限

新创建的用户登录控制台，验证FunctionGraph的函数查询权限。

- 在“服务列表”中选择“函数工作流 FunctionGraph”，进入“函数 > 函数列表”，单击“创建函数”进入到创建函数界面，发现无法创建函数，表示“FunctionGraph Invoker”已生效。
- 在“服务列表”中选择除FunctionGraph外的任一服务，若提示权限不足，表示“FunctionGraph Invoker”已生效。

## FunctionGraph 自定义策略

如果系统预置的FunctionGraph权限，不满足您的授权要求，可以创建自定义策略。具体创建步骤请参见：[创建自定义策略](#)。

目前华为云支持以下两种方式创建自定义策略：

- 可视化视图创建自定义策略：无需了解策略语法，按可视化视图导航栏选择云服务、操作、资源、条件等策略内容，可自动生成策略。
- JSON视图创建自定义策略：可以在选择策略模板后，根据具体需求编辑策略内容；也可以直接在编辑框内编写JSON格式的策略内容。

以下为您介绍常用的FunctionGraph自定义策略样例。

- 示例1：授权用户查询函数代码和配置

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "functiongraph:function:list",
        "functiongraph:function:getConfig",
        "functiongraph:function:getCode"
      ]
    }
  ]
}
```

- 示例2：拒绝用户删除函数

拒绝策略需要同时配合其他策略使用，否则没有实际作用。用户被授予的策略中，一个授权项的作用如果同时存在Allow和Deny，则遵循Deny优先。

如果您给用户授予FunctionGraph FullAccess的系统策略，但不希望用户拥有FunctionGraph FullAccess中定义的删除函数权限，您可以创建一条拒绝删除函数的自定义策略，然后同时将FunctionGraph FullAccess和拒绝策略授予用户，根据Deny优先原则，则用户可以对FunctionGraph执行除了删除函数外的所有操作。拒绝策略示例如下：

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "functiongraph:function:delete"
      ]
    }
  ]
}
```

- 示例3：特定资源权限配置

特定资源：授予IAM用户特定资源的相应权限。例如授予IAM用户所属应用Default下函数functionname的相应权限，需将函数functionname设置为指定资源路径，添加资源路径：FUNCTIONGRAPH:\*:\*function:Default/functionname。

**指定函数资源：【格式】FUNCTIONGRAPH:\*:\*function:所属应用/函数名称**

对于函数资源，IAM自动生成资源路径前缀

“FUNCTIONGRAPH:\*:\*function:”。通过所属应用和函数名称指定具体的资源路径，支持通配符\*。例如：FUNCTIONGRAPH:\*:\*function:Default/\*表示Default应用下的任意函数。

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "functiongraph:function:list"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "functiongraph:function:listAlias",
        "functiongraph:function:listVersion",
        "functiongraph:function:getConfig",
        "functiongraph:function:getCode",
        "functiongraph:function:updateCode",
        "functiongraph:function:invoke",
        "functiongraph:function:updateConfig",

```

```
    "functiongraph:function:createVersion",
    "functiongraph:function:updateAlias",
    "functiongraph:function:createAlias"
  ],
  "Resource": [
    "FUNCTIONGRAPH:*:*:function:Default/*"
  ]
}
]
```

FunctionGraph 资源

资源是服务中存在的对象。在FunctionGraph中，资源包括：函数和触发器，您可以在创建自定义策略时，通过指定资源路径来选择特定资源。

表 2-1 FunctionGraph 的指定资源与对应路径

| 资源类型     | 资源名称 | 资源路径                                                                                                                                                                                                                |
|----------|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| function | 函数   | <p>【格式】</p> <p>FunctionGraph:::function:分组/函数名称</p> <p>【说明】</p> <p>对于桶资源，IAM自动生成资源路径前缀 <b>FunctionGraph:*:*:function:</b> 通过函数名称指定具体的资源路径，支持通配符。例如：<b>FunctionGraph:*:*:function:default/*</b>表示default分组下任意函数。</p> |
| trigger  | 触发器  | <p>【格式】</p> <p>FunctionGraph:::trigger:触发器ID</p> <p>【说明】</p> <p>对于对象资源，IAM自动生成资源路径前缀 <b>FunctionGraph:*:*:trigger:</b> 通过触发器ID指定具体的资源路径，支持通配符。例如：<b>FunctionGraph:*:*:trigger:*</b>表示任意触发器。</p>                     |

# 3 FunctionGraph 支持的运行时语言

在函数工作流服务中创建函数时，需选定所需的运行时（Runtime），运行时为相应语言提供执行环境，以传递函数的调用事件、上下文信息和响应。用户可以使用函数工作流服务提供的运行时，或自行构建定制运行时。

FunctionGraph支持的运行时语言及其版本请参考[表3-1](#)。

表 3-1 FunctionGraph 支持的运行时语言

| 运行时语言   | 支持版本                                                                               | 开发指导                                                   |
|---------|------------------------------------------------------------------------------------|--------------------------------------------------------|
| Node.js | 6.10、8.10、10.16、12.13、14.18、16.17、18.15、20.15                                      | 接口定义、SDK接口说明和函数开发指导请参见 <a href="#">Node.js函数开发指南</a> 。 |
| Python  | 2.7、3.6、3.9、3.10、3.12                                                              | 接口定义、SDK接口说明和函数开发指导请参见 <a href="#">Python函数开发指南</a> 。  |
| Java    | 8、11、17、21（仅支持“中东-利雅得”、“土耳其-伊斯坦布尔”区域）                                              | 接口定义、SDK接口说明和函数开发指导请参见 <a href="#">Java函数开发指南</a> 。    |
| Go      | 1.x                                                                                | 接口定义、SDK接口说明和函数开发指导请参见 <a href="#">Go函数开发指南</a> 。      |
| C#      | .NET Core 2.1、.NET Core 3.1、.NET Core 6.0、.NET Core 8.0（仅支持“中东-利雅得”、“土耳其-伊斯坦布尔”区域） | 接口定义、SDK接口说明和函数开发指导请参见 <a href="#">C#函数开发指南</a> 。      |
| PHP     | 7.3、8.3                                                                            | 接口定义、SDK接口说明和开发指导请参见 <a href="#">PHP函数开发指南</a> 。       |
| 定制运行时   | -                                                                                  | -                                                      |

# 4 创建函数

## 4.1 创建空白函数

### 4.1.1 创建事件函数

创建函数是在FunctionGraph业务使用流程中的第一步操作，如需通过云产品事件触发函数执行特定任务，可创建事件函数，并根据业务需求完成函数代码、网络配置及函数触发器等必要配置。本章节介绍如何通过控制台创建事件函数。

FunctionGraph支持创建空白事件函数和HTTP函数，关于事件函数与HTTP函数的选型对比请参考[表4-1](#)。

表 4-1 事件函数与 HTTP 函数选型对比

| 对比项  | 事件函数                                                                                                                                                                     | HTTP函数                                                                                                                                                                                         |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 功能   | 用于处理文件和数据流，可以通过 <a href="#">各类云产品的事件触发</a> ，以及用于处理异步请求，能够追踪并保存每个异步调用的状态。                                                                                                 | 支持流行的Web应用框架和AI项目，可以通过浏览器访问，或通过URL调用。                                                                                                                                                          |
| 适用场景 | <ul style="list-style-type: none"><li>云产品集成：OBS实时文件处理、LTS日志加工等。</li><li>ETL数据加工：数据库数据清洗、消息队列处理等。</li><li>常规任务：定时任务、周期任务、脚本任务等。</li><li>多媒体处理：音视频转码、直播录制、图片加工等。</li></ul> | <ul style="list-style-type: none"><li>快速构建流行Web框架应用：Express、Flask等。</li><li>快速构建AI模型推理服务：如Stable Diffusion、ComfyUI、DeepSeek等。</li><li>迁移已有的应用和接口：HTML5网站、REST API接口、移动APP、小程序、游戏结算等。</li></ul> |

约束与限制

单个账户下最大允许创建的函数个数默认为400个，如需扩大配额请[提交工单](#)。

前提条件

- 使用函数工作流控制台的所有操作均默认具有操作权限，请确保您登录的用户已有“FunctionGraph Administrator”权限，即FunctionGraph服务所有权限，更多权限的说明请参考[权限管理](#)。
- 如果函数工作流服务需要访问云日志服务LTS、虚拟私有云VPC等其他云服务进行协同工作，则需[创建函数委托](#)并授权函数工作流服务访问所需云服务的权限；若函数工作流服务无需访问其他云服务，则无需创建和选择函数委托。
- 如果需要配置函数工作流服务访问VPC内资源，请参考[创建虚拟私有云和子网](#)创建VPC和子网。

创建事件函数

- 步骤1** 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
- 步骤2** 单击右上方的“创建函数”，进入“创建函数”页面。
- 步骤3** 如[图4-1](#)所示，创建方式选择“创建空白函数”，参考[表4-2](#)配置函数基本信息，参考[表4-3](#)配置函数高级设置参数。

图 4-1 函数基础信息

选择创建方式

创建空白函数  
从头开始创建函数

容器镜像  
选择一个容器镜像来部署你的函数

使用函数模板  
使用示例代码创建函数

基本信息

函数类型

事件函数 HTTP函数

处理事件请求的函数。您可以通过云服务平台触发函数并执行。

区域

华东-上海一

不同区域的资源之间内网不互通，请就近选择靠近您业务的区域，可以降低网络时延，提高访问速度。

函数名称

FG-demo

可包含字母、数字、下划线和中划线，以大/小写字母开头，以字母或数字结尾。长度不超过60个字符。

企业项目

default

函数归属的企业项目。您可以使用不同的企业项目对函数进行项目管理。

委托

fgs\_default\_agency

通过统一身份认证服务（IAM）的委托授予函数访问其他云服务的权限。如果您的函数不访问任何云服务，可不选择委托。

委托权限策略

DIS User SWR Admin fgs\_default\_region\_role fgs\_default\_global\_role

展示您选择委托 fgs\_default\_agency 中的权限策略。也可以在 IAM 控制台增加或删除策略。

运行时

Node.js 16.17

您用来编写函数的语言。请注意，控制台代码模板仅支持Node.js、Python和PHP。

表 4-2 创建函数基础信息参数说明

| 参数   | 说明                                                          | 取值样例   |
|------|-------------------------------------------------------------|--------|
| 函数类型 | 选择事件函数。<br>事件函数为可通过特定事件触发的函数，通常为JSON格式的请求事件。                | 事件函数   |
| 区域   | 选择函数所在的区域。<br>不同区域的资源之间内网不互通，请就近选择靠近您业务的区域，可以降低网络时延、提高访问速度。 | 华东-上海一 |

| 参数     | 说明                                                                                                                                                                                                     | 取值样例                                                                        |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------|
| 函数名称   | 输入自定义的函数名称，命名规则如下： <ul style="list-style-type: none"><li>可包含字母、数字、下划线和中划线，长度不超过60个字符。</li><li>以大/小写字母开头，以字母或数字结尾。</li></ul>                                                                            | FG-demo                                                                     |
| 企业项目   | 选择函数所属的企业项目。企业项目是一种云资源管理方式，企业项目管理服务提供统一的云资源按项目管理，以及项目内的资源管理、成员管理。<br>默认为“default”，支持用户选择已创建的企业项目。<br><b>如果您没有开通企业管理服务，将无法看到企业项目选项。开通方法请参见<a href="#">如何开通企业项目</a>。</b>                                 | default                                                                     |
| 委托     | 选择函数的委托。通过委托来授权函数工作流来访问其他云服务，若函数不访问任何云服务，则无需选择委托。<br>默认“未使用任何委托”，支持选择已创建的委托。<br>当华为云账号下无函数默认委托时，FunctionGraph提供快速创建默认委托“fgs_default_agency”的功能，详情请参见 <a href="#">默认委托</a> 。                            | fgs_default_agency                                                          |
| 委托权限策略 | <b>此参数仅在选择使用委托时显示。</b><br>选定委托后将展示该委托关联的权限策略，如需调整权限策略，请参考 <a href="#">修改函数委托</a> 通过IAM控制台进行操作。                                                                                                         | DIS User; SWR Admin;<br>fgs_default_region_role;<br>fgs_default_global_role |
| 运行时    | 选择编写函数的运行时语言。<br>FunctionGraph支持的运行时语言请参考 <a href="#">FunctionGraph支持的运行时语言</a> 。 <ul style="list-style-type: none"><li>控制台代码编辑器仅支持在线编辑Node.js、Python、PHP和定制运行时。</li><li>函数成功创建后，不支持修改运行时语言。</li></ul> | Node.js 16.17                                                               |

图 4-2 创建函数高级设置参数说明

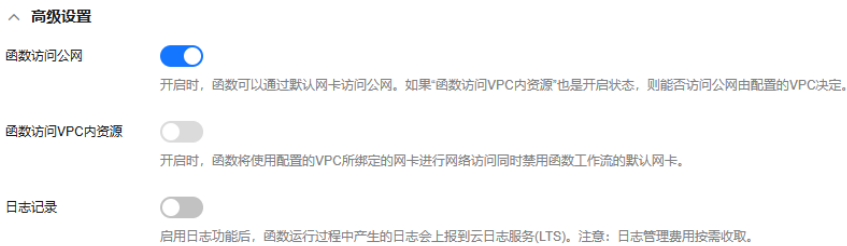


表 4-3 创建函数高级设置参数说明

| 参数         | 说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 取值样例 |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
| 函数访问公网     | 开启时，函数可以通过默认网卡访问公网上的服务，其公网访问带宽为用户间共享，仅适用于测试场景。                                                                                                                                                                                                                                                                                                                                                                                                                      | 开启   |
| 函数访问VPC内资源 | <b>开启此参数需要函数配置包含VPC管理权限的委托，若基本信息的委托中选择“未使用任何委托”则无法开启。</b><br>开启时，函数将使用配置的VPC所绑定的网卡进行网络访问，同时禁用函数工作流的默认网卡，即开关“函数访问公网”参数将不生效。<br>开启后可选择函数需访问的VPC及其子网。                                                                                                                                                                                                                                                                                                                  | 未开启  |
| 日志记录       | 启用日志功能后，函数运行过程中产生的日志会上报到云日志服务（LTS）。<br><b>注意</b><br>LTS将按需收取日志管理费用，详情请参见 <a href="#">云日志服务价格详情</a> 。<br>开启后可配置以下参数： <ul style="list-style-type: none"><li>● 日志配置方式<ul style="list-style-type: none"><li>- 自动配置：使用默认日志组和日志流。</li><li>- 自定义配置：选择自定义创建的<a href="#">日志组</a>和<a href="#">日志流</a>。</li></ul></li><li>● 日志标签<br/>函数执行时，会将日志标签配置上报到LTS，可通过标签在LTS控制台对函数日志进行过滤筛选，日志标签详情请参见<a href="#">日志管理</a>。<br/>标签键/标签值只能包含数字、字母、下划线和中划线，标签最大长度为64。最多支持添加10个标签。</li></ul> | 未开启  |

| 参数        | 说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 取值样例                                       |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------|
| KMS静态加密代码 | <p>当前仅“拉美-圣保罗”区域支持配置该参数。</p> <p>选择是否使用KMS静态加密函数代码。</p> <p><b>注意</b></p> <p>数据加密服务DEW将按需收取费用，详情请参见<a href="#">数据加密服务计费说明</a>。</p> <p>支持选择以下加密类型：</p> <ul style="list-style-type: none"><li>（默认）functiongraph/default：函数会在您账号下的DEW服务中自动创建默认密钥，您无需手动创建。</li><li>首次使用该默认密钥进行加解密，需确保函数委托中拥有以下权限：kms:dek:decrypt、kms:dek:create、kms:cmk:create、kms:cmk:get、kms:cmk:list。</li><li>使用用户主密钥：支持选择已创建的密钥加密函数代码，创建用户主密钥可参考<a href="#">创建自定义密钥</a>。使用用户主密钥，需确保函数委托中拥有以下权限：kms:dek:decrypt、kms:dek:create、kms:cmk:get、kms:cmk:list。</li></ul> <p><b>注意</b></p> <p>若选择“使用用户主密钥”，在后续使用中请勿在DEW服务中删除用于函数加密的主密钥，否则将因加密数据无法解密而导致函数执行失败。</p> <p>请参考<a href="#">JSON视图配置自定义策略</a>前往IAM控制台配置函数的委托权限策略。</p> | <p>（默认）<br/>functiongraph/<br/>default</p> |

**步骤4** 配置完成后单击“创建函数”，页面跳转至函数详情配置页面，界面上方显示成功创建函数。

**步骤5** 完成函数创建后，请参见[配置函数](#)，根据业务需求完成各项配置，即可使用函数。

----结束

相关文档

- 针对不同使用场景，函数工作流支持创建事件函数和HTTP函数，可使用内置运行时、定制运行时或自定义镜像编写函数，且支持为函数使用GPU计算资源。如何针对应用场景进行函数类型选型，详情请参见[函数类型选型](#)。
- 除使用控制台外，函数工作流支持通过API的方式创建函数，详情请参见[创建函数](#)。
- 创建函数的常见问题排查方法请参见[创建函数常见问题](#)。
- 创建简单事件函数的快速入门示例，请参见[使用空白模板创建并执行函数](#)、[使用容器镜像创建并执行事件函数](#)。

4.1.2 创建 HTTP 函数

如需使用流行Web应用框架编写函数代码和构建AI项目，可创建HTTP函数，并根据业务需求完成函数代码、网络配置及函数触发器等必要配置。本章节介绍如何通过控制台创建HTTP函数并提供HTTP使用示例。

FunctionGraph支持创建空白事件函数和HTTP函数，关于事件函数与HTTP函数的选型对比请参考[表4-4](#)。

表 4-4 事件函数与 HTTP 函数选型对比

| 对比项  | 事件函数                                                                                                                                                                     | HTTP函数                                                                                                                                                                                         |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 功能   | 用于处理文件和数据流，可以通过 <a href="#">各类云产品的事件触发</a> ，以及用于处理异步请求，能够追踪并保存每个异步调用的状态。                                                                                                 | 支持流行的Web应用框架和AI项目，可以通过浏览器访问，或通过URL调用。                                                                                                                                                          |
| 适用场景 | <ul style="list-style-type: none"><li>云产品集成：OBS实时文件处理、LTS日志加工等。</li><li>ETL数据加工：数据库数据清洗、消息队列处理等。</li><li>常规任务：定时任务、周期任务、脚本任务等。</li><li>多媒体处理：音视频转码、直播录制、图片加工等。</li></ul> | <ul style="list-style-type: none"><li>快速构建流行Web框架应用：Express、Flask等。</li><li>快速构建AI模型推理服务：如Stable Diffusion、ComfyUI、DeepSeek等。</li><li>迁移已有的应用和接口：HTML5网站、REST API接口、移动APP、小程序、游戏结算等。</li></ul> |

HTTP 函数概述

HTTP函数专注于优化 Web 服务场景，用户可以直接发送HTTP请求到URL触发函数执行，从而使用自己的Web服务。HTTP函数支持HTTP/1.1协议。

HTTP函数有以下优势：

- 丰富的框架支持  
您可以使用常见的Web框架（例如 Nodejs Web 框架：Express、Koa）编写HTTP函数，也可以将您本地的Web框架服务以极小的改造量快速迁移上云。
- 减少请求处理环节  
函数可以直接接收并处理HTTP请求，API 网关不再需要做JSON格式转换，减少请求处理环节，提升Web服务性能。
- 编写体验舒适化  
HTTP 函数的编写体验更贴近编写原生Web服务，可以使用Node.js原生接口，保证和本地开发服务体验一致。

HTTP 函数公共请求头

HTTP请求头是HTTP协议中用于传递元数据的重要组成部分，在函数调用时可以传递特定的元数据或配置信息，函数默认携带的公共请求头字段如[表4-5](#)所示。

HTTP函数的密钥信息仅能通过请求头传递，如需获取HTTP函数的AK、SK、Token，请参考[配置HTTP函数的请求头传入密钥](#)。

表 4-5 默认请求头

| 字段                 | 描述            |
|--------------------|---------------|
| X-CFF-Request-Id   | 当前的请求ID。      |
| X-CFF-Memory       | 函数分配的内存。      |
| X-CFF-Timeout      | 函数超时时间。       |
| X-CFF-Func-Version | 函数版本。         |
| X-CFF-Func-Name    | 函数名称。         |
| X-CFF-Project-Id   | 函数的ProjectID。 |
| X-CFF-Package      | 函数组。          |
| X-CFF-Region       | 函数当前所在Region。 |

约束与限制

表 4-6 创建 HTTP 函数约束限制

| 限制类别     | 具体限制                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 创建函数数量限制 | 单个账户下最大允许创建的函数个数默认为400个，如需扩大配额请 <a href="#">提交工单</a> 。                                                                                                                                                                                                                                                                                                                                                                                                             |
| HTTP函数限制 | <ul style="list-style-type: none"><li>• HTTP函数当前不区分编程语言，函数执行入口必须在bootstrap文件中设置，用户直接写启动命令，<b>端口统一开放成8000</b>，绑定IP为127.0.0.1。</li><li>• 函数需要返回一个合法的HTTP响应报文，用户的HTTP响应体不超过6M。<br/>函数的返回合法的HTTP响应报文中必须包含body(String)、statusCode(int)、headers(Map)和isBase64Encoded(boolean)，HTTP函数会默认对返回结果做Base64编码，isBase64Encoded默认为true，其它框架同理。</li><li>• HTTP函数只允许创建APIG和APIC触发器，其他触发器类型不支持。触发器使用方法请参考<a href="#">使用APIG专享版触发器</a>。</li><li>• HTTP函数不支持长时间运行和异步调用，不支持重试。</li></ul> |
| 其他限制     | <ul style="list-style-type: none"><li>• bootstrap文件是HTTP函数的启动文件，HTTP函数仅支持读取bootstrap作为启动文件名称，其它名称将无法启动服务，bootstrap启动文件请参见<a href="#">bootstrap文件示例</a>。<br/>若运行JAR包，bootstrap中建议增加JVM参数-Dfile.encoding=utf-8，否则可能会出现中文乱码。</li><li>• 函数发起HTTP请求时，如果是内网访问，则请求IP地址是动态；如果是公网访问，则请求IP地址是固定。如需了解更多关于请求IP地址的详情，请<a href="#">提交工单</a>咨询技术支持工程师。</li></ul>                                                                                                                  |

前提条件

- 使用函数工作流控制台的所有操作均默认具有操作权限，请确保您登录的用户已有“FunctionGraph Administrator”权限，即FunctionGraph服务所有权限，更多权限的说明请参考[权限管理](#)。
- 如果函数工作流服务需要访问云日志服务LTS、虚拟私有云VPC等其他云服务进行协同工作，则需[创建函数委托](#)并授权函数工作流服务访问所需云服务的权限；若函数工作流服务无需访问其他云服务，则无需创建和选择函数委托。
- 如果需要配置函数工作流服务访问VPC内资源，请参考[创建虚拟私有云和子网](#)创建VPC和子网。

创建 HTTP 函数

- 步骤1** 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
- 步骤2** 单击右上方的“创建函数”，进入“创建函数”页面。
- 步骤3** 如[图4-3](#)所示，创建方式选择“创建空白函数”，参考[表4-7](#)配置函数基本信息，参考[表4-3](#)配置函数高级设置参数。

图 4-3 创建 HTTP 函数基本信息

选择创建方式

创建空白函数  
从头开始创建函数

克隆模板  
选择一个克隆模板中已有的函数

使用函数模板  
使用示例代码创建函数

基本信息

函数类型 ①

事件函数 HTTP函数

处理HTTP请求的函数。您可以直接发送HTTP请求触发函数执行，从而使用自己的Web服务。

区域

华东-上海一

不同区域的资源之间内网不互通，请就近选择靠近您业务的区域，可以降低网络时延、提高访问速度。

函数名称

FG-demo

可包含字母、数字、下划线和中划线，以大小写字母开头，以字母或数字结尾，长度不超过60个字符。

企业项目 ①

default

函数归属的企业项目，您可以使用不同的企业项目对函数进行项目管理。

委托 ①

fgp\_default\_agency

通过统一身份认证服务（IAM）的委托授予函数访问其他云服务的权限。如果您的函数不访问任何云服务，可不选择委托。

表 4-7 函数基本信息

| 参数   | 说明                                                                                                                              | 取值样例    |
|------|---------------------------------------------------------------------------------------------------------------------------------|---------|
| 函数类型 | 选择HTTP函数。<br>处理HTTP请求的函数，可以直接发送HTTP请求到URL触发函数执行，从而使用Web服务。                                                                      | HTTP函数  |
| 区域   | 选择函数所在的区域。<br>不同区域的资源之间内网不互通，请就近选择靠近您业务的区域，可以降低网络时延、提高访问速度。                                                                     | 华东-上海一  |
| 函数名称 | 输入自定义的函数名称，命名规则如下： <ul style="list-style-type: none"><li>• 可包含字母、数字、下划线和中划线，长度不超过60个字符。</li><li>• 以大/小写字母开头，以字母或数字结尾。</li></ul> | FG-demo |

| 参数     | 说明                                                                                                                                                                          | 取值样例                                                                        |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------|
| 企业项目   | 选择函数所属的企业项目。企业项目是一种云资源管理方式，企业项目管理服务提供统一的云资源按项目管理，以及项目内的资源管理、成员管理。<br>默认为“default”，支持用户选择已创建的企业项目。<br><b>如果您没有开通企业管理服务，将无法看到企业项目选项。开通方法请参见<a href="#">如何开通企业项目</a>。</b>      | default                                                                     |
| 委托     | 选择函数的委托。通过委托来授权函数工作流来访问其他云服务，若函数不访问任何云服务，则无需选择委托。<br>默认“未使用任何委托”，支持选择已创建的委托。<br>当华为云账号下无函数默认委托时，FunctionGraph提供快速创建默认委托“fgs_default_agency”的功能，详情请参见 <a href="#">默认委托</a> 。 | fgs_default_agency                                                          |
| 委托权限策略 | <b>此参数仅在选择使用委托时显示。</b><br>选定委托后将展示该委托关联的权限策略，如需调整权限策略，请参考 <a href="#">修改函数委托</a> 通过IAM控制台进行操作。                                                                              | DIS User; SWR Admin;<br>fgs_default_region_role;<br>fgs_default_global_role |

图 4-4 创建函数高级设置参数说明

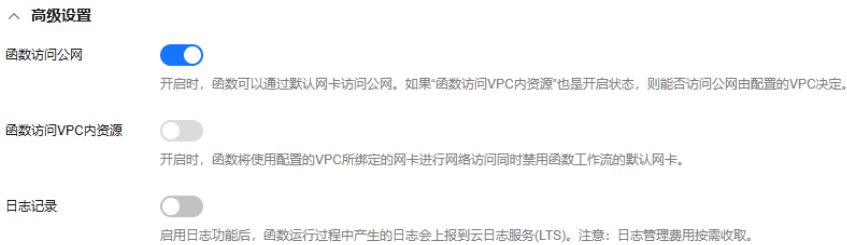


表 4-8 创建函数高级设置参数说明

| 参数     | 说明                                             | 取值样例 |
|--------|------------------------------------------------|------|
| 函数访问公网 | 开启时，函数可以通过默认网卡访问公网上的服务，其公网访问带宽为用户间共享，仅适用于测试场景。 | 开启   |

| 参数         | 说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | 取值样例 |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
| 函数访问VPC内资源 | <p>开启此参数需要函数配置包含VPC管理权限的委托，若基本信息的委托中选择“未使用任何委托”则无法开启。</p> <p>开启时，函数将使用配置的VPC所绑定的网卡进行网络访问，同时禁用函数工作流的默认网卡，即开关“函数访问公网”参数将不生效。</p> <p>开启后可选择函数需访问的VPC及其子网。</p>                                                                                                                                                                                                                                                                                                                           | 未开启  |
| 日志记录       | <p>启用日志功能后，函数运行过程中产生的日志会上报到云日志服务（LTS）。</p> <p><b>注意</b><br/>LTS将按需收取日志管理费用，详情请参见<a href="#">云日志服务价格详情</a>。</p> <p>开启后可配置以下参数：</p> <ul style="list-style-type: none"> <li>日志配置方式 <ul style="list-style-type: none"> <li>自动配置：使用默认日志组和日志流。</li> <li>自定义配置：选择自定义创建的<a href="#">日志组</a>和<a href="#">日志流</a>。</li> </ul> </li> <li>日志标签<br/>函数执行时，会将日志标签配置上报到LTS，可通过标签在LTS控制台对函数日志进行过滤筛选，日志标签详情请参见<a href="#">日志管理</a>。</li> </ul> <p>标签键/标签值只能包含数字、字母、下划线和中划线，标签最大长度为64。最多支持添加10个标签。</p> | 未开启  |

| 参数        | 说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | 取值样例                              |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| KMS静态加密代码 | <p>当前仅“拉美-圣保罗一”区域支持配置该参数。</p> <p>选择是否使用KMS静态加密函数代码。</p> <p><b>注意</b><br/>数据加密服务DEW将按需收取费用，详情请参见<a href="#">数据加密服务计费说明</a>。</p> <p>支持选择以下加密类型：</p> <ul style="list-style-type: none"><li>（默认）functiongraph/default：函数会在您账号下的DEW服务中自动创建默认密钥，您无需手动创建。<br/>首次使用该默认密钥进行加解密，需确保函数委托中拥有以下权限：kms:dek:decrypt、kms:dek:create、kms:cmk:create、kms:cmk:get、kms:cmk:list。</li><li>使用用户主密钥：支持选择已创建的密钥加密函数代码，创建用户主密钥可参考<a href="#">创建自定义密钥</a>。使用用户主密钥，需确保函数委托中拥有以下权限：kms:dek:decrypt、kms:dek:create、kms:cmk:get、kms:cmk:list。</li></ul> <p><b>注意</b><br/>若选择“使用用户主密钥”，在后续使用中请勿在DEW服务中删除用于函数加密的主密钥，否则将因加密数据无法解密而导致函数执行失败。</p> <p>请参考<a href="#">JSON视图配置自定义策略</a>前往IAM控制台配置函数的委托权限策略。</p> | （默认）<br>functiongraph/<br>default |

**步骤4** 配置完成后单击“创建函数”，页面跳转至函数详情配置页面，界面上方显示成功创建函数。

**步骤5** 完成函数创建后，请参见[配置函数](#)，根据业务需求完成各项配置，即可使用函数。

----结束

## HTTP 函数使用示例

以下以一个完整的HTTP函数创建和配置示例介绍HTTP函数的使用方法。

**在调用API前，请先确保您的业务系统所在网络与API的访问域名或地址互通：**

- 若业务系统与HTTP函数在相同VPC内时，可直接访问API。
- 若业务系统与HTTP函数在同一区域的不同VPC内时，可通过创建VPC对等连接，将两个VPC的网络打通，实现同一区域跨VPC访问API。具体步骤请参考[VPC对等连接说明](#)。
- 若业务系统与HTTP函数在不同区域的不同VPC内时，可通过创建云连接实例并加载需要互通的VPC，将两个VPC的网络打通，实现跨区域跨VPC访问API。具体步骤请参考[跨区域VPC互通](#)。
- 若业务系统与HTTP函数通过公网互通，请确保HTTP函数已绑定弹性IP。

## 步骤一：制作代码包

1. 准备一个node脚本文件，命名为“index.js”，代码示例如下：

```
const http = require('http'); // Import Node.js core module

var server = http.createServer(function (req, res) { //create web server
  res.writeHead(200, { 'Content-Type': 'text/html' });
  res.write('<html><body><h2>This is http function.</h2></body></html>');
  res.end();
});

server.listen(8000, '127.0.0.1'); //6 - listen for any incoming requests

console.log('Node.js web server at port 8000 is running..')
```

如需解析事件中的body，可参考以下代码示例：

```
const http = require('http');
const server = http.createServer((req, res) => {
  // 只处理 POST 请求 ( 也可以处理 PUT/PATCH/DELETE 等有请求体的方法 )
  if (req.url.startsWith("/apig-event-template")) {
    let body = "";
    // 1. 监听 'data' 事件收集数据块
    req.on('data', (chunk) => {
      body += chunk; // 将二进制数据块转为字符串
      // 可选：设置请求体大小限制 ( 防止内存溢出 )
      if (body.length > 1e6) { // 1MB
        req.destroy(); // 终止连接
      }
    });
    // 2. 监听 'end' 事件处理完整请求体
    req.on('end', () => {
      try {
        // 根据 Content-Type 处理不同格式
        const contentType = req.headers['content-type'] || "";
        // 处理 JSON 格式
        if (contentType.includes('application/json')) {
          const data = JSON.parse(body);
          console.log('Received JSON:', data);
          // 处理表单格式 (URL编码)
        } else if (contentType.includes('application/x-www-form-urlencoded')) {
          const params = new URLSearchParams(body);
          const data = Object.fromEntries(params);
          console.log('Received form data:', data);
          // 处理普通文本
        } else {
          console.log('Received raw text:', body);
        }
        // 响应客户端
        res.writeHead(200, { 'Content-Type': 'text/plain' });
        res.end('Body received');
      } catch (error) {
        // 处理错误 ( 如 JSON 解析失败 )
        console.error('Error processing body:', error);
        res.writeHead(400, { 'Content-Type': 'text/plain' });
        res.end('Invalid body format');
      }
    });
    // 3. 处理请求错误
    req.on('error', (error) => {
      console.error('Request error:', error);
      res.writeHead(500, { 'Content-Type': 'text/plain' });
      res.end('Internal server error');
    });
  } else {
    // 处理无请求体的方法
    res.writeHead(200, { 'Content-Type': 'text/plain' });
    res.end('No body needed');
  }
});
```

```
// 启动服务器
const PORT = 8000;
server.listen(PORT, () => {
  console.log(`Server running at http://localhost:${PORT}`);
});
```

2. 准备一个bootstrap启动文件，作为HTTP函数的启动文件。  
示例：bootstrap文件内容如下。  
`/opt/function/runtime/nodejs12.13/rtsp/nodejs/bin/node $RUNTIME_CODE_ROOT/index.js`
3. 将上述两个文件打包为zip文件。

图 4-5 文件打成 zip 包



如果执行HTTP类型是Python函数，则bootstrap文件中执行函数时，建议增加“-u”参数确保日志落盘。例如：

```
/opt/function/runtime/python3.6/rtsp/python/bin/python3 -u $RUNTIME_CODE_ROOT/index.py
```

如需使用其他语言，则参考表4-9更换语言路径，请注意根据使用的运行时语言版本更换路径信息，代码包路径无需更换。

表 4-9 多语言路径说明示例

| 语言                 | 路径                                                       |
|--------------------|----------------------------------------------------------|
| Java21             | /opt/function/runtime/java21/rtsp/jre/bin/java           |
| Python3.12         | /opt/function/runtime/python3.12/rtsp/python/bin/python3 |
| PHP8.3             | /opt/function/runtime/php8.3/rtsp/php/bin/php            |
| C# (.NET Core 8.0) | /opt/function/runtime/dotnet8.0/rtsp/dotnet/dotnet       |

步骤二：部署代码包

1. 参考[创建HTTP函数](#)创建一个HTTP函数。
2. 在HTTP函数的代码详情页中，单击“上传代码 > Zip文件”上传[步骤一：制作代码包](#)准备好的Zip包。
3. 创建触发器。

a. 选择“设置 > 触发器”页签，单击“创建触发器”。

b. 如[图4-6](#)所示配置触发器信息，此处以创建“API网关服务（APIG专享版）”触发器为例，参数详情请参见[使用API网关服务（APIG专享版）触发器](#)。

图 4-6 创建触发器

创建触发器

触发器类型 ?

API 网关服务 (APIG专享版)

可以通过HTTPS或者HTTP调用FunctionGraph函数，将各个API操作（如GET和PUT）映射到特定的FunctionGraph函数，当向该API发送HTTPS或者HTTP请求时，API 网关会调用相应的FunctionGraph函数。  
当使用APIG触发器时，函数返回体必须为如下json格式：  
{  
 "statusCode": 200,  
 "isBase64Encoded": false,  
 "headers": {  
 "Content-Type": "application/json; charset=UTF-8",  
 "body": "hello world"  
 }  
}  
(isBase64Encoded为true时，body必须为base64编码格式)

\* 实例

apig-

创建实例

\* API名称

API\_dedicated

支持汉字，英文，数字，下划线，且只能以英文和汉字开头，3-64字符。

\* 分组

APIGroup\_fg

创建分组

\* 发布环境

RELEASE

创建发布环境

\* 安全认证

None

无认证模式，所有用户均可访问，不推荐使用。

\* 请求协议

HTTPS

\* 请求方法

ANY

\* 后端超时(毫秒)

5000

后端超时设置1-60,000毫秒的范围内。

示例中“安全认证”选择“None”，生产环境强烈建议开启APP认证或IAM认证。

- c. 配置完成后，单击“确定”，完成APIG触发器创建。
4. 发布API。

a. 单击“触发器”页签下的API名称，跳转至API的总览页面。

图 4-7 API 触发器

API 网关服务 (APIG专享版) (共1个)

生成的调用URL使用默认域名，限制每天最多1000次访问，建议您绑定独立域名以便不限次数访问。[如何配置](#)

API\_dedicated 应用

删除

创建时间: 2025/04/30 16:05:53 GMT+08:00

调用URL: https://[id].apigw.cn/

实例: apig-[id]

分组: APIGroup\_fg

发布环境: RELEASE

安全认证: NONE

匹配模式: 前缀匹配

请求方法: ANY

请求协议: HTTPS

请求路径: /

后端超时: 5,000 ms

- b. 单击右上方的“编辑”，进入“基本信息”页面。

图 4-8 编辑 API



- c. 修改“路径”参数为“/user/get”并单击“完成”。

图 4-9 定义 API 请求



- d. 回到API详情界面，单击右上角的“发布最新版本”，在“发布API”页面单击“确定”完成API发布。

步骤三：触发函数执行

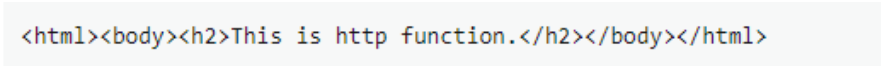
- 1. 返回函数工作流控制台，在左侧导航栏选择“函数 > 函数列表”，单击创建的 HTTP函数进入函数详情页。
- 2. 选择“设置 > 触发器”，如图4-10所示复制“调用URL”，在浏览器访问。

图 4-10 复制 URL



3. 查看请求结果。

图 4-11 查看请求结果



相关文档

- 使用Go语言构建FunctionGraph HTTP函数的实践，详情请参见[使用Go构建FunctionGraph HTTP函数](#)。
- 使用已有SpringBoot项目构建HTTP函数的实践，请参见[使用已有SpringBoot项目构建HTTP函数](#)。
- 使用Next.js项目构建HTTP函数的实践，详情请参见[使用Next.js项目构建HTTP函数](#)。
- 针对不同使用场景，函数工作流支持通过在线编写代码、上传代码文件或者使用容器镜像，创建事件函数和HTTP函数，且支持使用GPU计算资源。如何针对应用场景进行函数类型选型，详情请参见[函数类型选型](#)
- 除使用控制台外，函数工作流支持通过API的方式创建函数，详情请参见[创建函数](#)。
- 创建函数的常见问题排查方法请参见[创建函数常见问题](#)。

4.2 使用函数模板创建函数

函数模板概述

FunctionGraph提供了多种场景的函数模板，使用函数模板创建函数时，可实现函数代码、环境变量、函数触发器配置等的自动填充，助您快速构建函数应用。

支持根据函数类型、场景、云服务以及语言筛选函数模板，如[图4-12](#)所示单击函数模板的“详情”可查看该模板的描述、输入输出参数以及注意事项。

图 4-12 函数模板详情



前提条件

使用函数工作流控制台的所有操作均默认具有操作权限，请确保您登录的用户已有“FunctionGraph Administrator”权限，即FunctionGraph服务所有权限，更多权限的说明请参考[权限管理](#)。

使用函数模板创建函数

此处以Python 3.6的“context使用指导”举例，请根据实际需求选择函数模板。

1. 登录[函数工作流控制台](#)，可通过以下任一方式进入创建函数界面。
  - 在左侧的导航栏选择“函数 > 函数列表”，单击右上方的“创建函数”，进入“创建函数”页面。创建方式选择“使用函数模板”。

- 在左侧的导航栏选择“函数模板”，将鼠标放置在所需的函数模板上，单击“使用模板”，进入“创建函数”页面。
2. 选择函数模板后，如图4-13所示，会加载模板内置的代码、配置信息，具体详情参见表4-10。

图 4-13 使用函数模板创建函数

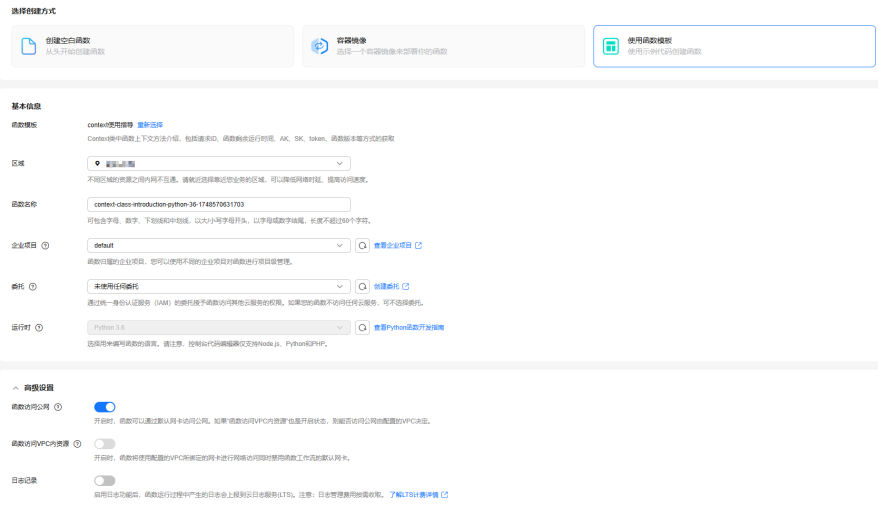


表 4-10 context 使用指导模板配置信息表

| 参数   | 说明                                                                                                                                                                         | 取值样例                                 |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------|
| 函数模板 | 默认展示已选择的函数模板。如需更换函数模板，请单击“重新选择”。                                                                                                                                           | context使用指导                          |
| 区域   | 选择函数创建的区域。<br><b>不同区域的资源之间内网不互通，请就近选择靠近您业务的区域，可以降低网络时延、提高访问速度。</b>                                                                                                         | 华东-上海一                               |
| 函数名称 | 输入自定义的函数名称，命名规则如下： <ul style="list-style-type: none"><li>可包含字母、数字、下划线和中划线，长度不超过60个字符。</li><li>以大/小写字母开头，以字母或数字结尾。</li></ul>                                                | context-class-introduction-python-36 |
| 企业项目 | 选择函数所属的企业项目。企业项目是一种云资源管理方式，由企业项目管理服务提供统一的云资源按项目管理功能，支持对项目内的资源和成员进行管理。<br>默认为“default”，支持用户选择已创建的企业项目。<br><b>如果您没有开通企业管理服务，将无法看到企业项目选项。开通方法请参见<a href="#">如何开通企业项目</a>。</b> | default                              |

| 参数   | 说明                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 取值样例                                                                                                                    |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------|
| 委托   | <p>选择函数的委托。通过委托函数工作流来访问其他云服务，例如：如果函数需要访问LTS、VPC等服务，则必须选择包含所需服务权限的委托；若函数不访问任何云服务，则无需选择委托。</p> <p>默认“未使用任何委托”，支持选择已创建的委托。</p> <p>当华为云账号下无函数默认委托时，FunctionGraph提供快速创建默认委托“fgs_default_agency”的功能，详情请参见<a href="#">默认委托</a>。</p>                                                                                                                                                                                                                            | 未使用任何委托                                                                                                                 |
| 运行时  | 选择用来编写函数的运行时语言，会根据选择的函数模板自动选择无法手动更改。                                                                                                                                                                                                                                                                                                                                                                                                                    | Python 3.6                                                                                                              |
| 高级设置 | <ul style="list-style-type: none"><li>函数访问公网：开启时，函数可以通过默认网卡访问公网上的服务，其公网访问带宽为用户间共享，仅适用于测试场景。</li><li>函数访问VPC内资源：开启时，函数将使用配置的VPC所绑定的网卡进行网络访问，同时禁用函数工作流的默认网卡，即开关“函数访问公网”参数将不生效。<br/>开启此参数需要函数配置包含VPC管理权限的委托，若基本信息的委托中选择“未使用任何委托”则无法开启。</li><li>日志记录：启用日志功能后，函数运行过程中产生的日志会上报到云日志服务（LTS）。LTS将按需收取日志管理费用，详情请参见<a href="#">云日志服务价格详情</a>。</li><li>KMS静态加密代码（仅“拉美-圣保罗一”区域支持）：选择是否使用KMS静态加密函数代码。<br/>数据加密服务DEW将按需收取费用，详情请参见<a href="#">数据加密服务计费说明</a>。</li></ul> | <ul style="list-style-type: none"><li>函数访问公网：开启</li><li>函数访问VPC内资源：未开启</li><li>日志记录：未开启</li><li>KMS静态加密代码：不使用</li></ul> |

3. 配置完成后单击“创建函数”，页面跳转至代码配置页面，各配置项操作请参见[配置函数](#)。

相关文档

- 创建函数的常见问题排查方法请参见[创建函数常见问题](#)。
- 除使用控制台外，函数工作流支持通过API的方式管理函数模板，详情请参见[函数模板API](#)。

4.3 使用容器镜像创建函数

FunctionGraph支持加载并运行容器镜像中的函数，相较于直接上传代码的方式，用户能够使用自定义代码包，灵活且减少了迁移成本。

支持使用公开和私有的镜像类型，具体详情请参考[编辑镜像属性](#)。

约束与限制

表 4-11 约束与限制

| 限制类别          | 具体限制                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 自定义容器镜像端口约束限制 | <ul style="list-style-type: none"><li>自定义容器镜像需包含HTTP Server，监听端口为8000。</li><li>自定义容器镜像开放端口限定为8000。</li></ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 容器镜像函数约束限制    | <ul style="list-style-type: none"><li>事件函数需创建一个HTTP Server并实现Method为POST和Path为/invoke的函数执行入口，可实现Method为POST和Path为/init的函数初始化入口。</li><li>函数返回要求：函数必须返回一个符合以下结构的合法HTTP响应。<pre>{  "isBase64Encoded": true false,  "statusCode": httpStatusCode,  "headers": {"headerName":"headerValue",...},  "body": "..."} </pre></li><li>Command、Args、Working dir三个参数之和不能超过5120。</li><li>初次执行时需要从SWR中拉取镜像，且冷启动时需要启动容器，所以自定义镜像冷启动比较慢。后续每次冷启动，如果节点上没有镜像，都需要从SWR中拉取。</li><li>可支持的镜像包最大为10G，当镜像包过大时可以采取一些方式缩容，比如在线题库场景中，可以把原来加载在容器中的题库数据通过外部文件系统挂载盘方式挂载到容器中。</li><li>oom错误时，内存占用大小可以在函数执行结果中查看。</li></ul> |

前提条件

- 使用函数工作流控制台的所有操作均默认具有操作权限，请确保您登录的用户已有“FunctionGraph Administrator”权限，即FunctionGraph服务所有权限，更多权限的说明请参考[权限管理](#)。
- 请参见[配置函数的委托权限](#)，创建一个包含“SWR Admin容器镜像服务（SWR）管理员”权限的委托。用户镜像储存在容器镜像服务SWR中，函数需拥有“SWR Admin”权限才能调用与获取镜像。

创建函数

- 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
- 单击右上方的“创建函数”，进入“创建函数”页面。
- 创建方式选择“容器镜像”，配置信息参见[表4-12](#)。

表 4-12 容器镜像配置信息

| 参数   | 说明                                                                                                                                                                                                              | 取值样例                                                     |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------|
| 函数类型 | 支持以下两种类型： <ul style="list-style-type: none"><li>事件函数：通常为JSON格式的请求，事件函数不受触发器类型的限制，当前FunctionGraph支持的所有类型触发器均可用于触发事件函数。</li><li>HTTP函数：用户可以直接发送HTTP请求到URL触发函数执行。</li></ul>                                        | 事件函数                                                     |
| 区域   | 选择要部署代码的区域。<br><b>不同区域的资源之间内网不互通，请就近选择靠近您业务的区域，可以降低网络时延、提高访问速度。</b>                                                                                                                                             | 华东-上海一                                                   |
| 函数名称 | 函数名称，命名规则如下： <ul style="list-style-type: none"><li>可包含字母、数字、下划线和中划线，长度不超过60个字符。</li><li>以大/小写字母开头，以字母或数字结尾。</li></ul>                                                                                           | SWR-demo                                                 |
| 企业项目 | 选择函数所属的企业项目。企业项目是一种云资源管理方式，由企业项目管理服务提供统一的云资源按项目管理功能，支持对项目内的资源和成员进行管理。<br>默认为“default”，支持用户选择已创建的企业项目。<br><b>如果您没有开通企业管理服务，将无法看到企业项目选项。开通方法请参见<a href="#">如何开通企业项目</a>。</b>                                      | default                                                  |
| 委托   | 选择函数的委托。通过委托函数工作流来访问其他云服务，需选择包含SWR Admin权限的委托，如需创建委托，请参见 <a href="#">创建委托</a> 。<br>当华为云账号下无函数默认委托时，FunctionGraph提供快速创建默认委托“fgs_default_agency”的功能，详情请参见 <a href="#">默认委托</a> 。                                  | fgs_default_agency                                       |
| 容器镜像 | 输入镜像URL，即用于函数的容器镜像的位置。您可以单击“查看镜像”，查看自有镜像及共享镜像。同时，也支持单击“选择镜像”功能，用户可在镜像列表中选择公开或私有镜像，选择成功后会自动填充镜像URL。如何制作镜像，请参见 <a href="#">制作镜像</a> 。<br>SWR中的镜像名，例如<br>swr.region_id.myhuaweicloud.com/my_group/my_image:latest。 | swr.region_id.myhuaweicloud.com/my_group/my_image:latest |
| 监听端口 | <b>当前仅支持“中东-利雅得”区域支持配置该参数。</b><br>配置代码中的HTTP Server所监听的端口，需要与镜像中配置的“EXPOSE”端口保持一致，端口取值范围是1025~65535，默认值为8000。                                                                                                   | 8000                                                     |

| 参数     | 说明                                                                                                                                                                                                                                                                                                                                                                                                                                | 取值样例 |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
| 容器镜像覆盖 | <p>可选参数。</p> <p>此处如果配置，则会覆盖dockerfile文件中的镜像配置，如果不配置则使用镜像默认配置。</p> <ul style="list-style-type: none"><li>• CMD：容器的启动命令，例如"/bin/sh"。该参数为可选参数，不填写，则默认使用镜像中的Entrypoint/CMD。字符串数组，以逗号分开。</li><li>• Args：容器的启动参数，例如"-args,value1"。该参数为可选参数，不填写，则默认使用镜像中的CMD。字符串数组，以逗号分开。</li><li>• Working Dir：容器的工作目录。该参数为可选参数，不填写，则默认使用镜像中的配置。文件夹路径，以/开头。</li><li>• 用户ID：镜像运行时的用户ID，若不填写，默认为1003。</li><li>• 用户组ID：镜像运行时的用户组ID，若不填写，默认为1003。</li></ul> | -    |

参考表4-3配置函数高级设置参数。

图 4-14 创建函数高级设置参数说明

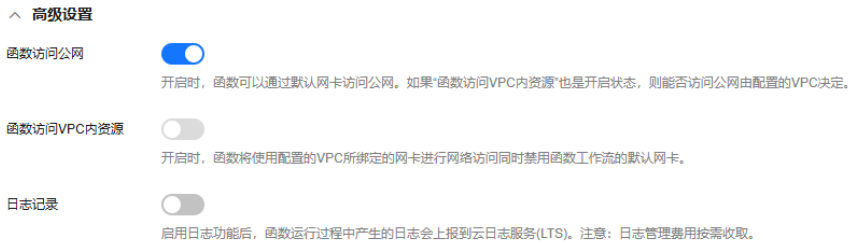


表 4-13 创建函数高级设置参数说明

| 参数         | 说明                                                                                                                                                         | 取值样例 |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
| 函数访问公网     | 开启时，函数可以通过默认网卡访问公网上的服务，其公网访问带宽为用户间共享，仅适用于测试场景。                                                                                                             | 开启   |
| 函数访问VPC内资源 | <p>开启此参数需要函数配置包含VPC管理权限的委托，若基本信息的委托中选择“未使用任何委托”则无法开启。</p> <p>开启时，函数将使用配置的VPC所绑定的网卡进行网络访问，同时禁用函数工作流的默认网卡，即开关“函数访问公网”参数将不生效。</p> <p>开启后可选择函数需访问的VPC及其子网。</p> | 未开启  |

| 参数        | 说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | 取值样例                              |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| 日志记录      | <p>启用日志功能后，函数运行过程中产生的日志会上报到云日志服务(LTS)。</p> <p><b>注意</b><br/>LTS将按需收取日志管理费用，详情请参见<a href="#">云日志服务价格详情</a>。</p> <p>开启后可配置以下参数：</p> <ul style="list-style-type: none"><li>● 日志配置方式<ul style="list-style-type: none"><li>- 自动配置：使用默认日志组和日志流。</li><li>- 自定义配置：选择自定义创建的日志组和日志流。</li></ul></li><li>● 日志标签<br/>函数执行时，会将日志标签配置上报到LTS，可通过标签在LTS控制台对函数日志进行过滤筛选，日志标签详情请参见<a href="#">日志管理</a>。<br/>标签键/标签值只能包含数字、字母、下划线和中划线，标签最大长度为64。最多支持添加10个标签。</li></ul>                                                                                                                                                                                                                              | 未开启                               |
| KMS静态加密代码 | <p>当前仅“拉美-圣保罗一”区域支持配置该参数。</p> <p>选择是否使用KMS静态加密函数代码。</p> <p><b>注意</b><br/>数据加密服务DEW将按需收取费用，详情请参见<a href="#">数据加密服务计费说明</a>。</p> <p>支持选择以下加密类型：</p> <ul style="list-style-type: none"><li>● （默认）functiongraph/default：函数会在您账号下的DEW服务中自动创建默认密钥，您无需手动创建。<br/>首次使用该默认密钥进行加解密，需确保函数委托中拥有以下权限：kms:dek:decrypt、kms:dek:create、kms:cmk:create、kms:cmk:get、kms:cmk:list。</li><li>● 使用用户主密钥：支持选择已创建的密钥加密函数代码，创建用户主密钥可参考<a href="#">创建自定义密钥</a>。使用用户主密钥，需确保函数委托中拥有以下权限：kms:dek:decrypt、kms:dek:create、kms:cmk:get、kms:cmk:list。</li></ul> <p><b>注意</b><br/>若选择“使用用户主密钥”，在后续使用中请勿在DEW服务中删除用于函数加密的主密钥，否则将因加密数据无法解密而导致函数执行失败。</p> <p>请参考<a href="#">JSON视图配置自定义策略</a>前往IAM控制台配置函数的委托权限策略。</p> | （默认）<br>functiongraph/<br>default |

4. 配置完成后单击“创建函数”，页面跳转至函数详情配置页面，各配置项操作请参见[配置函数](#)。

## 更新函数镜像

在容器镜像函数的代码配置页面中，选择“代码”页签，右侧单击“部署新镜像”，在输入框中输入新镜像的URL，单击“确定”完成部署。镜像URL获取方式如下：

1. 登录容器镜像服务控制台，左侧导航栏选择“我的镜像”。
2. 选择“自有镜像”或“他人共享”页签，在镜像列表中单击镜像名称，进入详情页面。
3. 选择“镜像版本”页签，在镜像版本列表中复制下载指令，指令中去掉“docker pull”，即为镜像URL。

## 相关文档

- 针对不同使用场景，函数工作流支持通过在线编写代码、上传代码文件或者使用容器镜像，创建事件函数和HTTP函数，且支持使用GPU计算资源。如何针对应用场景进行函数类型选型，详情请参见[函数类型选型](#)。
- 除使用控制台外，函数工作流支持通过API的方式创建函数，详情请参见[创建函数](#)。
- 使用容器镜像常见问题：[使用相同名称的镜像更新镜像时，预留实例无法自动更新，会一直使用老镜像。](#)

## 4.4 使用 Terraform 创建函数

Terraforms是一款开源工具，用于安全有效地构建、更改和版本控制基础设施。用户能够通过这些配置文件声明所需的基础设施最终状态，而不需要具体指定如何实现这个状态。

Terraform的优势：

- 提升架构一致性：减少手动配置过程中的错误和配置漂移。
- 降低运维成本：通过编程方式管理虚拟机，减少手动配置硬件及更新的需求。
- 提升操作效率：简化系统配置、维护和管理，加速软件开发和部署。
- 加快部署速度：将繁琐的配置工作转变为简单的脚本执行，缩短应用发布时间。
- 降低操作风险：支持版本控制，降低配置错误的风险。

本章节将指导开发者如何使用Terraform创建函数，方便开发者高效地创建函数资源。

## 前提条件

### 已获取访问密钥

获取访问密钥方式请参考[访问密钥](#)。建议使用安全性更高的临时访问密钥，具体请参考[临时访问密钥（企业联邦用户）](#)。

访问密钥（AK/SK，Access Key ID/Secret Access Key）包含访问密钥ID（AK）和秘密访问密钥（SK）两部分，是您在华为云的长期身份凭证，您可以通过访问密钥访问华为云API。

创建访问密钥成功后，您可以在访问密钥列表中查看访问密钥ID（AK），在下载.csv文件中查看访问密钥（SK）。

## 准备 Terraform 环境

- 安装Terraform执行环境。

Terraform提供了多种环境的安装包，具体可以参考[Terraform官网](#)。

下面以Linux CentOS (系统需要有访问公网权限)为例指导安装Terraform。

使用root用户登录系统，新建目录/home/Terraform，cd到Terraform目录执行如下命令：

```
sudo yum install -y yum-utils
sudo yum-config-manager --add-repo https://rpm.releases.hashicorp.com/RHEL/hashicorp.repo
sudo yum -y install terraform
```

- 基本的Terraform命令。

执行Terraform后会显示Terraform命令详情，查看更多命令详情请参考[Terraform CLI](#)。

- 基础Terraform语法。

Terraform配置语言主要基于HCL语法，具有配置简单、可读性强等特点，并且兼容JSON语法。详情参见[Terraform官网](#)介绍。

## 编写函数资源脚本

华为云在Terraform已经注册了provider，函数作为资源挂在华为云的provider下，可参考文档[huaweicloud\\_fgs\\_function](#)。

提供如下样例：

在服务器创建一个main.tf文件，将如下脚本拷贝到main.tf上并保存。

```
terraform {
  required_providers {
    huaweicloud = {
      source = "huaweicloud/huaweicloud"
      version = ">= 1.40.0"
    }
  }
}

provider "huaweicloud" {
  region = "cn-east-3" #实际的区域
  access_key = "*****" #替换为获取访问密钥生成的密钥
  secret_key = "*****" #替换为获取访问密钥生成的密钥
}

resource "huaweicloud_fgs_function" "fgs_function" {
  name      = "test_func_rf"
  app       = "default"
  agency    = "function-admin"
  description = "function test"
  handler   = "index.handler"
  memory_size = 128
  timeout   = 3
  runtime   = "Python3.6"
  code_type = "inline"
  func_code =
"aW1wb3J0IGpzb24KZGVmIGhhbmRsZXIgaGVhZGVmVudCkKICAgIHJldHVybiBvdXRwdXQ="
}
```

## 使用 Terraform 命令创建函数

1. 进入文件路径，执行以下命令初始化一个包含Terraform代码的工作目录。

```
terraform init
```

```
[root@function-deploy ~]# terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of local-registry.huaweicloud.huaweicloud from the dependency lock file
- Using previously-installed local-registry.huaweicloud.huaweicloud v1.45.1

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

2. 执行以下命令，将基础设施配置应用到函数。在Enter a value: 处输入yes。  
terraform apply

```
[root@function-deploy ~]# terraform apply
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# huaweicloud_fgs_function.fgs_function will be created
+ resource "huaweicloud_fgs_function" "fgs_function" {
+   agency           = "function-admin"
+   app              = (known after apply)
+   app_agency       = (known after apply)
+   code_filename     = (known after apply)
+   code_type        = "inline"
+   depend_list      = (known after apply)
+   description       = "function test"
+   enterprise_project_id = (known after apply)
+   func_code         = "D7a44bed8d0a78db5b8486f5e8ac18d59e8889"
+   functiongraph_version = (known after apply)
+   handler           = "index.handler"
+   id                = (known after apply)
+   initializer_handler = (known after apply)
+   initializer_timeout = (known after apply)
+   memory_size       = 128
+   mount_user_group_id = (known after apply)
+   mount_user_id      = (known after apply)
+   name              = "zyl_test_func_tf"
+   region            = (known after apply)
+   runtime            = "Python3.6"
+   timeout            = 3
+   upn                = (known after apply)
+   version            = (known after apply)
}

Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

huaweicloud_fgs_function.fgs_function: Creating...
huaweicloud_fgs_function.fgs_function: Creation complete after 1s [id=urn:fsc:cn-east-3:99d9f91fba989f5e12ffac98733b35532d:function:default:zyl_test_func_tf:latest]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```

3. 执行成功代表函数创建完成。

## 4.5 创建 GPU 函数

### 4.5.1 GPU 函数概述

Serverless GPU是一种高度灵活、高效利用、按需分配GPU计算资源的新兴云计算服务。GPU能力Serverless化，通过提供一种按需分配的GPU计算资源，在一定范围内有效地解决原有GPU长驻使用方式导致的低资源利用率、高使用成本和低弹性能力等痛点问题。本文将介绍Serverless GPU的详细功能和优势。

传统GPU长驻使用方式存在许多问题，例如，需要提前规划好资源需求并容易造成资源浪费。而Serverless GPU则提供了一种更加灵活的方式来利用GPU计算资源，用户只需选择合适的GPU型号和计算资源规模，就可以帮助用户有效地解决GPU长驻使用方式导致的资源浪费、高成本、低弹性等问题，为用户提供更加便捷、高效的GPU计算服务，有效承载AI模型推理、AI模型训练、音视频加速生产、图形图像加速等加速工作负载。

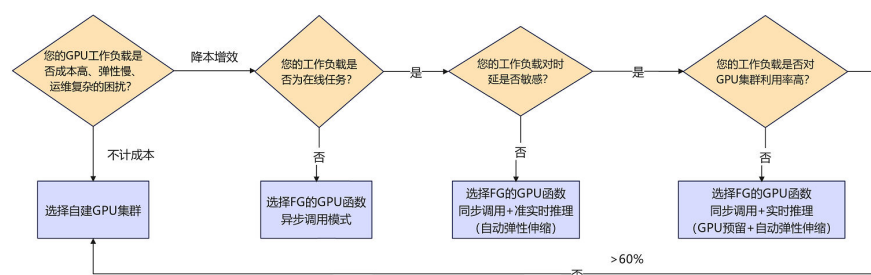
GPU函数主要用于：仿真、模拟、科学计算、音视频、AI和图像处理等场景下，使用GPU硬件加速，从而提高业务处理效率。

FunctionGraph使用GPU函数最佳实践：[使用FunctionGraph部署DeepSeek-R1蒸馏模型](#)。

表 4-14 GPU 函数规格

| 卡型        | vGPU 显存（GB）         | vGPU 算力（卡）        | 特点描述                                                                                                                                                                                                   |
|-----------|---------------------|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| NVIDIA-T4 | 1~16<br>取值说明：必须是整数。 | 说明：由系统自动分配，无需手动配置 | T4是一款独特的GPU产品，专为AI推理工作负载而设计，如处理视频，语音，搜索引擎和图像的神经网络。T4配备16GB GDDR6，GPU中集成320个Turing Tensor Core和2560个Turing CUDA Core，这款全新GPU具有突破性的性能，以及FP32/FP16/INT8/INT4等多种精度的运算能力，FP16的峰值性能为65T，INT8为130T，INT4为260T。 |

图 4-15 GPU 云产品选型决策指引



说明

GPU函数支持两种函数模式：按量模式和预留模式，详情请参见[为函数配置预留实例](#)。

约束与限制

- 目前该功能仅“华东-上海一”和“土耳其-伊斯坦布尔”区域支持，其中“土耳其-伊斯坦布尔”区域需[提交工单](#)开通白名单后支持。
- GPU函数不支持的网段：192.168.64.0/18，192.168.128.0/18，10.192.64.0/18，10.192.128.0/18。

应用场景一：准实时推理场景

特征

在准实时推理应用场景中，工作负载具有以下一个或多个特征：

- 调用稀疏  
日均调用几次到几万次，日均GPU实际使用时长远低于6~10小时，GPU存在大量闲置。
- 单次处理耗时长  
准实时推理业务的处理耗时一般在秒级~分钟级。例如，典型的CV任务处于秒级别，典型的视频处理和AIGC场景均处于分钟级别。
- 容忍冷启动  
业务可以容忍GPU冷启动耗时，或者业务流量波形对应的冷启动概率低。

### 功能优势

函数计算为准实时推理工作负载提供以下功能优势：

- 原生Serverless使用方式  
函数计算平台默认提供的按量GPU实例使用方式，会自动管理GPU计算资源。根据业务的请求数量，自动弹性GPU实例，最低0个实例，最大可配置实例数量。
- 规格最优  
函数计算平台提供的GPU实例规格，根据业务需求，选择卡型并配置使用的显存和内存的大小，为您提供最贴合业务的实例规格。
- 成本最优  
函数计算平台提供的按量付费能力，对于低GPU资源利用率的工作负载，降本幅度可达70%以上。

## 应用场景二：实时推理场景

### 特征

在实时推理应用场景中，工作负载具有低延迟特征，即单次请求的处理时效性要求高，RT（Response Time）延迟要求严格，90%的长尾延时普遍在百毫秒级别。

### 功能优势

函数计算为实时推理工作负载提供以下功能优势：

- 预留GPU实例  
函数计算平台提供了默认的按量GPU实例之外的另一种GPU使用方式——预留GPU实例。如果您希望消除冷启动延时的影响，满足实时推理业务低延迟响应的要求，可以通过配置预留GPU实例来实现。更多关于预留模式的信息，请参见[函数配置预留实例](#)。
- 服务质量优先，服务成本次优  
预留GPU实例的计费周期不同于按量GPU实例，预留GPU实例是以实例存活生命周期进行计费，而不考虑实例的活跃与闲置（不按请求计费）。因此，相较于按量GPU实例，总体使用成本较高，但相较于长期自建GPU集群，降本幅度达50%以上。
- 规格最优  
函数计算平台提供的GPU实例规格，允许您根据自己的工作负载选择不同的卡型，独立配置GPU/MEM。最小GPU规格小至1 GB显存/算力，将为您提供最贴合业务的实例规格。

- 突发流量支撑

函数计算平台提供充足的GPU资源供给，当业务遭遇突发流量时，函数计算将以秒级弹性供给海量GPU算力资源，避免因GPU算力供给不足、GPU算力弹性滞后导致的业务受损。

### 应用场景三：离线异步任务场景

#### 特征

在离线异步应用场景中，工作负载具有以下一个或多个特征：

- 执行时间长  
业务的处理耗时一般在分钟~小时级，Response Time不敏感。
- 提交后立即返回  
在触发调用后立即得到返回，从而不因长耗时处理阻塞业务主逻辑的执行。
- 实时感知任务状态  
无
- 并行处理  
离线GPU任务需要处理大量数据，对GPU资源供给要求高，通过API调用并行运行加快处理速度。
- 数据源集成  
离线GPU任务对数据源的需求多种多样，处理过程中需要与多种存储产品（例如对象存储OBS）和多种消息产品（例如消息队列）进行频繁交互。

#### 功能优势

函数计算为离线异步应用类工作负载提供以下功能优势：

- 业务架构简化  
对于长耗时，采用异步处理，提高系统响应速度、资源利用率和可用性。
- 充足的GPU资源供给  
函数计算平台提供充足的GPU资源供给，适合忙闲流量分明（长时空闲、短时繁忙）、忙闲流量不可预知的离线业务。
- 数据源集成  
函数计算支持多种数据源触发方式，例如对象存储OBS、消息队列等。

## 4.5.2 自定义镜像方式创建 GPU 函数

FunctionGraph支持通过容器镜像方式创建的函数启用GPU功能，进而帮助用户有效地解决GPU长驻使用方式导致的资源浪费、高成本、低弹性等问题，为用户提供更加便捷、高效的GPU计算服务。

### 约束与限制

- 目前该功能仅“华东-上海一”和“土耳其-伊斯坦布尔”区域支持，其中“土耳其-伊斯坦布尔”区域需[提交工单](#)开通白名单后支持。
- GPU函数不支持的网段：192.168.64.0/18，192.168.128.0/18，10.192.64.0/18，10.192.128.0/18。
- GPU型号当前仅支持NVIDIA-T4。

计费说明

函数工作流将按需收取GPU计量时间费用，详情请参见[函数工作流计费项](#)。

创建 GPU 函数

- 步骤1 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
- 步骤2 单击右上方的“创建函数”，进入“创建函数”页面。
- 步骤3 创建方式选择“容器镜像”，具体创建详情请参见[使用容器镜像创建函数](#)。
- 步骤4 容器镜像函数创建完成后，进入函数详情页面，选择“设置 > 常规设置”，单击“启用GPU”，配置GPU参数。

表 4-15 GPU 参数说明

| 参数名称      | 说明              |
|-----------|-----------------|
| GPU卡型     | 当前仅支持NVIDIA-T4。 |
| GPU规格（GB） | 支持1~16GB。       |

图 4-16 启用 GPU



----结束

相关文档

- 针对不同使用场景，函数工作流支持通过在线编写代码、上传代码文件或者使用容器镜像，创建事件函数和HTTP函数，且支持使用GPU计算资源。如何针对应用场景进行函数类型选型，详情请参见[函数类型选型](#)。
- 除使用控制台外，函数工作流支持通过API的方式创建函数，详情请参见[创建函数](#)。

### 4.5.3 定制运行时方式创建 GPU 函数

FunctionGraph支持运行时语言是“定制运行时”的函数启用GPU功能，进而帮助用户有效地解决GPU长驻使用方式导致的资源浪费、高成本、低弹性等问题，为用户提供更加便捷、高效的GPU计算服务。

#### 约束与限制

- 目前该功能仅“华东-上海一”和“土耳其-伊斯坦布尔”区域支持，其中“土耳其-伊斯坦布尔”区域需[提交工单](#)开通白名单后支持。
- GPU函数不支持的网段：192.168.64.0/18，192.168.128.0/18，10.192.64.0/18，10.192.128.0/18。
- GPU型号当前仅支持NVIDIA-T4。
- 定制运行时部署方式内置Python运行时版本，由于其内置cuda版本是11.6，因此函数需要基于cuda 11.6版本开发，使用其他版本的cuda请考虑使用自定义镜像函数。

#### 计费说明

函数工作流将按需收取GPU计量时间费用，详情请参见[函数工作流计费项](#)。

#### 创建 GPU 函数

- 步骤1** 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
- 步骤2** 单击右上方的“创建函数”，进入“创建函数”页面。
- 步骤3** 创建方式选择“创建空白函数”，具体创建详情请参见[创建空白函数](#)，本例“运行时”参数选择“定制运行时”。
- 步骤4** 定制运行时函数创建完成后，进入函数详情页面，选择“设置 > 常规设置”，单击“启用GPU”，配置GPU参数。

表 4-16 GPU 参数说明

| 参数名称      | 说明              |
|-----------|-----------------|
| GPU卡型     | 当前仅支持NVIDIA-T4。 |
| GPU规格（GB） | 支持1~16GB。       |

图 4-17 启用 GPU



----结束

相关文档

- 针对不同使用场景，函数工作流支持通过在线编写代码、上传代码文件或者使用容器镜像，创建事件函数和HTTP函数，且支持使用GPU计算资源。如何针对应用场景进行函数类型选型，详情请参见[函数类型选型](#)。
- 除使用控制台外，函数工作流支持通过API的方式创建函数，详情请参见[创建函数](#)。

# 5配置函数

## 5.1 函数配置概述

成功创建函数后，需配置函数运行所必须的资源与环境，确保函数满足业务需求。

### 函数配置流程

图5-1介绍了函数配置的总体流程。

图 5-1 函数配置流程



详细的步骤操作说明请参考表5-1。

表 5-1 配置函数操作步骤说明

| 序号 | 步骤       | 说明                                                                                                                                                                                                                                                       |
|----|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1  | 配置函数代码   | 函数代码包含函数实现业务需求的逻辑，函数工作流支持以下三种部署代码的方式： <ul style="list-style-type: none"><li>在线编辑函数代码：使用控制台代码编辑器在线编辑函数代码。</li><li>直接上传函数代码：直接上传本地开发完成的ZIP文件或JAR文件，适用于代码文件不超过40MB的场景使用。</li><li>通过OBS上传函数代码：通过输入OBS桶地址，上传OBS桶中的ZIP文件或JAR文件，适用于代码文件不超过300MB的场景使用。</li></ul> |
| 2  | 配置函数的依赖包 | 依赖包为函数提供依赖库、运行时环境及函数扩展的执行能力，确保函数代码能满足业务需求。                                                                                                                                                                                                               |

| 序号 | 步骤                             | 说明                                                                                                                                                                                                                                                                              |
|----|--------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 3  | <a href="#">配置函数的委托权限</a> （可选） | 如FunctionGraph服务需访问其他云服务以协同工作，则必须配置函数的委托权限，授权FunctionGraph使用这些云服务；若无需访问其他云服务，则保持默认配置，不使用任何委托。                                                                                                                                                                                   |
| 4  | <a href="#">配置函数的网络环境</a> （可选） | <p>函数创建后，默认启用“函数访问公网”功能，如实际业务需对函数的网络访问进行特定配置，应手动修改网络配置。</p> <p>函数支持访问以下两种网络环境：</p> <ul style="list-style-type: none"> <li>● 函数访问公网：支持函数访问公网资源，其应用场景包括调用外部API、访问其他云存储、集成第三方服务等。</li> <li>● 函数访问VPC网络：支持函数访问虚拟私有云（VPC）内的资源，对于需要高带宽、高性能和高可靠性的业务场景，建议手动启用“函数访问VPC内资源”功能。</li> </ul> |
| 5  | <a href="#">配置函数的触发器</a> （可选）  | <p>触发器是触发函数执行的一种机制，通过为函数创建触发器并定义其触发条件，当特定事件符合条件时，事件源将自动调用与触发器关联的函数。</p> <p>如FunctionGraph函数需自动响应特定事件，则应配置函数的触发器。使用前请参考<a href="#">FunctionGraph支持的触发事件</a>。</p>                                                                                                                 |
| 6  | <a href="#">在线调试函数</a>         | 完成函数的各项配置后，通过设置测试事件，验证并调试函数成功执行。                                                                                                                                                                                                                                                |

## 5.2 配置函数代码

### 5.2.1 在线编辑函数代码

本章节介绍如何通过函数工作流控制台在线编辑与部署函数代码，以及如何查看和修改函数执行入口。

#### 操作场景

编写函数业务代码是配置函数实现业务需求的第一步，FunctionGraph预装了适用于**Node.js、Python、PHP、定制运行时函数**的开发工具包。如果函数代码仅依赖于开发工具包库，则可以使用函数工作流控制台中“代码”页签下的代码编辑器在线编辑函数代码。如已在本地完成函数代码开发，请参考[直接上传函数代码](#)进行代码部署。

在线编辑代码区域支持工程方式的管理，请参考[在线编辑函数代码](#)使用。

#### 约束与限制

- 仅Node.js、Python、PHP和定制运行时函数支持在线编辑函数代码。  
使用Python语言在线编辑代码，需要输出中文时，请在代码编辑器中增加如下代码：

```
# -*- coding:utf-8 -*-  
import json  
def handler(event, context):  
    output = 'Hello message: ' + json.dumps(event,ensure_ascii=False)  
    return output
```

- 当代码编辑器中部署的代码大于20MB时将不展示代码，如图5-2所示，但仍可以正常测试函数代码。

图 5-2 编辑器不展示代码



## 在线编辑函数代码

- 步骤1** 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
- 步骤2** 在函数列表中，单击函数名称进入函数详情页。
- 步骤3** 在“代码”页签下，在线编辑函数代码。
- 步骤4** 完成代码编辑后，单击“部署代码”，控制台会将代码及相关的配置信息压缩到FunctionGraph服务可运行的代码文件中，无需其他操作。

如代码编辑区中的代码涉及修改，修改完成后需重新单击“部署代码”。

----结束

### 在线编辑代码区域管理：

如图5-3所示，在线编辑代码区域提供代码文件工程化管理能力，支持创建文件和文件夹、编辑和设置编码框中的代码。

- 文件：支持创建文件和文件夹功能。其中包括新建文件、新建文件夹、保存和关闭所有文件功能。
- 编辑：支持在编码框中，对代码进行撤销、恢复、剪切、复制、粘贴、查找和替换操作。
- 设置：支持设置编码框中代码字体大小、编码框主题颜色和是否自动格式化。

图 5-3 编码区管理



修改函数执行入口

函数执行入口是函数代码中指定的处理逻辑起点，通常是一个特定的函数或方法。当 FunctionGraph 函数被触发时，会从执行入口（handler）开始执行函数代码。

不同运行时的函数执行入口命名规则，请参见表5-2。

表 5-2 函数执行入口配置规则

| 运行时     | 函数执行入口配置规则                   | 示例                                     |
|---------|------------------------------|----------------------------------------|
| Node.js | [文件名].[执行函数名]                | myfunction.handler                     |
| Python  |                              |                                        |
| PHP     |                              |                                        |
| Java    | [包名].[类名].[执行函数名]            | com.xxxxx.exp.Myfunction.myHandler     |
| Go      | 与代码文件中的可执行文件名保持一致            | handler                                |
| C#      | [程序集名]::[命名空间].[类名]::[执行函数名] | CsharpDemo::CsharpDemo.Program::MyFunc |

以Python函数为例，介绍如何在函数详情页中查看和修改函数执行入口。

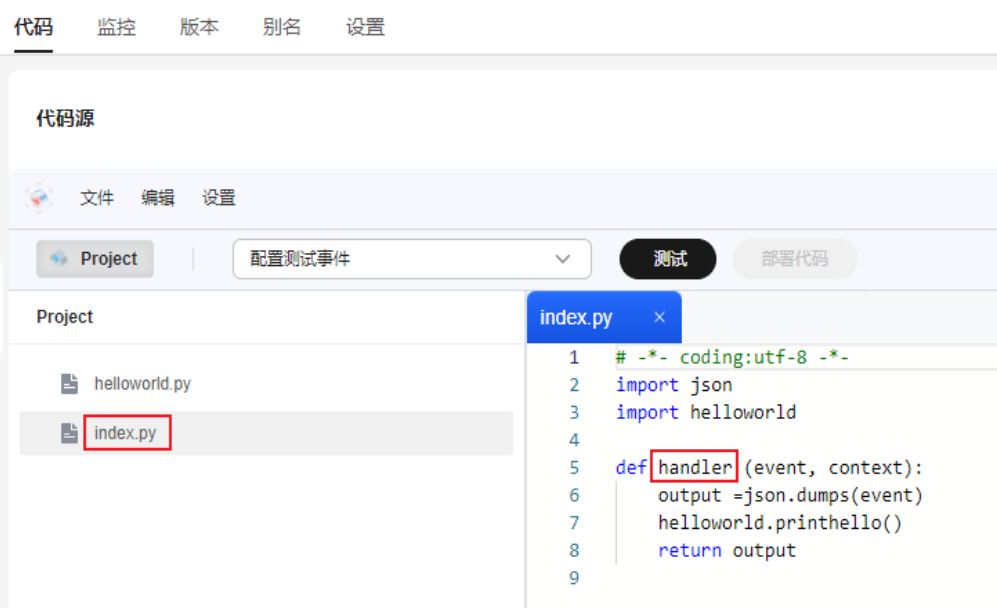
- 步骤1 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
- 步骤2 在函数列表中，单击函数名称进入函数详情页。
- 步骤3 单击“设置 > 常规设置”，如图5-4所示，可以查看和修改“函数执行入口”信息，修改完成后单击“保存”。

图 5-4 函数执行入口



**步骤4** 在“代码”页签，请根据设置的函数执行入口修改对应的文件名和执行函数名，如[图 5-5](#)所示。

图 5-5 文件名和执行函数名



----结束

相关文档

- 除使用控制台外，函数工作流支持通过API的方式管理函数代码，详情请参见[函数生命周期管理API](#)。
- 如需查看和配置函数的基本信息，请参考[配置函数的常规信息](#)。

5.2.2 直接上传函数代码

本章节介绍如何通过函数工作流控制台，直接上传函数代码文件进行部署。

操作场景

当函数的代码文件大小不超过40MB时，可以在函数工作流控制台直接上传本地开发完成的ZIP或JAR代码文件进行函数代码部署。运行时支持直接上传的代码文件类型请参考表5-3，请参见[函数工程打包规范](#)在本地打包代码文件。

表 5-3 直接上传代码支持的文件类型

| 运行时     | 上传ZIP文件 | 上传JAR文件 | 说明                                                                                                                                                                                                                                                                                                                                  |
|---------|---------|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Node.js | 支持      | 不支持     | -                                                                                                                                                                                                                                                                                                                                   |
| Python  | 支持      | 不支持     | 用python语言编写代码时，自己创建的依赖包名不能与python标准库同名，否则会提示module加载失败，例如“json”、“lib”、“os”等。                                                                                                                                                                                                                                                        |
| Java    | 支持      | 支持      | 部署Java代码文件支持上传ZIP文件或JAR文件。<br><b>上传JAR文件</b> <ul style="list-style-type: none"><li>如果函数代码没有引入依赖包，可以直接上传函数代码JAR文件。</li><li>如果函数代码引入了依赖包，可以在创建函数后先配置函数依赖包，再上传函数代码JAR文件部署。</li></ul> <b>上传ZIP文件</b><br>如果函数代码引入了依赖包，也可以制作并上传包含所有依赖包和函数代码JAR文件的ZIP代码文件。具体示例可参见 <a href="#">使用IDEA工具创建普通Java项目</a> 、 <a href="#">使用IDEA工具创建maven项目</a> 。 |
| Go      | 支持      | 不支持     | 编译后的动态库文件名称必须与函数执行入口的插件名称保持一致，例如：动态库名称为testplugin.so，则“函数执行入口”命名为testplugin.Handler。                                                                                                                                                                                                                                                |
| C#      | 支持      | 不支持     | -                                                                                                                                                                                                                                                                                                                                   |
| PHP     | 支持      | 不支持     | -                                                                                                                                                                                                                                                                                                                                   |
| 定制运行时   | 支持      | 不支持     | -                                                                                                                                                                                                                                                                                                                                   |
| Cangjie | 支持      | 不支持     | -                                                                                                                                                                                                                                                                                                                                   |

约束与限制

- 自行上传的代码文件编码格式建议设置为UTF-8。
- 直接上传的代码文件大小限制为40MB以内，如文件大小超过该限制，请使用[从OBS上传函数代码](#)的方式上传代码文件。
- 如果代码中包含敏感信息（如账户密码等），请自行加密，以防信息泄露。

- 当代码编辑器中部署的代码大于20MB时将不展示代码，如图5-6所示，但仍可以正常测试函数代码。

图 5-6 编辑器不展示代码



## 上传函数代码

- 步骤1** 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
- 步骤2** 在函数列表中，单击函数名称进入详情页。
- 步骤3** 选择“代码”页签，右侧选择“上传代码 > Zip文件”或“上传代码 > JAR文件”。
- 步骤4** 在弹出的窗口中，单击“添加文件”，选择本地函数代码文件。
- 步骤5** （可选，仅“拉美-圣保罗”区域支持配置）可根据实际需求选择使用KMS静态加密函数代码：
  - （默认）functiongraph/default：函数会在DEW服务中自动创建默认密钥，您无需手动创建。  
首次使用默认密钥进行加解密，需确保函数委托中拥有以下权限：  
kms:dek:decrypt、kms:dek:create、kms:cmk:create、kms:cmk:get、kms:cmk:list。
  - 使用用户主密钥：支持选择已创建的密钥加密函数代码，创建用户主密钥可参考[创建自定义密钥](#)。  
使用用户主密钥，需确保函数委托中拥有以下权限：kms:dek:decrypt、kms:dek:create、kms:cmk:get、kms:cmk:list。

### ⚠ 注意

若选择“使用用户主密钥”，在后续使用中请勿在DEW服务中删除用于函数加密的主密钥，否则将因加密数据无法解密而导致函数执行失败。

### 说明

请参考[JSON视图配置自定义策略](#)前往IAM控制台将“委托权限策略”（以选择默认密钥为例）添加至函数委托中，并参考[配置函数委托](#)为该函数配置委托权限。

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Action": [
        "kms:dek:decrypt",
        "kms:dek:create",
        "kms:cmk:create",
        "kms:cmk:get",
        "kms:cmk:list"
      ],
      "Effect": "Allow"
    }
  ]
}
```

**步骤6** 文件上传完成后，单击“确定”，代码将自动部署到代码编辑区中。

如需修改代码编辑区中的代码，请在每次修改完成后单击“部署代码”，进行代码更新部署。如需修改函数执行入口，请参见[修改函数执行入口](#)。

----结束

## 5.2.3 通过 OBS 上传函数代码

本章节介绍如何将存储在OBS中的函数代码文件，上传至函数工作流控制台。

### 操作场景

函数工作流控制台支持通过输入OBS地址，上传OBS桶中文件大小不超过300MB的函数代码文件对象，支持上传ZIP格式的代码文件进行函数代码部署。

当前所有运行时均支持从OBS上传函数代码，请参见[函数工程打包规范](#)在本地打包代码文件并上传至OBS桶。

### 约束与限制

- 请确保函数与OBS桶所在区域一致。
- 上传至OBS桶中的代码文件编码格式建议设置为UTF-8。
- 如OBS桶中的代码文件大小超过300MB，或代码文件解压后的源代码大小超过1.5G，请[提交工单](#)处理。
- 当代码编辑器中部署的代码大于20MB时将不展示代码，如[图5-7](#)所示，但仍可以正常测试函数代码。

图 5-7 编辑器不展示代码



## 前提条件

已将函数代码文件上传至OBS桶，并已设置匿名用户对该对象的读取权限，权限开启方法请参见[对所有账号授予指定对象的读权限](#)。

## 上传函数代码

- 步骤1** 登录[对象存储服务控制台](#)，请参考[通过URL访问OBS对象](#)复制所需的代码文件对象URL。
- 步骤2** 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
- 步骤3** 在函数列表中，单击函数名称进入函数详情页。
- 步骤4** 在“代码”页签，右侧选择“上传代码 > OBS地址”。
- 步骤5** 在弹出的窗口中，输入复制的OBS桶对象URL。
- 步骤6** （可选，仅“拉美-圣保罗一”区域支持配置）可根据实际需求选择使用KMS静态加密函数代码：
- （默认）functiongraph/default：函数会在DEW服务中自动创建默认密钥，您无需手动创建。  
首次使用默认密钥进行加解密，需确保函数委托中拥有以下权限：  
kms:dek:decrypt、kms:dek:create、kms:cmk:create、kms:cmk:get、kms:cmk:list。
  - 使用用户主密钥：支持选择已创建的密钥加密函数代码，创建用户主密钥可参考[创建自定义密钥](#)。  
使用用户主密钥，需确保函数委托中拥有以下权限：kms:dek:decrypt、kms:dek:create、kms:cmk:get、kms:cmk:list。

### 注意

若选择“使用用户主密钥”，在后续使用中请勿在DEW服务中删除用于函数加密的主密钥，否则将因加密数据无法解密而导致函数执行失败。

### 说明

请参考[JSON视图配置自定义策略](#)前往IAM控制台将“委托权限策略”（以选择默认密钥为例）添加至函数委托中，并参考[配置函数委托](#)为该函数配置委托权限。

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Action": [
        "kms:dek:decrypt",
        "kms:dek:create",
        "kms:cmk:create",
        "kms:cmk:get",
        "kms:cmk:list"
      ],
      "Effect": "Allow"
    }
  ]
}
```

- 步骤7** OBS地址输入完成后，单击“确定”，代码将自动部署到代码编辑区中。

如需修改代码编辑区中的代码，请在每次修改完成后单击“部署代码”，进行代码更新部署。如需修改函数执行入口，请参见[修改函数执行入口](#)。

----结束

## 5.3 配置函数的依赖包

### 5.3.1 函数依赖包概述

#### 函数依赖包简介

函数依赖包包含支持函数业务代码运行的公共库，可以将代码所需的公共库封装成依赖包进行单独管理，便于多函数共享，同时也能有效缩减函数代码在部署与更新过程中的体积。

FunctionGraph支持对同一依赖包进行版本迭代，即同一依赖包可拥有多个版本，便于系统化管理。

函数依赖包分为公共依赖包与私有依赖包。具体配置指导请参考[为函数配置依赖包](#)章节。

#### 公共依赖包

公共依赖包是FunctionGraph内置提供的依赖包，可在函数代码中直接添加所需运行时的依赖包使用，以支持代码实现业务逻辑。

相较于私有依赖包，公共依赖包有以下优势：

- 开箱即用：免去依赖包构建和上传的步骤，直接在函数详情页添加即可使用。无需过多关注如何配置代码运行环境，使您能更专注于代码和业务逻辑的优化。
- 减少延迟：FunctionGraph将公共依赖缓存在平台内，相较于私有依赖包，代码冷启动时无需额外时间从存储服务获取依赖文件，减少了网络延迟。
- 简化操作：私有依赖的文件大小限制为300MB，处理大文件时需拆分并多次上传；而公共依赖包不受300MB的文件限制，简化了添加与删除的操作。

#### 私有依赖包

私有依赖包是指在本地将业务代码所需的公共库自行封装的包，即自行创建的依赖包。各运行时函数的私有依赖包制作示例请参见[制作函数的私有依赖包](#)。

#### 支持引入的依赖库说明

FunctionGraph除了支持配置包外，还支持直接在代码中引入标准库和部分非标准库使用。

- 标准库  
对于标准库，无论是在线编辑或是本地开发代码打包上传至FunctionGraph，均可以直接在代码中引入使用。
- FunctionGraph内置的非标准库  
FunctionGraph内置了部分运行时的非标准库，如[表5-4](#)、[表5-5](#)所示。使用方法同标准库，在编写代码时直接引入即可。

表 5-4 Node.js Runtime 集成的三方件

| 名称              | 功能                                                                                                                   | 版本号     |
|-----------------|----------------------------------------------------------------------------------------------------------------------|---------|
| q               | 异步方法封装。                                                                                                              | 1.5.1   |
| co              | 异步流程控制。                                                                                                              | 4.6.0   |
| lodash          | 常用工具方法库。                                                                                                             | 4.17.10 |
| esdk-obs-nodejs | OBS sdk。                                                                                                             | 2.1.5   |
| express         | 极简web开发框架。                                                                                                           | 4.16.4  |
| fgs-express     | 在FunctionGraph和API Gateway之上使用现有的Node.js应用程序框架运行无服务器应用程序和REST API。提供的示例允许您使用Express框架轻松构建无服务器Web应用程序/服务和RESTful API。 | 1.0.1   |
| request         | 简化HTTP调用，支持HTTPS并默认遵循重定向。                                                                                            | 2.88.0  |

在函数代码中使用Node.js运行时引入依赖库示例如下：

```
const ObsClient = require('esdk-obs-nodejs');
```

表 5-5 Python Runtime 支持的非标准库

| 模块       | 功能                             | 版本号                                                                                                                           |
|----------|--------------------------------|-------------------------------------------------------------------------------------------------------------------------------|
| dateutil | 用于处理日期和时间的高级库，支持日期解析、格式化和时区操作。 | 2.6.0                                                                                                                         |
| requests | HTTP库，用于发送HTTP请求和处理响应。         | 2.7.0                                                                                                                         |
| httplib2 | HTTP客户端库，支持HTTP/HTTPS请求、认证和代理。 | 0.10.3                                                                                                                        |
| numpy    | 用于科学计算的库，支持高效的数组操作和数学函数。       | Python 2.7,<br>numpy==1.16.6<br>Python 3.6,<br>numpy==1.18.5<br>Python 3.9,<br>numpy==1.18.5<br>Python 3.10,<br>numpy==1.24.2 |
| redis    | 用于连接和操作Redis数据库的库，支持多种数据结构。    | 2.10.5                                                                                                                        |

| 模块        | 功能                                   | 版本号   |
|-----------|--------------------------------------|-------|
| ObsClient | 对象存储服务（OBS）的Python客户端，用于管理对象存储资源。    | 3.0.3 |
| smnsdk    | 消息通知服务（SMN）的Python SDK，用于访问公有云SMN服务。 | 1.0.1 |

- 其他第三方依赖库

需要在本地将依赖的第三方库打包为ZIP文件，上传至OBS桶或直接在函数控制台上传，在“代码”页签下添加相关的依赖包，即可在函数代码中使用。详细操作步骤请参见[为函数配置依赖包](#)。

## 相关文档

除使用控制台外，函数工作流支持通过API的方式管理函数依赖包，详情请参见[函数依赖包API](#)。

### 5.3.2 制作函数的私有依赖包

本章节介绍如何在本地制作函数的私有依赖包。

## 约束与限制

- 依赖包的文件大小限制为300MB，文件限制数为30000。
- 依赖包内文件名不能以~结尾，例如“module~”。
- 如果安装的依赖模块需要添加依赖库，请将依赖库归档到zip依赖包文件中，例如添加.dll、.so、.a等依赖库。
- **制作函数依赖包推荐在Huawei Cloud EulerOS 2.0环境中进行。**若所需依赖涉及操作系统相关的依赖包，使用其他操作系统环境打包时，可能因底层依赖库的差异而出现找不到动态链接库的问题。

## 搭建 EulerOS 环境

推荐在EulerOS环境中制作函数依赖包，EulerOS是基于开源技术的企业级Linux操作系统软件，具备高安全性、高可扩展性、高性能等技术特性，能够满足客户IT基础设施和云计算服务等多业务场景需求。

此处推荐[Huawei Cloud EulerOS](#)，可选择以下方法搭建环境：

- 在华为云购买一台EulerOS的ECS弹性云服务器，请参见[购买并登录Linux弹性云服务器](#)。在基础配置环节选择公共镜像时，选择“Huawei Cloud EulerOS操作系统”和具体的镜像版本。
- 下载[EulerOS镜像](#)，在本地使用虚拟化软件搭建EulerOS系统的虚拟机。

## 制作函数依赖包

以下为不同运行时函数制作依赖包示例。

## 为 Node.js 函数制作依赖包

制作依赖包前，请确认环境中已安装与函数运行时相匹配版本的Node.js。以Node.js 8.10安装MySQL依赖包为例，其他Node.js版本和依赖包制作过程相同。

**请注意不要使用CNPM命令制作Node.js依赖包。**

**步骤1** 执行如下命令，为Nodejs 8.10安装MySQL依赖包。

```
npm install mysql --save
```

命令执行后，在当前目录下会生成一个node\_modules文件夹。

**步骤2** 使用以下命令打包node\_modules文件夹生成ZIP包，即生成最终需要的依赖包。

```
zip -rq mysql-node8.10.zip node_modules
```

----结束

如需同时封装多个依赖包，建议参考以下步骤操作：

**步骤1** 新建一个package.json文件，在package.json中填入如下内容。

```
{
  "name": "test",           //自定义依赖包名称
  "version": "1.0.0",       //依赖包版本号
  "dependencies": {
    "redis": "~2.8.0",
    "mysql": "~2.17.1"
  }
}
```

**步骤2** 在包含package.json文件的目录下，执行如下命令。

```
npm install --save
```

**步骤3** 再将node\_modules打包成zip，即可生成一个既包含MySQL也包含redis的依赖包。

```
zip -rq node8.10.zip node_modules
```

----结束

## 为 Python 函数制作依赖包

制作依赖包前，请确认环境中已安装与函数运行时相匹配版本的Python。

以Python3.12安装PyMySQL依赖包为例，其他Python版本和依赖包制作过程相同。

**步骤1** 执行以下命令，指定PyMySQL依赖包的安装路径为本地的/tmp/pymysql下。

```
pip install PyMySQL --root /tmp/pymysql
```

**步骤2** 执行成功后，执行以下命令进入指定目录。

```
cd /tmp/pymysql/
```

**步骤3** 进入子目录直到site-packages路径下（一般路径为lib/python3.12/site-packages/，若此路径下无安装的依赖文件，请使用find命令找到并进入库文件所在路径），接下来执行以下命令压缩依赖文件。

所生成的包即为最终需要的依赖包。

```
zip -rq pymysql.zip *
```

----结束

### 📖 说明

如果需要安装存放在本地的wheel安装包，可执行以下命令：

```
pip install piexif-1.1.0b0-py2.py3-none-any.whl --root /tmp/piexif
//安装包名称以piexif-1.1.0b0-py2.py3-none-any.whl为例，请以实际安装包名称为准
```

## 为 PHP 函数制作依赖包

制作依赖包前，请确认环境中已安装与函数运行时相匹配版本的PHP。以PHP 7.3通过composer安装protobuf3.19依赖包为例，默认环境中已经安装了composer，其他PHP版本和依赖包制作过程相同。

**步骤1** 新建一个composer.json文件，在composer.json中填入以下内容。

```
{
  "require": {
    "google/protobuf": "^3.19"
  }
}
```

**步骤2** 执行如下命令。

```
Composer install
```

命令执行后，当前目录下会生成一个vendor文件夹，文件夹中有autoload.php、composer和google三个文件夹。

**步骤3** 使用以下命令生成zip包。

```
zip -rq vendor.zip vendor
```

----结束

如需同时封装多个依赖包，在composer.json文件中指定需要的依赖，把生成的vendor文件整体打包成zip文件上传即可。

### 📖 说明

在PHP项目中，使用composer下载的第三方依赖，需通过require "./vendor/autoload.php" 进行加载。FunctionGraph默认将上传的ZIP包解压后的内容，置于与项目代码同一级别的目录下。

## 为 Java 函数制作依赖包

使用Java编译型语言开发函数时，依赖包需要在本地编译。如何为Java函数制作依赖包请参考[Java函数开发指南（使用IDEA工具普通Java项目）](#)。

## 为 Go 函数制作依赖包

使用Go编译型语言开发函数时，依赖包需要在本地编译。如何为Go函数制作依赖包请参考[Go函数开发](#)。

## 为 C#函数制作依赖包

使用C#编译型语言开发函数时，依赖包需要在本地编译。如何为C#函数制作依赖包请参考[C#函数开发](#)。

## 相关文档

- 在本地完成依赖包制作后，请参考[为函数配置依赖包](#)在函数工作流控制台进行函数依赖包的创建和配置。

- 除使用控制台外，函数工作流支持通过API的方式管理函数依赖包，详情请参见[函数依赖包API](#)。

### 5.3.3 为函数配置依赖包

本章节介绍如何通过函数工作流控制台，对函数依赖包进行创建、配置和删除操作。

#### 操作场景

依赖包为函数提供依赖库、运行时环境及扩展能力。关于函数依赖包的相关介绍请参考[函数依赖包概述](#)。

私有依赖包需先在函数工作流控制台创建后再配置使用，公共依赖包可直接配置使用。如函数代码中需使用标准库和FunctionGraph内置的部分非标准库，可参考[支持引入的依赖库说明](#)在函数代码中引入使用。

#### 约束与限制

- 一个函数最多支持添加20个依赖包。
- 正在被函数使用的依赖包无法被删除。
- 各个依赖包与代码文件之间，应避免出现同名的目录或文件。例如依赖包depends.zip中包含index.py文件，若采用在线编辑方式的编写代码，函数执行入口为index.handler，函数执行将生成名为index.py的代码文件，与依赖包内的index.py文件重名，可能导致文件覆盖或合并错误。

#### 前提条件

如需为函数配置私有依赖包，应先在本地完成[函数私有依赖包的制作](#)。

#### 创建函数依赖包

仅函数私有依赖包需执行创建函数依赖包操作，公共依赖包可直接进行[配置函数依赖包操作](#)。

- 步骤1** 登录[函数工作流控制台](#)，在左侧导航栏选择“函数 > 依赖包管理”，进入“依赖包管理”界面。
- 步骤2** 单击“创建依赖包”，弹出“创建依赖包”对话框。
- 步骤3** 参考[表5-6](#)配置依赖包参数。

表 5-6 依赖包配置参数说明

| 参数    | 说明                                                                            |
|-------|-------------------------------------------------------------------------------|
| 依赖包名称 | 自定义依赖包名称，便于识别不同的依赖包。<br>可包含字母、数字、下划线、点和中划线，长度不超过96个字符。<br>以大/小写字母开头，以字母或数字结尾。 |
| 运行时   | 选择依赖包使用的运行时语言。<br>支持选择Cangjie、Node.js、Python、Java、PHP、Go、C#和Custom（定制运行时）。    |

| 参数     | 说明                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 代码上传方式 | 分为在线构建、上传ZIP文件和从OBS上传ZIP文件。 <ul style="list-style-type: none"><li>上传ZIP文件：单击“添加文件”，上传ZIP代码文件。上传的文件大小限制为10MB。</li><li>从OBS上传文件：填写OBS桶中zip文件的对象“OBS链接”。请注意，OBS桶与函数所在区域需一致。如何在OBS中上传文件对象请参见<a href="#">上传对象</a>，OBS存储链接获取方法请参见<a href="#">通过URL访问对象</a>。</li><li>在线构建：使用该功能需创建一个包含SWR Admin权限的<a href="#">函数委托</a>。<br/>当前支持为运行时“Node.js”和“Python”在线构建依赖包，仅需上传Node.js的package.json或Python的requirements.txt依赖文件，即可在线构建一个含有这些依赖的依赖包。</li></ul> |
| 描述     | 自定义填写依赖包的描述信息。                                                                                                                                                                                                                                                                                                                                                                                                                            |

**步骤4** 单击“创建”，完成依赖包的创建，进入依赖包详情页。默认首次创建的依赖包版本为“1”。

----结束

为依赖包创建版本

在依赖包详情页，如[图5-8](#)所示可以查看当前依赖包下的所有版本和版本相关信息。支持针对同一依赖包，进行不同版本的系统化管理。

- 单击“创建版本”，填写相关信息，可以创建新的依赖包版本，版本号默认+1。创建成功后详情页默认展示最新版本的依赖包。
- 单击“版本”列下具体的版本号，可以切换查看版本的信息。
- 单击版本号所在行的“下载”和“删除”，可以下载和删除该版本。

图 5-8 依赖包版本管理



| ID | 名称                 | 版本    | 创建时间                | 最后修改时间              | 操作    |
|----|--------------------|-------|---------------------|---------------------|-------|
| 2  | my-first-function  | 1.0.0 | 2025-09-08 10:00:00 | 2025-09-08 10:00:00 | 下载 删除 |
| 1  | my-second-function | 1.0.0 | 2025-09-08 10:00:00 | 2025-09-08 10:00:00 | 下载 删除 |

配置函数依赖包

如果函数配置的私有依赖包体积较大，建议在函数详情页的“设置 > 常规设置”中调整函数执行时间，在原有基础上适当延长超时时间，以防止函数因执行超时而执行失败。

配置函数依赖包操作步骤：

- 步骤1** 登录[函数工作流控制台](#)，在左侧导航栏选择“函数 > 函数列表”，进入函数列表界面。
- 步骤2** 单击函数名称，进入函数详情界面。
- 步骤3** 在“代码”页签的最下方，单击“添加依赖包”，弹出“选择依赖包”对话框。

**步骤4** 参考[表5-7](#)选择依赖包配置，单击“确定”完成依赖包添加。

表 5-7 依赖包配置说明

| 参数    | 说明                                                                                                                                                      |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| 运行时   | 默认展示当前函数的运行时语言，无法修改。                                                                                                                                    |
| 依赖包源  | 根据实际业务，选择“公共依赖包”或“私有依赖包”。 <ul style="list-style-type: none"><li>公共依赖包：函数工作流服务提供的系统内置依赖包。</li><li>私有依赖包：用户通过<a href="#">创建函数依赖包</a>步骤自行上传的依赖包。</li></ul> |
| 依赖包名称 | 选择当前运行时语言下所需的依赖包。                                                                                                                                       |
| 版本    | 选择当前依赖包的具体版本。                                                                                                                                           |

**步骤5** 代码依赖包添加成功后如[图5-9](#)所示。

图 5-9 成功添加代码依赖包



----结束

删除依赖包

依赖包无法在“依赖包管理”界面直接删除，如需删除，请先删除依赖包下的所有版本，当所有版本全部删除完成后，依赖包会自动删除。

- 步骤1** 登录函数工作流控制台，在左侧导航栏选择“函数 > 依赖包管理”，进入“依赖包管理”界面。
- 步骤2** 单击依赖包名称，进入依赖包详情页。
- 步骤3** 如[图5-10](#)所示，单击“版本管理”下依赖包版本号所在行的“删除”，即可删除该版本。

图 5-10 删除依赖包版本



----结束

相关文档

除使用控制台外，函数工作流支持通过API的方式管理函数依赖包，详情请参见[函数依赖包API](#)。

## 5.4 配置函数的委托权限

本章节介绍如何创建、配置和修改函数的委托权限。

### 操作场景

当函数工作流服务需与其他云服务协同工作时，应创建云服务委托，授权FunctionGraph使用这些云服务，并在FunctionGraph中为函数配置委托权限，以实现与其他云服务的协同工作。

对于常见函数应用场景的委托配置，可参考[常见应用场景授权项](#)配置。

### 默认委托

为方便创建和使用函数，当华为云账号下无函数默认委托时，函数工作流服务提供快速创建默认委托的功能。

默认委托包含了函数工作流服务需要访问的部分云资源权限，如[表5-8](#)所示，其中相关服务的[细粒度最小使用权限](#)介绍请参见[表5-11](#)。

表 5-8 默认委托权限说明

| 权限名称                    | 描述                                                  |
|-------------------------|-----------------------------------------------------|
| fgs_default_region_role | 函数默认委托使用的项目级服务策略，包含虚拟私有云服务（VPC）、弹性文件服务（SFS）等的最小权限集。 |
| fgs_default_global_role | 函数默认委托使用的全局服务策略，包含统一身份认证服务（IAM）和对象存储服务（OBS）的最小权限集。  |
| SWR Admin               | 容器镜像服务（SWR）管理员，拥有该服务下的所有权限。                         |
| DIS User                | 数据接入服务（DIS）通道使用权限。                                  |

#### 创建默认委托的3种方式：

方式一：首次登录自动弹出创建提示

- 首次登录函数工作流控制台，函数数量为0且无默认委托，在浏览“总览”页时，系统会弹窗提示是否需要创建默认委托，单击“同意授权并创建”即可自动创建名为“fgs\_default\_agency”的默认委托。

图 5-11 弹窗提示



方式二：创建函数时单击创建

- 进入创建函数页面，“委托名称”参数的下拉列表中提供创建默认委托的选项，如图5-12所示，单击后会弹出授权窗口，单击“确定”，即可自动创建名为“fgs\_default\_agency”的默认委托。

图 5-12 创建函数时创建默认委托



方式三：在函数详情页单击创建

- 进入函数详情页，如图5-13所示，选择“设置 > 权限”，单击“函数配置委托”参数的下拉列表，单击后会弹出授权窗口，单击“确定”，即可自动创建名为“fgs\_default\_agency”的默认委托。

图 5-13 权限中创建默认委托



说明

已创建默认委托后，以上3种方式不会再有创建默认委托的提示。

创建函数委托

以函数工作流服务获取VPC访问权限为例，当配置函数访问VPC内资源时，需为函数工作流服务创建委托并授予VPC访问权限，授权函数工作流服务使用VPC服务，实现云服务间的协同工作。

- 步骤1 登录[统一身份认证服务（IAM）控制台](#)。
- 步骤2 在左侧导航窗格中，选择“委托”页签，单击右上方的“创建委托”。

图 5-14 创建委托



- 步骤3 配置委托参数，完成参数配置后如[图5-15](#)所示，单击“完成”，系统弹出“创建成功”提示，单击“立即授权”。
- 委托名称：自定义填写，以“serverless-trust”为例。
  - 委托类型：选择“云服务”。
  - 云服务：选择“函数工作流 FunctionGraph”。
  - 持续时间：以选择“永久”为例，实际使用时请根据业务需求填写。
  - 描述（可选）：填写描述信息。

图 5-15 填写基本信息

委托 / 创建委托

★ 委托名称

serverless-trust

★ 委托类型

☐ 普通账号

将账号内资源的操作权限委托给其他华为云账号。

☒ 云服务

将账号内资源的操作权限委托给华为云服务。

★ 云服务

函数工作流 FunctionGraph

★ 持续时间

永久

描述

请输入委托信息。

0/255

完成

取消

**步骤4** 进入“选择策略”页面，如图5-16所示，在右方搜索框中搜索需要添加的权限并勾选，此处以添加VPC Administrator权限为例。勾选完成后单击“下一步”。

图 5-16 选择策略



表 5-9 委托权限示例

| 权限名称              | 使用描述/场景     |
|-------------------|-------------|
| VPC Administrator | 虚拟私有云服务管理员。 |

**步骤5** 进入“设置最小授权范围”页面，如图5-17所示，建议根据业务部署的区域选择“指定区域项目资源”，并勾选所需区域。勾选完成后单击“确定”，进入“授权成功”界面。

图 5-17 根据业务需要选择对应的权限



----结束

说明

如果系统策略不满足授权要求，管理员可以通过“可视化视图”和“JSON视图”两种方式创建自定义策略，并通过给用户组授予自定义策略来进行精细的访问控制，详情请参见[创建自定义策略](#)。

配置函数委托

- 步骤1 登录[函数工作流控制台](#)，在左侧导航栏选择“函数 > 函数列表”。
- 步骤2 选择待配置委托的函数，单击进入函数详情页。
- 步骤3 选择“设置 > 权限”，参考[表5-10](#)配置函数委托。配置完成后单击“保存”。

表 5-10 配置函数委托参数说明

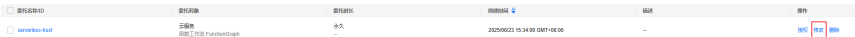
| 参数     | 说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 委托     | 在下拉框中选择已创建的函数委托。若无可用委托，可单击右侧“创建委托”进入统一身份认证服务IAM创建，具体操作步骤请参见 <a href="#">创建函数委托</a> 。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 函数执行委托 | <p>勾选“为函数执行单独设置委托”时，将弹出此参数进行配置，如<a href="#">图5-18</a>所示，配置后可以通过函数执行入口方法中的context参数获取具有委托中权限的Token或者SecurityAccessKey、SecuritySecretKey、SecurityToken，用于访问其他云服务，代码示例可参考<a href="#">6.4-环境变量应用示例</a>。</p> <p>函数执行委托与函数配置委托可独立设置，以确保业务使用过程中委托权限的明确管理。</p> <p><b>图 5-18 设置委托</b></p> <p><b>权限</b> ①</p> <p>用户委托函数工作流去访问其他的云服务。举例：如果用户函数需要访问OBS、DMS、DIS等服务，则需要提供权限委托名称，如果用户函数不访问任何云服务，则不用提供委托名称。</p> <p>函数配置委托 未使用任何委托 <input type="button" value="创建委托"/> <input checked="" type="checkbox"/> 为函数执行单独设置委托</p> <p>如果函数需要创建DMS或DIS触发器，则需要配置具有DMS或DIS访问权限的委托。</p> <p>函数执行委托 未使用任何委托</p> <p>配置后用户可以通过函数执行入口方法中的context参数获取具有委托中权限的token、ak、sk，用于访问其他云服务</p> <p><input type="button" value="保存"/></p> |

----结束

修改函数委托

如需修改委托的权限、持续时间和描述参数，如[图5-19](#)所示，需在[统一身份认证服务（IAM）控制台](#)中对相应的FunctionGraph委托进行修改。

图 5-19 修改函数委托



FunctionGraph相关委托修改后，约10分钟生效（如context.getSecurityToken()更新）。通过context获取的委托相关信息有效期为24h，请注意在失效前及时刷新。

常见应用场景授权项

如需使用[表5-11](#)中的场景，即需与其他服务协同工作，应先[创建函数委托](#)，并[配置函数委托](#)。

创建委托时，授予的权限类型应根据实际业务需要进行调整，**生产环境中建议调整为细粒度最小使用权限**，以确保业务运行所需的同时，有效降低权限过大的潜在风险。

表 5-11 常见授权项选择

| 场景                 | 策略名称                         | 策略说明                                                                                  | 细粒度最小使用权限                                                                                                                           |
|--------------------|------------------------------|---------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| 使用自定义镜像            | SWR Admin                    | 容器镜像服务（SWR）管理员，拥有该服务下的所有权限。<br>创建自定义镜像请参见 <a href="#">使用容器镜像创建函数</a> 。                | 暂不支持                                                                                                                                |
| 挂载SFS文件系统（仅存量用户支持） | SFS Administrator            | 弹性文件服务（SFS）管理员，拥有该服务下的所有权限。<br>挂载SFS文件系统请参见 <a href="#">挂载SFS容量型文件系统（仅存量用户支持使用）</a> 。 | 暂不支持                                                                                                                                |
| 挂载SFS Turbo文件系统    | SFS Turbo ReadOnlyAccess     | 弹性文件服务SFS Turbo的只读权限。<br>挂载挂载sfs turbo文件系统请参见 <a href="#">挂载SFS Turbo文件系统</a> 。       | <ul style="list-style-type: none"> <li>sfsturbo:shares:getShare（查询单个文件系统详情）</li> <li>sfsturbo:shares:showFsDir（查询目录是否存在）</li> </ul> |
| 挂载ECS共享目录          | ECS ReadOnlyAccess           | 弹性云服务器（ECS）的只读访问权限。<br>挂载ECS共享目录请参见 <a href="#">挂载ECS共享目录</a> 。                       | ecs:cloudServers:get（查询云服务器详情）                                                                                                      |
| 配置预留实例策略           | AOM ReadOnlyAccess           | 应用运维管理服务（AOM）的只读权限。                                                                   | <ul style="list-style-type: none"> <li>aom:metric:get（查询指标详情）</li> <li>aom:metric:list（查询指标列表详情）</li> </ul>                         |
|                    | FunctionGraph ReadOnlyAccess | 函数工作流服务（FunctionGraph）的只读权限。                                                          | functiongraph:functionion:getConfig（查询函数配置）                                                                                         |
| 使用DIS触发器           | DIS Administrator            | 数据接入服务（DIS）管理员，拥有该服务下的所有权限。<br>创建DIS触发器请参见 <a href="#">使用数据接入服务（DIS）触发器</a> 。         | 暂不支持                                                                                                                                |
| 使用分布式消息触发器         | DMS ReadOnlyAccess           | 分布式消息服务（DMS）的只读权限。                                                                    | dms:instance:get（查看实例详情）                                                                                                            |

| 场景        | 策略名称                            | 策略说明                                                                                        | 细粒度最小使用权限                                                                                                                                                                                                                                                          |
|-----------|---------------------------------|---------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 配置跨域VPC访问 | VPC Administrator               | 虚拟私有云VPC管理员，拥有对VPC内所有资源执行任意操作的权限。<br>配置跨域VPC访问请参见 <a href="#">函数访问VPC内资源</a> 。              | <ul style="list-style-type: none"> <li>vpc:ports:get ( 查询端口 )</li> <li>vpc:ports:create ( 创建端口 )</li> <li>vpc:vpcs:get ( 查询VPC )</li> <li>vpc:subnets:get ( 查询子网 )</li> <li>vpc:vips:delete ( 虚拟IP解绑VM )</li> <li>vpc:securityGroups:get ( 查询安全组列表或详情 )</li> </ul> |
| DNS域名解析   | DNS ReadOnlyAccess              | 云解析服务 ( DNS ) 的只读权限，拥有该权限的用户仅能查看云解析服务资源。<br>调用DNS服务的接口解析内网域名请参见 <a href="#">解析DNS内网域名</a> 。 | <ul style="list-style-type: none"> <li>dns:recordset:get ( 查询租户Record Set资源 )</li> <li>dns:zone:get ( 查询租户zone )</li> <li>dns:recordset:list ( 查询Record Set列表 )</li> <li>dns:zone:list ( 查询zone列表 )</li> </ul>                                                     |
| 开启异步通知    | 目标服务为OBS时：<br>OBS Administrator | 对象存储服务 ( OBS ) 管理员，拥有该服务下的所有权限。<br>配置异步通知请参见 <a href="#">配置函数的异步调用策略</a> 。                  | <ul style="list-style-type: none"> <li>obs:bucket:HeadBucket ( 获取桶元数据 )</li> <li>obs:bucket:CreateBucket ( 创建桶 )</li> <li>obs:object:PutObject ( PUT上传、POST上传、复制对象、追加写对象、初始化上传段任务、上传段、合并段 )</li> </ul>                                                             |
|           | 目标服务为SMN时：<br>SMN Administrator | 消息通知服务 ( SMN ) 管理员，拥有该服务下的所有权限。<br>配置异步通知请参见 <a href="#">配置函数的异步调用策略</a> 。                  | <ul style="list-style-type: none"> <li>smn:topic:publish ( 发布消息 )</li> <li>smn:topic:list ( 查询主题列表 )</li> </ul>                                                                                                                                                    |

| 场景     | 策略名称                            | 策略说明                                                                       | 细粒度最小使用权限                                                                                                                                                                                                                                          |
|--------|---------------------------------|----------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|        | 目标服务为DIS时：<br>DIS Administrator | 数据接入服务（DIS）管理员，拥有该服务下的所有权限。<br><br>配置异步通知请参见 <a href="#">配置函数的异步调用策略</a> 。 | 暂不支持                                                                                                                                                                                                                                               |
| 使用OBS桶 | OBS Administrator               | 对象存储服务（OBS）管理员，拥有该服务下的所有权限。                                                | <ul style="list-style-type: none"><li>• obs:bucket:GetBucketLocation（获取桶区域位置）</li><li>• obs:bucket:ListAllMyBuckets（获取桶列表）</li><li>• obs:bucket:GetBucketNotification（获取桶的消息通知配置）</li><li>• obs:bucket:PutBucketNotification（设置桶的消息通知配置）</li></ul> |

## 5.5 配置函数的网络环境

本章节介绍如何通过函数工作流控制台，配置函数访问公网或访问VPC内资源。

### 操作场景

函数的网络访问能力如[表5-12](#)所示，可根据业务需求配置。

表 5-12 网络配置说明

| 网络配置                   | 说明                                                                                                                                                                                                                   |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">函数访问公网</a> | 函数访问公网的配置方式有以下两种： <ul style="list-style-type: none"><li>• <a href="#">使用默认网卡</a>：函数创建成功后，无需手动配置即可使用默认网卡访问公网。</li><li>• <a href="#">配置VPC内固定公网IP</a>：通过手动配置函数访问VPC，并在VPC内配置公网NAT网关绑定EIP，分配独占公网访问带宽，以实现公网访问。</li></ul> |

| 网络配置    | 说明                                                                                                                                                                                                         |
|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 函数访问VPC | 函数访问VPC的配置方式有以下两种： <ul style="list-style-type: none"><li>允许函数访问VPC内资源：函数可访问所选VPC内的资源，默认禁用公网访问权限。若需访问公网，请参考<a href="#">配置VPC内固定公网IP</a>的方式开放公网访问。</li><li>仅允许指定的VPC调用函数：函数仅接受所选VPC的调用请求，禁止通过公网调用。</li></ul> |

## 约束与限制

- 在一个账号下所有的函数最多能绑定4个不同的子网。
- 开启“函数访问VPC内资源”时，函数将禁用默认网卡并使用VPC绑定的网卡，是否允许公网访问由配置的VPC决定，开关“函数访问公网”将不生效。

## 配置函数访问公网

配置函数访问公网有以下两种方式：

### 使用默认网卡

函数创建成功后，“函数访问公网”功能默认开启，无需手动配置即可使用默认网卡访问公网，允许函数访问公网上的服务。

其公网访问带宽为用户间共享，不保证可靠性，适用于测试场景，生产环境建议[配置VPC内固定公网IP](#)。

## 配置 VPC 内固定公网 IP

在生产环境中如需配置函数访问公网，以及在函数访问公网服务时需要固定公网出口IP的场景（例如被访问服务需要白名单验证），可通过在VPC内配置公网NAT网关绑定EIP的方式实现。开启是否允许公网访问将由配置的VPC决定，开关“允许函数访问公网”将不再生效。

前提条件：

1. 已创建虚拟私有云和子网，请参见[创建虚拟私有云基本信息及默认子网](#)。
2. 已申请弹性公网IP，请参见[申请弹性公网IP](#)。
3. 已[配置函数访问VPC](#)。

创建公网NAT网关步骤：

- 步骤1** 在服务控制台左侧导航栏，选择“网络> NAT网关”进入NAT网关控制台，单击“购买公网NAT网关”。
- 步骤2** 在公网NAT网关购买页面，参考[购买公网NAT网关](#)输入相关信息，确认规格信息后提交，完成购买。
- 步骤3** 购买完成后，单击公网NAT网关名称进入详情页面，选择“[添加SNAT规则](#)”，单击“确定”完成配置。

----结束

配置函数访问 VPC

FunctionGraph支持函数访问自行创建的VPC内资源，或配置其他用户共享的VPC子网，共享VPC的相关介绍请参考[共享VPC](#)。

在函数工作流控制台中，配置函数访问VPC有以下两种选项：

- 允许函数访问VPC内资源：允许函数访问所选VPC内的资源，不再具有默认的公网访问权限。如需访问公网，请先开启此功能，再参见[配置VPC内固定公网IP](#)进行公网访问配置。
- 仅允许指定的VPC调用函数：仅允许通过指定的VPC调用函数，并禁止通过公网调用。此配置适用于需严格控制函数调用来源的场景，确保函数只能通过特定网络环境访问。

前提条件：

需在默认“default”安全组中配置如下要求的入方向规则和出方向规则，添加规则详情请参见[添加安全组规则](#)。

- 入方向规则：配置“策略”为“允许”、“协议端口”为“ICMP”、“源地址”设置的最小范围为函数所选VPC网段的规则。  
例如图5-20所示，函数的VPC网段为192.168.x.x/24，安全组的入方向需增加一个规则：“策略”选择“允许”、“协议端口”选择“ICMP”、“源地址”设置最小范围的网段为192.168.x.x/24。

图 5-20 入方向规则



- 出方向规则：配置“策略”为“允许”的相关业务端口规则。

步骤一：配置函数的委托权限

函数使用VPC访问能力时，必须为函数配置VPC管理权限的委托，配置委托权限的操作步骤请参见[配置委托权限](#)。以下为涉及的委托权限：

- VPC Administrator：使用VPC功能需要为函数配置“VPC Administrator”委托权限，或参见[表5-13](#)和[创建自定义策略](#)授予访问VPC需要配置的最小权限，确保函数能操作相关云服务。

表 5-13 最小授权项配置

| 权限   | 授权项              |
|------|------------------|
| 删除端口 | vpc:ports:delete |

| 权限    | 授权项              |
|-------|------------------|
| 查询端口  | vpc:ports:get    |
| 创建端口  | vpc:ports:create |
| 查询VPC | vpc:vpcs:get     |
| 查询子网  | vpc:subnets:get  |

- DNS ReadOnlyAccess：如VPC中配置了内网域名，需要为函数配置“DNS ReadOnlyAccess”委托权限以解析内网域名。

步骤二：开启函数访问VPC内资源

- 步骤1** 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
- 步骤2** 选择待配置的函数，单击函数名称进入函数详情页。
- 步骤3** 选择“设置 > 网络设置”，如[图5-21](#)所示，开启“函数访问VPC内资源”，并参考[表5-14](#)配置参数。

开启时，函数将使用配置的VPC所绑定的网卡进行网络访问，同时禁用函数工作流的默认网卡。

图 5-21 配置 VPC



表 5-14 网络配置参数说明

| 参数  | 说明                                                                           |
|-----|------------------------------------------------------------------------------|
| VPC | 必选参数。<br>选择需要访问的虚拟私有云VPC。创建虚拟私有云VPC和子网请参见 <a href="#">创建虚拟私有云基本信息及默认子网</a> 。 |

| 参数        | 说明                                                                                                                                                                                                                                                                                                                                         |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 子网        | 必选参数。<br>选择VPC下的子网。                                                                                                                                                                                                                                                                                                                        |
| 域名        | 可选参数。<br>如果函数需要通过内网域名访问VPC内的服务，可配置和VPC绑定的域名，复选框中域名可以配置多个。<br>以下为与配置域名相关的操作，请根据业务需求参考使用： <ul style="list-style-type: none"><li>创建内网域名请参见<a href="#">创建内网域名</a>。</li><li>配置函数实现域名解析请参见<a href="#">内网域名配置及验证</a>。当前函数仅支持对A记录集类型的域名解析，记录集添加请参见<a href="#">记录集类型及配置规则</a>。</li><li>函数如何访问VPC内的Redis请参见<a href="#">函数访问VPC内的Redis</a>。</li></ul> |
| VPC对等连接网段 | 可选参数。<br>您可以声明函数代码中使用到的VPC网段，用以检测是否与服<br>务使用的VPC网段冲突。                                                                                                                                                                                                                                                                                      |
| 开启IPv6    | 可选参数。当前仅“华东二”区域支持函数VPC配置开启IPv6。<br>在创建虚拟私有云VPC时，确保默认子网配置中支持开启IPv6，此处将自动开启，具体详情请参见 <a href="#">创建虚拟私有云和子网</a> 。<br>开启IPv6功能后，将自动为子网分配IPv6网段，暂不支持自定义设置IPv6网段。该功能一旦开启，将不能关闭，更多介绍请参见 <a href="#">IPv6网络</a> 。                                                                                                                                  |

----结束

## 共享 VPC

共享VPC是基于资源访问管理（Resource Access Manager，简称RAM）服务的机制，VPC的所有者可以将VPC内的子网共享给其他账号使用，实现网络资源跨租户共享。有关VPC子网共享的更多信息，请参见《虚拟私有云用户指南》的“[共享VPC](#)”相关内容。

在函数中可以配置其他用户共享的子网，从而能够在函数中访问该子网下的资源。如需在函数中访问其他用户共享给您的子网，请先确保该子网的拥有者已经为您配置了子网共享，配置步骤请参考[VPC子网共享给其他账号](#)。成功配置子网共享后，参考[配置函数访问VPC](#)，在函数的网络配置中选择共享的子网即可。如果后续VPC子网拥有者取消了共享，将无法在函数中访问该子网。

## 5.6 配置函数的触发器

## 5.6.1 函数触发器概述

### 函数触发器介绍

触发器是FunctionGraph中用于启动函数执行的重要机制。当其他云服务作为事件源检测到特定事件时，触发器会根据预定义的规则自动调用关联的函数，从而实现事件的实时处理。FunctionGraph支持的触发事件请参见[FunctionGraph支持的触发事件](#)。

每个事件传递到函数时，仅包含与该触发器相关的事件数据，事件数据以结构化的JSON格式传递，FunctionGraph会将事件数据转换为函数处理程序可以使用的对象。根据具体业务需求，一个函数可以配置一个或多个触发器，每个触发器都可以独立地调用函数。

### 函数触发器的调用方式

根据函数调用方式，函数的触发器可分为同步调用和异步调用。关于函数调用的更多说明请参见[调用函数](#)。

- 同步调用：客户端调用函数后，FunctionGraph会立即执行函数，等函数执行完成后向客户端返回响应和执行结果。
- 异步调用：客户端调用函数后，FunctionGraph会将请求排队，不等待函数的执行结果，直接向客户端返回响应，FunctionGraph会在系统空闲的情况下逐个处理排队的请求。

表 5-15 函数触发器的调用机制

| 触发器                          | 调用方式                                                  |
|------------------------------|-------------------------------------------------------|
| API网关服务（APIG专享版）触发器          | 默认为同步调用，但可以修改为异步调用，具体配置方式请参考 <a href="#">配置异步调用</a> 。 |
| API Connect（APIC）触发器         | 默认为同步调用，但可以修改为异步调用，具体配置方式请参考 <a href="#">配置异步调用</a> 。 |
| 定时触发器（TIMER）触发器              | 默认为同步调用，且不可修改。                                        |
| 云数据库 GeminiDB<br>DynamoDB触发器 | 默认为同步调用，且不可修改。                                        |
| 云审计服务（CTS）触发器                | 默认为异步调用，且不可修改。                                        |
| 文档数据库服务（DDS）触发器              | 默认为异步调用，且不可修改。                                        |
| 数据接入服务（DIS）触发器               | 默认为异步调用，且不可修改。                                        |
| 分布式消息服务 Kafka版（KAFKA）触发器     | 默认为异步调用，且不可修改。                                        |
| 开源Kafka（OPENSOURCEKAFKA）触发器  | 默认为异步调用，且不可修改。                                        |

| 触发器                            | 调用方式           |
|--------------------------------|----------------|
| 分布式消息服务 RabbitMQ版（RABBITMQ）触发器 | 默认为异步调用，且不可修改。 |
| 云数据库 GeminiDB Mongo触发器         | 默认为异步调用，且不可修改。 |
| 设备接入（IoTDA）触发器                 | 默认为异步调用，且不可修改。 |
| 云日志服务（LTS）触发器                  | 默认为异步调用，且不可修改。 |
| 消息通知服务（SMN）触发器                 | 默认为异步调用，且不可修改。 |
| EventGrid触发器                   | 默认为异步调用，且不可修改。 |

FunctionGraph 支持的触发事件

**表5-16**为支持触发FunctionGraph函数的云服务，这些云服务可配置为FunctionGraph函数的事件源，即配置函数的触发器。配置事件源触发器后，只要检测到相应事件，将自动调用FunctionGraph函数。

表 5-16 FunctionGraph 支持的云服务触发事件

| 云服务/功能              | 触发事件                                                                                                                                                                                                                                                 |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 计划事件功能<br>TIMER     | 使用TIMER的计划事件功能定期调用函数代码，可以指定固定频率（分钟、小时、天数）或指定 Cron 表达式定期调用函数（ <a href="#">TIMER 示例事件</a> ）。<br>TIMER定时触发器的使用过程请参考 <a href="#">使用定时触发器</a> 。                                                                                                           |
| API网关服务<br>APIG     | 通过HTTPS或者HTTP调用FunctionGraph函数，使用API Gateway 自定义REST API和终端节点来实现。可以将各个API操作（如GET和PUT）映射到特定的FunctionGraph函数，当向该API终端节点发送HTTPS请求时（ <a href="#">APIG示例事件</a> ），APIG会调用相应的FunctionGraph函数。<br>APIG的触发使用过程请参考： <a href="#">使用APIG专享版触发器</a>               |
| API Connect<br>APIC | 通过HTTPS或者HTTP调用FunctionGraph函数，将各个API操作（如GET和PUT）映射到特定的FunctionGraph函数，当向该API发送HTTPS或者HTTP请求时，APIC 会调用相应的FunctionGraph函数。<br>APIC的触发使用过程请参考 <a href="#">使用APIC触发器</a> 。                                                                              |
| 云审计服务<br>CTS        | 编写FunctionGraph函数，根据CTS云审计服务类型和操作订阅所需要的事件通知，当CTS云审计服务获取已订阅的操作记录后，通过CTS触发器将采集到的操作记录作为参数传递（ <a href="#">CTS示例事件</a> ）来调用FunctionGraph函数。经由函数对日志中的关键信息进行分析和处理，对系统、网络等业务模块进行自动修复，或通过短信、邮件等形式产生告警，通知业务人员进行处理。<br>CTS的触发使用过程请参考 <a href="#">使用CTS触发器</a> 。 |

| 云服务/功能                 | 触发事件                                                                                                                                                                                                                                  |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 文档数据库服务 DDS            | 使用DDS触发器，每次更新数据库中的表时，都可以触发Functiongraph函数以执行额外的工作（ <a href="#">DDS示例事件</a> ）。<br>DDS的触发使用过程请参考 <a href="#">使用DDS触发器</a> 。                                                                                                             |
| 数据接入服务 DIS             | 将FunctionGraph函数配置为自动轮询流并处理任何新记录，例如网站点击流、财务交易记录、社交媒体源、IT日志和数据位置跟踪事件等（ <a href="#">DIS示例事件</a> ）。FunctionGraph会定期轮询DIS数据流中的新记录。<br>DIS的触发使用过程请参考 <a href="#">使用DIS触发器</a> 。                                                            |
| 分布式消息服务 Kafka版         | 当向Kafka实例的Topic生产消息时，FunctionGraph会消费消息，触发函数以执行额外的工作（ <a href="#">Kafka示例事件</a> ）。<br>分布式消息Kafka的触发使用过程请参见：<br><ul style="list-style-type: none"> <li>• <a href="#">使用Kafka触发器</a></li> <li>• <a href="#">使用开源Kafka触发器</a></li> </ul> |
| 分布式消息服务 RabbitMQ版      | FunctionGraph可以定期轮询RabbitMQ实例指定Exchange绑定的队列下的新消息，FunctionGraph将轮询得到的消息作为参数传递来调用函数（ <a href="#">分布式消息服务RabbitMQ示例事件</a> ）。<br>分布式消息RabbitMQ的触发使用过程请参见 <a href="#">使用RabbitMQ触发器</a> 。                                                 |
| 云数据库 GeminiDB MongoDB  | 使用GeminiDB MongoDB(MONGODB)触发器，每次更新数据库中的表时，都可以触发FunctionGraph函数以执行额外的工作（ <a href="#">GeminiDB MongoDB示例事件</a> ）。<br>GeminiDB MongoDB触发器使用请参见 <a href="#">使用云数据库GeminiDB MongoDB触发器</a> 。                                              |
| 云数据库 GeminiDB DynamoDB | 使用GeminiDB DynamoDB(DYNAMODB)触发器，实现定时拉取DynamoDB数据库流表中的信息，并触发函数执行（ <a href="#">云数据库GeminiDB DynamoDB示例</a> ）。<br>GeminiDB DynamoDB触发器使用请参见 <a href="#">使用云数据库GeminiDB DynamoDB触发器</a> 。                                                |
| 设备接入服务 IoTDA           | 使用IoTDA触发器，对于设备上报到平台的数据，FunctionGraph可跟踪设备的设备属性、消息上报，状态变更，分析、整理和计量数据流（ <a href="#">IoTDA示例事件</a> ）。<br>IoTDA触发器的使用请参见 <a href="#">使用IoTDA触发器</a> 。                                                                                    |
| 云日志服务 LTS              | 编写FunctionGraph函数来处理云日志服务订阅的日志，当云日志服务采集到订阅的日志后，可以通过将采集到的日志作为参数传递（ <a href="#">LTS示例事件</a> ）来调用FunctionGraph函数，FunctionGraph函数代码可以对其进行自定义处理、分析或将其加载到其他系统。<br>LTS的触发使用过程请参考 <a href="#">使用LTS触发器</a> 。                                  |

| 云服务/功能           | 触发事件                                                                                                                                                                                                                                                                                                                                                                                         |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 消息通知服务 SMN       | 编写FunctionGraph函数来处理SMN的通知，在将消息发布到SMN主题时，服务可以通过将消息负载作为参数传递（ <a href="#">SMN示例事件</a> ）来调用FunctionGraph函数，FunctionGraph函数代码可以处理事件，比如将消息发布到其他SMN主题或将消息发送到其他云服务。<br>SMN消息触发的使用过程请参考 <a href="#">使用SMN触发器</a> 。                                                                                                                                                                                   |
| 对象存储服务 OBS       | 可以编写FunctionGraph函数来处理OBS存储桶事件，例如对象创建事件或对象删除事件（ <a href="#">OBS示例事件</a> ）。当用户将一张照片上传到存储桶时，OBS存储桶调用FunctionGraph函数，实现读取图像和创建照片缩略图。<br>OBS对象操作触发函数的过程请参考： <ul style="list-style-type: none"><li>• <a href="#">使用OBS触发器</a></li><li>• <a href="#">使用EventGrid触发器（OBS应用事件源）</a></li></ul>                                                                                                        |
| 事件网格服务 EventGrid | 可以编写FunctionGraph函数来处理EventGrid的通知，在将消息发布到EG事件源时，服务可以通过将消息负载作为参数传递（ <a href="#">EG示例事件</a> ）来调用FunctionGraph函数。FunctionGraph函数代码可以处理事件，比如将消息发送到其他云服务。<br>EventGrid的触发使用支持以下事件源： <ul style="list-style-type: none"><li>• <a href="#">使用EventGrid触发器（OBS应用事件源）</a></li><li>• <a href="#">使用EventGrid触发器（RocketMQ自定义事件源）</a></li><li>• <a href="#">使用EventGrid触发器（RabbitMQ自定义事件源）</a></li></ul> |

云服务触发事件示例

定时触发器（TIMER）

- TIMER示例事件，具体参数解释参考[表5-17](#)。

```
{
  "version": "v2.0",
  "time": "2023-06-01T08:30:00+08:00",
  "trigger_type": "TIMER",
  "trigger_name": "Timer_001",
  "user_event": "User Event"
}
```

表 5-17 TIMER 示例事件参数说明

| 参数           | 类型     | 描述                      |
|--------------|--------|-------------------------|
| version      | String | 事件协议的版本。                |
| time         | String | 事件发生时间，ISO 8601 标准时间格式。 |
| trigger_type | String | 触发器类型。                  |
| trigger_name | String | 触发器名称。                  |

| 参数         | 类型     | 描述              |
|------------|--------|-----------------|
| user_event | String | 在创建触发器时配置的附加信息。 |

API网关服务（APIG专享版）触发器

- API网关示例事件，具体参数解释参考表5-18。

```
{
  "body": "",
  "requestContext": {
    "apId": "bc1dcffd-aa35-474d-897c-d53425a4c08e",
    "requestId": "11cdcdf33949dc6d722640a13091c77",
    "stage": "RELEASE"
  },
  "queryStringParameters": {
    "responseType": "html"
  },
  "httpMethod": "GET",
  "pathParameters": {},
  "headers": {
    "accept-language": "zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2",
    "accept-encoding": "gzip, deflate, br",
    "x-forwarded-port": "443",
    "x-forwarded-for": "103.218.216.98",
    "accept": "text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8",
    "upgrade-insecure-requests": "1",
    "host": "50eedf92-c9ad-4ac0-827e-d7c11415d4f1.apigw.region.cloud.com",
    "x-forwarded-proto": "https",
    "pragma": "no-cache",
    "cache-control": "no-cache",
    "x-real-ip": "103.218.216.98",
    "user-agent": "Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:57.0) Gecko/20100101 Firefox/57.0"
  },
  "path": "/apig-event-template",
  "isBase64Encoded": true
}
```

表 5-18 APIG 示例事件参数说明

| 参数                    | 类型     | 描述                              |
|-----------------------|--------|---------------------------------|
| body                  | String | 记录实际请求转换为String字符串后的内容。         |
| requestContext        | Map    | 请求来源的API网关的配置信息、请求标识、认证信息、来源信息。 |
| httpMethod            | String | 记录实际请求的HTTP方法。                  |
| queryStringParameters | Map    | 记录在API网关中配置过的Query参数以及实际取值。     |
| pathParameters        | Map    | 记录在API网关中配置过的Path参数以及实际取值。      |
| headers               | Map    | 记录实际请求的完整Header内容。              |

| 参数              | 类型      | 描述                |
|-----------------|---------|-------------------|
| path            | String  | 记录实际请求的完整的Path信息。 |
| isBase64Encoded | Boolean | 默认为true。          |

约束与限制：

- 通过APIG服务调用函数服务时，isBase64Encoded的值默认为true，表示APIG传递给FunctionGraph的请求体body已经进行Base64编码，需要先对body内容Base64解码后再处理。
- 函数必须按以下结构返回字符串。

```
{  "isBase64Encoded": true|false,  "statusCode": httpStatusCode,  "headers": {"headerName": "headerValue", ...},  "body": "..."}

```

云审计服务（CTS）触发器

- 云审计服务CTS示例事件，具体参数解释参考[表5-19](#)。

```
{  "cts": {    "time": 1529974447000,    "user": {      "name": "userName",      "id": "5b726c4fbfd84821ba866bafaaf56aax",      "domain": {        "name": "domainName",        "id": "b2b3853af40448fcb9e40dxj89505ba"      }    }  },  "request": {},  "response": {},  "code": 204,  "service_type": "FunctionGraph",  "resource_type": "graph",  "resource_name": "workflow-2be1",  "resource_id": "urn:fgs:region:projectId:graph:workflow-2be1",  "trace_name": "deleteGraph",  "trace_type": "ConsoleAction",  "record_time": 1529974447000,  "trace_id": "69be64a7-0233-11e8-82e4-e5d37911193e",  "trace_status": "normal"  }
}

```

表 5-19 CTS 示例事件参数说明

| 参数      | 类型   | 描述                |
|---------|------|-------------------|
| time    | Long | 本次请求的时间，13位时间戳格式。 |
| user    | Map  | 本次请求的发起用户信息。      |
| request | Map  | 事件请求内容。           |

| 参数            | 类型     | 描述                              |
|---------------|--------|---------------------------------|
| response      | Map    | 事件响应内容。                         |
| code          | Int    | 事件响应码，例如200、400。                |
| service_type  | String | 发送方的简写，比如vpc，ecs等。              |
| resource_type | String | 发送方资源类型，比如vm，vpn等。              |
| resource_name | String | 资源名称，例如ECS服务中某个虚拟机的名称。          |
| trace_name    | String | 事件名称，例如：startServer, shutDown等。 |
| trace_type    | String | 事件发生源头类型，例如ApiCall。             |
| record_time   | Long   | CTS服务接收到这条trace的时间，13位时间戳格式。    |
| trace_id      | String | 事件的唯一标识符。                       |
| trace_status  | String | 事件的状态。                          |

#### 文档数据库服务（DDS）触发器

- 文档数据库服务DDS示例事件，具体参数解释参考[表5-20](#)。

```
{
  "records": [
    {
      "event_source": "dds",
      "event_name": "insert",
      "region": "region",
      "event_version": "1.0",
      "dds": {
        "size_bytes": "100",
        "token": "{ \"_data\": \"825D8C2F4D0000001529295A100474039A3412A64BA89041DC952357FB4446645F696400645D8C2F8E5BECCB6CF5370D6A0004\" }",
        "full_document": "{ \"_id\": { \"$oid\": \"5d8c2f8e5beccb6cf5370d6a\" }, \"name\": \"dds\", \"age\": { \"$numberDouble\": \"52.0\" } }",
        "ns": "{ \"db\": \"functiongraph\", \"coll\": \"person\" }",
        "event_source_id": "e6065860-f7b8-4cca-80bd-24ef2a3bb748"
      }
    }
  ]
}
```

表 5-20 DDS 示例事件参数说明

| 参数              | 类型     | 描述            |
|-----------------|--------|---------------|
| region          | String | DDS实例所在区域。    |
| event_version   | String | 事件协议的版本。      |
| event_source    | String | 事件来源。         |
| event_name      | String | 事件名称。         |
| size_bytes      | Int    | 消息的字节数。       |
| token           | String | Base64编码后的数据。 |
| full_document   | String | 完整的文件信息。      |
| ns              | String | 列名。           |
| event_source_id | String | 事件源唯一标识符。     |

数据接入服务（DIS）触发器

- 数据接入服务DIS示例事件，具体参数解释参考[表5-21](#)。

```
{
  "ShardID": "shardId-0000000000",
  "Message": {
    "next_partition_cursor":
"eyJnZXRJdGVyYXRvclBhcmFtljp7InN0cmVhbS1uYW1lIjoZGlzLXN3dGVzdCIsInBhcnRpdGlubi1pZCI6InNoYXJkSWQtMDAwMDAwMDAwMCIsImN1cnNvci10eXBlljoiVFJT9IT1JJWk90Iiwic3RhcnRpbmctc2VxdWVudWY2UtnbVtYmVyljoiNCJ9LCJnZW5lcmF0ZVRpbWVzdGFtcCI6MTUwOTYwNjM5MjE5MX0",
    "records": [
      {
        "partition_key": "shardId_0000000000",
        "data": "d2VsY29tZQ==",
        "sequence_number": "0"
      },
      {
        "partition_key": "shardId_0000000000",
        "data": "dXNpbmc=",
        "sequence_number": "1"
      },
      {
        "partition_key": "shardId_0000000000",
        "data": "RnVuY3Rpb25TdGFnZQ==",
        "sequence_number": "2"
      },
      {
        "partition_key": "shardId_0000000000",
        "data": "c2VydmljZQ==",
        "sequence_number": "3"
      }
    ],
    "millis_behind_latest": ""
  },
  "Tag": "latest",
  "StreamName": "dis-swtest"
}
```

表 5-21 DIS 示例事件参数说明

| 参数                    | 类型     | 描述                     |
|-----------------------|--------|------------------------|
| ShardID               | String | 数据下载分区的ID。             |
| next_partition_cursor | String | 下一个分区的游标。              |
| Records               | Map    | 存储在DIS通道中的数据单元。        |
| partition_key         | String | 分区键。                   |
| data                  | String | 数据块，由数据生产者添加到数据通道。     |
| sequence_number       | Int    | 每个记录的唯一标识符，由DIS服务自动分配。 |
| Tag                   | String | 通道的标签。                 |
| StreamName            | String | 通道名称。                  |

分布式消息服务 Kafka（KAFKA）触发器/开源Kafka（OPENSOURCEKAFKA）触发器

- Kafka示例事件，具体参数解释参考[表5-22](#)。

```
{
  "event_version": "v1.0",
  "event_time": 1576737962,
  "trigger_type": "KAFKA",
  "region": "region",
  "instance_id": "81335d56-b9fe-4679-ba95-7030949cc76b",
  "records": [
    {
      "messages": [
        "kafka message1",
        "kafka message2",
        "kafka message3",
        "kafka message4",
        "kafka message5"
      ],
      "topic_id": "topic-test"
    }
  ]
}
```

表 5-22 Kafka 示例事件参数说明

| 参数            | 类型     | 描述           |
|---------------|--------|--------------|
| event_version | String | 事件协议的版本。     |
| event_time    | String | 事件发生时间。      |
| trigger_type  | String | 事件类型。        |
| region        | String | Kafka实例所在地域。 |

| 参数          | 类型     | 描述                |
|-------------|--------|-------------------|
| instance_id | String | 创建的Kafka实例的唯一标识符。 |
| messages    | String | 消息内容。             |
| topic_id    | String | 消息的唯一标识符。         |

分布式消息服务 RabbitMQ版（RABBITMQ）触发器

- 分布式消息服务RabbitMQ示例事件，具体参数解释参考[表5-23](#)。

```
{
  "event_version": "v1.0",
  "event_time": 1576737962,
  "trigger_type": "RABBITMQ",
  "region": "region",
  "records": [
    {
      "messages": [
        "rabbitmq message1",
        "rabbitmq message2",
        "rabbitmq message3",
        "rabbitmq message4",
        "rabbitmq message5"
      ],
      "instance_id": "81335d56-b9fe-4679-ba95-7030949cc76b",
      "exchange": "exchange-test"
    }
  ]
}
```

表 5-23 分布式消息服务 RabbitMQ 参数说明

| 参数            | 类型     | 描述                   |
|---------------|--------|----------------------|
| event_version | String | 事件协议的版本。             |
| region        | String | RabbitMQ实例所在的地域。     |
| instance_id   | String | 创建的RabbitMQ实例的唯一标识符。 |

云数据库GeminiDB MongoDB触发器

- 云数据库 GeminiDB MongoDB示例事件，具体参数解释参考[表5-24](#)。

```
{
  "records": [
    {
      "event_name": "\"insert\"",
      "event_version": "1.0",
      "event_source": "gauss_mongo",
      "region": "cn-north-xx",
      "gauss_mongo": {
        "full_document": "{\"_id\": {\"$oid\": \"5f61de944778db5fcded3f87\"}, \"zhangsan\": \"zhangsan\"}",
        "ns": "{\"db\": \"zhangsan\", \"coll\": \"zhangsan\"}",
        "size_bytes": "100",
        "token": "{\"_data\": \"825F61DE94000000129295A1004A2D9AE61206C43A5AF47CAF7C5C00C5946645F696400645F61DE\"}"
      }
    }
  ]
}
```

```
944778DB5FCDED3F870004\}"
    },
    "event_source_id": "51153d19-2b7d-402c-9a79-757163258a36"
  }
],
"vernier": "{\\_data\\":
\\\"825F61DE940000000129295A1004A2D9AE61206C43A5AF47CAF7C5C00C5946645F696400645F61DE
944778DB5FCDED3F870004\\\"}"
}
```

表 5-24 GeminiDB MongoDB 示例事件参数说明

| 参数              | 类型     | 描述               |
|-----------------|--------|------------------|
| region          | String | GeminiDB实例所在的地域。 |
| event_source    | String | 事件的来源。           |
| event_version   | String | 事件协议的版本。         |
| full_document   | String | 完整的文件信息。         |
| size_bytes      | Int    | 消息的字节数。          |
| token           | String | Base64编码后的数据。    |
| event_source_id | String | 事件源唯一标识符。        |
| vernier         | String | 游标。              |

云数据库GeminiDB DynamoDB触发器

- 云数据库 GeminiDB DynamoDB示例事件，具体参数解释参考[表5-25](#)。

```
{
  "event_time": 1747905781,
  "trigger_type": "DYNAMODB",
  "instance_ip": "10.0.0.205",
  "table_name": "test_052201",
  "records": [
    {
      "region": "ddbhws",
      "event_id": "000001747880625259000",
      "event_name": "INSERT",
      "sequence_number": "000001747880625259000",
      "size_bytes": 170,
      "stream_view_type": "NEW_AND_OLD_IMAGES",
      "approximate_arrival_timestamp": "2025-05-22T02:23:45.259Z",
      "keys": {
        "Title": {
          "S": "test"
        },
        "Year": {
          "N": "2025"
        }
      },
      "old_image": {
      },
      "new_image": {
        "Plot": {
          "S": "plot"
        },
        "Rating": {
```

```
        "N": "1"
      },
      "Title": {
        "S": "test"
      },
      "Year": {
        "N": "2025"
      }
    }
  },
  {
    "region": "ddbhws",
    "event_id": "000001747881021099000",
    "event_name": "MODIFY",
    "sequence_number": "000001747881021099000",
    "size_bytes": 323,
    "stream_view_type": "NEW_AND_OLD_IMAGES",
    "approximate_arrival_timestamp": "2025-05-22T02:30:21.099Z",
    "keys": {
      "Title": {
        "S": "test"
      },
      "Year": {
        "N": "2025"
      }
    },
    "old_image": {
      "Plot": {
        "S": "plot"
      },
      "Rating": {
        "N": "1"
      },
      "Title": {
        "S": "test"
      },
      "Year": {
        "N": "2025"
      }
    },
    "new_image": {
      "Plot": {
        "S": "plot"
      },
      "Rating": {
        "N": "2"
      },
      "Title": {
        "S": "test"
      },
      "Year": {
        "N": "2025"
      }
    }
  }
]
```

表 5-25 GeminiDB DynamoDB 示例事件参数说明

| 参数          | 类型     | 描述             |
|-------------|--------|----------------|
| EventTime   | Int    | 触发器拉取完消息时的时间。  |
| TriggerType | String | 触发器类型。         |
| InstanceIp  | String | DynamoDB的实例IP。 |

| 参数                          | 类型     | 描述                                                                                                                                                      |
|-----------------------------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| TableName                   | String | DynamoDB的表名。                                                                                                                                            |
| Region                      | String | 局点信息。                                                                                                                                                   |
| EventId                     | String | 事件ID。                                                                                                                                                   |
| EventName                   | String | 事件类名,枚举示例表示分别为插入，更改，删除。                                                                                                                                 |
| SequenceNumber              | String | 事件序列号。                                                                                                                                                  |
| SizeBytes                   | Int    | 事件消息大小。                                                                                                                                                 |
| StreamViewType              | String | 记录方式。<br>枚举值： <ul style="list-style-type: none"><li>NEW_IMAGE：新镜像</li><li>OLD_IMAGE：旧镜像</li><li>NEW_AND_OLD_IMAGES：新旧镜像</li><li>KEYS_ONLY：仅主键</li></ul> |
| ApproximateArrivalTimestamp | String | 事件入库时间。                                                                                                                                                 |
| Keys                        | Map    | 主键。                                                                                                                                                     |
| OldImage                    | Map    | 旧镜像。                                                                                                                                                    |
| NewImage                    | Map    | 新镜像。                                                                                                                                                    |

- 设备接入服务IoTDA示例事件，具体参数解释参考[表5-26](#)。

```
{
  "resource": "device",
  "event": "create",
  "event_time": "20240919T011335Z",
  "event_time_ms": "2024-09-19T01:13:35.854Z",
  "request_id": "75127474-1a26-4578-8847-3128d6101954",
  "notify_data": {
    "body": {
      "app_id": "3d40caf3ddfc4e83815b54b50f13aad7",
      "app_name": "DefaultApp_6439vdv2",
      "device_id": "66eb7a0ffa8d9c36870c6892_ttytytytytyt",
      "node_id": "ttytytytytyt",
      "gateway_id": "66eb7a0ffa8d9c36870c6892_ttytytytytyt",
      "node_type": "GATEWAY",
      "auth_info": {
        "auth_type": "SECRET",
        "secure_access": false,
        "timeout": 0
      },
      "product_id": "66eb7a0ffa8d9c36870c6892",
      "product_name": "test",
      "status": "INACTIVE",
      "create_time": "20240919T011335Z"
    }
  }
}
```

表 5-26 IoTDA 示例事件参数说明

| 参数            | 类型             | 描述                                                      |
|---------------|----------------|---------------------------------------------------------|
| resource      | string         | 数据来源，包括：设备、设置属性、设备消息、设备消息状态、设备状态、产品、设备异步命令状态、运行日志、批量任务。 |
| event         | string         | 触发事件。                                                   |
| event_time    | string         | 字符串格式的事件触发时间。                                           |
| event_time_ms | string         | datetime格式的事件触发时间。                                      |
| request_id    | string         | 请求id。                                                   |
| notify_data   | object 参见表5-27 | 推送消息。                                                   |

表 5-27 NotifyData

| 参数   | 类型            | 描述      |
|------|---------------|---------|
| body | object参见表5-28 | 推送消息内容。 |

表 5-28 NotifyDataBody

| 参数        | 类型     | 描述                                                                                                      |
|-----------|--------|---------------------------------------------------------------------------------------------------------|
| app_id    | string | 资源空间ID。                                                                                                 |
| app_name  | string | 资源空间名称。                                                                                                 |
| device_id | string | 设备ID，用于唯一标识一个设备。在注册设备时直接指定，或者由物联网平台分配获得。由物联网平台分配时，生成规则为"product_id" + "_" + "node_id" 拼接而成。<br>最大长度：256 |
| node_id   | string | 设备标识码，通常使用IMEI、MAC地址或Serial No作为nodeId。<br>最大长度：64                                                      |

| 参数           | 类型            | 描述                                                                                                                                                                  |
|--------------|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| gateway_id   | string        | 网关ID，用于标识设备所属的父设备，即父设备的设备ID。当设备是直连设备时，gateway_id与设备的device_id一致。当设备是非直连设备时，gateway_id为设备所关联的父设备的device_id。                                                          |
| node_type    | string        | 设备节点类型。                                                                                                                                                             |
| product_id   | string        | 设备关联的产品ID，用于唯一标识一个产品模型。                                                                                                                                             |
| product_name | string        | 设备关联的产品名称。                                                                                                                                                          |
| status       | string        | 设备的状态。 <ul style="list-style-type: none"><li>• ONLINE：设备在线</li><li>• OFFLINE：设备离线</li><li>• ABNORMAL：设备异常</li><li>• INACTIVE：设备未激活</li><li>• FREEZED：设备冻结</li></ul> |
| create_time  | string        | 在物联网平台注册设备的时间。格式：yyyyMMdd'T'HHmmss'Z'，如20151212T121212Z。                                                                                                            |
| auth_info    | Object参见表5-29 | 设备的接入认证信息。                                                                                                                                                          |

表 5-29 AuthInfo

| 参数            | 类型      | 描述                                                                                                                                                   |
|---------------|---------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| auth_type     | string  | 鉴权类型。支持密钥认证接入(SECRET)和证书认证接入(CERTIFICATES)两种方式。使用密钥认证接入方式(SECRET)填写secret字段，使用证书认证接入方式(CERTIFICATES)填写fingerprint字段，不填写auth_type默认为密钥认证接入方式(SECRET)。 |
| secure_access | Boolean | 指设备是否通过安全协议方式接入，默认值为true。 <ul style="list-style-type: none"><li>• true：通过安全协议方式接入。</li><li>• false：通过非安全协议方式接入。</li></ul>                            |

| 参数      | 类型      | 描述                                                                                                                                                                                  |
|---------|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| timeout | Integer | <p>设备验证码的有效时间，单位：秒，默认值：0。若设备在有效时间内未接入物联网平台并激活，则平台会删除该设备的注册信息。若设置为“0”，则表示设备验证码不会失效（建议填写为“0”）。注意：只有注册设备接口或者修改设备接口修改timeout时返回该参数。</p> <p>最小值：0</p> <p>最大值：2147483647</p> <p>缺省值：0</p> |

具体的设备相关消息请以IoTDA官网为准，例如设备添加：[设备添加通知](#)。

## 云日志服务（LTS）触发器

- 云日志服务LTS示例事件，具体参数解释参考表5-30。

```
{
  "lts": {
    "data":
      "eyJsb2dzIjoiW3t3clm1lc3NhZ2VcljpcjIwMTgtMDgtMDgvMDg6MDg6MDggW1dSTl0gW3Rlc3QuZ282MDEhZGhpcyBpcyBhIHRlc3QgbWVzc2FnZS5clixlcnRpbWVcljoxNTMwMDA5NjUzMDU5LFwiaG9zdF9uYW1lXCi6XCIyY3MtdGVzdFwiLFwiaXBcljpcjE5Mi4xNjguMS4xXClsXCJwYXRoXCI6XCJ2YXlvgG9nL3Rlc3QubG9nXClsXCjsb2dfdWlkXCI6XCI2NjNkNjkzMC03OTJkLTExZTgtOGIwOC0yODZlZDQ4OGNlNzBclixlcmxpbmVfbm9cljoxfV0iLCJvd25lcil6IjYyODBIMTcwYmQ5MzRmNjBhNGQ4NTFfZjZjYjYTA1MTI5IiwibG9nX2dyb3VwX2IkljoiOTdhOWQyODQ0NDQ0OC0xMWU4LTlmYTQtMjg2ZWQ0ODhjZTcwliwibG9nX3RvcGljX2kljoiMWE5Njc1YTctNzg0ZC0xMWU4LTlmNzAtMjg2ZWQ0ODhjZTcwIn0="
  }
}
```

### 表 5-30 LTS 示例事件参数说明

| 参数   | 类型    | 描述            |
|------|-------|---------------|
| data | Sting | Base64编码后的数据。 |

## 消息通知服务 (SMN) 触发器

- 消息通知服务SMN示例事件，具体参数解释参考表5-31。

```
{
  "record": [
    {
      "event_version": "1.0",
      "smn": {
        "topic_urn": "topicUrn",
        "timestamp": "2018-01-09T07:11:40Z",
        "message_attributes": null,
        "message": "this is smn message content",
        "type": "notification",
        "message_id": "a51671f77d4a479cacb09e2cd591a983",
        "subject": "this is smn message subject"
      },
      "event_subscription_urn": "functionUrn",
      "event_source": "smn"
    }
  ],
  "functionname": "test",
  "requestId": "7c307f6a-cf68-4e65-8be0-4c77405a1b2c",
}
```

```
"timestamp": "Tue Jan 09 2018 15:11:40 GMT+0800 (CST)"
}
```

表 5-31 SMN 示例事件参数说明

| 参数                     | 类型     | 描述                                     |
|------------------------|--------|----------------------------------------|
| event_version          | String | 事件协议的版本。                               |
| topic_urn              | String | SMN事件唯一编号，由SMN服务生成。                    |
| type                   | String | 事件的类型。                                 |
| requestId              | String | 请求ID，由FunctionGraph生成。<br>每个请求的ID取值唯一。 |
| message_id             | String | 消息ID，由SMN服务生成。<br>每条消息的ID取值唯一。         |
| message                | String | 消息内容。                                  |
| event_source           | String | 事件源。                                   |
| event_subscription_urn | String | 函数订阅的URN，取值唯一，可在函数详情页获取。               |
| timestamp              | String | 事件发生的时间。                               |

## 对象存储服务（OBS）触发器

- 对象存储服务OBS示例事件，具体参数解释参考[表5-32](#)。

```
{
  "Records": [
    {
      "eventVersion": "2.0",
      "eventTime": "2018-01-09T07:50:50.028Z",
      "requestParameters": {
        "sourceIPAddress": "103.218.216.125"
      },
      "s3": {
        "configurationId": "UK1DGFPYUKUZFHNQ00000160CC0B471D101ED30CE24DF4DB",
        "object": {
          "eTag": "9d377b10ce778c4938b3c7e2c63a229a",
          "sequencer": "000000000160D9E681484D6B4C00000000",
          "key": "job.png",
          "size": 777835
        },
        "bucket": {
          "arn": "arn:aws:s3:::syj-input2",
          "name": "functionstorage-template",
          "ownerIdentity": {
            "PrincipalId": "0ed1b73473f24134a478962e631651eb"
          }
        }
      },
      "Region": "{region}",
      "eventName": "ObjectCreated:Post",
      "userIdentity": {
        "principalId": "9bf43789b1ff4b679040f35cc4f0dc05"
      }
    }
  ]
}
```

```
    }  
  ]  
}
```

表 5-32 OBS 示例事件参数说明

| 参数              | 类型     | 描述                          |
|-----------------|--------|-----------------------------|
| eventVersion    | String | 事件协议的版本                     |
| eventTime       | String | 事件产生的时间<br>使用ISO-8601标准时间格式 |
| sourceIPAddress | String | 请求的源IP地址                    |
| s3              | Map    | OBS事件内容                     |
| object          | Map    | object参数内容                  |
| bucket          | Map    | bucket参数内容                  |
| arn             | String | Bucket的唯一标识符                |
| ownerIdentity   | Map    | 创建Bucket的用户ID               |
| Region          | String | Bucket所在的地域                 |
| eventName       | String | 配置的触发函数的事件                  |
| userIdentity    | Map    | 请求发起者的华为云账号ID               |

EventGrid触发器

- 事件网格服务EventGrid示例事件，具体参数解释参考[表5-33](#)。

RocketMQ自定义事件源：

```
{  
  "datacontenttype": "application/json",  
  "data": {  
    "context": "yyyyy"  
  },  
  "subject": "ROCKETMQ:region:domainId/projectId:ROCKETMQ:eventSourceName",  
  "specversion": "1.0",  
  "id": "016d5bd3-6231-4e9e-86ef-e451a070d598",  
  "source": "eventSourceName",  
  "time": "2023-04-07T11:51:10Z",  
  "type": "ROCKETMQ:CloudTrace:RocketmqCall"  
}
```

RabbitMQ自定义事件源：

```
{  
  "datacontenttype": "application/json",  
  "data": {  
    "context": "yyyyy"  
  },  
  "subject": "RABBITMQ:region:domainId/projectId:RABBITMQ:eventSourceName",  
  "specversion": "1.0",  
  "id": "016d5bd3-6231-4e9e-86ef-e451a070d598",  
  "source": "eventSourceName",  
  "time": "2023-04-07T11:51:10Z",  
}
```

```
"type": "RABBITMQ:CloudTrace:RabbitmqCall"
}
```

### OBS应用事件源:

```
{
  "channel_id":"b65779ed-d9d0-4a6c-b312-c767226964cf",
  "description":"",
  "name":"subscription-xeak",
  "sources":[
    {
      "id":null,
      "name":"HC.OBS.DWR",
      "detail":{
        "bucket":"eventbucket",
        "objectKeyEncode":true
      },
      "filter":{
        "source":[
          {
            "op":"StringIn",
            "values":[
              "HC.OBS.DWR"
            ]
          }
        ],
        "type":[
          {
            "op":"StringIn",
            "values":[
              "OBS:DWR:ObjectCreated:PUT",
              "OBS:DWR:ObjectCreated:POST"
            ]
          }
        ],
        "subject":{
          "and":[
            {
              "op":"StringStartsWith",
              "values":[
                "/ddd"
              ]
            }
          ]
        },
        "data":{
          "obs":{
            "bucket":{
              "name":[
                {
                  "op":"StringIn",
                  "values":[
                    "output-your"
                  ]
                }
              ]
            }
          }
        }
      },
      "provider_type":"OFFICIAL"
    }
  ],
  "targets":[
    {
      "id":null,
      "name":"HC.FunctionGraph",
      "detail":{
        "urn":"urn:fss:cn-north-7:c53626012ba84727b938ca8bf03108ef:function:A-nodejs-lqz:pylog:latest",
        "agency_name":"EG_AGENCY"
      }
    }
  ]
}
```

```
    },
    "dead_letter_queue":null,
    "provider_type":"OFFICIAL",
    "transform":{
      "type":"ORIGINAL",
      "value":""
    }
  }
}
```

表 5-33 EG 示例事件参数说明

| 参数              | 类型     | 描述     |
|-----------------|--------|--------|
| datacontenttype | String | 数据类型   |
| data            | Map    | 数据     |
| subject         | String | 目标值    |
| specversion     | String | 版本     |
| id              | String | 唯一键值   |
| source          | String | 来源名称   |
| time            | String | 发布订阅时间 |
| type            | String | 订阅类型   |

### 5.6.2 使用定时触发器（TIMER）

本节介绍如何通过函数工作流控制台创建定时触发器（TIMER），实现按照设置的频率定期触发函数运行。

常见应用场景示例：

- 备份重要数据，例如每7天执行备份数据的函数。
- 监控服务器状态和资源使用情况，例如每天上午10点执行函数并发送服务器使用报表。

#### 定时触发器触发事件说明

定时触发器使用TIMER的计划事件功能，能够根据设定的固定频率（分钟、小时、天数）或指定的Cron表达式规则，定期调用函数，适用于需要定期执行任务的场景。

“拉美-圣保罗一”区域支持为共享函数配置定时触发器。

定时触发器的示例事件

- 定时触发器会按照以下事件格式触发函数，具体参数解释请参考表5-34。

```
{
  "version": "v2.0",
  "time": "2023-06-01T08:30:00+08:00",
  "trigger_type": "TIMER",
  "trigger_name": "Timer_001",
  "user_event": "User Event"
}
```

表 5-34 TIMER 示例事件参数说明

| 参数           | 类型     | 描述              |
|--------------|--------|-----------------|
| version      | String | 事件函数的版本。        |
| time         | String | 事件发生时间。         |
| trigger_type | String | 触发器类型，此处为定时触发器。 |
| trigger_name | String | 触发器名称。          |
| user_event   | String | 在创建触发器时配置的附加信息。 |

约束与限制

HTTP函数不支持创建定时触发器。

前提条件

已[创建函数](#)。

创建定时触发器

1. 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。

2. 选择待配置的函数，单击函数名称进入函数详情页。  
“拉美-圣保罗一”区域支持为共享函数配置定时触发器，单击“共享函数”页签下的函数名称进入函数详情页即可。共享函数相关说明请参考[基于RAM共享函数资源](#)。

3. 如[图5-22](#)所示，选择“设置 > 触发器”，单击“创建触发器”，弹出“创建触发器”对话框。

图 5-22 创建触发器



4. 配置以下参数。

表 5-35 定时触发器参数说明

| 参数    | 说明                                              | 取值样例          |
|-------|-------------------------------------------------|---------------|
| 触发器类型 | 选择“定时触发器 (TIMER)”。                              | 定时触发器 (TIMER) |
| 定时器名称 | 自定义的定时器名称。支持字母、数字、下划线和中划线，必须以字母开头，且长度不能超过64个字符。 | Timer-fg      |

| 参数   | 说明                                                                                                                                                                                                                                                             | 取值样例        |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 触发规则 | 触发器的触发规则，可选择使用“固定频率”或“Cron表达式”。 <ul style="list-style-type: none"><li>固定频率：固定时间间隔触发函数，该类型下支持配置单位为分、时、天，每种类型仅支持整数配置，其中分钟支持范围(0, 60]，小时支持范围(0, 24]，天支持范围(0, 30]。</li><li>Cron表达式：设置更为复杂的函数执行计划，例如：周一到周五上午08:30:00执行函数等，具体请参见<a href="#">Cron表达式规则</a>。</li></ul> | 固定频率<br>1分钟 |
| 是否开启 | 选择是否开启定时触发器。不开启则不会按照设定频率触发函数。                                                                                                                                                                                                                                  | 开启          |
| 附加信息 | 可选参数。<br>如果用户配置了触发事件，会将该事件填写到TIMER事件源的“user_event”字段，详情请参见 <a href="#">支持的事件源</a> 。                                                                                                                                                                            | -           |

5. 单击“确定”，完成定时触发器的创建。

查看函数运行结果

函数的定时触发器创建后，将按配置的触发规则定期执行函数，可以在函数详情页中查看函数运行日志。

- 步骤1 返回函数工作流控制台，在左侧的导航栏选择“函数 > 函数列表”。
- 步骤2 选择“函数”，单击函数名称进入函数详情页。
- 步骤3 选择“监控 > 日志”，查询[函数运行日志](#)。
- 结束

Cron 表达式规则

函数Cron表达式下支持如下两种配置方式。

- @every格式**  
@every NUnit，其中N表示一个正整数，Unit可以为ns, μs, ms, s, m, h，表示每隔N个Unit时间触发一次函数，如[表5-36](#)所示。

表 5-36 表达式示例

| 表达式          | 说明               |
|--------------|------------------|
| @every 30m   | 每隔30分钟触发一次函数。    |
| @every 1h    | 每隔1小时触发一次函数。     |
| @every 2h30m | 每隔2小时30分钟触发一次函数。 |

- 标准Cron表达式**  
Cron表达式格式要求“秒 分 时 日 月 星期(可选)”，每个字段间以空格隔开，其中各字段说明如表5-37所示。其中特殊字符的含义说明请参考表5-40。

表 5-37 Cron 表达式字段说明

| 字段              | 说明                                                                                                                                                                                        | 取值范围                                    | 允许的特殊字符   |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------|-----------|
| CRON_TZ         | 可选。不设置则默认使用region所在时区。<br>如果您的任务需要按照特定时区运行，可以通过CRON_TZ指定，例如在北京时间每个月一号的04:00触发函数执行，则可以使用<br>CRON_TZ=Asia/Shanghai 0 0 4 1 *<br>*。<br>不同地域的时区表达式存在差异，请以实际情况自行查询所需地区的CRON_TZ表达方式，示例可参考表5-41。 | -                                       | -         |
| 秒               | 必选。                                                                                                                                                                                       | 0-59                                    | , - * /   |
| 分               | 必选。                                                                                                                                                                                       | 0-59                                    | , - * /   |
| 时               | 必选。                                                                                                                                                                                       | 0-23                                    | , - * /   |
| 日(Day of month) | 必选。                                                                                                                                                                                       | 1-31                                    | , - * ? / |
| 月               | 必选。                                                                                                                                                                                       | 1-12或者Jan-Dec（英文不区分大小写）如表5-38所示。        | , - * /   |
| 星期(Day of week) | 可选。                                                                                                                                                                                       | 0-6或者Sun-Sat（0表示星期天，英文不区分大小写），如表5-39所示。 | , - * ? / |

表 5-38 月份字段取值说明

| 月份 | 数字 | 英文简写 |
|----|----|------|
| 1月 | 1  | Jan  |

| 月份  | 数字 | 英文简写 |
|-----|----|------|
| 2月  | 2  | Feb  |
| 3月  | 3  | Mar  |
| 4月  | 4  | Apr  |
| 5月  | 5  | May  |
| 6月  | 6  | Jun  |
| 7月  | 7  | Jul  |
| 8月  | 8  | Aug  |
| 9月  | 9  | Sep  |
| 10月 | 10 | Oct  |
| 11月 | 11 | Nov  |
| 12月 | 12 | Dec  |

表 5-39 星期字段取值说明

| 星期  | 数字 | 英文简写 |
|-----|----|------|
| 星期一 | 1  | Mon  |
| 星期二 | 2  | Tue  |
| 星期三 | 3  | Wed  |
| 星期四 | 4  | Thu  |
| 星期五 | 5  | Fri  |
| 星期六 | 6  | Sat  |
| 星期日 | 0  | Sun  |

Cron表达式字段特殊字符说明如[表5-40](#)所示。

表 5-40 特殊字符说明

| 特殊字符 | 含义          | 说明                 |
|------|-------------|--------------------|
| *    | 表示该字段中的所有值。 | 在“分钟”字段中表示每一分钟都执行。 |

| 特殊字符 | 含义                       | 说明                                                                                                |
|------|--------------------------|---------------------------------------------------------------------------------------------------|
| ,    | 指定多个值（可以不连续）。            | 在“月”字段中指定“Jan, Apr, Jul, Oct”或者“1, 4, 7, 10”，表示1月，4月，7月和10月，在“星期几”字段中指定“Sat, Sun”或者“6, 0”表示周六，周日。 |
| -    | 指定一个范围。                  | 在“分钟”字段中使用0-3，表示从0分到3分。                                                                           |
| ?    | 指定一个或另一个。                | 仅“日”和“星期几”字段可以指定。例如，如果指定了一个特定的日期，但你不关心该日期对应星期几，那么“星期几”字段就可以使用该特殊字符。                               |
| /    | 表示起步和步幅，n/m表示从n开始，每次增加m。 | 在“分钟”字段1/3表示在满足其它字段情况下，从时间1分（例如00:01:00）开始，每隔3分钟触发一次。                                             |

Cron表达式配置示例如表5-41所示。

表 5-41 Cron 表达式配置示例

| 配置实例                          | Cron 表达式（以北京时区为例）                                |
|-------------------------------|--------------------------------------------------|
| 每天12点调度函数                     | CRON_TZ=Asia/Shanghai 0 0 12 * * *               |
| 每天12:30调度函数                   | CRON_TZ=Asia/Shanghai 0 30 12 * * *              |
| 每小时的26分，29分，33分调度函数           | CRON_TZ=Asia/Shanghai 0 26,29,33 * * * *         |
| 周一到周五的每天12:30调度函数             | CRON_TZ=Asia/Shanghai 0 30 12 ? * MON-FRI        |
| 周一到周五的每天12:00 ~ 14:59每5分钟调度函数 | CRON_TZ=Asia/Shanghai 0 0/5 12-14 ? * MON-FRI    |
| 一月到四月每天12:00调度函数              | CRON_TZ=Asia/Shanghai 0 0 12 ? JAN,FEB,MAR,APR * |

为共享函数创建定时触发器

“拉美-圣保罗一”区域支持为共享函数配置定时触发器，关于共享函数的相关介绍请参考[基于RAM共享函数资源](#)。

1. 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
2. 选择共享函数页签，单击函数名称进入函数详情页。
3. 后续创建步骤与普通函数相同，请参考[3](#)开始操作。

相关文档

- 除使用控制台外，函数工作流支持通过API的方式管理函数触发器，详情请参见[函数触发器API](#)。
- FunctionGraph中使用定时触发器的最佳实践：[使用FunctionGraph函数定时开关华为云ECS](#)。

5.6.3 使用 API 网关服务（APIG 专享版）触发器

本节介绍如何通过函数工作流控制台创建APIG触发器，实现使用API调用函数运行。

APIG 专享版触发器触发事件说明

APIG专享版触发器是FunctionGraph与[APIG\(API Gateway\)](#)结合使用的一种方式，支持通过HTTPS或HTTP协议调用FunctionGraph函数。通过APIG的自定义REST API和终端节点，可以将特定的API操作（如GET、POST、PUT等）映射到对应的FunctionGraph函数。当用户向APIG终端节点发送HTTPS请求时，APIG会自动触发相应的FunctionGraph函数，从而实现API请求与函数的无缝对接。

APIG专享版触发器的示例事件

- APIG触发器会按照以下事件格式触发函数，具体参数解释参考[表5-42](#)。

```
{
  "body": "",
  "requestContext": {
    "apiId": "bc1dcffd-aa35-474d-897c-d53425a4c08e",
    "requestId": "11cdcfcf33949dc6d722640a13091c77",
    "stage": "RELEASE"
  },
  "queryStringParameters": {
    "responseType": "html"
  },
  "httpMethod": "GET",
  "pathParameters": {},
  "headers": {
    "accept-language": "zh-CN;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2",
    "accept-encoding": "gzip, deflate, br",
    "x-forwarded-port": "443",
    "x-forwarded-for": "103.218.216.98",
    "accept": "text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8",
    "upgrade-insecure-requests": "1",
    "host": "50eedf92-c9ad-4ac0-827e-d7c11415d4f1.apigw.region.cloud.com",
    "x-forwarded-proto": "https",
    "pragma": "no-cache",
    "cache-control": "no-cache",
    "x-real-ip": "103.218.216.98",
    "user-agent": "Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:57.0) Gecko/20100101 Firefox/57.0"
  },
  "path": "/apig-event-template",
  "isBase64Encoded": true
}
```

表 5-42 APIG 示例事件参数说明

| 参数   | 类型     | 描述                      |
|------|--------|-------------------------|
| body | String | 记录实际请求转换为String字符串后的内容。 |

| 参数                    | 类型      | 描述                                                                             |
|-----------------------|---------|--------------------------------------------------------------------------------|
| requestContext        | Map     | 请求来源的API网关的配置信息、请求标识、认证信息、来源信息。                                                |
| httpMethod            | String  | 记录实际请求的HTTP方法。                                                                 |
| queryStringParameters | Map     | 记录在API网关中配置过的Query参数以及实际取值。                                                    |
| pathParameters        | Map     | 记录在API网关中配置过的Path参数以及实际取值。                                                     |
| headers               | Map     | 记录实际请求的完整Header内容。                                                             |
| path                  | String  | 记录实际请求的完整的Path信息。                                                              |
| isBase64Encoded       | Boolean | 默认为true，true表示APIG传递给FunctionGraph的请求体body已经进行Base64编码，需要先对body内容Base64解码后再处理。 |

## 约束与限制

- 当前除“华北-北京一”外，其余区域均支持使用APIG专享版触发器。具体运行时函数是否支持使用请以控制台为准。
- APIG专享版触发器创建以后，不能停用，只能删除。
- FunctionGraph函数对来自APIG调用的返回结果进行了封装，**APIG触发器要求函数的返回结果中必须包含isBase64Encoded(boolean)、statusCode(int)、headers(Map)和body(String)**，即函数的返回结果中必须按以下示例结构返回字符串才可以正确返回。

```
{
  "isBase64Encoded": true|false,
  "statusCode": httpStatusCode,
  "headers": {"headerName": "headerValue", ...},
  "body": "..."
}
```

isBase64Encoded的值默认为true，表示APIG传递给FunctionGraph的请求体body已经进行Base64编码，需要先对body内容Base64解码后再处理。

- APIG专享版触发器调用的请求正文有效负载大小为4M。

## 前提条件

- 已创建APIG专享版实例，创建过程请参见[创建APIG实例](#)。
  - 如需在公网调用函数，请在创建APIG实例时开启公网并配置所需带宽。
  - 如生产环境中配置函数访问VPC内资源，请在创建APIG实例时选择与函数相同的VPC环境。
- 已创建API分组，此处以“APIGroup\_test”分组为例，创建过程请参见[创建API分组](#)。

创建 APIG 专享版触发器

- 步骤1 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
- 步骤2 单击右上方的“创建函数”，进入“创建函数”页面。
- 步骤3 选择“创建空白函数”，配置以下函数信息，其他参数保持默认即可。
- 函数名称：输入自定义的函数名称，例如：apig\_demo。
  - 企业项目：选择“default”。
  - 委托：选择“未使用任何委托”。
  - 运行时语言：选择“Python 3.12”。
- 步骤4 单击“创建函数”，完成函数创建并进入函数详情页。
- 步骤5 在“代码”页签下，复制以下代码至代码编辑区并单击“部署代码”。

```
# -*- coding:utf-8 -*-
import json
def handler (event, context):
    body = "<html><title>Functiongraph Demo</title><body><p>Hello, FunctionGraph!</p></body></html>"
    print(body)
    return {
        "statusCode":200,
        "body":body,
        "headers": {
            "Content-Type": "text/html",
        },
        "isBase64Encoded": False
    }
```

- 步骤6 如[图5-23](#)所示选择“设置 > 触发器”，单击“创建触发器”，弹出“创建触发器”对话框。

图 5-23 创建触发器



- 步骤7 参考[表5-43](#)配置触发器参数。

表 5-43 APIG 专享版触发器参数说明

| 参数    | 说明                                                                                                  | 取值样例             |
|-------|-----------------------------------------------------------------------------------------------------|------------------|
| 触发器类型 | 选择“API网关服务（APIG专享版）”。                                                                               | API网关服务（APIG专享版） |
| 实例    | 选择APIG实例。若无实例，可单击“创建实例”完成创建。                                                                        | apig-fg          |
| API名称 | 填写专享版APIG触发器的名称。<br>支持汉字，英文，数字，下划线，且只能以英文和汉字开头，3-64字符。<br>关于API更多配置信息的介绍请参考 <a href="#">创建API</a> 。 | API_apig         |

| 参数   | 说明                                                                                                                                                                                                                                                                                                   | 取值样例          |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
| 分组   | 选择API分组。API分组相当于一个API集合，API提供方以API分组为单位，管理分组内的所有API。<br>若未创建分组，可单击“创建分组”完成创建。                                                                                                                                                                                                                        | APIGroup_test |
| 发布环境 | 选择API的发布环境。API可以同时提供给不同的场景调用，如生产、测试或开发，选择“RELEASE”，才能调用。<br>若未创建环境，可单击“创建发布环境”完成创建。                                                                                                                                                                                                                  | RELEASE       |
| 安全认证 | 选择API的安全认证方式。<br>API认证方式有以下三种： <ul style="list-style-type: none"> <li>App：采用Appkey&amp;Appsecret认证，安全级别高，推荐使用，详情请参见<a href="#">APP认证</a>。</li> <li>IAM：IAM认证，只允许IAM用户能访问，安全级别中等，详情请参见<a href="#">IAM认证</a>。</li> <li>None：无认证模式，所有用户均可访问。</li> </ul> <b>示例中“安全认证”选择“None”，生产环境强烈建议开启APP认证或IAM认证。</b> | None          |
| 请求协议 | 选择API的请求协议。<br>API的请求协议分为两种类型： <ul style="list-style-type: none"> <li>HTTP：数据在传输过程中不被加密。</li> <li>HTTPS：数据在传输过程中被加密。传输重要或敏感数据时推荐使用HTTPS。</li> <li>HTTP&amp;HTTPS</li> </ul>                                                                                                                          | HTTPS         |
| 请求路径 | 填写API的请求路径。<br>请求路径是API访问地址的一部分，用于指定资源的唯一标识或层级结构。<br><b>请求路径中的字母区分大小写。以斜杠(/)开头，不超过512个字符。请求路径可以包含请求参数，请求参数使用{}标识，例如/a/{b}，也可以通过配置“+”号做前缀匹配，例如：/a/{b+}。</b>                                                                                                                                           | /ttest        |
| 匹配模式 | 选择API的匹配模式。<br>匹配模式定义了如何匹配API请求的URL，API网关支持两种匹配模式： <ul style="list-style-type: none"> <li>绝对匹配：绝对匹配要求请求路径必须与定义的路径完全一致。</li> <li>前缀匹配：前缀匹配允许请求路径以定义的路径为前缀。</li> </ul>                                                                                                                                 | 前缀匹配          |

| 参数           | 说明                                                                                         | 取值样例 |
|--------------|--------------------------------------------------------------------------------------------|------|
| 请求方法         | 选择API的请求方法。<br>支持的请求方法：GET、POST、DELETE、PUT、PATCH、HEAD、OPTIONS、ANY。<br>其中ANY表示该API支持任意请求方法。 | ANY  |
| 后端超时<br>(毫秒) | 设置API的后端超时时间，单位为毫秒，设置范围为1~60000毫秒。                                                         | 5000 |

步骤8 单击“确定”，完成触发器的创建。

----结束

调用函数

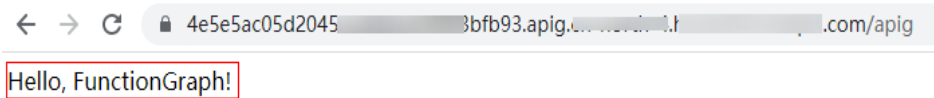
步骤1 如图5-24所示复制APIG触发器的调用URL，打开任意浏览器，在地址栏中输入调用地址并访问。

图 5-24 复制 APIG 调用 URL



步骤2 函数执行完毕，得到返回结果，如图5-25所示。

图 5-25 返回结果



----结束

查看函数运行结果

- 步骤1 返回函数工作流控制台，在左侧的导航栏选择“函数 > 函数列表”。
- 步骤2 单击函数名称，单击进入函数详情页。
- 步骤3 选择“监控 > 日志”，查询函数运行日志。
- 结束

相关文档

- 除使用控制台外，函数工作流支持通过API的方式管理函数触发器，详情请参见函数触发器API。

- 使用APIG触发器常见问题：[函数如何获取APIG触发器中的请求路径或请求参数？](#)
- 在FunctionGraph中使用APIG专享版触发器的最佳实践：
  - [使用FunctionGraph部署AI绘画Stable Diffusion应用](#)
  - [使用FunctionGraph部署DeepSeek-R1蒸馏模型](#)
  - [使用FunctionGraph部署AI绘画ComfyUI/ComfyUI+FLUX应用](#)
  - [使用FunctionGraph部署AI对话ChatGLM3应用](#)

5.6.4 使用 API Connect （ APIC ） 触发器

本节介绍如何通过函数工作流控制台创建APIC触发器，实现使用API调用函数运行。  
关于APIC触发器事件源的具体介绍，请参见[支持的事件源](#)。

约束与限制

APIC触发器当前支持“华东-上海一”、“华东-上海二”、“华南-广州”、“亚太-新加坡”区域使用。具体运行时函数是否支持使用请以控制台为准。

前提条件

- 已创建函数。
- 已创建APIC实例，创建过程请参见[创建APIC实例](#)。
- 已经创建API分组，此处以APIConnect\_test分组为例，创建过程请参见[创建API分组](#)。

创建 APIC 触发器

- 步骤1** 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
- 步骤2** 选择待配置的函数，单击进入函数详情页。
- 步骤3** 选择“设置 > 触发器”，单击“创建触发器”，弹出“创建触发器”对话框。

图 5-26 创建触发器



- 步骤4** 配置以下参数。

表 5-44 APIC 触发器参数说明

| 参数    | 说明                                    | 取值样例                 |
|-------|---------------------------------------|----------------------|
| 触发器类型 | 必选参数。<br>选择“API Connect （ APIC ） ”。   | API Connect （ APIC ） |
| 实例    | 必选参数。<br>选择APIC实例。若无实例，可单击“创建实例”完成创建。 | APIC_fg              |

| 参数       | 说明                                                                                                                                                                                                                                             | 取值样例            |
|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| API名称    | 必选参数。<br>APIC触发器的名称。支持汉字，英文，数字，下划线，且只能以英文和汉字开头，3-64字符。                                                                                                                                                                                         | API_apic        |
| 分组       | 必选参数。<br>选择API分组。API分组相当于一个API集合，API提供方以API分组为单位，管理分组内的所有API。<br>若未创建分组，可单击“创建分组”完成创建。                                                                                                                                                         | APIConnect_test |
| 发布环境     | 必选参数。<br>选择API的发布环境。API可以同时提供给不同的场景调用，如生产、测试或开发，选择“RELEASE”，才能调用。<br>若未创建环境，可单击“创建发布环境”完成创建。                                                                                                                                                   | RELEASE         |
| 安全认证     | 必选参数。<br>API认证方式有以下三种方式： <ul style="list-style-type: none"> <li>App：采用Appkey&amp;Appsecret认证，安全级别高，推荐使用，详情请参见<a href="#">APP认证</a>。</li> <li>IAM：IAM认证，只允许IAM用户能访问，安全级别中等，详情请参见<a href="#">IAM认证</a>。</li> <li>None：无认证模式，所有用户均可访问。</li> </ul> | None            |
| 请求协议     | 必选参数。<br>API的请求协议分为两种类型： <ul style="list-style-type: none"> <li>HTTP：数据在传输过程中不被加密。</li> <li>HTTPS：数据在传输过程中被加密。</li> </ul>                                                                                                                      | HTTPS           |
| 后端超时(毫秒) | 必选参数。<br>设置API的后端超时时间，单位为毫秒，设置范围为1~60000毫秒。                                                                                                                                                                                                    | 5000            |

**步骤5** 单击“确定”，完成触发器的创建。

#### 说明

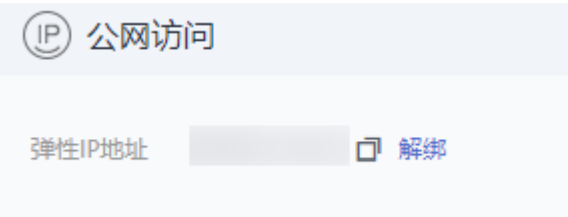
API触发器创建完成后，会在API网关生成名为API\_apic的API，单击API名称，跳转至API网关服务。

----结束

## 调用函数

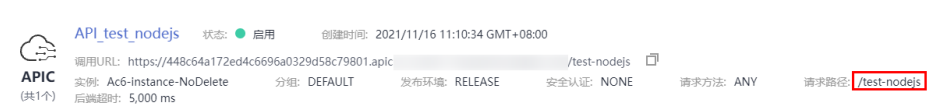
**步骤1** 进入应用与数据集成平台 ROMA Connect，找到所选实例（例如：Ac6-instance-NoDelete），查看公网IP。

图 5-27 公网 IP 地址



步骤2 在浏览器地址栏输入公网IP地址调用。

图 5-28 APIC 触发器调用地址



步骤3 函数执行完毕，得到返回结果。

图 5-29 返回结果



----结束

查看函数运行结果

- 步骤1 返回函数工作流控制台，在左侧的导航栏选择“函数 > 函数列表”。
- 步骤2 单击函数名称，进入函数详情页面。
- 步骤3 在函数详情页面，单击“日志”页签，查询函数运行日志。
- 步骤4 单击操作栏的“查看上下文”，查看日志详细信息。

----结束

相关文档

除使用控制台外，函数工作流支持通过API的方式管理函数触发器，详情请参见函数触发器API。

5.6.5 使用云审计服务（CTS）触发器

本节介绍如何通过函数工作流控制台创建CTS触发器，实现通过增加自定义操作触发函数运行，然后通过云审计服务获取操作记录。

CTS 触发器触发事件说明

**云审计服务**（Cloud Trace Service，以下简称CTS），是华为云安全解决方案中专业的日志审计服务，提供对各种云资源操作记录的收集、存储和查询功能，可用于支撑安全分析、合规审计、资源跟踪、问题回溯和问题定位等常见应用场景。

在FunctionGraph中使用CTS触发器，可以通过编写FunctionGraph函数，依据CTS云审计服务类型和操作订阅所需事件通知。当CTS获取已订阅的操作记录后，将通过CTS触发器将采集到的操作记录作为参数（**CTS示例事件**）传递给FunctionGraph函数。函数对日志中的关键信息进行分析和处理，对系统、网络等业务模块实施自动修复，或通过短信、邮件等形式发送告警，通知业务人员处理。

“拉美-圣保罗一”区域支持为共享函数配置CTS触发器。

云审计服务（CTS）触发器的示例事件

- 云审计服务CTS示例事件，具体参数解释参考**表5-45**。

```
{
  "cts": {
    "time": 1529974447000,
    "user": {
      "name": "userName",
      "id": "5b726c4fbfd84821ba866bafaaf56aax",
      "domain": {
        "name": "domainName",
        "id": "b2b3853af40448fcb9e40dxj89505ba"
      }
    },
    "request": {},
    "response": {},
    "code": 204,
    "service_type": "FunctionGraph",
    "resource_type": "graph",
    "resource_name": "workflow-2be1",
    "resource_id": "urn:fgs:region:projectId:graph:workflow-2be1",
    "trace_name": "deleteGraph",
    "trace_type": "ConsoleAction",
    "record_time": 1529974447000,
    "trace_id": "69be64a7-0233-11e8-82e4-e5d37911193e",
    "trace_status": "normal"
  }
}
```

表 5-45 CTS 示例事件参数说明

| 参数       | 类型   | 示例值    | 描述                |
|----------|------|--------|-------------------|
| time     | Long | 参考示例代码 | 本次请求的时间，13位时间戳格式。 |
| user     | Map  | 参考示例代码 | 本次请求的发起用户信息。      |
| request  | Map  | 参考示例代码 | 事件请求内容。           |
| response | Map  | 参考示例代码 | 事件响应内容。           |
| code     | Int  | 204    | 事件响应码，例如200、400。  |

| 参数            | 类型     | 示例值                                  | 描述                             |
|---------------|--------|--------------------------------------|--------------------------------|
| service_type  | String | FunctionGraph                        | 发送方的简写，比如vpc，ecs等。             |
| resource_type | String | graph                                | 发送方资源类型，比如vm，vpn等。             |
| resource_name | String | workflow-2be1                        | 资源名称，例如ECS服务中某个虚拟机的名称。         |
| trace_name    | String | deleteGraph                          | 事件名称，例如：startServer，shutDown等。 |
| trace_type    | String | ConsoleAction                        | 事件发生源头类型，例如ApiCall。            |
| record_time   | Long   | 参考示例代码                               | CTS服务接收到这条trace的时间，13位时间戳格式。   |
| trace_id      | String | 69be64a7-0233-11e8-82e4-e5d37911193e | 事件的唯一标识符。                      |
| trace_status  | String | normal                               | 事件的状态。                         |

约束与限制

- 当前除“华北-乌兰察布一”、“亚太-雅加达”、“非洲-约翰内斯堡”、“拉美-圣地亚哥”外，其余区域均支持使用CTS触发器。具体运行时函数是否支持使用请以控制台为准。
- CTS触发器最多支持添加10个服务，每个服务10个操作，总共可添加100个操作，服务及操作详情可参见[云审计服务支持的FunctionGraph操作列表](#)。

前提条件

- 已[开通CTS服务并授权](#)。
- 已配置函数的CTS委托权限，操作过程请参见[配置函数的委托权限](#)。

创建 CTS 触发器

以创建一个Python函数为例，介绍如何在函数中配置CTS触发器。

1. 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
2. 单击右上方的“创建函数”，进入“创建函数”页面。
3. 设置以下函数信息。

- 函数名称：输入您自定义的函数名称，例如：HelloWorld。
  - 委托名称：选择含CTS服务权限的委托。
  - 企业项目：选择“default”。
  - 运行时语言：选择“Python 2.7”。
4. 单击“创建函数”，完成函数的创建。
5. 在“代码”页签下，复制如下代码至代码编辑区，部分参数信息请参考表5-46补充，完成后单击“部署代码”。

```
# -*- coding:utf-8 -*-
"""
CTS trigger event:
{
  "cts": {
    "time": "",
    "user": {
      "name": "userName",
      "id": "",
      "domain": {
        "name": "domainName",
        "id": ""
      }
    },
    "request": {},
    "response": {},
    "code": 204,
    "service_type": "FunctionGraph",
    "resource_type": "",
    "resource_name": "",
    "resource_id": {},
    "trace_name": "",
    "trace_type": "ConsoleAction",
    "record_time": "",
    "trace_id": "",
    "trace_status": "normal"
  }
}
"""
def handler(event, context):
    trace_name = event["cts"]["resource_name"]
    timeinfo = event["cts"]["time"]
    print(timeinfo+' '+trace_name)
```

表 5-46 参数说明

| 参数          | 说明                          |
|-------------|-----------------------------|
| time        | 13位时间戳格式，例如“1738805309469”。 |
| record_time | 13位时间戳格式，例如“1738805309469”。 |

6. 选择“设置 > 触发器”，单击“创建触发器”，弹出“创建触发器”对话框。

图 5-30 创建触发器



7. 配置以下参数。

表 5-47 CTS 触发器参数说明

| 参数    | 说明                                                                                                                                                                                                                                                                                                 | 取值样例                                           |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------|
| 触发器类型 | 必选参数。<br>选择“云审计服务（CTS）”。                                                                                                                                                                                                                                                                           | 云审计服务（CTS）                                     |
| 通知名称  | 必选参数。<br>自定义CTS触发器的通知名称。支持汉字、字母、数字和下划线，且长度不能超过64个字节。                                                                                                                                                                                                                                               | CTS_fg                                         |
| 自定义操作 | 必选参数。<br>最多支持添加10个服务，100个操作，配置参数如下： <ul style="list-style-type: none"><li>服务类型：选择“FunctionGraph”。<br/>当服务类型选择全局级云服务时，例如对象存储服务（OBS）、统一身份认证服务（IAM）等，CTS触发器只能在“北京四”区域才能触发。如需了解更多全局级云服务请<a href="#">提交工单</a>咨询。</li><li>资源类型：所选服务下对应的资源类型，如触发器、实例、函数等。</li><li>操作名称：所选资源类型下对应的操作，如创建、删除触发器等。</li></ul> | FunctionGraph;<br>Functions;<br>createFunction |

8. 单击“确定”，完成CTS触发器的创建。

为共享函数配置 CTS 触发器

“拉美-圣保罗一”区域支持为共享函数配置CTS触发器，关于共享函数的相关介绍请参考[基于RAM共享函数资源](#)。

- 1. 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
- 2. 选择“共享函数”页签，单击函数名称进入函数详情页。
- 3. 后续创建步骤与普通函数相同，请参考[6](#)开始操作。

配置 CTS 事件触发函数

- 步骤1 返回函数工作流控制台，在左侧的导航栏选择“函数 > 函数列表”。
- 步骤2 选择待配置的函数，单击进入函数详情页。
- 步骤3 在HelloWorld函数详情页，选择函数版本，单击“测试”，弹出“配置测试事件”对话框。
- 步骤4 填写如[表5-48](#)所示测试信息后，单击“保存”。

表 5-48 测试信息

| 参数     | 说明                                     |
|--------|----------------------------------------|
| 配置测试事件 | 可创建新的测试事件也可编辑已有的测试事件。<br>选择“创建新的测试事件”。 |
| 事件模板   | 选择“云审计服务（CTS）”模板，使用系统内置CTS事件模板。        |
| 事件名称   | 自定义的事件名称，例如：cts-test。                  |
| 测试事件   | 自动加载系统内置CTS事件模板，可以根据实际情况修改。            |

**步骤5** 单击“测试”，可以得到函数运行结果记录。

----结束

相关文档

除使用控制台外，函数工作流支持通过API的方式管理函数触发器，详情请参见[函数触发器API](#)。

5.6.6 使用文档数据库服务（DDS）触发器（即将下线不建议使用）

本节介绍如何通过函数工作流控制台创建DDS触发器，实现每次更新DDS数据库中的表时，都可以触发函数执行。

**DDS触发器即将下线，不建议新建使用。**

约束与限制

- DDS触发器当前仅支持“华东-上海二”和“华南-广州”区域存量用户使用。具体运行时函数是否支持使用请以控制台为准。
- DDS触发器调用的请求正文有效负载大小为6M。

前提条件

- **函数及其配置：**
  - 已创建函数。
  - 已配置函数的DDS和VPC委托权限，操作过程请参见[配置函数的委托权限](#)。
  - 请参见[配置网络](#)开启函数访问VPC内资源的网络配置。
- **文档数据库服务DDS：**
  - 已创建DDS文档数据库实例，创建过程请参见[购买文档数据库实例](#)。
  - 已创建DDS文档数据库，请参见[新建数据库](#)。
  - 开启函数访问VPC内资源后，请参见[设置DDS安全组规则](#)在DDS服务安全组配置对应子网的权限。

创建 DDS 触发器

**步骤1** 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。

- 步骤2** 选择待配置的函数，单击进入函数详情页。
- 步骤3** 选择“设置 > 触发器”，单击“创建触发器”，弹出“创建触发器”对话框。

图 5-31 创建触发器



- 步骤4** 配置以下参数。

表 5-49 DDS 触发器参数说明

| 参数      | 说明                                                            | 取值样例          |
|---------|---------------------------------------------------------------|---------------|
| 触发器类型   | 必选参数。<br>选择“文档数据库服务 (DDS)”。                                   | 文档数据库服务 (DDS) |
| 文档数据库实例 | 必选参数。<br>选择已创建的文档数据库DDS实例。<br>若无实例，可单击“创建实例”完成创建。             | dds_fg        |
| 密码      | 必选参数。<br>输入DDS数据库实例管理员rwuser的密码。输入的密码必须是大写字母、小写字母、数字和特殊字符的组合。 | DDStest@123   |
| 数据库     | 必选参数。<br>输入自定义数据库名称。admin、local、config为保留数据库，不能使用。            | DDS_test      |
| 集合      | 必选参数。<br>输入自定义数据库集合名称。如何创建数据库集合，请参见 <a href="#">新建集合</a> 。    | DDS_set       |
| 批处理大小   | 必选参数。<br>单次函数执行处理的最大数据量，设置在1~10000范围内。                        | 100           |

- 步骤5** 单击“确定”，完成DDS触发器的创建。
- 结束

配置 DDS 事件触发函数

- 步骤1** 返回函数工作流控制台，在左侧的导航栏选择“函数 > 函数列表”。
- 步骤2** 选择待配置的函数，单击函数名称进入函数详情页。
- 步骤3** 选择函数版本，单击“测试”，弹出“配置测试事件”对话框。

**步骤4** 填写如表5-50所示测试信息后, 单击“保存”。

### 表 5-50 测试信息

| 参数     | 说明                                                                                    |
|--------|---------------------------------------------------------------------------------------|
| 配置测试事件 | 可创建新的测试事件也可编辑已有的测试事件。<br>选择默认值：“创建新的测试事件”。                                            |
| 事件模板   | 选择“文档数据库服务（DDS）”模板，使用系统内置dds事件模板。                                                     |
| 事件名称   | 事件名称必须以大写或小写字母开头，支持字母（大写或小写），数字和下划线“_”（或中划线“-”），并以字母或数字结尾，长度为1-25个字符，例如“dds-123test”。 |
| 测试事件   | 自动加载系统内置dds事件模板，本例不做修改。                                                               |

**步骤5** 单击“测试”，可以得到函数运行结果，函数会返回输入DDS数据。

----结束

## 相关文档

除使用控制台外，函数 workflows 支持通过 API 的方式管理函数触发器，详情请参见[函数触发器 API](#)。

### 5.6.7 使用数据接入服务（DIS）触发器

本节介绍如何通过函数 workflow 控制台创建数据接入服务 DIS 触发器，实现使用系统内置的事件模板配置 DIS 事件，触发函数运行。

## DIS 触发器触发事件说明

数据接入服务DIS（Data Ingestion Service）为处理或分析流数据的自定义应用程序构建数据流管道，主要解决云服务外的数据实时传输到云服务内的问题。数据接入服务每小时可从多种数据源中连续捕获、传送和存储数TB数据。

使用DIS触发器可以将FunctionGraph函数配置为自动轮询流并处理新记录，例如网站点击流、财务交易记录、社交媒体源、IT日志和数据位置跟踪事件等。FunctionGraph会定期轮询DIS数据流中的新记录。

## DIS触发器的示例事件

- DIS触发器会按照以下事件格式触发函数，具体参数解释参考表5-51。

```
{
  "ShardID": "shardId-0000000000",
  "Message": {
    "next_partition_cursor":
    "eyJnZXRjdGVyYXRvcjBhcmFtIjp7InN0cmVhbS1uYW1lIjoieGZlZXN3dGVzdCIsInBhcnRpdGlvbi1pZCI6InN0YXJkSWQtMDAwMDAwMDAwMCIsmN1cnNvci10eXB1IjoieVFJTjV9I11JWk9OIiwic3RhcncRpbmctc2VxdWVuY2UtnbVtYmVyljoicj9LCjZnZW5lcmF0ZVRpbWVzdGFtcCI6MTUwOTYwNjM5MjE5MX0",
    "records": [
      {
        "partition_key": "shardId_0000000000",
        "data": "d2VsY29tZQ==",
        "sequence number": "0"
      }
    ]
  }
}
```

```
    },
    {
      "partition_key": "shardId_0000000000",
      "data": "dXNpbmc=",
      "sequence_number": "1"
    },
    {
      "partition_key": "shardId_0000000000",
      "data": "RnVuY3Rpb25TdGFnZQ==",
      "sequence_number": "2"
    },
    {
      "partition_key": "shardId_0000000000",
      "data": "c2VydmljZQ==",
      "sequence_number": "3"
    }
  ],
  "millis_behind_latest": ""
},
"Tag": "latest",
"StreamName": "dis-swtest"
}
```

表 5-51 DIS 示例事件参数说明

| 参数                    | 类型     | 描述                     |
|-----------------------|--------|------------------------|
| ShardID               | String | 数据下载分区的ID。             |
| next_partition_cursor | String | 下一个分区的游标。              |
| Records               | Map    | 存储在DIS通道中的数据单元。        |
| partition_key         | String | 分区键。                   |
| data                  | String | 数据块，由数据生产者添加到数据通道。     |
| sequence_number       | Int    | 每个记录的唯一标识符，由DIS服务自动分配。 |
| Tag                   | String | 通道的标签。                 |
| StreamName            | String | 通道名称。                  |

约束与限制

- DIS触发器当前支持“华北-北京一”、“华北-北京四”、“华东-上海一”、“华东-上海二”、“华南-广州”、“中国-香港”、“亚太-新加坡”、“土耳其-伊斯坦布尔”区域使用。具体运行时函数是否支持使用请以控制台为准。
- DIS触发器调用的请求正文有效负载大小为6M。

前提条件

- 函数及其配置：
  - 已[创建函数](#)。

- 已配置函数的DIS委托权限，操作过程请参见[配置函数的委托权限](#)。
- **数据接入服务DIS：**  
已创建DIS接入通道，此处以dis-function为例，创建过程请参见[创建DIS通道](#)。

创建 DIS 触发器

- 步骤1** 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
- 步骤2** 选择待配置的函数，单击进入函数详情页。
- 步骤3** 选择“设置 > 触发器”，单击“创建触发器”，弹出“创建触发器”对话框。
- 步骤4** 参考[表5-52](#)配置DIS触发器参数。

表 5-52 DIS 触发器参数说明

| 参数    | 说明                                                                                                                                                                                                                                                                                                                                                                        | 取值样例         |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|
| 触发器类型 | 必选参数。<br>选择“数据接入服务 (DIS)”。                                                                                                                                                                                                                                                                                                                                                | 数据接入服务 (DIS) |
| 通道名称  | 必选参数。<br>选择已创建的DIS通道，支持选择普通通道和高级通道。若无接入通道，可单击“申请接入通道”完成创建，详情可参考 <a href="#">开通DIS通道</a> 。                                                                                                                                                                                                                                                                                 | dis-function |
| 起始位置  | 必选参数。<br>选择数据流中开始读取数据的位置，起始位置对应DIS服务中的游标类型，用来选择从DIS通道中读取数据的位置，详情可参考： <a href="#">获取数据游标</a> 。<br>起始位置有以下选择： <ul style="list-style-type: none"><li>● TRIM_HORIZON：从最早被存储至分区的有效记录开始读取。<br/>例如，某租户使用DIS的通道，分别上传了三条数据A1，A2，A3。N天后（设定A1已过期，A2和A3仍在有效期范围内），该租户需要下载此三条数据，并选择了TRIM_HORIZON这种下载方式。那么用户可下载的数据将从A2开始读取。</li><li>● latest：从分区中的最新记录开始读取，此参数可以保证总是读到分区中最新记录。</li></ul> | latest       |

| 参数     | 说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 取值样例 |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
| 最大字节数  | 必选参数。<br>每次触发时通道单次请求的记录的最大字节数，只有当分区中单条记录小于该值，才能被获取。单位可选择KB或MB。 <ul style="list-style-type: none"><li>通道类型为普通通道：单次请求的记录总大小不能超过1MB（不包含partitionKey数据大小），设置范围：0KB~1024KB（0MB~1MB）。</li><li>通道类型为高级通道：单次请求的记录总大小不能超过5MB（不包含partitionKey数据大小），设置范围：0KB~5120KB（0MB~5MB）。</li></ul>                                                                                                                                                                                                                                                                                 | 1 MB |
| 拉取周期   | 必选参数。<br>设置拉取流数据的周期，单位可选择秒或毫秒，设置范围：1ms~60000ms（1s~60s）。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 30 秒 |
| 串行处理数据 | 必选参数。 <ul style="list-style-type: none"><li>如果启用该选项，系统将在完成一次数据处理后，才获取下一次数据；否则只要到达拉取周期，系统就会获取数据进行处理。</li><li>如果关闭该选项，如图5-32所示，可根据业务需要配置并发数（范围：1-80）。<br/>功能说明：控制DIS触发器异步调用函数的并发数，防止单个触发器流量较大导致单租户并发跑满，进而影响其他DIS触发器执行。（当前仅“北京四”区域支持该功能）</li></ul> <p><b>图 5-32 关闭“串行处理数据”</b></p>  <p>串行处理数据 <input type="checkbox"/></p> <p>如果开启该选项，取一次数据处理完之后才会取下一次数据；否则只要拉取周期到了就会取数据进行处理。</p> <p>并发数 <input type="text" value="1-80"/></p> <p>该触发器触发函数执行的最大并发数，即并行处理数据时可同时调用函数的最大次数</p> | 开启   |

**步骤5** 单击“确定”，完成DIS触发器的创建。

----结束

修改 DIS 触发器配置

DIS触发器创建后，支持对部分参数进行修改。

- 步骤1** 返回函数工作流控制台，在左侧的导航栏选择“函数 > 函数列表”。
- 步骤2** 选择待配置的函数，单击进入函数详情页。
- 步骤3** 选择“设置 > 触发器”，单击DIS触发器上的“编辑”，弹出“创建触发器”对话框。

- 步骤4** 支持修改“最大字节数”、“拉取周期”和“串行处理数据”参数，修改完成后单击“确定”。
- 结束

配置 DIS 测试事件触发函数

- 步骤1** 返回函数工作流控制台，在左侧的导航栏选择“函数 > 函数列表”。
- 步骤2** 选择待配置的函数，单击进入函数详情页。
- 步骤3** 在函数详情页，选择函数版本。
- 步骤4** 在“代码”页签下，单击“测试”，弹出“配置测试事件”对话框。
- 步骤5** 填写如表5-53所示测试信息后，单击“保存”。

表 5-53 测试信息

| 参数     | 说明                                                                                      |
|--------|-----------------------------------------------------------------------------------------|
| 配置测试事件 | 可创建新的测试事件也可编辑已有的测试事件。<br>选择默认值：“创建新的测试事件”。                                              |
| 事件模板   | 选择"数据接入服务（DIS）"模板，使用系统内置dis事件模板。                                                        |
| 事件名称   | 事件名称必须以大写或小写字母开头，支持字母（大写或小写），数字和下划线“_”（或中划线“-”），并以字母或数字结尾，长度为1-25个字符。例如输入“dis-123test”。 |
| 测试事件   | 自动加载系统内置dis事件模板，本例不做修改。                                                                 |

- 步骤6** 单击“测试”，可以得到函数运行结果，函数会返回输入DIS数据。
- 结束

相关文档

除使用控制台外，函数工作流支持通过API的方式使用函数触发器，详情请参见[函数触发器API参考](#)。

5.6.8 使用分布式消息服务 Kafka 版（KAFKA）触发器

本节介绍如何通过函数工作流控制台创建Kafka触发器，以实现FunctionGraph对Kafka实例中指定Topic的新消息进行定期轮询。FunctionGraph会将轮询得到的消息作为参数，用于调用相应函数。

关于Kafka触发器的事件源介绍，请参见[支持的事件源](#)。

 说明

分布式消息服务Kafka版与开源Kafka的差异说明，请参见[Kafka与开源Kafka的差异](#)。

约束与限制

- 除“华北-乌兰察布一”、“拉美-圣保罗一”外，其他区域均支持使用Kafka触发器。具体运行时函数是否支持使用请以控制台为准。
- Kafka触发器调用的请求正文有效负载大小为6M。
- 对于处理失败的Kafka数据，Kafka触发器会丢弃过大记录（>6MB）。

前提条件

- 函数及其配置：**
  - 已创建函数。
  - 已配置函数的分布式消息服务DMS委托权限，委托的创建请参见[配置函数的委托权限](#)。
  - 请参见[配置网络](#)开启函数访问VPC内资源的网络配置。
- 分布式消息服务Kafka：**
  - 已创建Kafka实例，创建操作请参见[购买Kafka专享版实例](#)。
  - 在Kafka实例下创建主题，创建操作请参见[Kafka实例创建Topic](#)。
  - 开启函数访问VPC内资源后，请参见[选择和配置Kafka安全组](#)在Kafka服务安全组配置对应子网的权限。

创建 Kafka 触发器

- 步骤1** 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
- 步骤2** 选择待配置的函数，单击进入函数详情页。
- 步骤3** 选择“设置 > 触发器”，单击“创建触发器”，弹出“创建触发器”对话框。

图 5-33 创建触发器



- 步骤4** 配置以下参数。

表 5-54 Kafka 触发器参数说明

| 参数    | 说明                                                                                      | 取值样例                   |
|-------|-----------------------------------------------------------------------------------------|------------------------|
| 触发器类型 | 选择“分布式消息服务 Kafka版 (KAFKA)”。                                                             | 分布式消息服务 Kafka版 (KAFKA) |
| 实例    | 选择已创建的Kafka实例。若无实例，可单击“创建实例”完成创建。                                                       | kafka-fg               |
| 主题    | 选择专享版Kafka实例的Topic。Kafka触发器当前支持选择多个Topic主题，从而避免Topic过多导致创建的触发器数量被限制。若无主题，可单击“创建主题”完成创建。 | topic-fg               |
| 批处理大小 | 输入每次从Topic消费的消息数量，设置范围：1~1000。                                                          | 100                    |

| 参数  | 说明                                  | 取值样例 |
|-----|-------------------------------------|------|
| 用户名 | Kafka实例开启SSL时需要填写。连接Kafka专享版实例的用户名。 | user |
| 密码  | Kafka实例开启SSL时需要填写。连接Kafka专享版实例的密码。  | ***  |

步骤5 单击“确定”，完成kafka触发器的创建。

----结束

配置 Kafka 事件触发函数

- 步骤1 返回函数工作流控制台，在左侧的导航栏选择“函数 > 函数列表”。
- 步骤2 选择待配置的函数，单击进入函数详情页。
- 步骤3 在函数详情页，选择函数版本。
- 步骤4 在“代码”页签下，单击“测试”，弹出“配置测试事件”对话框。
- 步骤5 填写如表5-55所示测试信息后，单击“保存”。

表 5-55 测试信息

| 参数     | 说明                                                                    | 取值样例                         |
|--------|-----------------------------------------------------------------------|------------------------------|
| 配置测试事件 | 可创建新的测试事件也可编辑已有的测试事件。<br>默认值为“创建新的测试事件”。                              | 创建新的测试事件                     |
| 事件模板   | 选择"分布式消息服务 Kafka版（KAFKA）"模板，使用系统内置Kafka事件模板。                          | 分布式消息服务<br>Kafka版<br>（KAFKA） |
| 事件名称   | 事件名称必须以大写或小写字母开头，支持字母（大写或小写），数字和下划线“_”（或中划线“-”），并以字母或数字结尾，长度为1-25个字符。 | kafka-123test                |
| 测试事件   | 自动加载系统内置kafka事件模板，本例不做修改。                                             | -                            |

步骤6 单击“测试”，可以得到函数运行结果，函数会返回输入kafka消息数据。

----结束

相关文档

- 除使用控制台外，函数工作流支持通过API的方式使用函数触发器，详情请参见[函数触发器API参考](#)。
- 使用KAFKA触发器常见问题：[函数和Kafka必须在同一个子网内，才可以配置Kafka触发器吗？](#)

### 5.6.9 使用开源 Kafka（OPENSOURCEKAFKA）触发器

本节介绍如何通过函数工作流控制台创建开源Kafka（OPENSOURCEKAFKA）触发器，以实现FunctionGraph对Kafka实例中指定Topic的新消息进行定期轮询。FunctionGraph会将轮询得到的消息作为参数，用于调用相应函数。

 说明

分布式消息服务Kafka版与开源Kafka的差异说明，请参见[Kafka与开源Kafka的差异](#)。

#### 约束与限制

- 开源Kafka触发器当前支持“华北-北京一”、“华北-北京四”、“华东-上海一”、“华东-上海二”、“华南-广州”、“中国-香港”、“亚太-曼谷”、“亚太-新加坡”、“拉美-圣地亚哥”区域使用。具体运行时函数是否支持使用请以控制台为准。
- Kafka触发器调用的请求正文有效负载大小为6M。
- 对于处理失败的Kafka数据，Kafka触发器会丢弃过大记录（>6MB）。

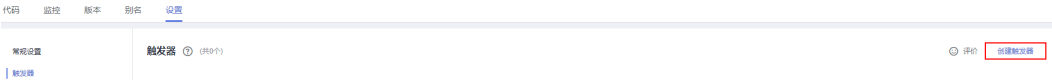
#### 前提条件

- 已创建函数。
- 已开启函数访问VPC内资源的网络配置，开启后需在Kafka服务安全组配置对应子网的权限，请参见[配置网络](#)。函数的网络配置需要和创建Kafka的ECS节点网络配置一样，包括VPC和子网。

#### 创建开源 Kafka 触发器

- 步骤1** 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
- 步骤2** 选择待配置的函数，单击进入函数详情页。
- 步骤3** 选择“设置 > 触发器”，单击“创建触发器”，弹出“创建触发器”对话框。

图 5-34 创建触发器



- 步骤4** 配置以下参数。

表 5-56 开源 Kafka 触发器参数说明

| 参数    | 说明                                      | 取值样例                                   |
|-------|-----------------------------------------|----------------------------------------|
| 触发器类型 | 必选参数。<br>选择“开源Kafka (OPENSOURCEKAFKA)”。 | 开源Kafka (OPENSOURCEKAFKA)              |
| 连接地址  | 必选参数。<br>输入搭建kafka的broker地址列表，以逗号分隔。    | 100.85.125.151:9094,100.95.145.47:9094 |

| 参数    | 说明                                                      | 取值样例    |
|-------|---------------------------------------------------------|---------|
| 主题    | 必选参数。<br>输入自行创建的topic。                                  | topic-1 |
| 批处理大小 | 必选参数。<br>输入每次从Topic消费的消息数量，单次函数执行处理的最大数据量，设置范围：1~10000。 | 100     |

**步骤5** 单击“确定”，完成开源Kafka触发器的创建。

----结束

激活 kafka 触发器

开源Kafka触发器创建完成后默认是停用状态，需要在触发器界面上单击“启动”。

 说明

如果启动失败可以在[工单系统](#)提交工单联系技术支持工程师。

配置 Kafka 事件触发函数

- 步骤1 返回函数工作流控制台，在左侧的导航栏选择“函数 > 函数列表”。
- 步骤2 选择待配置的函数，单击进入函数详情页。
- 步骤3 在函数详情页，选择函数版本。
- 步骤4 在“代码”页签下，单击“测试”，弹出“配置测试事件”对话框。
- 步骤5 填写如表5-57所示测试信息后，单击“保存”。

表 5-57 测试信息

| 参数     | 说明                                                                                     |
|--------|----------------------------------------------------------------------------------------|
| 配置测试事件 | 可创建新的测试事件也可编辑已有的测试事件。<br>选择默认值：“创建新的测试事件”。                                             |
| 事件模板   | 选择"开源Kafka（OPENSOURCEKAFKA）"模板，使用系统内置Kafka事件模板。                                        |
| 事件名称   | 事件名称必须以大写或小写字母开头，支持字母（大写或小写），数字和下划线“_”（或中划线“-”），并以字母或数字结尾，长度为1-25个字符，例如：kafka-123test。 |
| 测试事件   | 自动加载系统内置kafka事件模板，本例不做修改。                                                              |

**步骤6** 单击“测试”，可以得到函数运行结果，函数会返回输入kafka消息数据。

----结束

## 相关文档

除使用控制台外，函数工作流支持通过API的方式使用函数触发器，详情请参见[函数触发器API参考](#)。

### 5.6.10 使用分布式消息服务 RabbitMQ 版（RABBITMQ）触发器

本节介绍如何通过函数工作流控制台创建RabbitMQ触发器（当前只支持fanout路由模式），以实现FunctionGraph在RabbitMQ实例中，指定交换机绑定的队列下的新消息进行定期轮询。FunctionGraph会将轮询得到的消息作为参数，用于调用相应函数。

关于分布式消息服务RabbitMQ触发器的事件源介绍，请参见[支持的事件源](#)。

## 约束与限制

分布式消息服务RabbitMQ触发器当前支持“华北-北京四”、“华东-上海一”、“亚太-曼谷”、“亚太-新加坡”、“拉美-墨西哥城二”、“拉美-圣保罗一”区域使用。具体运行时函数是否支持使用请以控制台为准。

## 前提条件

- **函数及其配置：**
  - 已创建函数。
  - 已配置函数的分布式消息服务DMS委托权限，委托的创建请参见[配置函数的委托权限](#)。
  - 请参见[配置网络](#)开启函数访问VPC内资源的网络配置，开启后需在RabbitMQ服务安全组配置对应子网的权限，请参考RabbitMQ前提条件配置。
- **分布式消息服务RabbitMQ：**
  - 已创建RabbitMQ实例，创建操作请参见[购买RabbitMQ实例](#)。
  - 创建Vhost、Exchange和Queue。
    - i. 创建RabbitMQ Vhost，创建操作请参见[创建RabbitMQ Vhost](#)。
    - ii. 创建RabbitMQ Exchange，创建操作请参见[创建RabbitMQ Exchange](#)。
    - iii. 创建RabbitMQ Queue，创建操作请参见[创建RabbitMQ Queue](#)。
    - iv. 绑定RabbitMQ Exchange和RabbitMQ Queue，绑定操作请参见[绑定RabbitMQ Exchange](#)、[绑定RabbitMQ Queue](#)。

#### 说明

Vhost是一个相对独立的RabbitMQ服务，用于管理Exchange、Queue。一个RabbitMQ实例下可以有多个Vhost，一个Vhost里可以有若干个Exchange和Queue，具体详情请参见[RabbitMQ业务使用流程](#)。

- 确认实例安全组规则是否配置正确。
  - i. 在RabbitMQ实例详情页面的“基本信息 > 网络”，单击安全组名称，跳转到安全组页面。
  - ii. 选择“入方向规则”，查看安全组入方向规则。
    - 1) 实例未开启SSL开关
      - 如果是VPC内访问，实例安全组入方向规则，需要允许端口5672的访问。
      - 如果是公网访问，需要允许端口15672的访问。

- 2) 实例已开启SSL开关
- 如果是VPC内访问，实例安全组入方向规则，需要允许端口5671的访问。

▪ 如果是公网访问，需要运行端口15671的访问。

创建 RabbitMQ 触发器

- 步骤1 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
- 步骤2 选择待配置的函数，单击进入函数详情页。
- 步骤3 选择“设置 > 触发器”，单击“创建触发器”，弹出“创建触发器”对话框。

图 5-35 创建触发器



- 步骤4 配置以下参数。

表 5-58 RabbitMQ 触发器参数说明

| 参数    | 说明                                                               | 取值样例                         |
|-------|------------------------------------------------------------------|------------------------------|
| 触发器类型 | 必选参数。<br>选择“分布式消息服务 RabbitMQ版 (RABBITMQ)”。                       | 分布式消息服务 RabbitMQ版 (RABBITMQ) |
| 实例    | 必选参数。<br>选择已创建的RabbitMQ实例。若无实例，可单击“创建实例”完成创建。                    | rabbitmq-fg                  |
| 密码    | 必选参数。<br>填写创建的RabbitMQ实例的密码。                                     | testrabbitmq                 |
| 交换机名称 | 必选参数。<br>填写已创建的交换机名称，详情请参见 <a href="#">创建 RabbitMQExchange</a> 。 | rabbitmqEX                   |
| 虚拟机名称 | 可选参数。<br>填写已创建的vhost，详情请参见 <a href="#">创建 RabbitMQVhost</a> 。    | test                         |
| 批处理大小 | 必选参数。<br>输入每次从Topic消费的消息数量，设置范围：1~1000。                          | 100                          |

- 步骤5 单击“确定”，完成RabbitMQ触发器的创建。
- 结束

配置 RabbitMQ 事件触发函数

- 步骤1 返回函数工作流控制台，在左侧的导航栏选择“函数 > 函数列表”。
- 步骤2 选择待配置的函数，单击函数名称进入函数详情页。
- 步骤3 在函数详情页，选择函数版本。
- 步骤4 在“代码”页签下，单击“测试”，弹出“配置测试事件”对话框。
- 步骤5 填写如表5-59所示测试信息后，单击“保存”。

表 5-59 测试信息

| 参数     | 说明                                                                                         |
|--------|--------------------------------------------------------------------------------------------|
| 配置测试事件 | 可创建新的测试事件也可编辑已有的测试事件。<br>选择默认值：“创建新的测试事件”。                                                 |
| 事件模板   | 选择“分布式消息服务 RabbitMQ版 (RABBITMQ)”模板，使用系统内置RabbitMQ事件模板。                                     |
| 事件名称   | 事件名称必须以大写或小写字母开头，支持字母（大写或小写），数字和下划线“_”（或中划线“-”），并以字母或数字结尾，长度为1-25个字符，例如“rabbitmq-123test”。 |
| 测试事件   | 自动加载系统内置RabbitMQ事件模板，本例不做修改。                                                               |

- 步骤6 单击“测试”，可以得到函数运行结果，函数会返回输入RabbitMQ消息数据。
- 结束

相关文档

除使用控制台外，函数工作流支持通过API的方式使用函数触发器，详情请参见[函数触发器API参考](#)。

5.6.11 使用云数据库 GeminiDB MongoDB 触发器

本节介绍如何通过函数工作流控制台创建GeminiDB MongoDB触发器，实现每次更新数据库中的表时，都可以触发函数执行。

关于云数据库GeminiDB Mongo触发器事件源的具体介绍，请参见[支持的事件源](#)。

约束与限制

- GeminiDB Mongo触发器当前支持“华北-北京四”、“华东-上海一”、“华东-上海二”、“华南-广州”、“亚太-新加坡”。具体运行时函数是否支持使用请以控制台为准。
- GeminiDB Mongo触发器调用的请求正文有效负载大小为6M。

前提条件

- 函数及其配置：

- 已创建函数。
- 已配置函数的GeminiDB委托权限，委托的创建请参见[配置函数的委托权限](#)。
- 请参见[配置网络](#)开启函数访问VPC内资源的网络配置。
- **云数据库GeminiDB Mongo：**
  - 已创建GeminiDB Mongo云数据库实例，创建过程请参见[云数据库 GeminiDB 实例](#)。
  - 开启函数访问VPC内资源后，请参见[选择和配置GeminiDB Mongo安全组](#)在 GeminiDB Mongo服务安全组配置对应子网的权限。

创建 GeminiDB 触发器

- 步骤1** 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
- 步骤2** 选择待配置的函数，单击进入函数详情页。
- 步骤3** 选择“设置 > 触发器”，单击“创建触发器”，弹出“创建触发器”对话框。

图 5-36 创建触发器



- 步骤4** 配置以下参数。

表 5-60 GeminiDB 触发器参数说明

| 参数               | 说明                                                                  | 取值样例                            |
|------------------|---------------------------------------------------------------------|---------------------------------|
| 触发器类型            | 必选参数。<br>选择“云数据库 GeminiDB MongoDB(DYNAMODB)”。                       | 云数据库 GeminiDB MongoDB(DYNAMODB) |
| GeminiDB Mongo实例 | 必选参数。<br>选择已创建的GeminiDB Mongo实例。若无实例，可单击“创建GeminiDB Mongo实例”完成创建。   | geminidb-fg                     |
| 密码               | 必选参数。<br>输入GeminiDB Mongo数据库实例管理员rwuser的密码。                         | GeminiDB@123                    |
| 数据库              | 必选参数。<br>输入GeminiDB Mongo实例数据库名称。<br>admin、local、config为保留数据库，不能使用。 | GeminiDB-test                   |
| 集合               | 必选参数。<br>输入数据库集合名称。                                                 | GeminiDB-set                    |
| 批处理大小            | 必选参数。<br>输入每批从数据库读取的记录的数量，设置范围：1~10000。                             | 100                             |

**步骤5** 单击“确定”，完成GeminiDB触发器的创建。

----结束

配置 GeminiDB 事件触发函数

- 步骤1 返回函数工作流控制台，在左侧的导航栏选择“函数 > 函数列表”。
- 步骤2 选择待配置的函数，单击函数名称进入函数详情页。
- 步骤3 在函数详情页，选择函数版本。
- 步骤4 在“代码”页签下，单击“测试”，弹出“配置测试事件”对话框。
- 步骤5 填写如表5-61所示测试信息后，单击“保存”。

表 5-61 测试信息

| 参数     | 说明                                                                                      |
|--------|-----------------------------------------------------------------------------------------|
| 配置测试事件 | 可创建新的测试事件也可编辑已有的测试事件。<br>选择默认值：“创建新的测试事件”。                                              |
| 事件模板   | 选择"云数据库 GeminiDB Mongo"模板，使用系统内置GeminiDB Mongo事件模板。                                     |
| 事件名称   | 事件名称必须以大写或小写字母开头，支持字母（大写或小写），数字和下划线“_”（或中划线“-”），并以字母或数字结尾，长度为1-25个字符，例如：Gemini-123test。 |
| 测试事件   | 自动加载系统内置GeminiDB Mongo事件模板，本例不做修改。                                                      |

**步骤6** 单击“测试”，可以得到函数运行结果，函数会返回输入GeminiDB Mongo数据。

----结束

相关文档

除使用控制台外，函数工作流支持通过API的方式使用函数触发器，详情请参见[函数触发器API参考](#)。

5.6.12 使用云数据库 GeminiDB DynamoDB 触发器

本节介绍如何通过函数工作流控制台创建云数据库GeminiDB DynamoDB触发器。

云数据库 GeminiDB DynamoDB 触发器触发事件说明

GeminiDB推出了兼容DynamoDB API的服务，用户可以通过SDK/CLI访问数据库服务。关于GeminiDB兼容DynamoDB接口的详细介绍请参考[GeminiDB兼容DynamoDB接口](#)。

云数据库GeminiDB DynamoDB作为事件源，通过GeminiDB DynamoDB触发器可实现定时拉取DynamoDB数据库流表信息，并触发函数执行。GeminiDB DynamoDB触

发器每秒轮询四次以查找新记录，当出现新的流记录时，将同步调用FunctionGraph函数。

云数据库GeminiDB DynamoDB触发器的示例事件

- 以下为云数据库GeminiDB DynamoDB触发器的示例事件，具体参数解释请参考表5-62。

```
{
  "event_time": 1747905781,
  "trigger_type": "DYNAMODB",
  "instance_ip": "10.0.0.205",
  "table_name": "test_052201",
  "records": [
    {
      "region": "dcbhws",
      "event_id": "000001747880625259000",
      "event_name": "INSERT",
      "sequence_number": "000001747880625259000",
      "size_bytes": 170,
      "stream_view_type": "NEW_AND_OLD_IMAGES",
      "approximate_arrival_timestamp": "2025-05-22T02:23:45.259Z",
      "keys": {
        "Title": {
          "S": "test"
        },
        "Year": {
          "N": "2025"
        }
      },
      "old_image": {
        "Plot": {
          "S": "plot"
        },
        "Rating": {
          "N": "1"
        },
        "Title": {
          "S": "test"
        },
        "Year": {
          "N": "2025"
        }
      },
      "new_image": {
        "Plot": {
          "S": "plot"
        },
        "Rating": {
          "N": "1"
        },
        "Title": {
          "S": "test"
        },
        "Year": {
          "N": "2025"
        }
      }
    },
    {
      "region": "dcbhws",
      "event_id": "000001747881021099000",
      "event_name": "MODIFY",
      "sequence_number": "000001747881021099000",
      "size_bytes": 323,
      "stream_view_type": "NEW_AND_OLD_IMAGES",
      "approximate_arrival_timestamp": "2025-05-22T02:30:21.099Z",
      "keys": {
        "Title": {
          "S": "test"
        },
        "Year": {
          "N": "2025"
        }
      },
      "old_image": {
        "Plot": {
          "S": "plot"
        },
        "Rating": {
          "N": "1"
        },
        "Title": {
          "S": "test"
        },
        "Year": {
          "N": "2025"
        }
      },
      "new_image": {
        "Plot": {
          "S": "plot"
        },
        "Rating": {
          "N": "1"
        },
        "Title": {
          "S": "test"
        },
        "Year": {
          "N": "2025"
        }
      }
    }
  ]
}
```

```
    },
    "Title": {
      "S": "test"
    },
    "Year": {
      "N": "2025"
    }
  },
  "new_image": {
    "Plot": {
      "S": "plot"
    },
    "Rating": {
      "N": "2"
    },
    "Title": {
      "S": "test"
    },
    "Year": {
      "N": "2025"
    }
  }
}
]
```

表 5-62 GeminiDB DynamoDB 示例事件参数说明

| 参数                          | 类型     | 示例值                                                    | 描述                              |
|-----------------------------|--------|--------------------------------------------------------|---------------------------------|
| EventTime                   | Int    | 1747905781                                             | 触发器拉取完消息时的时间。                   |
| TriggerType                 | String | DYNAMODB                                               | 触发器类型。                          |
| Instancelp                  | String | 参考示例代码                                                 | DynamoDB的实例IP。                  |
| TableName                   | String | 参考示例代码                                                 | DynamoDB的表名。                    |
| Region                      | String | 参考示例代码                                                 | 局点信息。                           |
| EventId                     | String | 参考示例代码                                                 | 事件ID。                           |
| EventName                   | String | INSERT   MODIFY   REMOVE                               | 事件类名,枚举示例表示分别为插入，更改，删除。         |
| SequenceNumber              | String | 参考示例代码                                                 | 事件序列号。                          |
| SizeBytes                   | Int    | 100                                                    | 事件消息大小。                         |
| StreamViewType              | String | NEW_IMAGE   OLD_IMAGE   NEW_AND_OLD_IMAGES   KEYS_ONLY | 记录方式，枚举示例表示分别为新镜像，旧镜像，新旧镜像，仅主键。 |
| ApproximateArrivalTimestamp | String | 参考示例代码                                                 | 事件入库时间。                         |

| 参数       | 类型  | 示例值    | 描述   |
|----------|-----|--------|------|
| Keys     | Map | 参考示例代码 | 主键。  |
| OldImage | Map | 参考示例代码 | 旧镜像。 |
| NewImage | Map | 参考示例代码 | 新镜像。 |

约束与限制

- GeminiDB DynamoDB触发器当前支持“华北-北京四”。具体运行时函数是否支持使用请以控制台为准。
- 当前默认为每个DynamoDB实例上的每个表对应的流创建最多四个触发器。同一个FunctionGraph函数可以支持由不同表创建的多个DynamoDB触发器，但不允许由同一表创建多个DynamoDB触发器。
- 如果执行函数返回函数错误，FunctionGraph不会重试，同时这部分数据将被丢弃。
- 云数据库GeminiDB DynamoDB触发器调用的请求正文有效负载大小为6MB。

前提条件

- 云数据库GeminiDB兼容DynamoDB接口实例：
  - 已创建GeminiDB兼容DynamoDB接口实例，创建过程请参见[购买GeminiDB兼容DynamoDB接口实例](#)。
  - 请参见[给GeminiDB兼容DynamoDB接口实例设置安全组规则](#)配置对应的安全组规则。
- 函数及其配置：
  - 已创建函数。
  - 已配置函数的VPC委托权限，详情请参见[配置函数的委托权限](#)。
  - 已开启函数访问VPC内资源的网络配置，且为函数配置的VPC需要与云数据库实例配置中的VPC相同，详情请参见[配置函数访问VPC](#)。

创建 GeminiDB DynamoDB 触发器

- 步骤1 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
- 步骤2 选择待配置的函数，单击进入函数详情页。
- 步骤3 选择“设置 > 触发器”，单击“创建触发器”，弹出“创建触发器”对话框。

图 5-37 创建触发器



- 步骤4 参考[表5-63](#)配置触发器参数。

表 5-63 GeminiDB DynamoDB 触发器参数说明

| 参数    | 说明                                                                                                                                                                      | 取值样例                                    |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------|
| 触发器类型 | 必选参数。<br>选择“云数据库 GeminiDB<br>DynamoDB(DYNAMODB)”。                                                                                                                       | 云数据库 GeminiDB<br>DynamoDB(DYNAMO<br>DB) |
| 名称    | 必填参数。<br>填写触发器的名称。以中文或英文开头，可<br>以包含中文、英文、数字、下划线及中划<br>线，最大长度64位。                                                                                                        | Dynamodb-fg                             |
| 实例    | 必选参数。<br>选择已创建的GeminiDB DynamoDB数据库<br>实例。若无实例，可单击“创建实例”完成<br>创建。                                                                                                       | geminidb-fg                             |
| 用户名   | 必填参数。<br>填写DynamoDB数据库用户名。只能包含英<br>文字母、数字、下划线及中划线。                                                                                                                      | rwuser                                  |
| 密码    | 必填参数。<br>填写DynamoDB数据库密码。                                                                                                                                               | GeminiDB-test                           |
| 表名    | 必填参数。<br>填写DynamoDB数据库表名。只能包含英文<br>字母、数字、下划线及中划线。                                                                                                                       | table-test                              |
| 批处理大小 | 必选参数。<br>输入每批从数据库读取的记录的数量，设置<br>范围：1~10000。                                                                                                                             | 100                                     |
| 起始位置  | 必选参数。<br>选择流中开始读取数据的位置，有以下选<br>项： <ul style="list-style-type: none"><li>• TRIM_HORIZON：从最早被存储至分区的有效记录开始读取。</li><li>• LATEST：从分区中的最新记录开始读取，此设置可以保证总是读到分区中最新记录。</li></ul> | TRIM_HORIZON                            |

步骤5 单击“确定”，完成触发器的创建。

----结束

配置 GeminiDB DynamoDB 事件触发函数

步骤1 返回函数工作流控制台，在左侧的导航栏选择“函数 > 函数列表”。

步骤2 选择待配置的函数，单击函数名称进入函数详情页。

- 步骤3** 在函数详情页，选择函数版本。
- 步骤4** 在“代码”页签下，单击“测试”，弹出“配置测试事件”对话框。
- 步骤5** 填写如表5-64所示测试信息后，单击“保存”。

表 5-64 测试信息

| 参数     | 说明                                                                                      |
|--------|-----------------------------------------------------------------------------------------|
| 配置测试事件 | 可创建新的测试事件也可编辑已有的测试事件。<br>选择默认值：“创建新的测试事件”。                                              |
| 事件模板   | 选择空白模板，使用系统内置空白事件模板。                                                                    |
| 事件名称   | 事件名称必须以大写或小写字母开头，支持字母（大写或小写），数字和下划线“_”（或中划线“-”），并以字母或数字结尾，长度为1-25个字符，例如：Gemini-123test。 |
| 测试事件   | 自动加载系统内置空白事件模板，本例不做修改。                                                                  |

- 步骤6** 单击“测试”，可以得到函数运行结果。
- 结束

相关文档

除使用控制台外，函数工作流支持通过API的方式使用函数触发器，详情请参见[函数触发器API参考](#)。

5.6.13 使用设备接入（IoTDA）触发器

本节介绍如何通过函数工作流控制台创建IoTDA触发器，实现与设备接入服务IoTDA的协同工作。

关于设备接入服务IoTDA触发器事件源的具体介绍，请参见[支持的事件源](#)。

约束与限制

IoTDA触发器当前支持“华南-广州”、“华北-北京四”、“中国-香港”、“亚太-曼谷”区域使用。具体运行时函数是否支持使用请以控制台为准。

前提条件

- 已创建函数。
- 已创建IoTDA实例，创建详情请参见[购买IoTDA实例](#)。
- 已创建IoTDA实例的资源空间，创建详情请参见[创建资源空间](#)。

创建设备接入 IoTDA 触发器

- 步骤1** 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
- 步骤2** 选择待配置的函数，单击进入函数详情页。

**步骤3** 选择“设置 > 触发器”，右侧单击“创建触发器”，弹出“创建触发器”对话框。

**步骤4** 配置以下参数。

- 触发器类型：选择“设备接入（IoTDA）”。
- 实例：选择已经创建好的IoTDA实例。
- 数据来源、触发事件、资源空间：详情请参见表5-65。

表 5-65 配置说明

| 数据来源     | 触发事件           | 资源空间                                                                 |
|----------|----------------|----------------------------------------------------------------------|
| 设备       | 设备添加、设备删除、设备更新 | 选择资源空间。IoTDA实例支持创建多个资源空间，并指定其中一个作为“默认资源空间”。“默认资源空间”每个租户有且仅有一个，不允许删除。 |
| 设备属性     | 设备属性上报         |                                                                      |
| 设备消息     | 设备消息上报         |                                                                      |
| 设备消息状态   | 设备消息状态变更       |                                                                      |
| 设备状态     | 设备状态变更         |                                                                      |
| 产品       | 产品添加、产品删除、产品更新 |                                                                      |
| 设备异步命令状态 | 批量任务状态变更       |                                                                      |
| 运行日志     | 上报平台日志         |                                                                      |
| 批量任务     | 批量任务状态变更       |                                                                      |

**步骤5** 单击“确定”，完成IoTDA触发器的创建。

----结束

触发函数

**步骤1** 返回函数工作流控制台，在左侧的导航栏选择“函数 > 函数列表”。

**步骤2** 选择待配置的函数，单击函数名称进入函数详情页。

**步骤3** 在函数详情页，选择函数版本。

**步骤4** 在“代码”页签下，单击“测试”，弹出“配置测试事件”对话框。

**步骤5** 填写如表5-66参数后，单击“创建”。

表 5-66 测试信息

| 参数     | 说明                                   |
|--------|--------------------------------------|
| 配置测试事件 | 可创建新的测试事件，也可编辑已有的测试事件。默认是“创建新的测试事件”。 |
| 事件模板   | 选择“空白模板”。                            |

| 参数   | 说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 事件名称 | 事件名称必须以大写或小写字母开头，支持字母（大写或小写），数字和下划线“_”（或中划线“-”），并以字母或数字结尾，长度为1-25个字符，例如iotda-123test。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 测试事件 | <pre>{   "resource": "device",   "event": "create",   "event_time": "20240919T011335Z",   "event_time_ms": "2024-09-19T01:13:35.854Z",   "request_id": "75127474-1a26-4578-8847-3128d6101954",   "notify_data": {     "body": {       "app_id": "3d40caf3ddfc4e83815b54b50f13aad7",       "app_name": "DefaultApp_6439v2",       "device_id": "66eb7a0ffa8d9c36870c6892_ttytytytytyt",       "node_id": "ttytytytytyt",       "gateway_id": "66eb7a0ffa8d9c36870c6892_ttytytytytyt",       "node_type": "GATEWAY",       "auth_info": {         "auth_type": "SECRET",         "secure_access": false,         "timeout": 0       },       "product_id": "66eb7a0ffa8d9c36870c6892",       "product_name": "test",       "status": "INACTIVE",       "create_time": "20240919T011335Z"     }   } }</pre> |

**步骤6** 单击“测试”，可以得到函数运行结果，函数会返回输入的IoTDA消息数据。

----结束

相关文档

除使用控制台外，函数工作流支持通过API的方式使用函数触发器，详情请参见[函数触发器API参考](#)。

5.6.14 使用云日志服务（LTS）触发器

本节介绍如何通过函数工作流控制台创建LTS触发器，供您了解LTS触发器的使用方法。

关于云日志服务LTS触发器事件源的具体介绍，请参见[支持的事件源](#)。

约束与限制

- 当前除“华北-乌兰察布一”、“拉美-圣保罗一”外，其他区域均支持使用LTS触发器。具体运行时函数是否支持使用请以控制台为准。
- 当原始LTS事件消息超过75KB，会把原始LTS事件消息按照75KB维度拆分为多条消息触发执行函数。

前提条件

- 函数及其配置：

- 已创建函数。
- 已创建LTS FullAccess权限的委托，创建过程请参见[配置委托权限](#)。
- **云日志服务LTS：**
  - 已创建日志组，此处以LogGroup1为例，创建过程请参见[创建日志组](#)。
  - 已创建日志流，此处以LogTopic1为例，创建过程请参见[创建日志流](#)。
  - 已配置Agent，快速将ECS等服务器上日志采集到指定的日志组，详情请参见[安装ICAgent](#)。

创建 LTS 触发器

- 步骤1** 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
- 步骤2** 选择待配置的函数，单击进入函数详情页。
- 步骤3** 选择“设置 > 触发器”，单击“创建触发器”，弹出“创建触发器”对话框。

图 5-38 创建触发器



- 步骤4** 设置以下信息。
- 触发器类型：选择“云日志服务（LTS）”。
  - 日志组：选择已创建的日志组，例如：LogGroup1。
  - 日志流：选择已创建的日志流，例如：LogStream1。
- 步骤5** 单击“确定”，完成LTS触发器的创建。
- 结束

配置 LTS 事件触发函数

- 步骤1** 返回函数工作流控制台，在左侧的导航栏选择“函数 > 函数列表”。
- 步骤2** 选择待配置的函数，单击进入函数详情页。
- 步骤3** 在函数详情页，选择函数版本。
- 步骤4** 在“代码”页签下，单击“测试”，弹出“配置测试事件”对话框。
- 步骤5** 填写如[表5-67](#)所示测试信息后，单击“保存”。

表 5-67 测试信息

| 参数     | 说明                                         |
|--------|--------------------------------------------|
| 配置测试事件 | 可创建新的测试事件也可编辑已有的测试事件。<br>选择默认值：“创建新的测试事件”。 |
| 事件模板   | 选择“云日志服务（LTS）”模板，使用系统内置LTS事件模板。            |

| 参数   | 说明                                                                                   |
|------|--------------------------------------------------------------------------------------|
| 事件名称 | 事件名称必须以大写或小写字母开头，支持字母（大写或小写），数字和下划线“_”（或中划线“-”），并以字母或数字结尾，长度为1-25个字符，例如：lts-123test。 |
| 测试事件 | 自动加载系统内置lts事件模板，本例不做修改。                                                              |

**步骤6** 单击“测试”，可以得到函数运行结果，函数会返回输入LTS数据。

----结束

相关文档

除使用控制台外，函数工作流支持通过API的方式使用函数触发器，详情请参见[函数触发器API参考](#)。

5.6.15 使用消息通知服务（SMN）触发器

本节介绍如何通过函数工作流控制台创建SMN触发器，实现发布消息时触发函数运行。

关于消息通知服务SMN触发器事件源的具体介绍，请参见[支持的事件源](#)。

约束与限制

- 当前除“华北-乌兰察布一”、“亚太-雅加达”、“中东-利雅得”、“非洲-约翰内斯堡”外，其他区域均支持使用SMN触发器。具体运行时函数是否支持使用请以控制台为准。
- SMN触发器创建以后，不能停用，只能删除。

前提条件

- 已创建函数。
- 已创建SMN消息主题，此处以smn-test为例，创建过程请参见[创建消息主题](#)。

创建 SMN 触发器

- 步骤1** 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
- 步骤2** 选择待配置的函数，单击进入函数详情页。
- 步骤3** 选择“设置 > 触发器”，单击“创建触发器”，弹出“创建触发器”对话框。

图 5-39 创建触发器



- 步骤4** 设置以下信息。
- 触发器类型：选择“消息通知服务 (SMN)”。
  - 主题名称：选择主题名称，例如：smn-test。

**步骤5** 单击“确定”，完成SMN触发器的创建。

#### 📖 说明

SMN触发器创建完成后，会在SMN消息主题生成消息订阅。

----结束

## 触发函数运行

在“消息通知服务”控制台，为“smn-test”主题发布消息，具体操作步骤请参见[向主题发布文本消息](#)。消息发布以后，会自动触发函数运行。

发布消息的内容参见[表5-68](#)填写。

**表 5-68** 发布消息

| 字段   | 填写说明                        |
|------|-----------------------------|
| 消息标题 | 输入“SMN-Test”。               |
| 消息类型 | 选择“文本消息”。                   |
| 消息内容 | 输入以下内容：{"message":"hello"}。 |

## 查看函数运行结果

**步骤1** 返回函数工作流控制台，在左侧的导航栏选择“函数 > 函数列表”。

**步骤2** 选择函数，单击函数名称进入函数详情页。

**步骤3** 选择“监控 > 日志”，查询函数运行日志。

----结束

## 相关文档

除使用控制台外，函数工作流支持通过API的方式使用函数触发器，详情请参见[函数触发器API参考](#)。

### 5.6.16 使用对象存储服务（OBS）触发器

本节介绍如何通过函数工作流控制台创建OBS触发器，实现当OBS存储桶有文件更新时触发函数运行。

关于OBS触发器事件源的具体介绍，请参见[支持的事件源](#)。

#### OBS 触发器支持的事件类型

多个事件类型可以作用于同一个目标对象，即支持多选事件类型。

请参考[表5-69](#)选择需要使用的事件类型。

表 5-69 OBS 支持的事件类型

| 事件                       | 说明                                                                                                                           |
|--------------------------|------------------------------------------------------------------------------------------------------------------------------|
| ObjectCreated            | 表示所有创建对象的操作，包含Put、Post、Copy以及CompleteMultipartUpload。<br><b>约束与限制：</b> 如果选择了该事件，则不能再选择Put、Post、Copy或CompleteMultipartUpload。 |
| Put                      | 使用Put方法上传对象。                                                                                                                 |
| Post                     | 使用Post方法上传对象。                                                                                                                |
| Copy                     | 使用copy方法复制对象。                                                                                                                |
| CompleteMultipart Upload | 表示合并分段任务。                                                                                                                    |
| ObjectRemoved            | 表示所有删除对象的操作，包含Delete和DeleteMarkerCreated。<br><b>约束与限制：</b> 如果选择了该事件，则不能再选择Delete或DeleteMarkerCreated。                        |
| Delete                   | 指定对象版本号删除对象。                                                                                                                 |
| DeleteMarkerCreated      | 不指定对象版本号删除对象。                                                                                                                |

## 约束与限制

- OBS触发器当前支持“华北-北京一”、“华东-上海二”、“华东二”、“中国-香港”、“亚太-曼谷”、“亚太-新加坡”、“亚太-雅加达”、“土耳其-伊斯坦布尔”、“拉美-墨西哥城二”区域使用。具体运行时函数是否支持使用请以控制台为准。
- OBS触发器创建以后，不能停用，只能删除。
- 请确保创建的函数与OBS桶在同一区域下，不同区域创建的桶不通用。

## 前提条件

- **权限：**  
使用OBS作为事件订阅的事件源时，需要用户具备Tenant Administrator权限。相关操作请参考[给用户授权](#)。
- **函数及其配置：**
  - 已创建函数。
  - 已配置函数的OBS委托权限，委托的创建请参见[配置函数的委托权限](#)。
- **OBS桶：**  
已创建OBS桶，此处以名称为eventbucket的OBS桶为例。创建过程请参见[创建存储桶](#)。

创建 OBS 触发器

- 步骤1 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
- 步骤2 选择待配置的函数，单击进入函数详情页。
- 步骤3 选择“设置 > 触发器”，单击“创建触发器”，弹出“创建触发器”对话框。

图 5-40 创建触发器



- 步骤4 配置以下参数。

表 5-70 OBS 触发器参数说明

| 参数名称   | 参数说明                                                    | 取值样例                                                         |
|--------|---------------------------------------------------------|--------------------------------------------------------------|
| 触发器类型  | 必选参数。<br>选择“对象存储服务（OBS）”。                               | 对象存储服务（OBS）                                                  |
| 桶      | 必选参数。<br>用作事件源的OBS存储桶，创建成功后不支持修改。                       | obs-cff                                                      |
| 事件     | 必选参数。<br>选择触发函数的事件。支持选择的事件类型请参考 <a href="#">表5-69</a> 。 | “Put”、“Post”和“Delete”<br><br>（当对OBS桶中的文件进行更新、上传和删除操作时触发函数运行） |
| 事件通知名称 | 可选参数。<br>自定义的事件通知名称。用于在事件发生时，SMN服务给您推送消息。               | obs-event-test                                               |
| 前缀     | 可选参数。<br>用来限制以此关键字开头的对象的事件通知，该限制可以实现对OBS对象名的过滤。         | -                                                            |
| 后缀     | 可选参数。<br>用来限制以此关键字结尾的对象的事件通知，该限制可以实现对OBS对象名的过滤。         | -                                                            |

----结束

触发函数

登录对象存储服务控制台，将文件上传至“obs-cff”OBS存储桶，具体步骤请参见[上传文件](#)。

成功上传文件至“obs-cff” OBS存储桶后，会自动触发函数运行，可在函数执行日志查看执行结果。

## 查看函数运行结果

**步骤1** 返回函数工作流控制台，在左侧的导航栏选择“函数 > 函数列表”。

**步骤2** 选择函数，单击进入函数详情页。

**步骤3** 选择“监控 > 日志”，查询函数运行日志。

----结束

## 相关文档

除使用控制台外，函数工作流支持通过API的方式使用函数触发器，详情请参见[函数触发器API参考](#)。

### 5.6.17 使用 EventGrid 触发器（OBS 应用事件源）

本节介绍如何通过函数工作流控制台创建EventGrid触发器（OBS应用事件源），实现当OBS存储桶有文件更新时触发函数运行。

“拉美-圣保罗一”区域支持为共享函数配置EG（OBS应用事件源）触发器。

关于EventGrid触发器事件源的具体介绍，请参见[支持的事件源](#)。

## 约束与限制

- EventGrid触发器当前支持“华北-北京四”、“华北-乌兰察布一”、“华东-上海一”、“华南-广州”、“拉美-墨西哥城二”、“拉美-圣地亚哥”区域使用。具体运行时函数是否支持使用请以控制台为准。
- 确保创建的函数与OBS桶在同一区域下。

## 前提条件

- 权限：**  
使用OBS作为事件订阅的事件源时，需要用户具备Tenant Administrator权限。相关操作请参考[给用户授权](#)。
- 函数及其配置：**
  - 已创建函数。
  - 已配置函数的OBS委托权限，委托的创建请参见[配置函数的委托权限](#)。
- OBS桶：**  
已创建OBS桶，此处以名称为eventbucket的OBS桶为例。创建过程请参见[创建存储桶](#)。

### 创建 EventGrid 触发器（OBS 应用事件源）

- 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
- 选择待配置的函数，单击函数名称进入函数详情页。
- 选择“设置 > 触发器”，单击“创建触发器”，弹出“创建触发器”对话框。

图 5-41 创建触发器



4. 配置以下参数。

表 5-71 EG 触发器（OBS 应用事件源）参数说明

| 参数    | 说明                                                                                                                                                                                                                                                                                    | 取值样例               |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|
| 触发器类型 | 必选参数。<br>选择“OBS应用事件源”。                                                                                                                                                                                                                                                                | OBS应用事件源           |
| 触发器名称 | 必选参数。<br>触发器的名称。只能包含字母、数字、下划线和中划线。不能以数字、中划线开头。长度在 1-128 之间。                                                                                                                                                                                                                           | eg-obs             |
| 桶     | 必选参数。<br>选择已创建的OBS桶。若无OBS桶，可单击“创建桶”完成创建。                                                                                                                                                                                                                                              | eventbucket        |
| 事件类型  | 必选参数。<br>选择需要的触发事件类型，有以下事件类型： <ul style="list-style-type: none"><li>不指定对象版本号删除对象</li><li>指定对象版本号删除对象<br/>选择该事件类型需要开启多版本控制功能，具体详情请参见<a href="#">多版本控制功能启用时的对象删除机制</a>。</li><li>通过页面或Put请求创建或覆盖桶对象</li><li>通过页面或API请求合并分段任务</li><li>使用Copy请求创建或覆盖桶对象</li><li>使用Post请求创建或覆盖桶对象</li></ul> | 通过页面或Put请求创建或覆盖桶对象 |
| 对象名前缀 | 可选参数。<br>用来限制以此关键字开头的对象的事件通知，该限制可以实现对OBS对象名的过滤。                                                                                                                                                                                                                                       | -                  |
| 对象名后缀 | 可选参数。<br>用来限制以此关键字结尾的对象的事件通知，该限制可以实现对OBS对象名的过滤。                                                                                                                                                                                                                                       | -                  |
| 对象名编码 | 必选参数。<br>选择是否对对象名进行编码。默认为开启状态。                                                                                                                                                                                                                                                        | 开启                 |

5. 单击“确定”，完成触发器的创建。

为共享函数配置 EventGrid（OBS 应用事件源）触发器

“拉美-圣保罗一”区域支持为共享函数配置EG（OBS应用事件源）触发器，关于共享函数的相关介绍请参考[基于RAM共享函数资源](#)。

- 1. 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
- 2. 选择“共享函数”页签，单击函数名称进入函数详情页。
- 3. 后续创建步骤与普通函数相同，请参考[3](#)开始操作。

配置 EventGrid 事件触发函数

- 步骤1** 返回函数工作流控制台，在左侧的导航栏选择“函数 > 函数列表”。
- 步骤2** 选择待配置的函数，单击函数名称进入函数详情页。
- 步骤3** 在函数详情页，选择函数版本。
- 步骤4** 在“代码”页签下，单击“测试”，弹出“配置测试事件”对话框。
- 步骤5** 填写如下[表5-72](#)所示测试信息后，单击“保存”。

表 5-72 测试参数

| 参数     | 说明                                                                                     |
|--------|----------------------------------------------------------------------------------------|
| 配置测试事件 | 可创建新的测试事件，也可编辑已有的测试事件。<br>选择默认值：“创建新的测试事件”。                                            |
| 事件模板   | 选择“空白”模板，代码请参见 <a href="#">EG示例事件</a> 中“OBS应用事件源”。                                     |
| 事件名称   | 事件名称必须以大写或小写字母开头，支持字母（大写或小写），数字和下划线“_”（或中划线“-”），并以字母或数字结尾，长度为1-25个字符，例如：myobs-123test。 |
| 测试事件   | 使用新创建的测试事件。                                                                            |

----结束

相关文档

除使用控制台外，函数工作流支持通过API的方式使用函数触发器，详情请参见[函数触发器API参考](#)。

5.6.18 使用 EventGrid 触发器（RocketMQ 自定义事件源）

本节介绍如何通过函数工作流控制台创建EventGrid触发器（RocketMQ自定义事件源），实现使用RocketMQ自定义事件源触发函数执行。

关于EventGrid触发器事件源的具体介绍，请参见[支持的事件源](#)。

约束与限制

EventGrid触发器当前支持“华北-北京四”、“华北-乌兰察布一”、“华东-上海一”、“华南-广州”、“拉美-墨西哥城二”、“拉美-圣地亚哥”区域使用。具体运行时函数是否支持使用请以控制台为准。

前提条件

- **函数及其配置：**
  - 已创建函数。
  - 请参见[配置网络](#)开启函数访问VPC内资源的网络配置。
- **EventGrid：**
  - 已创建EG事件通道，创建过程请参见[创建eg事件通道](#)。
- **RocketMQ实例：**
  - 已创建RocketMQ实例，创建过程请参见[购买RocketMQ实例](#)。
  - 已创建RocketMQ topic，创建过程请参见[创建topic](#)。
  - 已创建RocketMQ 消费组，创建过程请参见[创建消费组](#)。

创建 EventGrid 触发器（RocketMQ 自定义事件源）

- 步骤1** 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
- 步骤2** 选择待配置的函数，单击进入函数详情页。
- 步骤3** 选择“设置 > 触发器”，单击“创建触发器”，弹出“创建触发器”对话框。

图 5-42 创建触发器



- 步骤4** 配置以下参数。

表 5-73 EG 触发器（RocketMQ）参数说明

| 参数    | 参数                                                          | 取值样例                            |
|-------|-------------------------------------------------------------|---------------------------------|
| 触发器类型 | 必选参数。<br>选择“分布式消息服务 RocketMQ版 (HC.ROCKETMQ)”。               | 分布式消息服务 RocketMQ版 (HC.ROCKETMQ) |
| 触发器名称 | 必选参数。<br>触发器的名称。只能包含字母、数字、下划线和中划线。不能以数字、中划线开头。长度在 1-128 之间。 | EG-RocketMQ                     |
| 事件通道  | 必选参数。<br>选择已创建的EG事件通道。若无可用事件通道，可单击“创建事件通道”完成创建。             | EGtest                          |

| 参数    | 参数                                              | 取值样例        |
|-------|-------------------------------------------------|-------------|
| 实例    | 必选参数。<br>选择已创建的RocketMQ实例。若无可用实例，可单击“创建实例”完成创建。 | rocketmq-fg |
| Topic | 必选参数。<br>选择已创建的RocketMQ topic。                  | topic-test  |
| 消费组   | 必选参数。<br>选择已经创建的RocketMQ 消费组。                   | fgtest      |

**步骤5** 单击“确定”，完成EG触发器的创建。

----结束

配置 EventGrid 事件触发函数

- 步骤1 返回函数工作流控制台，在左侧的导航栏选择“函数 > 函数列表”。
- 步骤2 选择待配置的函数，单击进入函数详情页。
- 步骤3 在函数详情页，选择函数版本。
- 步骤4 在“代码”页签下，单击“测试”，弹出“配置测试事件”对话框。
- 步骤5 填写如下表5-74所示测试信息后，单击“保存”。

表 5-74 测试参数

| 参数     | 说明                                                                                        |
|--------|-------------------------------------------------------------------------------------------|
| 配置测试事件 | 可创建新的测试事件，也可编辑已有的测试事件。<br>选择默认值：“创建新的测试事件”。                                               |
| 事件模板   | 选择"分布式消息服务 RocketMQ版 (HC.ROCKETMQ)"模板，使用系统内置 HC.ROCKETMQ事件模板。                             |
| 事件名称   | 事件名称必须以大写或小写字母开头，支持字母（大写或小写），数字和下划线“_”（或中划线“-”），并以字母或数字结尾，长度为1-25个字符，例如：rocketmq-123test。 |
| 测试事件   | 自动加载系统内置eg-RocketMQ事件模板，本例不做修改。                                                           |

**步骤6** 单击“测试”，可以得到函数运行结果，函数会返回输入eg-rocketmq消息数据。

----结束

## 相关文档

除使用控制台外，函数工作流支持通过API的方式使用函数触发器，详情请参见[函数触发器API参考](#)。

### 5.6.19 使用 EventGrid 触发器（RabbitMQ 自定义事件源）

本节介绍如何通过函数工作流控制台创建EventGrid触发器（RabbitMQ自定义事件源），实现使用RabbitMQ自定义事件源触发函数执行。

关于EventGrid触发器事件源的具体介绍，请参见[支持的事件源](#)。

## 约束与限制

EventGrid触发器当前支持“华北-北京四”、“华北-乌兰察布一”、“华东-上海一”、“华南-广州”、“拉美-墨西哥城二”、“拉美-圣地亚哥”区域使用。具体运行时函数是否支持使用请以控制台为准。

## 前提条件

- **函数及其配置：**
  - 已创建函数。
  - 请参见[配置网络](#)开启函数访问VPC内资源的网络配置，开启后需在RabbitMQ服务安全组配置对应子网的权限，参考RabbitMQ实例的前提条件配置。
- **EventGrid：**
  - 已创建EG事件通道，创建过程请参见[创建eg事件通道](#)。
- **RabbitMQ实例：**
  - 已创建RabbitMQ实例，创建操作请参见[购买RabbitMQ实例](#)。
  - 创建Vhost、Exchange和Queue。
    - i. 创建RabbitMQ Vhost，创建操作请参见[创建RabbitMQ Vhost](#)。
    - ii. 创建RabbitMQ Exchange，创建操作请参见[创建RabbitMQ Exchange](#)。
    - iii. 创建RabbitMQ Queue，创建操作请参见[创建RabbitMQ Queue](#)。
    - iv. 绑定RabbitMQ Exchange和RabbitMQ Queue，绑定操作请参见[绑定RabbitMQ Exchange](#)、[绑定RabbitMQ Queue](#)。

#### 说明

Vhost是一个相对独立的RabbitMQ服务，用于管理Exchange、Queue。一个RabbitMQ实例下可以有多个Vhost，一个Vhost里可以有若干个Exchange和Queue，具体详情请参见[RabbitMQ业务使用流程](#)。

- 确认实例安全组规则是否配置正确。
  - i. 在RabbitMQ实例详情页面的“基本信息 > 网络”，单击安全组名称，跳转到安全组页面。
  - ii. 选择“入方向规则”，查看安全组入方向规则。
    - 1) 实例未开启SSL开关
      - 如果是VPC内访问，实例安全组入方向规则，需要允许端口5672的访问。
      - 如果是公网访问，需要允许端口15672的访问。

- 2) 实例已开启SSL开关
- 如果是VPC内访问，实例安全组入方向规则，需要允许端口5671的访问。

▪ 如果是公网访问，需要运行端口15671的访问。

创建 EventGrid 触发器（ RabbitMQ 自定义事件源 ）

- 步骤1 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
- 步骤2 选择待配置的函数，单击进入函数详情页。
- 步骤3 选择“设置 > 触发器”，单击“创建触发器”，弹出“创建触发器”对话框。

图 5-43 创建触发器



- 步骤4 配置以下参数。

表 5-75 EG 触发器（ RabbitMQ ）参数说明

| 参数    | 说明                                                          | 取值样例                            |
|-------|-------------------------------------------------------------|---------------------------------|
| 触发器类型 | 必选参数。<br>选择“分布式消息服务 RabbitMQ版 (HC.RABBITMQ)”。               | 分布式消息服务 RabbitMQ版 (HC.RABBITMQ) |
| 触发器名称 | 必选参数。<br>触发器的名称。只能包含字母、数字、下划线和中划线。不能以数字、中划线开头。长度在 1-128 之间。 | EG-RabbitMQ                     |
| 事件通道  | 必选参数。<br>选择已创建的EG事件通道。若无事件通道，可单击“创建事件通道”完成创建。               | EGtest                          |
| 实例    | 必选参数。<br>选择已创建的RabbitMQ实例。若无实例，可单击“创建实例”完成创建。               | rabbitmq-fg                     |
| 用户名   | 必选参数。<br>填写连接RabbitMQ实例的用户名。                                | fgtest                          |
| 密码    | 必选参数。<br>填写连接RabbitMQ实例的密码。                                 | testrabbitmq                    |
| 虚拟机名称 | 必选参数。<br>填写RabbitMQ实例的虚拟机名称。                                | rabbitmqEX                      |
| 队列    | 必选参数。<br>填写RabbitMQ实例的队列。                                   | rabbitmqQueue                   |

**步骤5** 单击“确定”，完成EG触发器的创建。

----结束

配置 EventGrid 事件触发函数

- 步骤1** 返回函数工作流控制台，在左侧的导航栏选择“函数 > 函数列表”。
- 步骤2** 选择待配置的函数，单击进入函数详情页。
- 步骤3** 在函数详情页，选择函数版本。
- 步骤4** 在“代码”页签下，单击“测试”，弹出“配置测试事件”对话框。
- 步骤5** 填写如下表5-76所示测试信息后，单击“保存”。

表 5-76 测试参数

| 参数     | 说明                                                                                        |
|--------|-------------------------------------------------------------------------------------------|
| 配置测试事件 | 可创建新的测试事件，也可编辑已有的测试事件。<br>选择默认值：“创建新的测试事件”。                                               |
| 事件模板   | 选择"分布式消息服务 RabbitMQ版 (HC.RABBITMQ)"模板，使用系统内置 HC.RABBITMQ事件模板。                             |
| 事件名称   | 事件名称必须以大写或小写字母开头，支持字母（大写或小写），数字和下划线“_”（或中划线“-”），并以字母或数字结尾，长度为1-25个字符，例如：rabbitmq-123test。 |
| 测试事件   | 自动加载系统内置eg-RabbitMQ事件模板，本例不做修改。                                                           |

**步骤6** 单击“测试”，可以得到函数运行结果，函数会返回输入eg-rabbitmq消息数据。

----结束

相关文档

除使用控制台外，函数工作流支持通过API的方式使用函数触发器，详情请参见[函数触发器API参考](#)。

5.6.20 管理函数的触发器

本章节介绍如何通过函数工作流控制台，查看和管理创建过的触发器。

函数触发器列表

登录[函数工作流控制台](#)，在左侧导航栏选择“函数 > 触发器列表”，如图5-44所示，即可查看创建过的所有触发器，触发器类型页签会优先展示用户当前使用的触发器。

单击“所属函数”列下的函数链接，可跳转至函数详情页对触发器进行停用、启用或删除操作。

图 5-44 触发器展示



停用或启用触发器

已经创建的触发器，可以通过设置停用或启用，控制触发器的状态。**SMN触发器、APIG触发器创建以后，不能停用，只能删除。**

- 步骤1 登录[函数工作流控制台](#)，在左侧导航栏选择“函数 > 函数列表”。
- 步骤2 单击函数名称，进入函数详情界面。
- 步骤3 选择“设置 > 触发器”，进入“触发器”页签，在需要停用/启用的触发器所在行，单击“停用”或“启用”，停用或启用相应触发器。

----结束

删除触发器

- 步骤1 返回函数工作流控制台，在左侧的导航栏选择“函数 > 函数列表”。
- 步骤2 单击函数名称，进入函数详情界面。
- 步骤3 单击“触发器”，进入“触发器”页签，在需要删除的触发器所在行，单击“删除”，即可删除触发器。

----结束

相关文档

除使用控制台外，函数工作流支持通过API的方式使用函数触发器，详情请参见[函数触发器API参考](#)。

5.7 在线调试函数

本章节介绍如何通过函数工作流控制台管理测试事件，并通过测试事件在线调试函数。

操作场景

完成函数的各项配置后，在函数详情页的“代码”页签下，可自定义配置测试事件，验证函数能否成功执行。

有关函数在线调试测试事件的操作，可以参考指导视频[使用空白模板创建函数](#)中调试测试事件的操作介绍。

约束与限制

每个函数最多支持配置10个测试事件。

创建测试事件

测试事件配置并保存后可重复用于测试。

- 步骤1 登录[函数工作流控制台](#)，在左侧导航栏选择“函数 > 函数列表”，进入函数页面。
- 步骤2 单击函数名称，进入函数详情界面。
- 步骤3 在函数详情页，选择函数版本，单击“测试”，弹出“配置测试事件”弹窗。
- 步骤4 在“配置测试事件”弹窗参考[表5-77](#)填写测试事件信息。

表 5-77 测试事件信息

| 参数     | 说明                                                                                                                                             |
|--------|------------------------------------------------------------------------------------------------------------------------------------------------|
| 配置测试事件 | 必选参数。<br>可创建新的测试事件也可编辑已有的测试事件，选择“创建新的测试事件”。                                                                                                    |
| 事件模板   | 必选参数。 <ul style="list-style-type: none"><li>使用空白模板需要自行编辑测试事件。</li><li>使用其他普通事件模板或云事件模板，会自动加载相对应的测试事件，事件模板说明如<a href="#">表5-78</a>所示。</li></ul> |
| 事件名称   | 必填参数。<br>事件名称必须以大写或小写字母开头，支持字母（大写或小写），数字和下划线“_”（或中划线“-”），并以字母或数字结尾，长度为1-25个字符，例如：event-123test。                                                |
| 测试事件   | 必填。<br>在代码编辑区中编辑测试事件代码。                                                                                                                        |

表 5-78 事件模板说明

| 模板名称                | 模板说明                    |
|---------------------|-------------------------|
| API 网关服务（APIG专享版）   | 模拟APIG（专享版）事件触发函数。      |
| 云审计服务（CTS）          | 模拟CTS事件触发函数。            |
| 文档数据库服务（DDS）        | 模拟DDS事件触发函数。            |
| 云数据库 GeminiDB Mongo | 模拟GeminiDB Mongo事件触发函数。 |
| 数据接入服务（DIS）         | 模拟DIS事件触发函数。            |
| 云日志服务（LTS）          | 模拟LTS事件触发函数。            |
| 消息通知服务（SMN）         | 模拟SMN事件触发函数。            |

| 模板名称                           | 模板说明                             |
|--------------------------------|----------------------------------|
| 定时触发器（TIMER）                   | 模拟TIMER事件触发函数。                   |
| 分布式消息服务 Kafka版（KAFKA）          | 模拟Kafka事件触发函数。                   |
| 开源Kafka（OPENSOURCEKAFKA）       | 模拟开源Kafka事件触发函数。                 |
| 分布式消息服务 RabbitMQ版（RABBITMQ）    | 模拟RabbitMQ事件触发函数。                |
| 分布式消息服务 RabbitMQ版（HC.RABBITMQ） | 模拟RabbitMQ事件（通过EventGrid）触发函数。   |
| 分布式消息服务 RabbitMQ版（HC.ROCKETMQ） | 模拟RocketMQ事件（通过EventGrid）触发函数。   |
| 空白模板                           | 模板事件为：{"key": "value"}，可以根据需要修改。 |
| 登录安全实时分析                       | 可以作为“登录安全实时分析”函数模板的输入。           |
| 图片分类                           | 可以作为“实时图片分类（按图片内容）”函数模板的输入。      |
| 图片鉴黄                           | 可以作为“图片鉴黄”函数模板的输入。               |
| 语音识别                           | 可以作为“语音识别”函数模板的输入。               |

**步骤5** 单击“创建”，完成测试事件创建。

----结束

测试函数

- 步骤1** 返回函数工作流控制台，在左侧的导航栏选择“函数 > 函数列表”。
- 步骤2** 单击函数名称，进入函数详情界面。
- 步骤3** 在函数详情页，如图5-45所示，选择函数版本，选择测试事件，单击“测试”。

图 5-45 选择测试事件



- 步骤4** 单击“测试”，在“代码”页签下将弹出“执行结果”窗口。
- 在“执行结果”中可查看函数的日志，此处最多支持显示2K的日志。如需查看完整日志，请参见[配置日志组、日志流和查看函数日志](#)。
- 结束

修改测试事件

- 步骤1** 返回函数工作流控制台，在左侧的导航栏选择“函数 > 函数列表”。
- 步骤2** 单击函数名称，进入函数详情界面。
- 步骤3** 在函数详情页，单击“配置测试事件”，弹出“配置测试事件”页。
- 步骤4** 在“配置测试事件”界面修改测试信息，如[表5-79](#)所示。

表 5-79 测试信息

| 参数       | 说明          |
|----------|-------------|
| 创建新的测试事件 | 重新创建新的测试事件。 |
| 编辑已有测试事件 | 修改已有的测试事件。  |
| 测试事件     | 修改测试事件代码。   |

- 步骤5** 单击“保存”，完成测试事件修改。
- 结束

## 删除测试事件

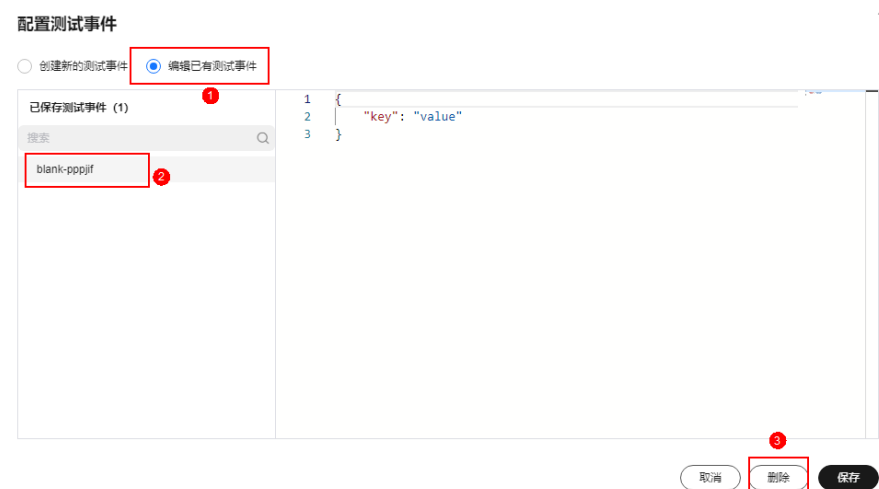
- 步骤1** 返回函数工作流控制台，在左侧的导航栏选择“函数 > 函数列表”。
- 步骤2** 单击函数名称，进入函数详情页。
- 步骤3** 在“代码”页签中选择“配置测试事件”进入编辑界面，如图5-46所示。

图 5-46 选择配置测试事件



- 步骤4** 如图5-47所示，在“配置测试事件”的编辑界面中，选择“编辑已有测试事件”，然后在左侧“已保存测试事件”列表中选中待删除事件名称，单击“删除”。

图 5-47 删除测试事件



----结束

## 相关文档

除使用控制台外，函数工作流支持通过API的方式管理函数测试事件，详情请参见[函数测试事件API参考](#)。

# 6 调用函数

函数创建和配置完成后，可通过RESTful API或者云服务事件源触发函数执行。

- RESTful API：通过调用API直接触发函数。
- 云服务事件源：由云服务触发函数，即使用函数触发器。

根据客户端是否需要等待函数的处理结果，函数的调用可分为同步调用和异步调用两种方式。

## 同步调用

同步调用是指客户端调用函数后，FunctionGraph会立即执行函数，等函数执行完成后再向客户端返回响应和执行结果。

### 同步调用应用场景：

同步调用的响应时间较短，适用于需要实时响应的业务场景。

#### 应用场景示例一：实时数据处理与查询

需要快速获取实时数据（如订单状态、支付结果）。同步调用能够立即处理请求并返回结果，满足实时性需求。

#### 应用场景示例二：即时交互与控制

需要即时看到操作结果（如消息发送成功）。同步调用能够确保操作的实时性和反馈的即时性。

## 异步调用

异步调用是指客户端调用函数后，FunctionGraph会将请求排队，不等待函数的执行结果，直接向客户端返回响应，FunctionGraph会在系统空闲的情况下逐个处理排队的请求。

异步调用场景下客户端无法实时感知函数的执行结果，如需获取异步请求的结果通知或者设置异步请求失败重试，请参见[配置函数的异步调用策略](#)。也可以使用[异步执行函数](#)接口异步触发函数。

### 异步调用应用场景：

异步调用适用于可以延迟处理，不需要实时响应，且可能需要较长时间或大量资源的任务。异步调用能够提升系统的响应速度和吞吐量，同时确保任务的可靠执行。

应用场景示例一：媒体处理与转换

- 图片压缩与格式转换：上传图片后，系统在后台异步处理图片，执行压缩、裁剪或格式转换，完成后将处理的图片存储于指定位置。图片处理需要一定时间，异步调用可避免等待。
- 视频转码与剪辑：上传视频后，系统在后台异步完成视频的转码、剪辑或添加水印。视频处理耗时较长，异步调用可以释放系统资源，保障其他任务正常运行。

应用场景示例二：数据处理与集成

- ETL任务（数据清洗与转换）：定期从数据库中提取数据，进行清洗、转换和加载到目标存储或分析平台，完成后生成报告并通知相关人员。ETL任务涉及大量数据和复杂计算，异步调用可避免系统阻塞，提高处理效率。
- 日志分析与统计：收集系统日志后，在后台异步进行日志分析、异常检测及统计，完成后将结果存储或发送至监控系统。日志分析需要较长时间，异步调用确保系统在高峰期间仍能正常运行。

约束与限制

- 同步调用场景下，函数最大运行时长限制为15分钟（900秒）。
- 异步调用场景下，函数最大运行时长限制为259200秒（3天）。如果业务有更大的最大执行时长需求，请[提交工单](#)申请。  
在函数详情页中支持自定义配置函数执行超时时长，具体请参见[配置函数的常规信息](#)，配置的函数执行超时时长需小于函数最大运行时长限制。
- 如果函数执行端到端时延超过90s，建议使用异步调用。如果使用同步调用，会因为网关限制，时延超过90s后无法收到同步响应。

函数触发器的调用方式

根据函数调用方式，函数的触发器可分为同步调用和异步调用。关于函数调用的更多说明请参见[调用函数](#)。

- 同步调用：客户端调用函数后，FunctionGraph会立即执行函数，等函数执行完成后向客户端返回响应和执行结果。
- 异步调用：客户端调用函数后，FunctionGraph会将请求排队，不等待函数的执行结果，直接向客户端返回响应，FunctionGraph会在系统空闲的情况下逐个处理排队的请求。

表 6-1 函数触发器的调用机制

| 触发器                       | 调用方式                                                  |
|---------------------------|-------------------------------------------------------|
| API网关服务（APIG专享版）触发器       | 默认为同步调用，但可以修改为异步调用，具体配置方式请参考 <a href="#">配置异步调用</a> 。 |
| API Connect（APIC）触发器      | 默认为同步调用，但可以修改为异步调用，具体配置方式请参考 <a href="#">配置异步调用</a> 。 |
| 定时触发器（TIMER）触发器           | 默认为同步调用，且不可修改。                                        |
| 云数据库 GeminiDB DynamoDB触发器 | 默认为同步调用，且不可修改。                                        |

| 触发器                                   | 调用方式           |
|---------------------------------------|----------------|
| 云审计服务（CTS）触发器                         | 默认为异步调用，且不可修改。 |
| 文档数据库服务（DDS）<br>触发器                   | 默认为异步调用，且不可修改。 |
| 数据接入服务（DIS）触发器                        | 默认为异步调用，且不可修改。 |
| 分布式消息服务 Kafka版<br>（KAFKA）触发器          | 默认为异步调用，且不可修改。 |
| 开源Kafka<br>（OPENSOURCEKAFKA）<br>触发器   | 默认为异步调用，且不可修改。 |
| 分布式消息服务<br>RabbitMQ版<br>（RABBITMQ）触发器 | 默认为异步调用，且不可修改。 |
| 云数据库 GeminiDB<br>Mongo触发器             | 默认为异步调用，且不可修改。 |
| 设备接入（IoTDA）触发器                        | 默认为异步调用，且不可修改。 |
| 云日志服务（LTS）触发器                         | 默认为异步调用，且不可修改。 |
| 消息通知服务（SMN）触<br>发器                    | 默认为异步调用，且不可修改。 |
| EventGrid触发器                          | 默认为异步调用，且不可修改。 |

配置函数的异步调用

以APIG触发器为例，在已创建函数并配置APIG触发器的情况下，配置异步调用。

1. 登录函数工作流控制台，在左侧导航栏选择“函数 > 函数列表”，单击函数名称进入函数详情页，选择“设置 > 触发器”。
2. 单击已配置的APIG触发器名称，跳转至APIG服务页面。

图 6-1 单击触发器名称



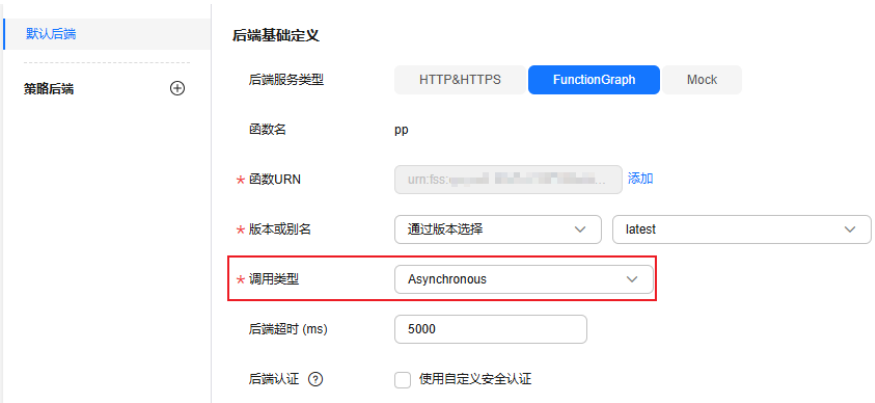
3. 单击右上角的”编辑”。

图 6-2 单击“编辑”



4. 单击“下一步”到“后端基础定义”界面，修改调用类型为“Asynchronous”。

图 6-3 修改调用类型



5. 单击“立即完成”，进行保存。

函数调用的重试机制

函数在同步调用或异步调用执行失败时，可以通过以下重试机制进行操作。

- 同步调用  
同步调用执行失败，需要自行重试。
- 异步调用  
异步调用可配置最大重试次数和消息最大有限期，具体配置方法请参见[配置函数的异步调用策略](#)。  
FunctionGraph会根据配置的最大重试次数和消息最大有限期（最大有限期为24小时）进行重试。重试次数和配置的最大重试次数一致，重试有效期和配置的消息最大有效期一致。

幂等性

在编程中，幂等性指应用程序或组件具备识别重复事件和防止重复、不一致或数据丢失的能力。若需要函数保持幂等性，可以通过函数逻辑设计来正确处理重复的事件。

幂等函数逻辑有助于减少以下问题：

- 不必要的API调用
- 代码处理时间
- 数据不一致
- 限制
- 延迟

请确保函数代码可以多次处理相同的事件，而不会导致重复的事务或其他不必要的副作用。如果函数不满足幂等性要求，则当函数调用异常、客户端重试或依赖函数内部重试时，可能会导致重复的事务或其他不必要的副作用。

## 相关文档

- 除使用控制台外，函数工作流支持通过API的方式调用函数，详情请参见[同步执行函数](#)和[异步执行函数](#)。
- 调用函数的常见问题排查方法，请参见[调用函数常见问题](#)。

# 7 管理函数

## 7.1 配置函数初始化

本章节指导如何通过函数工作流控制台，配置函数的初始化。

### 函数初始化概述

函数初始化是指调用函数时对其进行初始化设置的过程，包括设置函数的初始状态、分配资源和为函数内部的变量赋予初始值等。

FunctionGraph是在函数实例启动成功后执行函数的初始化，初始化执行成功后，函数实例才能开始处理函数的调用请求。

FunctionGraph保证一个函数实例在生命周期内只成功执行一次函数初始化。函数初始化的执行时间也会被计量，用户需要为此付费，计费方式与函数的调用处理一致。

一般场景下，多个请求处理可以共享的业务逻辑适合放到初始化函数，以降低函数时延。例如深度学习场景下加载规格较大的模型、数据库场景下连接池构建。关于使用函数初始化入口Initializer进行函数开发的更多说明请参见[函数初始化入口Initializer](#)。

### 约束与限制

- HTTP函数的“初始化配置”开关默认开启，且不可关闭。
- 函数初始化入口需要和函数执行入口在同一文件下。
- 开启函数初始化功能后，各个运行时的函数初始化入口命名规范与原有函数执行入口保持一致。函数执行入口信息可参考[函数执行入口](#)。
- 自定义镜像事件函数暂不支持进行初始化配置。
- 请勿在初始化函数中执行挂载磁盘的读写操作，这将导致函数无法读取磁盘中的内容。

### 配置函数初始化

1. 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
2. 单击函数的名称，进入函数详情页。
3. 在“设置”页签下选择“生命周期”。

4. 在设置页面开启“初始化配置”，启用函数的初始化，并参考[表7-1](#)配置初始化参数，配置完成后单击“保存”。

图 7-1 开启初始化配置

生命周期

初始化配置

\* 初始化超时时间（秒）

3

\* 函数初始化入口

index.initializer

函数初始化入口需要和函数执行入口在同一文件下; 格式为[文件名].[执行函数名], 不超过128个字符

表 7-1 初始化配置参数说明

| 参数         | 说明                                                                                                                                                                                                   |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 初始化超时时间（秒） | 函数初始化的超时时间，可设置范围为1-300秒。                                                                                                                                                                             |
| 函数初始化入口    | <b>仅事件函数可配置此参数，HTTP函数不显示该参数。</b><br>函数初始化入口需要和函数执行入口在同一个文件下。各运行时的函数初始化入口命名规范与原有函数执行入口保持一致，如Node.js和Python函数的执行入口命名规则为：[文件名].[执行函数名]，且不超过128个字符。<br><b>说明</b><br>函数代码配置信息请参见 <a href="#">配置函数代码</a> 。 |

初始化函数代码示例

- Node.js ([initializer入口介绍](#))

```
exports.initializer = function(context, callback) {  
  callback(null, "");  
};
```
- Python ([initializer入口介绍](#))

```
def my_initializer(context):  
    print("hello world!")
```
- Java ([initializer入口介绍](#))

```
public void my_initializer(Context context)  
{  
    RuntimeLogger log = context.getLogger();  
    log.log(String.format("ak:%s", context.getSecurityAccessKey()));  
}
```
- PHP ([initializer入口介绍](#))

```
<?php  
Function my_initializer($context) {  
    echo 'hello world' . PHP_EOL;  
}  
?>
```

相关文档

除使用控制台外，函数工作流支持通过API的方式管理函数的生命周期，详情请参见[函数生命周期管理API](#)。

## 7.2 配置函数的常规信息

函数的常规配置信息如函数执行入口、执行超时时间、内存等，会在创建函数时根据所选择的运行时语言默认填写，用户可根据实际业务需要修改常规信息的配置。

### 配置常规信息

1. 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
2. 单击函数的名称，进入函数详情页。
3. 在“设置”页签下选择“常规设置”。
4. 在常规设置页面参考[表1 基本信息配置说明](#)修改常规信息。

表 7-2 常规信息配置说明

| 参数     | 说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 函数名称   | 创建函数时配置的函数名称。<br>在创建函数时设置，无法修改。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 函数版本   | 函数的系统版本。<br>当前新创建函数的版本均为“v2”，无法修改。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 所属应用   | 函数所属应用，实际作用类似文件夹功能，未来会通过标签分组的方式来管理函数的分类。<br>当前新创建函数的所属应用均为“default”，无法修改。                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 运行时    | 函数的运行时语言。<br>运行时语言在创建函数时设置，无法修改。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 函数执行入口 | 函数执行入口是函数代码中指定的处理逻辑起点，通常是一个特定的函数或方法。当FunctionGraph函数被触发时，会从执行入口（handler）开始执行函数代码。<br>格式根据运行时语言的不同，函数执行入口的配置规则不同，不超过128个字符。 <ul style="list-style-type: none"><li>• Node.js、Python和PHP函数执行入口的配置规则：[文件名].[执行函数名]，必须包含“.”。<br/>默认：index.handler</li><li>• Java函数执行入口的配置规则：[包名].[类名].[执行函数名]。<br/>默认：com.huawei.demo.TriggerTests.apigTest</li><li>• Go函数执行入口的配置规则：与代码文件中的可执行文件名保持一致。<br/>默认：用户编译的可执行文件名为handler，则填handler。</li><li>• C#函数执行入口的配置规则：[程序集名]::[命名空间].[类名]::[执行函数名]。<br/>默认：CsharpDemo::CsharpDemo.Program::MyFunc</li></ul> |

| 参数        | 说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 企业项目      | 函数归属的企业项目，可以使用不同的企业项目对函数进行项目级管理。可选择已创建的企业项目，并将函数添加至企业项目中，默认选择“default”。<br>仅开通企业项目后可看到该参数，开通方法请参见 <a href="#">如何开通企业项目</a> 。                                                                                                                                                                                                                                                                                                                                                              |
| 执行超时时间（秒） | 函数执行的超时时间，超过时间还未执行完成的函数将被强行停止。如果函数的执行时间超过900秒，建议采用异步调用的方式。<br><b>说明</b><br>函数工作流服务中有三个关于函数运行时长限制： <ul style="list-style-type: none"><li>函数最大运行时长：系统内置的函数运行时长限制，旨在防止函数运行时间过长。同步调用场景下，函数最大运行时长限制为15分钟；异步调用场景下，函数最大运行时长限制为259200秒（3天）。具体可参见<a href="#">调用函数</a>。</li><li>执行超时时间（秒）：用户可根据具体业务需求，自行配置函数执行超时时间，须小于系统内置的函数最大运行时长。</li><li>前端超时时间：前端页面内置的超时提醒，前端页面等待后端返回结果的最大时长为90秒。如果设置的超时时间超过90秒，且函数实际运行时间也超过90秒，前端页面会提醒超时，但后端正常运行，运行结束后仍可在日志中查看返回结果。</li></ul> 函数执行超时时间设置范围为3~259,200秒。默认取值为3。 |
| 内存（MB）    | 函数执行可使用的内存大小。默认取值为128。<br>取值范围：128、256、512、768、1024、1280、1536、1792、2048、2560、3072、3584、4096、8192、10240                                                                                                                                                                                                                                                                                                                                                                                    |
| 描述        | 函数的描述信息。可自定义填写用于描述函数业务功能。<br>不超过512个字符。                                                                                                                                                                                                                                                                                                                                                                                                                                                   |

5. 修改完成后单击“保存”，保存配置。

相关文档

除使用控制台外，函数工作流支持通过API的方式管理函数的生命周期，详情请参见[函数生命周期管理API](#)。

7.3 配置函数的磁盘挂载

本章节介绍如何通过函数工作流控制台，配置函数的磁盘挂载，扩展函数的存储空间。

操作场景

FunctionGraph支持通过挂载文件系统为函数提供持久化存储能力，挂载文件系统可为函数带来如下优势：

- 多个函数之间可以挂载同一个文件系统，共享访问已经配置好的文件系统。
- 利用ECS已有的存储能力，可实现计算资源的动态扩展。

- 相比于临时存储空间/tmp，可以极大的扩展函数的执行存储空间。

#### 说明

函数的临时存储空间/tmp路径下可存储临时文件，默认为512MB，最大不能超过10,240MB。

FunctionGraph当前支持挂载以下文件系统：

- **SFS Turbo文件系统**  
专属弹性文件服务（SFS Turbo）为用户提供一个完全托管的共享文件存储，能够弹性伸缩至PB级规模，最高可提供亚毫秒级低时延、千万级IOPS、百GB带宽。具备高可用性和持久性，为海量的小文件、低延迟高IOPS型应用提供有力支持。**文件系统支持使用的区域详情请参见[SFS功能总览](#)。**
- **SFS通用文件系统**  
弹性文件服务（Scalable File Service, SFS）通用文件系统为用户提供一个完全托管的共享文件存储，能够弹性伸缩至PB规模、TB级带宽，具备高可用性和持久性，为海量数据、高带宽型应用提供有力支持。**通用文件系统支持使用的区域详情请参见[SFS功能总览](#)。**
- **弹性云服务器ECS共享目录**  
ECS共享目录，是通过NFS（网络文件系统）把ECS上的指定目录设置为共享目录，函数可以挂载对应目录进行读写等操作，实现计算资源的动态扩展，同时通过配置特定用户权限能有效进行共享数据的资源管理。适用于轻量级数据处理、文件共享等业务不太频繁的场景。
- **SFS容量型文件系统（仅存量用户支持使用）**  
弹性文件服务（Scalable File Service, SFS）容量型文件系统为用户提供一个完全托管的共享文件存储，能够弹性伸缩至PB规模，具备高可用性和持久性，为海量数据、高带宽型应用提供有力支持。详情请参见[SFS产品介绍](#)。

## 约束与限制

- **当前挂载SFS容量型文件系统仅支持存量用户使用，请选择SFS Turbo文件系统、通用文件系统和ECS共享目录进行挂载。各文件系统支持挂载的区域请以控制台展示为准。**
- 请勿在初始化函数中执行挂载磁盘的读写操作，这将导致函数无法读取磁盘中的内容。
- 使用磁盘挂载功能需要文件系统开放端口：111、445、2049、2051、2052、20048。  
对于Ubuntu系统还需再开放3个端口，获取方式请在任意目录下执行如下命令。  

```
rpcinfo -plgrep mountd|grep tcp
```

  
具体请参见[弹性文件服务会占用用户的哪些资源](#)。

前提条件

表 7-3 磁盘挂载前提条件

| 文件系统来源                | 前提条件                                                                                                                                                                                                                                               |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 挂载SFS Turbo文件系统       | <ul style="list-style-type: none"><li>已创建待挂载的SFS Turbo文件系统，具体操作请参见<a href="#">创建文件系统</a>。<br/>创建SFS Turbo文件系统时请记录文件系统所使用的VPC和子网信息。</li><li>已创建挂载SFS Turbo文件系统所需的相关委托，具体操作请参见<a href="#">配置函数的委托权限</a>。</li></ul>                                   |
| 挂载通用文件系统              | <ul style="list-style-type: none"><li>已创建待挂载的通用文件系统，具体操作请参见<a href="#">创建文件系统</a>。<br/>创建通用文件系统时请记录文件系统所使用的VPC和子网信息。</li><li>已创建挂载通用文件系统所需的相关委托，具体操作请参见<a href="#">配置函数的委托权限</a>。</li></ul>                                                        |
| 挂载ECS共享目录             | <ul style="list-style-type: none"><li>已创建待挂载的ECS，具体操作请参见<a href="#">购买ECS</a>。<br/>创建ECS时请记录ECS所使用的VPC和子网信息。</li><li>已在ECS上创建NFS共享目录，具体操作请参考<a href="#">在ECS创建NFS共享目录</a>。</li><li>已创建挂载ECS共享目录所需的相关委托，具体操作请参见<a href="#">配置函数的委托权限</a>。</li></ul> |
| 挂载SFS容量型文件系统（仅存量用户使用） | <ul style="list-style-type: none"><li>已创建待挂载的SFS容量型文件系统，具体操作请参见<a href="#">创建文件系统</a>。</li><li>已创建挂载SFS容量型文件系统所需的相关委托，具体操作请参见<a href="#">配置函数的委托权限</a>。</li></ul>                                                                                  |

挂载 SFS Turbo 文件系统

- 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
- 单击函数的名称，进入函数详情页。
- 在“设置”页签下选择“权限”，配置函数访问SFS Turbo文件系统的委托。
  - “函数配置委托”参数选择[前提条件](#)中已创建的委托权限。
  - 单击“保存”完成配置。
- 在“设置”页签下选择“网络配置”，配置函数访问SFS Turbo文件系统的VPC网络。
  - 开启“函数访问VPC内资源”。
  - 配置“VPC”和“子网”为[前提条件](#)中记录的VPC和子网。
  - 单击“保存”完成配置。
- 在“设置”页签下选择“磁盘挂载”。
- 单击“添加挂载”，在弹窗中配置SFS Turbo文件系统的挂载信息。

表 7-4 挂载 SFS Turbo 文件系统参数说明

| 参数     | 说明                                                                                                                                                                                                                                         | 取值样例         |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|
| 用户ID   | 函数访问文件系统所需要的身份标识，取值范围为-1或1~65534的整数且不包含1000和1002，默认值为-1，表示由系统自动分配用户ID。<br>对于SFS Turbo文件系统，保持默认值-1。                                                                                                                                         | -1           |
| 用户组ID  | 函数访问文件系统所需要的身份标识，取值范围为-1或1~65534的整数且不包含1000和1002，默认值为-1，表示由系统自动分配用户组ID。<br>对于SFS Turbo文件系统，保持默认值-1。                                                                                                                                        | -1           |
| 文件系统来源 | 选择要挂载的文件系统类型，此处选择“SFS Turbo”，即SFS Turbo文件系统。                                                                                                                                                                                               | SFS Turbo    |
| 文件系统名称 | 选择要挂载的SFS Turbo文件系统名称。                                                                                                                                                                                                                     | sfs-turbo-fg |
| 共享目录路径 | 文件系统共享的目录路径，路径须以“/”开头，例如：/a。<br>不配置表示函数能访问文件系统所有的目录路径。如果配置具体目录路径，则表示函数只能访问该目录路径。                                                                                                                                                           | /a           |
| 函数访问路径 | 函数访问文件系统中文件时所使用的目录路径，路径须以“/”开头，且最多设置为两级，建议配置为/mnt或/home开头的二级目录。<br><b>不能使用系统已存在的目录，否则会报“failed to mount exist system path”错误。</b><br>例如，共享目录路径为“/a”，其下有文件“/a/b”和“/a/c/d”，函数访问路径为“/mnt/test”，则函数访问文件b和文件d的路径为“/mnt/test/b”和“/mnt/test/c/d”。 | /mnt/test    |

7. 单击“确定”，完成挂载配置。

挂载通用文件系统

当前仅“华北-乌兰察布一”区域支持挂载通用文件系统。

- 1. 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
- 2. 单击函数的名称，进入函数详情页。
- 3. 在“设置”页签下选择“权限”，配置函数访问通用文件系统的委托。
  - a. “函数配置委托”参数选择[前提条件](#)中已创建的委托权限。
  - b. 单击“保存”完成配置。
- 4. 在“设置”页签下选择“网络配置”，配置函数访问通用文件系统的VPC网络。
  - a. 开启“函数访问VPC内资源”。
  - b. 配置“VPC”和“子网”为[前提条件](#)中记录的VPC和子网。

- c. 单击“保存”完成配置。
5. 在“设置”页签下选择“磁盘挂载”。
6. 单击“添加挂载”，在弹窗中配置通用文件系统的挂载信息。

表 7-5 挂载通用文件系统参数说明

| 参数     | 说明                                                                                                                                                                                                                                         | 取值样例      |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| 用户ID   | 函数访问文件系统所需要的身份标识，取值范围为-1或1~65534的整数且不包含1000和1002，默认值为-1，表示由系统自动分配用户ID。<br>对于SFS Turbo文件系统，保持默认值-1。                                                                                                                                         | -1        |
| 用户组ID  | 函数访问文件系统所需要的身份标识，取值范围为-1或1~65534的整数且不包含1000和1002，默认值为-1，表示由系统自动分配用户组ID。<br>对于SFS Turbo文件系统，保持默认值-1。                                                                                                                                        | -1        |
| 文件系统来源 | 选择要挂载的文件系统类型，此处选择“通用文件系统”。                                                                                                                                                                                                                 | 通用文件系统    |
| 文件系统名称 | 选择要挂载的通用文件系统名称。                                                                                                                                                                                                                            | sfs-fg    |
| 函数访问路径 | 函数访问文件系统中文件时所使用的目录路径，路径须以“/”开头，且最多设置为两级，建议配置为/mnt或/home开头的二级目录。<br><b>不能使用系统已存在的目录，否则会报“failed to mount exist system path”错误。</b><br>例如，共享目录路径为“/a”，其下有文件“/a/b”和“/a/c/d”，函数访问路径为“/mnt/test”，则函数访问文件b和文件d的路径为“/mnt/test/b”和“/mnt/test/c/d”。 | /mnt/test |

7. 单击“确定”，完成挂载配置。

## 挂载 ECS 共享目录

1. 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
2. 单击函数的名称，进入函数详情页。
3. 在“设置”页签下选择“权限”，配置函数访问ECS共享目录的委托权限。
  - a. “函数配置委托”参数选择[前提条件](#)中已创建的委托。
  - b. 单击“保存”完成配置。
4. 在“设置”页签下选择“网络配置”，配置函数访问ECS共享目录的VPC网络。
  - a. 开启“函数访问VPC内资源”。
  - b. 配置“VPC”和“子网”为[前提条件](#)中记录的VPC和子网。
  - c. 单击“保存”完成配置。
5. 在“设置”页签下选择“磁盘挂载”。

6. 单击“添加挂载”，在弹窗中配置ECS共享目录的挂载信息。

表 7-6 挂载 ECS 共享目录参数说明

| 参数     | 说明                                                                                                                                                                                                                                                        | 取值样例      |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| 用户ID   | 函数访问文件系统所需要的身份标识，取值范围为-1或1~65534的整数且不包含1000和1002，默认值为-1，表示由系统自动分配用户ID。 <ul style="list-style-type: none"><li>对于使用Windows系统的ECS，保持默认值-1。</li><li>对于使用Linux系统的ECS，为系统中的uid。例如Linux系统的用户名为test-user，则可以在系统中使用<code>id test-user</code>命令查询uid和gid。</li></ul> | -1        |
| 用户组ID  | 函数访问文件系统所需要的身份标识，取值范围为-1或1~65534的整数且不包含1000和1002，默认值为-1，表示由系统自动分配用户组ID。 <ul style="list-style-type: none"><li>对于使用Windows系统的ECS，保持默认值-1。</li><li>对于使用Linux系统的ECS，为系统中的gid。例如Linux系统的用户名为test-user，则可以在系统中使用<code>id test-user</code>命令查询gid。</li></ul>    | -1        |
| 文件系统来源 | 选择要挂载的文件系统类型，此处选择“ECS”，即ECS共享目录。                                                                                                                                                                                                                          | ECS       |
| 云服务器名称 | 选择要挂载的ECS名称。                                                                                                                                                                                                                                              | ecs-fg    |
| 共享目录路径 | 文件系统共享的目录路径，请输入实际存在的路径，路径须以“/”开头，例如：/a。<br>不配置表示函数能访问文件系统所有的目录路径，如果配置具体目录路径，则表示函数只能访问该目录路径。                                                                                                                                                               | /a        |
| 函数访问路径 | 函数访问文件系统中文件时所使用的目录路径，路径须以“/”开头，且最多设置为两级，建议配置为/mnt或/home开头的二级目录。<br><b>不能使用系统已存在的目录，否则会报“failed to mount exist system path”错误。</b><br>例如，共享目录路径为“/a”，其下有文件“/a/b”和“/a/c/d”，函数访问路径为“/mnt/test”，则函数访问文件b和文件d的路径为“/mnt/test/b”和“/mnt/test/c/d”。                | /mnt/test |

7. 单击“确定”，完成挂载配置。

## 在 ECS 创建 NFS 共享目录

Linux系统（CentOS、SUSE、Euler OS、Fedora、OpenSUSE等）

- Linux系统（CentOS、SUSE、Euler OS、Fedora、OpenSUSE等）创建NFS共享目录操作步骤：

- a. 配置Yum源。
- i. 在/etc/yum.repos.d目录下创建文件euleros.repo（ 文件名称euleros可自定义，但文件扩展名必须以 “.repo” 结尾 ）。

ii. 执行以下命令编辑euleros.repo文件。  
vi /etc/yum.repos.d/euleros.repo

iii. 在euleros.repo文件中添加如下配置信息。  
Euler 2.0SP3版本yum添加的配置信息如下：  
[base]  
name=EulerOS-2.0SP3 base  
baseurl=http://repo.huaweicloud.com/euler/2.3/os/x86\_64/  
enabled=1  
gpgcheck=1  
gpgkey=http://repo.huaweicloud.com/euler/2.3/os/RPM-GPG-KEY-EulerOS  
  
Euler 2.0SP5版本yum添加的配置信息如下：  
[base]  
name=EulerOS-2.0SP5 base  
baseurl=http://repo.huaweicloud.com/euler/2.5/os/x86\_64/  
enabled=1  
gpgcheck=1  
gpgkey=http://repo.huaweicloud.com/euler/2.5/os/RPM-GPG-KEY-EulerOS

表 7-7 参数说明

| 名称       | 说明                                                                                                                                                           |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| name     | 仓库的名称。                                                                                                                                                       |
| baseurl  | 仓库的地址。 <ul style="list-style-type: none"><li>使用http协议的网络地址：http://path/to/repo</li><li>使用本地仓库地址：file:///path/to/local/repo</li></ul>                         |
| gpgcheck | 表示是否进行gpg（ GNU Private Guard ）校验，以确定RPM包来源的有效性和安全性。 <ul style="list-style-type: none"><li>1：表示进行gpg校验。</li><li>0：表示不进行gpg校验。</li><li>不配置：默认行gpg校验。</li></ul> |

- iv. 按 “Esc” 退出编辑模式，然后输入 “:wq” 并回车，保存euleros.repo文件并退出。

v. 执行以下命令清理缓存。  
yum clean all
- b. 执行以下命令安装nfs-utils。  
yum install nfs-utils
- c. 设置共享目录。
- i. 执行以下命令编辑/etc/exports文件。  
vi /etc/exports

ii. 在exports文件中添加如下配置信息。  
/sharedata 192.168.0.0/24(rw, sync, no\_root\_squash)  
  
/sharedata表示要共享的目录路径，192.168.0.0/24表示目录要共享给该网段的其他服务器。

- iii. 按“Esc”退出编辑模式，然后输入“:wq”并回车，保存exports文件并退出。
- d. 执行以下命令启动nfs服务。  

```
systemctl start rpcbind  
service nfs start
```
- e. 执行以下命令判断是否设置成功。  

```
exportfs -v
```

如果回显信息有打印刚配置的共享目录路径，表示设置成功。
- f. 修改共享目录。  
如果需要修改或新增共享目录，参考**c**修改配置文件，修改完成后执行以下命令重启nfs服务。  

```
service nfs restart
```
- g. 设置rpcbind开机启动（可选）  
如果需要设置rpcbind服务开机启动，可执行以下命令。  

```
systemctl enable rpcbind
```

#### Linux系统（Ubuntu）

- Linux系统（Ubuntu）创建NFS共享目录操作步骤：
  - a. 执行以下命令安装nfs-kernel-server。  

```
sudo apt-get update  
sudo apt install nfs-kernel-server
```
  - b. 设置共享目录。
    - i. 执行以下命令编辑/etc/exports文件。  

```
vim /etc/exports
```
    - ii. 在exports文件中添加如下配置信息。  

```
/sharedata 192.168.0.0/24(rw,sync,no_root_squash)
```

**/sharedata**表示要共享的目录路径，**192.168.0.0/24**表示目录要共享给该网段的其他服务器。
    - iii. 按“Esc”退出编辑模式，然后输入“:wq”并回车，保存exports文件并退出。
  - c. 执行以下命令启动nfs服务。  

```
service nfs-kernel-server restart
```
  - d. 执行以下命令判断是否设置成功。  

```
exportfs -v
```

如果回显信息有打印刚配置的共享目录路径，表示设置成功。
  - e. 修改共享目录。  
如果需要修改或新增共享目录，参考**b**修改配置文件，修改完成后执行以下命令重启nfs服务。  

```
service nfs restart
```
  - f. 设置rpcbind开机启动（可选）  
如果需要设置rpcbind服务开机启动，可执行以下命令。  

```
systemctl enable rpcbind
```

#### Windows系统

- Windows系统创建NFS共享目录操作步骤：  
请参考在Windows系统中部署NFS的[官方文档](#)进行NFS安装和文件共享。

#### Windows系统（haneWIN NFS Server软件）

- Windows系统（haneWIN NFS Server软件）创建NFS共享目录操作步骤：
  - 从[官网](#)获取haneWIN NFS Server软件。  
**注意，haneWIN NFS Server为收费软件。**
  - 以Windows系统管理员身份运行软件包中的“.exe”文件，安装haneWIN NFS Server。
  - 安装完成后打开“NFS Server”软件，选择“Edit > Preferences”。
  - 选择“Exports”页签，单击“Edit exports file”，配置共享目录，“NFS”、“Server”和“PortMapper”页签下的配置保持默认，完成后单击“Save”保存。  
配置共享目录的格式为“D:\share -public -name:nfs”，表示将“D:\share”文件夹权限设置为“public”，同时定义了一个别名“nfs”。
  - 单击“确定”完成配置。
  - 关闭所有的防火墙，包括“域网络”、“专用网络”和“公用网络”，等使用完后，再打开防火墙。

 说明

可以在Linux系统中执行如下命令挂载目录，验证文件共享是否成功。

```
mount -t nfs -o nolock 192.168.xxx.xxx:/nfs /mnt
```

- “192.168.xxx.xxx”是Windows系统的IP地址。
- “nfs”是配置共享目录时创建的别名。
- “/mnt”是指定将远程目录挂载到本地的“/mnt”目录下。

挂载 SFS 容量型文件系统（仅存量用户支持使用）

- 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
- 单击函数的名称，进入函数详情页。
- 在“设置”页签下选择“权限”，配置函数访问SFS容量型文件系统的委托权限。
  - “函数配置委托”参数选择[前提条件](#)中已创建的委托。
  - 单击“保存”完成配置。
- 在“设置”页签下选择“磁盘挂载”。
- 单击“添加挂载”，在弹窗中配置SFS容量型文件系统的挂载信息。

表 7-8 挂载 SFS 容量型文件系统参数说明

| 参数    | 说明                                                                                               | 取值样例 |
|-------|--------------------------------------------------------------------------------------------------|------|
| 用户ID  | 函数访问文件系统所需要的身份标识，取值范围为-1或1~65534的整数且不包含1000和1002，默认值为-1，表示由系统自动分配用户ID。<br>对于SFS容量型文件系统，保持默认值-1。  | -1   |
| 用户组ID | 函数访问文件系统所需要的身份标识，取值范围为-1或1~65534的整数且不包含1000和1002，默认值为-1，表示由系统自动分配用户组ID。<br>对于SFS容量型文件系统，保持默认值-1。 | -1   |

| 参数     | 说明                                                                                                                                                                                                                                         | 取值样例      |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| 文件系统来源 | 选择要挂载的文件系统类型，此处选择“SFS”，即SFS容量型文件系统。                                                                                                                                                                                                        | SFS       |
| 文件系统名称 | 选择要挂载的SFS容量型文件系统名称。                                                                                                                                                                                                                        | SFS-fg    |
| 共享目录路径 | 文件系统共享的目录路径，路径须以“/”开头，例如：/a。<br>不配置表示函数能访问文件系统所有的目录路径，如果配置具体目录路径，则表示函数只能访问该目录路径。                                                                                                                                                           | /a        |
| 函数访问路径 | 函数访问文件系统中文件时所使用的目录路径，路径须以“/”开头，且最多设置为两级，建议配置为/mnt或/home开头的二级目录。<br><b>不能使用系统已存在的目录，否则会报“failed to mount exist system path”错误。</b><br>例如，共享目录路径为“/a”，其下有文件“/a/b”和“/a/c/d”，函数访问路径为“/mnt/test”，则函数访问文件b和文件d的路径为“/mnt/test/b”和“/mnt/test/c/d”。 | /mnt/test |

6. 单击“确定”，完成挂载配置。

## 7.4 配置函数的环境变量

本章节介绍如何通过函数工作流控制台，为函数配置环境变量，便于调整函数的执行行为。

### 操作场景

为函数配置环境变量，可以在不修改代码的情况下，将动态参数传递到函数，调整函数的执行行为。

环境变量在函数中适用于以下场景：

- 区分多环境：相同的函数逻辑，可根据部署环境的不同，配置不同的环境变量以区分。例如，通过环境变量给测试和开发环境配置不同的数据库。
- 加密配置：函数中访问其他服务的认证信息，如账号和密码，可通过配置加密环境变量在代码中动态获取，确保敏感数据的安全。
- 动态配置：函数逻辑中需要动态调整的配置，如查询周期、超时时间，可提取为环境变量，以避免业务每次变化都需要修改代码。

### 约束与限制

设置环境变量时，FunctionGraph会明文展示所有输入信息，请不要输入敏感信息（如账户密码等），以防止信息泄露。

配置环境变量

1. 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
2. 单击函数的名称，进入函数详情页。
3. 在“设置”页签下选择“环境变量”，单击“编辑环境变量”。

图 7-2 添加环境变量



4. 在“编辑环境变量”弹窗中单击“添加环境变量”，配置环境变量信息。  
使用表单编辑
- 使用表单编辑环境变量的参数说明请参考[表7-9](#)。

表 7-9 环境变量配置参数说明

| 参数   | 说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 键    | 环境变量的键（key）。<br>取值范围：以大小写字母开头，可包含大小写字母、数字、下划线（_）、中划线（-）和点（.），键和值的总长度不超过4096个字符。<br>系统预置了部分环境变量配置，您无法配置和预置值同名的环境变量，预置的环境变量请参见 <a href="#">环境变量预置值</a> 。                                                                                                                                                                                                                                                                                                                                    |
| 值    | 环境变量的值（value）。<br>取值范围：键和值的总长度不超过4096个字符。                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 加密参数 | 开启加密参数后，环境变量的值将以“*”号加密显示，参数传输过程中键值也处于加密状态。<br>已开启加密并保存的环境变量值，取消加密后将重置为空。<br><div><div>图 7-3 使用表单编辑环境变量</div><div><div>编辑环境变量</div><div><div>AES加密并保存的环境变量值取消加密将重置为空。<br/>JSON格式下为键添加前缀_encrypt_表示该键为AES加密变量。</div><div><div>使用表单编辑</div><div>使用 JSON 格式编辑</div></div><div><div>键</div><div>值</div><div>加密参数</div></div><div><div>obs_output_bucket</div><div>*****</div><div><input checked="" type="checkbox"/></div><div>删除</div></div><div><div>+ 添加环境变量</div></div></div></div></div> |

| 参数    | 说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 传输中加密 | <p>此参数仅“拉美-圣保罗一”区域支持配置。</p> <p><b>注意</b></p> <p>数据加密服务DEW将按需收取费用，详情请参见<a href="#">数据加密服务计费说明</a>。</p> <p>单击“加密”按钮为环境变量加密，支持以下加密类型：</p> <ul style="list-style-type: none"><li>• AES加密：使用AES加密环境变量，无需手动创建密钥。</li><li>• KMS加密：支持选择已创建的KMS密钥加密函数环境变量。</li></ul> <p><b>前提条件</b></p> <p>使用KMS加密环境变量前，需向函数授予调用Decrypt操作的权限，请参考<a href="#">JSON视图配置自定义策略</a>前往IAM控制台将“委托权限策略”添加至函数委托中，并参考<a href="#">配置函数委托</a>为该函数配置委托权限。</p> <p><b>选择KMS密钥</b></p> <p>请选择已启用的自主管理型密钥，如无可用密钥，可参考创建用户主密钥<a href="#">创建自定义密钥</a>进行创建。</p> <p><b>注意</b></p> <p>在后续使用中请勿在DEW服务中删除用于函数加密的主密钥，否则将因加密数据无法解密而导致函数执行失败。</p> <p><b>解密环境变量</b></p> <p>使用KMS加密环境变量后，需在函数代码中添加“解密密钥代码段”用以解密环境变量，请注意将代码段中的 {endpoint} 替换为对应区域的<a href="#">终端节点</a>。</p> <p><b>图 7-4 传输中加密</b></p> <div><p>传输中加密</p><p>加密类型</p><p><input type="radio"/> AES 加密 <input checked="" type="radio"/> KMS 加密</p><p>要在传输中加密的 KMS 密钥</p><div><div></div><div>查看密钥</div></div><p>选择已启用的密钥或创建新的 KMS 密钥</p><div><p>要使用KMS加密的环境变量，您的函数需要访问用于加密的密钥。要向您的函数授予调用 Decrypt 操作的权限，请将以下策略添加到函数委托。在您的函数中添加代码以解密环境变量。请参阅“解密密钥代码段”示例。 <a href="#">更多代码示例</a></p><p>委托权限策略</p><p>解密密钥代码段</p></div></div> <p>开启加密参数后，环境变量的值将以“*”号加密显示，参数传输过程中键值也处于加密状态。</p> <p>已开启加密并保存的环境变量值，取消加密后将重置为空。</p> |

使用JSON格式编辑

- 使用JSON格式编辑环境变量的示例说明。

```
{  "obs_output_bucket": "obs_image"}
```

如需使用AES加密环境变量，需为键添加前缀 “\_encrypt\_” 。

```
{
  "_encrypt_obs_output_bucket": "obs_image"
}
```

5. （可选，仅“拉美-圣保罗—”区域支持配置）选择为环境变量是否使用“静态加密配置”。
- 如选择使用KMS静态加密，请先将“委托权限策略”添加至该函数的委托中。支持选择以下两种KMS静态加密：
- （默认）functiongraph/default：函数会在DEW服务中自动创建默认密钥，您无需手动创建。

- 使用用户主密钥：选择已创建的用户主密钥加密函数代码。创建用户主密钥可参考[创建自定义密钥](#)。

 **注意**

若选择“使用用户主密钥”，在后续使用中请勿在DEW服务中删除用于函数加密的主密钥，否则将因加密数据无法解密而导致函数执行失败。

6. 配置完成后，单击“确定”保存环境变量。
- 环境变量配置完成后，即可在函数中使用环境变量，具体应用示例可参见[环境变量应用示例](#)。

环境变量预置值

FunctionGraph预置的环境变量如[表7-10](#)所示。

表 7-10 环境变量预置值说明

| 环境变量名                | 含义                           | 获取方式                     |
|----------------------|------------------------------|--------------------------|
| RUNTIME_PROJECT_ID   | 函数的项目ID。                     | Context类提供接口或通过系统环境变量获取。 |
| RUNTIME_FUNC_NAME    | 函数名称。                        | Context类提供接口或通过系统环境变量获取。 |
| RUNTIME_FUNC_VERSION | 函数版本。                        | Context类提供接口或通过系统环境变量获取。 |
| RUNTIME_HANDLER      | 函数执行入口。                      | 通过系统环境变量获取。              |
| RUNTIME_TIMEOUT      | 函数执行的超时时间。                   | 通过系统环境变量获取。              |
| RUNTIME_USERDATA     | 用户通过环境变量传入的值。                | Context类提供接口或通过系统环境变量获取。 |
| RUNTIME_CPU          | 函数占用的CPU资源，取值与MemorySize成比例。 | Context类提供接口或通过系统环境变量获取。 |
| RUNTIME_MEMORY       | 函数配置的内存大小，单位MB。              | Context类提供接口或通过系统环境变量获取。 |

| 环境变量名                       | 含义                                       | 获取方式        |
|-----------------------------|------------------------------------------|-------------|
| RUNTIME_MAX_RESP_BODY_SIZE  | 最大返回值限制，系统默认值为6,291,456 Byte。            | 通过系统环境变量获取。 |
| RUNTIME_INITIALIZER_HANDLER | 函数初始化入口。                                 | 通过系统环境变量获取。 |
| RUNTIME_INITIALIZER_TIMEOUT | 函数初始化超时时间。                               | 通过系统环境变量获取。 |
| RUNTIME_ROOT                | Runtime包的路径，系统默认路径为/home/snuser/runtime。 | 通过系统环境变量获取。 |
| RUNTIME_CODE_ROOT           | 代码在容器中的存放目录，系统默认路径为/opt/function/code    | 通过系统环境变量获取。 |
| RUNTIME_LOG_DIR             | 系统日志在容器中存放的目录，系统默认路径为/home/snuser/log。   | 通过系统环境变量获取。 |

环境变量应用示例

使用环境变量设置以下信息：安装文件的目录、存储输出的位置、存储连接和日志记录设置等。这些设置与应用程序逻辑解耦，在需要变更设置时，无需更新函数代码。

1. 设置环境变量obs\_output\_bucket，用于灵活设置存储输出图片的OBS桶。

图 7-5 环境变量



2. 在函数代码中使用环境变量。
- 在如下函数代码片段中，参数“obs\_output\_bucket”为图片处理后存储地址。

说明

- 非HTTP函数使用context.getUserData('xxx')方式获取环境变量。
- HTTP函数使用系统方法获取环境变量，例如Python函数使用os.Environ['xx']; Node.js函数使用process.env.xx，Node.js获取加密环境变量使用process.env.RUNTIME\_USERDATA。

Python语言示例

```
def handler(event, context):
    srcBucket, srcObjName = getObsObjInfo4OBSTrigger(event)
    obs_address = context.getUserData('obs_address')
    outputBucket = context.getUserData('obs_output_bucket')
    if obs_address is None:
        obs_address = '{obs_address_ip}'
    if outputBucket is None:
        outputBucket = 'casebucket-out'
```

```
ak = context.getSecurityAccessKey()
sk = context.getSecuritySecretKey()
st = context.getSecurityToken()

# download file uploaded by user from obs
# TODO: Replace with actual implementation
GetObject(obs_address, srcBucket, srcObjName, ak, sk, st)

outFile = watermark_image(srcObjName)

# 将转换后的文件上传到新的obs桶中
# TODO: Replace with actual implementation
PostObject(obs_address, outputBucket, outFile, ak, sk, st)

return 'OK'
```

#### – NodeJS语言示例

```
exports.handler = async (event, context) => {
  let bucket = context.getUserData('obs_output_bucket');
  console.log(bucket);
  const output =
  {
    'statusCode': 200,
    'headers':
    {
      'Content-Type': 'application/json'
    },
    'isBase64Encoded': false,
    'body': JSON.stringify(event),
  }
  return output;
}
```

## 相关文档

- 除使用控制台外，函数工作流支持通过API的方式管理函数的生命周期，详情请参见[函数生命周期管理API](#)。
- 函数环境变量相关常见问题详情请参见[配置函数常见问题](#)。

## 7.5 配置函数的异步调用策略

在异步调用函数的场景下，FunctionGraph会将请求排队，不等待函数的执行结果，直接向客户端返回响应，这种情况下客户端无法实时感知函数的执行结果。如果需要设置异步请求失败重试或者获取异步请求的结果通知，可以通过设置异步配置实现。

- 失败重试：用户代码异常造成的失败，FunctionGraph默认不重试。如果函数中有需要重试的场景，例如调用三方服务经常失败，可配置重试提升成功率。
- 结果通知：FunctionGraph可以将异步函数的执行结果传递给下游服务做进一步处理。例如把执行失败信息保存到OBS，用于后续分析失败原因，或者把执行成功信息推送到DIS或再次触发函数做处理等。

约束与限制

表 7-11 函数异步配置约束限制

| 限制场景                         | 具体限制                                                                                                          |
|------------------------------|---------------------------------------------------------------------------------------------------------------|
| 使用异步调用的函数                    | 使用异步调用的函数，返回值的大小不能超过256KB，否则会返回空值。                                                                            |
| 使用异步配置通知目标服务                 | 请务必保证不要出现循环调用的情况。<br>例如：函数A配置了成功调用时的异步通知目标为函数B，函数B配置了成功调用时的异步通知目标为函数A，当异步触发函数A并且执行成功后，则可能出现A>B>A.....循环调用的情况。 |
| 使用异步配置失败时通知分布式消息服务（RocketMQ） | 仅支持“华东-上海一”、“西南-贵阳一”区域配置。                                                                                     |

前提条件

- 使用异步配置通知目标服务，需配置具有目标服务操作权限的函数委托，配置函数委托详情请参考[配置函数的委托权限](#)。
- 使用异步配置失败时通知分布式消息服务（RocketMQ），需要[配置函数访问VPC](#)且与通知的RocketMQ实例在同一个VPC中。  
同时，需要确保RocketMQ实例所在的安全组入方向规则已放通实例的连接端口，否则无法通知消息，可参考[连接RocketMQ网络要求](#)。（当前处于公测阶段，仅支持华东-上海一、西南-贵阳一。）

配置异步调用策略

- 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
- 单击函数的名称，进入函数详情页。
- 在“设置”页签下选择“异步配置”。
- 在异步配置页面，单击“配置异步调用”，编辑异步配置信息。

图 7-6 填写配置参数

异步策略配置

最大重试次数

-

3

+

取值范围：0-3

异步调用失败后的最大重试次数，单位：次

消息最大有效期(s)

-

3,600

+

取值范围：1-86,400

消息最大存活时长，单位：秒

成功时通知

执行成功时发送通知到以下目标

目标服务

函数工作流 (FunctionGraph)

函数名称

shili3

创建函数

版本/别名

latest

失败时通知

执行失败时发送通知到以下目标

目标服务

函数工作流 (FunctionGraph)

函数名称

shili3

创建函数

版本/别名

latest

表 7-12 异步配置参数说明

| 参数     | 说明                                                                                                                                                                                       |
|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 异步策略配置 | <div><div>最大重试次数：异步调用失败后最大重试次数，默认为3次，取值范围：0-3。</div><div>消息最大有效期(s)：消息最大存活时长，取值范围：1-86400秒。</div><div>开启异步调用状态持久化：开通云日志服务（LTS）后显示该配置开关，具体配置方式请参考<a href="#">配置异步调用状态持久化</a>。</div></div> |

| 参数    | 说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 成功时通知 | <p>目标服务：函数执行成功时发送通知到以下目标服务。</p> <ul style="list-style-type: none"><li>● 函数工作流（FunctionGraph）<ul style="list-style-type: none"><li>- 函数名称：选择接收通知消息的函数名称。</li><li>- 版本/别名：选择函数对应的版本或别名。</li></ul></li><li>● 对象存储服务（OBS）<ul style="list-style-type: none"><li>- 目标桶目录：选择存放通知消息的OBS桶路径。</li><li>- 对象前缀目录：输入对象前缀目录，以筛选通知消息要存放的对象文件。</li><li>- 对象过期时间(天)：过期后的对象会被OBS服务端自动清理。单位：天，取值范围：0-365，0表示不过期。</li></ul></li><li>● 数据接入服务（DIS）<br/>目标通道：选择接收通知消息的目标通道。</li><li>● 消息通知服务（SMN）<br/>主题名称：选择接收通知消息的主题。</li></ul>                                                                                                                                                                       |
| 失败时通知 | <p>目标服务：函数执行失败时发送通知到以下目标服务。</p> <ul style="list-style-type: none"><li>● 函数工作流（FunctionGraph）<ul style="list-style-type: none"><li>- 函数名称：选择接收通知消息的函数名称。</li><li>- 版本/别名：选择函数对应的版本或别名。</li></ul></li><li>● 对象存储服务（OBS）<ul style="list-style-type: none"><li>- 目标桶目录：选择存放通知消息的OBS桶路径。</li><li>- 对象前缀目录：输入对象前缀目录，以筛选通知消息要存放的对象文件。</li><li>- 对象过期时间(天)：过期后的对象会被OBS服务端自动清理。单位：天，取值范围：0-365，0表示不过期。</li></ul></li><li>● 数据接入服务（DIS）<br/>目标通道：选择接收通知消息的目标通道。</li><li>● 消息通知服务（SMN）<br/>主题名称：选择接收通知消息的主题。</li><li>● 分布式消息服务 RocketMQ版（RocketMQ）（仅支持华东-上海一、西南-贵阳一区域配置。）<ul style="list-style-type: none"><li>- 实例：选择接收通知消息的RocketMQ实例。</li><li>- Topic：选择接收通知消息的Topic。</li></ul></li></ul> |

#### 5. 单击“确定”保存异步配置。

## 通知消息结构说明

FunctionGraph向目标发送异步调用通知的消息示例如下所示。

```
{
  "timestamp": "2020-08-20T12:00:00.000Z+08:00",
  "request_context": {
    "request_id": "1167bf8c-87b0-43ab-8f5f-26b16c64f252",
    "function_urn": "urn:fss:xx-xxx-x:xxxxxxx:function:xxxx:xxxx:latest",
  }
}
```

```
    "condition": "",
    "approximate_invoke_count": 0
  },
  "request_payload": "",
  "response_context": {
    "status_code": 200,
    "function_error": ""
  },
  "response_payload": "hello world!"
}
```

表 7-13 消息参数说明

| 参数                                           | 说明                                |
|----------------------------------------------|-----------------------------------|
| timestamp                                    | 函数开始执行的时间。                        |
| request_context                              | 请求上下文。                            |
| request_context.request_id                   | 异步调用的请求ID。                        |
| request_context.function_urn                 | 异步执行的函数URN。                       |
| request_context.condition                    | 调用错误类别。                           |
| request_context.<br>approximate_invoke_count | 异步调用的执行次数。当该值大于1时，说明函数的执行进行了重试。   |
| request_payload                              | 请求函数的原始负载。                        |
| response_context                             | 返回上下文。                            |
| response_context.statusCode                  | 调用函数的系统返回码。当该返回码不为200时，说明出现了系统错误。 |
| response_context.function_err<br>or          | 调用错误信息。                           |
| response_payload                             | 执行函数返回的原始负载。                      |

配置异步调用状态持久化

异步调用状态持久化可以提供更好的可观测性，开启后函数的异步调用日志会上报到 LTS，通过日志可以查询每次异步调用的执行状态。

说明

用户默认没有权限开启异步调用状态持久化，如果需要开启异步调用状态持久化，查看调用记录，请在[工单系统](#)提交工单添加白名单。

1. 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。

2. 单击函数的名称，进入函数详情页，在“设置”页签下选择“异步配置”。

3. （如已开通云日志服务，可跳过该步骤）在异步配置页面，单击页面上方说明信息中的“点击开通”，开通云日志服务（LTS）。
- 此过程中系统会自动调用LTS接口，在LTS控制台创建日志组（名称为function-async-log-group开头的日志组）和日志流（名称为function-async-log-stream开头的日志流），用户无需手动创建日志组和日志流。

4. 开通完成后，在页面上单击“配置异步调用”，页面上新增参数“开启异步调用状态持久化”。
5. 启用“开启异步调用状态持久化”参数，并选择3中自动创建的日志组和日志流。

图 7-7 开启异步调用状态持久化



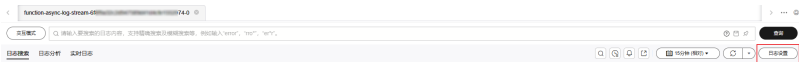
6. 在LTS服务配置日志结构化。
- a. 登录云日志服务LTS控制台，左侧导航栏选择“日志管理”，在日志组列表模块可查看自动创建的日志组和日志流。

图 7-8 日志组列表



- b. 单击日志流名称，进入日志流详情页。
- c. 在日志流详情页，如图7-9所示单击右上角的日志设置。

图 7-9 日志设置



- d. 在设置弹窗中，选择“云端结构化解析”页签，如图7-10所示配置日志结构化。
- 场景：选择“分隔符”。

■ 选择示例日志：填入“x|x|0|0|x”。

此处表示占位符，代表前2个字段为string类型，中间3个字段为long类型，最后1个字段为string类型。

■ 指定分隔符：“|”。

图 7-10 云端结构化解析

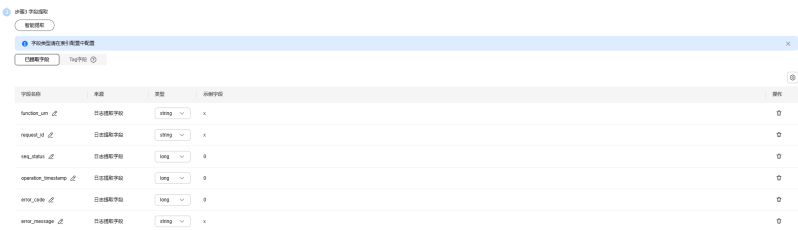


- e. 单击步骤3的“智能提取”智能生成字段，并单击 参考表7-14修改字段名称，修改完成后如图7-11所示。

表 7-14 修改字段

| 字段名称   | 修改后字段               |
|--------|---------------------|
| field1 | function_urn        |
| field2 | request_id          |
| field3 | seq_status          |
| field4 | operation_timestamp |
| field5 | error_code          |
| field6 | error_message       |

图 7-11 修改提取字段



- f. 修改完成后单击“保存”。

相关文档

- 除使用控制台外，函数工作流支持通过API的方式管理函数的异步配置，详情请参见[函数异步配置API](#)。
- 若修改异步配置时报“用户权限不足”，请添加“FunctionGraph Administrator”权限，具体操作方法请参见[创建用户并授权使用FunctionGraph](#)。

## 7.6 配置函数的并发处理

### 函数并发概述

默认情况下，每个函数实例同一时刻只处理一个请求。函数收到并发请求时，例如并发三个请求，FunctionGraph会启动三个函数实例处理请求。FunctionGraph提供了单实例多并发能力，可以在一个实例上并发处理多个请求。

函数的单实例多并发适合处理逻辑中有较长时间等待下游服务响应的场景，也适合函数逻辑中初始化时间较长的场景。

单实例多并发具备以下优势：

- 降低冷启动概率，优化函数处理时延：例如并发三个请求，不配置单实例多并发，FunctionGraph默认启动三个实例处理请求，会有三次冷启动。若配置了单实例支持三并发，三个并发请求，FunctionGraph只启动一个实例处理请求，减少了两次冷启动。
- 减少总请求处理时长，节省费用：单实例单并发下，多个请求的总处理时长为每个请求的处理时长相加。单实例多并发下，同一个实例对并发的多个请求的计费时间为，从第一个请求开始处理计时，到最后一个并发的请求处理结束计一次时长费用。

FunctionGraph会根据实际的请求情况自动弹性伸缩函数实例，并发变高时，会分配更多的函数实例来处理请求，并发减少时，相应的实例也会变少。

函数实例数=函数并发数/该函数的单实例并发数。

- 函数并发数：指某一刻该函数同时执行的请求数。
- 该函数的单实例并发数：指单个实例最多允许的函数并发数，即函数并发配置界面的“单实例并发数”。

### 单实例单并发与单实例多并发的对比

当一个函数执行需要花费5秒，若配置为单实例单并发，三次函数调用请求分别在三个函数实例执行，总执行时长为15秒。

若配置为单实例多并发，设置单实例并发数为5，即单个实例最多支持5个并发请求，如果有三次函数调用请求，将在一个实例内并发处理，总执行时间为5秒。

#### 说明

单实例并发数大于1，在您设置的“单函数最大实例数”范围内，超过单实例并发处理能力时会自动扩容新实例。

表 7-15 单并发与多并发对比

| 对比项  | 单实例单并发     | 单实例多并发                                                                                                                                                                                                                                                                                              |
|------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 日志打印 | 不涉及。       | Node.js使用console.info()函数，Python使用print()函数，Java使用System.out.println()函数打印日志，该方式会把当前请求的Request ID包含在日志内容中。<br><br>当多请求在同一个实例并发处理时，当前请求可能有很多个，继续使用这些函数打印日志会导致Request ID错乱。此时应该使用context.getLogger()，获取一个日志输出对象，通过这个日志输出对象打印日志。<br>例如Python：<br><pre>log = context.getLogger() log.info("test")</pre> |
| 共享变量 | 不涉及。       | 单实例多并发处理时，修改共享变量会导致错误。这要求您在编写函数时，对于非线程安全的变量修改要进行互斥保护。                                                                                                                                                                                                                                               |
| 监控指标 | 按实际情况进行监控。 | 相同负载下，函数的实例数明显减少。                                                                                                                                                                                                                                                                                   |
| 流控错误 | 不涉及。       | 请求数超过实例可处理能力时，FunctionGraph会对请求进行流控。Body中的ErrorCode为“FSS.0429”，响应头中的Status为429，错误信息提示：Your request has been controlled by overload sdk, please retry later。                                                                                                                                         |

约束与限制

- 该功能仅FunctionGraph v2版本支持。
- 当前仅通过空白函数和函数模板创建的HTTP函数，以及通过容器镜像创建的函数支持“单实例并发数”配置。
  - 如果其他方式创建的存量函数仍有“单实例并发数”配置，当该配置修改为1时，后续将不再支持修改。
  - 其他方式创建的函数如需使用单实例多并发功能，请[提交工单](#)申请白名单开通。
- 对于Python函数，由于Python GIL锁导致实例上的线程被绑定到一个核上，造成多并发无法使用多核，即使配置更大资源规格也无法提升函数处理性能。
- 对于Node.js函数，由于V8引擎的单进程单线程，造成多并发无法使用多核，即使配置更大资源规格也无法提升函数处理性能。

配置单实例多并发

- 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
- 单击函数的名称，进入函数详情页。
- 在“设置”页签下选择“并发”。
- 在并发页面设置函数的并发配置。

图 7-12 并发基础配置

基础配置

单实例并发数 ⓘ

1

函数支持一个实例同时并发执行多个请求，这个值用来配置单个函数实例可以同时处理多少个请求。

单函数最大实例数

400

单个函数最多运行在多少个按量实例中，最大值1000；-1表示不限制实例数

表 7-16 并发参数说明

| 参数       | 说明                                                                                                                                                                                                                                                                                       |
|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 单实例并发数   | <p>单个函数实例支持的请求并发数。</p> <p><b>约束说明：</b></p> <ul style="list-style-type: none"><li>当前仅通过空白函数和函数模板创建的“HTTP函数”，以及通过容器镜像创建的函数支持“单实例并发数”参数配置，其他方式创建的函数无该配置参数。</li><li>若该参数已设置为“1”，则后续将不再展示该参数，并且无法再次修改；若该参数设置大于1，则会继续展示该参数。</li></ul> <p><b>取值范围：</b><br/>1~1000</p> <p><b>默认取值：</b><br/>1</p> |
| 单函数最大实例数 | <p>单个函数最大可启用的按量实例数量。</p> <p><b>约束说明：</b></p> <ul style="list-style-type: none"><li>超过实例数处理能力的请求会被直接丢弃，而不是重试。</li><li>超过实例数处理能力导致的请求错误不会直接显示在函数日志中，可通过<a href="#">配置函数的异步调用策略</a>来获取错误详细信息。</li></ul> <p><b>取值范围：</b><br/>-1或1~1000的整数，-1表示不限制实例数。</p> <p><b>默认取值：</b><br/>400</p>        |

5. 完成后单击“保存”保存并发配置。

7.7 配置函数的版本

函数在开发、测试到生产全生命周期中支持发布多个版本，实现对函数代码的迭代管理。每个版本是函数某一时刻的完整快照，可对应代码里的Tag，包含独立的代码、配置及依赖信息，便于回滚、并行部署或功能验证。

函数创建以后，默认版本为latest版本，每个函数都有一个latest版本。

函数版本发布后，可根据实际需要修改部分版本配置信息，但版本的代码无法更新，以保证版本的稳定性和可追溯性。

约束与限制

- 单个函数下您最多可以发布20个版本（含latest版本）。
- 新发布的版本默认不绑定触发器，不会挂载磁盘，不带预留实例。
- latest版本不能被删除。
- 如果函数版本关联了别名，则删除版本时会把关联的别名删除。
- 删除版本将永久删除关联的代码、配置、别名及事件源映射，但不会删除日志。删除操作无法恢复，请谨慎操作。

发布新版本

1. 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
2. 单击函数的名称，进入函数详情页。
3. 在“版本”页签下，单击“发布新版本”，新版本将以latest版本内容发布。

图 7-13 发布新版本参数配置

版本号

请输入版本号

非必填项，若不指定则默认按时间生成版本号，如v20180116-202722  
可包含字母、数字、中划线、下划线和点，长度不超过42个字符，以字母/数字开头和结尾

描述

请输入版本描述，不超过512个字符

0/512

表 7-17 版本配置参数说明

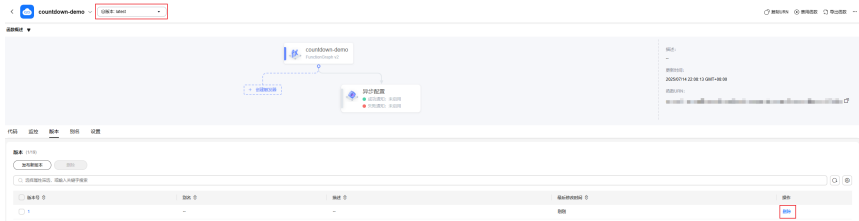
| 参数  | 说明                                                                                                      |
|-----|---------------------------------------------------------------------------------------------------------|
| 版本号 | 自定义版本号，用于区分不同的版本。<br>可包含字母、数字、中划线、下划线和点，长度不超过42个字符，以字母或数字开头和结尾。当版本号未设置时，系统以时间生成版本号，例如：v20220510-190658。 |
| 描述  | 版本的描述信息，长度不超过512个字符。                                                                                    |

4. 完成后单击“确定”，系统自动完成版本发布，当前函数版本也会切换至新创建的版本。

删除函数版本

1. 在函数详情页的最上方选择“latest”版本。
2. 在“版本”页签下，可看到已创建的版本列表。
3. 单击版本右侧的“删除”，在弹窗中输入“DELETE”后单击确定，可删除版本。

图 7-14 删除版本



4. 单击弹框中的“确认”，删除函数版本。

相关文档

除使用控制台外，函数工作流支持通过API的方式管理函数版本，详情请参见[函数版本别名API](#)。

7.8 配置函数的别名

FunctionGraph支持为函数创建别名，并关联指定版本的函数，通过别名调用函数时，即可调用到指定版本的函数。在实际业务场景下，可以通过修改在别名上配置的版本，实现函数版本的更新和回滚，客户端无感知。

一个别名支持配置最多2个函数版本，并为两个版本分配不同的权重，实现灰度发布。

约束与限制

- 单个函数最多可以创建10个别名。
- 一个别名支持配置最多2个函数版本。

创建别名

1. 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
2. 单击函数的名称，进入函数详情页。
3. 在“别名”页签下，单击“创建别名”。创建指向现有版本的别名，可通过该别名来调用对应版本的函数。

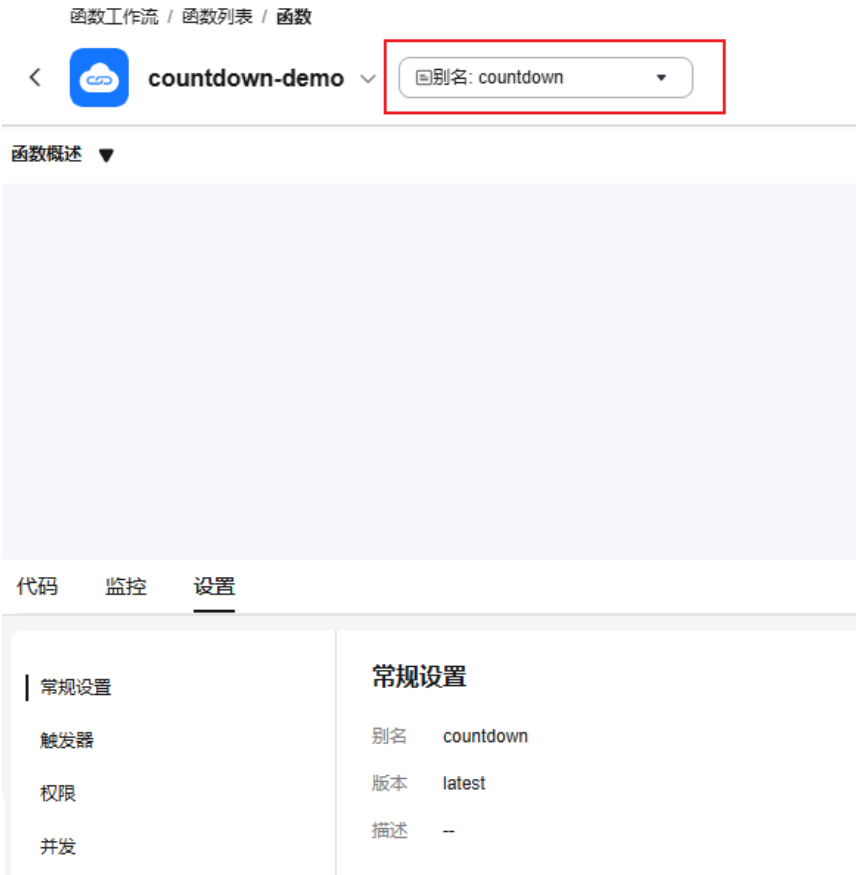
表 7-18 别名配置参数说明

| 参数     | 说明                                                                  |
|--------|---------------------------------------------------------------------|
| 别名名称   | 自定义的别名名称，用于区分不同的别名。以大写或小写字母开头，以字母或数字结尾，可包含字母、数字、下划线和中划线，长度不超过63个字符。 |
| 对应版本   | 选择需要关联的函数版本。仅可以为每个版本创建1个别名。                                         |
| 开启灰度版本 | 选择是否开启灰度版本，开启灰度版本后，一个别名可以同时关联两个版本，根据设置的权重比例，函数切换部分主版本的请求到灰度版本运行。    |

| 参数   | 说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 灰度版本 | <p>仅“开启灰度版本”开启时需要配置。</p> <p>选择需要关联的灰度版本，latest版本不能作为灰度版本。如需创建版本可参考<a href="#">配置函数的版本</a>。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 灰度类型 | <p>仅“开启灰度版本”开启时需要配置。</p> <p>选择灰度版本的流量切换类型。</p> <ul style="list-style-type: none"> <li>按百分比随机灰度：发给主版本的请求中，根据权重值把对应比例的请求切换到灰度版本。比如权重值为5%，则会将5%的主版本请求切到灰度版本，95%的请求继续发给主版本。</li> <li>按指定规则灰度：把满足设定规则的请求切换到灰度版本，该模式仅支持HTTP函数或使用APIG触发器的函数。</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                   |
| 权重   | <p>仅“开启灰度版本”开启，且“灰度类型”选择“按百分比随机灰度”时需要配置。</p> <p>发给主版本的请求中，切到灰度版本的比例。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 规则模式 | <p>仅“开启灰度版本”开启，且“灰度类型”选择“按指定规则灰度”时需要配置。</p> <p>选择满足规则条件的模式。</p> <ul style="list-style-type: none"> <li>满足下列所有规则：只有该请求满足规则列表中的所有规则时，才会发送给灰度版本。</li> <li>满足下列任意规则：只要请求满足规则列表中的任意一条规则，就会发送给灰度版本。</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 规则列表 | <p>仅“开启灰度版本”开启，且“灰度类型”选择“按指定规则灰度”时需要配置。</p> <p>设置函数请求要满足的规则条件：</p> <ul style="list-style-type: none"> <li>参数类型：默认为“Header”，即HTTP请求中的Header参数。</li> <li>参数：HTTP请求中Header参数的名称，不区分大小写。</li> <li>条件：Header参数要满足的条件类型，有“=”和“in”两种条件类型。</li> <li>值：Header参数的值，字符串类型。 <ul style="list-style-type: none"> <li>条件为“=”时，表示请求中Header参数的值等于设定值时才满足条件。</li> <li>条件为“in”时，可以设置多个值，并用英文逗号(,)隔开，表示Header参数的值满足其中一个值即满足条件。</li> </ul> </li> </ul> <p>例如，规则1中“参数”为“aaa”，“条件”设置为“=”，“值”设置为“123”，规则2中“参数”为“bbb”，“条件”设置为“in”，“值”设置为“111,222,333”。如果HTTP请求中有Header参数aaa=123，则满足规则1；如果HTTP请求中有Header参数bbb=222，则满足规则2；如果HTTP请求中有Header参数aaa=123和bbb=111，则同时满足规则1和规则2。</p> |
| 描述   | <p>别名的描述信息，长度不超过512个字符。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |

4. 单击“确定”，完成别名的创建，将自动跳转到别名函数的详情页。

图 7-15 创建别名

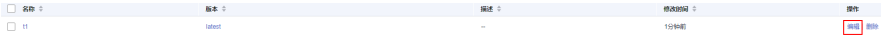


管理函数别名

1. 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
2. 单击函数的名称，进入函数详情页。
3. 在函数详情页的最上方选择“latest”版本。
4. 在“别名”页签下，可看到已创建的别名列表。

- 单击别名右侧的“编辑”，可修改别名的配置信息。

图 7-16 修改别名



- 单击别名右侧的“删除”，在弹窗中单击确定，可删除别名。

图 7-17 删除别名



相关文档

除使用控制台外，函数工作流支持通过API的方式管理函数版本，详情请参见[函数版本别名API](#)。

7.9 配置函数的标签

标签用于标识和分类函数资源，标签与函数资源间为多对多关系。为函数添加标签后，可通过标签快速检索和定位函数，便于系统化管理资源。

为函数配置标签，支持以下两种标签来源。

- 使用标签管理服务（TMS）创建的全局资源标签，支持在TMS服务中通过标签筛选函数资源。
- 使用FunctionGraph创建标签，支持在函数列表中通过标签筛选函数。

约束与限制

- 一个函数最多可以添加20个标签。
- 如您的组织已经设定函数工作流服务的相关标签策略，则需按照标签策略规则为函数添加标签。标签如果不符合标签策略的规则，则可能会导致函数创建失败，请联系组织管理员了解标签策略详情。

前提条件

如需使用标签管理服务（TMS）提供的预定义标签能力，请开通TMS服务，TMS服务开通具体请参见[权限管理](#)。

添加标签

1. 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
  2. 单击函数的名称，进入函数详情页。
  3. 在“设置”页签下选择“标签”。
  4. 单击“编辑标签”，在编辑标签弹框中单击“添加标签”。
  5. 配置标签键和标签值。
- 每个标签由一对键值对（Key-Value）组成，且每个标签键（Key）都必须是唯一的，每个标签键（Key）只能有一个值（Value）。

表 7-19 标签命名规则

| 参数  | 规则                                                                                                                                                                            |
|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 标签键 | <ul style="list-style-type: none"><li>• 不能为空</li><li>• 不能以_sys_或空格开头，不能以空格结尾</li><li>• 可用UTF-8格式表示的字母（包含中文）、数字和空格，以及以下字符：_ . : = + - @</li><li>• 128个字符以内且不与其他标签键重复</li></ul> |

| 参数  | 规则                                                                                                                                                      |
|-----|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| 标签值 | <ul style="list-style-type: none"> <li>可以为空字符串</li> <li>可用UTF-8格式表示的字母（包含中文）、数字、空格，以及以下字符：_ . : / = + - @</li> <li>长度0~255个字符（中文也可以输入255个字符）</li> </ul> |

- 完成后单击“确定”保存标签。

## 使用标签检索函数

- 返回函数工作流控制台，在左侧的导航栏选择“函数 > 函数列表”。
- 在搜索框中先选择筛选条件“标签”，再勾选标签键值对，一次可添加多个标签作为筛选条件。
- 单击“确定”，可以在函数列表查看检索结果。

## 7.10 配置函数的动态内存

本章节介绍如何通过函数工作流控制台，配置函数的动态内存，节省成本。

### 操作场景

默认情况下，一个函数唯一绑定了一个资源规格。开启动态内存可以让函数在处理指定请求时，设置本次处理中函数实例使用的资源规格。如果不指定资源规格，函数将使用默认配置的资源规格。

以使用函数做视频转码为例，视频文件大小从MB到GB，不同编码格式和分辨率对转码需要的计算资源要求差别很大。为了保证转码性能，通常需要配置一个很大的资源规格，但是在处理低分辨率视频（例如短视频）时，会造成资源浪费。用户可以把转码业务实现为元数据获取和转码两个函数，根据元数据信息指定转码函数的资源规格，最小化资源占用，达到更低的成本开销。

### 约束与限制

- 动态内存设置值必须为128、256、512、768、1024、1280、1536、1792、2048、2560、3072、3584、4096、8192、10240中的一个，单位为MB。
- 未开启动态内存，调用API执行函数时使用[配置常规信息](#)中设置的内存大小。
- 若配置了动态内存，未设置内存值，调用同步执行函数或异步执行函数API时使用[配置常规信息](#)中设置的内存大小。
- 若配置了动态内存，内存值设置错误，未包含在指定值的范围内，调用API时，返回错误码“FSS.0406”。

### 配置动态内存

- 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
- 单击函数的名称，进入函数详情页。
- 在“设置”页签下选择“高级设置”。
- 开启“动态内存”开关，然后单击“保存”保存配置。

调用同步执行函数或异步执行函数API的操作，请参见[同步执行函数](#)和[异步执行函数](#)接口的具体说明。

## 7.11 配置函数的心跳检测

本章节介绍如何通过函数工作流控制台，配置心跳函数，检测函数运行时的异常场景。

### 操作场景

心跳函数用于检测函数运行时的异常场景，如函数死锁、函数内存溢出、函数网络异常等。

配置了心跳函数后，当函数运行时，FunctionGraph每隔5秒向函数实例发送一次心跳请求，触发心跳函数。如果心跳请求返回异常，FunctionGraph会认为函数实例异常，将终止此函数实例。

FunctionGraph心跳请求的超时时间是3秒，如果连续6次心跳请求未响应，函数实例将被终止。

### 约束与限制

- 当前Java函数支持配置心跳函数。
- 心跳函数入口需要与函数执行入口在同一文件下。

Java心跳函数格式为：

```
public boolean heartbeat() {  
    // 自定义检测逻辑  
    return true  
}
```

- 心跳函数目前无输入参数，返回值为Boolean类型。

### 配置心跳函数

1. 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
2. 单击函数的名称，进入函数详情页。
3. 在“设置”页签下选择“高级设置”。
4. 开启“配置心跳函数”开关，并配置心跳函数的入口。

心跳函数入口需要与函数执行入口在同一文件下，格式为：[包名].[类名].[执行函数名]，不超过128个字符。

图 7-18 配置心跳函数

★ 心跳函数入口

com.example.Test.test

心跳函数入口需要与函数执行入口在同一文件下

格式为[包名].[类名].[执行函数名]，不超过128个字符

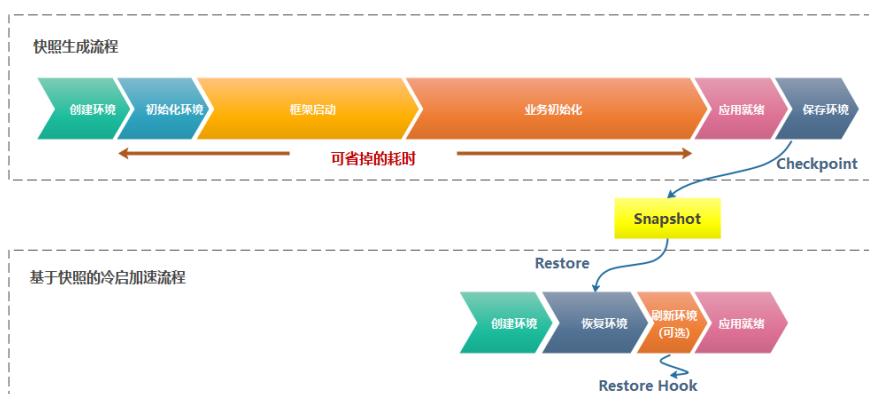
5. 完成后单击“保存”，保存心跳函数配置。

## 7.12 配置函数的快照式冷启动

### 快照式冷启动概述

FunctionGraph支持基于进程级快照的冷启动加速方案，用户无需额外付费，只需进行简单的配置、少量的代码修改，即可享受到该方案带来的冷启动性能提升。

当用户的Java函数启用快照式冷启动后，FunctionGraph会预先执行函数对应的初始化代码，获取其初始化执行上下文环境的快照，并进行加密缓存。后续调用该函数时会触发冷启动，**直接从提前初始化后的应用快照来恢复执行环境**，而非重新走一遍初始化流程，以此达到极大提升启动性能的效果。



### 约束与限制

- 当前仅Java运行时函数支持配置快照式冷启动。
- 如果函数强依赖于有状态，需要考虑使用Restore Hook进行状态刷新。
- 对于强依赖CPU指令集特性的函数，请先[工单系统](#)提前联系技术工程师确认是否可以使用快照式冷启动。
- 依赖硬编码Host环境（如hostname或者hostip）的函数，在迁移到其他主机上时，使用快照式冷启动可能会有问题。请先提前在[工单系统](#)提交工单联系技术工程师进行确认，建议避免依赖这些变量。
- 快照式冷启动当前仅支持基于x86机器开发的应用。

### 配置快照式冷启动

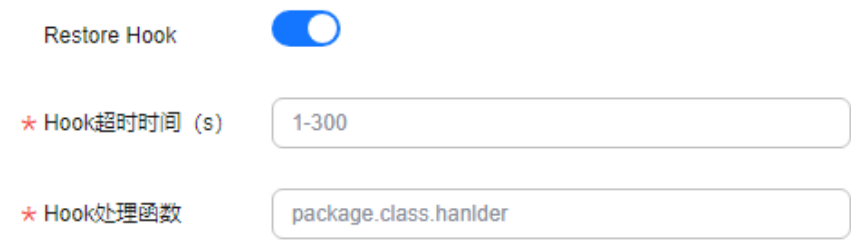
1. 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
2. 单击函数的名称，进入函数详情页。在“设置”页签下选择“高级设置”。
3. 开启“快照式冷启动”开关。

图 7-19 开启快照式冷启动



4. （可选）如果函数强依赖于有状态，还需配置Restore Hook进行状态刷新，并在函数代码中实现对应的Hook逻辑。
- Hook超时时间（秒）：Hook处理函数的执行超时时间。
  - Hook处理函数：Hook处理函数的入口，格式：[包名].[类名].[执行函数名]，不超过128个字符。

图 7-20 开启 Restore Hook



函数代码中Restore Hook示例如下：

```
2 usages
public class App
{
    public void restore(Context context) {
        System.setProperty("myKey", "restoreValue");
    }

    public void init(Context context) {
        System.setProperty("myKey", "initValue");
    }

    2 usages
    public void print(APIGTriggerEvent event, Context context) {
        System.out.println("value: " + System.getProperty("myKey"));
    }
}
```

5. 完成后单击“保存”，保存快照式冷启动配置。
6. 参考[发布新版本](#)发布函数版本，发布版本时开启“快照式冷启动”，触发快照的自动化制作。

图 7-21 发布新版本

发布新版本

1

新版本将以LATEST版本内容发布，版本发布后代码和配置将无法编辑。单个函数下您最多可以发布20个版本（含latest版本）。[了解更多](#)

×

版本号

请输入版本号

非必填项，若不指定则默认按时间生成版本号，如v20180116-202722  
可包含字母、数字、中划线、下划线和点，长度不超过42个字符，以字母/数字开头和结尾

快照式冷启动

描述

请输入版本描述，不超过512个字符

0/512

7. 等待快照制作完成（5分钟超时时间）。

图 7-22 快照制作中

代码

监控

设置

常规设置

触发器

权限

网络配置

磁盘挂载

环境变量

函数URL

开发

异步配置

日志配置

标签

生命周期

高级设置

高级设置

配置心跳函数

动态内存

在执行函数的时候根据需要动态指定内存规格，节省成本

快照式冷启动

快照正在制作中

Restore Hook

类隔离

开启后将使用单独的类加载器加载用户代码和依赖，解决用户依赖和运行时依赖的冲突，如无依赖冲突问题无需打开。

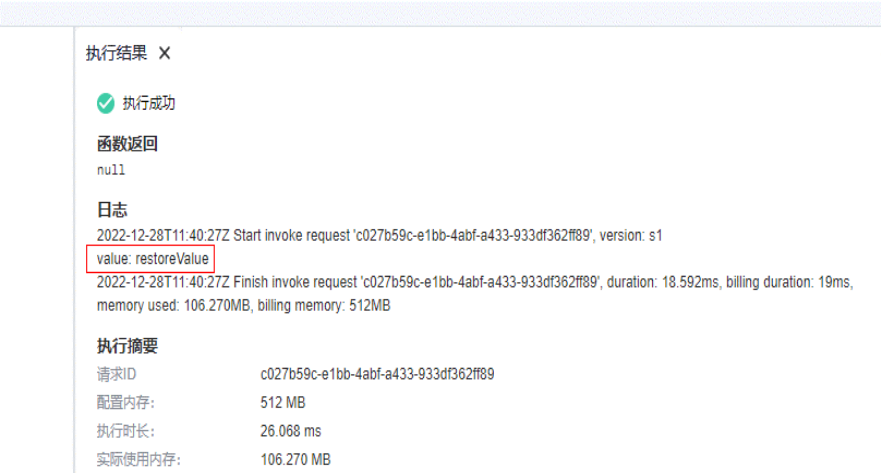
保存

图 7-23 快照制作成功



8. 调用开启了快照式冷启动的Java函数，查看执行日志中的快照式冷启动信息，如图7-24所示。

图 7-24 启动日志



相关文档

- 除使用控制台外，函数工作流支持通过API的方式管理函数的生命周期，详情请参见[函数生命周期管理API](#)。
- 在函数实际使用过程中如何进行冷启动优化，请参考[FunctionGraph冷启动优化实践](#)。

7.13 配置函数的 WebSocket 连接

本章节介绍通过函数工作流控制台，配置函数的WebSocket连接。

WebSocket 连接概述

在FunctionGraph中，可以通过配置APIG触发器，使函数能够响应WebSocket请求。配置完成后，相关联的函数可以作为一个Web服务器，处理WebSocket连接和消息，并将结果返回给客户端。适用于长连接、实时消息收发和实时数据监控等场景，能够实现高效的实时通信和数据交互。

使用WebSocket的计费方式与使用HTTP的计费方式完全一致，可以将WebSocket看作连接时间较长的HTTP调用。关于计费的更多信息，请参见[函数工作流计费概述](#)。

约束与限制

- 仅HTTP函数支持配置WebSocket，事件函数不支持配置。
- WebSocket配置当前仅支持“华东-上海一”区域使用。
- FunctionGraph在设置执行超时时间时，不区分WebSocket请求和HTTP请求。若WebSocket连接的持续时间超过设定的执行超时时间，该连接将被强制关闭，客户端将接收到1006状态码。

步骤一：创建 HTTP 函数

- 步骤1** 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
- 步骤2** 单击右上方的“创建函数”，进入“创建函数”页面。
- 步骤3** 选择“创建空白函数”，参考[表7-20](#)配置函数信息。

表 7-20 函数配置

| 参数   | 说明                                                                                                                                                                                                 | 取值样例           |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| 函数类型 | 选择函数类型，支持以下两种函数类型： <ul style="list-style-type: none"><li>• 事件函数：通过特定事件触发的函数，通常为JSON格式的请求事件。</li><li>• HTTP函数：处理HTTP请求的函数，可以直接发送HTTP请求到URL触发函数执行，从而使用Web服务。仅FunctionGraph v2版本支持创建HTTP函数。</li></ul> | HTTP函数         |
| 区域   | 选择要部署代码的区域。<br>当前WebSocket配置仅支持“华东-上海一”区域。                                                                                                                                                         | 华东-上海一         |
| 函数名称 | 填写函数名称，命名规则如下：<br>可包含字母、数字、下划线和中划线，长度不超过60个字符。以大/小写字母开头，以字母或数字结尾。                                                                                                                                  | WebSocket-demo |
| 企业项目 | 选择函数所属的企业项目。企业项目是一种云资源管理方式，企业项目管理服务提供统一的云资源按项目管理，以及项目内的资源管理、成员管理。<br>默认选择“default”，支持用户选择已创建的企业项目。如果没有开通企业管理服务，将无法看到企业项目选项。开通方法请参见 <a href="#">如何开通企业项目</a> 。                                      | default        |

| 参数   | 说明                                                                                                                                                                                                                                             | 取值样例                                    |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------|
| 委托   | <p>选择函数的委托。通过委托函数工作流来访问其他云服务，例如：如果函数需要访问LTS、VPC等服务，则必须选择包含所需服务权限的委托；若函数不访问任何云服务，则无需选择委托。</p> <p>默认“未使用任何委托”，支持选择已创建的委托。</p> <p>当华为云账号下无函数默认委托时，FunctionGraph提供快速创建默认委托“fgs_default_agency”的功能，详情请参见<a href="#">默认委托</a>。</p>                   | 未使用任何委托                                 |
| 高级设置 | <ul style="list-style-type: none"><li>函数访问公网：开启后允许函数访问公网上的服务，其公网访问带宽为用户间共享。</li><li>函数访问VPC内资源：开启时，函数将使用配置的VPC所绑定的网卡进行网络访问同时禁用函数工作流的默认网卡。</li><li>日志记录：启用日志功能后，函数运行过程中产生的日志会上报到云日志服务（LTS）。LTS将按需收取费用，具体请参见<a href="#">云日志服务价格详情</a>。</li></ul> | 函数访问公网：开启<br>函数访问VPC内资源：未开启<br>日志记录：未开启 |

**步骤4** 配置完成后单击“创建函数”，页面跳转至函数详情配置页面。

----结束

步骤二：配置函数代码

**步骤1** 在函数详情页的“代码”页签下，打开“index.js”文件，使用代码编辑器复制以下代码并部署，部署完成后如[图7-25](#)所示。

```
const WebSocket = require('ws');
const WebSocketServer = WebSocket.Server;

const wss = new WebSocketServer({
  host: '127.0.0.1',
  port: 8000,
  path: '/websocket_demo'
});

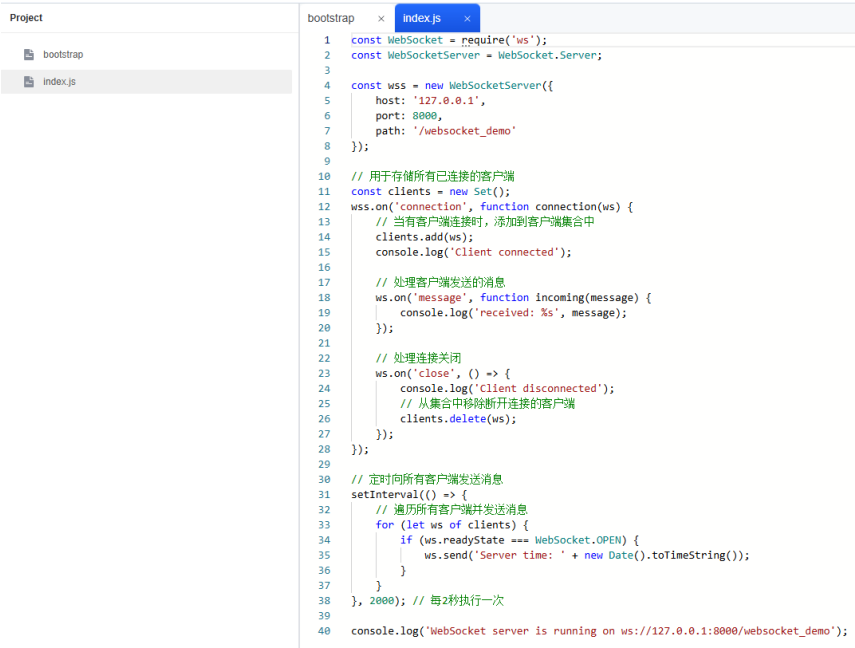
// 用于存储所有已连接的客户端
const clients = new Set();
wss.on('connection', function connection(ws) {
  // 当有客户端连接时，添加到客户端集合中
  clients.add(ws);
  console.log('Client connected');

  // 处理客户端发送的消息
  ws.on('message', function incoming(message) {
    console.log('received: %s', message);
  });

  // 处理连接关闭
  ws.on('close', () => {
    console.log('Client disconnected');
    // 从集合中移除断开连接的客户端
    clients.delete(ws);
  });
});
```

```
});  
});  
  
// 定时向所有客户端发送消息  
setInterval(() => {  
  // 遍历所有客户端并发送消息  
  for (let ws of clients) {  
    if (ws.readyState === WebSocket.OPEN) {  
      ws.send('Server time: ' + new Date().toString());  
    }  
  }  
}, 2000); // 每2秒执行一次  
  
console.log('WebSocket server is running on ws://127.0.0.1:8000/websocket_demo');
```

图 7-25 index.js 文件

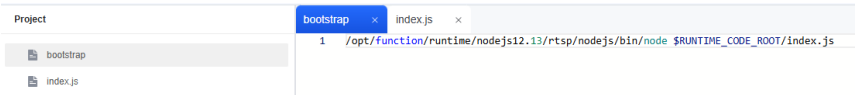


WebSocket Server监听的IP地址是127.0.0.1，监听的端口号默认值为8000端口。

**步骤2** 在“bootstrap”文件中编写如下内容，完成后如图7-26所示。

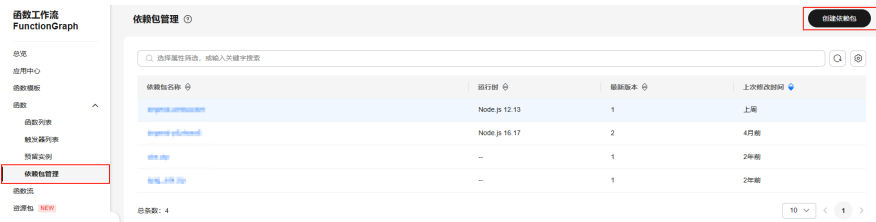
```
/opt/function/runtime/nodejs12.13/rtsp/nodejs/bin/node $RUNTIME_CODE_ROOT/index.js
```

图 7-26 bootstrap 文件



**步骤3** 返回函数工作流控制台，如图7-27所示在左侧的导航栏选择“函数 > 依赖包管理”，单击“创建依赖包”。

图 7-27 依赖包管理



**步骤4** 下载WebSocket依赖包“wss-nodejs12”文件，在“创建依赖包”界面如图7-28所示上传并配置WebSocket依赖包信息，填写完成后单击“创建”。

图 7-28 创建依赖包



表 7-21 创建依赖包

| 参数    | 参数说明                                                                  | 取值样例             |
|-------|-----------------------------------------------------------------------|------------------|
| 依赖包名称 | 自定义填写依赖包的名称。<br>以大/小写字母开头，以字母或数字结尾。<br>可包含字母、数字、下划线、点和中划线，长度不超过96个字符。 | depend-websocket |
| 运行时   | 选择依赖包的运行时。                                                            | Node.js 12.13    |

| 参数         | 参数说明                                                                                                                                                                                                                                                          | 取值样例                                             |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|
| 代码上传方式     | 选择代码上传方式。支持“上传ZIP文件”或通过OBS链接“从OBS上传文件”。                                                                                                                                                                                                                       | 上传ZIP文件                                          |
| 文件上传/OBS链接 | <ul style="list-style-type: none"><li>代码上传方式为“上传ZIP文件”：单击“添加文件”上传打包为ZIP格式文件的依赖包。上传的ZIP文件大小限制为10M，超过10M请使用从OBS上传文件。</li><li>代码上传方式为“从OBS上传文件”：填写指向代码文件对象的OBS对象URL，需将对象放入与函数在同一区域下的OBS桶中，该对象必须为ZIP格式文件。可参考<a href="#">通过URL访问OBS对象</a>复制所需的代码文件对象URL。</li></ul> | 添加文件，并上传 <a href="#">步骤4</a> 下载的“wss-nodejs12”文件 |
| 描述         | 填写依赖包的描述说明。                                                                                                                                                                                                                                                   | -                                                |

- 步骤5** 返回“WebSocket-demo”函数详情页，在“代码”页签最下方的“代码依赖包”模块，单击“添加依赖包”。
- 步骤6** 在选择依赖包弹窗中，依赖包源选择“私有依赖包”，添加[步骤4](#)创建的“depend-websocket”依赖包，单击“确定”完成依赖包添加。
- 步骤7** 选择“设置 > 高级设置”，如[图7-29](#)所示开启“WebSocket”参数开关。

图 7-29 开启 WebSocket



----结束

步骤三：创建 APIG 专享版实例

请根据业务需要，参考[创建APIG实例](#)购买所需规格的专享版APIG实例。

购买过程中请参照以下注意事项：

表 7-22 创建 APIG 实例注意事项

| 参数   | 说明                                                                                                                                                              |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 区域   | 选择与所创建函数一致的区域。                                                                                                                                                  |
| 公网入口 | <ul style="list-style-type: none"><li>如使用<a href="#">步骤五：测试函数</a>进行本地测试，则需“开启公网入口”，请根据实际需求选择入公网带宽。</li><li>如在生产环境中使用VPC内访问函数的WebSocket特性，则无需“开启公网入口”。</li></ul> |
| 网络   | 选择创建的VPC和子网。<br>如在生产环境中使用可 <a href="#">配置函数访问VPC</a> ，并在此参数中配置与函数相同的VPC与子网。                                                                                     |

步骤四：创建 APIG 触发器

**步骤1** 选择“设置 > 触发器”，单击“创建触发器”，参考[表7-23](#)创建一个API 网关服务（APIG专享版）触发器。

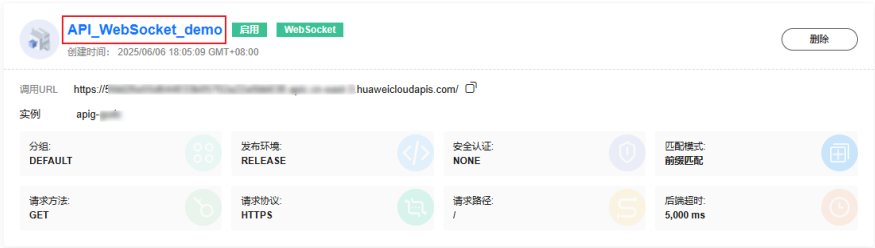
表 7-23 APIG 专享版触发器参数说明

| 参数    | 说明                                                                                                                                                                                       | 取值样例               |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|
| 触发器类型 | 必选参数。<br>选择“API网关服务（APIG专享版）”。                                                                                                                                                           | API网关服务（APIG专享版）   |
| 实例    | 必选参数。<br>选择APIG专享版实例。若无实例，可单击“创建实例”完成创建。                                                                                                                                                 | apig-ws            |
| API名称 | 必选参数。<br>填写专享版APIG触发器的名称。支持汉字、英文、数字和下划线，且只能以英文和汉字开头，3-64字符。                                                                                                                              | API_websocket_demo |
| 分组    | 必选参数。<br>选择API分组。API分组可以视为一组API的集合，API提供方以API分组为单位，管理分组内的所有API。<br>若未创建分组，可单击“创建分组”完成创建，创建完成单击右侧  即可。 | APIGroup_ws        |

| 参数       | 说明                                                                                                                                                                                                                                                                                                    | 取值样例    |
|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| 发布环境     | 必选参数。<br>选择API的发布环境。API可以同时提供给不同的场景调用，如生产、测试或开发，选择“RELEASE”，才能调用。<br>若未创建环境，可单击“创建发布环境”完成创建。                                                                                                                                                                                                          | RELEASE |
| 安全认证     | 必选参数。<br>API认证方式有以下三种方式： <ul style="list-style-type: none"><li>• App：采用Appkey&amp;Appsecret认证，安全级别高，推荐使用，详情请参见<a href="#">APP认证</a>。</li><li>• IAM：IAM认证，只允许IAM用户能访问，安全级别中等，详情请参见<a href="#">IAM认证</a>。</li><li>• None：无认证模式，所有用户均可访问。</li></ul> <b>注意</b><br>当前示例中安全认证选择“None”，实际业务使用推荐选择安全级别高的认证方式。 | None    |
| 请求协议     | 必选参数。<br>在使用WebSocket时HTTPS对应WSS，HTTP对应WS。<br>API的请求协议分为两种类型： <ul style="list-style-type: none"><li>• HTTP：数据在传输过程中不被加密。</li><li>• HTTPS：数据在传输过程中被加密。</li></ul>                                                                                                                                       | HTTPS   |
| 请求方法     | 必选参数。<br>如需使用WebSocket协议，请求方法至少需要支持GET。<br>支持的请求方法：GET、POST、DELETE、PUT、PATCH、HEAD、OPTIONS、ANY。                                                                                                                                                                                                        | GET     |
| 后端超时(毫秒) | 必选参数。<br>设置API的后端超时时间，单位为毫秒，设置范围为1~60000毫秒。<br>后端超时时间表示WebSocket的 <b>最大空闲连接时间</b> ，例如设置为5000毫秒，即当WebSocket超过5000毫秒未收发消息时，连接断开。                                                                                                                                                                        | 5000    |

**步骤2** 如[图7-30](#)所示，创建成功后单击已创建的触发器名称，进入APIG服务界面。

图 7-30 APIG 触发器



说明

APIG触发器创建成功后，访问地址不会变更。

**步骤3** 在API实例详情界面，如图7-31所示单击“编辑”，进入编辑API界面。

图 7-31 编辑 API



**步骤4** 单击“下一步”至“默认后端”页签，如图7-32所示配置API参数，单击“完成”。

- 后端服务类型：选择“FunctionGraph”。
  - 网络架构：选择“V2”。（如没有此参数，请向APIG服务[提交工单](#)开启白名单配置）
  - 调用类型：选择“Synchronous”。WebSocket场景下，API的后端调用类型只支持Synchronous同步调用。
- 其他参数保持默认即可。

图 7-32 编辑 API 默认后端



**步骤5** 返回API实例详情界面，如图7-33所示单击“发布最新版本”。

图 7-33 发布最新版本 api

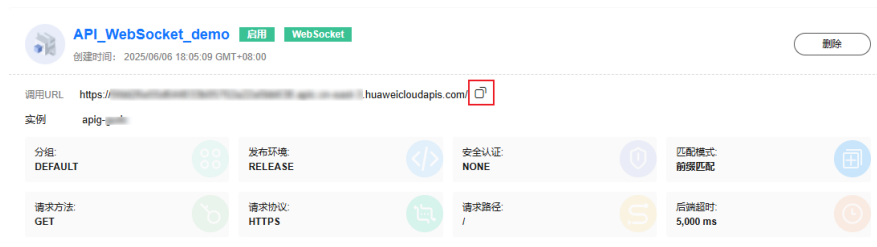


----结束

## 步骤五：测试函数

**步骤1** 返回函数 workflow 控制台，进入函数详情页面，选择“设置 > 触发器”，复制APIG触发器的访问地址。

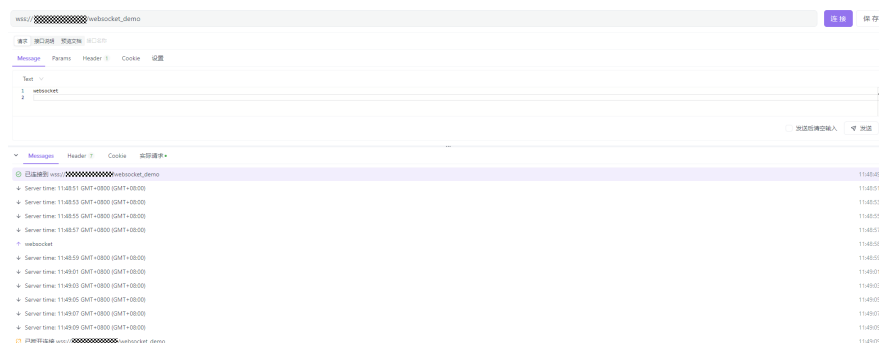
图 7-34 复制 APIG 触发器 URL



### 步骤2 使用Postman测试函数的正确性。

1. 在Postman中创建WebSocket请求。将APIG触发器的调用URL复制到Postman中，并将Scheme由HTTPS变更为WSS。
2. 根据业务需求配置Params和Headers参数。
3. 连接WebSocket，连接成功后即可发送消息，可查看消息接收情况。
4. 超过执行超时时间后，会断开与WebSocket服务器的连接。

图 7-35 WebSocket 测试



---结束

## WebSocket 相关说明

- **连接保活和超时重连**

当WebSocket连接建立后，除连接持续时间超过设定的执行超时时间会导致连接断开外，函数工作流不会干预任何逻辑。若在存续期间，您的WebSocket连接在特定时段内无数据传输，该连接可能被网络中间节点（例如NAT网关）关闭。在该场景下，您需要利用WebSocket协议的Ping、Pong帧来保持连接活动或验证WebSocket连接的有效性。

若业务需求超出函数工作流所能提供的最大请求超时时间，或应用需要在运行期间保持逻辑上的连接稳定，建议在客户端代码中加入超时重连机制。可使用 **Reconnecting-WebSocket** 库或 **SocketIO** 库以实现该机制。

- 会话亲和性

FunctionGraph作为无服务器计算平台，其函数实例采用请求触发式生命周期管理。在高并发场景下，系统通过自动扩缩容机制创建多个实例副本以应对流量压力，但无法保证同一客户端的连续请求路由至固定实例。对于需要维持会话状态的WebSocket应用，推荐采用Redis、Kafka、数据库等外部存储系统实现跨实例的状态同步。

例如，在聊天室应用中，由于函数工作流无法保证所有用户同时连接到同一个函数实例，因此可以利用Redis的发布订阅功能来实现消息广播。当用户加入一个聊天室时，函数会订阅该聊天室对应的频道。当用户发送消息时，函数会将消息发布到Redis中该聊天室对应的频道。由于同一聊天室的所有用户都已订阅了该频道，因此所有用户都会接收到该消息。

## 7.14 配置函数的流式返回

### 函数流式返回概述

为满足当前Web应用及AI应用在实时数据传输和大报文传输方面的需求，函数支持配置流式返回开关，将函数的响应报文以HTTP流式传输的方式返回给客户端。

**采用流式返回具有以下优势：**

- 支持更大规模的响应报文（最大可达200M）。
- 由于流式传输无需一次性传输全部报文，可在函数代码中分批处理和传输数据，有效降低内存占用。
- 通过流式传输可实现更快的首字节响应时间。

**适用场景包括：**

- 大文件或大数据集的处理场景（如图片、视频处理等）。
- 实时数据处理场景（例如AI大模型应用需使用Server-Sent Events (SSE) 协议进行数据传输），需要快速返回响应数据，而非一次性返回全部数据。

### 约束与限制

- 流式返回配置当前支持“西南-贵阳一”和“华北-北京四”区域使用。
- 仅容器镜像函数、定制运行时函数和HTTP函数支持配置流式返回。
- 流式返回最大支持响应报文大小为200MB。
- 流式返回仅支持同步调用，执行超时时间限制最大300s。

### 配置流式返回

1. 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
2. 单击函数的名称，进入函数详情页。在“设置”页签下选择“高级设置”。
3. 开启“流式返回”，单击“保存”。

图 7-36 开启流式返回



编写流式返回函数代码

以下为容器镜像函数、HTTP函数和定制运行时函数的代码示例。

容器镜像函数示例代码

参考[使用容器镜像创建函数](#)完成容器镜像函数创建，并开启流式返回。请注意，[函数超时时间配置](#)需控制在300s以内，否则将导致函数保存失败。

为了确保函数能够正确返回流式数据，在函数的响应体生成过程中，建议分批次写入流式数据，以避免内存浪费。以下提供一个简单的Java SpringBoot应用返回流式数据的示例：

```
@SpringBootApplication
@RestController
public class SpringBootDemoHelloworldApplication {
    private ExecutorService executor = Executors.newCachedThreadPool();
    public static void main(String[] args) {
        SpringApplication.run(SpringBootDemoHelloworldApplication.class, args);
    }
    /**
     * Stream Response
     * @return
     */
    @PostMapping(value = "/invoke")
    public ResponseEntity<ResponseBodyEmitter> invoke(@RequestBody Map<String, Object> body) {
        ResponseBodyEmitter emitter = new ResponseBodyEmitter();
        executor.submit(() -> {
            try {
                emitter.send("hello \n", MediaType.TEXT_EVENT_STREAM);
                emitter.send("world \n", MediaType.TEXT_EVENT_STREAM);
                emitter.complete();
            } catch (Exception ex) {
                emitter.completeWithError(ex);
            }
        });
        HttpHeaders responseHeaders = new HttpHeaders();
        responseHeaders.setContentType(MediaType.TEXT_EVENT_STREAM);
        return new ResponseEntity(emitter, responseHeaders, HttpStatus.OK);
    }
}
```

## HTTP 函数示例代码

参考[创建HTTP函数](#)完成HTTP函数创建，并开启流式返回。请注意，[函数超时时间配置](#)需控制在300s以内，否则将导致函数保存失败。

HTTP函数的示例代码可参考[容器镜像函数示例代码](#)。

## 定制运行时函数示例代码

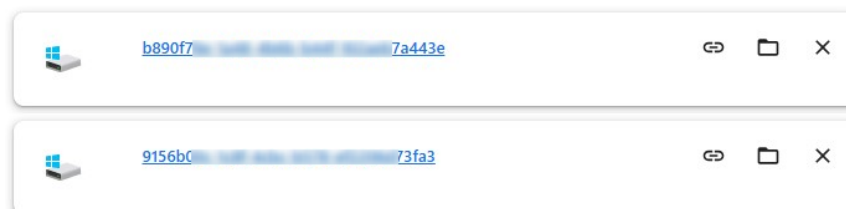
定制运行时函数需调用http://\$RUNTIME\_API\_ADDR/v1/runtime/invoke/response/\$REQUEST\_ID的POST方法返回流式数据。以下为一个简单的Shell类型bootstrap示例：

```
#!/bin/sh
set -o pipefail
#Processing requests loop
while true
do
  HEADERS="$(mktemp)"
  # Get an event
  EVENT_DATA=$(curl -sS -LD "$HEADERS" -X GET "http://$RUNTIME_API_ADDR/v1/runtime/invoke/request")
  # Get request id from response header
  REQUEST_ID=$(grep -Fi x-cff-request-id "$HEADERS" | tr -d '[:space:]' | cut -d: -f2)
  if [ -z "$REQUEST_ID" ]; then
    continue
  fi
  # Send streamdata file content back to client
  curl -X POST "http://$RUNTIME_API_ADDR/v1/runtime/invoke/response/$REQUEST_ID" -H "Content-Type:application/octet-stream" --data-binary "@code/streamdata.txt"
done
```

## 测试流式返回函数

1. 在函数详情页的“代码”页签下，单击“测试”，弹出“配置测试事件”弹窗。
2. 选择“空白模板”，无需修改单击“创建”。
3. 选择2创建的测试事件，再次单击“测试”。
4. 函数执行成功后，将自动启动下载，下载文件名为此次请求的ID，文件内容为此次请求的返回值。

图 7-37 文件下载



## 7.15 配置 Java 函数类隔离和预停止

类隔离是指使用单独的类加载器加载用户代码和依赖，解决用户依赖和运行时依赖的冲突，如无依赖冲突问题则无需使用类隔离。

预停止是指每次FunctionGraph停止当前函数实例前，用户希望FunctionGraph调用指定回调函数，然后再停止函数实例。

约束与限制

当前仅使用Java语言的函数支持配置类隔离和预停止。

配置 Java 函数的类隔离

- 1. 登录[函数工作流控制台](#)，在左侧导航栏选择“函数 > 函数列表”。
- 2. 单击函数的名称，进入函数详情页。
- 3. 在“设置”页签下选择“高级设置”。
- 4. 开启“类隔离”开关，单击“保存”类隔离配置。

配置 Java 函数的预停止

- 1. 登录[函数工作流控制台](#)，在左侧导航栏选择“函数 > 函数列表”。
- 2. 单击函数的名称，进入函数详情页。
- 3. 在“设置”页签下选择“生命周期”。
- 4. 开启“预停止配置”开关，并配置相关参数。

表 7-24 预停止配置说明

| 参数         | 说明                                                                      |
|------------|-------------------------------------------------------------------------|
| 预停止超时时间（秒） | 停止当前函数实例前，执行回调函数的超时时间，取值范围为1-90的整数。                                     |
| 函数预停止入口    | 停止当前函数实例前，需要FunctionGraph调用的回调函数执行入口。格式为：[包名].[类名].[执行函数名]，长度不超过128个字符。 |

- 5. 单击“保存”，保存预停止配置。

7.16 配置 HTTP 函数的请求头传入密钥

配置请求头传入密钥

HTTP函数的密钥信息仅能通过请求头传递，如需获取HTTP函数的AK、SK、Token等如[表7-25](#)所示的信息，请参考以下步骤操作：

- 1. 登录[函数工作流控制台](#)，进入需配置的HTTP函数详情页。
- 2. 单击“设置 > 高级设置”，开启“请求头传入密钥”开关，单击“保存”，即可使用请求头传入密钥。

图 7-38 请求头传入密钥



表 7-25 请求头传入密钥

| 字段                        | 说明                                                                                                            |
|---------------------------|---------------------------------------------------------------------------------------------------------------|
| X-CFF-Auth-Token          | Token是系统颁发给IAM用户的访问令牌，承载用户的身份、权限等信息。                                                                          |
| X-CFF-Security-Access-Key | 临时AK/SK和securitytoken是系统颁发给IAM用户的临时访问令牌。<br>临时AK/SK和securitytoken遵循权限最小化原则。<br>临时AK/SK和securitytoken两者必须同时使用。 |
| X-CFF-Security-Secret-Key |                                                                                                               |
| X-CFF-Security-Token      |                                                                                                               |

相关文档

关于HTTP函数的更多说明请参考[创建HTTP函数](#)。

7.17 导入导出函数

FunctionGraph支持导入导出函数，可以把函数以文件的形式导出到本地，并在其他区域或其他用户的函数工作流控制台导入，实现函数配置数据的迁移。

约束与限制

表 7-26 导入导出函数约束限制

| 限制类别   | 具体限制                                                                                                                                                                                                                                                              |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 导出函数限制 | <ul style="list-style-type: none"><li>导出函数资源包大小不超过50MB。</li><li>导出的函数资源中配置信息不包含别名信息。</li><li>通过函数列表界面导出函数时，仅能导出latest版本的函数。</li><li>支持加密环境变量导出。</li></ul>                                                                                                         |
| 导入函数限制 | <ul style="list-style-type: none"><li>在跨区域导入函数时（例如从“华东-上海一”导出的函数在“华北-北京四”导入），由于不同区域的密钥可能不一致，函数的加密环境变量可能无法解密，导致无法正常生效，需重新配置加密环境变量。</li><li>上传代码时，如果代码中包含敏感信息（如账户密码等），请您自行加密，以防止信息泄露。</li><li>上传的文件大小限制为10MB。</li><li>上传文件是包含函数代码(zip)和函数配置文件(yaml)的zip包。</li></ul> |

导出函数

1. 登录[函数工作流控制台](#)，在左侧导航栏选择“函数 > 函数列表”。

2. 在“函数”页面，可以通过以下两种方式导出函数。

- 批量导出函数：在函数列表中勾选要导出的函数，单击“导出函数”，导出的函数将下载为zip格式文件。

**批量导出函数仅能导出函数的latest版本。**

图 7-39 批量导出函数



- 导出单个函数：
  - i. 单击函数名称，进入函数详情页面。
  - ii. 在函数详情页的最上方选择要导出的函数版本。
  - iii. 在页面右上角单击“导出函数”，导出对应版本的函数。

图 7-40 导出函数



## 导入函数

1. 登录[函数工作流控制台](#)，在左侧导航栏选择“函数 > 函数列表”。
2. 单击“导入函数”，在导入函数弹窗中单击“添加文件”，选择本地已导出的zip格式函数文件。

图 7-41 导入函数



3. 系统将自动读取和配置函数的运行时，自定义编辑函数名称后单击“确定”，完成函数导入。

## 相关文档

除使用控制台外，函数工作流支持通过API的方式管理函数导入导出，详情请参见[函数导入导出API](#)。

## 7.18 启用禁用函数

FunctionGraph支持对函数进行禁用和启用操作，已禁用的函数不能被调用执行。

### 约束与限制

- 只能禁用“latest”版本的函数，不能禁用已经发布的函数版本。
- 基于已禁用的“latest”版本重新发布新版本，发布后的新版本也处于禁用状态且不能启用。

- 当函数处于禁用状态时可以修改代码，但不能执行函数。

## 禁用函数

1. 登录[函数工作流控制台](#)，在左侧导航栏选择“函数 > 函数列表”。
2. 单击函数名称，进入函数详情页面。
3. 在页面右上角单击“禁用函数”，在弹窗中单击“确定”，禁用函数。

## 启用函数

1. 登录[函数工作流控制台](#)，在左侧导航栏选择“函数 > 函数列表”。
2. 单击被禁用的函数名称，进入函数详情页面。
3. 在页面右上角单击“启用函数”，启用函数。

# 7.19 为函数配置预留实例

## 实例类型介绍

FunctionGraph提供了按量和预留两种类型的实例。

- 按量实例是由FunctionGraph根据用户使用函数的实际情况来创建和释放，当FunctionGraph收到函数的调用请求时，自动为此请求分配执行环境。
- 预留实例是将函数实例的创建和释放交由用户管理，如果一个函数创建了预留实例，FunctionGraph收到此函数的调用请求时，会优先将请求转发给预留实例。当请求的峰值超过预留实例处理能力时，剩余的请求将会转发给按量实例，由函数工作流自动为函数分配执行环境。

预留实例在创建完成后，会自动加载该函数的代码、依赖包以及执行初始化入口函数，且预留实例会常驻环境，消除冷启动对业务的影响。**请注意不要依赖预留实例本身的初始化函数去执行一次性业务。**

函数当前支持[配置固定数量的预留实例策略](#)、[配置定时伸缩的预留实例策略](#)和[配置智能伸缩的预留实例策略](#)。

### 说明

用户默认没有权限使用指标策略和智能推荐策略，如果需要使用该功能，请在[工单系统](#)提交工单添加白名单。

## 约束与限制

- 别名和对应的版本不可以同时配置预留实例。比如，latest版本对应的别名为1.0，在latest版本下配置了预留实例后，不能在别名1.0再配置预留实例，反之同理。
- 开启闲置模式后，因为在最初阶段会涉及实例的初始化及模式转换，因此该段时间会以预留实例的非闲置模式计费标准进行计费。
- 当函数调用并发数大于预留实例数量时，超出部分会分配给按量实例，这部分请求仍然有冷启动。

计费说明

为函数配置预留实例将按需收取费用，预留实例计费规则与计费项详情请参考[函数工作流计费项](#)。

前提条件

配置按指标伸缩的预留实例策略前，需为函数配置包含AOM服务指标查询权限和函数服务配置查询权限的委托，具体操作请参见[配置函数的委托权限](#)。

配置固定数量的预留实例策略

1.

登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
2.

单击函数的名称，进入函数详情页。
3.

在“设置”页签下选择“并发”。
4.

单击“添加”，配置预留实例策略。

预留实例的数量不能超过并发实例数配额和单函数最大实例数。

图 7-42 基础配置

基础配置

函数名称test-lts

类型

版本

别名

选择版本latest

预留实例数0

配置预留实例数后，函数工作流会为您创建固定数目的函数实例，并且在您将预留实例数设置为 0 之前预留实例会持续运行。

闲置模式

预留实例在无调用的时候暂停CPU，节省资源，降低费用成本

表 7-27 基础配置说明

| 参数    | 说明                                                                   |
|-------|----------------------------------------------------------------------|
| 函数名称  | 展示当前配置预留实例的函数名称。                                                     |
| 类型    | 选择是要为函数的“版本”还是“别名”配置预留实例。                                            |
| 选择版本  | 仅当“类型”选择“版本”时需要配置。<br>选择要配置预留实例的函数版本。                                |
| 选择别名  | 仅当“类型”选择“别名”时需要配置。<br>选择要配置预留实例的函数别名，支持选择灰色版本的别名。                    |
| 预留实例数 | 配置预留实例的数量，取值范围为1-1000的整数。配置预留实例数后，FunctionGraph会为函数创建固定数目的函数实例并持续运行。 |

| 参数     | 说明                                     |
|--------|----------------------------------------|
| 闲置模式   | 开启闲置模式后，预留实例会在无调用的时候暂停CPU，节省资源，降低费用成本。 |
| 弹性预留策略 | 固定数量的预留实例无需配置。                         |

5. 单击“确定”，完成预留实例策略的配置。

配置定时伸缩的预留实例策略

配置预留实例时，可以为预留实例添加定时策略，函数可以在指定时间段内更新预留实例的数量。超过了该时间段，函数将预留实例数恢复到基础预留实例数量。

1. 参见[配置固定数量的预留实例策略](#)进行预留实例策略的基础配置。
2. 在添加预留实例策略弹窗中，单击“添加策略”，参考[表7-28](#)添加弹性预留策略。

表 7-28 定时策略配置说明

| 参数           | 说明                                                                                                                                                        |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| 策略类型         | 配置定时伸缩的预留实例策略时选择“定时策略”。                                                                                                                                   |
| 策略名称         | 自定义策略名称，以大写或小写字母开头，以字母或数字结尾，可包含字母、数字、下划线和中划线，长度不超过60个字符。                                                                                                  |
| Cron表达式(UTC) | 通过Cron表达式设置定时策略中预留实例数生效的时间，Cron表达式填写规则请参见 <a href="#">Cron表达式规则</a> 。<br>如果按照北京时间配置，请将当前时间减去8小时。支持指定时区，格式为： <b>CRON_TZ=Asia/Shanghai 0 */10 * * * *</b> 。 |
| 生效时间         | 定时策略生效的时间窗。只有当时间在时间窗内时，定时策略才会生效。当预留实例策略中的所有弹性策略都不生效时，预留实例数就会还原到基础配置中的预留实例数。                                                                               |
| 预留实例数        | 根据实际业务场景，定时策略生效时需要创建的预留实例数，该参数值必须大于或等于 <a href="#">配置固定数量的预留实例策略</a> 中配置的基础预留实例数。                                                                         |

3. 单击“确定”，完成定时策略的添加。

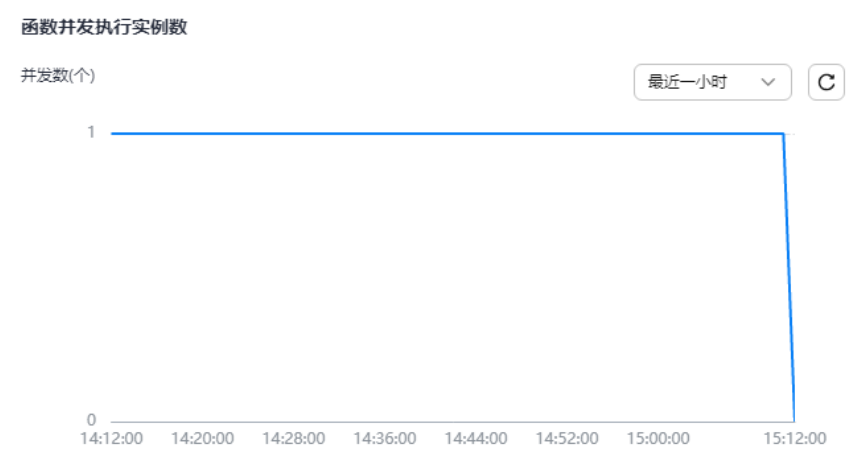
图 7-43 列表展示



4. 单击“确定”，完成预留实例策略的配置。

- 可同时配置多条定时策略，如配置早晨8点的时候，配置策略更新预留实例数为100，而21点的时候配置策略更新为10个。
- 配置完成后，您可以在“预留实例策略配置”列表，单击预留实例策略的名称进入预留实例策略配置页面，再单击弹性策略名称，查看函数并发执行的实例数统计。

图 7-44 查看并发执行实例数



说明

时间业务是可以配置多条定时策略，如配置早晨8点的时候，配置策略更新预留实例数为100，而21点的时候配置策略更新为10个。

配置智能伸缩的预留实例策略

FunctionGraph基于特征画像与负载预测技术，提供了预留实例的智能推荐策略，使预留实例随函数的负载动态变化，波峰时提前扩容，波谷时释放多余预留实例。

配置预留实例时，可以为预留实例添加智能推荐策略，函数可以根据负载情况更新预留实例的数量。

用户配置预留实例时，能够选择智能推荐策略，支持高性能、均衡、低成本三种选项，由系统根据用户负载模式，基于负载预测动态调整预留实例数量，适应负载的波峰波谷变化，并对相应的预留实例成本与性能提供直观展示。（注：智能推荐策略与其他预留实例弹性策略不能共存，且同一版本或别名只能存在一个。）

1. 参见[配置固定数量的预留实例策略](#)进行预留实例策略的基础配置。
2. 在添加预留实例策略弹窗中，单击“添加策略”，参考[表7-29](#)添加弹性预留策略。

图 7-45 智能推荐策略

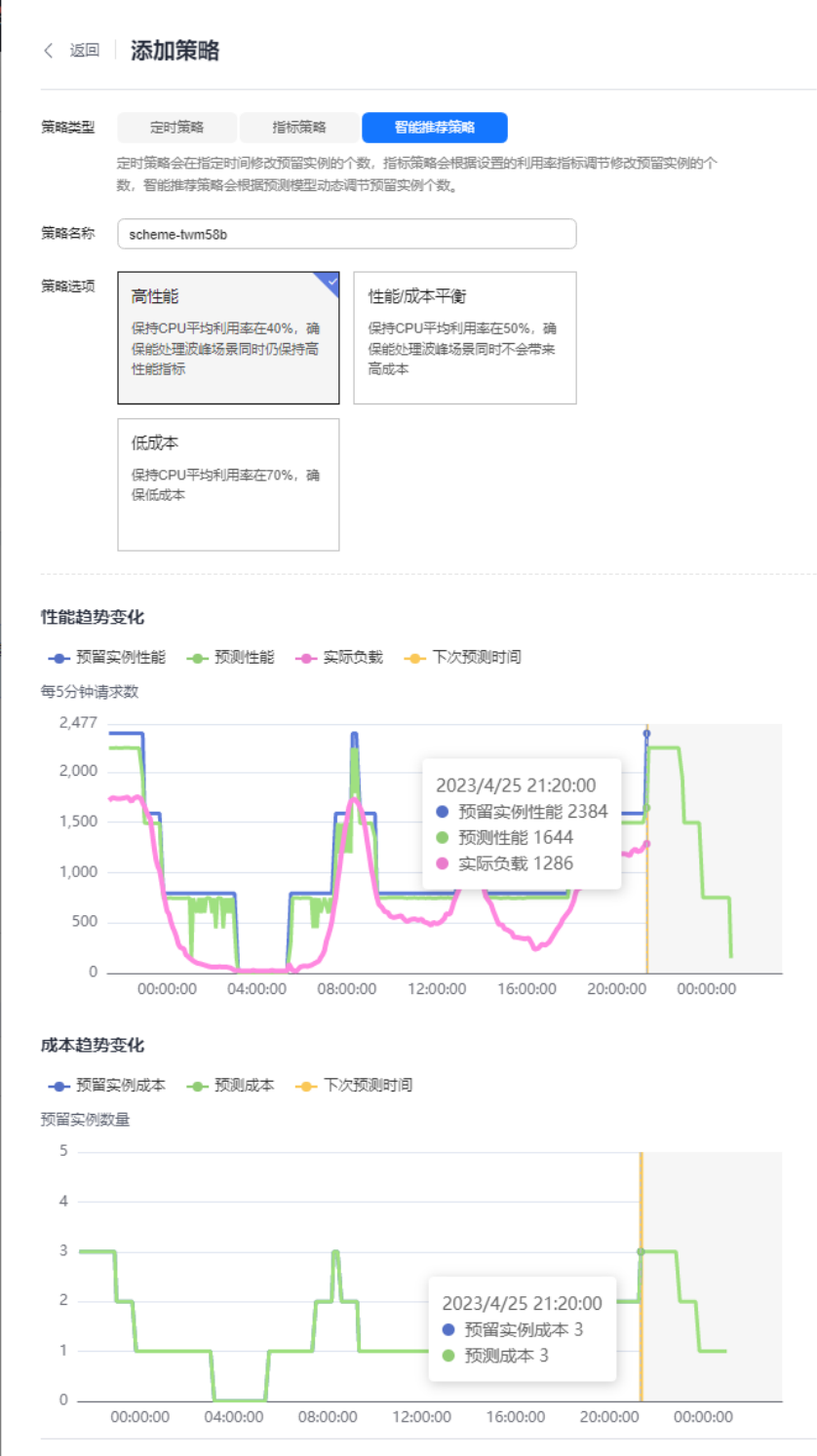


表 7-29 智能推荐策略配置说明

| 参数   | 说明                        |
|------|---------------------------|
| 策略类型 | 配置智能推荐的预留实例策略时选择“智能推荐策略”。 |

| 参数   | 说明                                                                                                                                                                                                                                      |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 策略名称 | 自定义策略名称，以大写或小写字母开头，以字母或数字结尾，可包含字母、数字、下划线和中划线，长度不超过60个字符。                                                                                                                                                                                |
| 策略选项 | <div>根据业务需要选择智能策略的执行模式，可选择“高性能”、“性能/成本平衡”和“低成本”三种模式。</div> <ul style="list-style-type: none"><li>高性能：保持CPU平均利用率在40%，确保能处理波峰场景同时仍保持高性能指标。</li><li>性能/成本平衡：保持CPU平均利用率在40%，确保能处理波峰场景同时不会带来高成本。</li><li>低成本：保持CPU平均利用率在70%，确保低成本。</li></ul> |

3. 单击“确定”，完成指标策略的添加。

图 7-46 预留实例策略配置

| 策略名称 | 策略选项 | 策略名称 | 策略选项 | 策略名称 | 策略选项 |
|------|------|------|------|------|------|
| 策略名称 | 策略选项 | 策略名称 | 策略选项 | 策略名称 | 策略选项 |

4. 单击“确定”，完成预留实例策略的配置。

配置完成后，您可以在“预留实例策略配置”列表，单击预留实例策略的名称进入预留实例策略配置页面，再单击弹性策略名称，查看函数预留实例成本和性能统计。

相关文档

除使用控制台外，函数工作流支持通过API的方式管理函数的预留实例，详情请参见[函数预留实例API](#)。

7.20 基于 RAM 共享函数资源

基于资源访问管理RAM服务，FunctionGraph可以实现跨账号共享函数资源，资源所有者将资源同时共享给其他多个账号使用，资源使用者接受共享邀请后就可以访问和使用共享的函数资源。

资源所有者可以依据最小权限原则和不同的使用诉求，选择不同的共享权限，资源使用者只能对资源进行权限内的访问，保证共享资源在满足资源使用者业务诉求的同时，提升资源管理的安全性。关于RAM服务的更多信息请参见[什么是资源访问管理](#)。

当您的账号由华为云组织管理时，您还可以利用此优势更轻松地共享资源。如果您的账号在组织中，则您可以与单个账号共享，也可以与组织或OU中的所有账号共享，而不必枚举每个账号，具体请参见[启用与组织共享资源](#)。

约束与限制

- 只有资源的所有者才能把资源共享给其他账号，资源使用者无法把其他账号共享的函数资源再共享给其他账号。资源使用者只能查看和调用共享的函数。
- 当用户需要与组织或组织单元共享函数资源时，必须启用与组织共享资源功能。更多信息请参考[启用与组织共享资源](#)。
- 单个账号最多只能被共享50个函数资源。

- 指定共享资源时，资源类型选择“functiongraph:function”。
- 共享函数资源创建完成后，需要使用者在一定时间内接受共享申请，才可以使用该函数资源，详细请参见[接受/拒绝共享邀请](#)。
- 当前FunctionGraph仅支持共享函数资源（资源类型为“function: 函数”）。

### 创建函数资源共享

资源所有者需要在RAM管理控制台中创建共享资源，具体操作请参见[创建共享](#)。

### 查看和使用共享的函数

1. 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
2. 在页面上选择“共享函数”页签，可以查看其他账号共享的函数资源。
  - 如果您是函数资源的所有者，您可以通过共享名称，在RAM管理控制台，找到对应的共享，查看共享内的资源情况、资源的权限以及资源的使用者，具体操作请参见[查看共享](#)。
  - 如果您是函数资源的使用者，您可以通过共享名称，在RAM管理控制台，找到对应的共享，查看共享内的资源情况、资源的权限以及资源的所有者，具体操作请参见[查看共享给您的资源](#)。

图 7-47 查看共享资源



3. 单击函数名称，可以查看函数的详情信息并调用函数。

### 停止函数资源共享

- 资源所有者如果不再需要某个共享时，可以随时将其删除，删除共享不会删除共享的资源。共享删除后，共享资源指定的使用者将无法继续使用该共享中的资源，详情请参见[删除共享](#)。
- 资源所有者可以随时更新资源共享实例，支持更新资源共享实例的名称、描述、标签、共享的资源、共享权限以及共享使用者，详情请参见[更新共享](#)。
- 资源使用者如果不再需要访问共享给您的资源，可以随时退出共享。退出共享后，将失去对共享资源的访问权限。

只有当共享资源的指定使用者是华为云账号而不是组织内共享时，才可以退出此共享。如果共享资源的指定使用者是组织，而资源使用者的账号由组织管理，则无法退出此共享，详情请参见[退出共享](#)。

# 8 配置函数流

## 8.1 创建函数流

函数流是一个面向无服务器计算领域，编排无服务器分布式应用的工作流服务。基于该服务，用户可以通过Low Code以及可视化的方式将多个独立的无服务器函数用顺序、分支、并行等方式轻松编排成一个完整的应用，并提供监控和管理平台，用于诊断和调试应用。

本章节主要介绍如何创建函数流任务和编排函数流任务。您可以根据实际业务场景来创建标准函数流或快速函数流。

- 标准模式面向普通的业务场景，支持长时间任务，支持执行历史持久化和查询，只支持异步调用，在函数流运行记录页面查询执行结果。
- 快速模式面向业务执行时长较短，需要极致性能的场景，只支持流程执行时长低于5分钟的场景，不支持执行历史持久化（比如不支持查询执行节点的历史信息），支持同步和异步调用。通过同步执行函数流接口进行函数流的同步执行，接口直接返回函数流执行结果，同时日志页面查看上报到LTS的函数流执行日志。

### 说明

快速函数流限时免费，欢迎体验！

## 约束与限制

当前仅“华北-北京四、华东-上海二、华南-广州、西南-贵阳一、华东-上海一、亚太-新加坡”区域支持函数流功能。

## 创建函数流

### 操作步骤

1. 登录[函数工作流控制台](#)，左侧导航栏单击“函数流”。
2. 单击“创建标准函数流”或“创建快速函数流”，进入新建函数流编排页面。
3. 在函数流编排区域，通过拖拽组件进行流程编排，详情请参见[函数流组件配置说明](#)。

设计的函数流必须是一个有向无环图，从开始节点出发，开始节点后续必须且只能连接一个节点（除了异常处理和结束节点）；流程必须在某一个节点结束，结束流程有两种形式：

- 流程中存在的节点没有任何后继节点，且后续节点非条件分支，并行分支或开始节点。
  - 流程中存在结束节点，且结束节点后续无其他节点。
4. 流程中的所有节点参数配置完成后，单击右上角的“保存”，进行参数配置，最后单击“确定”。

表 8-1 配置信息

| 参数       | 说明                                                                                                                                       |
|----------|------------------------------------------------------------------------------------------------------------------------------------------|
| 名称       | 输入函数流名称。                                                                                                                                 |
| 企业项目     | 选择企业项目。                                                                                                                                  |
| 日志记录     | 创建快速函数流，保存时需要选择此参数。 <ul style="list-style-type: none"><li>• ALL：为所有事件启用日志记录。</li><li>• ERROR：仅启用错误日志记录。</li><li>• NONE：关闭日志记录。</li></ul> |
| 合并参数     | 将上一个节点的输出与下一个节点的输入合并为输入。                                                                                                                 |
| 支持返回流式数据 | 创建快速函数流时可配置此参数。开启后，函数流执行支持返回流式数据，可参见流式文件处理。<br><b>需函数流中函数节点的代码调用流式数据接口。</b>                                                              |
| 描述       | 输入函数流的简要描述。                                                                                                                              |

表达式说明

函数流的异常处理控制器组件中配置“重试条件(JSONPath)”和条件分支控制器组件中“输入过滤表达式/输出过滤表达式”的JSONPath表达式结构为[JsonPath] + [逻辑运算符] + [对比数据]，如示例：\$.age >= 20。

表 8-2 JSONPath 说明

| 操作符         | 是否支持 | 说明                     |
|-------------|------|------------------------|
| \$          | 是    | 执行查询的root，所有正则表达式由此启动。 |
| @           | 是    | 过滤正在处理的当前位置。           |
| .           | 是    | 子节点。                   |
| [ ( , ) ]   | 是    | 数组索引。                  |
| [start:end] | 是    | 数组切片运算符。               |

| 操作符   | 是否支持 | 说明                 |
|-------|------|--------------------|
| [?()] | 是    | 过滤表达式，表达式必须计算为布尔值。 |

示例

```
{
  "fruits": [ "apple", "orange", "pear" ],
  "vegetables": [{
    "veggieName": "potato",
    "veggieLike": true
  },
  {
    "veggieName": "broccoli",
    "veggieLike": false
  }]
}
```

- 简单取值  
\$.fruits表达式含义：取出fruits下对应的所有value。  
\$.fruits解析结果：["apple","orange","pear"]。
- 简单过滤  
表达式：\$.vegetables[?(@.veggieLike == true)].veggieName。  
表达式含义：取出key值vegetables对应的所有value，并根据过滤条件输出veggieLike为True的veggieName。  
取值结果：[potato]

运算符说明

使用以下数据作为例子中的输入参数：

```
{
  "name" : "apple",
  "weight": 13.4,
  "type": [3,4,6,8],
  "obj": {
    "a" : 1
  }
}
```

表 8-3 逻辑运算符

| 符号 | 作用 | 例子                 | 返回值   | 备注                                       |
|----|----|--------------------|-------|------------------------------------------|
| == | 相等 | \$.name == 'apple' | true  | 支持的数据类型包括：<br>int,float,string, bool,nil |
| != | 不等 | \$.name != 'apple' | false | 支持的数据类型包括：<br>int,float,string, bool,nil |

| 符号  | 作用   | 例子                                   | 返回值   | 备注             |
|-----|------|--------------------------------------|-------|----------------|
| <   | 小于   | \$.weight < 12                       | false | 只支持数字类型        |
| >   | 大于   | \$.weight > 12                       | true  | 只支持数字类型        |
| <=  | 小于等于 | \$.weight <= 13.4                    | true  | 只支持数字类型        |
| >=  | 大于等于 | \$.weight >= 13.4                    | true  | 只支持数字类型        |
| '*' | 通配符  | \$.weight == '*'                     | true  | 只支持在== 比较中使用   |
|     | 或    | \$.name == 'apple'    \$.weight < 12 | true  | 支持使用（ ）的复杂与或逻辑 |
| &&  | 且    | \$.name == 'apple' && \$.weight < 12 | false | 支持使用（ ）的复杂与或逻辑 |

- 字符串格式常量需要使用"包含，例如：'apple'。
- jsonpath表达式中不能出现上述保留字符'=', '!=', '<', '>', '|', '&'。

## 8.2 启动函数流

函数流配置完成后，需要启动函数流才能完成业务功能实现。

### 约束与限制

当函数流的配置有改动后，必须先保存信息，才能启动函数流任务。

### 启动函数流

1. 在函数流编排页面右上角单击“启动”或在函数流列表页面选择“更多 > 启动”，弹出的启动执行页面。
2. 在弹出的启动执行页面，支持定义输入值或者直接启动。此处选择“定义输入值”。（输入定义值必须是JSON格式的内容。）

图 8-1 启动执行配置



3. 单击“开始执行”，页面右上角提示“启动函数流xxx成功”。

## 8.3 函数流组件配置说明

### 8.3.1 配置函数流的 EG 服务组件

函数流支持配置事件网格服务（EG）服务组件，EG节点会发布已配置的事件至指定的EG事件通道，如何创建EG资源请参见[开始使用事件网格](#)。

#### 约束与限制

- EG服务组件当前仅支持华北-北京四。
- 通过数据工坊DWR服务创建的函数流，在函数工作流控制台只能查看，不能编辑和删除，相关操作请在DWR服务中执行。
- 当前允许流程中配置0-10个EG节点。
- 当EG节点连接异常处理节点时，最多可以再连接一个非开始节点和非异常处理节点。
- 当EG节点不连接异常处理节点时，只能连接一个非开始节点。

#### 前提条件

- 已经在事件网格（EG）控制台创建自定义事件源和自定义事件通道，具体创建详情请参见[自定义事件源](#)、[自定义事件通道](#)。

- 使用函数流功能前，请先了解[表达式说明](#)和[运算符说明](#)。

配置说明

单击编辑框中的EG节点进行编辑。配置EG参数，参数说明如[表8-4](#)所示。

图 8-2 EG 节点配置



表 8-4 EG 节点参数说明

| 参数      | 说明                                                                                             |
|---------|------------------------------------------------------------------------------------------------|
| 事件通道    | 事件通道负责接收来自事件源的事件。函数流仅支持编排自定义事件通道：您自行创建的事件通道，用于接收自定义事件源产生的事件。<br>详情请参见 <a href="#">事件通道概述</a> 。 |
| 事件源     | 事件源是事件的来源，函数流仅支持自定义的应用作为事件源，通过自定义的事件通道发布事件到事件网格。<br>详情请参见 <a href="#">事件源概述</a> 。              |
| 事件内容格式  | 参数“事件负载内容”的内容格式。<br>( 目前只支持application/json格式。 )                                               |
| 事件负载内容  | 事件内容。                                                                                          |
| 事件发生主题  | 事件发生的主题或对象，用以标识哪个具体对象发生了当前事件                                                                   |
| 输入过滤表达式 | 通过JSONPath表达式对节点的输入信息进行过滤。                                                                     |
| 输出过滤表达式 | 通过JSONPath表达式对节点的输出信息进行过滤。                                                                     |

8.3.2 配置函数流的函数服务组件

函数流支持配置函数服务组件，通过该组件可以关联已创建的函数，进而实现业务需求。

约束与限制

- 通过数据工坊DWR服务创建的函数流，在函数工作流控制台只能查看，不能编辑和删除，相关操作请在DWR服务中执行。
- **配置的函数节点返回的数据格式必须是json格式，否则会解析失败。**
- 同步调用不支持长时间函数，最大执行时长为15分钟。
- 异步调用支持长时间函数，单函数节点的最大执行时长以函数服务支持的最大执行时长为准。
- 当前允许流程中配置0-99个函数节点。
- 当函数连接异常处理节点时，最多可以再连接一个非开始节点和非异常处理节点。
- 当函数不连接异常处理节点时，只能连接一个非开始节点。

前提条件

- 已经在函数工作流控制台创建函数，如何创建请参见[创建空白函数](#)。
- 使用函数流功能前，请先了解[表达式说明](#)和[运算符说明](#)。

配置说明

单击编辑框中函数节点进行编辑。配置函数参数，参数说明如[表8-5](#)所示。

图 8-3 函数节点配置

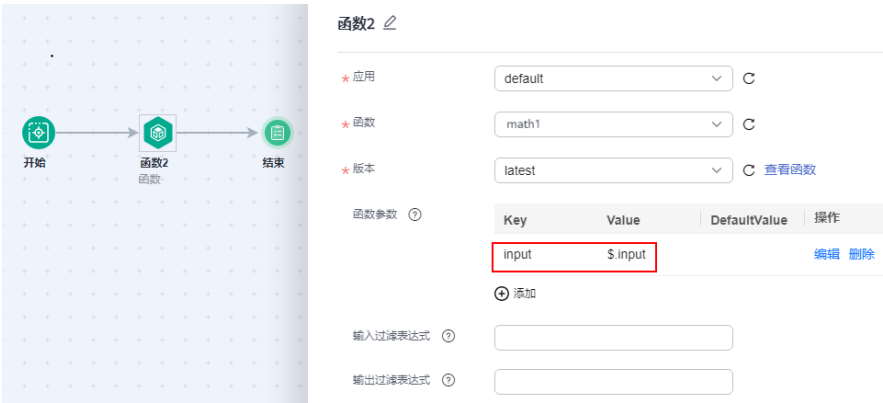


表 8-5 函数参数说明

| 参数   | 说明                                                          |
|------|-------------------------------------------------------------|
| 应用   | 函数所属应用，用户创建函数时可以进行分组，每个函数应用下面可以创建多个函数，在函数创建时可以指定其归属于某个函数应用。 |
| 函数   | FunctionGraph中对应的函数。                                        |
| 版本   | FunctionGraph中函数对应的版本。                                      |
| 调用方式 | 创建标准函数流时，需要配置此参数。函数流执行函数节点时的调用方式，默认为同步调用。                   |

| 参数      | 说明                                                                                                                                                                                                                      |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 函数参数    | 函数执行时的入参，以键值对形式传入，'值'可自定义常量或通过JSONPath表达式从启动时的'定义输入值'和前置节点的输出信息中获取，请谨慎输入密码等敏感信息，敏感信息请自行加密，防止信息泄露！<br>流程中以json格式作为body参数在执行时传入函数。<br>Key：填写参数。<br>Value：填写参数值。<br>DefaultValue：设置默认值，参数未获取到值时，默认获取默认值。<br>操作：编辑或删除设置的参数。 |
| 输入过滤表达式 | 通过JSONPath表达式对节点的输入信息进行过滤。                                                                                                                                                                                              |
| 输出过滤表达式 | 通过JSONPath表达式对节点的输出信息进行过滤。                                                                                                                                                                                              |
| 开启容灾函数  | 开启后，当前节点名称不能与其他函数节点名称重复。                                                                                                                                                                                                |

## 函数服务组件示例

本章节以函数的运行时语言使用Python 3.9为例介绍，代码及功能介绍如下所示。如何创建函数请参见[创建空白函数](#)。

函数功能：函数执行返回result的值为函数调用事件内的input输入值。

```
import json
def handler (event, context):
    input = event.get('input',0)
    return {
        "result": input
    }
```

### 8.3.3 配置函数流的子流程控制器组件

函数工作流服务可以把一些“函数流”作为“子流程”组合成一个新的函数流，这样可以抽取一些函数流作为公共流进行使用，减少函数流的重复编排。

## 约束与限制

通过数据工坊DWR服务创建的函数流，在函数工作流控制台只能查看，不能编辑和删除，相关操作请在DWR服务中执行。

## 前提条件

- 已经在函数工作流控制台创建函数流，可作为配置子流程使用。
- 使用函数流功能前，请先了解[表达式说明](#)和[运算符说明](#)。

配置说明

在函数编排页面，单击“子流程”节点，参考表8-6配置子流程参数。

表 8-6 子流程参数

| 参数        | 说明                         |
|-----------|----------------------------|
| 选择子流程     | 下拉列表中会展示所有已创建的函数流。         |
| 是否等待子流程完成 | 默认开启。                      |
| 输入过滤表达式   | 通过JSONPath表达式对节点的输入信息进行过滤。 |
| 输出过滤表达式   | 通过JSONPath表达式对节点的输出信息进行过滤。 |

8.3.4 配置函数流的并行分支控制器组件

函数流的并行分支控制器组件支持函数流同时执行多个分支任务，并可根据分支执行结束后控制下一步流程。

约束与限制

- 通过数据工坊DWR服务创建的函数流，在函数工作流控制台只能查看，不能编辑和删除，相关操作请在DWR服务中执行。
- 并行分支控制器组件的后继节点允许连接1-20个节点（除了异常处理、开始节点和结束节点），至少连接一个节点。

前提条件

使用函数流功能前，请先了解表达式说明和运算符说明。

配置说明

在函数编排页面，单击“并行分支”节点，配置并行分支参数。

图 8-4 并行分支节点配置



表 8-7 并行分支参数

| 参数       | 说明                                                                                                                                                                             |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 分支执行完成条件 | 下拉列表可选择如下3种条件： <ul style="list-style-type: none"><li>• 所有分支执行完成。</li><li>• 一个分支执行完成。</li><li>• 指定数目分支执行完成。<br/>当选择该条件时，需要配置“指定分支执行完成数目”参数。（最大分支数目20，指定分支数目不能大于总分支数。）</li></ul> |
| 输入过滤表达式  | 通过JSONPath表达式对节点的输入信息进行过滤。                                                                                                                                                     |
| 输出过滤表达式  | 通过JSONPath表达式对节点的输出信息进行过滤。                                                                                                                                                     |
| 结果输出路径   | 输入并行分支执行结果输出位置，输入值作为key，并行分支执行结果作为value，以json形式输出。若未填写，默认输出路径为：result。                                                                                                         |

### 8.3.5 配置函数流的开始控制器组件

函数流的开始控制器组件是用于标识流程的开始。

#### 约束与限制

- 通过数据工坊DWR服务创建的函数流，在函数工作流控制台只能查看，不能编辑和删除，相关操作请在DWR服务中执行。
- 每个函数流必须以“开始节点”作为流程开始且只能有一个“开始节点”，同时以“结束节点”作为流程结束。
- 开始节点后面必须接1个节点，且后续节点类型不能是结束节点或者异常处理。
- “开始节点”只能配置触发器：
  - 函数流触发器当前支持**APIG触发器(专享版)**、**定时触发器**、**SMN触发器**、**EG触发器**。
  - 当前允许流程中配置0-10个触发器。
  - 触发器必须配置在开始节点内。
  - 触发器不允许连接其他任何节点，也不允许被其他节点连接。

#### 定时触发器配置说明

单击“开始”节点，在右侧弹出的属性页面添加触发器，触发器类型选择“定时触发器”，填写触发器配置信息，如表8-8所示。

图 8-5 定时触发器



表 8-8 定时触发器配置信息

| 参数      | 说明                                                            |
|---------|---------------------------------------------------------------|
| 触发器类型   | 选择“定时触发器”。                                                    |
| 触发规则    | 定时触发器的触发规则，当前只支持Cron表达式。                                      |
| Cron表达式 | 用于表示任务调度的表达式，能够表示特定周期进行的特定的时间、日期等。具体请参见Cron表达式规则。             |
| 附加信息    | 附加信息为json格式，输入必须包含input，在input内输入需要的json体。input的内容会作为流程的输入参数。 |

APIG 触发器(专享版)配置说明

- 函数流APIG触发器目前仅支持IAM认证方式。
- 前提条件：需要预先创建APIG专享版实例，具体请参见[购买专享版实例](#)。

在函数流编排页面，单击“开始”节点，在右侧弹出的属性页面添加触发器，触发器类型选择“APIG触发器(专享版)”，填写触发器配置信息，如表8-9。



SMN 触发器配置说明

在函数流编排页面，单击“开始”节点，在右侧弹出的属性页面添加触发器，触发器类型选择“SMN触发器”、“主题名称”选择已创建的SMN主题（[创建主题](#)）。

图 8-7 SMN 触发器



EG 触发器配置说明

在函数流编排页面，单击“开始”节点，在右侧弹出的属性页面添加触发器，触发器类型选择“EG触发器”，填写触发器配置信息，如[表8-10](#)所示。

图 8-8 EG 触发器



表 8-10 定时触发器配置信息

| 参数    | 说明                                                 |
|-------|----------------------------------------------------|
| 触发器类型 | 选择“EG触发器”。                                         |
| 提供方   | EG事件源提供方，支持华为云和自定义两种方式。                            |
| 事件通道  | 当“提供方”选择“自定义”时，配置此参数。事件通道负责接收来自事件源的事件来触发函数流执行。     |
| 事件源   | 事件源是事件的来源，负责将华为云服务、自定义应用等应用程序生产的事件发布到事件网格。         |
| 事件类型  | 当“提供方”选择“华为云”时，配置此参数。可以指定事件类型进行过滤，满足事件类型才会触发函数流执行。 |

### 8.3.6 配置函数流的异常处理控制器组件

异常处理控制器组件用于控制函数执行失败后的下一步流程，当函数流里面的函数执行异常时，可以通过“异常处理”来处理执行失败的函数并可添加重试。函数执行失败可分为多种情况：函数执行异常；函数内部业务失败并在返回内容中添加了错误码，例如status，200代表成功，500和404等代表失败。

#### 约束与限制

- 通过数据工坊DWR服务创建的函数流，在函数工作流控制台只能查看，不能编辑和删除，相关操作请在DWR服务中执行。
- 异常处理控制器组件后面可以接0-10个节点，后继节点不能是开始节点、结束节点和异常处理节点。

#### 配置说明

在函数流编排页面，单击“异常处理”节点，在右侧弹出页面中配置异常处理参数。

图 8-9 配置异常处理

异常处理

★ 是否重试

☒

★ 重试条件(JSONPath)

重试间隔(1-30秒)

最大重试次数(1-8)

表 8-11 异常处理参数配置

| 参数             | 说明                                                                                                                         |
|----------------|----------------------------------------------------------------------------------------------------------------------------|
| 是否重试           | 默认关闭。                                                                                                                      |
| 重试条件(JSONPath) | 当开启“是否重试”后，需要配置此项。该参数用以判断满足重试条件时，则按照设置的重试间隔和最大重试次数进行重试。当最大重试次数执行完后仍满足重试条件，则执行其后继节点；在最大重试次数执行期间或达到最大执行次数后不满足重试条件时，执行其他分支节点。 |
| 重试间隔(1-30秒)    | 当开启“是否重试”后，需要配置此项。默认重试间隔1秒。                                                                                                |
| 最大重试次数(1-8)    | 当开启“是否重试”后，需要配置此项。默认重试次数3次。                                                                                                |

8.3.7 配置函数流的循环控制器组件

循环控制器组件可以用来对输入数组中每个元素进行循环处理，每次循环会执行一次循环内部的子流程或函数。

约束与限制

- 通过数据工坊DWR服务创建的函数流，在函数工作流控制台只能查看，不能编辑和删除，相关操作请在DWR服务中执行。
- 循环节点内部子流程需要满足如下规则：
  - 只能有一个起始节点（没有前驱节点），起始节点只能使用函数，时间等待节点。
  - 循环节点内部只允许编排函数、时间等待和异常处理节点。

前提条件

- 已经在函数工作流控制台创建函数，如何创建请参见[创建空白函数](#)。
- 使用函数流功能前，请先了解[表达式说明](#)和[运算符说明](#)。

配置说明

在函数流编排区域，单击循环组件进行配置。

图 8-10 配置循环组件

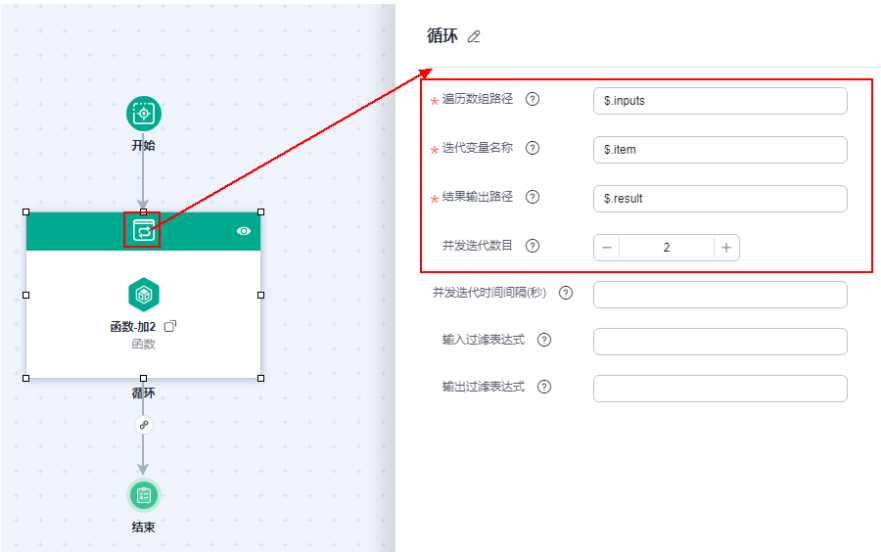


表 8-12 循环组件参数配置

| 参数     | 说明                                                                         |
|--------|----------------------------------------------------------------------------|
| 遍历数组路径 | 通过JSONPath表达式从输入的数据中获取数组。例如此时有输入:{"arr":[1,2,3]}，则可在“遍历数组路径”填入\$.arr获取该数组。 |
| 迭代变量名称 | 在遍历循环数组时作为数组中每一个元素的临时变量来承载循环的值，例如使用item作为临时变量，则设置为\$.item。                 |

| 参数       | 说明                                                                          |
|----------|-----------------------------------------------------------------------------|
| 结果输出路径   | 通过JSONPath表达式将结果指定输出到一个json中，例如在“结果输出路径”填入\$.result，则输出{"result": [2,3,4]}。 |
| 并发迭代数目   | 限制0-100，0代表并发拉起的数目无限制。                                                      |
| 并发迭代时间间隔 | 每次迭代间隔的时间。                                                                  |
| 输入过滤表达式  | 通过JSONPath表达式对节点的输入信息进行过滤。                                                  |
| 输出过滤表达式  | 通过JSONPath表达式对节点的输出信息进行过滤。                                                  |

### 8.3.8 配置函数流的时间等待控制器组件

时间等待控制器组件可以用来控制当前子流程或函数在指定时间延迟后再调用下一个子流程或函数。

#### 约束与限制

- 通过数据工坊DWR服务创建的函数流，在函数工作流控制台只能查看，不能编辑和删除，相关操作请在DWR服务中执行。
- 时间等待控制器组件后面可以连接0个或1个节点，节点类型不能是开始节点和异常处理节点。

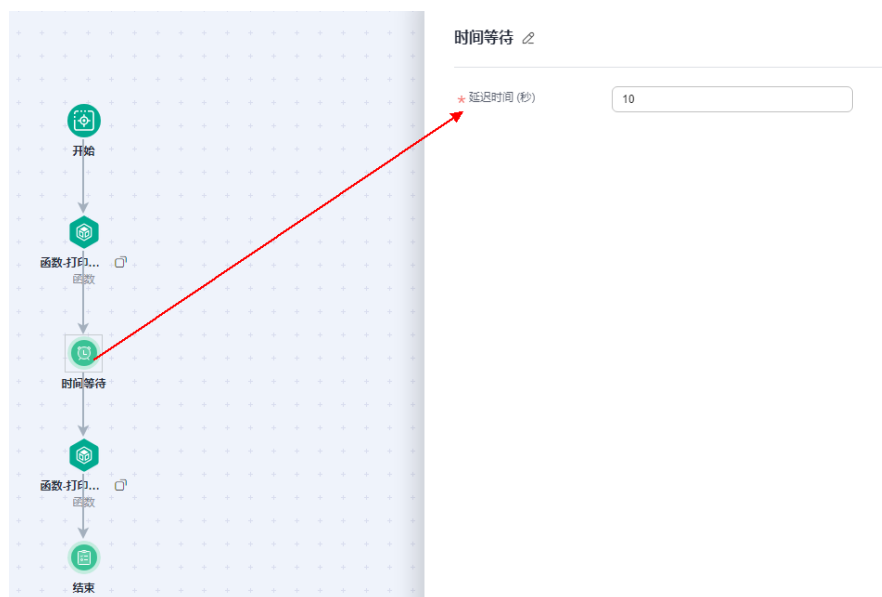
#### 前提条件

使用函数流功能前，请先了解[表达式说明](#)和[运算符说明](#)。

#### 配置说明

在函数流编排区域，单击时间等待节点，修改“延迟时间（秒）”参数（输入值必须在0.001到86400之间）。

图 8-11 配置时间等待节点



8.3.9 配置函数流的服务控制器组件

服务控制器组件可以将多个函数操作合并成一个原子节点进行管理。  
函数流的服务控制器组件支持两种执行模式：串行模式和并行模式。

约束与限制

- 通过数据工坊DWR服务创建的函数流，在函数工作流控制台只能查看，不能编辑和删除，相关操作请在DWR服务中执行。
- 服务控制器组件由多个函数节点组成，后续节点可以是结束节点或异常处理节点。

前提条件

使用函数流功能前，请先了解[表达式说明](#)和[运算符说明](#)。

配置说明

在函数流编排区域，单击服务节点，配置服务组件参数。

图 8-12 串行模式



图 8-13 并行模式



表 8-13 服务组件参数配置

| 参数      | 说明                                                                                                                                    |
|---------|---------------------------------------------------------------------------------------------------------------------------------------|
| 执行模式    | 下拉列表可选择“串行模式”和“并行模式”。 <ul style="list-style-type: none"><li>串行模式：当选择该模式时，服务组件中函数节点需要连线。</li><li>并行模式：当选择该模式时，服务组件中函数节点无需连线。</li></ul> |
| 输入过滤表达式 | 通过JSONPath表达式对节点的输入信息进行过滤。                                                                                                            |
| 输出过滤表达式 | 通过JSONPath表达式对节点的输出信息进行过滤。                                                                                                            |

## 8.3.10 配置函数流的条件分支控制器组件

函数流的条件分支控制器组件可以根据设置的条件判断是否执行其他分支。

### 约束与限制

- 通过数据工坊DWR服务创建的函数流，在函数工作流控制台只能查看，不能编辑和删除，相关操作请在DWR服务中执行。
- 条件分支控制器组件后面可以连接2-20个后继节点，后继节点类型不能为开始节点、结束节点和异常处理节点。

### 前提条件

使用函数流功能前，请先了解[表达式说明](#)和[运算符说明](#)。

### 配置说明

在函数流编排区域，配置“条件分支”节点参数和连线参数。

图 8-14 条件分支节点配置

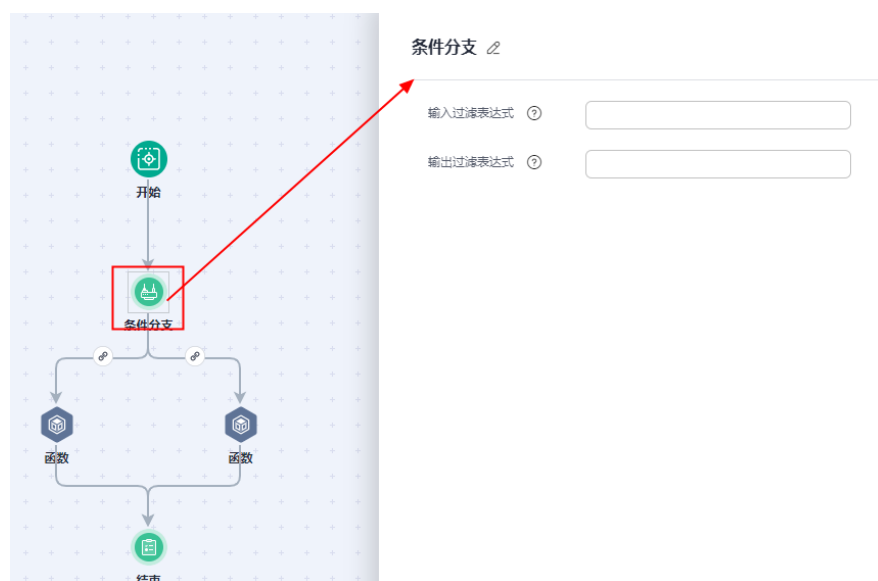


图 8-15 连线配置



表 8-14 条件分支节点参数配置

| 参数      | 说明                         |
|---------|----------------------------|
| 输入过滤表达式 | 通过JSONPath表达式对节点的输入信息进行过滤。 |
| 输出过滤表达式 | 通过JSONPath表达式对节点的输出信息进行过滤。 |

表 8-15 连线参数配置

| 参数   | 说明                                                                                              |
|------|-------------------------------------------------------------------------------------------------|
| 分支类型 | 下拉列表可选择“条件分支”和“默认分支”。                                                                           |
| 表达式  | 当“分支类型”选择“条件分支”时，需要配置该项。该参数通过表达式确定当前分支是否满足执行条件。例如，\$.input<2，表示当如输入值小于2时，执行该条件分支；大于等于2时，执行默认分支。 |

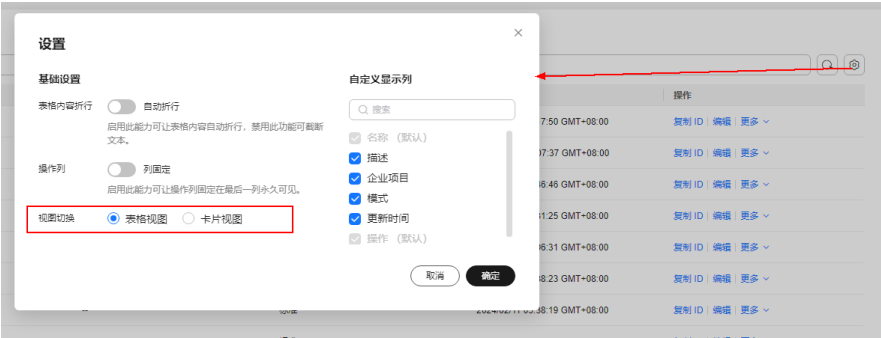
## 8.4 管理函数流

函数流创建后，用户可以查看函数流的配置、执行历史、执行日志等信息，可以对失败的流程进行重试，也可以终止正在执行的流程。

### 查看函数流

- 步骤1 登录[函数工作流控制台](#)，左侧导航栏单击“函数流”。
- 步骤2 在函数流列表页面，支持设置函数流任务卡片显示和列表显示。

图 8-16 设置显示方法



- 卡片显示

图 8-17 卡片显示



- 列表显示

图 8-18 列表显示



**步骤3** 在函数流页面，可以查看所有已创建的函数流任务，您可以执行如表8-16中的管理操作。

表 8-16 操作说明

| 操作项 | 说明                                                                                                                                                |
|-----|---------------------------------------------------------------------------------------------------------------------------------------------------|
| 编辑  | 在函数流任务卡片或列表中，单击“编辑”，进入函数流编辑页面，更新修改函数流任务信息。                                                                                                        |
| 启动  | 在函数流任务卡片或列表中，单击“更多 > 启动”，启动该函数流任务。                                                                                                                |
| 删除  | <ul style="list-style-type: none"><li>• 支持单个删除任务：在函数流任务卡片或列表中，单击“更多 &gt; 删除”，删除该函数流任务。</li><li>• 批量删除任务：勾选多个函数流任务，单击左上角的“删除”，删除所选函数流任务。</li></ul> |

**步骤4** 单击某一个函数流任务名称，查看任务详情。

- 查看任务基本信息  
在“基本信息”页签，查看任务名称、ID、更新时间和创建时间等信息。
- 查看任务运行记录（仅支持标准函数流）  
在“运行记录”页签，查看任务的执行历史记录、输入值、输出值、节点日志等信息。

若需要修改任务信息，单击右上角的“函数流设计”，进入函数流编辑页面，更新修改任务。

- 查看任务日志（仅支持快速函数流）  
在“日志”页签，单击“请求ID”链接，可以查看执行结果的日志信息，同时支持一次最多下载5000条日志。
- 查看函数流指标  
在“监控”页签，查看任务的调用次数、运行时间、错误次数、运行中、被拒绝次数信息。监控指标说明如表8-17所示。

表 8-17 函数流监控指标

| 指标    | 单位 | 说明                                              |
|-------|----|-------------------------------------------------|
| 调用次数  | 次  | 函数流总的调用请求数，包含了正确、错误和运行中的调用。异步函数流在请求被系统执行时才开始计数。 |
| 运行时间  | 毫秒 | 时间段内单次函数流执行平均的运行时间。                             |
| 错误次数  | 次  | 指发生异常请求的函数流不能正确执行完，会计入错误次数。                     |
| 运行中   | 个  | 正在运行中的函数流的数量。                                   |
| 被拒绝次数 | 次  | 指被限流无法执行函数流的次数。                                 |

----结束

查询标准函数流的执行历史

- 步骤1 在函数流列表页面，单击标准函数流名称，然后选择“运行记录”页签。
- 步骤2 左侧为执行历史记录列表，展示最近100次执行记录。

图 8-19 执行历史记录



- 单击左侧的执行记录，中间画布展示流程的执行结果，如果节点执行成功，图标背景为绿色，如果执行失败背景为红色。
- 画布下方输入输出展示区默认展示流程的输入值和输出值，单击上方任意节点，即可展示节点的输入值和输出值。（对于函数流及函数流各个节点，若输出值中有字段的值为null，则该字段会被直接过滤，不予展示。）

图 8-20 输入输出展示区



- 在“执行失败”的记录右侧单击图标，重试成功后会生成一条执行记录。
- 单击“执行中”任务右侧停止图标，终止成功后流程会进入取消状态。
- “节点日志”区域展示流程从开始到结束所有节点的执行记录。

图 8-21 节点日志

| 节点日志 |      |                                        |        |                               |  |
|------|------|----------------------------------------|--------|-------------------------------|--|
| 节点名称 | 状态   | 请求ID                                   | 耗时     | 开始时间                          |  |
| 函数   | 执行失败 | 7c9d9e9f-9b9f-4b9f-9b9f-9b9f-9b9f-9b9f | 451 ms | 2024/04/15 18:55:18 GMT+08:00 |  |

----结束

## 查询快速函数流的执行日志

**步骤1** 在函数流列表页面，单击快速函数流名称，进入函数流详情页，然后选择“日志”页签。

**步骤2** 单击“请求ID”链接查看执行日志详情。支持最近1小时、最近1天、最近3天及自定义时间过滤日志。

----结束

# 9 通过应用中心部署函数应用

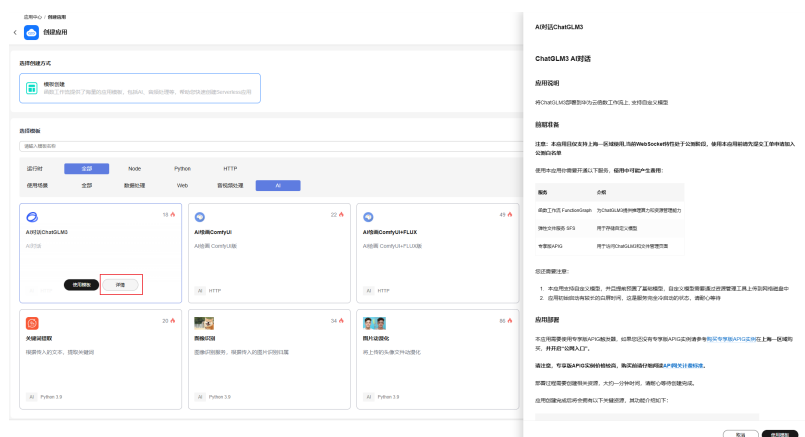
应用中心使用资源编排服务来部署应用需要的周边资源（包含函数、委托、触发器等），使这些资源相互配合，共同执行任务。

## 应用中心模板介绍

FunctionGraph应用中心提供了多种热门应用的函数模板，使用模板创建函数时，可实现函数代码、环境变量、函数触发器配置等的自动填充，助您快速构建函数应用。

支持根据函数“运行时”和“使用场景”筛选函数模板，如图9-1所示单击函数模板的“详情”可查看该模板的使用描述以及注意事项。

图 9-1 模板详情



## 应用中心计费说明

使用FunctionGraph应用中心创建应用不收取费用，但是使用不同应用模板进行应用创建的过程中，可能涉及创建其他付费的云服务资源，关于其他云服务的计费详情，请参考具体云服务的计费说明。

## 约束与限制

当前仅“华北-北京四”、“华东-上海一”和“西南-贵阳一”区域支持应用中心功能。

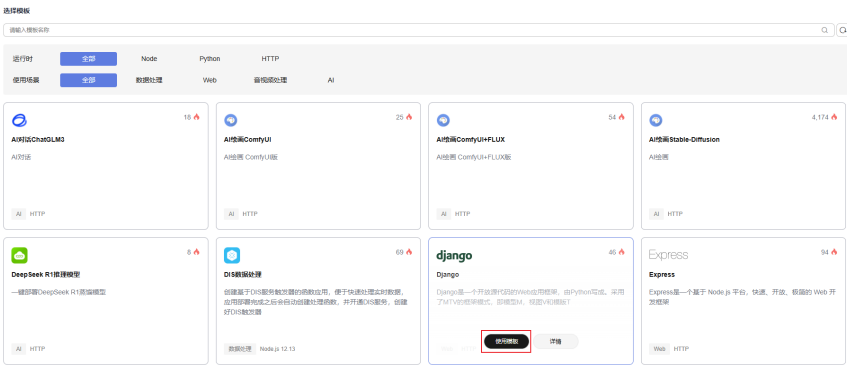
前提条件

如果函数工作流服务需要访问云日志服务LTS、虚拟私有云VPC等其他云服务进行协同工作，则需[创建函数委托](#)并授权函数工作流服务访问所需云服务的权限。

创建应用

- 步骤1 登录[函数工作流控制台](#)，在左侧导航栏选择“应用中心”。
- 步骤2 单击右上方的“创建应用”，进入“选择模板”页面。
- 步骤3 以创建“Django”应用为例进行介绍，确定好模板后，如[图9-2](#)所示单击“使用模板”进入应用配置页面。

图 9-2 选择应用模板



- 步骤4 在“创建应用”界面，参考[表9-1](#)填写基本信息，配置完成后单击“立即创建”。

表 9-1 Django 数据处理模板参数说明

| 参数    | 说明                                                                                                | 取值样例                |
|-------|---------------------------------------------------------------------------------------------------|---------------------|
| 模板    | 默认展示已选择的函数应用模板。<br>如需更换函数模板，请单击“重新选择”。                                                            | Django              |
| 区域    | 函数应用创建的区域。<br>默认选择控制台所在区域，如需修改，需返回函数工作流控制台切换左上方的区域选项。                                             | 华东-上海一              |
| 应用名称  | 自定义填写函数应用名称。<br>可包含字母、数字、下划线和中划线，以大/小写字母开头，以字母或数字结尾，长度在2-60个字符之间。                                 | django-framework    |
| 运行时语言 | 函数应用模板内置的运行时语言，不支持修改。                                                                             | http                |
| 委托    | 选择函数工作流服务的委托。<br>选择“未使用任何委托”时，系统将会使用名称为“fgs-app-adminagency”的委托以保证应用正常创建（该委托不存在时将在单击“立即创建”时提示创建）。 | fgs-app-adminagency |

| 参数     | 说明                                                                  | 取值样例    |
|--------|---------------------------------------------------------------------|---------|
| 创建仓库   | 开启时，应用创建时会为您在代码托管服务创建代码仓库，之后您可以通过更新代码并部署来更新您的应用。                    | 开启      |
| 仓库所属项目 | 代码仓库从属于需求管理服务的项目。应用创建时会在您选择的项目下创建代码仓库；如果未选择项目，则会新建一个项目并在该项目下创建代码仓库。 | -       |
| 描述     | 自定义输入关于本应用的描述，最长支持1024个字符。                                          | -       |
| APIG实例 | 选择可用的APIG专享版实例，如无实例，可单击右侧“创建实例”进入创建页面。                              | apig-fg |

**步骤5** 创建成功后如图9-3所示，将自动创建所需的函数资源及API网关资源，单击“物理资源名称”下的蓝色链接，可跳转至资源详情页。

图 9-3 应用创建成功

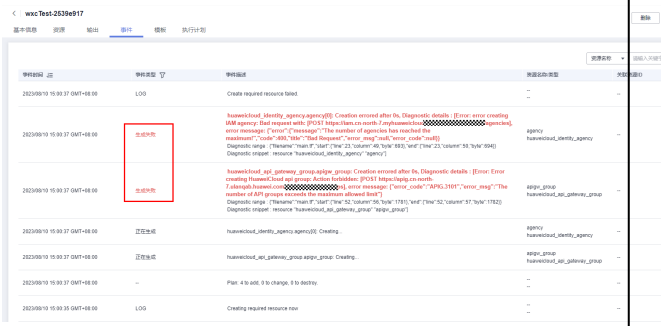


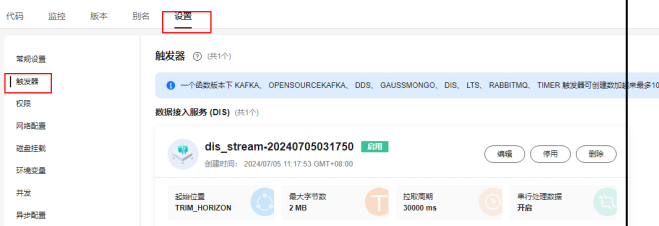
----结束

应用中心常见问题排查方法

表 9-2 常见问题排查方法

| 常见问题                                                                                                                                       | 排查方法                                                                                                                                       |
|--------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| <p>创建代码仓库失败，提示如图9-4所示。</p> <p><b>图 9-4 创建代码仓失败</b></p>  | <p>请检查您账户是否开通了CodeArts服务，具体请参考<a href="#">登录软件开发生产线（CodeArts）</a>检查并开通服务。</p> <p>如果开通后问题仍存在，请<a href="#">提交工单</a>联系华为云函数工作流服务工程师进一步帮助。</p> |

| 常见问题                                                                                                                                  | 排查方法                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|---------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>堆栈部署失败，提示如图9-5所示。</p> <p><b>图 9-5 堆栈部署失败</b></p>  | <p>1. 单击应用程序名称链接进入应用详情页，单击“资源栈名称”链接，跳转到资源编排服务页面。</p> <p><b>图 9-6 总览页</b></p>  <p>2. 在资源编排服务页面，单击“查看失败原因”链接，查看具体问题。</p> <p><b>图 9-7 资源编排服务页面</b></p>  <p><b>图 9-8 查看失败原因</b></p>  |

| 常见问题                                                                                                                                | 排查方法                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>您的权限不足，提示如图9-9所示。</p> <p><b>图 9-9 权限不足</b></p>  | <p>首次创建应用时报错“权限不足”，请配置参考<a href="#">配置函数的委托权限</a>配置当前账号的委托权限，然后重试。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <p>删除失败，提示如图9-10所示。</p> <p><b>图 9-10 删除失败</b></p>  | <ol style="list-style-type: none"><li>单击应用程序名称链接进入应用详情页，单击“资源栈名称”链接，跳转到资源编排服务页面。</li><li>单击“查看失败原因”链接，查看具体问题。以图9-10提示为例，删除失败原因是API组中包含API导致删除失败。</li><li>重新进入应用详情页，单击函数服务的“物理资源名称/ID”链接，进入函数详情页。</li></ol> <p><b>图 9-11 单击函数服务链接</b></p>  <ol style="list-style-type: none"><li>选择“设置”-&gt;“触发器”，查看API名称。</li></ol> <p><b>图 9-12 查看 API 名称</b></p>  <ol style="list-style-type: none"><li>进入API网关服务控制台，将该API下线并删除，再返回函数应用重试删除。</li></ol> |

相关文档

除使用控制台外，函数工作流支持通过API的方式管理函数应用中心，详情请参见[函数应用中心API](#)。

# 10 通过 KooCLI 管理函数

KooCLI工具支持FunctionGraph服务，用户可以通过CLI工具执行函数管理，别名管理，触发器管理函数调用等操作。

## 约束与限制

- 当通过API访问公有云系统时，需要使用访问密钥（AK/SK）进行身份认证并对请求进行加密，确保请求的机密性、完整性和请求双方身份的正确性。请妥善保存config.yaml配置文件，确保访问密钥不被非法使用。
- 当发现访问密钥被非法使用（包括丢失、泄露等情况），可以自行删除或者通知管理员重置访问密钥，重新配置。
- 删除的访问密钥将无法恢复。

## 获取 CLI 工具

CLI工具支持64位的Linux x86系统及64位Windows系统，以及Mac系统，具体下载地址如表10-1所示。

表 10-1 下载地址

| 系统      | 软件包及校验文件                                           | 参考文档                     |
|---------|----------------------------------------------------|--------------------------|
| Linux   | 各系统类型软件包及校验文件下载请参见 <a href="#">CLI软件包和校验文件下载</a> 。 | <a href="#">KooCLI介绍</a> |
| Windows |                                                    |                          |
| Mac     |                                                    |                          |

## 安装 KooCLI 命令行工具

- 安装KooCLI命令行工具，请参见[在Linux系统上安装KooCLI](#)。
- 获取访问密钥（Access Key ID和Secret Access key）
  - 如果您有登录密码，可以登录控制台，可以在我的凭证中获取自己的访问密钥AK/SK。请参见：[新增访问密钥](#)。可以下载得到AK/SK文件，文件名一般为：credentials.csv。如下图所示，文件包含了用户名称（User Name），AK（Access Key Id），SK（Secret Access Key）。

图 10-1 credentials.csv 文件内容

| A         | B             | C                 |
|-----------|---------------|-------------------|
| User Name | Access Key Id | Secret Access Key |
|           | CI            | PI zr17 5uCy      |

- 如果您没有登录密码，不能登录控制台，在访问密钥异常丢失或者需要重置时，可以请账号管理员在IAM中生成您的访问密钥，并发送给您。请参见：[管理IAM用户访问密钥](#)

3. 获取Region Name，请参见：[地区和终端节点](#)。

图 10-2 获取区域

| 区域名称     | 区域         |
|----------|------------|
| 非洲-约翰内斯堡 | af-south-1 |
| 华北-北京四   | cn-north-4 |

4. 初始化KooCLI命令行工具  
执行如下命令初始化KooCLI命令行工具：

```
hcloud configure init
```

需要输入Access Key ID、Secret Access key和Region Name，初始化成功如[图3](#)所示。

图 10-3 初始化成功

```
[root@ecs-function-deploy ~]# hcloud configure init
开始初始化配置,其中"Secret Access Key"输入内容匿名化处理,获取参数可参考'https://.../usermanual-hcli/hcli_09.ht
Access Key ID [required]:
Secret Access Key [required]: *****
Region Name:
*****
*****
*****      初始化配置成功      *****
*****
```

5. 执行如下命令，查看FunctionGraph支持命令。如[图4](#)所示，“Available Operations”中返回了FunctionGraph服务支持的operation列表。  
hcloud FunctionGraph --help

图 10-4 FunctionGraph 服务支持的 operation 列表

```
[root@ ~]# hcloud FunctionGraph --help
KooCLI(Koo Command Line Interface) Version 3.2.8 Copyright(C) 2020-2022 www.huaweicloud.com

Usage:
  hcloud FunctionGraph <operation> --param1=value1 --param2=value2 ...

Service:
  FunctionGraph

Available Operations:
  AsyncInvokeFunction          InvokeFunction          ShowLtsLogDetails
  AsyncInvokeReservedFunction ListDependencies      ShowTenantMetric
  BatchDeleteFunctionTriggers ListEvents            ShowTracing
  BatchDeleteWorkflows        ListFunctionAsyncInvocations ShowVersionAlias
  CancelAsyncInvocation        ListFunctionAsyncInvokeConfig ShowWorkflow
  CreateDependency             ListFunctionStatistics  ShowWorkflowMetric
  CreateEvent                  ListFunctionTriggers    ShowWorkflowExecution
  CreateFunction                ListFunctionVersions    StartSyncWorkflowExecution
  CreateFunctionTrigger         ListFunctions            StartWorkflowExecution
  CreateFunctionVersion         ListQuotas               StopWorkflow
  CreateVersionAlias            ListStatistics           UpdateDependency
  CreateWorkflow                ListVersionAliases       UpdateEvent
  DeleteDependency              ListWorkflowExecutions   UpdateFunctionAsyncInvokeConfig
  DeleteEvent                   ListWorkflows            UpdateFunctionCode
  DeleteFunction                RetryWorkflow             UpdateFunctionConfig
  DeleteFunctionAsyncInvokeConfig ShowDependency            UpdateFunctionMaxInstanceConfig
  DeleteFunctionTrigger         ShowEvent                 UpdateFunctionReservedInstances
  DeleteVersionAlias            ShowFunctionAsyncInvokeConfig UpdateTracing
  EnableLtsLogs                 ShowFunctionCode          UpdateTrigger
  ExportFunction                 ShowFunctionConfig        UpdateVersionAlias
  ImportFunction                 ShowFunctionTrigger        UpdateWorkflow

''' hcloud FunctionGraph <operation> --help''' ''' operation'''
```

执行如下命令，查看“调用函数”操作的帮助信息。如图5所示，返回如下信息。  
hcloud FunctionGraph InvokeFunction --help

图 10-5 “调用函数”操作的帮助信息

```
[root@ ~]# hcloud FunctionGraph InvokeFunction --help
KooCLI(Koo Command Line Interface) Version 3.2.8 Copyright(C) 2020-2022 www.huaweicloud.com

Service:
  FunctionGraph

Description:
  .....

Method:
  POST

Params:
  --cli-region
    required string ..... cli-region
  --function_urn
    required string path ..... URN, ..... FunctionGraph .....
  --project_id
    required string path ..... ID, ..... cli-project-id
  --(*)
    required object body
  --X-CFF-Request-Version
    optional string header ..... v0,v1
  v0: .....
  v1: ..... json, sdk .....
  --X-Cff-Log-Type
    optional string header ..... :tail( ..... 4K ..... ), ..... ( ..... )
```

函数调用

调用函数首先需要获取函数的urn，如图10-6所示。

图 10-6 获取函数 URN



• 同步调用

执行同步调用示例命令，参数说明请参见表1。

```
hcloud FunctionGraph InvokeFunction --cli-region="ap-southeast-1" --X-Cff-Log-Type="tail" --X-CFF-Request-Version="v1" --function_urn="urn:fss:cn-east-3:*****:function:default:hcloud-invoke:latest" --project_id="*****" --key="value"
```

表 10-2 参数说明

| 参数                    | 是否必须 | 描述                                                                                                           |
|-----------------------|------|--------------------------------------------------------------------------------------------------------------|
| --cli-region          | 是    | 调用函数所在region。                                                                                                |
| --function_urn        | 是    | 函数的URN（Uniform Resource Name），唯一标识函数。                                                                        |
| --project_id          | 是    | 租户Project ID。                                                                                                |
| --X-Cff-Log-Type      | 否    | tail（header里面返回函数执行后的4K日志），或者为空（不返回日志）。                                                                      |
| X-CFF-Request-Version | 否    | 返回体格式：取值v0，v1。 <ul style="list-style-type: none"><li>v0：默认返回文本格式。</li><li>v1：默认返回json格式，sdk需要使用此值。</li></ul> |
| Body                  | 是    | 通过--key="value"方式指定请求体，其对应的json结构：{ "key":"value"}。                                                          |

输出结果如图2所示，输出的响应参数请参见表2。

图 10-7 输出结果



表 10-3 响应参数

| 参数         | 参数类型    | 描述       |
|------------|---------|----------|
| request_id | String  | 请求ID     |
| result     | String  | 函数执行结果   |
| log        | String  | 函数执行返回日志 |
| status     | Integer | 函数执行返回状态 |
| error_code | String  | 错误码      |

- 异步调用  
执行异步调用示例命令，参数说明请参见表3。

```
hcloud FunctionGraph AsyncInvokeFunction --cli-region="cn-east-3" --function_urn="urn:fss:cn-east-3:*****:function:default:hcloud-invoke:latest" --project_id="*****" --key="value"
```

表 10-4 参数说明

| 参数             | 是否必须 | 描述                                                    |
|----------------|------|-------------------------------------------------------|
| --cli-region   | 是    | 调用函数所在region。                                         |
| --function_urn | 是    | 函数的URN ( Uniform Resource Name ) ， 唯一标识函数。            |
| --project_id   | 是    | 租户Project ID。                                         |
| 请求体            | 是    | 通过--key="value"方式指定请求体，其对应的json结构：{ "key": "value" }。 |

输出结果如图3所示，输出的响应参数请参见表2。

图 10-8 输出结果

```
{
  "request_id": "3977431d-c254-4e65-979e-30541936651b"
}
```

表 10-5 响应参数说明

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| request_id | String | 请求ID |

使用 KooCLI 配置网络代理

设置网络代理命令如下：

```
export HTTP_PROXY="http://user:password@proxyIp:proxyPort"
```

详情请参考：[https://www.cyberciti.biz/faq/unix-linux-export-variable-http\\_proxy-with-special-characters/](https://www.cyberciti.biz/faq/unix-linux-export-variable-http_proxy-with-special-characters/)。

# 11 申请扩大 FunctionGraph 配额

## 概述

为防止资源滥用，平台限制了各服务资源的配额，对用户的资源数量和容量做了限制。如您最多可以创建多少台弹性云服务器、多少块云硬盘。

如果当前资源配额限制无法满足使用需要，您可以申请扩大配额。

## 查看配额

1. 登录管理控制台。
2. 在页面右上角，选择“资源 > 我的配额”。系统进入“服务配额”页面。

图 11-1 进入服务配额页面



3. 您可以在“服务配额”页面，查看各项资源的总配额及使用情况。如果当前配额不能满足业务要求，请参见后续操作，申请扩大配额。

## 申请扩大配额

1. 登录管理控制台。
2. 在页面右上角，选择“资源 > 我的配额”。系统进入“服务配额”页面。

图 11-2 进入“服务配额”页面



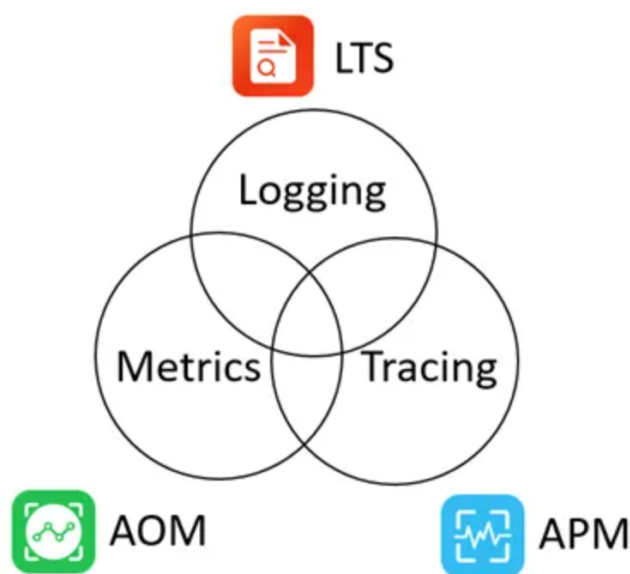
3. 单击“申请扩大配额”。
4. 在“新建工单”页面，根据您的需求，填写相关参数。  
其中，“问题描述”项请填写需要调整的内容和申请原因。
5. 填写完毕后，勾选协议并单击“提交”。

# 12 查看监控指标与配置告警

## 12.1 FunctionGraph 监控概述

华为云函数工作流FunctionGraph与华为云其他云服务紧密集成，为您提供整体解决方案，旨在解决Serverless应用中的可观测性问题。通过与华为云平台上监控、日志、跟踪和分析服务的深度集成，用户可以实现函数功能、性能和成本的有效监控与优化。

图 12-1 函数监控概述



### 联手云日志服务 LTS 打造一站式日志采集存储能力

华为云函数工作流FunctionGraph支持函数一键接入LTS。LTS满足Serverless应用的日志收集、存储、搜索与分析需求，可以高效处理大规模日志，并提供实时日志告警功能。其广泛日志加工与统计功能，能够迅速优化日志数据。借助LTS，Serverless应用更易实现日志管理和监控，提高日志运维效率和应用稳定性。

配置详情请参考[配置和查看函数的调用日志](#)。

## 基于应用运维管理 AOM 构建丰富的指标和监控系统

华为云函数工作流FunctionGraph向AOM上报了超过20项丰富的指标，这些指标包括调用结果、调用次数、调用时长等调用类指标和实例数、异步堆积及内存占用等实例类指标，提供全面而深入的函数运行状态感知，能够反映函数的应用性能与运行状况，有助于进行精确的调优与优化，从而提高其在华为云平台上的运行效率与稳定性。

通过实时和准确的数据支持，基于该监控系统能够使您更好地了解和优化函数的运行情况，提升整体管理和运行效率。

## 通过应用性能管理 APM 实现调用链路追踪

通过整合华为云APM的监控技术和OpenTelemetry的分布式跟踪功能，能够深入监控单个函数内部的调用链，追踪函数执行过程中的各个调用阶段和依赖关系，精确捕获函数调用链上的关键信息，涵盖调用耗时、调用参数以及调用成功情况等，为系统正常运行和故障诊断提供重要数据支持。

同时，当Serverless函数调用出现异常时，华为云APM的分析能力可对特定请求进行数据汇聚，实现以请求为单位的数据聚合与分析，使您能够直观掌握该请求在指定时间内的耗时变化、请求结果等信息，迅速定位异常请求。通过深入分析异常请求，包括异常调用信息与异常触发时间点，可以精准定位问题根源，高效执行故障排查与修复操作。

配置详情请参考[配置和查看函数的调用链信息](#)。

## 运行时扩展和遥测接口

**扩展接口**（Extensions API）允许通过自定义扩展（Extensions）来增强函数的功能。

**遥测接口**（Telemetry API）提供了一种标准化的方式，使扩展程序能够订阅并接收函数的遥测数据（如日志、指标、跟踪数据）。

扩展接口和遥测接口的结合使用，可以实现高度定制化的运维监控和功能增强，同时保持与函数原生服务的无缝集成。

# 12.2 FunctionGraph 支持的监控指标

## 功能说明

本节定义了FunctionGraph上报云监控服务的监控指标的命名空间，监控指标列表和维度定义，用户可以通过云监控服务提供管理控制台或API接口来检索FunctionGraph产生的监控指标和告警信息。

## 命名空间

SYS.FunctionGraph

## 约束与限制

具体展示的监控指标因区域选择而异，请以控制台显示的监控指标为准，此处列举FunctionGraph支持的全部监控指标。

函数监控指标

表 12-1 FunctionGraph 支持的监控指标

| 指标ID                | 指标名称   | 指标含义                                                                                                                                 | 取值范围              | 单位        | 进制          | 测量对象（维度） | 监控周期（原始指标） |
|---------------------|--------|--------------------------------------------------------------------------------------------------------------------------------------|-------------------|-----------|-------------|----------|------------|
| count               | 调用次数   | 该指标用于统计函数调用次数。                                                                                                                       | ≥ 0<br>counts     | Cou<br>nt | 不<br>涉<br>及 | 函数       | 5分钟        |
| failcount           | 错误次数   | 该指标用于统计函数调用错误次数。<br>以下两种情况都会计入错误次数： <ul style="list-style-type: none"><li>函数请求异常，导致无法执行完成且返回200。</li><li>函数自身语法错误或者自身执行错误。</li></ul> | ≥ 0<br>counts     | Cou<br>nt | 不<br>涉<br>及 | 函数       | 5分钟        |
| failRate            | 调用错误率  | 该指标用于统计函数调用错误率。                                                                                                                      | 0%<br>≤X≤<br>100% | %         | 不<br>涉<br>及 | 函数       | 5分钟        |
| rejectcount         | 被拒绝次数  | 该指标用于统计函数调用被拒绝次数。<br>被拒绝次数是指并发请求太多，系统流控而被拒绝的请求次数。                                                                                    | ≥ 0<br>counts     | Cou<br>nt | 不<br>涉<br>及 | 函数       | 5分钟        |
| concurrency         | 并发数    | 该指标用于统计函数同时调用处理的最大并发请求个数。                                                                                                            | ≥ 0<br>counts     | cou<br>nt | 不<br>涉<br>及 | 函数       | 5分钟        |
| reservedinstancesum | 预留实例数  | 该指标用于统计函数运行的预留实例数。                                                                                                                   | ≥ 0<br>counts     | cou<br>nt | 不<br>涉<br>及 | 函数       | 5分钟        |
| duration            | 平均运行时间 | 该指标用于统计函数调用平均运行时间。                                                                                                                   | ≥ 0 ms            | ms        | 不<br>涉<br>及 | 函数       | 5分钟        |
| maxDuration         | 最大运行时间 | 该指标用于统计函数调用最大运行时间。                                                                                                                   | ≥ 0 ms            | ms        | 不<br>涉<br>及 | 函数       | 5分钟        |

| 指标ID               | 指标名称        | 指标含义                      | 取值范围       | 单位    | 进制  | 测量对象（维度） | 监控周期（原始指标） |
|--------------------|-------------|---------------------------|------------|-------|-----|----------|------------|
| minDuration        | 最小运行时间      | 该指标用于统计函数最小运行时间。          | ≥ 0 ms     | ms    | 不涉及 | 函数       | 5分钟        |
| systemErrorCount   | 系统错误次数      | 该指标用于统计函数调用系统错误次数。        | ≥ 0 counts | count | 不涉及 | 函数       | 5分钟        |
| functionErrorCount | 函数错误次数      | 该指标用于统计函数调用函数错误次数。        | ≥ 0 counts | count | 不涉及 | 函数       | 5分钟        |
| payPerUseInstance  | 弹性实例数       | 该指标用于统计函数排除预留实例后实际使用的实例数。 | ≥ 0 counts | count | 不涉及 | 函数       | 5分钟        |
| instances          | 实例数         | 该指标用于统计函数调用实例的个数。         | ≥ 0 counts | count | 不涉及 | 函数       | 5分钟        |
| duration_p500      | 函数执行时间 p50  | 该指标用于统计函数执行时间的 p50。       | ≥ 0 ms     | ms    | 不涉及 | 函数       | 5分钟        |
| duration_p800      | 函数执行时间 p80  | 该指标用于统计函数执行时间的 p80。       | ≥ 0 ms     | ms    | 不涉及 | 函数       | 5分钟        |
| duration_p950      | 函数执行时间 p95  | 该指标用于统计函数执行时间的 p95。       | ≥ 0 ms     | ms    | 不涉及 | 函数       | 5分钟        |
| duration_p990      | 函数执行时间 p990 | 该指标用于统计函数执行时间的 p990。      | ≥ 0 ms     | ms    | 不涉及 | 函数       | 5分钟        |
| duration_p999      | 函数执行时间 p999 | 该指标用于统计函数执行时间的 p999。      | ≥ 0 ms     | ms    | 不涉及 | 函数       | 5分钟        |
| memoryUsed         | 函数执行内存使用    | 该指标用于统计函数执行内存使用量。         | ≥ 0 MB     | MB    | 不涉及 | 函数       | 5分钟        |

| 指标ID          | 指标名称  | 指标含义                                | 取值范围        | 单位 | 进制  | 测量对象（维度） | 监控周期（原始指标） |
|---------------|-------|-------------------------------------|-------------|----|-----|----------|------------|
| functionCost  | 资源使用量 | 该指标用于统计函数执行资源使用量，函数执行内存使用 * 函数执行时间。 | ≥ 0 MB      | MB | 不涉及 | 函数       | 5分钟        |
| instanceUsage | 实例使用率 | 该指标用于统计租户所有函数实例实际使用百分比。             | 0% ≤X≤ 100% | %  | 不涉及 | 函数       | 5分钟        |

表 12-2 函数流支持的监控指标

| 指标ID              | 指标名称      | 指标含义                     | 取值范围       | 单位    | 进制  | 测量对象（维度） | 监控周期（原始指标） |
|-------------------|-----------|--------------------------|------------|-------|-----|----------|------------|
| totalCount        | 调用次数      | 工作流的执行总数。                | ≥ 0 counts | count | 不涉及 | 函数流      | 1分钟        |
| errorCount        | 错误次数      | 工作流的执行失败总数。              | ≥ 0 counts | count | 不涉及 | 函数流      | 1分钟        |
| running           | 运行中个数     | 工作流运行中的流程个数。             | ≥ 0 counts | count | 不涉及 | 函数流      | 1分钟        |
| rejectCount       | 被限流次数     | 工作流被限流的流程个数。             | ≥ 0 counts | count | 不涉及 | 函数流      | 1分钟        |
| averageDuration   | 平均运行时间    | 工作流的平均执行时长。              | ≥ 0 ms     | ms    | 不涉及 | 函数流      | 1分钟        |
| notifyFailedCount | 通知失败次数    | 工作流执行结果通知失败的次数。          | ≥ 0 counts | count | 不涉及 | 函数流      | 1分钟        |
| ExecutionsStarted | 工作流实例启动个数 | 该指标用于统计用户工作流实例开始运行的实例个数。 | ≥ 0 counts | count | 不涉及 | 函数流      | 1分钟        |

| 指标ID                | 指标名称         | 指标含义                       | 取值范围       | 单位    | 进制  | 测量对象（维度） | 监控周期（原始指标） |
|---------------------|--------------|----------------------------|------------|-------|-----|----------|------------|
| ExecutionsSucceeded | 工作流实例执行成功个数  | 该指标用于统计用户工作流实例成功运行完成的实例个数。 | ≥ 0 counts | count | 不涉及 | 函数流      | 1分钟        |
| ExecutionsFailed    | 工作流实例执行失败个数  | 该指标用于统计用户工作流实例运行失败的实例个数。   | ≥ 0 counts | count | 不涉及 | 函数流      | 1分钟        |
| ExecutionsAborted   | 工作流实例执行被中断个数 | 该指标用于统计用户工作流实例被终止运行的个数。    | ≥ 0 counts | count | 不涉及 | 函数流      | 1分钟        |
| ExecutionsTimedOut  | 工作流实例执行超时个数  | 该指标用于统计用户工作流实例超时运行的个数。     | ≥ 0 counts | count | 不涉及 | 函数流      | 1分钟        |

维度

| key                  | value                                     |
|----------------------|-------------------------------------------|
| package-functionname | 应用名-函数名。<br>示例：default-myfunction_Python。 |
| graph_name           | 函数流。                                      |
| projectId            | 租户                                        |

12.3 查看 FunctionGraph 的监控数据

FunctionGraph函数实现了与应用运维管理AOM服务的对接，用户无需任何配置，即可查询函数监控信息。

查看函数监控信息

FunctionGraph会统计函数的运行时指标，显示的指标是函数运行时活动的聚合视图。要查看不同函数版本的指标，可在查看指标前切换函数版本，查询不同版本对应的监控信息。

1. 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
2. 选择待配置的函数，单击进入函数详情页。
3. 选择“监控 > 指标”，选择时间粒度（最近1小时、最近3小时、最近12小时、最近1天、最近3天、自定义），查看函数运行状态。

函数指标说明

函数运行监控指标说明如[表12-3](#)所示。

表 12-3 监控指标说明表

| 指标       | 单位  | 说明                                                                                                                                                          |
|----------|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 调用次数     | 次   | 函数总的调用请求数，包含了错误和被拒绝的调用。异步调用在该请求实际被系统执行时才开始计数。                                                                                                               |
| 运行时间     | 毫秒  | 最大运行时间：某统计周期下，即某一时间段内单次函数执行最大的运行时间。<br>最小运行时间：某统计周期下，即某一时间段内单次函数执行最小的运行时间。<br>平均运行时间：某统计周期下，即某一时间段内单次函数执行平均的运行时间，计算公式：执行时间/调用次数。                            |
| 错误次数/错误率 | 次/% | 系统错误次数：该指标用于统计函数调用系统错误次数，不包含限流错误和执行错误。<br>函数错误次数：指发生异常请求的函数不能正确执行完并且返回200，都计入错误次数。函数自身的语法错误或自身执行错误也会计入该指标。<br>错误率：该指标用于统计函数调用错误率，计算公式：（函数错误次数+系统错误次数）/调用次数。 |
| 被拒绝次数    | 次   | 由于并发请求太多，系统流控而被拒绝的请求次数。                                                                                                                                     |
| 资源统计     | 个   | 并发数：该函数的请求并发数。<br>预留实例：该函数的预留实例数。<br>弹性实例数：该函数的弹性实例数。                                                                                                       |

| 指标    | 单位 | 说明                                                                                                                                                                       |
|-------|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 内存使用量 | MB | 当前仅“华东-上海一”和“华北-北京四”支持查看该指标。<br>最大内存使用量：某统计周期下，即某一时间段内单次函数执行最大的内存使用量。<br>最小内存使用量：某统计周期下，即某一时间段内单次函数执行最小的内存使用量。<br>平均内存使用量：某统计周期下，即某一时间段内单次函数执行平均的内存使用量，计算公式：总内存使用量/调用次数。 |

## 12.4 配置 FunctionGraph 的监控告警

函数及触发器创建以后，可以实时监控函数被调用及运行情况。

### 约束与限制

删除函数后，已创建的告警规则在CES服务控制台里不会实时更新，可能会继续在CES服务控制台里显示最多 7 天。

### 监控函数

不同版本函数的监控信息做了区分，查询函数指标之前设置函数版本，可以查询不同版本对应的监控信息。

### 操作步骤

函数实现与云监控服务的对接，函数上报云监控服务的监控指标，用户可以通过云监控服务来查看函数产生的监控指标和告警信息。

- 步骤1 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。
- 步骤2 单击函数名称，进入函数详情界面。
- 步骤3 选择函数对应的版本或者别名，选择“监控 > 指标”。
- 步骤4 单击“创建告警规则”，弹出“创建告警规则”对话框。
- 步骤5 输入告警参数，单击“下一步”。如[图12-2](#)所示。

图 12-2 创建告警规则

创建告警规则

**i** 告警基础功能免费，触发产生的告警消息由SMN发送，可能产生少量费用，具体费用由SMN结算。[了解计费详情](#)

\* 名称

\* 监控指标

\* 告警策略

当聚合方式为原始值时，无需选择监控周期

\* 告警级别 ☐ 紧急 ☒ 重要 ☐ 次要 ☐ 提示

发送通知 ☐

描述 

0/255

**步骤6** 输入告警规则名称，单击“确定”。

----结束

## 12.5 配置和查看函数的调用日志

FunctionGraph函数实现了与云日志服务的对接，用户无需任何配置，即可查询函数日志信息。

在FunctionGraph函数控制台，可以通过以下两种方式查看函数日志。

- 在测试页签查看日志  
函数创建完成后，可以测试函数，在执行结果页，可以查看函数测试日志。操作步骤请参见[在线调试函数](#)。  
此处最多显示2KB字节日志，如果日志太多，可以去函数详情页日志页签查询日志。
- 在日志页签查看日志  
在函数详情页“监控 > 日志”页签，查询日志信息，操作步骤请参见[云日志服务（LTS）管理函数日志](#)。

### 配置日志组、日志流和查看函数日志

用户可以针对某个函数自行关联日志组和日志流，管理函数日志，即进行函数调用后，调用日志会保存到指定的日志组和日志流下。若未指定，函数调用日志会保持原逻辑自动生成在系统默认创建的日志流下（即创建一个函数默认创建一个对应的日志流），具体请参见[云日志服务（LTS）管理函数日志](#)。

#### 操作步骤

**步骤1** 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。

**步骤2** 单击函数名称进入函数详情页。

**步骤3** 选择“设置 > 日志配置”，参见表12-4进行配置。

表 12-4 日志配置参数说明

| 参数     | 说明                                                                                                                                      |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------|
| 启动日志记录 | V2版本默认为开启状态，V1版本当前不支持此特性。                                                                                                               |
| 日志组    | 为当前函数指定日志组。禁选functiongraph默认创建的日志组functiongraph.log.group.xxx。如何创建日志组详情请参见 <a href="#">管理日志组</a> 。<br>默认展示系统自动生成的日志组（以functiongraph开头）。 |
| 日志流    | 指定日志组下的日志流。如何创建日志流详情请参见 <a href="#">管理日志流</a> 。<br>默认展示为创建函数时自动生成的日志流（以函数名称开头）。                                                         |

**步骤4** 配置完成后单击“保存”。

 说明

函数的日志流可以在更新时随时修改。

**步骤5** 选择“代码”页签，单击“测试”执行函数。

**步骤6** 函数执行完成后，选择“监控 > 日志”，查看函数日志。

- 如图12-3，表示当前函数日志指定生成在该日志组及日志流下。
- 若未指定，此处的日志组显示为系统默认创建的日志组及日志流，日志组名称以“functiongraph”开头。

图 12-3 查看函数日志



- 图中日志列表中可以区分是否请求经过冷启动，下边一条请求显示的冷启动成功，其中耗时13.100ms，表示当前请求是经过冷启动的调用总耗时，日志中上边一条日志显示调用成功，其中耗时1.671ms，表示当前请求不经过冷启，调用耗时。

图 12-4 日志



----结束

日志标签

函数执行时，可以按照自定义标签配置上报标签到云日志服务(LTS)，用户可以通过标签对日志进行过滤筛选。

- 步骤1 新创建函数时，可在“高级设置”中开启“日志记录”，并单击“添加标签”增加日志标签；如已创建函数，请从步骤2开始配置。
- 步骤2 在函数详情页中，选择“设置 > 日志配置”，开启“日志记录”，选择日志组和日志流。
- 步骤3 单击“添加新标签”，输入标签键和标签值，最多支持添加10个标签，配置完成后单击“保存”。

表 12-5 日志标签介绍

| 参数名称 | 说明                             |
|------|--------------------------------|
| 标签键  | 不能为空，只能包含数字、字母、下划线和中划线，最大长度64。 |
| 标签值  |                                |

图 12-5 添加日志标签



- 步骤4 选择“代码”页签，配置测试事件后，单击“测试”，等待函数执行完成。

- 步骤5 选择“监控 > 日志”，在“请求列表”中查看函数请求记录（日志上报会有延迟，请稍等几分钟）。
- 步骤6 登录云日志服务控制台，左侧选择“日志管理”，在日志组列表中选择筛选条件后，搜索函数配置的日志组名称，在搜索结果中再选择日志流的筛选条件，并搜索函数配置的日志流名称，单击日志流名称进入详情页面。

图 12-6 搜索日志流



- 步骤7 在日志流详情页面，右侧单击“日志设置”，进入“索引配置”面板，单击“添加字段”，新增在函数中新添加的日志标签键，如何设置索引请参见[设置LTS日志索引配置](#)。配置完成后，单击“确定”。
- 步骤8 返回日志流详情页面，在搜索框中输入函数配置的日志标签键和值，即可查看匹配的日志内容。

----结束

云日志服务（LTS）管理函数日志

FunctionGraph支持开通云日志服务(LTS)，使用更丰富的函数日志管理功能。开通云日志服务后，FunctionGraph会自动创建1个日志组（functiongraph开头），创建函数后，会默认生成一个日志流（函数名称开头）。

您也可以针对某个函数自行关联日志组和日志流，管理函数日志，即进行函数调用后，调用日志会保存到指定的日志组和日志流下。具体操作请参见[配置日志组、日志流和查看函数日志](#)。

约束与限制：

- 默认创建的20个日志流，您无法自定义。您可以在函数的“日志”页签下，按“F12”，找到query接口里的日志流ID，再到lts里找到对应的日志流ID。



- 若在LTS控制台误删函数日志组，之前的日志数据不可找回，FunctionGraph服务不感知该操作。此时您可以通过修改函数常规设置中的描述信息，保存后触发重建函数日志组。
- 用户普通实例的初始化阶段的日志大小限制为（10MB），超过大小限制的日志进行滚动更新，为您保留最新的日志。

- 步骤1 开通云日志服务（LTS）管理函数日志。

在“日志”页签，单击“去开启”跳转至设置中的“日志配置”界面，单击开启“日志记录”。

图 12-7 开启日志功能



步骤2 设置查询条件。

- 请求列表：支持设置请求ID、调用结果（执行成功、执行失败）、原因分析（初始化失败、加载失败、系统错误、调用超时、内存超限、磁盘超限、代码异常）、耗时、实际使用内存。  
在使用查询字段前，需先在云日志服务LTS中配置相关字段信息，具体配置详情请参见配置函数的日志过滤字段。
- 请求日志：支持关键字、请求ID、实例ID。

表 12-6 调用结果

| 调用结果 | 说明                                                                                            |
|------|-----------------------------------------------------------------------------------------------|
| 执行成功 | 函数执行成功打印的日志。                                                                                  |
| 执行失败 | 函数执行失败打印的日志，包含调用超时、内存超限、磁盘超限、代码异常四种情况。<br>若想查看调用超时的日志信息，请将“日志类型”切换为调用超时，另外3种执行失败下的日志类型查看方法相同。 |

表 12-7 原因分析

| 原因分析  | 说明                        |
|-------|---------------------------|
| 初始化失败 | 函数初始化失败打印的日志。             |
| 加载失败  | runtime加载用户函数文件失败打印的日志    |
| 系统错误  | 内部错误。                     |
| 调用超时  | 函数调用时间超过配置的“执行超时时间”打印的日志。 |
| 内存超限  | 函数内存大小超过配置的“内存”大小打印的日志。   |
| 磁盘超限  | 磁盘超出限制大小打印的日志。            |
| 代码异常  | 代码出现异常情况打印的日志。            |

说明

- 支持的时间条件：最近1小时、最近1天、最近3天及自定义。
- 您可以单击“到LTS进行日志分析等更多操作”，前往LTS控制台管理函数日志。

步骤3 开始日志分析。

“日志分析”页签下内置了LTS的日志分析功能模块，可参考[配置函数的日志过滤字段](#)为函数配置日志过滤字段，关于在LTS上分析日志的具体使用方法请参考[搜索与分析日志](#)。

----结束

配置函数的日志过滤字段

函数工作流的日志查询功能支持多种过滤字段。您在使用前，需按照如下步骤完成配置后方可使用：

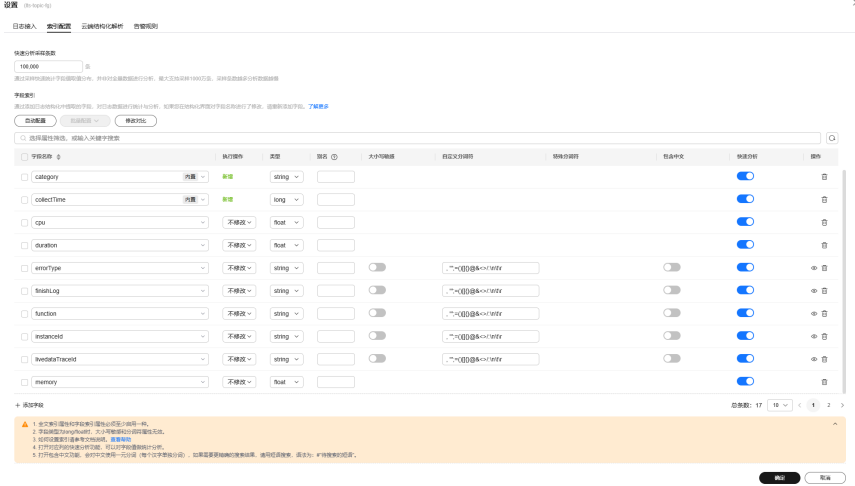
- 步骤1 返回函数工作流控制台，在左侧导航栏选择“函数 > 函数列表”，进入函数列表页面。
- 步骤2 单击函数名称，进入函数详情界面。
- 步骤3 单击“监控”，进入“日志”页签，单击“日志分析”页签，可在函数工作流控制台使用LTS功能进行日志分析。
- 步骤4 单击右侧“日志设置”，进入“索引配置”页，单击“自动配置”后，等待后台自动配置全量字段信息。

图 12-8 日志设置



- 步骤5 自动配置完成后，将“memory”和“duration”两个字段的类型修改为“float”。同时，排查是否存在“errorType”字段，如没有请手动添加，完成后如[图12-9](#)所示。

图 12-9 配置字段信息



**步骤6** 配置完成后，单击“确定”。返回函数的“监控 > 日志 > 请求列表”页签下，即可使用相关字段过滤信息。

----结束

## 日志下载

**约束与限制：**

- 一次最多只能下载5000条日志，请合理选择时间范围，避免下载的日志缺失。
- 当前日志时间戳打印的时间为 UTC 时间。

**步骤1** 选择“监控 > 日志”，单击“请求日志”页签。

**步骤2** 在“请求日志”页签下，选择时间范围，单击“下载日志”。

----结束

## 12.6 配置和查看函数的调用链信息

### 概述

用户在函数的“监控”页签，开启调用链能力。开启后可以在函数“监控 > 调用链”页面或跳转至APM服务“应用监控 > 调用链”页面，查看函数调用链信息。当前仅支持Java8和Java11函数。

### 约束与限制

当前特性仅“中东-利雅得”、“华北-北京四”、“华东-上海一”、“华南-广州”、“亚太-雅加达”、“土耳其-伊斯坦布尔”区域支持。

### 前提条件

- 调用链只支持512MB内存以上的函数，若函数内存低于512MB，无法使用调用链功能，请在“设置 > 常规信息”中，扩大函数“内存”后再使用调用链功能。
- 已开通应用性能管理APM服务的使用权限，具体请参见[APM权限管理](#)。若未开通，则无法获取调用链相关数据。

### 开通调用链

**步骤1** 登录[函数工作流控制台](#)，在左侧的导航栏选择“函数 > 函数列表”。

**步骤2** 单击函数名称，进入函数详情界面。

**步骤3** 选择“监控 > 调用链”。

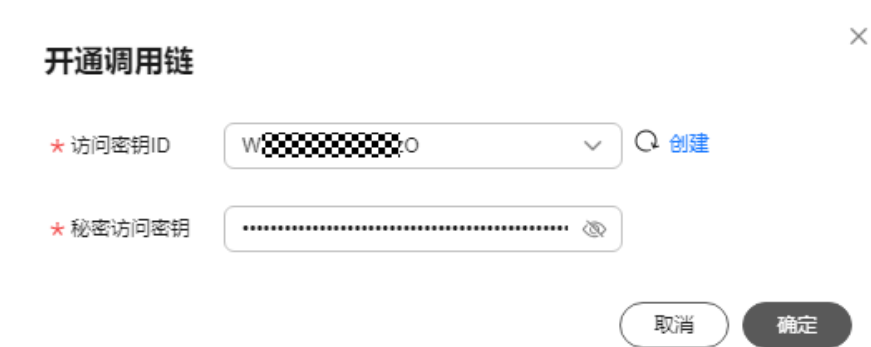
**步骤4** 单击“点击开通”。

图 12-10 开通调用链监控



**步骤5** 此时调用链会从应用性能管理APM服务自动获取一个访问密钥AK/SK，如图12-11。

图 12-11 获取访问密钥



- 已有访问密钥：在“访问密钥ID”的下拉列表中选择“访问密钥ID”，选择完成后单击“确定”，完成开通。
- 无访问密钥：单击“创建”进入APM控制台，参见[新增访问密钥](#)，新增后的AK/SK会同步至函数工作流控制台。

**步骤6** 管理调用链。

- 单击右上方“关闭调用链”，可以直接关闭调用链,关闭调用链后将不能查看函数调用链。
- 若在APM控制台对AK/SK进行了编辑修改，返回函数工作流控制台，单击右上方“更新AKSK”，在弹框中更新AK/SK。

----结束

查询调用链详情

- 步骤1** 返回函数工作流控制台，在左侧导航栏选择“函数 > 函数列表”，单击已开启调用链的函数，进入函数详情页面。
- 步骤2** 在“监控”页签下，选择“调用链”。
- 步骤3** 在左侧设置查询条件，完成后单击“查询Trace”。

图 12-12 设置查询条件

查询条件

清空

时间

2024/07/04 14:07:35 – 2024/...

响应时间

ms

 -- 

ms

执行结果

全部

Trace ID

查询Trace

- 时间：设置查询时间，注意起止查询时间间隔不能超过24小时。
- 响应时间：设置响应时间。
- 执行结果：选择“全部/执行成功/执行失败”。
- Trace ID：调用链的TraceID，填写该搜索条件后，其他搜索条件全部失效，只根据该TraceID搜索。

步骤4 在右侧查看调用链详情。

查看单条调用链详情：在查询到的调用链结果中，单击调用链名称，进入APM控制台查看。

图 12-13 单击调用链名称

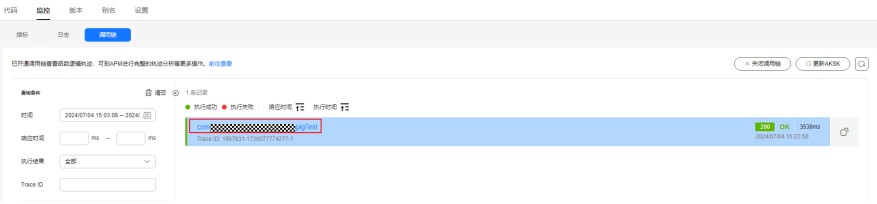
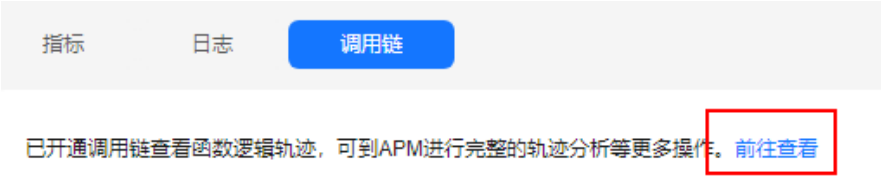


图 12-14 单条详情



查看所有调用链信息：单击“前往查看”，到APM进行完整的路径分析等更多操作。

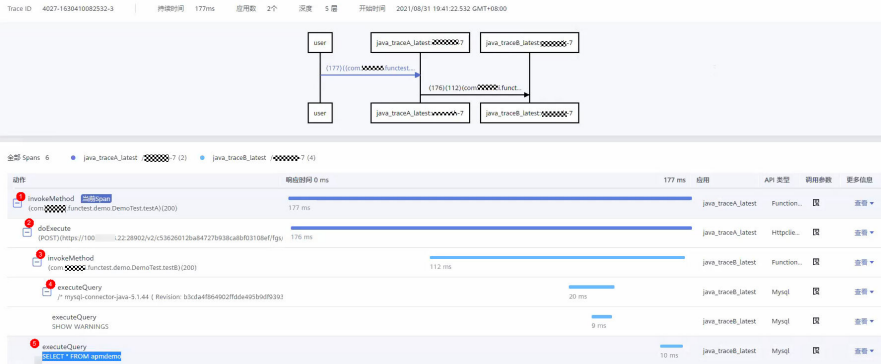
图 12-15 前往 APM



示例

1. 如下是在函数A（DemoTestA）中通过HTTP请求方式调用函数B（DemoTestB）的完整函数执行过程。

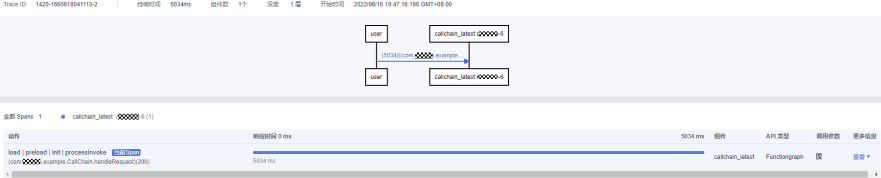
图 12-16 函数 A 调用函数 B 详情



- ①函数A方法的总耗时。
- ②通过HTTP请求调用函数B。
- ③进入函数B。
- ④执行executeQuery。
- ⑤执行Select查询语句。

2. 如下是在函数首次调用包含冷启动的完整函数执行过程。

图 12-17 示例



Spans动作说明如下：

- **load**：是下载解压用户函数代码文件和依赖包的时间。
- **preload**：运行时加载用户函数代码和初始化函数执行环境的时间。
- **init**：初始化函数的执行时间，初始化函数只有在冷启动时才会被执行。
- **processInvoke**：函数的执行时间。

### 说明

您可以参见[调用链](#)，进一步了解调用链详情。

----结束

# 13查看审计日志

## 13.1 CTS 支持的 FunctionGraph 操作列表

云审计支持的FunctionGraph操作列表如表13-1所示。

表 13-1 云审计服务支持的 FunctionGraph 操作列表

| 操作名称     | 资源类型                  | 事件名称                   |
|----------|-----------------------|------------------------|
| 创建函数     | Functions             | createFunction         |
| 删除函数     | Functions             | deleteFunction         |
| 修改函数信息   | Functions             | updateFunctionConfig   |
| 发布函数版本   | FunctionVersions      | publishFunctionVersion |
| 删除函数版本别名 | FunctionVersionsAlias | deleteVersionAlias     |
| 删除函数触发器  | Trigger               | deleteTrigger          |
| 创建函数触发器  | Trigger               | createTrigger          |
| 停用函数触发器  | Trigger               | disableTrigger         |
| 启用函数触发器  | Trigger               | enableTrigger          |

## 13.2 在 CTS 事件列表查看云审计事件

### 场景描述

云审计服务能够为您提供云服务资源的操作记录，记录的信息包括发起操作的用户身份、IP地址、具体的操作内容的信息，以及操作返回的响应信息。根据这些操作记录，您可以很方便地实现安全审计、问题跟踪、资源定位，帮助您更好地规划和利用已有资源、甄别违规或高危操作。

## 什么是事件

事件即云审计服务追踪并保存的云服务资源的操作日志，操作包括用户对云服务资源新增、修改、删除等操作。您可以通过“事件”了解到谁在什么时间对系统哪些资源做了什么操作。

## 什么是管理类追踪器和数据类追踪器

管理追踪器会自动识别并关联当前用户所使用的所有云服务，并将当前用户的所有操作记录在该追踪器中。管理追踪器记录的是管理类事件，即用户对云服务资源新建、修改、删除等操作事件。

数据追踪器会记录用户对OBS桶中的数据操作的详细信息。数据类追踪器记录的是数据类事件，即OBS服务上报的用户对OBS桶中数据的操作事件，例如上传数据、下载数据等。

## 约束与限制

- 管理类追踪器未开启组织功能之前，单账号跟踪的事件可以通过云审计控制台查询。管理类追踪器开启组织功能之后，多账号的事件只能在账号自己的事件列表页面去查看，或者到组织追踪器配置的OBS桶中查看，也可以到组织追踪器配置的CTS/system日志流下面去查看。组织追踪器的详细介绍请参见[组织追踪器概述](#)。
- 用户通过云审计控制台只能查询最近7天的操作记录，过期自动删除，不支持人工删除。如果需要查询超过7天的操作记录，您必须配置转储到对象存储服务（OBS）或云日志服务（LTS），才可在OBS桶或LTS日志组里面查看历史事件信息。否则，您将无法追溯7天以前的操作记录。
- 用户对云服务资源做出创建、修改、删除等操作后，1分钟内可以通过云审计控制台查询管理类事件操作记录，5分钟后才可通过云审计控制台查询数据类事件操作记录。
- CTS新版事件列表不显示数据类审计事件，您需要在旧版事件列表查看数据类审计事件。

## 前提条件

### 1. 注册华为云并实名认证。

如果您已有一个华为账户，请跳到下一个任务。如果您还没有华为账户，请参考以下步骤创建。

- a. 打开[华为云官网](#)，单击“注册”。
- b. 根据提示信息完成注册，详细操作请参见[如何注册华为云管理控制台的用户？](#)。

注册成功后，系统会自动跳转至您的个人信息界面。

- c. 参考[实名认证](#)完成个人或企业账号实名认证。

### 2. 为用户添加操作权限。

如果您是以主账号登录华为云，请跳到下一个任务。

如果您是以IAM用户登录华为云，需要联系CTS管理员（主账号或admin用户组中的用户）对IAM用户授予CTS FullAccess权限。授权方法请参见[给IAM用户授权](#)。

查看审计事件

用户进入云审计服务创建管理类追踪器后，系统开始记录云服务资源的操作。在创建数据类追踪器后，系统开始记录用户对OBS桶中数据的操作。云审计服务管理控制台会保存最近7天的操作记录。

本节介绍如何在云审计服务管理控制台查看或导出最近7天的操作记录。

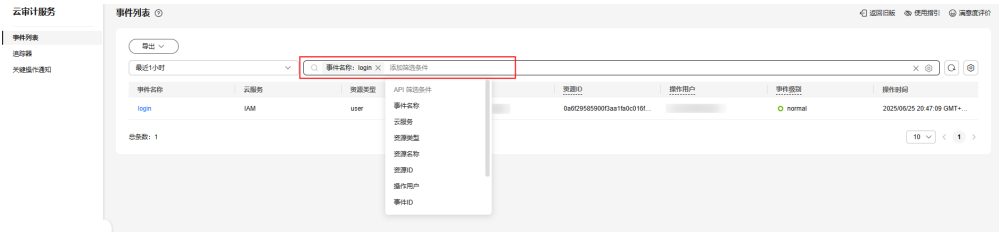
在 CTS 新版事件列表查看审计事件

- 步骤1 登录[CTS控制台](#)。
- 步骤2 登录控制台，单击左上角，选择“管理与部署 > 云审计服务 CTS”，进入云审计服务页面。
- 步骤3 单击左侧导航栏的“事件列表”，进入事件列表信息页面。
- 步骤4 在列表上方，可以通过筛选时间范围，查询最近1小时、最近1天、最近1周的操作事件，也可以自定义最近7天内任意时间段的操作事件。
- 步骤5 事件列表支持通过高级搜索来查询对应的操作事件，您可以在筛选器组合一个或多个筛选条件。

表 13-2 事件筛选参数说明

| 参数名称 | 说明                                                                                                                                             |
|------|------------------------------------------------------------------------------------------------------------------------------------------------|
| 事件名称 | 操作事件的名称。<br>输入的值区分大小写，需全字符匹配，不支持模糊匹配模式。<br>各个云服务支持审计的操作事件的名称请参见 <a href="#">支持审计的服务及详细操作列表</a> 《云审计服务用户指南》的“支持审计的服务及操作列表”章节。<br>示例：updateAlarm |
| 云服务  | 云服务的名称缩写。<br>输入的值区分大小写，需全字符匹配，不支持模糊匹配模式。<br>示例：IAM                                                                                             |
| 资源名称 | 操作事件涉及的云资源名称。<br>输入的值区分大小写，需全字符匹配，不支持模糊匹配模式。<br>当该事件所涉及的云资源无资源名称或对应的API接口操作不涉及资源名称参数时，该字段为空。<br>示例：ecs-name                                    |
| 资源ID | 操作事件涉及的云资源ID。<br>输入的值区分大小写，需全字符匹配，不支持模糊匹配模式。<br>当该资源类型无资源ID或资源创建失败时，该字段为空。<br>示例： <i>{虚拟机ID}</i>                                               |

| 参数名称   | 说明                                                                                                                                                                         |
|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 事件ID   | 操作事件日志上报到CTS后，查看事件中的trace_id参数值。<br>输入的值需全字符匹配，不支持模糊匹配模式。<br>示例：01d18a1b-56ee-11f0-ac81-*****1e229                                                                         |
| 资源类型   | 操作事件涉及的资源类型。<br>输入的值区分大小写，需全字符匹配，不支持模糊匹配模式。<br>各个云服务的资源类型请参见 <a href="#">支持审计的服务及详细操作列表</a> 《云审计服务用户指南》的“支持审计的服务及操作列表”章节。<br>示例：user                                       |
| 操作用户   | 触发事件的操作用户。<br>下拉选项中选择一个或多个操作用户。<br>查看事件中的trace_type的值为“SystemAction”时，表示本次操作由服务内部触发，该条事件对应的操作用户可能为空。<br>IAM身份与操作用户对应关系，以及操作用户名称的格式说明，请参见 <a href="#">IAM身份与操作用户对应关系</a> 。  |
| 事件级别   | 下拉选项包含“normal”、“warning”、“incident”，只可选择其中一项。 <ul style="list-style-type: none"><li>normal代表操作成功。</li><li>warning代表操作失败。</li><li>incident代表比操作失败更严重的情况，如引起其他故障等。</li></ul> |
| 企业项目ID | 资源所在的企业项目ID。<br>查看企业项目ID的方式：在EPS服务控制台的“项目管理”页面，可以查看企业项目ID。<br>示例：b305ea24-c930-4922-b4b9-*****1eb2                                                                         |
| 访问密钥ID | 访问密钥ID，包含临时访问凭证和永久访问密钥。<br>查看访问密钥ID的方式：在控制台右上方，用户名下拉选项中，选择“我的凭证 > 访问密钥”，可以查看访问密钥ID。<br>示例：HSTAB47V9V*****TLN9                                                              |



- 步骤6** 在事件列表页面，您还可以导出操作记录文件、刷新列表、设置列表展示信息等。
- 在搜索框中输入任意关键字，按下Enter键，可以在事件列表搜索符合条件的数据。
  - 单击“导出”按钮，云审计服务会将查询结果以.xlsx格式的表格文件导出，该.xlsx文件包含了本次查询结果的所有事件，且最多导出5000条信息。

- 单击按钮，可以获取到事件操作记录的最新信息。
- 单击按钮，可以自定义事件列表的展示信息。启用表格内容折行开关，可让表格内容自动折行，禁用此功能将会截断文本，默认停用此开关。

**步骤7** （可选）在新版事件列表页面，单击右上方的“返回旧版”按钮，可切换至旧版事件列表页面。

----结束

在 CTS 旧版事件列表查看审计事件

- 步骤1** 登录[CTS控制台](#)。
- 步骤2** 登录控制台，单击左上角，选择“管理与部署 > 云审计服务 CTS”，进入云审计服务页面。
- 步骤3** 单击左侧导航栏的“事件列表”，进入事件列表信息页面。
- 步骤4** 用户每次登录云审计控制台时，控制台默认显示新版事件列表，单击页面右上方的“返回旧版”按钮，切换至旧版事件列表页面。
- 步骤5** 在页面右上方，可以通过筛选时间范围，查询最近1小时、最近1天、最近1周的操作事件，也可以自定义最近7天内任意时间段的操作事件。
- 步骤6** 事件列表支持通过筛选来查询对应的操作事件，如[图13-1](#)所示。

图 13-1 筛选框

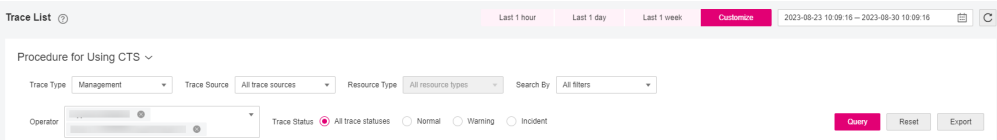


表 13-3 事件筛选参数说明

| 参数名称 | 说明                                                                                                                                                        |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| 事件类型 | 事件类型分为“管理事件”和“数据事件”。 <ul style="list-style-type: none"><li>• 管理类事件，即用户对云服务资源新建、修改、删除等操作事件。</li><li>• 数据类事件，即OBS服务上报的OBS桶中的数据的操作事件，例如上传数据、下载数据等。</li></ul> |
| 云服务  | 在下拉选项中，选择触发操作事件的云服务名称。                                                                                                                                    |
| 资源类型 | 在下来选项中，选择操作事件涉及的资源类型。<br>各个云服务的资源类型请参见 <a href="#">支持审计的服务及详细操作列表</a> 《云审计服务用户指南》的“支持审计的服务及操作列表”章节。                                                       |

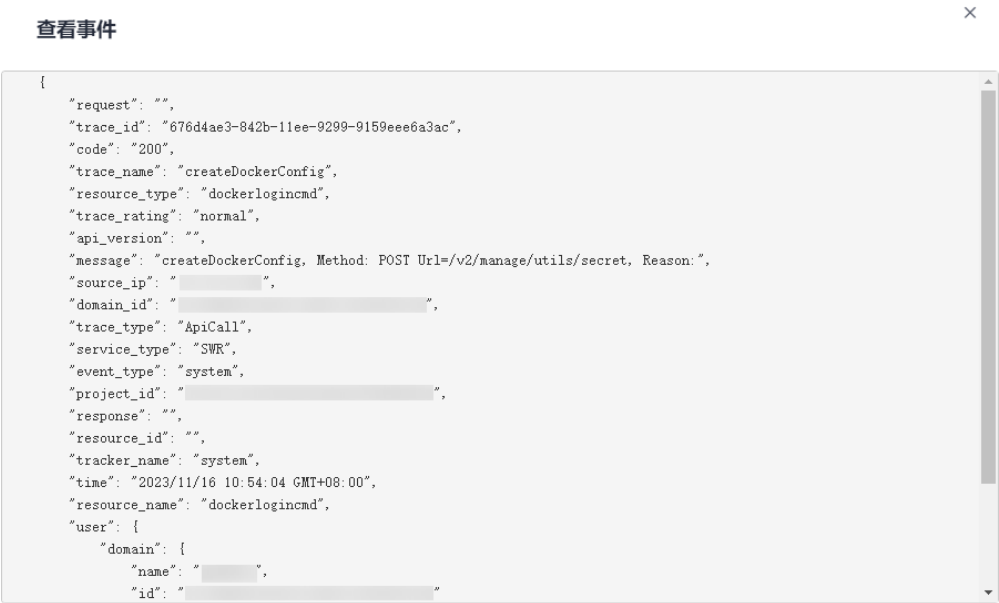
| 参数名称 | 说明                                                                                                                                                                                                                                                                                                                   |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 筛选类型 | 筛选类型分为“资源ID”、“事件名称”和“资源名称”。 <ul style="list-style-type: none"><li>资源ID：操作事件涉及的云资源ID。<br/>当该资源类型无资源ID，或资源创建失败时，该字段为空。</li><li>事件名称：操作事件的名称。<br/>各个云服务支持审计的操作事件的名称请参见<a href="#">支持审计的服务及详细操作列表</a>《云审计服务用户指南》的“支持审计的服务及操作列表”章节。</li><li>资源名称：操作事件涉及的云资源名称。<br/>当事件所涉及的云资源无资源名称，或对应的API接口操作不涉及资源名称参数时，该字段为空。</li></ul> |
| 操作用户 | 触发事件的操作用户。<br>下拉选项中选择一个或多个操作用户。<br>查看事件中的trace_type的值为“SystemAction”时，表示本次操作由服务内部触发，该条事件对应的操作用户可能为空。<br>IAM身份与操作用户对应关系，以及操作用户名称的格式说明，请参见 <a href="#">IAM身份与操作用户对应关系</a> 。                                                                                                                                            |
| 事件级别 | 可选项包含“所有事件级别”、“Normal”、“Warning”、“Incident”，只可选择其中一项。 <ul style="list-style-type: none"><li>Normal代表操作成功。</li><li>Warning代表操作失败。</li><li>Incident代表比操作失败更严重的情况，如引起其他故障等。</li></ul>                                                                                                                                   |

- 步骤7 选择完查询条件后，单击“查询”。
- 步骤8 在事件列表页面，您还可以导出操作记录文件和刷新列表。
  - 单击“导出”按钮，云审计服务会将查询结果以CSV格式的表格文件导出，该CSV文件包含了本次查询结果的所有事件，且最多导出5000条信息。
  - 单击按钮，可以获取到事件操作记录的最新信息。
- 步骤9 在事件的“是否篡改”列中，您可以查看该事件是否被篡改：
  - 上报的审计日志没有被篡改，显示“否”；
  - 上报的审计日志被篡改，显示“是”。
- 步骤10 在需要查看的事件左侧，单击展开该记录的详细信息。

| 事件名称                                                                                                                                                                                                                                                                                                                                                                                                                                                   | 资源类型           | 云服务 | 资源ID | 资源名称           | 事件级别                                                                                       | 操作用户 | 操作时间                          | 操作                   |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|-----|------|----------------|--------------------------------------------------------------------------------------------|------|-------------------------------|----------------------|
|  createDockerConfig                                                                                                                                                                                                                                                                                                                                                 | dockerlogincmd | SWR | --   | dockerlogincmd |  normal |      | 2023/11/16 10:54:04 GMT+08:00 | <a href="#">查看事件</a> |
| <div><div>request</div><div>trace_id</div><div>code</div><div>trace_name</div><div>resource_type</div><div>trace_iding</div><div>api_version</div><div>message</div><div>source_ip</div><div>domain_id</div><div>trace_type</div></div> <div><div>299</div><div>createDockerConfig</div><div>dockerlogincmd</div><div>normal</div><div>createDockerConfig: Method: POST Uri: v2/manage/ulists/secret: Reason:</div><div></div><div>ApiCall</div></div> |                |     |      |                |                                                                                            |      |                               |                      |



**步骤11** 在需要查看的记录右侧，单击“查看事件”，会弹出一个窗口显示该操作事件结构的详细信息。



**步骤12** （可选）在旧版事件列表页面，单击右上方的“体验新版”按钮，可切换至新版事件列表页面。

----结束

相关文档

- 关于事件结构的关键字段详解，请参见[事件结构事件结构](#)“《云审计服务用户指南》>事件参考>事件结构”章节和[事件样例事件样例](#)“《云审计服务用户指南》>事件参考>事件样例”章节。
- 您可以通过以下示例，来学习如何查询具体的事件：
  - 使用云审计服务，审计最近两周内云硬盘服务的创建和删除操作。具体操作，请参见[安全审计](#)。
  - 使用云审计服务，定位现网某个弹性云服务器在某日上午发生的故障，以及定位现网创建弹性云服务器操作失败的问题。具体操作，请参见[问题定位](#)。
  - 使用云审计服务，查看某个弹性云服务器的所有的操作记录。具体操作，请参见[资源跟踪](#)。