函数工作流 FunctionGraph

快速入门

文档版本 01

发布日期 2025-09-11

华为云计算技术有限公司

商标声明


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 入门实践

操作指引

您可以使用下述方式创建函数并部署代码。

- 函数工作流控制台：基于界面化的操作快速创建出简易可执行的函数，适用于构建轻量级应用及快速验证等场景。具体操作步骤请参见**使用空白模板创建并执行函数**。
- VSCode：基于华为云函数工作流VSCode插件，您可以在本地代码编辑器便捷地创建、执行函数，适用于小型团队、个人开发者等。具体操作步骤请参见**VSCode本地调试**。
- Serverless Devs：基于开源CLI工具使用Yaml配置文件便捷管理多个函数，适用于拥有较多函数的复杂项目或自动化部署等场景。具体操作步骤请参见**Serverless Devs使用**。
- Serverless Framework：基于V3版本的Serverless Framework使用Yaml配置文件维护函数的全生命周期，适用于有该工具使用基础的用户。具体操作步骤请参见**Serverless Framework快速入口**。

FunctionGraph 最佳实践

当您了解如何创建函数等基本操作后，可以根据自身的业务需求使用函数工作流 FunctionGraph 提供的一系列常用实践。

本文介绍函数工作流 FunctionGraph 常用实践，帮助您更好的使用函数工作流，更多最佳实践请参考**FunctionGraph最佳实践汇总**。

表 1-1 常用最佳实践

实践	描述
使用FunctionGraph函数对OBS中的图片进行压缩	本实践基于函数工作流服务实践所编写，用于指导您使用函数工作流服务实现图片压缩的功能。
使用FunctionGraph函数为OBS中的图片打水印	本实践基于函数工作流服务实践所编写，用于指导您使用函数工作流服务实现为图片打水印的功能。

实践	描述
使用已有SpringBoot项目构建HTTP函数	本章节指导使用Springboot开发应用的用户，将业务通过构建HTTP函数的方式部署到FunctionGraph。
函数+LTS：日志实时分析实战	• 通过LTS云日志服务，快速完成ECS等服务器的任务运行日志采集、加工和转换。 • 通过函数工作流服务中的函数创建LTS触发器获取日志数据，经由自定义函数对日志中的关键信息进行分析和处理，过滤出告警日志。 • SMN消息通知服务通过短信和邮件推送告警信息，通知业务人员进行处理。 • 将函数处理后的日志数据投递至OBS桶中集中存储，便于后续处理。
函数+CTS：登录/登出安全分析实战	• 通过CTS云审计服务，完成对公有云账户对各个云服务资源操作动作和结果的实时记录。 • 通过在函数工作流服务中创建CTS触发器获取订阅的资源操作信息，经由自定义函数对资源操作的信息进行分析和处理，产生告警日志。 • SMN消息通知服务通过短信和邮件推送告警信息，通知业务人员进行处理。
使用函数处理IOT数据	该案例演示您如何使用FunctionGraph与IotDA 服务组合，处理物联网设备上报以及设备状态变动的相关数据。物联网设备在IotDA 平台进行管理，设备产生的数据可以从IotDA直接流转触发FunctionGraph 的函数运行。用户可以根据需要编写函数处理这些数据。 通常该组合，可以适用于以下场景，如将设备上报的数据在处理后进行存储到如OBS；对上报的数据进行结构化，清洗然后存储到数据库；根据设备状态变化进行事件通知等。
工作流+函数：自动化处理OBS中数据	本实践基于函数流服务实践所编写，用于指导您使用函数流服务实现OBS数据处理的功能。
使用FunctionGraph部署AI绘画Stable Diffusion应用	本章节介绍如何通过FunctionGraph的应用中心，部署AI绘画Stable-Diffusion应用，并提供多种自定义使用AI绘画应用的方法。

2 使用空白模板创建并执行函数

本章节介绍如何在函数工作流控制台使用空白模板快速开发一个简单的Hello World函数。该章节以创建HelloWorld事件函数为例，介绍函数的创建及测试过程，供您快速体验FunctionGraph函数的基本功能。

准备工作

1. 注册华为账号并实名认证。

   在创建函数前，请先注册华为账号并实名认证，具体步骤请参考**注册华为账号并开通华为云**和**实名认证介绍**。

   如果您已有一个华为账号并实名认证，请跳过此步骤。

2. 免费额度。

   函数工作流服务每个月都会提供一定数量的免费额度，免费额度是子主账户共同使用，具体详情请参见**免费额度**。

   当免费额度使用完后，若您继续使用函数工作流时，账户的可用额度小于待结算的账单时，即被判定为账户欠费。欠费后，可能会影响您的服务资源的正常运行，请及时充值，具体详情请参考**账户充值**。

3. 为用户添加函数的操作权限。

   本章节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**。

步骤一：创建函数

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. 单击右上方的“创建函数”，进入“创建函数”页面，开始创建空白函数。
3. 参考**图2-1**配置函数基本信息，具体参数说明如下，完成后单击“创建函数”。

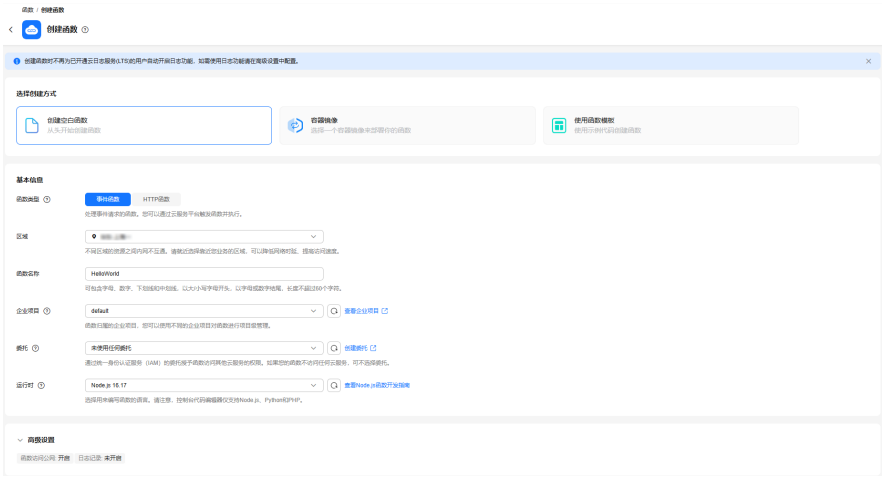
表 2-1 创建函数参数说明

参数	说明	取值样例
函数类型	选择事件函数。 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

参数	说明	取值样例
区域	选择函数所在的区域。 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： • 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。	HelloWorld
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。 默认为“default”，支持用户选择已创建的企业项目。 **如果您没有开通企业管理服务，将无法看到企业项目选项。开通方法请参见如何开通企业项目。**	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。 默认“未使用任何委托”，支持选择已创建的委托。 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**默认委托**。	未使用任何委托
委托权限策略	**此参数仅在选择使用委托时显示。** 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**修改函数委托**通过IAM控制台进行操作。	-
运行时	选择编写函数的运行时语言。 FunctionGraph支持的运行时语言请参考**FunctionGraph支持的运行时语言**。 • 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。 • 函数成功创建后，不支持修改运行时语言。	Node.js 16.17

参数	说明	取值样例
高级设置	• 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。 • 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。 **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。** • 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**云日志服务价格详情**。 • 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。 数据加密服务DEW将按需收取费用，详情请参见**数据加密服务计费说明**。	• 函数访问公网：开启 • 函数访问VPC内资源：未开启 • 日志记录：未开启 • KMS静态加密代码：不使用

图 2-1 创建函数



4. 配置代码源，复制如下代码至代码窗，单击“部署代码”。

   样例代码实现的功能是：获取测试事件，打印测试事件信息。

```
exports.handler = function (event, context, callback) {
    const error = null;
    const output = `Hello message: ${JSON.stringify(event)}`;
    callback(error, output);
}
```

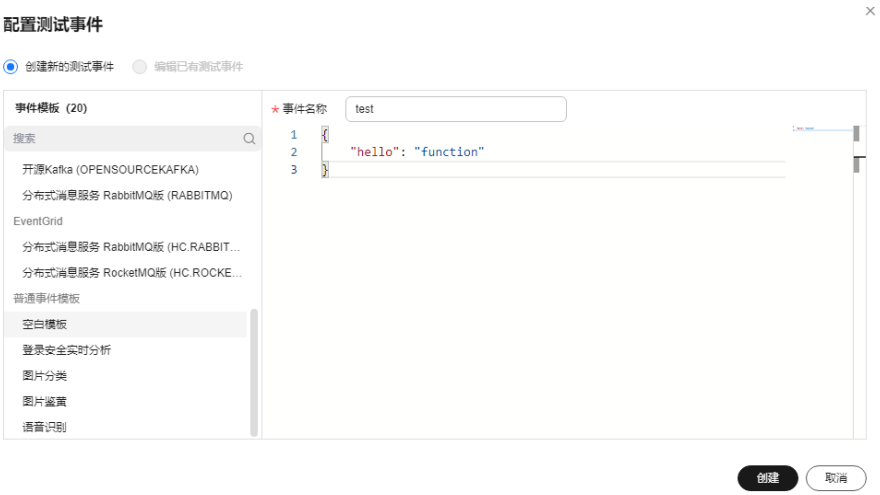
步骤二：测试函数

1. 在函数详情页，单击“测试”，在弹窗中创建新的测试事件。

2. 选择“空白模板”，事件名称输入“test”，测试事件修改为如下所示，完成后单击“创建”。

```
{
  "hello": "function"
}
```

图 2-2 配置测试事件



步骤三：查看执行结果

单击test事件的“测试”，执行后，在右侧查看执行结果。

- “函数返回”显示函数的返回结果。
- “日志”部分显示函数执行过程中生成的日志。
- “执行摘要”部分显示“日志”中的关键信息。

图 2-3 查看执行结果

说明

此页面最多显示2K日志，了解函数更多日志信息，请参考**配置和查看函数的调用日志**。

相关信息

- 了解函数工作流的相关概念，请参考**基本概念**。
- 了解函数工作流的价格，请参考**函数工作流价格详情**。
- 了解函数工作流的约束与限制，请参考**约束与限制**。

3 使用函数模板创建并执行函数

FunctionGraph平台提供了多种函数模板，本章节将以“context使用指导”函数模板介绍如何在创建函数时选择模板，实现模板代码、运行环境自动填充，快速构建应用程序。

准备工作

1. 注册华为账号并实名认证。

   在创建函数前，请先注册华为账号并实名认证，具体步骤请参考**注册华为账号并开通华为云**和**实名认证介绍**。

   如果您已有一个华为账号并实名认证，请跳过此步骤。

2. 免费额度。

   函数工作流服务每个月都会提供一定数量的免费额度，免费额度是子主账户共同使用，具体详情请参见**免费额度**。

   当免费额度使用完后，若您继续使用函数工作流时，账户的可用额度小于待结算的账单时，即被判定为账户欠费。欠费后，可能会影响您的服务资源的正常运行，请及时充值，具体详情请参考**账户充值**。

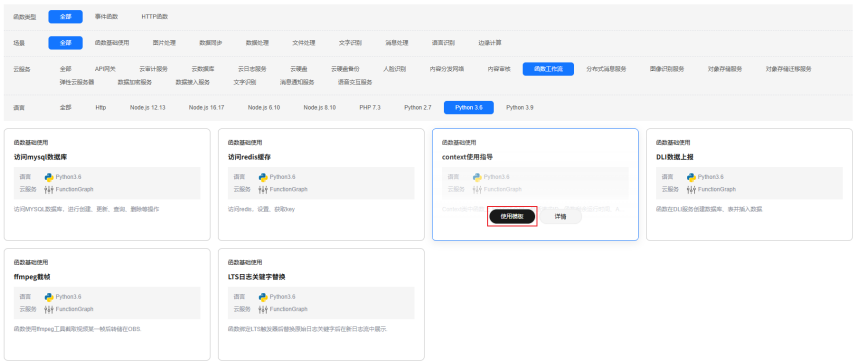
3. 为用户添加函数的操作权限。

   本章节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**。

步骤一：创建函数

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. 单击右上方的“创建函数”，进入“创建函数”页面，使用模板创建函数。
3. 参考**图3-1**，选择如下模板并单击“使用模板”。

图 3-1 选择模板



4. 进入创建函数界面，填写函数参数，单击“创建函数”。
   - 函数模板：用户已选择的模板名称，如需更改模板，请单击右侧“重新选择”。
   - 区域：默认，支持用户选择其他区域。

     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：输入“context”。
   - 企业项目：默认“default”，支持用户选择已创建的企业项目。

     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。
   - 委托名称：默认未使用任何委托，支持用户选择已创建的委托。

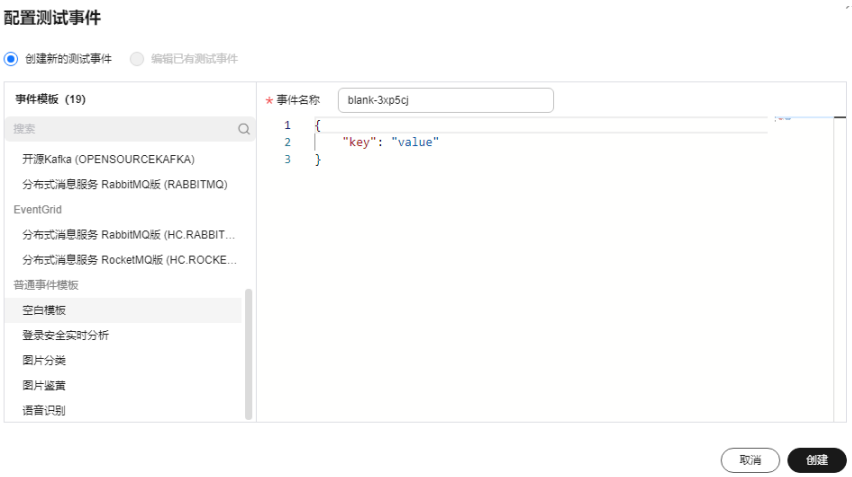
     用户委托函数工作流去访问其他的云服务，举例：如果用户函数需要访问LTS、VPC等服务，则需要提供权限委托名称，如果用户函数不访问任何云服务，则不用提供委托名称。
   - 运行时：选择用来编写函数的语言，默认“Python 3.6”，此处不支持用户选择其他运行时语言。

图 3-2 填写基本信息

步骤二：测试函数

1. 在函数详情页，单击“测试”，在弹窗中创建新的测试事件。
2. 选择“空白模板”，事件名称输入“test”，完成后单击“创建”。

图 3-3 配置测试事件



步骤三：查看执行结果

单击test事件的“测试”，成功执行后，在右侧查看执行结果。

- “函数返回”显示函数的返回结果。
- “日志”部分显示函数执行过程中生成的日志。
- “执行摘要”部分显示“日志”中的关键信息。

说明

此页面最多显示2K日志，了解函数更多日志信息，请参考**配置和查看函数的调用日志**。

相关信息

- 了解函数工作流的相关概念，请参考**基本概念**。
- 了解函数工作流的价格，请参考**函数工作流价格详情**。
- 了解函数工作流的约束与限制，请参考**约束与限制**。

4 使用容器镜像创建并执行 HTTP 函数

本章节将以使用容器镜像方式创建HTTP类型函数为例，介绍容器镜像函数的创建及测试过程。

此过程中用户需要在镜像中实现一个http server，并监听**8000（下文示例中提及的8000端口请不要变动）端口**接收请求。备注：可以使用任意基础镜像。

约束与限制

- HTTP函数只支持使用APIC/APIG触发器。
- 在云上环境会默认使用uid 1003，gid 1003 启动容器。uid、gid可以在函数页面的“设置 > 常规设置 >容器镜像覆盖”板块中修改，但不可以是root或其他保留id。
- **HTTP函数示例中涉及的8000端口请勿修改。**
- **如果使用Alpine版的基础镜像，请使用“addgroup”和“adduser”命令。**

准备工作

1. 注册华为账号并实名认证。

   在创建函数前，请先注册华为账号并实名认证，具体步骤请参考**注册华为账号并开通华为云**和**实名认证介绍**。

   如果您已有一个华为账号并实名认证，请跳过此步骤。

2. 免费额度。

   函数工作流服务每个月都会提供一定数量的免费额度，免费额度是子主账户共同使用，具体详情请参见**免费额度**。

   当免费额度使用完后，若您继续使用函数工作流时，账户的可用额度小于待结算的账单时，即被判定为账户欠费。欠费后，可能会影响您的服务资源的正常运行，请及时充值，具体详情请参考**账户充值**。

3. 为用户添加函数的操作权限。

   本章节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**。

步骤一：制作镜像

以在linux x86 64位系统上制作镜像为例。（系统配置无要求）

1. 创建一个空文件夹
   ```
   mkdir custom_container_http_example && cd custom_container_http_example
   ```
2. 以Nodejs语言为例，实现一个Http Server，其他语言请参考开发指南中的。

   创建一个main.js文件，引入express框架，接收POST请求，打印请求Body到标准输出并返回Hello FunctionGraph, method POST给客户端。
   ```
   const express = require('express');

   const PORT = 8000;

   const app = express();
   app.use(express.json());

   app.post('/*', (req, res) => {
       console.log('receive', req.body);
       res.send('Hello FunctionGraph, method POST');
   });

   app.listen(PORT, () => {
     console.log(`Listening on http://localhost:${PORT}`);
   });
   ```
3. 创建一个package.json文件，此文件用于向npm提供信息，使其能够识别项目以及处理项目的依赖关系。
   ```
   {
     "name": "custom-container-http-example",
     "version": "1.0.0",
     "description": "An example of a custom container http function",
     "main": "main.js",
     "scripts": {},
     "keywords": [],
     "author": "",
     "license": "ISC",
     "dependencies": {
         "express": "^4.17.1"
     }
   }
   ```
   - name：值为项目名。
   - version：值为项目版本。
   - main：列举文件为程序的入口文件。
   - dependencies：列出npm上可用的项目的所有依赖项。
4. 创建Dockerfile文件
   ```
   FROM node:12.10.0

   ENV HOME=/home/custom_container
   ENV GROUP_ID=1003
   ENV GROUP_NAME=custom_container
   ENV USER_ID=1003
   ENV USER_NAME=custom_container

   RUN mkdir -m 550 ${HOME} && groupadd -g ${GROUP_ID} ${GROUP_NAME} && useradd -u $
   {USER_ID} -g ${GROUP_ID} ${USER_NAME}

   COPY --chown=${USER_ID}:${GROUP_ID} main.js ${HOME}
   COPY --chown=${USER_ID}:${GROUP_ID} package.json ${HOME}

   RUN cd ${HOME} && npm install

   RUN chown -R ${USER_ID}:${GROUP_ID} ${HOME}

   RUN find ${HOME} -type d | xargs chmod 500
   RUN find ${HOME} -type f | xargs chmod 500

   USER ${USER_NAME}
   ```

```
WORKDIR ${HOME}

EXPOSE 8000
ENTRYPOINT ["node", "/home/custom_container/main.js"]
```

- FROM：指定基础镜像为node:12.10.0，基础镜像必须设置，值可修改。
- ENV：设置环境变量，设置HOME环境变量为/home/custom_container，设置GROUP_NAME和USER_NAME为custom_container，**USER_ID**和**GROUP_ID为1003**，这些环境变量必须设置，值可修改。
- RUN：格式为RUN <命令>，例如RUN mkdir -m 550 ${HOME}表示构建容器时创建${USER_NAME}用户的home目录。
- USER：切换${USER_NAME}用户。
- WORKDIR：切换工作目录到${USER_NAME}用户的“${HOME}”目录下。
- COPY：将main.js和package.json拷贝到容器的${USER_NAME}用户的home目录下。
- **EXPOSE：暴露容器的8000端口，请勿修改。**
- ENTRYPOINT：使用node main.js命令启动容器，请勿修改。

5. 构建镜像

   指定镜像的名称为custom_container_http_example，版本为latest，“.”指定Dockerfile所在目录，镜像构建命令将该路径下所有的内容打包给容器引擎帮助构建镜像。

   ```
   docker build -t custom_container_http_example:latest .
   ```

步骤二：本地验证

1. 启动docker容器

   ```
   docker run -u 1003:1003 -p 8000:8000 custom_container_http_example:latest
   ```

2. 打开一个新的命令行窗口，向开放的8000端口发送消息，支持访问模板代码中根目录“/”下所有路径，以下以“helloworld”为例。

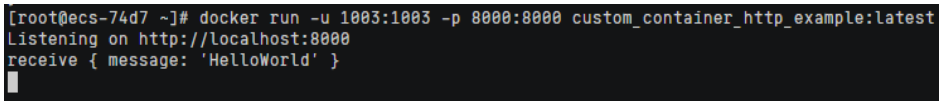
   ```
   curl -XPOST -H 'Content-Type: application/json' -d '{"message":"HelloWorld"}' localhost:8000/helloworld
   ```

   按照模块代码中返回

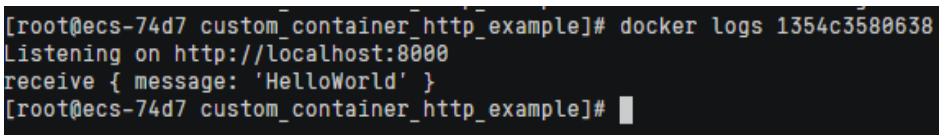
   ```
   Hello FunctionGraph, method POST
   ```

3. 在容器启动端口可以看到

   ```
   receive {"message":"HelloWorld"}
   ```

   或者使用docker logs命令获取容器的日志

步骤三：上传镜像

1. 登录容器镜像服务控制台，在左侧导航栏选择“我的镜像”。
2. 单击右上角的“客户端上传”或“页面上传”。
3. 根据指示上传镜像。

4. 上传成功后，在“我的镜像”界面可查看。

步骤四：创建函数

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”。
3. 填写函数基本信息。
   - 函数类型：选择“HTTP函数”。
   - 区域：默认，支持用户选择其他区域。

     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：输入“custom_container_http”。
   - 企业项目：默认“default”，支持用户选择已创建的企业项目。

     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。
   - 委托：使用包含SWR Admin权限的委托，如果没有委托，请参考**创建委托**。
   - 容器镜像：输入**步骤三**上传到SWR的镜像。（示例填写：swr.{局点id}.myhuaweicloud.com/{组织名称}/{镜像名称}:{版本名称}
4. 填写容器镜像覆盖参数（可选）。
   - 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。
   - 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。
   - 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。
   - 用户ID：输入用户ID。
   - 用户组ID：输入用户组ID。
5. 完成后单击“创建函数”。

步骤五：测试函数

1. 在函数详情页，单击“测试”，在弹窗中创建新的测试事件。
2. 选择“apig-event-template”，事件名称输入“helloworld”，测试事件修改为如下所示，完成后单击“创建”。

```
{
    "body": "{\"message\": \"helloworld\"}",
    "requestContext": {
        "requestId": "11cdcdcf33949dc6d722640a13091c77",
        "stage": "RELEASE"
    },
    "queryStringParameters": {
        "responseType": "html"
    },
    "httpMethod": "POST",
    "pathParameters": {},
    "headers": {
        "Content-Type": "application/json"
    },
```

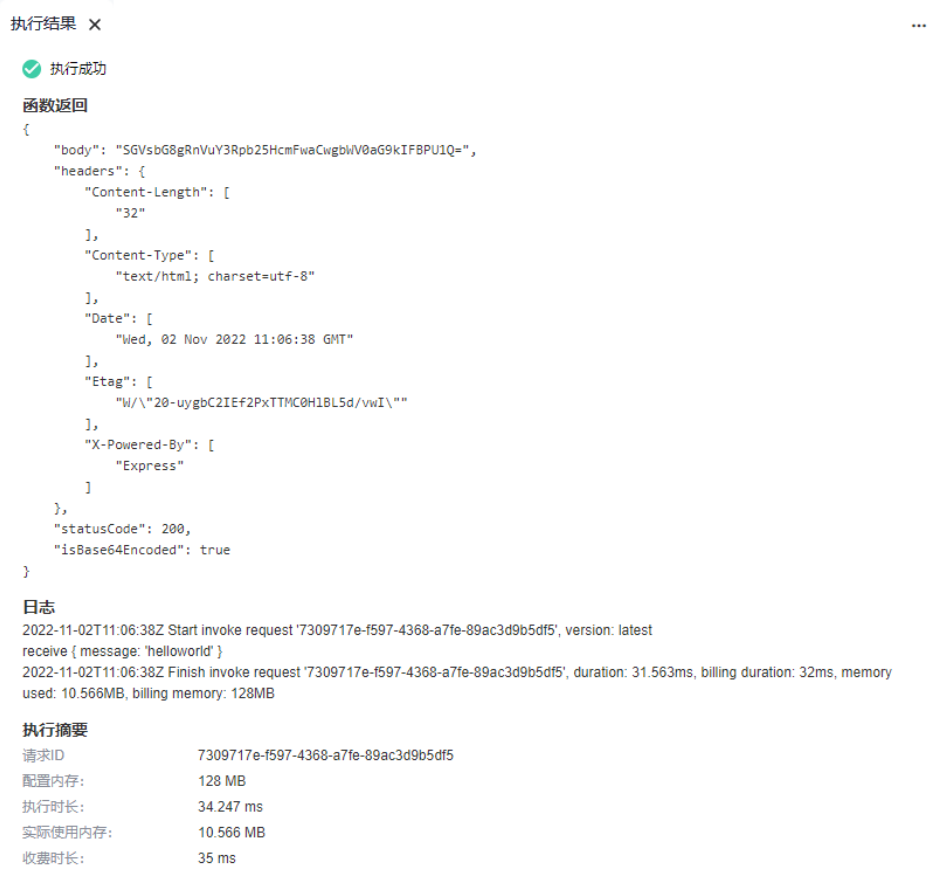


```
    "path": "/helloworld",
    "isBase64Encoded": false
}
```

步骤六：查看执行结果

单击helloworld事件的“测试”，执行后，在右侧查看执行结果，执行结果如下图。

图 4-1 执行结果



- “函数返回”显示函数的返回结果。
- “日志”部分显示函数执行过程中生成的日志。
- “执行摘要”部分显示“日志”中的关键信息。

说明

此页面最多显示2K日志，了解函数更多日志信息，请参考**配置和查看函数的调用日志**。

相关信息

- 了解函数工作流的相关概念，请参考**基本概念**。
- 了解函数工作流的价格，请参考**函数工作流价格详情**。
- 了解函数工作流的约束与限制，请参考**约束与限制**。

5 使用容器镜像创建并执行事件函数

本章节将以使用容器镜像方式创建事件函数为例，介绍容器镜像函数的创建及测试过程。

用户需要在镜像中实现一个http server，并监听8000端口接收请求。其中，请求路径/init 默认为函数初始化入口，请根据需要实现该接口。请求路径/invoke为函数执行入口，触发器事件转到该接口处理，请求参数请参见**函数支持的事件源**。备注：可以使用任意基础镜像。

约束与限制

- 在云上环境会默认使用uid 1003，gid 1003 启动容器。uid、gid可以在函数页面的“设置 > 常规设置 >容器镜像覆盖”板块中修改，但不可以是root或其他保留id。
- **如果使用Alpine版的基础镜像，请使用“addgroup”和“adduser”命令。**

准备工作

1. 注册华为账号并实名认证。

   在创建函数前，请先注册华为账号并实名认证，具体步骤请参考**注册华为账号并开通华为云**和**实名认证介绍**。

   如果您已有一个华为账号并实名认证，请跳过此步骤。

2. 免费额度。

   函数工作流服务每个月都会提供一定数量的免费额度，免费额度是子主账户共同使用，具体详情请参见**免费额度**。

   当免费额度使用完后，若您继续使用函数工作流时，账户的可用额度小于待结算的账单时，即被判定为账户欠费。欠费后，可能会影响您的服务资源的正常运行，请及时充值，具体详情请参考**账户充值**。

3. 为用户添加函数的操作权限。

   本章节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**。

步骤一：制作镜像

以在linux x86 64位系统上制作镜像为例。（ 系统配置无要求 ）

1. 创建一个空文件夹
   ```
   mkdir custom_container_event_example && cd custom_container_event_example
   ```
2. 以Nodejs语言为例，实现一个Http Server，处理函数初始化init请求和函数调用invoke请求并响应。

   创建一个main.js文件，引入express框架，实现Method为POST和Path为/invoke的函数执行入口，实现Method为POST和Path为/init的函数初始化入口。
   ```
   const express = require('express');

   const PORT = 8000;

   const app = express();
   app.use(express.json());

   app.post('/init', (req, res) => {
     console.log('receive', req.body);
     res.send('Hello init\n');
   });

   app.post('/invoke', (req, res) => {
     console.log('receive', req.body);
     res.send('Hello invoke\n');
   });

   app.listen(PORT, () => {
     console.log(`Listening on http://localhost:${PORT}`);
   });
   ```
3. 创建一个package.json文件，此文件用于向npm提供信息，使其能够识别项目以及处理项目的依赖关系。
   ```
   {
     "name": "custom-container-event-example",
     "version": "1.0.0",
     "description": "An example of a custom container event function",
     "main": "main.js",
     "scripts": {},
     "keywords": [],
     "author": "",
     "license": "ISC",
     "dependencies": {
        "express": "^4.17.1"
     }
   }
   ```
   - name：值为项目名。
   - version：值为项目版本。
   - main：列举文件为程序的入口文件。
   - dependencies：列出npm上可用的项目的所有依赖项。
4. 创建Dockerfile文件
   ```
   FROM node:12.10.0

   ENV HOME=/home/custom_container
   ENV GROUP_ID=1003
   ENV GROUP_NAME=custom_container
   ENV USER_ID=1003
   ENV USER_NAME=custom_container

   RUN mkdir -m 550 ${HOME} && groupadd -g ${GROUP_ID} ${GROUP_NAME} && useradd -u $
   {USER_ID} -g ${GROUP_ID} ${USER_NAME}

   COPY --chown=${USER_ID}:${GROUP_ID} main.js ${HOME}
   COPY --chown=${USER_ID}:${GROUP_ID} package.json ${HOME}

   RUN cd ${HOME} && npm install
   ```

```
RUN chown -R ${USER_ID}:${GROUP_ID} ${HOME}

RUN find ${HOME} -type d | xargs chmod 500
RUN find ${HOME} -type f | xargs chmod 500

USER ${USER_NAME}
WORKDIR ${HOME}

EXPOSE 8000
ENTRYPOINT ["node", "/home/custom_container/main.js"]
```

- FROM：指定基础镜像为node:12.10.0，基础镜像必须设置，值可修改。
- ENV：设置环境变量，设置HOME环境变量为/home/custom_container，设置GROUP_NAME和USER_NAME为custom_container，**USER_ID**和**GROUP_ID**为**1003**，这些环境变量必须设置，值可修改。
- RUN：格式为RUN <命令>，例如RUN mkdir -m 550 ${HOME}表示构建容器时创建${USER_NAME}用户的home目录。
- USER：切换${USER_NAME}用户。
- WORKDIR：切换工作目录到${USER_NAME}用户的“${HOME}”目录下。
- COPY：将main.js和package.json拷贝到容器的${USER_NAME}用户的home目录下。
- EXPOSE：暴露容器的8000端口，请勿修改。
- ENTRYPOINT：使用node /home/tester/main.js命令启动容器。

5. 构建镜像

   指定镜像的名称为custom_container_event_example，版本为latest，“.”指定Dockerfile所在目录，镜像构建命令将该路径下所有的内容打包给容器引擎帮助构建镜像。

   ```
   docker build -t custom_container_event_example:latest .
   ```

步骤二：本地验证

1. 启动docker容器

   ```
   docker run -u 1003:1003 -p 8000:8000 custom_container_event_example:latest
   ```

2. 打开一个新的命令行窗口，向开放的8000端口发送消息，访问模板代码中指定的/init路径

   ```
   curl -XPOST -H 'Content-Type: application/json' localhost:8000/init
   ```

   按照模块代码中返回

   ```
   Hello init
   ```

3. 打开一个新的命令行窗口，向开放的8000端口发送消息，访问模板代码中指定的/invoke路径

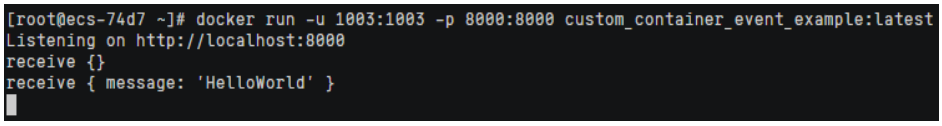
   ```
   curl -XPOST -H 'Content-Type: application/json' -d '{"message":"HelloWorld"}' localhost:8000/invoke
   ```

   按照模块代码中返回

   ```
   Hello invoke
   ```

4. 在容器启动端口可以看到

   ```
   Listening on http://localhost:8000
   receive {}
   receive { message: 'HelloWorld' }
   ```

   或者使用docker logs命令获取容器的日志

```
[root@ecs-74d7 custom_container_event_example]# docker logs 5560e1ec09d3
Listening on http://localhost:8000
receive {}
receive { message: 'HelloWorld' }
[root@ecs-74d7 custom_container_event_example]#
```

步骤三：上传镜像

1. 登录容器镜像服务控制台，在左侧导航栏选择“我的镜像”。
2. 单击右上角的“客户端上传”或“页面上传”。
3. 根据指示上传镜像。

4. 上传成功后，在“我的镜像”界面可查看。

步骤四：创建函数

1. 在服务控制台左侧导航栏，选择“计算 > 函数工作流”。进入函数工作流控制台后在左侧导航栏选择“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，创建方式选择“容器镜像”。
3. 填写函数基本信息。
   - 函数类型：选择“事件函数”。
   - 区域：默认，支持用户选择其他区域。

     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：输入“custom_container_http”。
   - 企业项目：默认“default”，支持用户选择已创建的企业项目。

     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。
   - 委托：使用包含SWR Admin权限的委托，如果没有委托，请参考**创建委托**。
   - 容器镜像：输入**步骤三**上传到SWR的镜像。（示例填写：swr.{局点id}.myhuaweicloud.com/{组织名称}/{镜像名称}:{版本名称}
4. 填写容器镜像覆盖参数（可选）。
   - 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。
   - 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。
   - 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。
   - 用户ID：输入用户ID。
   - 用户组ID：输入用户组ID。
5. 完成后单击“创建函数”。
6. 在函数详情页“设置 > 生命周期”，开启“初始化配置”，即调用init接口进行初始化。

步骤五：测试函数

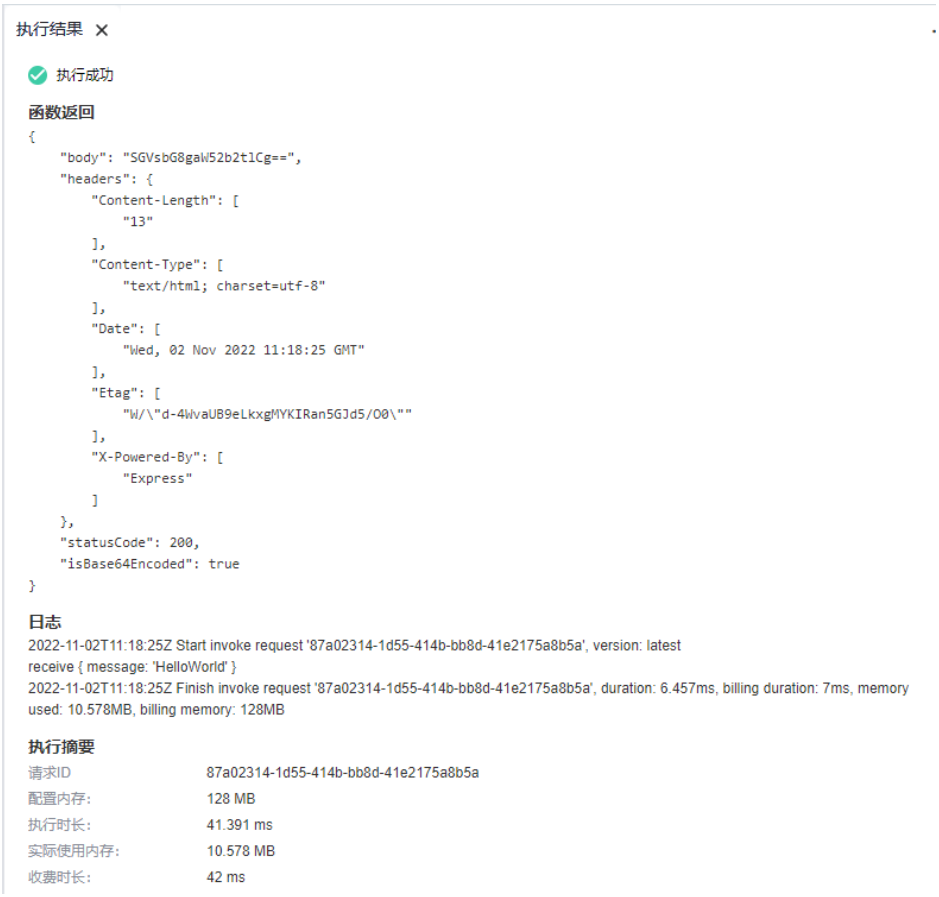
1. 在函数详情页，单击“测试”，在弹窗中创建新的测试事件。
2. 选择“空白模板”，事件名称输入“helloworld”，测试事件修改为如下所示，完成后单击“创建”。

```
{
  "message": "HelloWorld"
}
```

步骤六：查看执行结果

单击helloworld事件的“测试”，执行后，在右侧查看执行结果，执行结果如下图。

图 5-1 执行结果



- “函数返回”显示函数的返回结果。
- “日志”部分显示函数执行过程中生成的日志。
- “执行摘要”部分显示“日志”中的关键信息。

说明

此页面最多显示2K日志，了解函数更多日志信息，请参考**配置和查看函数的调用日志**。

相关信息

- 了解函数工作流的相关概念，请参考**基本概念**。

- 了解函数工作流的价格，请参考**函数工作流价格详情**。
- 了解函数工作流的约束与限制，请参考**约束与限制**。