



函数计算 快速入门

文档版本: 20220527



法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。 如果您阅读或使用本文档,您的阅读或使用行为将被视为对本声明全部内容的认可。

- 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档,且仅能用 于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息,您应当严格 遵守保密义务;未经阿里云事先书面同意,您不得向任何第三方披露本手册内容或 提供给任何第三方使用。
- 未经阿里云事先书面许可,任何单位、公司或个人不得擅自摘抄、翻译、复制本文 档内容的部分或全部,不得以任何方式或途径进行传播和宣传。
- 由于产品版本升级、调整或其他原因,本文档内容有可能变更。阿里云保留在没有 任何通知或者提示下对本文档的内容进行修改的权利,并在阿里云授权通道中不时 发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠 道下载、获取最新版的用户文档。
- 4. 本文档仅作为用户使用阿里云产品及服务的参考性指引,阿里云以产品及服务的"现状"、"有缺陷"和"当前功能"的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引,但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的,阿里云不承担任何法律责任。在任何情况下,阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害,包括用户使用或信赖本文档而遭受的利润损失,承担责任(即使阿里云已被告知该等损失的可能性)。
- 5. 阿里云网站上所有内容,包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计,均由阿里云和/或其关联公司依法拥有其知识产权,包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意,任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外,未经阿里云事先书面同意,任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称(包括但不限于单独为或以组合形式包含"阿里云"、"Aliyun"、"万网"等阿里云和/或其关联公司品牌,上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司)。
- 6. 如若发现本文档存在任何错误,请与阿里云取得直接联系。

通用约定

格式	说明	样例
⚠ 危险	该类警示信息将导致系统重大变更甚至故 障,或者导致人身伤害等结果。	⚠ 危险 重置操作将丢失用户配置数据。
▲ 警告	该类警示信息可能会导致系统重大变更甚 至故障,或者导致人身伤害等结果。	警告 重启操作将导致业务中断,恢复业务 时间约十分钟。
〔〕 注意	用于警示信息、补充说明等,是用户必须 了解的内容。	▶ 注意 权重设置为0,该服务器不会再接受新 请求。
⑦ 说明	用于补充说明、最佳实践、窍门等,不是 用户必须了解的内容。	⑦ 说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击设置> 网络> 设置网络类型。
粗体	表示按键、菜单、页面名称等UI元素。	在 结果确认 页面,单击 确定 。
Courier字体	命令或代码。	执行 cd /d C:/window 命令,进入 Windows系统文件夹。
斜体	表示参数、变量。	bae log listinstanceid
[] 或者 [alb]	表示可选项,至多选择一个。	ipconfig [-all -t]
{} 或者 {alb}	表示必选项,至多选择一个。	switch {act ive st and}

目录

1.使用流程	05
2.开通服务	07
3.使用控制台创建函数	09
4.使用Serverless Devs管理函数资源	11
5.使用VSCode插件管理函数资源	18

1.使用流程

本文介绍使用函数计算的流程,包括创建服务、创建函数、触发函数、查看执行日志和查看服务监控。

背景信息

您可以使用函数计算控制台、Serverless Devs或者VSCode插件创建函数和查询日志等。更多信息,请参见 以下文档:

- 使用控制台创建函数
- 使用Serverless Devs管理函数资源
- 使用VSCode插件管理函数资源

使用流程

使用函数计算前, 您需要在产品详情页开通服务。

函数计算使用流程图如下所示。



流程说明如下:

- 1. 创建服务。
- 2. 创建函数, 编写代码, 将应用部署到函数中。
- 3. 触发函数。
- 4. 查看执行日志。
- 5. 查看服务监控。

创建服务

服务(Service)是函数计算的基本资源单位。您可以在服务级别上授权、配置日志和创建函数等。更多信息,请参见管理服务。

创建函数

函数(Function)是调度与运行的基本单位,更是一段代码的处理逻辑。您需要根据函数计算提供的函数接 口形式编写代码,并将代码以函数的形式部署到函数计算。函数计算中的服务对应于软件应用架构领域中的 微服务。在函数计算平台构建应用时,首先根据需求将业务逻辑抽象为微服务,然后再实现为函数计算中的 服务。

一个服务下可以创建多个函数,每个函数可以设置不同的内存规格、环境变量等属性,并可以结合您的实际 业务场景来决定是否开启Initializer功能。更多信息,请参见Initializer函数。

服务是函数层次化的抽象,在系统抽象和实现灵活度上能够取得平衡。例如,实现一个微服务,需要调用阿 里云语音合成服务,将文字转成语音,再把这段语音和一系列图片组合为视频。其中文字转语音函数是调用 其他服务,可以设置很小的内存规格。而视频合成函数是计算密集型,需要更大的内存。因此您可以组合多 个不同规格的函数实现微服务,优化成本。关于函数的创建、更新和删除等,请参见管理函数。

触发函数

函数计算支持直接触发函数或通过事件触发函数。您可以根据需要选择合适的触发方式:

- 使用函数计算控制台、Serverless Devs或SDK等方式直接触发函数的执行。更多信息,请参见以下文档:
 - 使用控制台创建函数
 - o 使用s invoke的相关命令调用函数
 - 使用SDK执行HTTP触发器函数
- 配置函数计算触发器,通过事件触发函数的执行。例如配置OSS触发器后,当OSS对应的Bucket中有对象 新增或删除后都会触发函数的执行,方便您处理上传的对象;配置日志服务触发器后,当日志服务对应的 Logstore中有新日志写入后可以触发函数的执行,方便您处理写入的日志。您需要设置触发器来设置事件 触发的方式。更多信息,请参见以下文档:
 - o 触发器简介
 - 触发器管理

查看执行日志

查看日志是帮助您调试的一个重要环节。关于使用函数计算配置日志并查看日志的操作步骤,请参见配置日 志。

查看服务监控

您可以在函数计算控制台上查看服务监控。更多信息,请参见以下文档:

- 监控指标
- 监控数据

2.开通服务

本文介绍如何开通函数计算服务。

前提条件

您已成功注册阿里云账号,并完成实名认证。具体信息,请参见阿里云账号注册流程。

操作步骤

- 1. 进入函数计算首页。
- 2. 单击管理控制台。

< 查看全部产品	
函数计算 FC ® 購放 服務	
函数计算(Function Compute)是一个事件驱动的全托管 Serverless 计算服务,您无需管理服务器等基础设施,只 需编写代码并上传,函数计算会为您准备好计算资源,并以弹性、可靠的方式运行您的代码。	
阿里云云原生 Servertess 技术实践营(广州站)——精彩回顾	
购买资源包 管理控制台 产品文档 产品价格	

- 3. 在概览页面,单击免费开通。
- 4. 在开通说明面板,认真阅读并选中**函数计算服务协议**复选框,单击**立即开通**,然后单击**关闭**,进入函 数计算控制台。
- 5. (可选)在左侧导航栏,单击**概览**,在**云资源访问授权**对话框,单击**同意授权**给您的账号授 予AliyunFCDef ault Role角色。

云资源访问授权 如素修改角色权限,请前往 RAM 控制台角色管理中设置,需要注意的是,错误的 配置可能导致 CloudMonitor 无法获取到必要的权限。	×
 FC 请求获取访问您云资源的权限。 下方是系统创建的可供 FC 使用的角色,授权后,FC 拥有对您云资源相应的访问权限。 AliyunFCDefaultRole 展开详情 	

。 授权成功

授权成功即函数计算具有访问云资源的权限。

- 授权失败
 - 阿里云账号

可能由于一些功能无法授权造成的。但函数计算仍可以正常使用。如果您仍需授予系统默认角色, 请提交工单咨询。

■ RAM用户

当您授权失败后您需要判断您的账号是否有授权能力。

- 如果没有RAM权限,您需要通过阿里云账号登录函数计算控制台,然后进行授权操作。授权成功 后,您通过RAM用户再次登录函数计算控制台时将不再提示您进行授权。
- 如果已有RAM权限依旧授权失败,可能由于一些功能无法授权造成的。但函数计算仍可以正常使用。如果您仍需授予系统默认角色,请提交工单咨询。

关于权限管理的详细信息,请参见权限管理。

3.使用控制台创建函数

本文介绍如何在函数计算控制台快速开发一个简单的Serverless Hello World应用。

前提条件

开通函数计算

背景信息

函数计算为您提供了简单的Hello World示例,您可以直接在函数计算控制台创建服务和函数,成功创建后可 以立马测试该函数查看函数执行的详细信息,例如返回结果、执行摘要和日志输出。

本文以从零开始创建的方式为例,介绍如何创建函数。

操作步骤

- 1. 创建服务。
 - i. 登录函数计算控制台。
 - ii. 在左侧导航栏, 单击**服务及函数**。
 - iii. 在顶部菜单栏,选择地域。
 - iv. 在服务列表页面, 单击创建服务。
 - v. 在创建服务面板,填写服务名称和描述,然后单击确定。
 您也可以在创建服务面板,设置是否启用阿里云日志服务和是否启用阿里云链路追踪功能。详细信息,请参见管理服务。

当您成功创建服务后,页面会跳转至该服务页面的函数管理页面。

- 2. 创建函数。
 - i. 在管理函数页面, 单击创建函数。
 - ii. 在创建函数页面,选择从零开始创建。
 - iii. 在基本设置区域,设置相关参数,然后单击创建。
 参数配置如下:
 - (可选)函数名称: 自定义函数的名称为Function。
 - 运行环境:选择Node.js 12。
 - 请求处理程序类型:选择处理事件请求。
 - **实例类型**:选择弹性实例。
 - 内存规格:设置函数执行内存为512 MB。
 - 实例并发度:设置为1。
 - 请求处理程序:设置为index.handler。

当您成功创建函数后,页面会跳转至该函数详情页面的函数代码页签。

3. 执行函数。

在目标函数详情页面的**函数代码**页签,单击测试函数。

⑦ 说明 您也可以在代码编辑器内编辑您的代码,实现二次开发。

执行完毕后,您可以在函数代码页签查看执行结果和详细的日志信息。

4.使用Serverless Devs管理函数资源

Serverless Devs是一个开源开放的Serverless开发者平台,通过该平台您可以快速地开发、创建、测试和部署项目,实现项目的全生命周期管理。本文以部署运行环境是Node.js 12的函数为例,介绍如何使用 Serverless Devs下载对应的函数计算应用模板,然后快速开发部署函数应用。

背景信息

本文介绍两种方式初始化项目。当您初次使用Serverless Devs时,建议您使用方式一体验Serverless Devs。 如果您想了解关于Serverless Devs的更多信息,例如本地调试、远程调用、删除资源等操作,建议您使用方 式二体验。

前提条件

- 1. 安装Serverless Devs和Docker
- 2. 配置Serverless Devs

方式一:使用s命令管理资源

1. 执行以下命令, 初始化项目。

S

请根据输出的交互提示,初始化项目。输出示例:

```
No Serverless-Devs project is currently detected. Do you want to create a new project?
Yes//在项目文件内未检测到Serverless-Devs项目。选择是否创建一个项目。
Serverless Awesome: https://github.com/Serverless-Devs/Serverless-Devs/blob/master/doc
s/zh/awesome.md
Hello Serverless for Cloud Vendors Alibaba Cloud Serverless//选择目标云厂商,本示例以阿里
云为例。
Hello, serverlesser. Which template do you like? Quick start [Deploy a Hello World fun
ction to FaaS]//选择您需要创建的模板。本示例以快速部署运行环境为Node.js 12的事件函数为例。
Which template do you like? [Event] Node.js 12
Create application command: [s init devsapp/start-fc-event-nodejs12]
Please input your project name (init dir) start-fc-event-nodejs12//Serverless Devs默认
为您生成一个项目名称,您也可以自定义该名称。
file decompression completed
   | / \| | | | \ / | | || |\ | | _ \| /
   please select credential alias default//选择您在配置Serverless Devs时设置的别名。
   Welcome to the Aliyun FC start application
    This application requires to open these services:
       FC : https://fc.console.aliyun.com/
    * 项目初始化完成,您可以直接进入项目目录下,并使用s deploy进行项目部署。
Thanks for using Serverless-Devs
You could [cd /test/start-fc-event-nodejs12] and enjoy your serverless journey!
If you need help for this example, you can use [s -h] after you enter folder.
Document Star: https://github.com/Serverless-Devs/Serverless-Devs
Do you want to deploy the project immediately? Yes//选择是否部署项目,本示例以选择部署项目
为例。
. . . . . .
helloworld:
 region: cn-hangzhou
 service:
  name: hello-world-service
 function:
            event-nodejs12
   name:
            nodejs12
   runtime:
   handler: index.handler
   memorySize: 128
   timeout:
            60
```

方式二: 使用s init命令管理资源

1. 初始化项目。

在目标文件内执行以下命令,初始化项目:

```
s init devsapp/start-fc-http-nodejs12

⑦ 说明 本示例以部署一个运行环境为Node.js12的HTTP函数为例,介绍如何快速初始化一个项
目。当需要部署其他运行环境的示例时,您可以将 http 替换为其他类型的触发方式例如 event
,也可以将 nodejs12 替换为其他类型的运行环境。
```

执行输出:



- *s.yaml*:函数计算资源,该资源是以YAML规范定义的。关于YAML规范的详细信息,请参见YAML规范。
- 。 code: code文件夹内包含 index.js文件,可在该文件内编辑函数代码逻辑。
- readme.md: 依赖清单文件, 该文件内定义了项目所需的各个模块和项目的配置信息。
- 2. 执行以下命令,进入项目目录内:

```
cd start-fc-http-nodejs12
```

3. (可选)构建应用。

在项目目录内执行以下命令,构建应用:

s build

执行成功该步骤后,会在当前目录中生成一个.*s*文件,存放输出的构建产物,该文件可用于本地调试和 部署。

4. (可选)本地调试应用。

当您在本地完成应用的开发构建后,可能需要在本地调试验证功能是否正常。您可以按需选择命令调试 函数:

○ HTTP函数

s local start

```
[2021-12-16 07:17:05] [INFO] [FC-LOCAL-INVOKE] - Using trigger for start: name: httpT
rigger
type: http
config:
  authType: anonymous
 methods:
   - GET
[2021-12-16 07:17:05] [INFO] [FC-LOCAL-INVOKE] - HttpTrigger httpTrigger of hello-wor
ld-service/http-trigger-nodejs12 was registered
       url: http://localhost:7013/2016-08-15/proxy/hello-world-service/http-trigger-
nodejs12/
       methods: GET
        authType: anonymous
Tips: you can also use these commands to run/debug custom domain resources:
Start with customDomain:
* s local start auto
Debug with customDomain:
* s local start -d 3000 auto
Tips for next step
     _____
* Deploy Resources: s deploy
helloworld:
 status: succeed
function compute app listening on port 7013!
```

• 事件函数

s local invoke

当您执行调试命令后,由于函数类型不同返回结果也会不同:

○ HTTP函数

当您的函数是HTTP函数时,执行完调试命令后,将在执行输出中返回一个URL地址,此时调试流程阻 塞在这里,您可以使用以下方式触发函数执行:

■ 使用cURL工具触发函数执行

curl http://localhost:7013/2016-08-15/proxy/hello-world-service/http-trigger-nodejs
12/

■ 使用浏览器访问HTTP函数

http://localhost:7013/2016-08-15/proxy/hello-world-service/http-trigger-nodejs12/

事件函数

如果您的函数是事件函数,执行完调试命令后,将立即返回调试结果。

5. 部署应用。

执行以下命令,将应用部署至函数计算:

s deploy -y

```
Checking Service, Function, Triggers (0.82s)
Creating Service, Function, Triggers (0.6s)
Creating custom domain (0.31s)
Tips for next step
_____
* Display information of the deployed resource: s info
* Display metrics: s metrics
* Display logs: s logs
* Invoke remote function: s invoke
* Remove Service: s remove service
* Remove Function: s remove function
* Remove Trigger: s remove trigger
* Remove CustomDomain: s remove domain
helloworld:
 region: cn-hangzhou
 service:
   name: hello-world-service
 function:
   name: http-trigger-nodejs12
   runtime: nodejs12
   handler: index.handler
   memorySize: 128
   timeout: 60
 url:
   custom domain:
       domain: http://http-trigger-nodejs12.hello-world-service.188077086902****.cn-ha
ngzhou.fc.devsapp.net
 triggers:
   _
     type: http
     name: httpTrigger
```

执行成功后,该命令会将应用部署至函数计算。

6. (可选)远程调试应用。

执行以下命令,远程调试应用:

```
s invoke -e '{"body": "","method":"GET","headers":{"header_1":"v1"},"queries":{"key":"v
alue"},"path":"/"}'
```

```
函数计算
```

```
Request url: https://188077086902****.cn-hangzhou.fc.aliyuncs.com/2016-08-15/proxy/hell
o-world-service/http-trigger-nodejs12/
====== FC invoke Logs begin =======
FC Invoke Start RequestId: 6a3adedb-5d5d-4ac5-aa0b-2d43f4ace65c
load code for handler:index.handler
2022-03-04T08:39:14.770Z 6a3adedb-5d5d-4ac5-aa0b-2d43f4ace65c [verbose] hello world
FC Invoke End RequestId: 6a3adedb-5d5d-4ac5-aa0b-2d43f4ace65c
Duration: 66.23 ms, Billed Duration: 67 ms, Memory Size: 128 MB, Max Memory Used: 42.48
MB
====== FC invoke Logs end =======
FC Invoke Result[Code: 200]:
{
 path: '//',
 queries: { key: 'value' },
 headers: {
   accept: 'application/json',
   authorization: 'FC LTAI4G4cwJkK4Rza6xd9****:KZsoanKle4SoRO8EjyOjY3duLlVlR1mp78hjV9W
G****',
   date: 'Fri, 04 Mar 2022 08:39:14 GMT',
   header 1: 'v1',
    host: '188077086902****.cn-hangzhou.fc.aliyuncs.com',
    'user-agent': 'Node.js(v14.17.4) OS(linux/x64) SDK(@alicloud/fc2@v2.5.0)',
    'x-forwarded-proto': 'http'
 },
 method: 'GET',
 requestURI: '/2016-08-15/proxy/hello-world-service/http-trigger-nodejs12//?key=value'
 clientIP: '47.97.XX.XX',
 body: ''
}
End of method: invoke
```

由于本示例部署的函数是HTTP函数,所以在成功执行远程调试命令后,执行输出会返回一个URL,您可 以使用cURL工具触发执行线上的函数:

curl https://188077086902****.cn-hangzhou.fc.aliyuncs.com/2016-08-15/proxy/hello-worldservice/http-trigger-nodejs12/

7. (可选)删除应用。

执行以下命令删除应用:

s remove service

Need to delete the resource in the cn-hangzhou area, the operation service is hello-wor ld-service: Delete trigger hello-world-service/http-trigger-nodejs12/httpTrigger success. Delete function hello-world-service/http-trigger-nodejs12 success. Delete service hello-world-service success. [2022-03-04 08:40:26] [INFO] [FC] - Getting list on-demand: FC NAS hello-world-service -ensure-nas-dir-exist-service [2022-03-04 08:40:26] [INFO] [FC] - Getting list provision: _FC_NAS_hello-world-service -ensure-nas-dir-exist-service [2022-03-04 08:40:26] [INFO] [FC] - Getting listAliases: _FC_NAS_hello-world-service-en sure-nas-dir-exist-service [2022-03-04 08:40:26] [INFO] [FC] - Getting list on-demand: _FC_NAS_hello-world-service [2022-03-04 08:40:26] [INFO] [FC] - Getting list provision: _FC_NAS_hello-world-service [2022-03-04 08:40:26] [INFO] [FC] - Getting listAliases: _FC_NAS_hello-world-service End of method: remove

当成功执行该命令时,表示应用已被删除。

5.使用VSCode插件管理函数资源

本文介绍了如何通过VSCode插件创建、部署、调试函数资源。

背景信息

Aliyun Serverless VSCode Extension是函数计算提供的VSCode插件。通过该插件,您可以完成以下操作:

- 在本地快速地初始化项目和创建函数。
- 运行、调试本地函数。调试功能支持Node.js、Python和PHP编程语言。
- 拉取云端的服务和函数至VSCode中。
- 部署资源至云端,并更新相关参数配置。

前提条件

您已完成以下操作:

- 开通服务
- 下载安装Visual Studio Code
- 安装Docker

编译代码、安装依赖和在本地运行调试函数等操作都是在Docker镜像中进行的。

• 安装Aliyun Serverless VSCode Extension插件

操作步骤

- 1. 绑定阿里云账号。
 - i. 打开VSCode,在左侧导航栏,单击 🚾 图标,单击Bind New Account 绑定阿里云账号。



ii. 依次输入阿里云账号(主账号)、AccessKey ID、AccessKey Secret和自定义的账号别名(即本地账号名称)。

Bind Aliyun Account (Input Account ID) (1/4)

Input Account ID (Press 'Enter' to confirm or 'Escape' to cancel)

绑定完成后,您可以看到成功绑定的阿里云账号在云端的服务和函数。

~ R	EMOTE RESOURCES
>	😭 demo
>	😭 demo1
>	😚 demo2
>	😚 demo3
>	😭 guide-hello_world
>	🕥 myapp1-FcOssFFmpeg-B6DFA
>	😚 service
>	😚 test123-QualifierHelper-C5F46
>	😚 test123-test123-D7F8559D2BCC

您也可以在云端资源面板切换地域查看不同地域的服务和函数。



- 2. 创建函数。
 - i. 通过VSCode插件, 打开一个空文件夹。
 - ii. 在LOCAL RESOURCES区域,单击加号图标,在本地初始化一个函数计算项目。

ALIYUN: FUNCTION COM				
✓ LOCAL RESOURCES	+	ß	Q	•••

iii. 按照导航依次输入或选择服务名称、函数名称、函数运行环境和函数类型。配置完成后,插件会自动创建函数并在本地资源面板中会展示新建的本地服务与函数。

ALIYUN: FUNCTION COMPUTE	•••	JS index.js	×
 ✓ LOCAL RESOURCES ✓ template.yml 		demo01 > te 1 'us	testFuncO1 > JS index.js > ☆ handler > ☆ handler use strict';
✓ ⑦ demo01 <i>f</i> testFunc01			<pre> f you open the initializer feature, please implement the initializer fun odule.exports.initializer = function(context, callback) { console.log('initializing'); callback(null, ''); ; </pre>
		8 7 Loc 9 moo 10 0 11 0 12 }	<pre>ccalRun LocalDebug InvokePanel ocalRun LocalDebug InvokePanel odule.exports.handler = function(event, context, callback) { console.log(new String(event)); callback(null, 'hello world');</pre>

您也可以直接单击LOCAL RESOURCES中服务名右侧的加号图标,为该服务创建函数。按照导航依次输入或选择函数名称、函数运行时、函数类型即可。

ALIYUN: FUNCTION COMPUTE ····	JS index.js ×
✓ LOCAL RESOURCES + ♠ ♂ ···	demoO1 > testFuncO1 > JS index.js > 1 'use strict':
✓ I demo01 + <i>f</i> testFunc01	<pre>2 /* 3 if you open the initializer feature, please implement the initializer fun 4 module.exports.initializer = function(context, callback) { 5 console.log('initializing'); 6 callback(null, ''); 7 };</pre>
	<pre>8 */ Local Run Local Debug Invoke Panel 9 module.exports.handler = function(event, context, callback) { 10 console.log(new String(event)); 11 callback(null, 'hello world'); 12 }</pre>

3. 本地调试函数。

↓ 注意

- 调试Python 2.7或Python 3函数, 需要安装Python插件。
- 调试PHP 7.2函数, 需要安装PHP Debug插件。
- i. 在本地资源面板中,单击函数名称右侧的调试图标或 handler 文件中的Local Debug,调试该 函数。

ALIYUN: FUNCTION COMPUTE ····	JS index.js X
〜 LOCAL RESOURCES 🛛 🕂 🔥 😷	demo01 > testFunc01 > JS index.js > \bigcirc handler > \bigcirc handler
✓ template.yml ✓	<pre>4 module.exports.initializer = function(context, callback) { 5</pre>
f testFunc01 <⁄> 🗼 🕨	6 callback(null, '');
	Local Run Local Debug Invoke Panel
	<pre>9 module.exports.handler = function(event, context, callback) { 40</pre>
	<pre>10 console.log(new string(even()); 11 callback(null, 'hello world');</pre>
	12 }

ii. 在代码文件中插入断点,启动调试后可查看调试信息。



插件会为您在函数入口文件同目录下创建 event.dat 文件,您可以通过修改该文件设置每次调试函数 时触发的事件信息。

4. (可选)本地调用函数。

在LOCAL RESOURCES区域中,单击函数名称右侧的执行图标或 handler 文件中的Local Run,调用 该函数。



函数日志和结果会输出到终端。



插件会为您在函数入口文件同目录下创建 event.dat 文件,您可以通过修改该文件设置每次调用函数 时触发的事件信息。

EXPLORER	Js index.js ×		{} event.dat ×
<pre>> OPEN EDITORS > SAMPLE10 demo01 testFunc01 ly event.dat js index.js template.yml</pre>	<pre>demo01 > testFunc01 > Js index.js > @ handler > @ handler 1 'use-strict'; 2 /# 3 if you open the initializer - feature, please implement th 4 module exports.initializer = function(context, callback) 5 - console.log('initializing'); 6 - callback(null, ''); 7); 8 */ Local Run Local Debug Invoke Panel module exports.handler = function(event, context, callback) 1 - callback(null, 'hello world'); 12 }</pre>	Generation Technology	<pre>demc01 + testFunc01 > () event.dat > 1 2 - "*key": "value", 3 - "name": "Function Compute" 4 </pre>
	PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL FC Invoke Start RequestId: load code for handler: index.handler 2019-09-1811:228:35,752 FC Invoke End RequestId: hello world RequestId: Billed	erbose] [St Duration: 5	T: Function Compute#1 + C C ring: '{ "key": "value", "name": "Function Compute"}'] 9 ms Memory Size: 1998 MB Max Memory Used: 19 MB

- 5. 部署服务与函数。
 - i. 单击本地资源面板中的部署图标,将本地的服务与函数部署到云端。

ALIYUN: FUNCTION COMPUTE		JS			
▲ LOCAL RESOURCES + ▲ ♂ … ▲ ♂ demo01 +					
f testFunc01					

ii. 部署完成后, 单击云端资源面板中的刷新图标, 查询部署到云端的服务与函数。



6. 执行云端函数。

单击云端资源面板中函数右侧的执行图标,执行云端函数。



函数日志和结果会输出在TERMINAL中。



插件会为您在项目根目录下创建 event.dat 文件,您可以通过修改 event.dat 文件设置每次调用云 端函数时触发的事件信息。

7. 云端测试。

您可以在函数计算控制台查看是否部署成功。

- i. 登录函数计算控制台。
- ii. 在左侧导航栏, 单击**服务及函数**。
- iii. 在顶部菜单栏,选择地域。
- iv. 在**服务列表**页面,单击目标服务。
- v. 在函数管理页面,单击目标函数名称。
- vi. 在函数详情页面的函数代码页签,单击测试函数,即可在函数计算控制台执行函数。
- 8. 查看日志。

每次执行完毕,可以在当前页面查看本次执行日志。如果需要查看历史执行日志,可以单击**调用日志**页 签,查看历史执行日志时需为函数配置日志仓库。详细信息,请参见<mark>配置日志</mark>。