

函数 workflow

# 常见问题

文档版本 01  
发布日期 2024-11-11



版权所有 © 华为云计算技术有限公司 2024。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。

# 目录

<b>1 通用问题</b>	<b>1</b>
1.1 FunctionGraph 是什么	1
1.2 使用 FunctionGraph 是否需要开通计算、存储、网络等服务?	1
1.3 使用 FunctionGraph 开发程序之后是否需要部署?	1
1.4 如何获取 Token?	1
1.5 FunctionGraph 函数支持哪些编程语言?	2
1.6 FunctionGraph 函数分配磁盘空间有多少?	3
1.7 FunctionGraph 函数是否支持版本控制?	3
1.8 函数中如何读写文件?	3
1.9 使用 CLI 工具, 如何配置网络代理?	4
1.10 FunctionGraph 函数是否支持扩展?	4
1.11 IAM 子账号使用 FunctionGraph 需要设置哪些权限?	4
1.12 如何制作基于 ODBC 驱动的 Python 依赖包用于查询数据库?	4
1.13 FunctionGraph 配额	5
1.14 FunctionGraph 函数支持哪些中文字体?	5
1.15 FunctionGraph 函数如何解析 DNS 内网域名?	5
1.16 容器镜像函数如何解析 DNS 内网域名?	6
1.17 如何通过域名访问专享版 APIG 中注册的接口?	7
1.18 函数工作流的常见使用场景?	7
1.19 函数调用绑定在 APIG 的域名的服务, 报域名无法解析?	7
1.20 同步函数工作流能否支持到内网最大带宽的同步传输?	8
1.21 单租户的 VPC 超过默认配额时, 需要怎么做?	8
1.22 如何打印 info、error、warn 级别的日志?	8
1.23 函数是否可以把 API 的接口域名配置成自己的域名?	8
1.24 函数工作流是否支持修改运行时语言?	8
1.25 已创建的函数是否支持修改函数名称?	8
1.26 挂载文件系统时, 报“failed to mount exist system path”, 应如何处理?	8
1.27 如何获取上传的文件?	9
1.28 同步调用响应未收到的可能原因?	9
1.29 os.system("command &")执行日志未采集, 应如何处理?	9
1.30 自定义运行时, 都能操作哪些目录?	9
1.31 运行时语言支持的 python3.6 和 3.9 具体指哪个版本?	9
1.32 用户想使用 vpc 功能, 但不想配置 VPC Administrator 委托, 应配置哪些授权项?	9

1.33 函数执行超时的可能原因有哪些? .....	10
1.34 如何获取函数代码? .....	10
1.35 是否有 initializer 的代码示例? .....	10
1.36 如何开启结构化日志查询.....	10
1.37 函数服务是否支持在函数中启动 TCP 的监听端口, 通过 EIP 接收外部发送过来的 TCP 请求? .....	14
1.38 FunctionGraph 是否支持域名解析? .....	14
1.39 函数发起 HTTP 请求的源地址如何获取? .....	14
<b>2 创建函数.....</b>	<b>16</b>
2.1 能否在函数代码中使用线程和进程? .....	16
2.2 FunctionGraph 函数工程打包有哪些规范 ( 限制 ) ? .....	16
2.3 FunctionGraph 如何隔离代码? .....	20
2.4 HTTP 函数 bootstrap 启动文件如何创建? .....	20
<b>3 触发器管理.....</b>	<b>21</b>
3.1 哪些事件可以触发 FunctionGraph 函数? .....	21
3.2 使用 APIG 触发器调用一个返回 String 的 FunctionGraph 函数, 报 500 错误, 该如何解决? .....	21
3.3 DIS 触发器中起始位置 LATEST 和 TRIM_HORIZON 是什么意思? .....	22
3.4 如何使用 APIG 触发器调用函数? .....	22
3.5 使用 APIG 触发器, 函数如何获取请求路径或请求参数? .....	22
3.6 FunctionGraph 和 kafka 必须在同一个子网内, 才可以在 FunctionGraph 中配置 Kafka 触发器吗? .....	23
<b>4 依赖包管理.....</b>	<b>24</b>
4.1 依赖包是什么? .....	24
4.2 什么场景下需要引入依赖? .....	24
4.3 使用依赖包时, 有哪些注意事项? .....	24
4.4 函数支持引入的依赖库有哪些? .....	24
4.5 使用 FunctionGraph 开发函数能否引入类库? .....	25
4.6 如何在 FunctionGrap 上使用第三方依赖? .....	26
4.7 如何制作函数依赖包? .....	26
4.8 如何在函数平台创建依赖包? .....	28
4.9 如何为函数添加依赖包? .....	29
<b>5 函数执行.....</b>	<b>30</b>
5.1 FunctionGraph 函数的执行需要多长时间? .....	30
5.2 FunctionGraph 函数的执行包含了哪些过程? .....	30
5.3 FunctionGraph 函数的并发处理过程是什么? .....	30
5.4 FunctionGraph 函数如何处理长时间不执行的实例? .....	30
5.5 首次访问函数慢, 如何优化? .....	31
5.6 怎样获取在函数运行过程中实际使用了多少内存? .....	31
5.7 为什么第一次请求会比较慢? .....	31
5.8 调用 API 时, 报错怎么办? .....	31
5.9 如何读取函数的请求头? .....	31
5.10 API 同步执行函数接口, 是否支持内网调用? .....	32
5.11 为什么函数实际使用内存大于预估内存, 甚至触发 OOM? .....	32

5.12 函数内存超限返回“runtime memory limit exceeded”，如何查看内存占用大小？	32
5.13 如何定位自定义镜像执行失败“CrashLoopBackOff”的原因？	32
5.14 用户使用相同的镜像名更新镜像，预留实例无法自动更新，会一直使用老镜像，应如何处理？	33
<b>6 函数配置</b>	<b>34</b>
6.1 FunctionGraph 函数是否支持环境变量？	34
6.2 能否在函数环境变量中存储敏感信息？	34
6.3 如何通过父函数调用子函数？	34
6.4 如何通过版本和别名实现绑定 APIG 触发器的 HTTP 函数的灰度升级？	34
<b>7 函数访问外部资源</b>	<b>38</b>
7.1 函数如何访问 MySQL 数据库？	38
7.2 函数如何访问 Redis？	39
7.3 函数无法通过 VPC 连接对应的 Redis？	39
7.4 如何配置外网访问？	40
<b>8 其他问题</b>	<b>41</b>
8.1 如何查看给函数配置的告警规则？	41
8.2 视频转码，上传的 zip 文件是否能支持反编译？	41
8.3 申请的 FunctionGraph2.0 公测资源，到期是否自动释放？如何计费？	41
8.4 函数常规信息中的“应用”应如何理解？	41
8.5 用户需要为冷启动时间付费吗？	41
8.6 创建函数时，为什么提示“已限制，不能执行此操作”？	41
8.7 函数计费中的调用次数，是某一账号下在不同 region 的所有函数的调用次数总和吗？	42
<b>9 V1 迁移 V2 相关问题</b>	<b>43</b>
9.1 Python 语言从 V1 迁移到 V2 时需注意哪些兼容性问题？	43
9.2 Python2.7 在执行 reload(sys)后无法通过 print 打印日志应如何解决？	43

# 1 通用问题

## 1.1 FunctionGraph 是什么

函数工作流（FunctionGraph）是一项基于事件驱动的函数托管计算服务。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。

## 1.2 使用 FunctionGraph 是否需要开通计算、存储、网络等服务？

用户使用FunctionGraph时，不需要开通或者预配置计算、存储、网络等服务，由FunctionGraph提供和管理底层计算资源，包括服务器CPU、内存、网络和其他配置/资源维护、代码部署、弹性伸缩、负载均衡、安全升级、资源运行情况监控等，用户只需要按照FunctionGraph支持的编程语言提供程序包，上传即可运行。

## 1.3 使用 FunctionGraph 开发程序之后是否需要部署？

用户在本地开发程序之后打包，必须是ZIP包（Java、Node.js、Python、Go）或者JAR包（Java），上传至FunctionGraph即可运行，无需其它的部署操作。

制作ZIP包的时候，单函数入口文件必须在根目录，保证解压后，直接出现函数执行入口文件，才能正常运行。

对于Go runtime，必须在编译之后打zip包，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。

## 1.4 如何获取 Token？

调用接口时，可以使用Token进行认证鉴权，可以通过IAM标准接口获取Token，示例如下。

- 获取华南区的Token示例命令如下。

```
curl -k -i -X POST https://iam.cn-south-1.myhuaweicloud.com/v3/auth/tokens -H
'Content-Type: application/json' -d '{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "name": "华为云账号",
          "password": "华为云登录密码",
          "domain": {
            "name": "华为云账号"
          }
        }
      }
    },
    "scope": {
      "project": {
        "name": "cn-south-1"
      }
    }
  }
}'
```

- 获取华北区的Token示例命令如下。

```
curl -k -i -X POST https://iam.cn-north-1.myhuaweicloud.com/v3/auth/tokens -H
'Content-Type: application/json' -d '{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "name": "华为云账号",
          "password": "华为云登录密码",
          "domain": {
            "name": "华为云账号"
          }
        }
      }
    },
    "scope": {
      "project": {
        "name": "cn-north-1"
      }
    }
  }
}'
```

#### 📖 说明

返回头中的X-Subject-Token对应的值即为获取的token值。例如获取华北-北京一区域的token，只能用于调用华北-北京一的FunctionGraph服务。

具体请参考[获取用户Token](#)。

## 1.5 FunctionGraph 函数支持哪些编程语言？

FunctionGraph目前支持的编程语言，如[表1-1](#)所示。

表 1-1 支持的编程语言和版本

语言	支持版本
Python	2.7、3.6、3.9、3.10
Node.js	6.10、8.10、10.16、12.13、14.18、16.17、18.15
Java	8、11
Go	1.x
C#.NET Core	2.1、3.1
PHP	7.3

## 1.6 FunctionGraph 函数分配磁盘空间有多少？

对于每个FunctionGraph函数分配了512MB临时存储空间，单个租户下最大允许部署包大小为10G，更多函数的资源限制，请参考[使用限制](#)。

## 1.7 FunctionGraph 函数是否支持版本控制？

FunctionGraph函数支持版本控制，具体请参见[版本管理](#)。

## 1.8 函数中如何读写文件？

### 函数工作目录权限说明

函数可以读取代码目录下的文件，函数工作目录在入口文件的上一级，例如用户上传了文件夹backend，需要读取与入口文件同级目录的文件test.conf，可以用相对路径“code/backend/test.conf”，或者使用全路径（相关目录为RUNTIME\_CODE\_ROOT环境变量对应的值）。如果需要写文件（如创建新文件或者下载文件等），可以在/tmp目录下进行或者使用函数提供的挂载文件系统功能。

#### 说明

- 若容器回收，文件的读写就会失效。
- 函数目前不支持持久化。

### 典型场景

- 需要对OBS上的文件进行处理，可以先把文件下载到/tmp目录。
- 函数运行过程中产生了一些数据想保存到OBS，可以先在/tmp目录下创建新文件，然后把这些数据写到里面，接下来上传该文件到OBS。

## 1.9 使用 CLI 工具，如何配置网络代理？

### 问题描述

使用CLI工具，在上传程序zip包时，是否可以设置代理服务器和身份认证信息，通过内部网络的代理网关认证。

### 问题解决

设置网络代理命令如下：

```
export HTTP_PROXY="http://user:password@proxyp: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/)。

## 1.10 FunctionGraph 函数是否支持扩展？

FunctionGraph目前已经集成了一些非标准库如：redis、http、obs\_client等，开发函数时可以直接使用，详情请参考[开发指南](#)。

用户可以通过维护属于自己的依赖代码库，供所有函数使用，请参考[依赖包管理](#)。

## 1.11 IAM 子账号使用 FunctionGraph 需要设置哪些权限？

使用IAM子账号登录、使用函数工作流服务，对函数和函数下的触发器进行增删改查，如果出现权限不足情况，需要主账号对IAM子账号所属的用户组设置相应的权限。同时，若您想通过函数调用其他云服务，如OBS服务时，需要您配置委托并根据实际业务需求配置服务相关权限。为保障服务的安全，请您按照最小权限原则配置权限，详情请参考[权限说明](#)。

## 1.12 如何制作基于 ODBC 驱动的 Python 依赖包用于查询数据库？

对于依赖操作系统的包（以unixODBC为例），需要下载源码编译制作依赖包：

1. 通过ecs控制台页面登录ecs机器（确保gcc、make工具安装完成），执行如下命令下载相关源码包。

```
wget 源码路径
```

若下载包为zip文件，执行如下命令进行解压：

```
unzip xxx/xx.zip
```

若下载包为tar.gz文件，执行如下命令进行解压：

```
tar -zxvf xxx/xx.tar.gz
```

2. 执行如下命令，创建/opt/function/code目录。

```
mkdir /opt/function/code
```

3. 进入解压目录执行如下命令

```
./configure --prefix=/opt/function/code --sysconfdir=/opt/function/code;make;make install
```

4. 进入/opt/function/code/lib/pkgconfig检查配置，确认prefix目录是/opt/function/code。  

```
cd /opt/function/code/lib/pkgconfig
```
5. 将/opt/function/code/lib中的文件都拷到/opt/function/code。  

```
cp -r /opt/function/code/lib/* /opt/function/code
```
6. 切换到/opt/function/code目录下，将/opt/function/code下的文件都打入xxx.zip，依赖包制作完成。  

```
cd /opt/function/code  
zip -r xxx.zip *
```

## 1.13 FunctionGraph 配额

FunctionGraph服务的账户资源配额请参考[账户资源限制](#)，如果您需要扩大配额，请参考[如何申请扩大配额](#)。

## 1.14 FunctionGraph 函数支持哪些中文字体？

FunctionGraph函数支持以下四种中文字体：

- NotoSansTC-Regular.otf
- NotoSerifTC-Regular.otf
- NotoSansSC-Regular.otf
- NotoSerifSC-Regular.otf

## 1.15 FunctionGraph 函数如何解析 DNS 内网域名？

当前FunctionGraph函数无法直接解析华为云解析服务（DNS）的内网域名，当需要在函数中解析DNS域名，可参考本章节操作，通过调用DNS服务的接口，实现解析功能。

### 解析 DNS 内网域名

您需要提前创建VPC和DNS内网域名，再按照如下步骤操作。

#### 步骤1 内网域名关联VPC并添加记录集

登录云解析服务控制台，将内网域名关联VPC。

点击进入域名并添加记录集，类型选择A。

#### 步骤2 创建函数

创建一个运行时语言为Python 2.7版本的函数，代码示例如下：

```
# -*- coding:utf-8 -*-  
import json  
import os  
def handler(event, context):  
    os.system("curl -iv www.test.com")
```

#### 步骤3 为函数配置DNS与VPC委托

在IAM页面，为函数 workflow 服务配置“DNS ReadOnlyAccess”以及“VPC Administrator”权限的委托。

### 📖 说明

在进行解析域名时，函数需要查看云解析服务资源，所以必须要配置DNS资源数据读取权限，否则在执行函数时，会提示如下错误，获取不成功。

```
2020/08/20 10:37:12 GMT+08:00 Start invoke request 'a2f105b4-2e72-4fda-94a5-86d3837e961d', version: latest  
[GET /v2/zones/{zone_id}/recordsets] failed, response: {"code":"DNS.1802","message":"Policy doesn't allow dns:recordset:list to be performed."}  
2020/08/20 10:37:13 GMT+08:00 Finish invoke request 'a2f105b4-2e72-4fda-94a5-86d3837e961d', duration: 1030.072ms, billing duration: 1100ms, memory used: 77.039MB.
```

### 步骤4 配置函数

进入**步骤2**创建的函数详情页面，在“设置”页签下，执行以下配置。

- “权限”选择**步骤3**创建的委托。
- 打开VPC访问开关，配置上述操作中创建的VPC、子网、域名，详情参考**图1-1**。

图 1-1 配置函数



### 步骤5 验证结果

保存后执行函数时，就能在代码中解析该域名中配置的所有IPV4类型的域名。

----结束

### 📖 说明

用户配置了VPC域名解析后，如果在DNS云解析服务中修改域名对应的IP，10min后稳定生效。

## 1.16 容器镜像函数如何解析 DNS 内网域名?

当前FunctionGraph容器镜像函数无法直接解析华为云解析服务（DNS）的内网域名，当需要在函数中解析DNS域名，可参考本章节操作，通过调用DNS服务的接口，实现解析功能。

## 解析 DNS 内网域名

1. 已获取内网域名和域名ID。

以添加解析记录的域名为例，获取方法如下：

- a. 登录云解析服务控制台。
- b. 获取域名ID。



单击“”，在搜索框中勾选“域名”，获取域名ID。

- c. 获取对应解析记录的域名。

单击域名进入记录集列表，选择指定记录集。

2. 编写解析逻辑。

调试[查询单个Zone下Record Set列表](#)接口。

- 参数zone\_id即上述步骤中获取的“域名ID”，单击“调试”，响应体中即可获得内网域名对应的IP。
- 切换到代码示例获取完整的代码，相关依赖请参见SDK信息。

## 1.17 如何通过域名访问专享版 APIG 中注册的接口？

以域名www.test.com为例，具体请参考如下步骤。

**步骤1** 登录API网关控制台，在左侧导航栏选择“专享版”，单击实例名称，进入“实例概览”页面，在“入口地址”区域查看“弹性IP地址”，获取APIG的访问地址（ip格式）。

**步骤2** 在DNS控制台，配置用户域名www.test.com解析到apig地址的ipv4规则。

**步骤3** 最后在函数服务配置该域名的解析配置（可参考[FunctionGraph函数如何解析DNS内网域名](#)），这样就能在函数中通过域名(www.test.com)访问专享版APIG中注册的接口了。

----结束

## 1.18 函数工作流的常见使用场景？

1. Web类应用：比如小程序、网页/App、聊天机器人、BFF等。
2. 事件驱动类应用：文件处理、图片处理、视频直播/转码、实时数据流处理、IoT规则/事件处理等。
3. AI类应用：三方服务集成、AI推理、车牌识别。

具体请参考[应用场景](#)。

## 1.19 函数调用绑定在 APIG 的域名的服务，报域名无法解析？

函数服务目前只能解析pod域的域名或者在华为dns服务购买的域名。

## 1.20 同步函数工作流能否支持到内网最大带宽的同步传输？

暂不支持。

## 1.21 单租户的 VPC 超过默认配额时，需要怎么做？

单租户的VPC默认配额为4，超过默认配额时，请在[工单系统](#)，提交工单申请配额。

## 1.22 如何打印 info、error、warn 级别的日志？

此处以java语言为例，提供[日志级别打印demo](#)进行日志打印。

## 1.23 函数是否可以把 API 的接口域名配置成自己的域名？

可以，您可以参考如下步骤进行操作。

**步骤1** 您可以登录API网关控制台，参考[绑定域名](#)，完成域名绑定。

**步骤2** 在已创建的API分组的“域名管理”下，单击“绑定独立域名”。比如将xxxx.apig.x配置为test.com/user/get。

----结束

## 1.24 函数工作流是否支持修改运行时语言？

不支持，函数一旦创建完成，就不能修改运行时语言。

## 1.25 已创建的函数是否支持修改函数名称？

不支持，函数一旦创建完成，就不能修改函数名称。

## 1.26 挂载文件系统时，报“failed to mount exist system path”，应如何处理？

您可以将文件重新挂载在新的路径下。

用户/用户组ID: 可设置为1000之外的其他数字id，如果为-1默认设置为1003。用户/用户组ID用于对访问远端文件系统的目录权限进行控制。

文件系统/云服务器名称: 选择创建的文件系统或者云服务器资源，注意函数配置的VPC和委托要有访问权限。

共享目录路径: 如果选择ECS挂载需要配置远端共享目录，请参见[ECS创建nfs共享目录](#)。

函数访问路径: 为本地文件系统挂载目录，不能是系统已存在目录。建议使用/mnt/下二级子目录，例如/mnt/test。

## 1.27 如何获取上传的文件？

以Python语言为例，如果用户用`os.getcwd()`查看当前目录的话，会发现当前目录是`/opt/function`，但实际代码是传到`/opt/function/code`里的。

有2种方法可以获取到上传的文件：

1. 函数里使用`cd`命令切换路径到`/opt/function/code`
2. 使用全路径（相关目录为`RUNTIME_CODE_ROOT`环境变量对应的值）

### 📖 说明

其他语言同理，可参考如上方法获取上传的文件。

## 1.28 同步调用响应未收到的可能原因？

如果函数执行端到端时延超过90s，建议使用异步不使用同步，否则会因为网关限制，超过90s后无法收到同步响应。

## 1.29 `os.system("command &")`执行日志未采集，应如何处理？

不建议使用`os.system("command &")`后台运行命令，其产生的输出函数不进行采集。如果要得到后台运行命令的输出，建议使用`subprocess.Popen`的方式获取其输出。

## 1.30 自定义运行时，都能操作哪些目录？

目前默认只能操作`/tmp`目录，在`/tmp`下可以写文件（如创建新文件或者下载文件等）。

## 1.31 运行时语言支持的 python3.6 和 3.9 具体指哪个版本？

3.6.8、3.9.2。

## 1.32 用户想使用 vpc 功能，但不想配置 VPC Administrator 委托，应配置哪些授权项？

用户若不想配置VPC Administrator委托，可授予最小权限，如[表1 授权项配置](#)所示。

表 1-2 授权项配置

权限	授权项
删除端口	<code>vpc:ports:delete</code>
查询端口	<code>vpc:ports:get</code>

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

### 1.33 函数执行超时的可能原因有哪些？

- 自身代码执行逻辑超时，建议优化代码或增加超时时间。
- 网络请求超时，建议增加超时时间。
- 函数进行冷启动时，Java加载类时间过长，建议增加超时时间或增加内存。

### 1.34 如何获取函数代码？

1. 登录函数工作流控制台，单击函数名称进入函数详情页，单击右上方操作栏下的“导出函数”，继续单击“导出函数代码”。
2. 通过导出函数API接口获取函数代码。

### 1.35 是否有 initializer 的代码示例？

有，请参考如下示例。

- 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.getAccessKey()));
}
```
- PHP ([initializer入口介绍](#))

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

### 1.36 如何开启结构化日志查询

#### 使用场景

客户如果异步执行函数，需要查询请求状态，可以在异步配置->异步调用记录中查询异步调用记录：如图1-2。

图 1-2 异步调用记录



## 前提条件

需要开启异步状态持久化。

## 具体步骤

**步骤1** 联系函数客服配置白名单。

**步骤2** 开通云日志服务，在异步配置页面，单击“点击开通”，如图1-3。

图 1-3 开通云日志服务



**步骤3** 配置异步调用持久化，在异步配置页面单击“配置异步调用”->“开启异步调用持久化”，如和图1-5图1-4。

图 1-4 异步策略配置



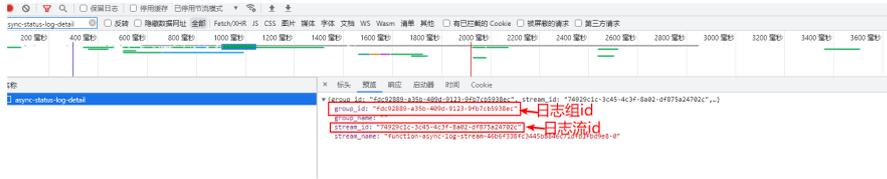
图 1-5 开启异步调用持久化



**步骤4** 在lts页面配置结构化查询。

1. 在函数页面查看函数配置的日志组和日志流。按F12->Network, 过滤“async-status-log-detail”, 获取日志组id和日志流id, 如图1-6。

图 1-6 获取日志组 ID 和日志流 ID



2. 在lts页面根据日志组和日志流的ID来进入日志流, 如图1-7。

图 1-7 进入日志流

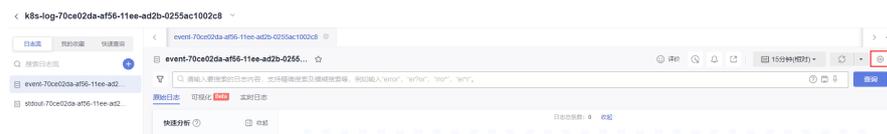


图 1-8 进入日志流



3. 在日志流页面添加结构化配置, 单击右上角的齿轮进行配置, 如图1-9。

图 1-9 日志流页面添加结构化配置



4. 设置结构化配置, 如图1-10。

图 1-10 结构化配置



5. 单击智能提取生成字段，如图1-11。

图 1-11 智能提取字段



6. 修改字段定义，单击  进行修改，其中各字段介绍如下：

- a. field1修改为function\_urn，类型为string；
- b. field2修改为request\_id，类型为string；
- c. field3修改为seq\_status,类型为long；
- d. field4修改为operation\_timestamp,类型为long；
- e. field5修改为error\_code,类型为long；
- f. field6修改为error\_message，类型为string；

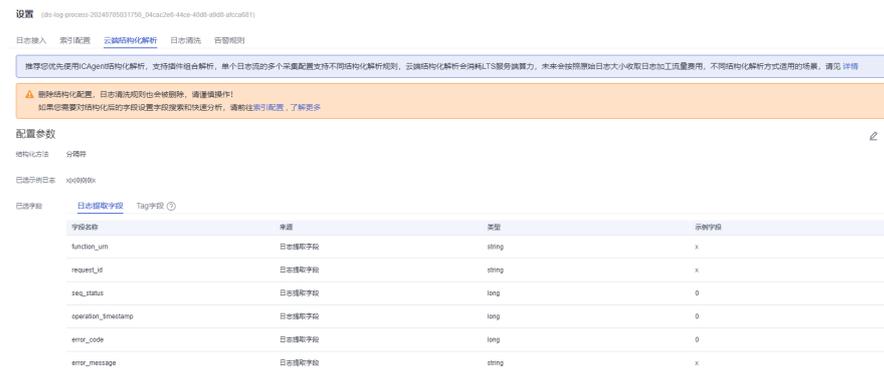
同时需要打开快速分析按钮，如图1-12。

图 1-12 打开快速分析按钮



7. 修改完成单击保存，最终界面如图1-13。

图 1-13 单击保存



----结束

## 1.37 函数服务是否支持在函数中启动 TCP 的监听端口，通过 EIP 接收外部发送过来的 TCP 请求？

目前函数暂不支持这种方式。函数的理念是无服务器计算，计算资源只会在运行期分配，这种自定义监听端口的场景并不适合。

## 1.38 FunctionGraph 是否支持域名解析？

- 1、函数服务不支持华为云解析服务上的自定义域名。
- 2、如果要配置华为云解析上注册域名的话，需要在自己的函数内部简单实现下域名转IP的操作。
- 3、调用DNS的API来解析域名对应的IP，然后函数通过调用IP访问对应的服务。

## 1.39 函数发起 HTTP 请求的源地址如何获取？

### 公网访问

- 函数未开启“函数访问VPC内资源”功能时访问公网。  
访问公网时使用函数 workflow 服务的SNAT地址，该地址是**固定的**，如何获取请咨询技术支持工程师。

图 1-14 未开启函数访问 VPC 内资源



- 函数开启“函数访问VPC内资源”功能时访问公网。（相关VPC配置详情请参见[配置网络](#)）

图 1-15 开启函数访问 VPC 内资源



访问公网时使用用户VPC中配置的SNAT地址，该地址是**固定的**，查看公网IP方法如下：



# 2 创建函数

---

## 2.1 能否在函数代码中使用线程和进程？

用户可使用编程语言和操作系统的功能，在函数中创建额外的线程和进程。

## 2.2 FunctionGraph 函数工程打包有哪些规范（限制）？

函数除了支持在线编辑代码，还支持上传ZIP、JAR、引入OBS文件等方式上传代码。具体规范请参考[打包规范说明](#)、[ZIP工程包示例](#)。

### 打包规范说明

函数除了支持在线编辑代码，还支持上传ZIP、JAR、引入OBS文件等方式上传代码，函数工程的打包规范说明如[表2-1](#)所示。

表 2-1 函数工程打包规范

编程语言	JAR包	ZIP包	OBS文件
Node.js	不支持该方式	<ul style="list-style-type: none"> <li>假如函数工程文件保存在“~/Code/”文件夹下，在打包的时候务必进入Code文件夹下选中所有工程文件进行打包，这样做的目的是：入口函数是程序执行的入口，确保解压后，入口函数所在的文件位于根目录。</li> <li>如果函数工程引入了第三方依赖，可以将第三方依赖打成ZIP包，在函数代码界面设置外部依赖包；也可以将第三方依赖和函数工程文件一起打包。</li> </ul>	将工程打成ZIP包，上传到OBS存储桶。
PHP	不支持该方式	<ul style="list-style-type: none"> <li>假如函数工程文件保存在“~/Code/”文件夹下，在打包的时候务必进入Code文件夹下选中所有工程文件进行打包，这样做的目的是：入口函数是程序执行的入口，确保解压后，入口函数所在的文件位于根目录。</li> <li>如果函数工程引入了第三方依赖，可以将第三方依赖打成ZIP包，在函数代码界面设置外部依赖包；也可以将第三方依赖和函数工程文件一起打包。</li> </ul>	将工程打成ZIP包，上传到OBS存储桶。

编程语言	JAR包	ZIP包	OBS文件
Python 2.7	不支持该方式	<ul style="list-style-type: none"> <li>假如函数工程文件保存在“~/Code/”文件夹下，在打包的时候务必进入Code文件夹下选中所有工程文件进行打包，这样做的目的是：入口函数是程序执行的入口，确保解压后，入口函数所在的文件位于根目录。</li> <li>如果函数工程引入了第三方依赖，可以将第三方依赖打成ZIP包，在函数代码界面设置外部依赖包；也可以将第三方依赖和函数工程文件一起打包。</li> </ul>	将工程打成ZIP包，上传到OBS存储桶。
Python 3.6	不支持该方式	<ul style="list-style-type: none"> <li>假如函数工程文件保存在“~/Code/”文件夹下，在打包的时候务必进入Code文件夹下选中所有工程文件进行打包，这样做的目的是：入口函数是程序执行的入口，确保解压后，入口函数所在的文件位于根目录。</li> <li>如果函数工程引入了第三方依赖，可以将第三方依赖打成ZIP包，在函数代码界面设置外部依赖包；也可以将第三方依赖和函数工程文件一起打包。</li> </ul>	将工程打成ZIP包，上传到OBS存储桶。
Java 8	如果函数没有引用第三方件，可以直接将函数工程编译成Jar包。	如果函数引用第三方件，将函数工程编译成Jar包后，将所有依赖三方件和函数jar包打成ZIP包。	将工程打成ZIP包，上传到OBS存储桶。

编程语言	JAR包	ZIP包	OBS文件
Go 1.x	不支持该方式	必须在编译之后打zip包，编译后的二进制文件必须与执行函数入口保持一致，例如二进制名称为Handler，则执行入口为Handler。	将工程打成ZIP包，上传到OBS存储桶。
C#	不支持该方式	必须在编译之后打zip包，必须包含“工程名.deps.json”，“工程名.dll”，“工程名.runtimeconfig.json”，“工程名.pdb”和“HC.Serverless.Function.Common.dll”文件。	将工程打成ZIP包，直接上传到OBS存储桶。
定制运行时	不支持该方式	打zip包，必须包含“bootstrap”可执行引导文件。	将工程打成ZIP包，直接上传到OBS存储桶。
Cangjie	不支持该方式	必须在编译之后打zip包，编译后的二进制文件必须与执行函数入口保持一致，例如二进制名称为libuser_func_test_success.so，则执行入口为libuser_func_test_success.so。	将工程打成ZIP包，上传到OBS存储桶。

## ZIP 工程包示例

- Nods.js工程ZIP包目录示例

```

Example.zip          示例工程包
|--- lib             业务文件目录
|--- node_modules   npm三方件目录
|--- index.js        入口js文件（必选）
|--- package.json    npm项目管理文件
                    
```
- PHP工程ZIP包目录示例

```

Example.zip          示例工程包
|--- ext             扩展库目录
|--- pear            PHP扩展与应用仓库
|--- index.php       入口PHP文件
                    
```
- Python工程ZIP包目录示例

```

Example.zip          示例工程包
|--- com             业务文件目录
|--- PLI             第三方依赖PLI目录
|--- index.py        入口py文件（必选）
|--- watermark.py    实现打水印功能的py文件
|--- watermark.png   水印图片
                    
```

- Java工程ZIP包目录示例

Example.zip	示例工程包
--- obstest.jar	业务功能JAR包
--- esdk-obs-java-3.20.2.jar	第三方依赖JAR包
--- jackson-core-2.10.0.jar	第三方依赖JAR包
--- jackson-databind-2.10.0.jar	第三方依赖JAR包
--- log4j-api-2.12.0.jar	第三方依赖JAR包
--- log4j-core-2.12.0.jar	第三方依赖JAR包
--- okhttp-3.14.2.jar	第三方依赖JAR包
--- okio-1.17.2.jar	第三方依赖JAR包
- Go工程ZIP包目录示例

Example.zip	示例工程包
--- testplugin.so	业务功能包
- C#工程ZIP包目录示例

Example.zip	示例工程包
--- fssExampleCsharp2.0.deps.json	工程编译产生文件
--- fssExampleCsharp2.0.dll	工程编译产生文件
--- fssExampleCsharp2.0.pdb	工程编译产生文件
--- fssExampleCsharp2.0.runtimeconfig.json	工程编译产生文件
--- Handler	帮助文件, 可直接使用
--- HC.Serverless.Function.Common.dll	函数工作流提供的dll
- Cangjie工程ZIP包目录示例

fss_example_cangjie.zip	示例工程包
--- libuser_func_test_success.so	业务功能包
- 定制运行时

Example.zip	示例工程包
--- bootstrap	可执行引导文件

## 2.3 FunctionGraph 如何隔离代码？

每个FunctionGraph函数都运行在其自己的环境中，有其自己的资源和文件系统。

## 2.4 HTTP 函数 bootstrap 启动文件如何创建？

如果您需要创建HTTP函数，需要用到bootstrap启动文件，具体创建方法请参考[bootstrap文件创建](#)。

# 3 触发器管理

## 3.1 哪些事件可以触发 FunctionGraph 函数？

有关事件源的完整列表，请参考[支持的事件源](#)。

## 3.2 使用 APIG 触发器调用一个返回 String 的 FunctionGraph 函数，报 500 错误，该如何解决？

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body(String)、statusCode(int)、headers(Map)和isBase64Encoded(boolean)，才可以正确返回。

Node.js函数APIG触发器调用返回结果定义示例如下：

```
exports.handler = function (event, context, callback) {
  const response = {
    'statusCode': 200,
    'isBase64Encoded': false,
    'headers': {
      "Content-type": "application/json"
    },
    'body': 'Hello, FunctionGraph with APIG',
  }
  callback(null, response);
}
```

Java函数APIG触发器调用返回结果定义示例如下：

```
import java.util.Map;

public HttpTriggerResponse index(String event, Context context){
  String body = "<html><title>FunctionStage</title>"
    + "<h1>This is a simple APIG trigger test</h1><br>"
    + "<h2>This is a simple APIG trigger test</h2><br>"
    + "<h3>This is a simple APIG trigger test</h3>"
    + "</html>";
  int code = 200;
  boolean isBase64 = false;
  Map<String, String> headers = new HashMap<String, String>();
  headers.put("Content-Type", "text/html; charset=utf-8");
}
```



- queryStringParameters: GET请求中URL后面要带的参数, 当发起一次GET请求时, 参数会以url string的形式进行传递。即?后的字符串则为其请求参数, 并以&作为分隔符。
- path: API访问地址。

您可以直接通过请求路径调用: <https://464d86ec641d45a683c5919ac57f3823.apig.projectID.huaweicloudapis.com/apig-demo/subpath>

也可以通过添加请求参数调用:

<https://464d86ec641d45a683c5919ac57f3823.apig.projectID.huaweicloudapis.com/apig-demo/subpath?a=1&b=2>



### 3.6 FunctionGraph 和 kafka 必须在同一个子网内, 才可以在 FunctionGraph 中配置 Kafka 触发器吗?

是必须的。在创建函数的时候会判断子网是否相同, 不允许一个VPC的不同子网进行连接。

# 4 依赖包管理

## 4.1 依赖包是什么？

依赖包是程序包，可以理解为某软件包运行所需的依赖环境，软件包和依赖环境相互依赖，若无此依赖环境，则软件包无法正常使用。

## 4.2 什么场景下需要引入依赖？

当您安装了某个程序或开发了某段代码，需要依赖某个环境运行时，此时您需要引入依赖。

## 4.3 使用依赖包时，有哪些注意事项？

- 依赖包内文件名不能以~结尾。
- 依赖包当前文件限制数为30000。
- 在函数页面上传依赖包的ZIP包文件大小限制为10M，如超过10M，需通过OBS上传（依赖包大小限制为最大300M）。
- 如果函数配置了私有依赖包且依赖包很大，建议在函数详情页的“设置 > 常规设置”重新设置函数执行时间，在原基础上增加超时时间。

## 4.4 函数支持引入的依赖库有哪些？

### 支持的依赖库说明

FunctionGraph支持引入标准库及第三方依赖库。

- 标准库  
对于标准库，无论是在线编辑或是线下开发打包上传至FunctionGraph，均可以直接在代码中引入，使用其功能。
- FunctionGraph支持的非标准库  
FunctionGraph内置一些三方件，如[表4-1](#)、[表4-2](#)所示。像标准库一样，在编写代码时直接引入，使用其功能。

表 4-1 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

表 4-2 Python Runtime 支持的非标准库

模块	功能	版本号
dateutil	日期/时间处理	2.6.0
requests	http库	2.7.0
httplib2	httpclient	0.10.3
numpy	数学计算	1.13.1
redis	redis客户端	2.10.5
obsclient	OBS客户端	-
smnsdk	访问公有云smn服务	1.0.1

- 其他第三方库（除了上面表格列举的非标准三方库，FunctionGraph没有内置别的非标准三方库）

将依赖的第三方库打包，上传至OBS桶或在函数界面上传，具体请参见[如何创建依赖包](#)，在函数代码中即可使用其功能。

## 4.5 使用 FunctionGraph 开发函数能否引入类库？

使用FunctionGraph编写函数支持引入标准库及非标准三方库，请参见[函数支持引入的依赖库有哪些？](#)

## 4.6 如何在 FunctionGrap 上使用第三方依赖？

1. 请参见[如何制作函数依赖包](#)，将依赖的第三方库打包成zip包。
2. 在函数平台，请参见[如何在函数平台创建依赖包](#)，完成依赖包创建。
3. 进入待配置依赖包的函数配置详情页，在“代码”页签下，请参见[如何为函数添加依赖包](#)，添加制作成功的私有依赖包。在函数代码中即可使用其功能。

## 4.7 如何制作函数依赖包？

制作函数依赖包推荐在EulerOS环境中进行。使用其他系统打包可能会因为底层依赖库的原因，运行出问题，比如找不到动态链接库。

### 📖 说明

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。

### 搭建 EulerOS 环境

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](#)。

1. 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。
2. 下载EulerOS镜像，在本地使用虚拟化软件搭建EulerOS系统的虚拟机。

### 为 Python 函数制作依赖包

打包环境中的Python版本要和对应函数的运行时版本相同，如Python2.7建议使用2.7.12及以上版本，Python3.6建议使用3.6.3以上版本。

为Python 2.7安装PyMySQL依赖包，并指定此依赖包的安装路径为本地的/tmp/pymysql下，可以执行如下命令。

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

执行成功后，执行以下命令。

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

进入子目录直到site-packages路径下（一般路径为usr/lib64/python2.7/site-packages/），接下来执行以下命令。

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

所生成的包即为最终需要的依赖包。

### 📖 说明

如果需要安装存放在本地的wheel安装包，直接输入：

```
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 Nodejs 函数制作依赖包

需要先保证环境中已经安装了对应版本的Nodejs。

为Nodejs 8.10安装MySQL依赖包，可以执行如下命令。

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

可以看到当前目录下会生成一个node\_modules文件夹。

- Linux系统

Linux系统下可以使用以下命令生成zip包。

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

即可生成最终需要的依赖包。

- windows系统

用压缩软件将node\_modules目录压缩成zip文件即可。

如果需要安装多个依赖包，也可以先新建一个package.json文件，例如在package.json中填入如下内容后，执行如下命令。

```
{
  "name": "test",
  "version": "1.0.0",
  "dependencies": {
    "redis": "~2.8.0",
    "mysql": "~2.17.1"
  }
}
npm install --save
```

### 📖 说明

不要使用CNPM命令制作nodejs依赖包。

然后将node\_modules打包成zip即可生成一个既包含MySQL也包含redis的依赖包。

Nodejs其他版本制作依赖包过程与上述相同。

## 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。开发Java函数中如何添加依赖包请参考[Java函数开发指南（使用IDEA工具普通Java项目）](#)。

## 为 PHP 函数制作依赖包

制作函数依赖包推荐在Huawei Cloud EulerOS 2.0环境中进行。

为php7.3通过composer安装protobuf3.19依赖包，默认环境中已经安装了composer和对应版本的php。

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

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

执行如下命令。

```
Composer install
```

可以看到当前目录底下生成一个vendor文件夹，文件夹中有autoload.php、composer和google三个文件夹。

- Linux系统

Linux系统下可以使用以下命令生成zip包。

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

- windows系统

用压缩软件将vendor目录压缩成zip文件即可。

如果要安装多个依赖包，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上传。

### 说明

php工程代码中使用通过composer下载的第三方依赖时，需要通过require "./vendor/autoload.php" 加载，平台默认把上传的zip包解压后的内容置于项目代码的同级目录下。

## 4.8 如何在函数平台创建依赖包？

1. 登录FunctionGraph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。
2. 单击的“创建依赖包”，弹出“创建依赖包”对话框。
3. 设置以下信息。

表 4-3 依赖包配置参数说明

参数	说明
依赖包名称	自定义的依赖包名称，用于识别不同的依赖包。
代码上传方式	分为上传ZIP文件和从OBS上传文件。 <ul style="list-style-type: none"> <li>• 上传ZIP文件：需单击“添加文件”，上传ZIP文件。</li> <li>• OBS链接URL：需填写“OBS链接URL”，OBS存储链接获取方法请参见<a href="#">OBS对象存储服务</a>。</li> </ul>
运行时语言	选择运行时语言。
描述	对于依赖包的描述信息，可以不填。

4. 单击“确定”，完成依赖包的创建。默认首次创建的依赖包版本为“1”。
5. 单击列表中的依赖包名称，进入版本历史界面，可以查看当前依赖包下的所有版本和版本相关信息。当前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。
  - 单击具体的版本号，可以查看版本地址。
  - 单击版本号所在行的删除，可以删除该版本。

图 4-1 删除依赖包版本



## 4.9 如何为函数添加依赖包？

1. 进入函数详情页面，在“代码”页签，单击“依赖代码包”所在行的“添加依赖包”。
  - 公共依赖包：此处的依赖包为函数平台提供，您可以直接添加使用。
  - 私有依赖包：此处的依赖包为用户自行制作上传的依赖包。
2. 完成后单击“保存”，完成依赖包的添加。

# 5 函数执行

## 5.1 FunctionGraph 函数的执行需要多长时间？

同步调用函数的执行时间在900秒内，异步调用函数的执行时间在72小时内。

FunctionGraph函数默认的执行超时时间为3秒，您可以自行设置执行超时时间为3 ~ 259200秒之间的任何整数。如果执行超时时间设置为3秒，超过3秒后，函数将终止执行。

## 5.2 FunctionGraph 函数的执行包含了哪些过程？

FunctionGraph函数的执行过程包含两步：

1. 选择一个相应内存的空闲实例。
2. 执行用户的指定运行代码。

## 5.3 FunctionGraph 函数的并发处理过程是什么？

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。

用户函数实例数=用户函数并发数/该函数的单实例并发数。

- 用户函数并发数：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”。

### 📖 说明

与FunctionGraph函数并发的其他相关服务，请参见[函数支持的事件源](#)。

## 5.4 FunctionGraph 函数如何处理长时间不执行的实例？

如果一个函数在一段时间内一直没有执行，那么所有与之相关的实例都会被释放。

## 5.5 首次访问函数慢，如何优化？

如果您使用的是C#或者Go语言，因为机制原因，启动速度会比其他语言慢。此时，您可以通过以下设置，增加运行速度。

- 适当增加函数的内存。
- 精简函数代码，例如：删除不必要的依赖包。
- 使用C#语言时，除了以上两种方法，在非并发场景下，您还可以通过以下方法增加运行速度。

创建一个一分钟一次的定时触发器，确保至少有一个存活的实例。

## 5.6 怎样获取在函数运行过程中实际使用了多少内存？

函数调用的返回信息中会包含最大内存消耗等信息，请参考开发指南中的[SDK接口](#)。也可以在执行结果界面查看。

## 5.7 为什么第一次请求会比较慢？

因为函数是冷启动的，所以如果有初始化或者函数中有第一次执行比较耗时的操作，第一次请求会比较慢，后面接着的请求就会很快，因为此时容器还没有销毁。如果间隔一分钟没有请求，容器就会销毁。

## 5.8 调用 API 时，报错怎么办？

您可参考[错误码参考](#)尝试解决，若无法解决，请联系技术支持工程师解决。

## 5.9 如何读取函数的请求头？

函数入口中的第一个参数里面包含请求头，您可以打印函数执行结果，从而获取想要的字段。

如下图，event为函数入口的第一个参数，headers为请求头。

```
index.py x
1  # -*- coding:utf-8 -*-
2  import json
3  def handler (event, context):
4      body = "<html><title>Functiongraph Demo</title><body><p>Hello, FunctionGraph!"
5      print(body)
6      return {
7          "statusCode":200,
8          "body":body,
9          "headers": {
10             "Content-Type": "text/html",
11         },
12         "isBase64Encoded": False
13     }
```

## 5.10 API 同步执行函数接口，是否支持内网调用？

默认使用内网调用，若跨区域，请开通公网调用。

## 5.11 为什么函数实际使用内存大于预估内存，甚至触发OOM？

1. 函数调用过程中，运行时解析和缓存传入的event事件，这部分操作会消耗额外的内存。
2. 函数调用结束后，回收的内存首先会放入内部内存池中，并不一定归还给操作系统，导致内存偏高，在高并发场景下这种现象会更加明显。

## 5.12 函数内存超限返回“runtime memory limit exceeded”，如何查看内存占用大小？

请在函数请求返回界面查看。

图 5-1 查看 oom 内存大小

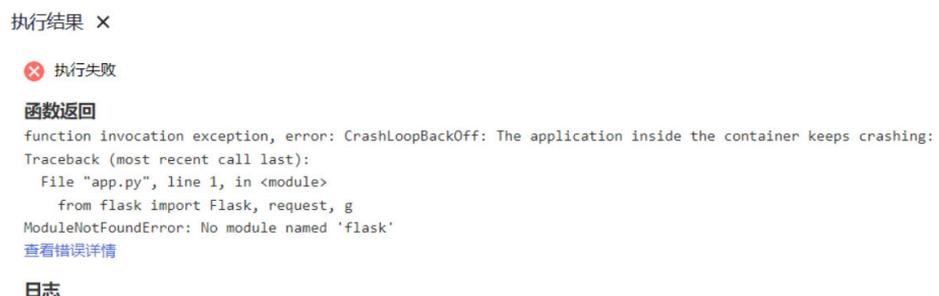


## 5.13 如何定位自定义镜像执行失败“CrashLoopBackOff”的原因？

若出现“CrashLoopBackOff: The application inside the container keeps crashing”错误字段：

1. 请根据页面提示信息诊断原因。

图 5-2 查看执行结果



2. 请参见[使用容器镜像部署函数](#)章节进行容器镜像自验证。
3. 排查镜像是否为x86 linux架构，目前仅支持x86 linux架构镜像。

## 5.14 用户使用相同的镜像名更新镜像，预留实例无法自动更新，会一直使用老镜像，应如何处理？

建议使用非latest的镜像tag管理镜像更新，避免使用完全相同的镜像名。

# 6 函数配置

## 6.1 FunctionGraph 函数是否支持环境变量？

创建函数时可以设置环境变量，无需对代码进行任何更改，可以设置动态参数，传递到函数代码和库，请参考[使用环境变量](#)。

## 6.2 能否在函数环境变量中存储敏感信息？

定义环境变量时，系统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。

## 6.3 如何通过父函数调用子函数？

请参考示例[通过父函数调用子函数](#)。

## 6.4 如何通过版本和别名实现绑定 APIG 触发器的 HTTP 函数的灰度升级？

1. 基于latest创建版本V1、V2，创建别名alias绑定对应版本V1，权重100%，为alias创建APIG触发器。

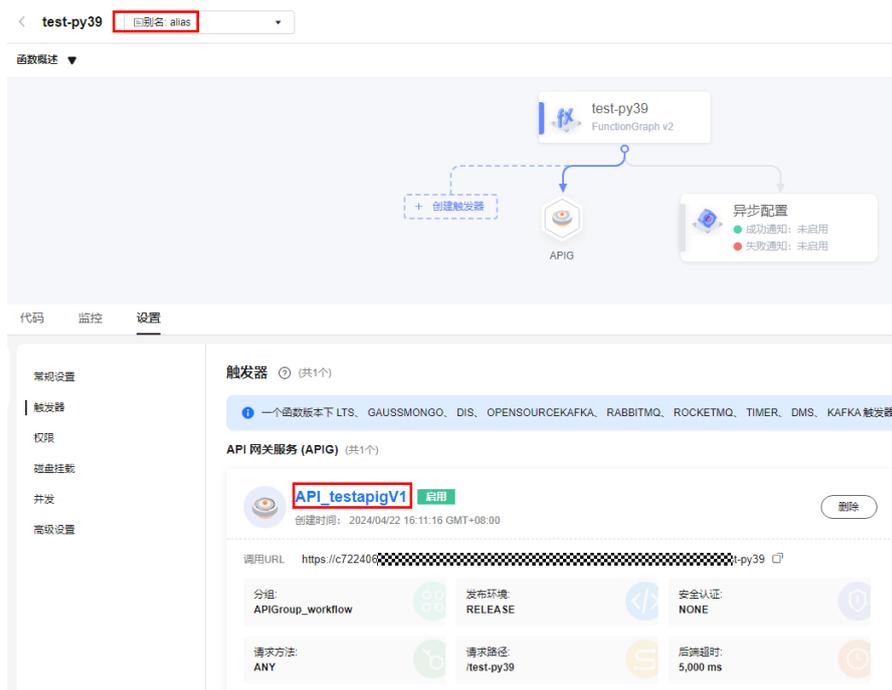
图 6-1 发布新版本 V1 和 V2



图 6-2 创建 alias 别名并绑定 V1 版本



图 6-3 在 alias 下创建触发器



### 说明

函数支持别名和版本分别创建APIG触发器，APIG触发器名称默认为API\_{函数名}{版本名}，请求Path默认是/{函数名}-{版本名}。

同一个APIG触发器(URL Path)不能同时存在于别名和它的对应版本下。比如V1版本和alias别名存在绑定关系，在V1创建触发器，默认名称为API\_testapigV1，再在alias别名下创建触发器，默认名称也为API\_testapigV1，且调用URL相同。此时返回V1版本下的触发器列表查看，发现已无API\_testapigV1触发器。

2. 回到latest版本别名界面，编辑alias，灰色版本选择V2，调整权重，实现V1到V2的灰度升级。

### 说明

权重用于分派接收到的流量给后端版本，具体数值视业务情况自定义。

图 6-4 编辑别名

编辑别名

别名名称 alias

\* 对应版本 V1 (含别名数: 1)

开启灰度版本

\* 灰度版本 V2 (含别名数: 0)

灰度类型

您可以按照流量比例随机灰度，您也可以根据请求中的 Header 进行灰度。

\* 权重  %

切流的比例。假设值为 5%，函数计算会将 5% 的流量打到灰度版本，95% 的流量打到主版本。

描述

0/512

3. 创建V3版本，编辑alias，对应版本设置为V2，灰度版本选择V3，调整权重，实现V2到V3的灰度升级。

图 6-5 创建 V3 版本

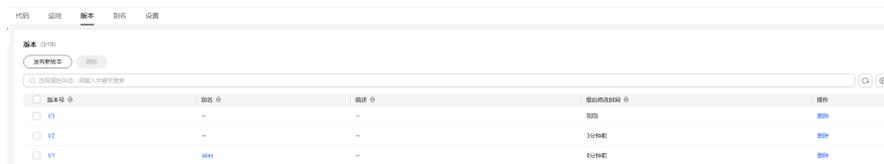


图 6-6 编辑别名

编辑别名

别名名称	alias
* 对应版本	V2 (含别名数: 0)
开启灰度版本	<input checked="" type="checkbox"/>
* 灰度版本	V3 (含别名数: 0)
灰度类型	<input checked="" type="button" value="按百分比随机灰度"/> <input type="button" value="按指定规则灰度"/>
	您可以按照流量比例随机灰度，您也可以根据请求中的 Header 进行灰度。
* 权重	<input type="text" value="-"/> <input type="text" value="90"/> <input type="text" value="+"/> %
	切流的比例。假设值为 5%，函数计算会将 5% 的流量打到灰度版本，95% 的流量打到主版本。
描述	<input type="text" value="请输入描述"/>
	0/512

须知

不能基于latest重复发布相同版本，必须修改配置或代码等信息后发布新版本。  
不能删除绑定了触发器的别名。

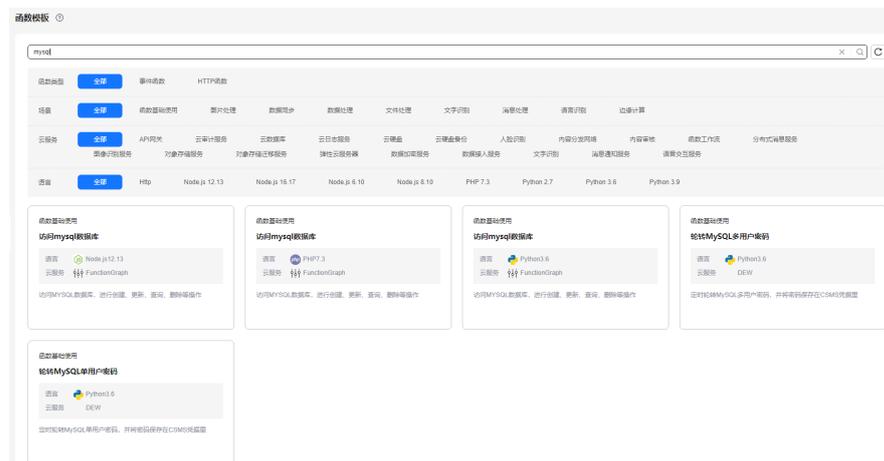
# 7 函数访问外部资源

## 7.1 函数如何访问 MySQL 数据库？

本章介绍如何访问MySQL数据库，具体操作步骤如下：

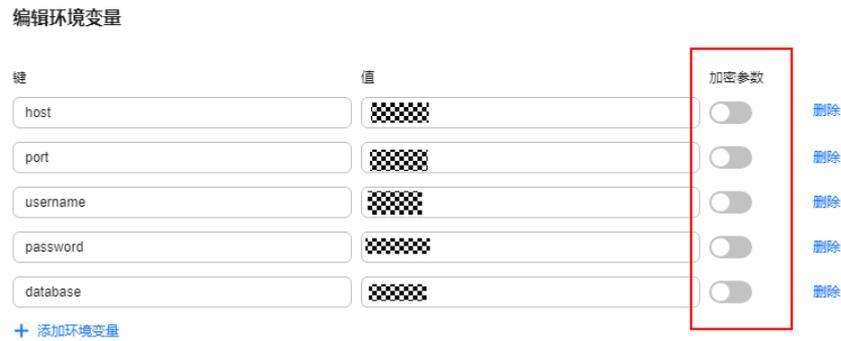
1. 确认MySQL数据库是否搭建在VPC的网络中？
  - 是，为函数设置与MySQL数据库相同的VPC、子网，具体请参考[函数配置VPC](#)。
  - 否，具体请参考[如何配置外网访问](#)。
2. 在函数模板中搜索mysql，根据使用开发语言选择对应MySQL数据库模板，如[图7-1](#)。模板参数根据使用情况按需配置，最后单击创建函数。

图 7-1 函数模板选择



3. MySQL函数模板创建好后，选择设置->环境变量，在环境变量列表中可按需开启加密参数功能，如[图7-2](#)，配置完保存。

图 7-2 开启加密参数



### 说明

如果函数需要访问RDS的接口，参考[创建委托](#)，获取RDS的授权。

## 7.2 函数如何访问 Redis?

本章介绍函数如何访问Redis，具体操作步骤如下：

1. 确认Redis实例是否搭建在VPC的网络中？
  - 是，为函数设置与Redis实例相同的VPC、子网，具体请参考[函数配置VPC](#)。
  - 否，Redis实例搭建在公网中，获取Redis实例的公网IP地址。
2. 在函数中，编写对接Redis实例的代码，示例如下。

FunctionGraph提供的Python 2.7和Python 3.6都内置了第三方库[redis-py](#)，因此不需要下载任何Redis第三方库。

```
# -*- coding:utf-8 -*-
import redis
def handler(event, context):
    r = redis.StrictRedis(host="host_ip",password="passwd",port=6379)
    print(str(r.get("hostname")))
    return "^_^"
```

### 说明

- 如果在FunctionGraph服务中远程访问公网上的Redis失败，请检查以下几个方面：
  - 查看redis.conf里面的设置，设置成允许任何IP访问。
  - 在redis.conf中设置Redis的访问密码。
  - 关闭防火墙。
- 如果函数需要访问DCS的接口，参考[创建委托](#)，获取DCS的授权。

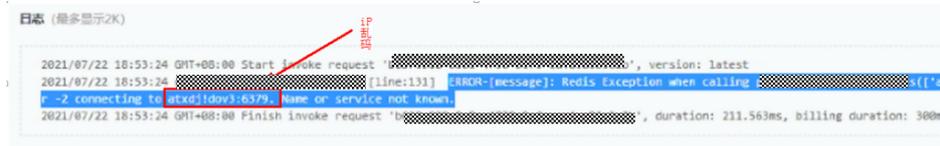
## 7.3 函数无法通过 VPC 连接对应的 Redis?

### 问题现象

相同的代码，函数能通过VPC1连通Redis1，但是不能通过VPC2连通Redis2。

## 解决方案

1. 与客户确认问题现象，Redis1在VPC1中，Redis2在VPC2中，Redis客户端代码是同一套代码。
2. 查看连接报错信息如下，通过VPC2连接Redis2时，Redis IP地址变成乱码。



```
日志 (最多显示2K)
2021/07/22 18:53:24 GMT+08:00 Start invoke request '...', version: latest
2021/07/22 18:53:24 [line:131] [ERROR-[message]: Redis Exception when calling '...'
-2 connecting to [redacted] name or service not known
2021/07/22 18:53:24 GMT+08:00 Finish invoke request '...', duration: 211.563ms, billing duration: 300ms
```

3. 分析Redis1与Redis2除了IP地址和密码不一样之外，其他没有什么不同，让客户比较两个Redis密码后发现，Redis2的密码中含有@符号，导致请求redis的时候IP地址截取有误变成乱码了。
4. 修改Redis2的密码后正常。

## 7.4 如何配置外网访问?

部署在VPC中的函数默认是和外网隔离开的，如果您想让函数同时具备内网访问和外网访问能力，您可以选择给VPC添加公网NAT网关。

### 前提条件:

1. 已创建虚拟私有云和子网，请参考[创建虚拟私有云基本信息及默认子网](#)。
2. 已申请弹性云公网IP，请参考[申请弹性公网IP](#)。

### 创建公网NAT网关步骤如下:

- 步骤1** 登录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，输入相关信息，选择已创建的虚拟私有云及子网（此处以vpc-01为例），在确认规格信息后提交，完成购买。具体操作步骤请参考[购买公网NAT网关](#)。
- 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

# 8 其他问题

## 8.1 如何查看给函数配置的告警规则？

请登录云监控控制台，查看“告警规则”。

## 8.2 视频转码，上传的 zip 文件是否能支持反编译？

函数不支持反编译，需要用户反编译后上传上来。

## 8.3 申请的 FunctionGraph2.0 公测资源，到期是否自动释放？如何计费？

functiongraph2.0到期自动结束释放，不会计费。

## 8.4 函数常规信息中的“应用”应如何理解？

“应用”实际作用就是文件夹功能。当前创建的新函数所属应用均为“default”应用，且无法更改，新版本里会逐步弱化并下线老界面的“应用”概念，未来会通过标签分组的方式来管理函数的分类等。

## 8.5 用户需要为冷启动时间付费吗？

不需要，冷启动时间不会被计量，因此用户无需为冷启动时间付费。

## 8.6 创建函数时，为什么提示“已限制，不能执行此操作”？

因为账号处于欠费状态。

## 8.7 函数计费中的调用次数，是某一账号下在不同 region 的所有函数的调用次数总和吗？

是的，每个 region 都有一套函数，如果是同一个账号在不同的 region 都有函数，则函数计费中的调用次数为调用次数总和。

# 9 V1 迁移 V2 相关问题

## 9.1 Python 语言从 V1 迁移到 V2 时需注意哪些兼容性问题？

### 1. args 的区别

V1 使用的语法：

```
args = parser.parse_args()
```

如果迁移到 V2，需修改为：

```
args = parser.parse_args(args=[])
```

因为 V1 和 V2 python runtime 的 sys.argv 不一样。

V2 是 ['/home/snuser/runtime/python3.6/server.py', '127.0.0.1:31536', '/opt/function/code']，比 v1 多了后面 2 个参数。

### 2. asyncio 的区别

V1 使用的语法：

```
loop = asyncio.get_event_loop()
loop.run_until_complete(func(arg1, arg2))
loop.close()
```

如果迁移到 V2，需修改为：

```
loop_tmp = asyncio.new_event_loop()
asyncio.set_event_loop(loop_tmp)
loop = asyncio.get_event_loop()
loop.run_until_complete(func(arg1, arg2))
loop.close()
```

因为 asyncio.get\_event\_loop() 是从 OS 线程（主线程）获取当前事件循环，而 V2 的 Python runtime 不是在主线程运行用户函数，所以函数内 asyncio.get\_event\_loop() 会抛出 RuntimeError。

V2 使用 asyncio 需要新建并设置事件循环。

## 9.2 Python2.7 在执行 reload(sys) 后无法通过 print 打印日志应如何解决？

建议使用 context.getLogger() 打印日志：

```
log = context.getLogger()  
log.info("test")
```