

Guía del usuario de FunctionGraph

Edición 01
Fecha 2025-12-08



Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2025. Todos los derechos reservados.

Quedan terminantemente prohibidas la reproducción y/o la divulgación totales y/o parciales del presente documento de cualquier forma y/o por cualquier medio sin la previa autorización por escrito de Huawei Cloud Computing Technologies Co., Ltd.

Marcas registradas y permisos



El logotipo **HUAWEI** y otras marcas registradas de Huawei pertenecen a Huawei Technologies Co., Ltd. Todas las demás marcas registradas y los otros nombres comerciales mencionados en este documento son propiedad de sus respectivos titulares.

Aviso

Es posible que la totalidad o parte de los productos, las funcionalidades y/o los servicios que figuran en el presente documento no se encuentren dentro del alcance de un contrato vigente entre Huawei Cloud y el cliente. Las funcionalidades, los productos y los servicios adquiridos se limitan a los estipulados en el respectivo contrato. A menos que un contrato especifique lo contrario, ninguna de las afirmaciones, informaciones ni recomendaciones contenidas en el presente documento constituye garantía alguna, ni expresa ni implícita.

Huawei está permanentemente preocupada por la calidad de los contenidos de este documento; sin embargo, ninguna declaración, información ni recomendación aquí contenida constituye garantía alguna, ni expresa ni implícita. La información contenida en este documento se encuentra sujeta a cambios sin previo aviso.

Huawei Cloud Computing Technologies Co., Ltd.

Dirección: Huawei Cloud Data Center Jiaoxinggong Road
 Avenida Qianzhong
 Nuevo distrito de Gui'an
 Gui Zhou, 550029
 República Popular China

Sitio web: <https://www.huaweicloud.com/intl/es-us/>

Índice

1 Antes de comenzar.....	1
1.1 Uso de FunctionGraph.....	1
1.2 Gestión de permisos.....	4
1.2.1 Cómo crear un usuario y otorgar permisos.....	4
1.2.2 Cómo crear una política personalizada.....	5
1.3 Lenguajes de programación compatibles.....	7
1.3.1 Node.js.....	7
1.3.2 Python.....	7
1.3.3 Java.....	8
1.3.4 Go.....	8
1.3.5 C#.....	8
1.3.6 PHP.....	9
1.3.7 Tiempo de ejecución personalizado.....	9
2 Creación de una función de FunctionGraph.....	17
2.1 Creación de un paquete de implementación.....	17
2.2 Creación de una función desde cero.....	23
2.2.1 Creación de una función de evento.....	23
2.2.2 Creación de una función HTTP.....	27
2.3 Creación de una función mediante una plantilla.....	32
2.4 Implementación de una función mediante una imagen de contenedor.....	33
3 Configuración de la función.....	38
3.1 Configuración de la inicialización.....	38
3.2 Configuración de los parámetros básicos.....	39
3.3 Configuración de permisos de delegación.....	41
3.4 Configuración de VPC.....	46
3.5 Configuración del montaje en disco.....	47
3.6 Configuración de variables de entorno.....	52
3.7 Configuración de encriptación.....	54
3.8 Configuración de la notificación de ejecución asincrónica.....	54
3.9 Configuración de Single-Instance Multi-Concurrency.....	57
3.10 Gestión de versiones.....	60
3.11 Gestión de alias.....	61

3.12 Configuración de memoria dinámica.....	63
4 Creación de activadores.....	66
4.1 Gestión de activadores.....	66
4.2 Uso del activador de temporizador.....	66
4.3 Uso de un activador APIG (dedicado).....	68
4.4 Uso de un activador OBS.....	70
4.5 Uso de un activador Kafka.....	72
4.6 Uso de un activador DIS.....	74
4.7 Uso de un activador SMN.....	77
4.8 Uso de un activador LTS.....	78
4.9 Uso de un activador CTS.....	80
4.10 Uso de un activador DDS.....	83
4.11 Uso de un activador de GaussDB (for Mongo).....	85
4.12 Uso de un activador APIG (compartido).....	87
4.13 Uso de un activador de APIC.....	89
4.14 Uso de un activador DMS (for RabbitMQ).....	92
4.15 Apéndice: Expresiones de Cron para un activador de temporizador de función.....	94
5 Depuración de la función.....	98
5.1 Depuración en línea.....	98
5.2 Depuración local con VSCode.....	102
6 Invocación de la función.....	107
6.1 Invocación síncrona.....	107
6.2 Invocación asíncrona.....	107
6.3 Mecanismo de reintento.....	108
7 Monitoreo.....	109
7.1 Métricas.....	109
7.1.1 Monitoreo de funciones.....	109
7.1.2 Métricas de función.....	110
7.1.3 Creación de una regla de alarma.....	112
7.2 Registros.....	114
7.2.1 Consulta de registros de funciones.....	115
7.2.2 Gestión de registros de funciones.....	115
8 Gestión de funciones.....	121
9 Gestión de dependencias.....	129
10 Gestión de instancias reservadas.....	136
11 Aumento de la cuota de recursos.....	140
12 Auditoría.....	142
12.1 Operaciones registradas por CTS.....	142

1 Antes de comenzar

1.1 Uso de FunctionGraph

FunctionGraph permite ejecutar código sin aprovisionar ni gestionar servidores, mientras garantiza altos niveles de disponibilidad y escalabilidad. Todo lo que necesitas hacer es cargar tu código y establecer las condiciones de ejecución, y FunctionGraph se va a encargar del resto. Pague solo lo que utilice y olvídate de los cobros por el código que no se está ejecutando.

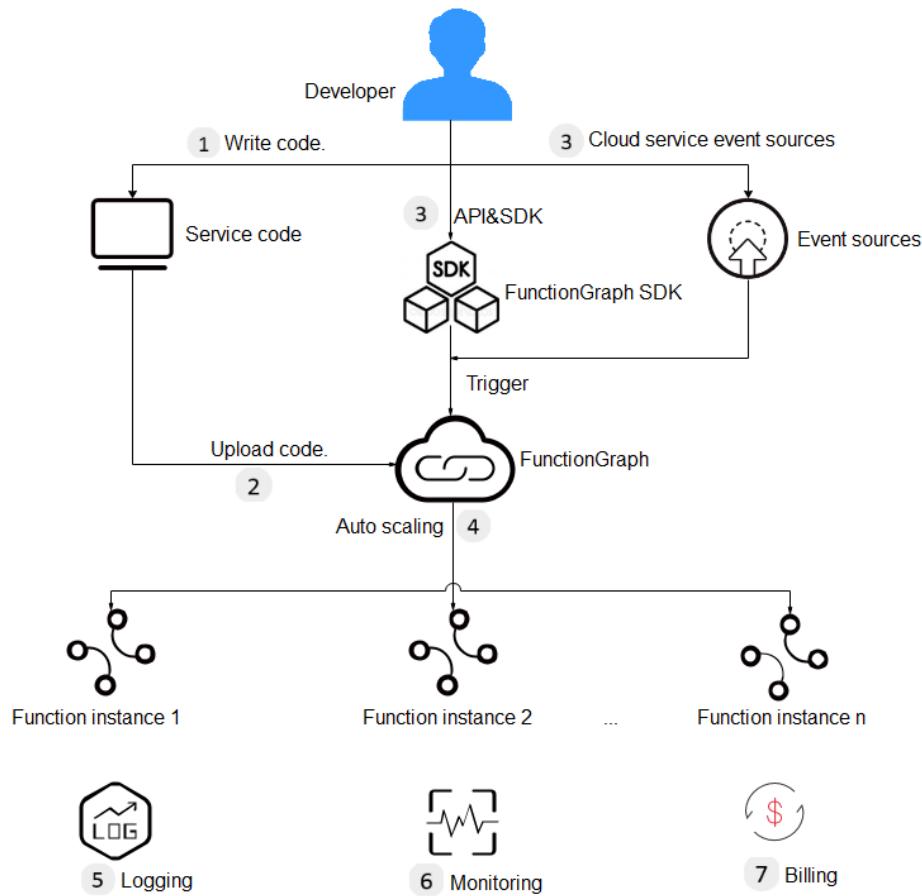
NOTA

FunctionGraph v2 está disponible en CN North-Beijing4, CN East-Shanghai1, AP-Bangkok, AP-Singapore, LA-Sao Paulo1, and LA-Santiago.

Proceso

Figura 1-1 muestra el proceso de uso de funciones.

1. Escribe código, empaquétalo y súbelo a FunctionGraph y agrega fuentes de eventos como notificaciones simples de mensajes (SMN), servicios de almacenamiento de objetos (OBS) y fuentes de eventos API Gateway (APIG) para crear aplicaciones.
2. Las funciones se activan por llamadas de la RESTful API o fuentes de eventos para lograr los fines de servicio esperados. Durante este proceso, FunctionGraph programa automáticamente los recursos.
3. Consulte registros y métricas. Tenga en cuenta que se le facturará en base a la duración de ejecución del código.

Figura 1-1 Diagrama de flujo

A continuación se muestran los detalles:

1. Escribir código.

Escribe código en Node.js, Python, Java, Go, C# o PHP. Para obtener más información, consulte la [Guía para desarrolladores de FunctionGraph](#).

2. Cargar código.

Edite el código en línea, cargue un archivo ZIP o JAR local, o cargue un archivo ZIP desde OBS. Para más detalles, consulte [Creación de un paquete de implementación](#).

3. Activar las funciones por llamadas de API o eventos de servicio en la nube.

Las funciones se activan por llamadas a la API o eventos de servicio en la nube. Para más detalles, consulte [Creación de activadores](#).

4. Aplicar el escalado automático.

FunctionGraph aplica el escalado automático en función del número de solicitudes. Para obtener más información, consulte [Cuotas y restricciones de uso](#).

5. Consultar registros.

Consulte los registros de ejecución de la función ya que FunctionGraph está interconectado con Log Tank Service (LTS). Para más detalles, consulte [Registros](#).

6. Consultar información de monitoreo.

Consulte la información de monitoreo gráfico ya que FunctionGraph está interconectado con Cloud Eye. Para más detalles, consulte [Métricas](#).

7. Modos de facturación

Después de ejecutar una función, se le facturará en función del número de solicitudes de ejecución de función y la duración de la ejecución. (v1: facturado por cada 100 ms; v2: facturado por cada 1 ms)

Introducción del Panel de Control

Inicie sesión en la consola de FunctionGraph y elija **Dashboard** en el panel de navegación de la izquierda.

- Consulte sus funciones creadas/la cuota de función, almacenamiento usado/la cuota de almacenamiento, las invocaciones mensuales y el uso de recursos.

Figura 1-2 Estadísticas mensuales



- Puede consultar la información de monitoreo a nivel de inquilino, incluido el número de invocaciones, el número de errores, la duración y el número de aceleradores, como se muestra en **Figura 1-3**.

Figura 1-3 Métricas

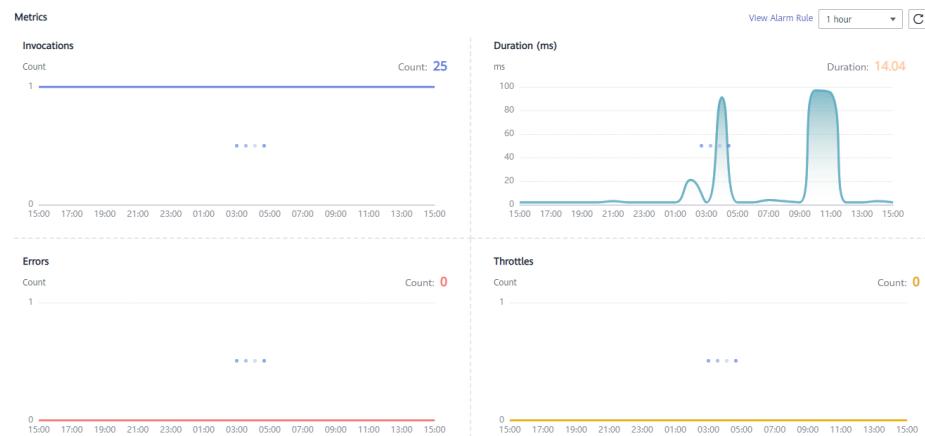


Tabla 1-1 describe las métricas de la función.

Tabla 1-1 Métricas de función

Métrica	Unidad	Descripción
Invocaciones	Cantidad	Número total de solicitudes de invocación, incluidos los errores de invocación y las invocaciones rechazadas. En caso de invocación asíncrona, el cuento comienza solo cuando se ejecuta una función en respuesta a una solicitud.

Métrica	Unidad	Descripción
Duración	milisegundo	Duración máxima: la duración máxima de todas las funciones se ejecutan a la vez dentro de un período. Duración mínima: la duración mínima de todas las funciones se ejecutan a la vez dentro de un período. Duración media: la duración media de todas las funciones se ejecutan a la vez dentro de un período.
Errores	Cantidad	Número de veces que sus funciones fallan al devolver el código de error 200 . También se incluyen los errores causados por la sintaxis de la función o la ejecución.
Rechazo	Cantidad	Número de veces que FunctionGraph rechaza sus funciones debido al límite de recursos.

1.2 Gestión de permisos

1.2.1 Cómo crear un usuario y otorgar permisos

En esta sección se describe cómo utilizar la [Identity and Access Management \(IAM\)](#) para implementar un control de permisos detallado para los recursos de FunctionGraph. Con IAM, usted puede:

- Crear los usuarios de IAM para empleados en base a la estructura organizativa de su empresa. Cada usuario de IAM tiene sus propias credenciales de seguridad para acceder a los recursos de FunctionGraph.
- Otorgar únicamente los permisos necesarios para que los usuarios realicen una tarea.
- Confiar en otras cuentas o servicios en la nube para realizar operaciones y mantenimientos profesionales y eficientes en sus recursos de FunctionGraph.

Si su cuenta no necesita usuarios individuales de IAM, puede omitir este capítulo.

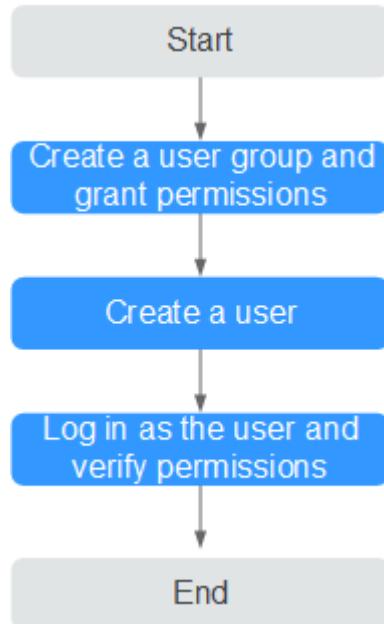
En esta sección se describe el procedimiento para conceder permisos. Para más detalles, consulte [Figura 1-4](#).

Prerrequisitos:

Antes de asignar permisos a grupos de usuarios, debe obtener información sobre los permisos del sistema enumerados en [Descripción de permisos](#). Para ver las políticas del sistema de otros servicios, consulte [Permisos del sistema](#).

Proceso

Figura 1-4 Proceso para conceder permisos de FunctionGraph



1. Creación de un grupo de usuarios y asignación de permisos

Cree un grupo de usuarios en la consola de IAM y asigne el rol **FunctionGraph Invoker** al grupo.

2. Creación de un usuario de IAM

Cree un usuario en la consola de IAM y agregue el usuario al grupo creado en 1.

3. Inicio de sesión como usuario de IAM y verificación de permisos

Inicie sesión en la consola de gestión como usuario creado y compruebe si este usuario solo tiene permisos de lectura para FunctionGraph:

- Seleccione **Service List > FunctionGraph** para acceder a la consola de FunctionGraph. En el panel de navegación, seleccione **Functions > Function List**. A continuación, haga clic en **Create Function**. Si aparece un mensaje que indica permisos insuficientes para realizar la operación, el rol FunctionGraph Invoker ya ha tenido efecto.
- Elija cualquier otro servicio en **Service List**. Si aparece un mensaje que indica permisos insuficientes para acceder al servicio, el rol FunctionGraph Invoker ya ha tenido efecto.

1.2.2 Cómo crear una política personalizada

Las políticas personalizadas se pueden crear como un suplemento a las políticas del sistema de FunctionGraph.

Puede crear políticas personalizadas de cualquiera de las siguientes maneras:

- Visual editor: Seleccione servicios en la nube, acciones, recursos y condiciones de solicitud. Esto no requiere conocimiento de la sintaxis de políticas.

- JSON: Edite las políticas JSON desde cero o basándose en una política existente.

Para obtener más información, consulte [Creación de una política personalizada](#). En esta sección se presentan los ejemplos de políticas personalizadas comunes de FunctionGraph.

Ejemplo de políticas personalizadas

- Ejemplo 1: Autorizar a un usuario a consultar el código de función y la configuración

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

- Ejemplo 2: Denegar la eliminación de funciones

Una política con solo los permisos "Denegar" debe usarse junto con otras políticas para que surtan efecto. Si ambos permisos "Permitir" y "Denegar" se asignan a un usuario, los permisos "Denegar" tienen prioridad sobre los permisos "Permitir".

Si necesita asignar permisos de la política **FunctionGraph FullAccess** a un usuario pero evitar que el usuario elimine funciones, cree una política personalizada para denegar la eliminación de funciones y adjunte ambas políticas al grupo al que pertenece el usuario. De esta manera, el usuario puede realizar todas las operaciones de FunctionGraph excepto la eliminación de funciones. A continuación se muestra un ejemplo de una política de denegación:

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

- Ejemplo 3: Configuración de permisos para recursos específicos

Puede conceder permisos de usuario de IAM para recursos específicos. Por ejemplo, para conceder permisos de usuario para la función **functionname** en la aplicación **Default**, establezca **functionname** en una ruta de recurso especificada que es:

FUNCTIONGRAPH:*.*:function:Default/functionname.

NOTA

Especificar recursos de función:

Formato: **FUNCTIONGRAPH:*.*:function: application or function name**

Para los recursos de función, IAM genera automáticamente el prefijo de ruta de recursos

FUNCTIONGRAPH:*.*:function: Puede especificar una ruta de recurso agregando el nombre

de la aplicación o función junto al prefijo de ruta de acceso. Se admiten comodines Wildcards (*).

Por ejemplo, **FUNCTIONGRAPH:*.*:function:Default/*** indica cualquier función en la aplicación **Default**.

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

```

        "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/*"
        ]
    }
]
}

```

1.3 Lenguajes de programación compatibles

1.3.1 Node.js

√: Soportado. ✗: No soportado.

Tiempo de ejecución	FunctionGraph V1	FunctionGraph V2	Guía del desarrollador
Node.js 6.10	√	√	Para obtener más información sobre la sintaxis de funciones, las SDK API y el desarrollo de funciones, consulte Desarrollo de funciones en Node.js .
Node.js 8.10	√	√	
Node.js 10.16	√	√	
Node.js 12.13	√	√	
Node.js 14.18	✗	√	

1.3.2 Python

√: Soportado. ✗: No soportado.

Tiempo de ejecución	FunctionGraph V1	FunctionGraph V2	Guía del desarrollador
Python 2.7	✓	✓	Para obtener más información sobre la sintaxis de funciones, las SDK API y el desarrollo de funciones, consulte Desarrollo de funciones en Python .
Python 3.6	✓	✓	
Python 3.9	✗	✓	

1.3.3 Java

✓: Soportado. ✗: No soportado.

Tiempo de ejecución	FunctionGraph V1	FunctionGraph V2	Guía del desarrollador
Java 8	✓	✓	Para obtener más información sobre la sintaxis de funciones, las SDK API y el desarrollo de funciones, consulte Desarrollo de funciones en Java .
Java 11	✗	✓	

1.3.4 Go

✓: Soportado. ✗: No soportado.

Tiempo de ejecución	FunctionGraph V1	FunctionGraph V2	Guía del desarrollador
Go 1.8	✓	✗	Para obtener más información sobre la sintaxis de funciones, las SDK API y el desarrollo de funciones, consulte Desarrollo de funciones en Go .
Go 1.x	✓	✓	

1.3.5 C#

✓: Soportado. ✗: No soportado.

Tiempo de ejecución	Function Graph V1	FunctionGraph V2	Guía del desarrollador
C# (.NET Core 2.0)	√	✗	Para obtener más información sobre la sintaxis de funciones, las SDK API y el desarrollo de funciones, consulte Desarrollo de funciones en C# .
C# (.NET Core 2.1)	√	✗	
C# (.NET Core 3.1)	√	✗	

1.3.6 PHP

√: Soportado. ✗: No soportado.

Tiempo de ejecución	Function Graph V1	FunctionGraph V2	Guía del desarrollador
PHP 7.3	√	√	Para obtener más información sobre la sintaxis de funciones, las SDK API y el desarrollo de funciones, consulte Desarrollo de funciones en PHP .

1.3.7 Tiempo de ejecución personalizado

Escenarios

Un tiempo de ejecución ejecuta el código de una función, lee el nombre del controlador de una variable de entorno y lee los eventos de invocación de las API de tiempo de ejecución de FunctionGraph. El tiempo de ejecución pasa los datos de evento al controlador de función y devuelve la respuesta del controlador a FunctionGraph.

FunctionGraph admite tiempos de ejecución personalizados. Puede utilizar un archivo ejecutable denominado bootstrap para incluir un tiempo de ejecución en el paquete de implementación de funciones. El tiempo de ejecución ejecuta el método manejador de la función cuando se invoca la función.

El tiempo de ejecución se ejecuta en el entorno de ejecución de FunctionGraph. Puede ser un script de shell o un archivo ejecutable binario que se compila en Linux.

NOTA

Después de la programación, simplemente empaqueta tus códigos en un archivo ZIP (Java, Node.js, Python y Go) o un archivo JAR (Java), y sube el archivo a FunctionGraph para su ejecución. Cuando cree un archivo ZIP, coloque el archivo de controlador bajo el directorio **root** para asegurarse de que su código puede ejecutarse normalmente después de ser descomprimido.

Si edita código en Go, comprima el archivo compilado y asegúrese de que el nombre del archivo de biblioteca dinámica sea coherente con el nombre del complemento del controlador. Por ejemplo, si el nombre del archivo de biblioteca dinámica es **testplugin.so**, establezca el nombre del controlador en **testplugin.Handler**.

tiempo de ejecución de archivo bootstrap

Si hay un archivo llamado bootstrap en su paquete de implementación de funciones, el FunctionGraph ejecuta ese archivo. Si el archivo bootstrap no se encuentra o no es ejecutable, su función devolverá un error cuando se invoque.

El código de tiempo de ejecución es responsable de completar las tareas de inicialización. Él procesa eventos de invocación en un bucle hasta que sea terminado.

Las tareas de inicialización se ejecutan una vez por cada instancia de la función para preparar el entorno para el manejo de invocaciones.

API de tiempo de ejecución

FunctionGraph proporciona API de tiempo de ejecución HTTP para recibir eventos de invocación de funciones y devuelve datos de respuesta en el entorno de ejecución.

● Siguiente Invocación

Método – Get

Ruta – [http://\\$RUNTIME_API_ADDR/v1/runtime/invocation/request](http://$RUNTIME_API_ADDR/v1/runtime/invocation/request)

Esta API se utiliza para recuperar un evento de invocación. El cuerpo de la respuesta contiene los datos del evento. En la siguiente tabla se describen datos adicionales sobre la invocación contenida en el encabezado de respuesta.

Tabla 1-2 Información del encabezado de respuesta

Parámetro	Descripción
X-Cff-Request-Id	ID de solicitud.
X-CFF-Access-Key	Llave de acceso de la cuenta. Se debe configurar una agencia para la función si se utiliza esta variable.
X-CFF-Auth-Token	Token de la cuenta. Se debe configurar una agencia para la función si se utiliza esta variable.
X-CFF-Invoke-Type	Tipo de invocación de la función.
X-CFF-Secret-Key	Llave secreta de la cuenta. Se debe configurar una agencia para la función si se utiliza esta variable.

Parámetro	Descripción
X-CFF-Security-Token	Token de seguridad de la cuenta. Se debe configurar una agencia para la función si se utiliza esta variable.

- **Respuesta de invocación**

Método – POST

Ruta – `http://$RUNTIME_API_ADDR/v1/runtime/invocation/response/$REQUEST_ID`

Esta API se utiliza para enviar una respuesta de invocación correcta a FunctionGraph. Después de que el tiempo de ejecución invoque al manejador de función, envía la respuesta de la función a la ruta de respuesta de invocación.

- **Error de invocación**

Método – POST

Ruta – `http://$RUNTIME_API_ADDR/v1/runtime/invocation/error/$REQUEST_ID`

\$REQUEST_ID es el valor de la variable **X-Cff-Request-Id** en el encabezado de una respuesta de recuperación de evento. Para obtener más información, consulte [Tabla 1-2](#).

\$RUNTIME_API_ADDR es una variable de entorno del sistema. Para obtener más información, consulte [Tabla 1-3](#).

Esta API se utiliza para enviar una respuesta de invocación de error a FunctionGraph. Después de que el tiempo de ejecución invoque al manejador de función, envía la respuesta de la función a la ruta de respuesta de invocación.

VARIABLES DE ENTORNO EN TIEMPO DE EJECUCIÓN

Puede usar variables de entorno personalizadas y en tiempo de ejecución en el código de función. En la siguiente tabla se enumeran las variables de entorno en tiempo de ejecución que se utilizan en el entorno de ejecución del FunctionGraph.

Tabla 1-3 Variables de entorno

Clave	Descripción
RUNTIME_PROJECT_ID	ID del proyecto
RUNTIME_FUNC_NAME	Nombre de la función
RUNTIME_FUNC_VERSION	Versión de la función
RUNTIME_PACKAGE	Aplicación a la que pertenece la función
RUNTIME_HANDLER	Manejador de funciones
RUNTIME_TIMEOUT	Duración del tiempo de espera de la función
RUNTIME_USERDATA	Valor pasado a través de una variable de entorno
RUNTIME_CPU	Número de núcleos de CPU asignados

Clave	Descripción
RUNTIME_MEMORY	Memoria asignada
RUNTIME_CODE_ROOT	Directorio que almacena el código de función
RUNTIME_API_ADDR	Dirección IP del host y puerto de una API de tiempo de ejecución personalizada

El valor de una variable de entorno personalizada se puede recuperar de la misma manera como el valor de una variable de entorno de FunctionGraph.

Ejemplo

Este ejemplo contiene un archivo llamado bootstrap. El archivo se implementa en Bash.

El tiempo de ejecución carga el script de función desde el paquete de implementación por el uso de dos variables.

El archivo bootstrap es como el siguiente:

```
#!/bin/sh

set -o pipefail

#Processing requests loop
while true
do
  HEADERS=$(mktemp)
    # Get an event
    EVENT_DATA=$(curl
    -sS -L "$HEADERS" -X GET
    "http://$RUNTIME_API_ADDR/v1/runtime/invocation/request")
      # Get request id
  from response header
    REQUEST_ID=$(grep
    -F x-cff-request-id "$HEADERS" | tr -d '[:space:]' | cut -d: -f2)
      if [ -z
    "$REQUEST_ID" ]; then
      continue
    fi
    # Process request
  data
  RESPONSE="Echoing request: '$EVENT_DATA'"
    # Put response
    curl -X POST
    "http://$RUNTIME_API_ADDR/v1/runtime/invocation/response/$REQUEST_ID"
    -d "$RESPONSE"
done
```

Después de cargar el script, el tiempo de ejecución procesa los eventos de invocación en un bucle hasta que sea terminado. Utiliza la API para recuperar eventos de invocación de FunctionGraph, pasa los eventos al controlador y luego envía las respuestas a FunctionGraph.

Para obtener el ID de solicitud, el tiempo de ejecución guarda el encabezado de respuesta de la API en un archivo temporal y luego lee el ID de solicitud del campo de encabezado **x-cff-request-id**. El tiempo de ejecución procesa los datos de eventos recuperados y envía una respuesta a FunctionGraph.

El siguiente es un ejemplo del código fuente en Go. Solo se puede ejecutar después de la compilación.

```
package main

import (
    "bytes"
    "encoding/json"
    "fmt"
    "io"
    "io/ioutil"
    "log"
    "net"
    "net/http"
    "os"
    "strings"
    "time"
)

var (
    getRequestUrl          = os.ExpandEnv("http://$"
{RUNTIME_API_ADDR}/v1/runtime/invocation/request")
    putResponseUrl         = os.ExpandEnv("http://$"
{RUNTIME_API_ADDR}/v1/runtime/invocation/response/{REQUEST_ID}")
    putErrorResponseUrl   = os.ExpandEnv("http://$"
{RUNTIME_API_ADDR}/v1/runtime/invocation/error/{REQUEST_ID}")
    requestIdNotFoundError = fmt.Errorf("request id invalid")
    noRequestAvailableError = fmt.Errorf("no request available")
    putResponseFailedError = fmt.Errorf("put response failed")
    functionPackage         = os.Getenv("RUNTIME_PACKAGE")
    functionName           = os.Getenv("RUNTIME_FUNC_NAME")
    functionVersion         = os.Getenv("RUNTIME_FUNC_VERSION")

    client = http.Client{
        Transport: &http.Transport{
            DialContext: (&net.Dialer{
                Timeout: 3 * time.Second,
            }).DialContext,
        },
    }
)

func main() {
    // main loop for processing requests.
    for {
        requestId, header, payload, err := getRequest()
        if err != nil {
            time.Sleep(50 * time.Millisecond)
            continue
        }

        result, err := processRequestEvent(requestId, header, payload)
        err = putResponse(requestId, result, err)
        if err != nil {
            log.Printf("put response failed, err: %s.", err.Error())
        }
    }
}

// event processing function
func processRequestEvent(requestId string, header http.Header, evtBytes
[]byte) ([]byte, error) {
```

```
log.Printf("processing request '%s'.", requestId)
result := fmt.Sprintf("function: %s:%s:%s, request id: %s, headers: %s,
+v, payload: %s", functionPackage, functionName,
    functionVersion, requestId, header, string(evtBytes))
var event FunctionEvent
err := json.Unmarshal(evtBytes, &event)
if err != nil {
    return (&ErrorMessage{ErrorType: "invalid event", ErrorMessage:
"invalid json formated event"}).ToJsonBytes(), err
}

return (&APIGFormatResult{StatusCode: 200, Body:
result}).ToJsonBytes(), nil
}

func getRequest() (string, http.Header, []byte, error) {
    resp, err := client.Get(getRequestUrl)
    if err != nil {
        log.Printf("get request error, err: %s.", err.Error())
        return "", nil, nil, err
    }
    defer resp.Body.Close()

    // get request id from response header
    requestId := resp.Header.Get("X-CFF-Request-Id")
    if requestId == "" {
        log.Printf("request id not found.")
        return "", nil, nil, requestIdInvalidError
    }

    payload, err := ioutil.ReadAll(resp.Body)
    if err != nil {
        log.Printf("read request body error, err: %s.", err.Error())
        return "", nil, nil, err
    }

    if resp.StatusCode != 200 {
        log.Printf("get request failed, status: %d, message: %s.",
resp.StatusCode, string(payload))
        return "", nil, nil, noRequestAvailableError
    }

    log.Printf("get request ok.")
    return requestId, resp.Header, payload, nil
}

func putResponse(requestId string, payload []byte, err error) error {
    var body io.Reader
    if payload != nil && len(payload) > 0 {
        body = bytes.NewBuffer(payload)
    }

    url := ""
    if err == nil {
        url = strings.Replace(putResponseUrl, "{REQUEST_ID}", requestId,
-1)
    } else {
        url = strings.Replace(putErrorResponseUrl, "{REQUEST_ID}",
requestId, -1)
    }

    resp, err := client.Post(strings.Replace(url, "{REQUEST_ID}",
```

```
requestId, -1), "", body)
    if err != nil {
        log.Printf("put response error, err: %s.", err.Error())
        return err
    }
    defer resp.Body.Close()

    responsePayload, err := ioutil.ReadAll(resp.Body)
    if err != nil {
        log.Printf("read request body error, err: %s.", err.Error())
        return err
    }

    if resp.StatusCode != 200 {
        log.Printf("put response failed, status: %d, message: %s.",
        resp.StatusCode, string(responsePayload))
        return putResponseFailedError
    }

    return nil
}

type FunctionEvent struct {
    Type string `json:"type"`
    Name string `json:"name"`
}

type APIGFormatResult struct {
    StatusCode      int          `json:"statusCode"`
    IsBase64Encoded bool        `json:"isBase64Encoded"`
    Headers        map[string]string `json:"headers,omitempty"`
    Body           string       `json:"body,omitempty"`
}

func (result *APIGFormatResult) toJsonBytes() []byte {
    data, err := json.MarshalIndent(result, "", " ")
    if err != nil {
        return nil
    }

    return data
}

type ErrorMessage struct {
    ErrorType      string `json:"errorType"`
    ErrorMessage   string `json:"errorMessage"`
}

func (errMsg *ErrorMessage) toJsonBytes() []byte {
    data, err := json.MarshalIndent(errMsg, "", " ")
    if err != nil {
        return nil
    }

    return data
}
```

Tabla 1-4 describe las variables de entorno utilizadas en el código anterior.

Tabla 1-4 Variables de entorno

Variable de entorno	Descripción
RUNTIME_FUNC_NAME	Nombre de la función
RUNTIME_FUNC_VERSION	Versión de la función
RUNTIME_PACKAGE	Aplicación a la que pertenece la función

2 Creación de una función de FunctionGraph

2.1 Creación de un paquete de implementación

Para crear una función, debe crear un paquete de implementación que incluya sus códigos y todas las dependencias. Puede crear un paquete de implementación localmente o editar códigos en la consola de FunctionGraph. Si edita códigos en línea, el FunctionGraph crea y carga automáticamente un paquete de implementación para su función. FunctionGraph le permite editar el código de función de la misma manera que administrar un proyecto. Puede crear y editar archivos y carpetas. Después de cargar un paquete ZIP de códigos, puede consultar y editar los códigos en la consola.

NOTA

- Después de la programación, simplemente empaqueta tus códigos en un archivo ZIP (Java, Node.js, Python y Go) o un archivo JAR (Java), y sube el archivo a FunctionGraph para su ejecución.
- Cuando cree un archivo ZIP, coloque el archivo de controlador bajo el directorio **root** para asegurarse de que su código puede ejecutarse normalmente después de ser descomprimido.
- Si edita código en Go, comprima el archivo compilado y asegúrese de que el nombre del archivo de biblioteca dinámica sea coherente con el nombre del complemento del controlador. Por ejemplo, si el nombre del archivo de biblioteca dinámica es **testplugin.so**, establezca el nombre del controlador en **testplugin.Handler**.

Tabla 2-1 enumera los modos de entrada de código admitidos por FunctionGraph a cada tiempo de ejecución.

Tabla 2-1 Modos de entrada de código

Tiempo de ejecución	Edición de código en línea	Carga de un archivo ZIP	Carga de un archivo JAR	Carga de un archivo ZIP desde OBS
Node.js	Soportado	Soportado	No soportado	Soportado
Python	Soportado	Soportado	No soportado	Soportado
Java	No soportado	Soportado	Soportado	Soportado
Go	No soportado	Soportado	No soportado	Soportado

Tiempo de ejecución	Edición de código en línea	Carga de un archivo ZIP	Carga de un archivo JAR	Carga de un archivo ZIP desde OBS
C#	No soportado	Soportado	No soportado	Soportado
PHP	Soportado	Soportado	No soportado	Soportado
Custom runtime	Soportado	Soportado	No soportado	Soportado

AVISO

Si el código que se va a cargar contiene informaciones confidenciales (como contraseñas de cuenta), cifrelas para evitar fugas.

Node.js

Edición de código en línea

FunctionGraph proporciona un kit de desarrollo de software (SDK) para editar código en Node.js. Si su código personalizado solo utiliza la biblioteca de SDK, puede editar códigos con el editor en línea en la consola FunctionGraph. Después de editar el código en línea y cargarlo a FunctionGraph, la consola comprime el código y las configuraciones relacionadas en un paquete de implementación que FunctionGraph puede ejecutar.

Carga de un paquete de implementación

Si el código utiliza otros recursos, como una biblioteca de gráficos para el procesamiento de imágenes, primero cree un paquete de implementación y luego cargue el paquete en la consola de FunctionGraph. Puede cargar un paquete de implementación de Node.js de dos maneras.

AVISO

- Cuando cree un archivo ZIP, coloque el archivo de controlador bajo el directorio **root** para asegurarse de que su código puede ejecutarse normalmente después de ser descomprimido.
- El tamaño del código fuente descomprimido no puede exceder de 1.5 GB. Si el código es demasiado grande, póngase en contacto con el especialista.
- Carga directa de un paquete de implementación local

Después de crear un paquete de implementación ZIP, súbelo a la consola de FunctionGraph. Si el tamaño del paquete supera los 50 MB, cargue el paquete desde OBS.

Para obtener más información sobre las restricciones de recursos de función, consulte [Cuotas y restricciones de uso](#).
- Carga de un paquete de implementación por un bucket de OBS

Después de crear un paquete de implementación ZIP, súbelo a un depósito OBS en la misma región que tu FunctionGraph y, a continuación, pega la URL de enlace del

depósito OBS en la función. El tamaño máximo del archivo ZIP que se puede subir a OBS es de 300 MB.

Para obtener más información sobre las restricciones de recursos de función, consulte [Cuotas y restricciones de uso](#).

Python

Edición de código en línea

FunctionGraph proporciona un SDK para editar código en Python. Si su código personalizado solo utiliza la biblioteca de SDK, puede editar códigos con el editor en línea en la consola FunctionGraph. Después de editar el código en línea y cargarlo a FunctionGraph, la consola comprime el código y las configuraciones relacionadas en un paquete de implementación que FunctionGraph puede ejecutar.

Carga de un paquete de implementación

Si el código utiliza otros recursos, como una biblioteca de gráficos para el procesamiento de imágenes, primero cree un paquete de implementación y luego cargue el paquete en la consola de FunctionGraph. Puede cargar un paquete de implementación de Python de dos maneras.

AVISO

- Cuando cree un archivo ZIP, coloque el archivo de controlador bajo el directorio **root** para asegurarse de que su código puede ejecutarse normalmente después de ser descomprimido.
- El tamaño del código fuente descomprimido no puede exceder de 1.5 GB. Si el código es demasiado grande, póngase en contacto con el especialista.
- Cuando escriba código en Python, no nomine su paquete con el mismo sufijo que una biblioteca estándar de Python, como json, lib y os. De lo contrario, se informará de un error que indica un fallo de carga del módulo.

● Carga directa de un paquete de implementación local

Después de crear un paquete de implementación ZIP, súbelo a la consola de FunctionGraph. Si el tamaño del paquete supera los 50 MB, cargue el paquete desde OBS.

Para obtener más información sobre las restricciones de recursos de función, consulte [Cuotas y restricciones de uso](#).

● Carga de un paquete de implementación por un bucket de OBS

Después de crear un paquete de implementación ZIP, súbelo a un depósito OBS en la misma región que tu FunctionGraph y, a continuación, pega la URL de enlace del depósito OBS en la función. El tamaño máximo del archivo ZIP que se puede subir a OBS es de 300 MB.

Para obtener más información sobre las restricciones de recursos de función, consulte [Cuotas y restricciones de uso](#).

Java

Java es un lenguaje compilado, que no admite la edición de código en línea. Solo puede cargar un paquete de implementación local, que puede ser un archivo ZIP o JAR.

Carga de un archivo JAR

- Si su función no utiliza ninguna dependencia, cargue directamente un archivo JAR.
- Si tu función utiliza dependencias, súbelas a un depósito de OBS, configúralas durante la creación de la función y sube el archivo JAR.

Carga de un archivo ZIP

Si su función utiliza dependencias de terceros, comprima las dependencias y el archivo JAR de función en un archivo ZIP y luego cargue el archivo ZIP.

Puede cargar un paquete de implementación de Java de dos maneras.

AVISO

- Cuando cree un archivo ZIP, coloque el archivo de controlador bajo el directorio **root** para asegurarse de que su código puede ejecutarse normalmente después de ser descomprimido.
- El tamaño del código fuente descomprimido no puede exceder de 1.5 GB. Si el código es demasiado grande, póngase en contacto con el especialista.

- Carga directa de un paquete de implementación local

Después de crear un paquete de implementación ZIP, súbelo a la consola de FunctionGraph. Si el tamaño del paquete supera los 50 MB, cargue el paquete desde OBS.

Para obtener más información sobre las restricciones de recursos de función, consulte [Cuotas y restricciones de uso](#).

- Carga de un paquete de implementación por un bucket de OBS

Después de crear un paquete de implementación ZIP, súbelo a un depósito OBS en la misma región que tu FunctionGraph y, a continuación, pega la URL de enlace del depósito OBS en la función. El tamaño máximo del archivo ZIP que se puede subir a OBS es de 300 MB.

Para obtener más información sobre las restricciones de recursos de función, consulte [Cuotas y restricciones de uso](#).

Go

Carga de un paquete de implementación

Solo puede cargar un paquete de implementación de Go en formato de ZIP. Hay dos maneras de subirlo.

AVISO

- Cuando cree un archivo ZIP, coloque el archivo de controlador bajo el directorio **root** para asegurarse de que su código puede ejecutarse normalmente después de ser descomprimido.
- El tamaño del código fuente descomprimido no puede exceder de 1.5 GB. Si el código es demasiado grande, póngase en contacto con el especialista.

- Carga directa de un paquete de implementación local

Después de crear un paquete de implementación ZIP, súbelo a la consola de FunctionGraph. Si el tamaño del paquete supera los 50 MB, cargue el paquete desde OBS.

Para obtener más información sobre las restricciones de recursos de función, consulte [Cuotas y restricciones de uso](#).

- Carga de un paquete de implementación por un bucket de OBS

Después de crear un paquete de implementación ZIP, súbelo a un depósito OBS en la misma región que tu FunctionGraph y, a continuación, pega la URL de enlace del depósito OBS en la función. El tamaño máximo del archivo ZIP que se puede subir a OBS es de 300 MB.

Para obtener más información sobre las restricciones de recursos de función, consulte [Cuotas y restricciones de uso](#).

C#

Carga de un paquete de implementación

Solo puede cargar un paquete de implementación de C# en formato ZIP. Hay dos maneras de subirlo.

AVISO

- Cuando cree un archivo ZIP, coloque el archivo de controlador bajo el directorio **root** para asegurarse de que su código puede ejecutarse normalmente después de ser descomprimido.
 - El tamaño del código fuente descomprimido no puede exceder de 1.5 GB. Si el código es demasiado grande, póngase en contacto con el especialista.
-
- Carga directa de un paquete de implementación local

Después de crear un paquete de implementación ZIP, súbelo a la consola de FunctionGraph. Si el tamaño del paquete supera los 50 MB, cargue el paquete desde OBS.

Para obtener más información sobre las restricciones de recursos de función, consulte [Cuotas y restricciones de uso](#).
 - Carga de un paquete de implementación por un bucket de OBS

Después de crear un paquete de implementación ZIP, súbelo a un depósito OBS en la misma región que tu FunctionGraph y, a continuación, pega la URL de enlace del depósito OBS en la función. El tamaño máximo del archivo ZIP que se puede subir a OBS es de 300 MB.

Para obtener más información sobre las restricciones de recursos de función, consulte [Cuotas y restricciones de uso](#).

PHP

Edición de código en línea

FunctionGraph proporciona un SDK para editar código en PHP. Si su código personalizado solo utiliza la biblioteca de SDK, puede editar códigos con el editor en línea en la consola FunctionGraph. Después de editar el código en línea y cargarlo a FunctionGraph, la consola comprime el código y las configuraciones relacionadas en un paquete de implementación que FunctionGraph puede ejecutar.

Carga de un paquete de implementación

Si el código utiliza otros recursos, como una biblioteca de gráficos para el procesamiento de imágenes, primero cree un paquete de implementación y luego cargue el paquete en la consola de FunctionGraph. Puede cargar un paquete de implementación de PHP de dos maneras.

AVISO

- Cuando cree un archivo ZIP, coloque el archivo de controlador bajo el directorio **root** para asegurarse de que su código puede ejecutarse normalmente después de ser descomprimido.
 - El tamaño del código fuente descomprimido no puede exceder de 1.5 GB. Si el código es demasiado grande, póngase en contacto con el especialista.
-
- Carga directa de un paquete de implementación local
Después de crear un paquete de implementación ZIP, súbelo a la consola de FunctionGraph. Si el tamaño del paquete supera los 50 MB, cargue el paquete desde OBS.
Para obtener más información sobre las restricciones de recursos de función, consulte [Cuotas y restricciones de uso](#).
 - Carga de un paquete de implementación por un bucket de OBS
Después de crear un paquete de implementación ZIP, súbelo a un depósito OBS en la misma región que tu FunctionGraph y, a continuación, pega la URL de enlace del depósito OBS en la función. El tamaño máximo del archivo ZIP que se puede subir a OBS es de 300 MB.
Para obtener más información sobre las restricciones de recursos de función, consulte [Cuotas y restricciones de uso](#).

Tiempo de ejecución personalizado

Edición de código en línea

Después de editar el código en línea y cargarlo a FunctionGraph, la consola comprime el código y las configuraciones relacionadas en un paquete de implementación que FunctionGraph puede ejecutar.

Carga de un paquete de implementación

Si el código utiliza otros recursos, como una biblioteca de gráficos para el procesamiento de imágenes, primero cree un paquete de implementación y luego cargue el paquete en la consola de FunctionGraph. Puede cargar un paquete de implementación para un tiempo de ejecución personalizado de dos maneras.

AVISO

-
- Cuando cree un archivo ZIP, coloque el archivo de controlador bajo el directorio **root** para asegurarse de que su código puede ejecutarse normalmente después de ser descomprimido.
 - El tamaño del código fuente descomprimido no puede exceder de 1.5 GB. Si el código es demasiado grande, póngase en contacto con el especialista.
-

- Carga directa de un paquete de implementación local
Después de crear un paquete de implementación ZIP, súbelo a la consola de FunctionGraph. Si el tamaño del paquete supera los 50 MB, cargue el paquete desde OBS.
Para obtener más información sobre las restricciones de recursos de función, consulte [Cuotas y restricciones de uso](#).
- Carga de un paquete de implementación por un bucket de OBS
Después de crear un paquete de implementación ZIP, súbelo a un depósito OBS en la misma región que tu FunctionGraph y, a continuación, pega la URL de enlace del depósito OBS en la función. El tamaño máximo del archivo ZIP que se puede subir a OBS es de 300 MB.
Para obtener más información sobre las restricciones de recursos de función, consulte [Cuotas y restricciones de uso](#).

2.2 Creación de una función desde cero

2.2.1 Creación de una función de evento

Información general

Una función es un código personalizado para procesar eventos. Puede crear una función desde cero y configurarla en función de los requisitos del sitio.

FunctionGraph gestiona los recursos de cómputo necesarios para la ejecución de la función. Después de editar el código para su función, configure los recursos informáticos en la consola FunctionGraph.

Puede crear una función desde cero o usando [una plantilla](#) o [una imagen contenedora](#).

NOTA

Al crear una función desde cero, configure la información básica y la de código basada en [Tabla 2-2](#). Los parámetros marcados con un asterisco (*) son obligatorios.

Después de crear una función, modifique su configuración en función de [Modificación de configuraciones de función](#) si es necesario.

Cada función de FunctionGraph se ejecuta en su propio entorno y tiene sus propios recursos y sistema de archivos.

Prerrequisitos:

1. Debe estar familiarizado con los lenguajes de programación compatibles con FunctionGraph. Para más detalles, consulte [Lenguajes de programación compatibles](#).
2. Ha creado un paquete. Para más detalles, consulte [Creación de un paquete de implementación](#).
3. (Opcional) Ha creado una agencia. Para más detalles, consulte [Configuración de permisos de delegación](#).

Procedimiento

1. Inicie sesión en [la consola de FunctionGraph](#). En el panel de navegación, seleccione **Functions > Function List**.

2. En la página **Function List**, haga clic en **Create Function** en la esquina superior derecha.
3. Haga clic en **Create from scratch** y configure la información de la función haciendo referencia a **Tabla 2-2**. Los parámetros marcados con un asterisco (*) son obligatorios.

Figura 2-1 Creación de una función desde cero

Basic Information

* FunctionGraph Version

FunctionGraph v2

* Function Type

Event Function HTTP Function

* Function Name

functiongraph

Enter 1 to 60 characters, starting with a letter and ending with a letter or digit. Only letters, digits, hyphens (-), and underscores (_) are allowed.

Agency ?

Use no agency C Create Agency

* Enterprise Project ?

default C View Enterprise Project

Runtime ?

Node.js 10.16 Learn how to develop functions in Node.js.

Tabla 2-2 Información básica

Parámetro	Descripción
*FunctionGraph Version	Seleccione una versión de FunctionGraph. <ul style="list-style-type: none">● v1: aloja funciones controladas por eventos en un contexto sin servidor.● FunctionGraph v2: un servicio de alojamiento de funciones de última generación impulsado por la arquitectura YuanRong de Huawei.

Parámetro	Descripción
* Function Type	<p>Si establece FunctionGraph Version en FunctionGraph v2, debe seleccionar un tipo de función.</p> <ul style="list-style-type: none"> Funciones de eventos: se activan por activadores. Funciones HTTP: se activan una vez que las solicitudes HTTP se envían a URLs específicas. <p>NOTA</p> <ul style="list-style-type: none"> Las funciones HTTP no distinguen entre lenguajes de programación. El controlador debe estar establecido en el archivo bootstrap. Puede escribir directamente el comando startup y permitir el acceso a través del puerto 8000. Las funciones HTTP solo admiten activadores APIG y APIC. Para obtener más información sobre cómo usar las funciones HTTP, consulte Creación de una función HTTP.
*Function Name	<p>Nombre de la función, que debe cumplir los siguientes requisitos:</p> <ul style="list-style-type: none"> Consta de 1 a 60 caracteres y puede contener letras, dígitos, guiones (-) y guiones bajos (_). Comienza con una letra y termina con una letra o dígito.
Agency	<p>Se requiere una delegación si FunctionGraph accede a otros servicios en la nube. Para obtener más información sobre cómo crear una agencia, consulte Configuración de permisos de delegación.</p> <p>No se requiere delegación si FunctionGraph no accede a ningún servicio en la nube.</p>
*Enterprise Project	<p>Seleccione un proyecto de empresa creado y agréguele la función. De forma predeterminada, se selecciona default.</p> <p>NOTA</p> <p>Si Enterprise Project Management Service (EPS) no está habilitado, este parámetro no estará disponible. Para obtener más información, consulte Habilitación de la función del proyecto empresarial.</p>
Runtime	<p>Seleccione un tiempo de ejecución para compilar la función.</p> <p>AVISO</p> <p>CloudIDE solo admite Node.js, Python y PHP.</p>

4. Haga clic en **Create Function**. En la página de ficha **Code** que se muestra, continúe configurando el código.

Configuración del código

1. Puede implementar el código en función del tiempo de ejecución que seleccione. Para más detalles, consulte [Creación de un paquete de implementación](#). Una vez completada la implementación, haga clic en **Deploy**.
Por ejemplo, para implementar código (**Figura 2-2**) en Node.js 10.16, puede editar código en línea, cargar un archivo ZIP local o cargar un archivo ZIP desde OBS.

Figura 2-2 Despliegue de código

2. Puede modificar el código y hacer clic en **Deploy** para implementar el código de nuevo.

Consulta de información de código

1. Consultar los atributos de código.

Los atributos de código muestran el tamaño del código y la hora en que se modificó el código.

Figura 2-3 Consulta de atributos de código

2. Consultar información básica.

Figura 2-4 muestra la memoria predeterminada y el tiempo de espera de ejecución en cada tiempo de ejecución. Puede hacer clic en **Edit** para cambiar a la página **Basic Settings** y modificar **Handler**, **Memory (MB)** y **Execution Timeout (s)** según sea necesario. Para más detalles, consulte [Configuración de los parámetros básicos](#).

Figura 2-4 Edición de información básica

AVISO

Una vez se crea una función, no se puede cambiar el tiempo de ejecución.

Tabla 2-3 Información básica predeterminada de cada tiempo de ejecución

Tiempo de ejecución	Información básica predeterminada
JAVA	Memoria (MB): 512 Controlador: com.huawei.demo.TriggerTests.apigTest Tiempo de espera de ejecución (s): 15
Node.js	Memoria (MB): 128 Controlador: index.handler Tiempo de espera de ejecución (s): 3

Tiempo de ejecución	Información básica predeterminada
Personalizado	Memoria (MB): 128 Controlador: bootstrap Tiempo de espera de ejecución (s): 3
PHP	Memoria (MB): 128 Controlador: index.handler Tiempo de espera de ejecución (s): 3

2.2.2 Creación de una función HTTP

NOTA

Esta función solo es compatible con FunctionGraph v2.

Información general

Se diseñan las funciones HTTP para optimizar los servicios web. Puede enviar solicitudes HTTP a direcciones URL para activar la ejecución de la función. Las funciones HTTP solo admiten activadores APIG y APIC.

NOTA

- Las funciones HTTP no distinguen entre lenguajes de programación. El controlador debe estar establecido en el archivo bootstrap. Puede escribir directamente el comando startup y permitir el acceso a través del puerto 8000. La dirección IP enlazada es **127.0.0.1**.
- El archivo bootstrap es el archivo de inicio de una función HTTP. La función HTTP solo puede leer bootstrap como el nombre del archivo de inicio. Si el nombre del archivo no es bootstrap, el servicio no se puede iniciar. Para obtener más información, consulte el [ejemplo del archivo bootstrap](#).
- Las funciones HTTP soportan múltiples lenguajes de programación.
- Esta sección utiliza Java como ejemplo. Para utilizar otro tiempo de ejecución, simplemente cambie la ruta de acceso del tiempo de ejecución. No es necesario cambiar la ruta del paquete de código. Para las rutas de otros tiempos de ejecución, consulte [Tabla 2-4](#).

Prerrequisitos:

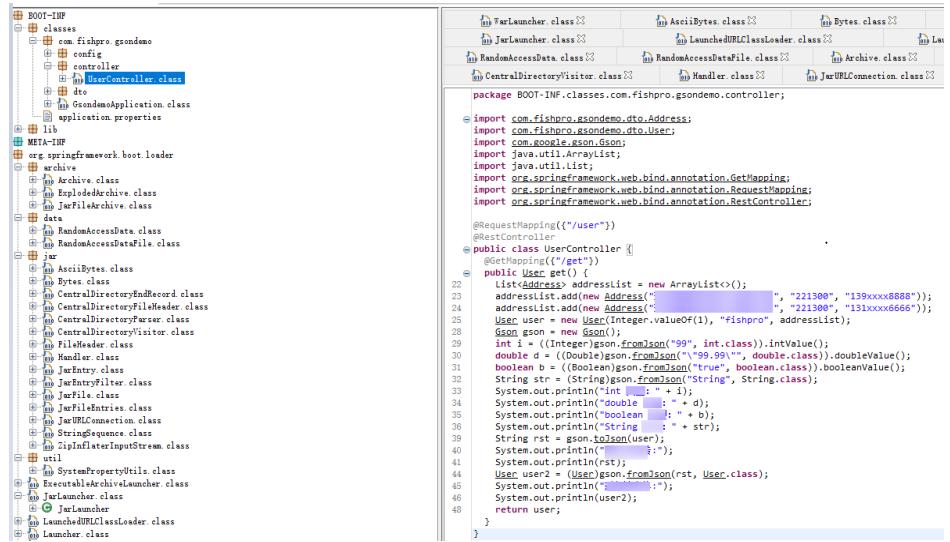
- Ha preparado un paquete de Java JAR.
- Ha preparado un archivo bootstrap como el archivo de inicio de la función HTTP.

Ejemplo

El contenido del archivo bootstrap es el siguiente:

```
/opt/function/runtime/java8/rtsp/jre/bin/java -jar -Dfile.encoding=utf-8 /opt/function/code/gsondemo-0.0.1-SNAPSHOT.jar
```

- /opt/function/runtime/java8/rtsp/jre/bin/java**: ruta de Java.
- /opt/function/code**: ruta del paquete de código de función.
- gsondemo-0.0.1-SNAPSHOT.jar**: ejemplo de paquete JAR. La ruta de servicio es **/user/get**.



Para utilizar otro tiempo de ejecución, cambie la ruta de acceso en tiempo de ejecución haciendo referencia a [Tabla 2-4](#). No es necesario cambiar la ruta del paquete de código.

Tabla 2-4 Rutas de acceso para diferentes tiempos de ejecución

Tiempo de ejecución	Ruta
Java 8	/opt/function/runtime/java8/rtsp/jre/bin/java
Java 11	/opt/function/runtime/java11/rtsp/jre/bin/java
Node.js 6	/opt/function/runtime/nodejs6.10/rtsp/nodejs/bin/node
Node.js 8	/opt/function/runtime/nodejs8.10/rtsp/nodejs/bin/node
Node.js 10	/opt/function/runtime/nodejs10.16/rtsp/nodejs/bin/node
Node.js 12	/opt/function/runtime/nodejs12.13/rtsp/nodejs/bin/node
Node.js 14	/opt/function/runtime/nodejs14.18/rtsp/nodejs/bin/node
Python 2.7	/opt/function/runtime/python2.7/rtsp/python/bin/python
Python 3.6	/opt/function/runtime/python3.6/rtsp/python/bin/python3
Python 3.9	/opt/function/runtime/python3.9/rtsp/python/bin/python3
PHP 7.3	/opt/function/runtime/php7.3/rtsp/php/bin/php

Procedimiento

1. Creación de una función.
 - a. Creación de una función HTTP. Para más detalles, consulte [Creación de una función de evento](#). Preste especial atención a los siguientes parámetros:
 - **FunctionGraph Version:** v2 de FunctionGraph
 - **Tipo de función:** Función HTTP
 - b. Carga del código. Por ejemplo, cargue un archivo ZIP desde OBS. Una vez completada la carga, haga clic en **Deploy**.

Comprima el paquete JAR y el archivo de arranque, y elija **Upload > OBS ZIP**.

Figura 2-5 Carga de un archivo ZIP desde OBS



2. Creación de un activador.

NOTA

Las funciones HTTP solo admiten activadores APIG y APIC.

- En la página de detalles de la función, seleccione **Configuration > Trigger** y haga clic en **Create Trigger**.
- Configure la información del activador. A continuación se describe cómo crear un activador APIG. Para más detalles, consulte [Uso de un activador APIG \(dedicado\)](#).

Figura 2-6 Creación de un activador

Create Trigger

Trigger Type API Gateway (APIG)

For a function using APIG triggers, set the response body in the following JSON format:
`{"statusCode": 200, "isBase64Encoded": false, "headers": {"Content-Type": "application/json;charset=UTF-8"}, "body": "hello world"}`
 (The request body will be encrypted using Base64.)

★ API Name Enter 3 to 64 characters, starting with a letter. Only letters, digits, and underscores (_) are allowed.

★ API Group C Create API Group

★ Environment C Create Environment

★ Security Authentication None Authentication will not be performed and all users will be granted access. (Not recommended)

★ Protocol

★ Timeout(ms) Set a backend timeout from 1 ms to 60,000 ms.

NOTA

En este ejemplo, **Security Authentication** se establece en **None**. Debe seleccionar un modo de autenticación basado en los requisitos del sitio.

- **App**: Autenticación de AppKey y AppSecret. Este modo es de alta seguridad y se recomienda.
 - **IAM**: Autenticación de IAM. Este modo solo otorga permisos de acceso a los usuarios de IAM y es de seguridad media.
 - **None**: No hay autenticación. Este modo otorga permisos de acceso a todos los usuarios.
- Cuando se complete la configuración, haga clic en **OK**. Una vez creado el activador, **API_test_http** se generará en la consola de API Gateway.

3. Publica la API.

- En la página de la pestaña **Triggers**, haga clic en un nombre de API para ir a la página de descripción general de la API.

Figura 2-7 Activador APIG

The screenshot shows the 'Configuration' tab of the APIG interface. On the left, a sidebar lists 'Basic Settings', 'Trigger' (which is selected and highlighted in blue), 'Permissions', 'VPC', 'Disk Mounting', 'Environment Variables', 'Concurrency', 'Configure Async Notification', and 'Advanced Settings'. The main content area is titled 'Trigger' and shows 'Total triggers: 1'. Below this, there is a table for the API 'API_http_test'. The table includes columns for 'APIG' (Subtotal: 1), 'Trigger' (API_http_test, Enabled), 'URL' (https://52fc3b7a1be.dapis.com/http_test), 'API Group' (group_test), 'Environment' (RELEASE), 'Security Authentication' (IAM), 'Method' (ANY), 'Path' (/http_test), and 'Timeout' (5000 ms). A 'Delete' button is located in the top right corner of the table.

- Haga clic en **Edit** en la esquina superior derecha. Se muestra la página **Basic Information**.

Figura 2-8 Edición de una API

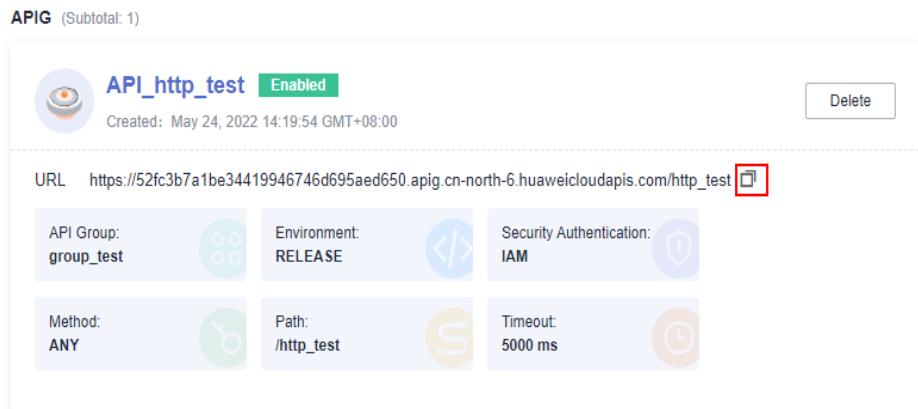
The screenshot shows the 'Basic Information' tab of the API editing interface. The top navigation bar includes 'APIs / API_http_test / Switch API', 'Dashboard', 'API Call', 'Authorization', 'Request Throttling Policies', 'Access Control Policies', 'Signature Keys', 'Publication History', 'Export', 'Debug', 'Publish', 'Take Offline', 'Edit' (which is highlighted in red), and 'Delete'. The main content area shows the 'Basic Information' for the API 'API_http_test'.

- Haga clic en **Next**. En la página **Define API Request** que se muestra, cambie **Path** a **/user/get** y haga clic en **Finish**.

Figura 2-9 Definición de una solicitud de API

The screenshot shows the 'Define API Request' configuration page. At the top, it says 'Define API Request'. Below that, 'Domain Name' is set to '52fc3b7a1be.dapis.com'. The 'Protocol' section shows 'HTTP' (selected) and 'HTTPS'. A note says 'WebSocket is supported for HTTP and HTTPS.' The 'Path' field is set to '/user/get' (highlighted with a red box). A note says 'Enclose parameters in braces, for example, /a/{b}. You can also use a plus sign (+) to match parameters starting with specific characters, for example, /a/{b+}.' The 'Matching' section shows 'Exact match' (selected) and 'Prefix match'. A note says 'API requests will be forwarded to paths starting with the specified characters, for example, /a.' The 'Method' dropdown is set to 'ANY'. The 'CORS' section has a switch that is turned off. A note says 'Enable cross-origin resource sharing (CORS) if you want to allow restricted resources on a web page to be requested from other domains. Learn more about CORS.'

- d. Haz clic en **Publish API**. En la página mostrada, haga clic en **Publish**.
4. Activar una función.
 - a. Vaya a la consola de FunctionGraph y seleccione **Functions > Function List** en el panel de navegación y haga clic en la función HTTP creada para ir a su página de detalles.
 - b. Seleccione Configuration > Trigger, copie la URL y acceda a ella mediante un navegador.

Figura 2-10 Copia de la URL

- c. Consulte el resultado de la solicitud.

Figura 2-11 Consulta del resultado de la solicitud

```
[{"userId": 1, "username": "fishpro", "addresses": [{"street": "123 Main Street", "city": "Anytown", "state": "CA", "zipCode": "1234567890", "mobile": "139xxxx8888"}, {"street": "456 Elm Street", "city": "Anytown", "state": "CA", "zipCode": "221300", "mobile": "131xxxx6666"}], "created": "2021-01-01T00:00:00Z", "updated": "2021-01-01T00:00:00Z", "lastUpdated": "2021-01-01T00:00:00Z"}]
```

Encabezados de solicitud de función común

En la siguiente tabla se enumeran los campos de encabezado de solicitud predeterminados de una función HTTP.

Tabla 2-5 Campos de encabezado de solicitud predeterminados

Campo	Descripción
X-CFF-Request-Id	ID de la solicitud actual
X-CFF-Memory	Memoria asignada
X-CFF-Timeout	Duración del tiempo de espera de la función
X-CFF-Func-Version	Versión de la función
X-CFF-Func-Name	Nombre de la función
X-CFF-Project-Id	ID del proyecto
X-CFF-Package	Aplicación a la que pertenece la función

Campo	Descripción
X-CFF-Region	Región actual

2.3 Creación de una función mediante una plantilla

Información general

FunctionGraph proporciona plantillas para completar código automáticamente y ejecutar configuraciones de entorno cuando crea una función, lo que le ayuda a crear aplicaciones rápidamente.

Creación de una función

1. Inicie sesión en [la consola de FunctionGraph](#). En el panel de navegación, elija **Templates**.
 2. En la página que se muestra, seleccione el servicio de **FunctionGraph** y seleccione la plantilla de **context-class-introduction** para Python 2.7 y haga clic en **Configure**.

NOTA

La plantilla **context-class-introduction** para Python 2.7 se usa como ejemplo. También puede seleccionar otras plantillas.

3. Despues de seleccionar una plantilla de función, el código integrado y las configuraciones de la plantilla se cargan automáticamente. Se muestra la página **Create Function**.
 4. Establezca **Function Name** en **context**, seleccione una delegación creada, conserve los valores predeterminados para otros parámetros y haga clic en **Create Function**.

NOTA

Si no se configura ninguna agencia, se mostrará el siguiente mensaje cuando se active la función:

Error al acceder a otros servicios porque no se ha obtenido ningún AK, SK o token temporal. Por favor, establezca una delegación.

- ## 5. Haga clic en **Save**.

Activación de una función

1. En la página de la ficha **Code** de la función de **context**, haga clic en **Test** en la esquina superior derecha.
 2. En el cuadro de diálogo **Configure Test Event**, seleccione **Blank Template** y haga clic en **Create**.
 3. Haga clic en **Test**. Una vez completada la prueba, vea el resultado de la prueba.

Figura 2-12 Resultado de ejecución correcto

2.4 Implementación de una función mediante una imagen de contenedor

Información general

Puede empaquetar y cargar directamente imágenes de contenedores. Las imágenes son cargadas e iniciadas por la plataforma y se pueden llamar de manera similar a las funciones HTTP. A diferencia del modo de carga de código anterior, puede usar un paquete de código personalizado, que es flexible y reduce los costos de migración. Puede crear imágenes personalizadas tanto para funciones de evento como para las de HTTP.

Se admiten las siguientes características:

- Descarga de imágenes

Las imágenes se almacenan en SWR y solo pueden ser descargadas por los usuarios con el permiso de **SWR Admin**. FunctionGraph llamará a la API de SWR para generar y establecer comandos de inicio de sesión temporales antes de crear instancias.

- Definición de variables de entorno

Se admite la configuración de cifrado y las variables de entorno. Para más detalles, consulte [Configuración de variables de entorno](#).

- Conexión de discos de datos externos

Se pueden conectar con los discos de datos externos. Para más detalles, consulte [Configuración del montaje en disco](#).

- Facturación

No se le facturará cuando descargue imágenes o espere a que las imágenes estén listas.

- Instancias reservadas:

Se admiten las instancias reservadas. Para más detalles, consulte [Gestión de instancias reservadas](#).

NOTA

Los contenedores de usuario se iniciarán usando UID 1003 y GID 1003, que son los mismos que otros tipos de funciones.

Prerrequisitos:

Ha creado una delegación con los permisos de **SWR Admin** haciendo referencia [Configuración de permisos de delegación](#). Las imágenes se almacenan en SWR, y solo los usuarios con este permiso pueden invocar y extraer imágenes.

Procedimiento

1. Inicie sesión en [la consola de FunctionGraph](#). En el panel de navegación, seleccione **Functions > Function List**.
2. En la página **Function List**, haga clic en **Create Function** en la esquina superior derecha.
3. Seleccione **Use container image**. Para más detalles, consulte [Tabla 2-6](#).

Figura 2-13 Creación de una función mediante una imagen contenedora

Basic Information

Function Type

Event Function HTTP Function

Function Name

image

Enter 1 to 60 characters, starting with a letter and ending with a letter or digit. Only letters, digits, hyphens (-), and underscores (_) are allowed.

Enterprise Project

default

Use container image

swr.my cloud.com/my_group/my_image:latest

Container Image Override

CMD

Enter one or more CMD.

Args

Enter one or more arguments.

Working Dir

Enter a working directory.

User ID

This field cannot be left blank.

Group ID

This field cannot be left blank.

Permissions

No agency Use agency

Agency

test-trust

To create a function using an image, select an agency with the SWR Admin permission.

Tabla 2-6 Descripción del parámetro

Parámetro	Descripción
*Function Type	<p>Seleccione un tipo de función.</p> <ul style="list-style-type: none"> ● Funciones de eventos: se activan por activadores. ● Funciones HTTP: se activan una vez que las solicitudes HTTP se envían a URLs específicas. <p>NOTA</p> <ul style="list-style-type: none"> ● Las funciones HTTP no distinguen entre lenguajes de programación. El controlador debe estar establecido en el archivo bootstrap. Puede escribir directamente el comando startup y permitir el acceso a través del puerto 8000. ● Las funciones HTTP solo admiten activadores APIG y APIC. ● Para obtener más información sobre cómo usar las funciones HTTP, consulte Creación de una función HTTP.
*Function Name	<p>Nombre de la función, que debe cumplir los siguientes requisitos:</p> <ul style="list-style-type: none"> ● Consta de 1 a 60 caracteres y puede contener letras, dígitos, guiones (-) y guiones bajos (_). ● Comienza con una letra y termina con una letra o dígito.
*Enterprise Project	<p>Seleccione un proyecto de empresa creado y agréguele la función. De forma predeterminada, se selecciona default.</p> <p>NOTA</p> <p>Si EPS no está habilitado, este parámetro no estará disponible. Para obtener más información, consulte Habilitación de la función del proyecto empresarial.</p>
Container Image	<p>Introduzca una URL de imagen, es decir, la ubicación de la imagen contenedora. Puede hacer clic en View Image para ver imágenes privadas y compartidas.</p> <p>Imagen en SWR, por ejemplo, swr.myhuaweicloud.com/my_group/my_image:latest.</p>
Container Image Override	<ul style="list-style-type: none"> ● CMD: comando de inicio de contenedor. Ejemplo: /bin/sh. Si no se especifica ningún comando, se usará el punto de entrada o CMD en la configuración de imagen. Introduzca uno o más comandos separados por comas (,). ● Args: parámetro de inicio de contenedor. Ejemplo: -args,value1. Si no se especifica ningún argumento, se usará CMD en la configuración de imagen. Introduzca uno o más argumentos separados por comas (,). ● Working Dir: directorio de trabajo que ejecuta un contenedor. Si no se especifica ningún directorio, se utilizará el directorio en la configuración de la imagen. El directorio debe comenzar con una barra diagonal (/). ● User ID: ID de usuario para ejecutar la imagen. Si no se especifica ningún ID de usuario, se usará el valor por defecto 1003. ● Group ID: ID de grupo de usuario. Si no se especifica ningún ID de grupo de usuario, se usará el valor por defecto 1003.

Parámetro	Descripción
Agency	Seleccione una delegación con permisos de SWR Admin . Para crear una delegación, consulte Creación de una delegación .

NOTA

- Command, Args, y Working dir pueden contener hasta 5120 caracteres.
- Después de crear una función, puede determinar si desea habilitar la inicialización. Si la inicialización está habilitada para una función, se invocará un inicializador especificado para comenzar la función y, a continuación, se invocará un controlador para procesar solicitudes. Para más detalles, consulte [Configuración de la inicialización](#).
- Si la inicialización está habilitada, se accederá a **127.0.0.1:8000/init** cuando se inicie la imagen por primera vez, y se accederá a **127.0.0.1:8000/invoke** durante las llamadas a la API posteriores.
- La imagen debe ser pública.
- El puerto de ejecución de la función HTTP se establece en **8000**.

Código de muestra

Lo siguiente utiliza **NodeJS Express** como ejemplo. Durante la inicialización de la función, el FunctionGraph utiliza el método POST para acceder a la ruta **/init** (opcional). Cada vez que se llama a una función, FunctionGraph utiliza el método POST para acceder a la ruta **/invoke**. La función obtiene **context** desde **req.headers**, obtiene **event** desde **req.body** y devuelve las estructuras de respuesta HTTP como resultado.

```
const express = require('express');
const app = express();
const PORT = 8000;

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

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

app.listen(PORT, () => {
  console.log(`Listening on http://localhost:${PORT}`);
});
```

Diferencias entre imágenes de contenedores, funciones de eventos y funciones HTTP

Tabla 2-7 Diferencias entre imágenes de contenedores, funciones de eventos y funciones HTTP

Artículo de comparación	Función de evento	Función HTTP	Imagen del contenedor
Lenguaje	Cualquier lenguaje	Cualquier lenguaje	Cualquier lenguaje
Modo de inicio	Inicio del proceso	Inicio del proceso	Inicio de contenedores
Entorno	Entornos especificados	Entornos especificados	Alta flexibilidad y personalización
Modo de conexión de FunctionGraph	Sirve como un cliente de HTTP	Sirve como un servidor de HTTP	Sirve como un servidor de HTTP
Modo de implementación de dependencias	Capa	Capa	En la imagen
Latencia de arranque en frío	Corto	Corto	Largo

3 Configuración de la función

3.1 Configuración de la inicialización

Información general

Después de crear una función, habilite la inicialización. Si la inicialización está habilitada para una función, se invocará un inicializador especificado para comenzar la función y, a continuación, se invocará un controlador para procesar solicitudes.

Prerrequisitos:

Ha creado una función.

Inicialización de una función

- Paso 1** Inicie sesión en [la consola de FunctionGraph](#). En el panel de navegación, seleccione **Functions > Function List**.
- Paso 2** Haga clic en la función que se va a configurar para ir a la página de detalles de la función.
- Paso 3** Haga clic en la ficha **Configuration** y elija **Configuración avanzada**.

Figura 3-1 Habilitar la inicialización

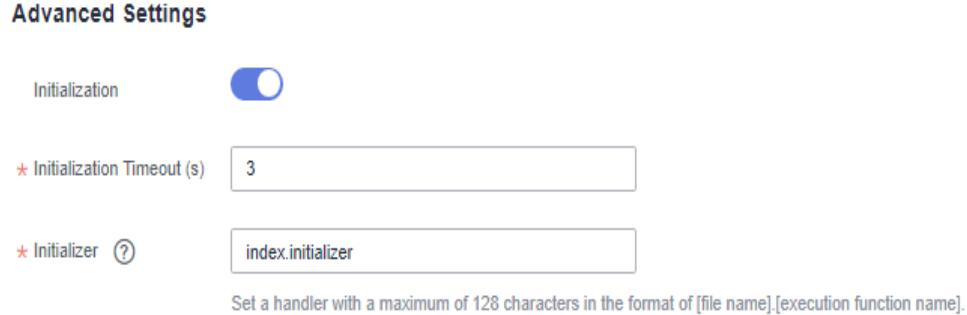


Tabla 3-1 Configuración de parámetros

Parámetro	Descripción
Initialization	Habilite la inicialización si es necesario.
Initialization Timeout (s)	Duración máxima de la función puede ser inicializada. Establezca este parámetro si habilita la inicialización de funciones. El valor oscila entre 1 y 300s.
Initializer	Puede habilitar la inicialización de funciones en la página de ficha Configuration . El inicializador debe ser nombrado del mismo modo que el manejador. Por ejemplo, para una función Node.js o Python, establezca un nombre de inicializador en el formato de <i>[file name].[initialization function name]</i> . NOTA Este parámetro no es necesario si la inicialización de la función está deshabilitada.

NOTA

- Establezca el inicializador de la misma manera que el manejador. Por ejemplo, para una función Node.js o Python, establezca un nombre de inicializador en el formato de *[file name].[initialization function name]*.
- Para obtener más información sobre la configuración del código de función, consulte [Creación de un paquete de implementación](#).

----Fin

3.2 Configuración de los parámetros básicos

Información general

Después de crear una función, **Memory**, **Handler** y **Execution Timeout (s)** se establecen automáticamente en función del tiempo de ejecución. Si es necesario, modificarlos según esta sección.

Prerrequisitos:

Ha creado una función.

Procedimiento

- Inicie sesión en [la consola de FunctionGraph](#). En el panel de navegación, seleccione **Functions > Function List**.
- Haga clic en la función que se va a configurar para ir a la página de detalles de la función.
- Elija **Configuration > Basic Settings** y configure los parámetros basados en [Tabla 3-2](#). Los parámetros marcados con un asterisco (*) son obligatorios.

Tabla 3-2 Parámetros básicos

Parámetro	Descripción
App	<p>Después de crear una función, se categoriza automáticamente en la aplicación default y no se puede cambiar a otras aplicaciones.</p> <p>AVISO Una aplicación actúa como una carpeta. En el futuro, las funciones se gestionarán por etiqueta para una mejor experiencia.</p>
*Handler	<ul style="list-style-type: none"> Para una función Node.js, Python o PHP, el controlador debe ser nombrado en el formato de <i>[file name].[function name]</i>, que debe contener un punto (.). Ejemplo: myfunction.handler Para una función Java, el controlador debe ser nombrado en el formato de <i>[package name].[file name].[function name]</i>. Ejemplo: com.xxxxx.exp.Myfunction.myHandler Para una función Go, el controlador debe ser nombrado en el formato de <i>[plug-in name].[function name]</i>. El nombre de la función debe comenzar con una letra mayúscula. El nombre del controlador puede contener un máximo de 128 caracteres. Ejemplo: Function.Handler Para una función de C#, el controlador debe ser nombrado en el formato de <i>[.NET assembly file name]::[namespace and class of the handler function]::[handler function name]</i>. Ejemplo: HelloCsharp::Example.Hello::Handler
*Enterprise Project	<p>Seleccione un proyecto de empresa creado y agréguele la función. De forma predeterminada, se selecciona default.</p> <p>NOTA Si EPS no está habilitado, este parámetro no estará disponible. Para obtener más información, consulte Habilitación de la función del proyecto empresarial.</p>
*Execution Timeout (s)	<p>Duración máxima de la función puede ser ejecutada. Puede establecer este parámetro en la página de ficha Configuration. Si la ejecución tarda más de 90s, utilice la invocación asincrónica.</p> <p>El valor oscila entre 3s y 900s.</p> <p>NOTA Para establecer el tiempo de espera de ejecución en 900s a 43,200s, póngase en contacto con el soporte técnico.</p>
Memory (MB)	Memoria de una instancia de función. Opciones: 128, 256, 512, 768, 1024, 1280, 1536, 1792, 2048, 2560, 3072, 3584 y 4096.
Description	Descripción de la función, que no puede superar los 512 caracteres.

4. Haga clic en **Save**.

3.3 Configuración de permisos de delegación

Información general

Los servicios en Huawei Cloud interactúan entre sí, y algunos servicios en la nube dependen de otros servicios. Para delegar un servicio en la nube para acceder a otros servicios y realizar operaciones y mantenimiento de recursos, cree una agencia para el servicio.

Creación de una delegación

NOTA

En el siguiente ejemplo, el permiso **Tenant Administrator** se asigna a FunctionGraph y esta configuración solo tiene efecto en CN North-Beijing4.

Crear una delegación basada en [Creación de una delegación](#) y establecer los parámetros siguientes:

1. Inicie sesión en la consola IAM.
2. En la consola de IAM, seleccione **Agencies** en el panel de navegación y haga clic en **Create Agency** en la esquina superior derecha.

Figura 3-2 Creación de una delegación

Agency Name	Delegated Party	Validity Period	Created	Description	Operation
modelarts_agency	Cloud service Modelarts	Unlimited	Jan 18, 2022 16:08:44 GMT+00:00	Created by ModelArts service.	Authorize Modify Delete
ctg_admin_trust	Cloud service Cloud Trace Service (CTS)	Unlimited	Nov 24, 2021 16:09:44 GMT+00:00	Created by CTS: To ensure that services run properly, d...	Authorize Modify Delete
romalab_admin_trust	Account op_src_ROMALab	Unlimited	Apr 23, 2021 16:13:25 GMT+00:00	Created By ROMALab Team	Authorize Modify Delete

3. Configura la delegación.

Figura 3-3 Configuración de información básica

★ Agency Name

★ Agency Type Account Delegate another HUAWEI CLOUD account to perform operations on your resources.
 Cloud service Delegate a cloud service to access your resources in other cloud services.

★ Cloud Service

★ Validity Period

Description
0/255

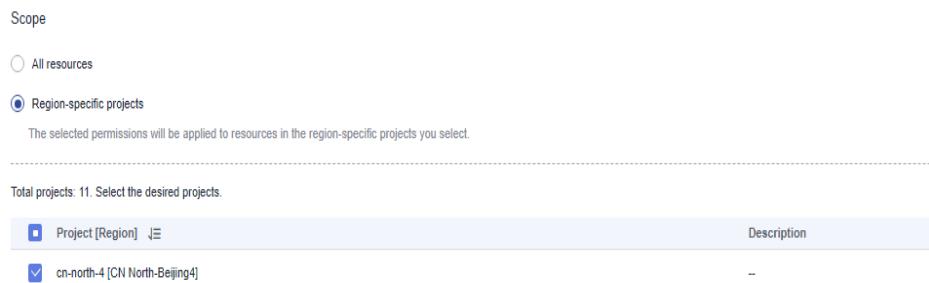
Next **Cancel**

- En **Agency Name**, escriba **serverless-trust**.
 - Para **Agency Type**, seleccione **Cloud service**.
 - En **Cloud Service**, seleccione **FunctionGraph**.
 - Para **Período de validez**, seleccione **Ilimitado**.
 - **Description**: Introduzca la descripción.
4. Haga clic en **Next**. En la página mostrada, busque los permisos que se agregarán en el cuadro de búsqueda de la derecha y seleccione los permisos. El permiso **Tenant Administrator** se utiliza como ejemplo.

Figura 3-4 Selección de políticas**Tabla 3-3** Ejemplo de permisos de agencia

Nombre de la política	Escenario
Tenant Administrator	Administrador para todos los servicios en la nube excepto IAM. Este usuario puede realizar cualquier operación en todos los recursos en la nube de la empresa.

5. Haga clic en **Next** y seleccione el ámbito, por ejemplo, **Region-specific project**.

Figura 3-5 Selección de los permisos necesarios

Configuración de una delegación

1. Inicie sesión en **la consola de FunctionGraph**. En el panel de navegación, seleccione **Functions > Function List**.
2. Haga clic en la función que se va a configurar para ir a la página de detalles de la función.
3. Elija **Configuration > Permissions**, haga clic en **Create Agency** y establezca una agencia en función de los requisitos del sitio haciendo referencia a [2-5](#).

Tabla 3-4 Parámetros de configuración de la delegación

Parámetro	Descripción
Configuration Agency	Seleccione una función que ha creado.
Execution Agency	Obligatorio si selecciona Specify an exclusive agency for function execution .

NOTA

- Para garantizar un rendimiento óptimo, seleccione **Specify an exclusive agency for function execution** y establecer diferentes delegaciones para la configuración y ejecución de funciones. También puede utilizar ninguna delegación o especificar la misma delegación para ambos fines. [Figura 3-6](#) muestra las opciones de la delegación.

[Figura 3-6](#) Delegaciones de establecimiento

Permissions [?](#)



- Configuration Agency:** por ejemplo, para crear activadores del Servicio de ingestión de datos (DIS), primero especifique una delegación con permisos DIS. Si no se especifica una delegación de este tipo o no existe la delegación especificada, no se pueden crear activadores DIS.
- Execution Agency:** Este tipo de delegación permite obtener un token y AK/SK del contexto en el gestor de funciones para acceder a otros servicios en la nube.

4. Haga clic en **Save**.

Configuración de acciones en escenarios comunes

Antes de usar FunctionGraph [cree una delegación](#) y seleccione la acción requerida haciendo referencia a [Tabla 3-5](#).

Tabla 3-5 Acciones comunes

Escenario	Acción	Descripción
Uso de una imagen personalizada	SWR Admin	<p>SWR Admin: administrador que tiene todos los permisos para el servicio Repositorio SoftWare para Contenedor (SWR).</p> <p>Para obtener más información sobre cómo crear una imagen personalizada, consulte Implementación de una función mediante una imagen de contenedor.</p>
Montaje de un sistema de archivos SFS Turbo	SFS Administrator o Tenant administrator	<p>SFS Administrator: administrador que tiene todos los permisos para el servicio SFS.</p> <p>Tenant administrator: administrador de todos los servicios en la nube excepto IAM. Este usuario puede realizar cualquier operación en todos los recursos en la nube de la empresa.</p> <p>Para obtener más información sobre cómo montar un sistema de archivos SFS, consulte Montaje de un sistema de archivos SFS Turbo.</p>

Escenario	Acción	Descripción
Montaje de un directorio compartido de ECS	Tenant Guest y VPC Administrator	Tenant Guest: usuario con permisos de solo lectura para todos los servicios en la nube (excepto IAM) VPC Administrator: administrador de red La delegación debe tener al menos los permisos de inquilino invitado y administrador de VPC. Para obtener más información sobre cómo montar un directorio compartido de ECS, consulte Montaje de un directorio compartido de ECS .
Uso de un activador DIS	DIS Administrator	Administrador que tiene todos los permisos para el servicio DIS. Para obtener más información sobre cómo crear un disparador DIS, consulte Uso de un activador DIS .
Configuración del acceso de VPC entre dominios	VPC Administrator	Los usuarios con permisos de VPC Administrator pueden realizar cualquier operación en todos los recursos de nube de la VPC. Para configurar el acceso entre VPC, especifique una delegación con permisos de administración de VPC. Para obtener más información sobre cómo configurar el acceso de VPC entre dominios, consulte Configuración de VPC .
Lectura de recursos DNS	DNS ReadOnlyAccess	DNS ReadOnlyAccess: usuario con los permisos solo para ver recursos DNS. Para llamar a una API de DNS para resolver nombres de dominio privados, especifique una delegación con los permisos para leer recursos DNS. Para obtener más información sobre cómo llamar a la API DNS para resolver nombres de dominio privados, consulte ¿Cómo resuelve FunctionGraph un nombre de dominio privado de DNS?
Creación de un bucket y disparador de OBS	Tenant Administrator	Tenant administrator: administrador de todos los servicios en la nube excepto IAM. Este usuario puede realizar cualquier operación en todos los recursos en la nube de la empresa. Para obtener más información sobre cómo crear un activador OBS, consulte Uso de un activador OBS .

Modificación de una delegación

Modificar una delegación: puede modificar los permisos, el período de validez y la descripción de una delegación en la consola de IAM.

⚠️ ATENCIÓN

Después de que se modifica una delegación, se tarda unos 10 minutos para que la modificación (por ejemplo, `context.getToken`) surta efecto.

3.4 Configuración de VPC

Información general

Las funciones se pueden configurar para acceder a recursos en sus nubes privadas virtuales (Virtual Private Clouds, VPC) o para acceder a Internet mediante direcciones IP elásticas enlazadas a través de la traducción de direcciones de red de origen (SNAT).

Configuración de acceso de VPC

Para configurar el acceso de VPC para una función, realice los siguientes pasos:

1. Inicie sesión en [la consola de FunctionGraph](#). En el panel de navegación, seleccione **Functions > Function List**.
2. Haga clic en la función que se va a configurar para ir a la página de detalles de la función.
3. Seleccione **Configuration > VPC**, habilite el acceso de VPC y configure la información relacionada.

💡 NOTA

1. Para obtener más información sobre cómo crear una VPC y una subred, consulte [Creación de una VPC](#).
2. Especifique una delegación con permisos de administrador de VPC para la función. Para más detalles, consulte [Configuración de permisos de delegación](#).
3. Puede enlazar funciones de un proyecto a hasta cuatro subredes diferentes en cualquier VPC. (Cada proyecto tiene un ID de proyecto único de 32 dígitos, que se asigna cuando se crea su cuenta. Los ID de proyecto de su cuenta y del usuario de IAM son los mismos.)
4. Haga clic en **Save**.

💡 NOTA

Preguntas frecuentes:

- [Acceso a Internet desde una VPC](#)

Ejemplo

A continuación se muestra un ejemplo de configuración de acceso de VPC para conectarse a una instancia de Redis del Servicio de caché distribuida (DCS).

Paso 1 En la consola de DCS, vea la VPC y la subred de la instancia de DCS Redis. Si no hay instancias de Redis disponibles, cree una.

Paso 2 En la pestaña **Configuration** de una función, configure el acceso de VPC para la función de la siguiente manera:

- **VPC Access:** Habilite el acceso a VPC.
- **VPC:** Seleccione la VPC consultada en **Paso 1**.

- **Subnet:** Seleccione la subred consultada en [Paso 1](#).

Paso 3 Una vez completada la configuración, utilice el siguiente código para acceder a la instancia de Redis:

```
# -*- coding:utf-8 -*-
import redis

def handler(event, context):
    r = redis.StrictRedis(host='192.168.1.143', port=6379, db=0)
    r.set('product', 'FunctionGraph')
    print r.keys('*')
    print r.get('product')
```

----Fin

Acceso a Internet desde una VPC

De forma predeterminada, las funciones desplegadas en una VPC están aisladas de Internet. Si una función necesita acceder a redes internas y externas, agregue una puerta de enlace NAT pública para la VPC.

Prerrequisitos:

1. Ha creado una VPC y una subred de acuerdo con [Creación de una VPC](#).
2. Ha obtenido una dirección IP elástica según [la asignación de un EIP](#).

Procedimiento de creación de una puerta de enlace NAT pública

Paso 1 Inicie sesión en la consola de NAT Gateway y haga clic en [Comprar NAT pública Gateway](#).

Paso 2 En la página mostrada, introduzca la información de la puerta de enlace, seleccione una VPC y una subred (por ejemplo, vpc-01), y confirme y envíe la configuración para comprar NAT Gateway. Para obtener más información, consulte [Comprar un NAT Gateway público](#).

Paso 3 Haga clic en el nombre de NAT Gateway público. En la página de detalles que se muestra, haga clic en [Add an SNAT Rule](#) y haga clic en **OK**.

----Fin

3.5 Configuración del montaje en disco

Escenarios

Se pueden montar sistemas de archivos a la función para proporcionar almacenamiento de archivos escalable. La función puede entonces leer y escribir datos en los sistemas de archivos como lo haría en los sistemas de archivos locales. Cada sistema de archivos puede ser compartido por diferentes funciones e instancias. Solo necesita especificar información como sistemas de archivos y rutas de acceso a funciones.

FunctionGraph admite los siguientes tipos de sistemas de archivos:

- **SFS Turbo**

SFS Turbo admite las siguientes clases de almacenamiento: Standard, Standard-Enhanced, Performance, y Performance-Enhanced. SFS Turbo es ampliable a 320 TB y proporciona almacenamiento de archivos compartidos totalmente alojado. Cuenta con alta disponibilidad y durabilidad, y es compatible con cantidades masivas de archivos.

pequeños y aplicaciones que requieren baja latencia y altas operaciones de entrada/salida por segundo (IOPS). SFS Turbo es adecuado para sitios web de alto rendimiento, almacenamiento de registros, compresión y descompresión, DevOps, oficinas empresariales y aplicaciones en contenedores.

- **ECS**

Un directorio en un ECS se especifica como un sistema de archivos compartido (consulte [Montaje de un directorio compartido de ECS](#)) mediante el uso del servicio de sistema de archivos de red (NFS). El directorio puede montarse entonces en una función en la misma VPC que el ECS de modo que la función pueda leer y escribir datos en el directorio. Los sistemas de archivos ECS permiten la expansión dinámica de los recursos informáticos. Este tipo de sistema de archivos es adecuado para escenarios de baja demanda de servicio.

Beneficios del uso de estos sistemas de archivos:

- El espacio de ejecución de la función se puede ampliar en gran medida en comparación con **/tmp**.
- Un sistema de archivos puede ser compartido por múltiples funciones.
- Los recursos informáticos de ECS se pueden ampliar dinámicamente y la capacidad de almacenamiento de ECS existente se puede utilizar para lograr un rendimiento informático más fuerte.

 **NOTA**

Puede escribir archivos temporales en el directorio **/tmp**. El tamaño total de estos archivos no puede superar los 512 MB.

Creación de una delegación

Antes de agregar sistemas de archivos a una función, especifique una delegación con permisos para acceder a los servicios del sistema de archivos para la función.

Hay un límite en el número máximo de agencias que puede crear, y las agencias de servicios en la nube no se pueden modificar. Por lo tanto, se recomienda crear una delegación con permisos de alto nivel, por ejemplo, **Tenant Administrator**, para permitir que una función acceda a todos los recursos de la región seleccionada. Para obtener más información, consulte [Configuración de permisos de delegación](#).

Montaje de un sistema de archivos SFS Turbo

Establecimiento de una delegación

Antes de montar un sistema de archivos SFS Turbo en una función, especifique una delegación a la que se le hayan concedido permisos de **SFS Administrator** y **VPC Administrator** para la función. Si no hay delegaciones disponibles, cree una en IAM.

Configuración de acceso de VPC

Solo se puede acceder a un sistema de archivos SFS Turbo en la VPC donde se ha creado. Antes de montar un sistema de archivos de este tipo en una función, habilite el acceso de VPC para la función.

1. En la consola SFS, obtenga la información sobre la VPC y la subred donde se va a montar un sistema de archivos en su función. Para obtener más información, consulte [Gestión del sistema de archivos](#).

2. Habilite el acceso a la VPC consultando **Configuración de VPC** e introduzca la VPC y la subred obtenidas en 1.

Montaje de un sistema de archivos SFS Turbo

Los sistemas de archivos SFS Turbo se pueden montar de la misma manera que los sistemas de archivos SFS. Seleccione un sistema de archivos y establezca la ruta de acceso.

Montaje de un directorio compartido de ECS

Especificación de una delegación

Antes de montar un directorio compartido de ECS en una función, especifique una delegación a la que se le hayan concedido permisos de **Tenant Guest** y **VPC Administrator** para la función. Si no hay delegaciones disponibles, cree una en IAM.

Configuración de acceso de VPC

Antes de agregar un directorio compartido de ECS, especifique la VPC donde se implementa el ECS. Vea la información de la VPC en la página de detalles del ECS. Haga clic en el nombre de la VPC para ir a la página de detalles de la VPC y ver la subred.

Establezca la VPC y la subred adquiridas para la función.

Montaje de un directorio ECS

Introduzca un directorio compartido y una ruta de acceso a funciones.

Figura 3-7 Configuración de la ruta



Operaciones consecuentes

Una función puede leer y escribir datos en una ruta de acceso de la misma manera que en el sistema de archivos montado.

Los registros de funciones pueden persistir configurando la ruta de acceso como un subdirectorio en la ruta de acceso.

Cree una función utilizando la plantilla "Web-Server-Access-Log-Statistics" para analizar los registros del servidor web.

Figura 3-8 Plantilla de función



Creación de un directorio compartido NFS en ECS

1. Linux

- CentOS, SUSE, EulerOS, Fedora o openSUSE
 - i. Configure un repositorio YUM.
 1. Cree un archivo llamado **euleros.repo** en el directorio **/etc/yum.repos.d**. Asegúrese de que el nombre del archivo debe terminar con **.repo**.
 2. Ejecute el siguiente comando para ingresar **euleros.repo** y edite la configuración:

```
vi /etc/yum.repos.d/euleros.repo
```

La configuración de EulerOS 2.0 SP3 YUM es la siguiente:

```
[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
```

La configuración de EulerOS 2.0 SP5 YUM es la siguiente:

```
[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
```

📘 NOTA

Descripción

name: nombre del repositorio

baseurl: URL del repositorio

- Dirección de red basada en HTTP: **http://path/to/repo**
- Dirección del repositorio local: **file:///path/to/local/repo**

gpgcheck: indica si se debe habilitar la protección de privacidad de GNU (GPG) para verificar la validez y seguridad de los recursos de paquetes RPM. **0:** La comprobación de GPG está desactivada. **1:** La comprobación de GPG está activada. Si no se especifica esta opción, la comprobación GPG está activada de forma predeterminada.

3. Guarde la configuración.

4. Ejecute el siguiente comando para borrar la caché:

```
yum clean all
```

ii. Ejecute el siguiente comando para instalar nfs-utils:

```
yum install nfs-utils
```

iii. Cree un directorio compartido.

Cuando abra **/etc/exports** y necesite crear el directorio compartido **/sharedata**, agregue la siguiente configuración:

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

📘 NOTA

La configuración anterior se utiliza para compartir el directorio **/sharedata** con otros servidores de la subred **192.168.0.0/24**.

Después de ejecutar el comando anterior, ejecute el comando **exportfs -v** para ver el directorio compartido y comprobar si la configuración es correcta.

- iv. Ejecute los siguientes comandos para iniciar el servicio NFS:

```
systemctl inicio rpcbind
inicio del servicio nfs
```

- v. Cree otro directorio compartido.

Por ejemplo, para crear el directorio **/home/myself/download**, agregue la siguiente configuración a **/etc(exports**:

```
/home/myself/download 192.168.0.0/24 (rw, sync, no_root_squash)
```

Reinicie el servicio NFS.

```
service nfs restart
```

Alternativamente, ejecute el siguiente comando sin reiniciar el servicio NFS:

```
exportfs -rv
```

- vi. (Opcional) Habilite el inicio automático del servicio rpcbind.

Ejecute el siguiente comando:

```
systemctl enable rpcbind
```

– Ubuntu

- i. Ejecute los siguientes comandos para instalar nfs-kernel-server:

```
sudo apt-get update
sudo apt install nfs-kernel-server
```

- ii. Cree un directorio compartido.

Cuando abra **/etc(exports** y necesite crear el directorio compartido **/sharedata**, agregue la siguiente configuración:

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

□ NOTA

La configuración anterior se utiliza para compartir el directorio **/sharedata** con otros servidores de la subred **192.168.0.0/24**.

Después de ejecutar el comando anterior, ejecute el comando **exportfs -v** para ver el directorio compartido y comprobar si la configuración es correcta.

- iii. Reinicie el servicio NFS.

```
service nfs-kernel-server restart
```

- iv. Cree otro directorio compartido.

Por ejemplo, para crear el directorio **/home/myself/download**, agregue la siguiente configuración a **/etc(exports**:

```
/home/myself/download 192.168.0.0/24 (rw, sync, no_root_squash)
```

Reinicie el servicio NFS.

```
service nfs restart
```

Alternativamente, ejecute el siguiente comando sin reiniciar el servicio NFS:

```
exportfs -rv
```

2. Windows

1. Instale el servidor NFS.

Software de pago: haneWIN. Descargue el software en el [sitio web oficial de haneWIN](#).

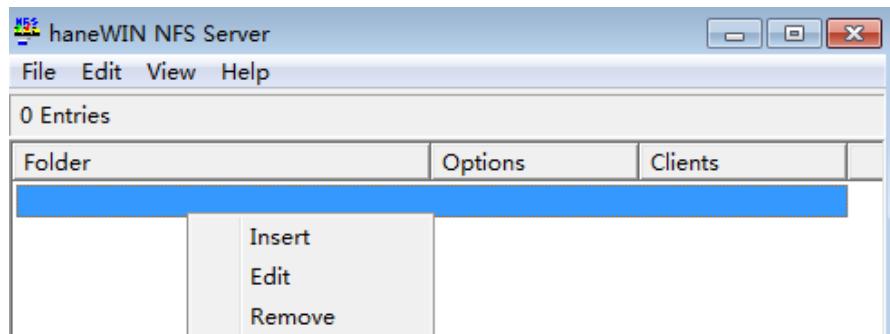
Software libre: FreeNFS y WinNFSD. Descargue el software en el [sitio web del SourceForge](#).

2. Habilite la función NFS.

- En el caso de WinNFSD, consulte [Configuración de WinNFSD](#).

- En el caso de haneWIN, realice los siguientes pasos:
 - i. Ejecute **nfscctl.exe** como administrador.
 - ii. Haga clic con el botón derecho en el área en blanco y elija **Insert** en el menú contextual.

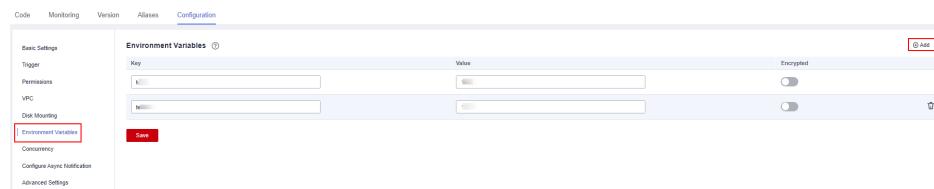
Figura 3-9 Insertar



3.6 Configuración de variables de entorno

Puede configurar la configuración de cifrado y las variables de entorno para pasar dinámicamente la configuración al código de función y las bibliotecas sin cambiar el código.

Figura 3-10 Adición de variables de entorno



Por ejemplo, para Node.js, la configuración de cifrado y los valores de las variables de entorno se pueden obtener de `getUserData(string key)` en **Context**. Para obtener más información, consulte [Desarrollo de funciones en Node.js](#).

⚠️ ADVERTENCIA

- Las variables de entorno y la configuración de cifrado son pares clave-valor definidos por el usuario que almacenan la configuración de función. Las teclas pueden contener letras, dígitos y guiones bajos (_), y deben comenzar con una letra.
- La longitud total de la clave y el valor no puede superar los 2048 caracteres.

Parámetros preestablecidos

A continuación se enumeran los parámetros preestablecidos. No configure las variables de entorno con los mismos nombres que cualquiera de estos parámetros.

RUNTIME_PROJECT_ID

RUNTIME_FUNC_NAME

RUNTIME_FUNC_VERSION
RUNTIME_PACKAGE
RUNTIME_HANDLER
RUNTIME_TIMEOUT
RUNTIME_USERDATA
RUNTIME_CPU
RUNTIME_MEMORY
RUNTIME_MAX_RESP_BODY_SIZE
RUNTIME_INITIALIZER_HANDLER
RUNTIME_INITIALIZER_TIMEOUT
RUNTIME_ROOT
RUNTIME_CODE_ROOT
RUNTIME_LOG_DIR
RUNTIME_SHARE_DIR
RUNTIME_TMP_DIR
RUNTIME_SOCKET_ADDRESS
RUNTIME_USE_REPLAYABLE
RUNTIME_FSS_REPOSITORY_ROOT
APP_SHARE_DIR

Ejemplo

Puede usar variables de entorno para configurar en qué directorio instalar archivos, dónde almacenar los resultados y cómo almacenar la configuración de conexión y registro. Estos ajustes están desacoplados de la lógica de la aplicación, por lo que no es necesario actualizar el código de función al cambiar los ajustes.

En el siguiente fragmento de código, **obs_output_bucket** es el bucket utilizado para almacenar imágenes procesadas.

```
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.getAccessKey()
    sk = context.getSecretKey()

    # download file uploaded by user from obs
    GetObject(obs_address, srcBucket, srcObjName, ak, sk)

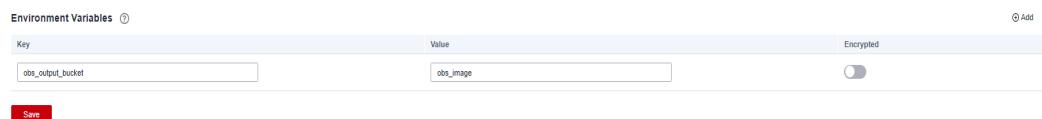
    outFile = watermark_image(srcObjName)
```

```
# Upload converted files to a new OBS bucket.
PostObject (obs_address, outputBucket, outFile, ak, sk)

return 'OK'
```

Usando la variable de entorno **obs_output_bucket**, puede configurar de forma flexible el bucket OBS utilizado para almacenar imágenes de salida.

Figura 3-11 Variables de entorno



3.7 Configuración de encriptación

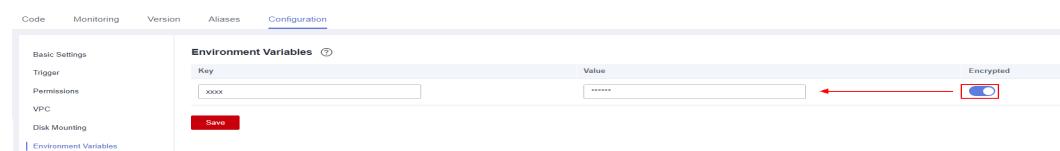
La configuración de encriptación hace referencia a las variables de entorno cifradas, que se pueden establecer en la página de detalles de la función.

Para obtener más información sobre cómo agregar y eliminar variables, consulte [Configuración de variables de entorno](#).

AVISO

Después de habilitar la encriptación, los pares clave-valor se cifrarán en la consola. También se cifrarán durante la transmisión.

Figura 3-12 Configuración de encriptación



3.8 Configuración de la notificación de ejecución asincrónica

Después de crear una función, configure la notificación de ejecución asíncrona para enviar los resultados de la ejecución al servicio de destino.

NOTA

- Si se muestra un mensaje que indica permisos insuficientes al configurar la notificación de ejecución asincrónica, agregue el permiso de **FunctionGraph Administrator**. Para obtener más información, consulte [Creación de un usuario y concesión de permisos](#).
- Establezca una delegación que permita a FunctionGraph acceder al servicio de destino.
- Para evitar la invocación ciclica, no establezca dos funciones como objetivos de ejecución asincrónica entre sí.

Procedimiento

- Paso 1** Inicie sesión en [la consola de FunctionGraph](#). En el panel de navegación, seleccione **Functions > Function List**.
- Paso 2** Haga clic en la función que se va a configurar para ir a la página de detalles de la función.
- Paso 3** Seleccione **Configuration > Configure Async Notification**. En la página mostrada, haga clic en **Edit** junto a **Asynchronous Notification Policy**.

Figura 3-13 Configuración de una política de notificación asincrónica



- Paso 4** Establezca los parámetros haciendo referencia a [Tabla 3-6](#). Por ejemplo, especifique **FunctionGraph** para el **Target Service**.

Figura 3-14 Configuración de parámetros

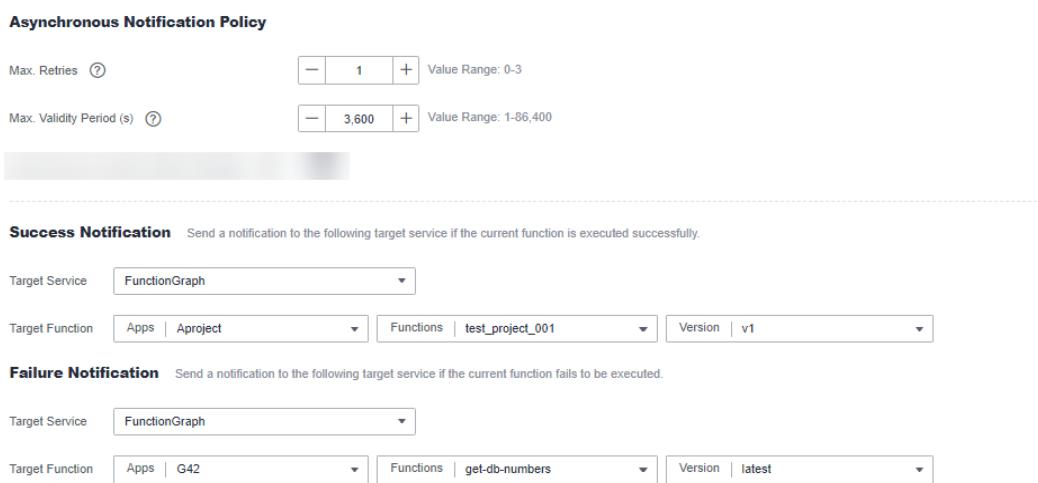


Tabla 3-6 Descripción del parámetro

Parámetro	Descripción
Asynchronous Execution Notification Policy	<ul style="list-style-type: none"> Max. Retries: número máximo de reintentos cuando falla la invocación asincrónica. Rango de valores: 0–3. Valor predeterminado: 1. Max. Período de validez (s): duración máxima de un mensaje en segundos. Rango de valores: 1–86,400.

Parámetro	Descripción
Success Notification	Target Service: al que se enviará una notificación si una función se ejecuta correctamente. <ol style="list-style-type: none"> 1. FunctionGraph 2. OBS 3. DIS 4. SMN
Failure Notification	Target Service: al que se enviará una notificación si una función no se ejecuta. <ol style="list-style-type: none"> 1. FunctionGraph 2. OBS 3. DIS 4. SMN

Paso 5 Haz clic en **OK**.

----Fin

Descripción de la configuración

Para obtener detalles sobre cómo establecer el destino para la invocación asincrónica, consulte [Tabla 3-7](#). A continuación se muestra un ejemplo:

```
{
  "timestamp": "2020-08-20T12:00:00.000Z",
  "request_context": {
    "request_id": "1167bf8c-87b0-43ab-8f5f-26b16c64f252",
    "function_urn": "urn:fss:xx-xxxx-x:xxxxxxx:function:xxxx:xxxx:latest",
    "condition": "",
    "approximate_invoke_count": 0
  },
  "request_payload": "",
  "response_context": {
    "status_code": 200,
    "function_error": ""
  },
  "response_payload": "hello world!"
}
```

Tabla 3-7 Descripción del parámetro

Parámetro	Descripción
timestamp	Hora en que comienza la invocación.
request_context	Contexto de solicitud.
request_context.request_id	ID de una solicitud de invocación asincrónica.
request_context.function_urn	URN de la función que se va a ejecutar asincrónicamente.

Parámetro	Descripción
request_context.condition	Tipo de error de invocación.
request_context.approximate_invoke_count	Número de tiempos de invocación asíncrona. Si el valor es mayor que 1, se ha reintentado la ejecución de la función.
request_payload	Carga útil de solicitud original.
response_context	Contexto de respuesta.
response_context.statusCode	Código devuelto después de la invocación de la función. Si el código no es 200, se produjo un error del sistema.
response_context.function_error	Información de error de invocación.
response_payload	Carga útil devuelta después de la ejecución de la función.

3.9 Configuración de Single-Instance Multi-Concurrency

NOTA

Esta función solo es compatible con FunctionGraph v2.

Información general

La multiconcurrencia de instancia única permite que una instancia procese simultáneamente un número especificado de solicitudes. Esta característica tiene las siguientes ventajas:

- Mejora la eficiencia de la ejecución de la función y reduce la duración de la ejecución.
- Reduce la probabilidad de arranques en frío.
- Reduce el número total de instancias y ahorra recursos.

Escenarios de la aplicación

Si las solicitudes de función tardan mucho tiempo en esperar la respuesta de los servicios descendentes, puede configurar una instancia multiconcurrencia para que las solicitudes se puedan procesar simultáneamente.

Comparación

Si una función tarda 5 segundos en ejecutarse cada vez y establece el número de solicitudes que una instancia puede procesar simultáneamente en 1, tres solicitudes deben procesarse en tres instancias, respectivamente. Por lo tanto, la duración total de la ejecución es 15s.

Cuando se establece **Max. Requests per Instance** a **5**, si se envían tres solicitudes, serán procesadas simultáneamente por una instancia. El tiempo total de ejecución es de 5s.

 NOTA

Cuando **Max. Requests per Instance** son mayores que 1, se debe considerar la competencia de recursos y la seguridad de simultaneidad.

Las solicitudes excesivas deben esperar un tiempo antes de ser procesadas. Por ejemplo, si el número de solicitudes concurrentes de una única instancia se establece en 10, el sistema no devuelve un error inmediatamente cuando el número de solicitudes excede la capacidad de procesamiento. Las solicitudes excesivas se procesarán después de un período de tiempo.

Tabla 3-8 Comparación

Artículo de comparación	Single-Instance Single-Concurrency	Single-Instance Multi-Concurrency
Impresión de registros	-	Para imprimir registros, Node.js Runtime utiliza la función console.info() , Python Runtime utiliza la función print() , y Java Runtime utiliza la función System.out.println() . En este modo, los ID de solicitud actuales se incluyen en el contenido del registro. Sin embargo, cuando una instancia procesa varias solicitudes simultáneamente, los ID de solicitud son incorrectos si continúa utilizando las funciones anteriores para imprimir registros. Para resolver este problema, utilice la función context.getLogger().log() en su lugar. Este método conserva los ID de solicitud por separado.
Variables compartidas	Que no participan.	La modificación de las variables compartidas causará errores. La protección de exclusión mutua es necesaria cuando se modifican variables no seguras para subprocesos durante la escritura de funciones.
Métricas de monitoreo	Realizar monitoreo basado en la situación real.	Bajo la misma carga, el número de instancias de función disminuye significativamente.
Error de control de flujo	Que no participan.	Cuando hay demasiadas solicitudes, el código de error en el cuerpo es FSS.0429 , el estado en el encabezado de respuesta es 429 , y el mensaje de error es Your request has been controlled by overload sdk, please retry later.

Configuración de Single-Instance Multi-Concurrency

1. Inicie sesión en [la consola de FunctionGraph](#). En el panel de navegación, seleccione **Functions > Function List**.
2. Haga clic en la función que se va a configurar para ir a la página de detalles de la función.
3. Seleccione **Configuration > Concurrency**.

Defina los parámetros haciendo referencia a [Tabla 3-9](#) y haga clic en **Save**.

Figura 3-15 Configuración de simultaneidad

The screenshot shows a configuration interface for 'Concurrency'. At the top, the word 'Concurrency' is in bold. Below it are two input fields: 'Max. Requests per Instance' with a value of '1' and 'Max. Instances per Function' with a value of '-1'. At the bottom is a red 'Save' button.

Tabla 3-9 Descripción

Parámetro	Descripción
Max. Requests per Instance	Número de solicitudes simultáneas admitidas por una sola instancia. Rango de valores: 1–100. Para más detalles, consulte Configuración de Single-Instance Multi-Concurrency .
Max. Instances per Function	Número máximo de instancias en las que se puede ejecutar una función. -1 (predeterminado): La función puede ejecutarse en cualquier número de instancias. 0: La función está deshabilitada. NOTA Las solicitudes que excedan la capacidad de procesamiento de las instancias serán descartadas. Los errores causados por solicitudes excesivas no se mostrarán en los registros de funciones. Puede obtener detalles de error haciendo referencia a Configuración de la notificación de ejecución asincrónica .

Restricciones de configuración

Para las funciones Java, la multiconcurrencia de instancia única tiene las siguientes restricciones:

- 128 MB de memoria: hasta 25 solicitudes simultáneas
- 256 MB de memoria: hasta 50 solicitudes simultáneas
- 512 MB o más de memoria: hasta 100 solicitudes simultáneas

3.10 Gestión de versiones

Información general

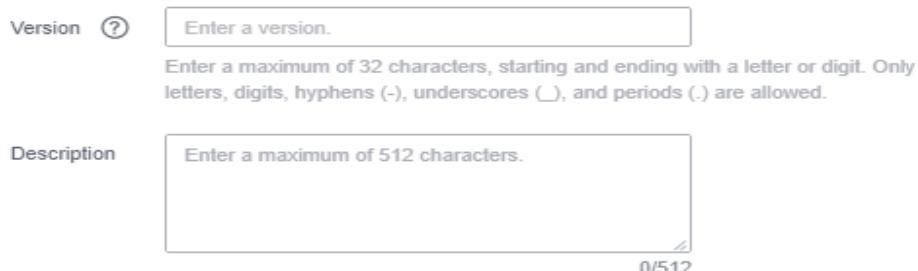
FunctionGraph le permite publicar una o más versiones a lo largo de los procesos de desarrollo, pruebas y producción para administrar su código de función. El código y las variables de entorno de cada versión se guardan como una instantánea. Una vez publicado el código de función, puede modificar la configuración según sea necesario.

Después de crear una función, la versión predeterminada es la última. Cada función tiene la última versión. Una vez publicado el código de función, puede modificar la configuración de la versión según sea necesario.

Publicación de una versión

1. Inicie sesión en [la consola de FunctionGraph](#). En el panel de navegación, seleccione **Functions > Function List**.
2. Haga clic en la función que se va a configurar para ir a la página de detalles de la función.
3. En la página de la ficha **Version**, haga clic en **Publish new version**.

Figura 3-16 Parámetros para publicar una nueva versión



- **Version:** Introduzca un número de versión. Si no se especifica ningún número de versión, el sistema genera automáticamente un número de versión basado en la fecha actual, por ejemplo, **v20220510-190658**.
 - **Description:** Introduzca una descripción para la versión. Este parámetro es por elección.
4. Haz clic en **OK**. El sistema publica automáticamente una versión. A continuación, se le redirigirá a la nueva versión.

NOTA

- Puede publicar hasta 10 versiones para una función.
- La última versión tiene instancias reservadas y su configuración no se puede cambiar. Para modificar la configuración, publique una nueva versión. La configuración de la versión original se sincronizará con la nueva versión. De forma predeterminada, una nueva versión no más reciente también tendrá instancias reservadas, que se facturarán en función de los precios estándar.

Eliminación de una versión

1. Inicie sesión en **la consola de FunctionGraph**. En el panel de navegación, seleccione **Functions > Function List**.
2. Haga clic en la función que se va a configurar para ir a la página de detalles de la función.
3. En la ficha **Version** de la última versión, seleccione la versión que desea eliminar.

Figura 3-17 Eliminación de una versión



NOTA

- No se puede eliminar la última versión de una función.
- Si se elimina una versión de función asociada con alias, los alias también se eliminarán.

4. Haga clic en **OK** para eliminar la versión.

ADVERTENCIA

Las versiones eliminadas no se pueden recuperar. Tenga cuidado cuando realice esta acción.

3.11 Gestión de alias

Información general

Un alias representa una versión de función específica y se puede usar para invocar esta versión. Para volver a una versión anterior, utilice el alias correspondiente para representar la versión en lugar de modificar el código de función.

Cada alias de función se puede vincular a una versión principal y a una versión adicional para el desplazamiento del tráfico.

Creación de un alias

1. Inicie sesión en **la consola de FunctionGraph**. En el panel de navegación, seleccione **Functions > Function List**.
2. Haga clic en la función que se va a configurar para ir a la página de detalles de la función.
3. En la página de la ficha **Aliases**, haga clic en **Create Alias**.

Figura 3-18 Creación de un alias

Formulario para la creación de un alias:

- Alias:** Entrada: Enter an alias. (Entrada obligatoria)
- Version:** Selección desplegable: --Select--
- Weight:** Valor: 100 %
- Traffic Shifting:** Botón de control (encendido)
- Additional Version:** Selección desplegable
- Weight:** Botones para ajustar el peso: - 0~100 + %
- Description:** Entrada: Enter a description. (0/512)

- **Alias:** Introduzca un alias.
- **Version:** seleccione una versión que se asociará con el alias.
- **Traffic Shifting:** elija si desea habilitar el cambio de tráfico. Si esta función está habilitada, puede distribuir un porcentaje específico de tráfico a la versión adicional.
- **Additional Version:** seleccione una versión adicional a asociar. La última versión no se puede utilizar como una versión adicional.
- **Weight:** Introduzca un número entero de 0 a 100.
- **Description:** Introduzca una descripción para el alias.

4. Haga clic en **OK**.

📘 NOTA

Puede crear hasta 10 alias para una función.

Modificación de un alias

1. Inicie sesión en **la consola de FunctionGraph**. En el panel de navegación, seleccione **Functions > Function List**.
2. Haga clic en la función que se va a configurar para ir a la página de detalles de la función.
3. En la ficha **Aliases** de la última versión, seleccione el alias que desea modificar.

Figura 3-19 Modificación de un alias

Name	Version	Description	Modified	Actions
test	1.0 (Weight: 100 %)	–	just now	

4. Modifique la información de alias y haga clic en **OK**.

Supresión de un alias

1. Inicie sesión en **la consola de FunctionGraph**. En el panel de navegación, seleccione **Functions > Function List**.

2. Haga clic en la función que se va a configurar para ir a la página de detalles de la función.
3. En la ficha **Aliases** de la última versión, seleccione el alias que desea eliminar.

Figura 3-20 Supresión de un alias

Name	Version	Description	Modified	Delete
test	1.0 (Weight: 100 %)	-	just now	

4. Haga clic en **OK** para eliminar la versión.

3.12 Configuración de memoria dinámica

NOTA

Esta función solo es compatible con FunctionGraph v2.

Información general

Al ejecutar una función, especifique los recursos de forma dinámica en función del tamaño de los datos para reducir los costos.

Los recursos informáticos necesarios para la transcodificación varían mucho en función de los tamaños de archivo de vídeo, los formatos de codificación y las resoluciones. Para garantizar un rendimiento óptimo, generalmente se configura una especificación de recursos de gran tamaño. En ese caso, muchos recursos se desperdiciarán en un escenario de baja resolución (por ejemplo, video corto). Para resolver el problema, el FunctionGraph admite la asignación dinámica de memoria durante la ejecución de la función. Minimiza el consumo de recursos y logra un control detallado de los recursos a menores costos.

Prerrequisitos:

Ha creado una función de acuerdo con [Creación de una función desde cero](#).

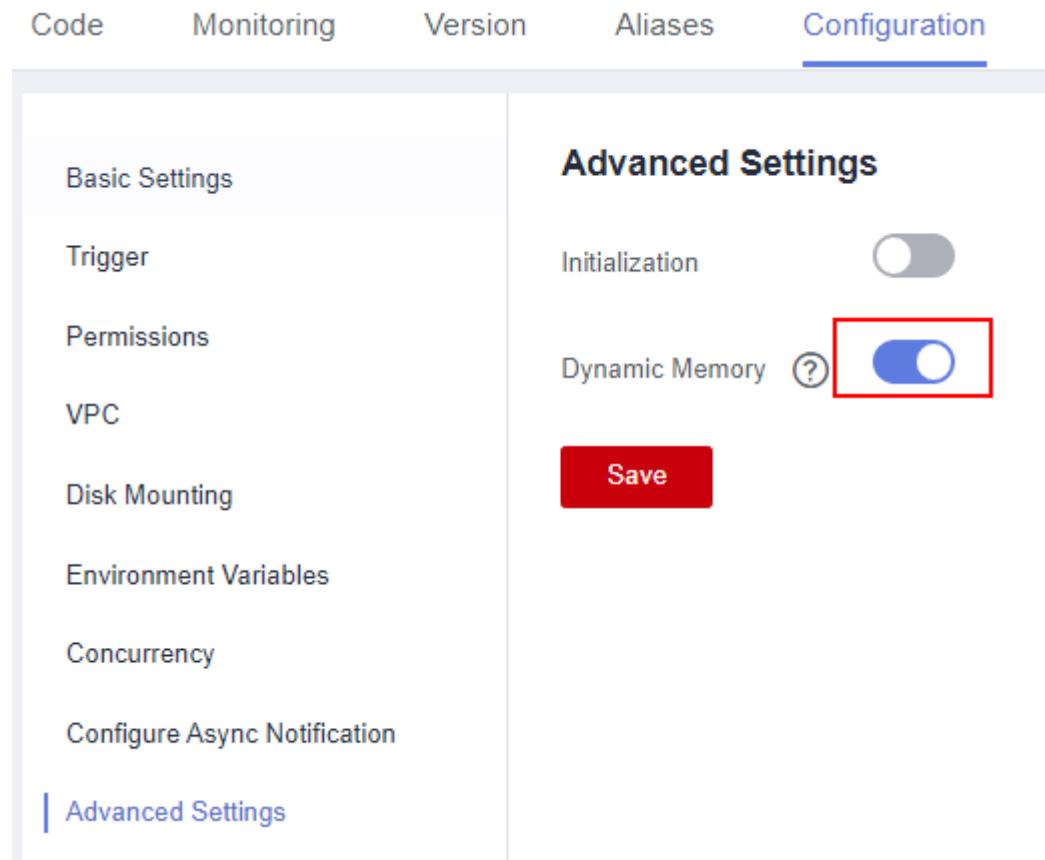
Procedimiento

Paso 1 Inicie sesión en la consola FunctionGraph y seleccione **Functions** > **Function List** en el panel de navegación y haga clic en el nombre de la función creada.

Figura 3-21 Selección de una función creada

<input type="checkbox"/> Function Name	Package Type	Runtime	Last Modified
 test	Zip	Node.js 10.16	1 minutes ago

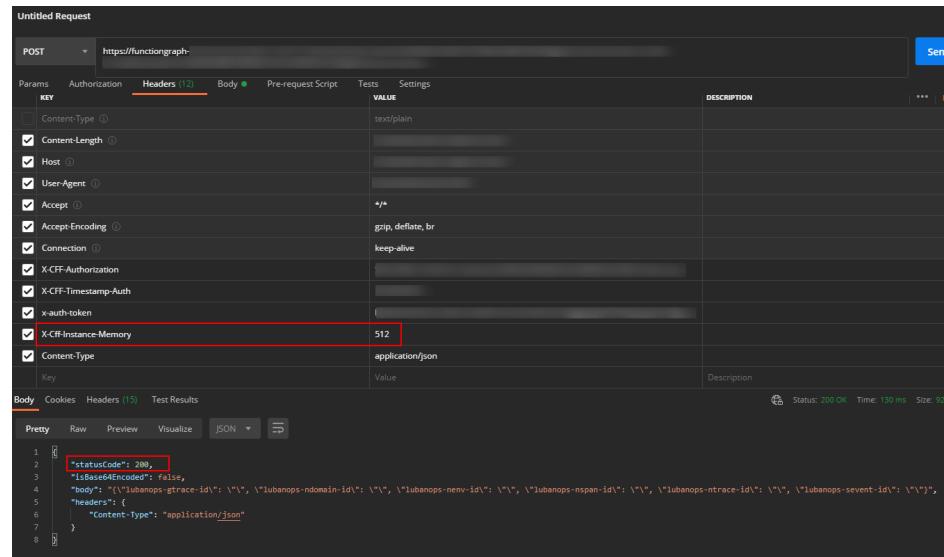
Paso 2 En la página de detalles de la función, seleccione **Configuration** > **Advanced Settings** y habilite **Dynamic Memory**.

Figura 3-22 Configuración de memoria dinámica

Paso 3 Llamar a la ejecución de la función síncrona o ejecución de la función asíncrona API, añadir **X-Cff-Instance-Memory** a la cabecera de la solicitud, y establecer el valor en 128, 256, 512, 768, 1024, 1280, 1536, 1792, 2048, 2560, 3072, 3584 o 4096

A continuación se describe cómo llamar a una API usando Postman. Agregue **X-Cff-Instance-Memory** a **Headers** y establezca el valor en 512. Si se llama correctamente a la API, se devolverá el código de error 200.

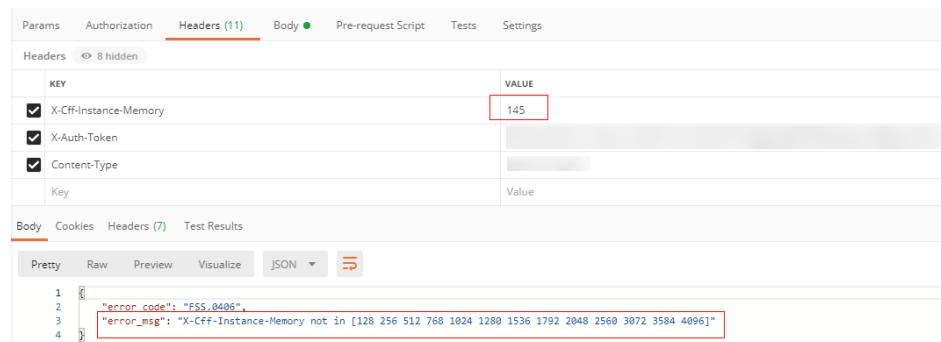
Figura 3-23 Agregar un encabezado de solicitud y llamar a la función



NOTA

- Si **Dynamic Memory** no está habilitada, el tamaño de memoria establecido cuando se crea la función se utilizará de forma predeterminada.
- Si **Dynamic Memory** está habilitada pero el valor de memoria no se ha establecido, el tamaño de memoria establecido cuando se crea la función se utilizará de forma predeterminada. Si se llama correctamente a la API, se devolverá el código de error 200.
- Si **Dynamic Memory** está habilitada pero el valor de memoria no es 128, 256, 512, 768, 1024, 1280, 1536, 1792, 2048, 2560, 3072, 3584 o 4096, código de error FSS.0406 se devolverá cuando se llame a la API. Solo necesita restablecer el valor de memoria.

Figura 3-24 Fallo de invocación



----Fin

4 Creación de activadores

4.1 Gestión de activadores

Habilitar o deshabilitar un activador

Puede habilitar o deshabilitar activadores según sea necesario. **Tenga en cuenta que los activadores SMN, OBS y APIG no se pueden deshabilitar y solo se pueden eliminar.**

- Paso 1** Inicie sesión en [la consola de FunctionGraph](#). En el panel de navegación, seleccione **Functions > Function List**.
 - Paso 2** Haga clic en el nombre de la función deseada.
 - Paso 3** Seleccione **Configuration > Trigger**. En la página mostrada, busque la fila que contiene el disparador de destino y haga clic en **Disable** o **Enable**.
- Fin

Eliminación de un activador

Puede eliminar activadores que ya no se utilizarán.

- Paso 1** Inicie sesión en [la consola de FunctionGraph](#). En el panel de navegación, seleccione **Functions > Function List**.
 - Paso 2** Haga clic en el nombre de la función deseada.
 - Paso 3** Seleccione **Configuration > Trigger**. En la página mostrada, busque la fila que contiene el disparador de destino y haga clic en **Delete**.
- Fin

4.2 Uso del activador de temporizador

Esta sección describe cómo crear un activador de temporizador para invocar su función basándose en una expresión de cron o de tasa fija.

Para obtener más información sobre el origen de eventos del temporizador, consulte [Orígenes de eventos compatibles](#).

Prerrequisitos:

Ha creado una función en la consola FunctionGraph. Para más detalles, consulte [Creación de una función desde cero](#).

Creación de un activador de temporizador

Paso 1 Inicie sesión en [la consola de FunctionGraph](#). En el panel de navegación, seleccione **Functions > Function List**.

Paso 2 Haga clic en la función que se va a configurar para ir a la página de detalles de la función.

Paso 3 Seleccione **Configuration > Trigger** y haga clic en **Create Trigger**.

Figura 4-1 Creación de un activador



Paso 4 Establezca los siguientes parámetros:

- **Trigger Type:** Seleccione **Timer**.
- **Timer Name:** Ingrese un nombre del temporizador, por ejemplo, **Timer**.
- **Rule:** Establece una tasa fija o una expresión cron.
 - **Fixed rate:** La función se activa a una velocidad fija de minutos, horas o días. Puede establecer una tarifa fija de 1 a 60 minutos, de 1 a 24 horas o de 1 a 30 días.
 - **Cron expression:** La función se activa en base a una regla compleja. Por ejemplo, puede configurar una función para que se ejecute a las 08:30:00 de lunes a viernes. Para obtener más información, consulte [Apéndice: Expresiones de Cron para un activador de temporizador de función](#).
- **Enable Trigger:** Elija si desea activar el activador del temporizador.
- **Additional Information:** La información adicional que configure se pondrá en el campo **user_event** de la fuente del evento del temporizador. Para obtener más información, consulte [Orígenes de eventos compatibles](#).

Paso 5 Haga clic en **OK**.

----Fin

Consulta del resultado de la ejecución

Después de crear el activador del temporizador, la función se ejecuta cada 1 minuto. Para ver los registros de la función que ejecuta, realice los siguientes pasos:

Paso 1 Inicie sesión en [la consola de FunctionGraph](#). En el panel de navegación, seleccione **Functions > Function List**.

Paso 2 Haga clic en una función para ir a la página de detalles de la función.

Paso 3 Seleccione **Monitoring > Logs** para consultar los registros de ejecución de la función.

----Fin

4.3 Uso de un activador APIG (dedicado)

Esta sección describe cómo crear un activador APIG y llamar a una API para activar una función.

Para obtener más información sobre el origen de eventos APIG, consulte [Orígenes de eventos compatibles](#).

Prerrequisitos:

Ha creado un grupo de API, por ejemplo, APIGroup_test. Para obtener más información, consulte [Crear un grupo de API](#).

Creación de un activador APIG

- Paso 1** Inicie sesión en [la consola de FunctionGraph](#). En el panel de navegación, seleccione **Functions > Function List**.
- Paso 2** En la página **Function List**, haga clic en **Create Function** en la esquina superior derecha.
- Paso 3** Establezca los siguientes parámetros:
 - **Function Name**: Ingrese un nombre de función, por ejemplo, **apig**.
 - **Agency**: Seleccionar **Use no agency**.
 - **Enterprise Project**: seleccione **default**.
 - **Runtime**: Selecciona **Python 2.7**.
- Paso 4** Haga clic en **Create Function**.

- Paso 5** En la página de ficha **Code**, copie el siguiente código en la ventana de código y haga clic en **Deploy**.

```
# -*- 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
    }
```

- Paso 6** Seleccione **Configuration > Trigger** y haga clic en **Create Trigger**.

Figura 4-2 Creación de un activador



- Paso 7** Configuración de la información del activador.

Tabla 4-1 Información de activador

Parámetro	Descripción
Trigger Type	Seleccione API Gateway (APIG) .
Instance	Seleccione una instancia. Si no hay ninguna instancia disponible, haga clic en Create Instance .
API Name	Introduce un nombre de API, por ejemplo, API_apig .
API Group	Un grupo de API es una colección de API. Puede administrar las API por grupo de API. Seleccione APIGroup_test .
Environment	Una API se puede llamar en diferentes entornos, como entornos de producción, prueba y desarrollo. API Gateway admite la gestión del entorno, lo que le permite definir diferentes rutas de solicitud para una API en diferentes entornos. Para asegurarse de que se puede llamar a la API, seleccione RELEASE .
Security Authentication	Hay tres modos de autenticación: <ul style="list-style-type: none"> ● App: Autenticación de AppKey y AppSecret. Este modo es de alta seguridad y se recomienda. Para obtener más información, consulte Autenticación de aplicaciones. ● IAM: Autenticación de IAM. Este modo solo otorga permisos de acceso a los usuarios de IAM y es de seguridad media. Para obtener más información, consulte Autenticación de IAM. ● None: No hay autenticación. Este modo otorga permisos de acceso a todos los usuarios. Seleccione None .
Protocol	Hay dos tipos de protocolos: <ul style="list-style-type: none"> ● HTTP ● HTTPS Seleccione HTTPS .
Timeout (ms)	Ingrese 5000 .

Paso 8 Haga clic en **OK**.

 **NOTA**

1. URL: dirección del activador APIG.
2. Después de crear el activador APIG, se genera una API denominada **API_apig** en la consola API Gateway. Puede hacer clic en el nombre de la API en la lista de activadores para ir a la consola API Gateway.

----Fin

Invocar la función

Paso 1 Introduzca la URL del activador APIG en la barra de direcciones de un navegador y pulse **Enter**.

Paso 2 Después de ejecutar la función, compruebe el resultado de la ejecución, como se muestra en [Figura 4-3](#).

Figura 4-3 Resultado devuelto



NOTA

1. La entrada para la invocación de API Gateway proviene de una plantilla de evento proporcionada por la función. Para más detalles, consulte [Tabla 5-2](#).
2. La respuesta de función para la invocación de API Gateway está encapsulada y debe contener `body(String)`, `statusCode(int)`, `headers(Map)`, y `isBase64Encoded(boolean)`.

----Fin

Consulta del resultado de la ejecución

Paso 1 Inicie sesión en [la consola de FunctionGraph](#). En el panel de navegación, seleccione **Functions > Function List**.

Paso 2 Haga clic en una función para ir a la página de detalles de la función.

Paso 3 Seleccione **Monitoring > Logs** para consultar los registros de ejecución de la función.

----Fin

4.4 Uso de un activador OBS

Esta sección describe cómo crear un activador OBS y cargar un paquete de imagen en un bucket OBS especificado para activar una función.

Para obtener más información sobre el origen de eventos del OBS, consulte [Orígenes de eventos compatibles](#).

Prerrequisitos:

Antes de crear un activador, asegúrese de haber preparado lo siguiente:

- Ha creado una función en la consola FunctionGraph. Para más detalles, consulte [Creación de una función desde cero](#).
- Ha creado un bucket OBS, por ejemplo, `obs_cff`. Para obtener más información, consulte [Creación de un bucket](#).

Creación de un activador OBS

- Paso 1** Inicie sesión en [la consola de FunctionGraph](#). En el panel de navegación, seleccione **Functions > Function List**.
- Paso 2** Haga clic en la función que se va a configurar para ir a la página de detalles de la función.
- Paso 3** Seleccione **Configuration > Trigger** y haga clic en **Create Trigger**.

Figura 4-4 Creación de un activador



- Paso 4** Establezca los siguientes parámetros:

- **Trigger Type:** seleccione **Object Storage Service (OBS)**.
- **Bucket Name:** Especifique el bucket OBS que se utilizará como fuente de eventos, por ejemplo, **obs-cff**.

AVISO

Los bucket no se pueden utilizar en todas las regiones. Por ejemplo, si ha creado una función en CN North-Beijing4, sólo se muestran los bucket de esta región.

- **Events:** Seleccione los eventos que activarán la función. En este ejemplo, seleccione **Put, Post, y Delete**. Cuando los archivos en el bucket obs_cff se actualizan, se cargan o se eliminan, la función se activa.
- **Event Notification Name:** especifique el nombre de la notificación de evento que enviará SMN cuando ocurra un evento.
- **Prefix:** introduzca una palabra clave para limitar las notificaciones a las de los objetos cuyos nombres comiencen por los caracteres coincidentes. Este límite se puede utilizar para filtrar los nombres de los objetos OBS.
- **Suffix:** introduzca una palabra clave para limitar las notificaciones a las relativas a los objetos cuyos nombres terminan con los caracteres coincidentes. Este límite se puede utilizar para filtrar los nombres de los objetos OBS.

- Paso 5** Haga clic en **OK**.

----Fin

Activación de una función

En la consola OBS, cargue un paquete ZIP de imagen en el bucket obs-cff. Para obtener más información, consulte [Cargar un archivo](#).

NOTA

Después de cargar el paquete ZIP en el bucket obs-cff, se activa la función **HelloWorld**.

Consulta del resultado de la ejecución

- Paso 1** Inicie sesión en [la consola de FunctionGraph](#). En el panel de navegación, seleccione **Functions > Function List**.

Paso 2 Haga clic en una función para ir a la página de detalles de la función.

Paso 3 Seleccione **Monitoring > Logs** para consultar los registros de ejecución de la función.

----Fin

4.5 Uso de un activador Kafka

Esta sección describe cómo crear un activador de Kafka y configurar un evento de Kafka para activar una función.

Después de usar un activador de Kafka, el FunctionGraph sondea periódicamente nuevos mensajes en un tema específico en una instancia de Kafka y pasa los mensajes como parámetros de entrada para invocar funciones. Para obtener más información sobre el origen de eventos del DMS for Kafka, consulte [Orígenes de eventos compatibles](#).

Prerrequisitos:

Antes de crear un activador, asegúrese de haber preparado lo siguiente:

- Ha creado una función en FunctionGraph. Para más detalles, consulte [Creación de una función desde cero](#).
- Ha habilitado el acceso de VPC para la función. Para más detalles, consulte [Configuración de VPC](#).
- Ha creado una instancia de Kafka. Para obtener más información, consulte [Compra de una instancia](#).
- Ha creado un tema bajo una instancia de Kafka. Para obtener más información, consulte [Creación de un tema](#).

Creación de un activador Kafka

Paso 1 Inicie sesión en [la consola de FunctionGraph](#). En el panel de navegación, seleccione **Functions > Function List**.

Paso 2 Haga clic en la función que se va a configurar para ir a la página de detalles de la función.

Paso 3 Seleccione **Configuration > Trigger** y haga clic en **Create Trigger**.

Figura 4-5 Creación de un activador



Paso 4 Establezca los siguientes parámetros:

- **Trigger Type:** Seleccione **Distributed Message Service for Kafka (Kafka)**.
- **Instance:** Seleccione una instancia premium de Kafka.
- **Topic:** Seleccione un tema de la instancia premium de Kafka.
- **Batch Size:** Establezca el número de mensajes que se recuperarán de un tema cada vez.
- **Username:** Introduzca el nombre de usuario de la instancia si SSL ha sido habilitado para ella.
- **Password:** Introduzca la contraseña de la instancia si SSL ha sido habilitado para ella.

Paso 5 Haga clic en **OK**.

 **NOTA**

Después de habilitar el acceso a la VPC, debe configurar los permisos de subred correspondientes para el grupo de seguridad de Kafka. Para obtener más información sobre cómo habilitar el acceso de VPC, consulte [Configuración de VPC](#).

----Fin

Configuración de un evento Kafka para activar la función

Paso 1 Inicie sesión en [la consola de FunctionGraph](#). En el panel de navegación, seleccione **Functions > Function List**.

Paso 2 Haga clic en la función que se va a configurar para ir a la página de detalles de la función.

Paso 3 En la página de detalles de la función, seleccione una versión.

Paso 4 En la página de la ficha **Code**, haga clic en **Test**. Aparece el cuadro de diálogo **Configure Test Event**.

Paso 5 Defina los parámetros descritos en [Tabla 4-2](#) y haga clic en **Save**.

Tabla 4-2 Información del evento de prueba

Parámetro	Descripción
Configure Test Event	Puede elegir crear un evento de prueba o editar uno existente. Utilice la opción predeterminada Create new test event .
Event Template	Seleccione kafka-event-template .
Event Name	El nombre del evento puede contener de 1 a 25 caracteres y debe comenzar con una letra y terminar con una letra o un dígito. Solo se permiten letras, dígitos, guiones bajos (_) y guiones (-). Por ejemplo, kafka-123test .
Event data	El sistema carga automáticamente la plantilla de eventos Kafka integrada, que se utiliza en este ejemplo sin modificaciones.

 NOTA

La plantilla de evento es la siguiente:

```
{  
  "event_version": "v1.0",  
  "event_time": 1576737962,  
  "trigger_type": "KAFKA",  
  "region": "xx-xxxx-1",  
  "records": [  
    {  
      "messages": [  
        "kafka message1",  
        "kafka message2",  
        "kafka message3",  
        "kafka message4",  
        "kafka message5"  
      ],  
      "instance_id": "81335d56-b9fe-4679-ba95-7030949cc76b",  
      "topic_id": "topic-test"  
    }]  
}
```

Paso 6 Haga clic en **Test**. Se muestra el resultado de la prueba de función.

----Fin

4.6 Uso de un activador DIS

Esta sección describe cómo crear un activador DIS para una función y configurar un evento DIS usando la plantilla de evento integrada para activar la función.

Para obtener más información sobre el origen de eventos de DIS, consulte [Orígenes de eventos compatibles](#).

Prerrequisitos:

Antes de crear un activador, asegúrese de haber preparado lo siguiente:

- Ha creado una función en la consola FunctionGraph. Para más detalles, consulte [Creación de una función desde cero](#).
- Ha creado un flujo DIS, por ejemplo, [dis-function](#). Para obtener más información, consulte [Creación de un flujo DIS](#).

Establecimiento de una delegación

Antes de crear un activador DIS, establezca una delegación para delegar FunctionGraph para acceder a DIS. Para obtener más información sobre cómo crear una agencia, consulte [Configuración de permisos de delegación](#).

Dado que no especificó una delegación al crear la función HelloWorld, especifique una primero.

Paso 1 Inicie sesión en [la consola de FunctionGraph](#). En el panel de navegación, seleccione **Functions > Function List**.

Paso 2 Haga clic en la función que se va a configurar para ir a la página de detalles de la función.

Paso 3 Seleccione **Configuration > Permissions** y cambie la delegación a **serverless-trust** creada en [Configuración de permisos de delegación](#).

Paso 4 Haga clic en **Save**.

----Fin

Creación de un activador DIS

Paso 1 Inicie sesión en [la consola de FunctionGraph](#). En el panel de navegación, seleccione **Functions > Function List**.

Paso 2 Haga clic en la función que se va a configurar para ir a la página de detalles de la función.

Paso 3 Seleccione **Configuration > Trigger** y haga clic en **Create Trigger**.

Paso 4 Establezca los siguientes parámetros:

- **Trigger Type:** seleccione **Data Ingestion Service (DIS)**.
- **Stream Name:** Seleccione un flujo DIS, por ejemplo, **dis-function**.
- **Max. Fetch Bytes:** Volumen máximo de datos que se pueden obtener en cada solicitud. Solo se recuperarán los registros más pequeños que este valor. El valor oscila entre 1 KB y 4 MB.
- **Starting Position:** especifique una posición en la secuencia especificada desde la que se iniciará la lectura de datos.
 - **TRIM_HORIZON:** Los datos se leen de los registros válidos más antiguos que se almacenan en la partición.
 - **latest:** Los datos se leen justo después del registro más reciente en la partición. Esta configuración garantiza que siempre lea los datos más recientes.
- **Pull Period:** establece un período para extraer datos de la secuencia.
- **Serial Data Processing:** Si se selecciona esta opción, el FunctionGraph extrae datos de la secuencia solo después de procesar los datos anteriores. Si esta opción no está seleccionada, el FunctionGraph extrae datos de la secuencia siempre que finalice el período de extracción.

Paso 5 Haga clic en **OK**.

----Fin

Configuración de un evento DIS para activar la función

Paso 1 Inicie sesión en [la consola de FunctionGraph](#). En el panel de navegación, seleccione **Functions > Function List**.

Paso 2 Haga clic en la función que se va a configurar para ir a la página de detalles de la función.

Paso 3 En la página de detalles de la función, seleccione una versión.

Paso 4 En la página de la ficha **Code**, haga clic en **Test**. Aparece el cuadro de diálogo **Configure Test Event**.

Paso 5 Defina los parámetros descritos en [Tabla 4-3](#) y haga clic en **Save**.

Tabla 4-3 Información del evento de prueba

Parámetro	Descripción
Configure Test Event	Puede elegir crear un evento de prueba o editar uno existente. Utilice la opción predeterminada Create new test event .
Event Template	Seleccione dis-event-template .
Event Name	El nombre del evento puede contener de 1 a 25 caracteres y debe comenzar con una letra y terminar con una letra o un dígito. Solo se permiten letras, dígitos, guiones bajos (_) y guiones (-). Por ejemplo, dis-123test .
Event data	El sistema carga automáticamente la plantilla de eventos DIS integrada, que se utiliza en este ejemplo sin modificaciones. El código de esta plantilla es el siguiente:

NOTA

La plantilla de evento es la siguiente:

```
{
  "ShardID": "shardId-0000000000",
  "Message": {
    "next_partition_cursor": "eyJnZXRJdGVyYXRvc1BhcmFtIjp7InN0cmVhbSluYW1lIjoizGlzLXN3dGVzdCIsInBhc
nRpdGlvbi1pZCI6InNoYXJkSWQtMDAwMDAwMCIsImN1cnNvcil0eXB1IjoivFJJTV9
IT1JJWk9OIiwic3RhcnRpbmctc2VxdWVuY2UtbnVtYmVyIjoicNCJ9LCJnZW5lcmF0ZVRpb
WVzdGFtcCI6MTUwOTYwNjM5MjE5MX0",
    "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"
}
```

Paso 6 Haga clic en **Test**. Se muestra el resultado de la prueba de función.

----Fin

4.7 Uso de un activador SMN

Esta sección describe cómo crear un activador SMN y publicar un mensaje para activar una función.

Para obtener más información sobre el origen de eventos del SMN, consulte [Orígenes de eventos compatibles](#).

Prerrequisitos:

- Ha creado un tema SMN, por ejemplo, **smn-test**. Para obtener más información, consulte [Creación de un tema](#).
- Ha creado una función en la consola FunctionGraph. Para más detalles, consulte [Creación de una función desde cero](#).

Creación de un activador SMN

Paso 1 Inicie sesión en [la consola de FunctionGraph](#). En el panel de navegación, seleccione **Functions > Function List**.

Paso 2 Haga clic en la función que se va a configurar para ir a la página de detalles de la función.

Paso 3 Seleccione **Configuration > Trigger** y haga clic en **Create Trigger**.

Figura 4-6 Creación de un activador



Paso 4 Establezca los siguientes parámetros:

- **Trigger Type:** Seleccione **SMN**.
- **Topic Name:** Seleccione un tema, por ejemplo, **smn-test**.

Paso 5 Haga clic en **OK**.

BOOK NOTA

Después de crear el activador SMN, se genera una suscripción para el tema correspondiente en la consola SMN.

----Fin

Publicación de un mensaje para activar la función

En la consola SMN, publique un mensaje en el tema de **smn-test**. Para obtener más información, consulte [Publicar un mensaje de texto](#).

Tabla 4-4 describe los parámetros necesarios para publicar un mensaje.

Tabla 4-4 Parámetros necesarios para publicar un mensaje

Parámetro	Descripción
Subject	Ingrese SMN-Test.
Message Format	Seleccione Text.
Message	Ingrese {"message":"hello"}.

 NOTA

Después de publicar un mensaje, la función se activa automáticamente. Para obtener más información sobre los eventos de ejemplo, consulte [Orígenes de eventos compatibles](#).

Consulta del resultado de la ejecución

- Paso 1** Inicie sesión en [la consola de FunctionGraph](#). En el panel de navegación, seleccione **Functions > Function List**.
 - Paso 2** Haga clic en una función para ir a la página de detalles de la función.
 - Paso 3** Seleccione **Monitoring > Logs** para consultar los registros de ejecución de la función.
- Fin

4.8 Uso de un activador LTS

Esta sección describe cómo crear un activador LTS para una función e invocar la función cuando ocurren eventos de registro.

Para obtener más información sobre el origen de eventos del temporizador, consulte [Orígenes de eventos compatibles](#).

Prerrequisitos:

Ha creado una función en la consola FunctionGraph. Para más detalles, consulte [Creación de una función desde cero](#).

- Ha creado un grupo de registro, por ejemplo, LogGroup1. Para obtener más información, consulte [Crear un grupo de registros](#).
- Ha creado un flujo de registro, por ejemplo, LogTopic1. Para obtener más información, consulte [Crear un flujo de registro](#).

Creación de un activador LTS

- Paso 1** Inicie sesión en [la consola de FunctionGraph](#). En el panel de navegación, seleccione **Functions > Function List**.
- Paso 2** Haga clic en la función que se va a configurar para ir a la página de detalles de la función.
- Paso 3** Seleccione **Configuration > Trigger** y haga clic en **Create Trigger**.

Figura 4-7 Creación de un activador

Paso 4 Establezca los siguientes parámetros:

- **Trigger Type:** seleccione **Log Tank Service (LTS)**.
- **Log Group:** seleccione un grupo de registro, por ejemplo, **LogGroup1**.
- **Log Topic:** seleccione un tema de registro, por ejemplo, **LogTopic1**.

Paso 5 Haga clic en **OK**.

---Fin

Configuración de un evento LTS para activar la función

NOTA

Cuando el tamaño de un mensaje de evento LTS supera los 75 KB, se dividirá en varios mensajes por 75 KB para activar la función.

Paso 1 Inicie sesión en la [consola de FunctionGraph](#). En el panel de navegación, seleccione **Functions > Function List**.

Paso 2 Haga clic en la función que se va a configurar para ir a la página de detalles de la función.

Paso 3 En la página de detalles de la función, seleccione una versión.

Paso 4 En la página de la ficha **Code**, haga clic en **Test**. Aparece el cuadro de diálogo **Configure Test Event**.

Paso 5 Defina los parámetros descritos en [Tabla 4-5](#) y haga clic en **Save**.

Tabla 4-5 Información del evento de prueba

Parámetro	Descripción
Configure Test Event	Puede elegir crear un evento de prueba o editar uno existente. Utilice la opción predeterminada Create new test event .
Event Template	Seleccione lts-event-template .
Event Name	El nombre del evento puede contener de 1 a 25 caracteres y debe comenzar con una letra y terminar con una letra o un dígito. Solo se permiten letras, dígitos, guiones bajos (_) y guiones (-). Por ejemplo, lts-123test .
Event data	El sistema carga automáticamente la plantilla de evento LTS integrada, que se utiliza en este ejemplo sin modificaciones.

NOTA

La plantilla de evento es la siguiente:

Paso 6 Haga clic en **Test**. Se muestra el resultado de la prueba de función.

-----Fin

4.9 Uso de un activador CTS

Esta sección describe cómo crear un activador CTS para una función e invocar la función en respuesta a las operaciones de recursos en la nube registradas por CTS.

Para obtener más información sobre el origen de eventos del CTS, consulte [Orígenes de eventos compatibles](#).

Prerrequisitos:

Ha creado una delegación en IAM. Para más detalles, consulte [Configuración de permisos de delegación](#).

Creación de un activador CTS

Paso 1 Inicie sesión en [la consola de FunctionGraph](#). En el panel de navegación, seleccione **Functions > Function List**.

Paso 2 En la página **Function List**, haga clic en **Create Function** en la esquina superior derecha.

Paso 3 Establezca los siguientes parámetros:

- **Function Name:** Ingrese un nombre de función, por ejemplo, **HelloWorld**.
- **Agency:** Seleccionar **Use no agency**.
- **Enterprise Project:** seleccione **default**.
- **Runtime:** Selecciona **Python 2.7**.

Paso 4 Haga clic en **Create Function**.

Paso 5 En la página de ficha **Code**, copie el siguiente código en la ventana de código y haga clic en **Deploy**.

```
# -*- 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)
```

Paso 6 Seleccione **Configuration > Trigger** y haga clic en **Create Trigger**.

Figura 4-8 Creación de un activador



Paso 7 Configuración de la información del activador.

Tabla 4-6 Información de activador

Parámetro	Descripción
Trigger Type	Selecciona Cloud Trace Service (CTS) .
Notification Name	Introduzca un nombre de notificación, por ejemplo, Test .
Tipo de servicio	Seleccione FunctionGraph .
Resource Type	Tipos de recursos admitidos por el servicio seleccionado, como activadores, instancias y funciones.
Trace Name	Operaciones que se pueden realizar en el tipo de recurso seleccionado, como crear o eliminar un activador.

Paso 8 Haga clic en **OK**.

----Fin

Configuración de un evento CTS para activar la función

Paso 1 Inicie sesión en [la consola de FunctionGraph](#). En el panel de navegación, seleccione **Functions > Function List**.

Paso 2 Haga clic en la función que se va a configurar para ir a la página de detalles de la función.

Paso 3 En la página de detalles de la función, seleccione una versión y haga clic en **Test**. Aparece el cuadro de diálogo **Configure Test Event**.

Paso 4 Defina los parámetros descritos en [Tabla 4-7](#) y haga clic en **Save**.

Tabla 4-7 Información del evento de prueba

Parámetro	Descripción
Configure Test Event	Puede elegir crear un evento de prueba o editar uno existente. Utilice la opción predeterminada Create new test event .
Event Template	Seleccione cts-event-template .
Event Name	Introduzca un nombre de evento, por ejemplo, cts-test .
Event data	El sistema carga automáticamente los datos de eventos en la plantilla de eventos CTS. Puede modificar los datos del evento según sea necesario.

Paso 5 Haga clic en **Test**. Se muestra el resultado de la prueba de función.

----Fin

4.10 Uso de un activador DDS

Esta sección describe cómo crear un activador DDS para una función e invocar la función cuando cambia una tabla de base de datos.

Una función que usa un activador DDS se activará cada vez que se actualiza una tabla de base de datos. Para obtener más información sobre el origen de eventos de DDS, consulte [Orígenes de eventos compatibles](#).

Prerrequisitos

Antes de crear un activador, asegúrese de haber preparado lo siguiente:

- Ha creado una función en la consola FunctionGraph. Para más detalles, consulte [Creación de una función desde cero](#).
- Ha habilitado el acceso de VPC para la función. Para más detalles, consulte [Configuración de VPC](#).
- Ha creado una instancia de base de datos de DDS.
- Ha creado una base de datos de DDS.

Creación de un activador de DDS

Paso 1 Inicie sesión en [la consola de FunctionGraph](#). En el panel de navegación, seleccione **Functions > Function List**.

Paso 2 Haga clic en la función que se va a configurar para ir a la página de detalles de la función.

Paso 3 Seleccione **Configuration > Trigger** y haga clic en **Create Trigger**.

Figura 4-9 Creación de un activador



Paso 4 Establezca los siguientes parámetros:

- **Trigger Type:** seleccione **Document Database Service (DDS)**.
- **DB Instance:** seleccione una instancia de base de datos de DDS.
- **Password:** Introduzca la contraseña del administrador **rwuser** de instancia de base de datos de DDS.
- **Database:** Introduzca el nombre de una base de datos. Tenga en cuenta que **admin**, **local**, y **config** son nombres de base de datos reservados y no se pueden usar aquí.
- **Collection:** introduzca el nombre de una colección de base de datos.
- **Batch Size:** Establece el número de registros que se van a leer de la base de datos a la vez.

Paso 5 Haz clic en **OK**.

 NOTA

Después de habilitar el acceso a la VPC, debe configurar los permisos de subred correspondientes para el grupo de seguridad de DDS. Para obtener más información sobre cómo habilitar el acceso de VPC, consulte [Configuración de VPC](#).

----Fin

Configuración de un evento DDS para activar la función

- Paso 1** Inicie sesión en [la consola de FunctionGraph](#). En el panel de navegación, seleccione **Functions > Function List**.
- Paso 2** Haga clic en la función que se va a configurar para ir a la página de detalles de la función.
- Paso 3** En la página de detalles de la función, seleccione una versión y haga clic en **Test**. Aparece el cuadro de diálogo **Configure Test Event**.
- Paso 4** Defina los parámetros descritos en [Tabla 4-8](#) y haga clic en **Save**.

Tabla 4-8 Información del evento de prueba

Parámetro	Descripción
Configure Test Event	Puede elegir crear un evento de prueba o editar uno existente. Utilice la opción predeterminada Create new test event .
Event Template	Seleccione dds-event-template .
Event Name	El nombre del evento puede contener de 1 a 25 caracteres y debe comenzar con una letra y terminar con una letra o un dígito. Solo se permiten letras, dígitos, guiones bajos (_), y guiones (-). Por ejemplo, dds-123test .
Event data	El sistema carga automáticamente la plantilla de eventos DDS integrada, que se utiliza en este ejemplo sin modificaciones.

NOTA

La plantilla de evento es la siguiente:

```
{
  "records": [
    {
      "event_source": "dds",
      "event_name": "insert",
      "region": "xx-xxxxxx-1",
      "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"
      }
    }
  ]
}
```

Paso 5 Haga clic en **Test**. Se muestra el resultado de la prueba de función, como se muestra en [Figura 4-10](#).

Figura 4-10 Resultado de la prueba

----Fin

4.11 Uso de un activador de GaussDB (for Mongo)

Esta sección describe cómo crear un activador de GaussDB (for Mongo) para una función.

Una función que usa un activador de GaussDB (para Mongo) se activará cada vez que se actualiza una tabla de base de datos. Para obtener más información sobre el origen de eventos de GaussDB (for Mongo), consulte [Orígenes de eventos compatibles](#).

Prerrequisitos:

Antes de crear un activador, asegúrese de haber preparado lo siguiente:

- Ha creado una función en la consola FunctionGraph. Para más detalles, consulte [Creación de una función desde cero](#).
- Ha habilitado el acceso de VPC para la función. Para más detalles, consulte [Configuración de VPC](#).
- Ha creado una instancia de GaussDB (for Mongo).

Creación de un activador de GaussDB (for Mongo)

Paso 1 Inicie sesión en [la consola de FunctionGraph](#). En el panel de navegación, seleccione **Functions > Function List**.

Paso 2 Haga clic en la función que se va a configurar para ir a la página de detalles de la función.

Paso 3 Seleccione **Configuration > Trigger** y haga clic en **Create Trigger**.

Figura 4-11 Creación de un activador



Paso 4 Establezca los siguientes parámetros:

- **Trigger Type:** seleccione **GaussDB(for Mongo)**.
- **GaussDB(for Mongo) Instance:** seleccione una instancia de GaussDB (for Mongo).
- **Password:** Introduzca la contraseña del administrador **rwuser** de instancia de GaussDB (for Mongo).
- **Database:** Introduzca el nombre de una base de datos GaussDB (for Mongo). Tenga en cuenta que **admin**, **local**, y **config** son nombres de base de datos reservados y no se pueden usar aquí.
- **Collection:** introduzca el nombre de una colección de base de datos.
- **Batch Size:** Establece el número de registros que se van a leer de la base de datos a la vez.

Paso 5 Haga clic en **OK**.

NOTA

Después de habilitar el acceso a VPC, debe configurar los permisos de subred correspondientes para el grupo de seguridad GaussDB (for Mongo). Para obtener más información sobre cómo habilitar el acceso de VPC, consulte [Configuración de VPC](#).

----Fin

Configuración de un evento GaussDB (for Mongo) para activar la función

Paso 1 Inicie sesión en [la consola de FunctionGraph](#). En el panel de navegación, seleccione **Functions > Function List**.

- Paso 2** Haga clic en la función que se va a configurar para ir a la página de detalles de la función.
- Paso 3** En la página de detalles de la función, seleccione una versión.
- Paso 4** En la página de la ficha **Code**, haga clic en **Test**. Aparece el cuadro de diálogo **Configure Test Event**.
- Paso 5** Defina los parámetros descritos en **Tabla 4-9** y haga clic en **Save**.

Tabla 4-9 Información del evento de prueba

Parámetro	Descripción
Configure Test Event	Puede elegir crear un evento de prueba o editar uno existente. Utilice la opción predeterminada Create new test event .
Event Template	Seleccione gaussmongo-event-template .
Event Name	El nombre del evento puede contener de 1 a 25 caracteres y debe comenzar con una letra y terminar con una letra o un dígito. Solo se permiten letras, dígitos, guiones bajos (_) y guiones (-). Por ejemplo, gaussmongo-123test .
Event data	El sistema carga automáticamente la plantilla de eventos GaussDB (for Mongo) incorporada, que se utiliza en este ejemplo sin modificaciones.

- Paso 6** Haga clic en **Test**. Se muestra el resultado de la prueba de función.

----Fin

4.12 Uso de un activador APIG (compartido)

Esta sección describe cómo crear un activador APIG y llamar a una API para activar una función.

Para obtener más información sobre el origen de eventos APIG, consulte [Orígenes de eventos compatibles](#).

NOTA

API Gateway ya no proporciona puertas de enlace compartidas. Solo los clientes que se habían registrado antes de que se eliminara esta función pueden seguir utilizándola.

Prerrequisitos:

Ha creado un grupo de API, por ejemplo, **APIGroup_test**. Para obtener más información, consulte [Crear un grupo de API](#).

Creación de un activador APIG

- Paso 1** Inicie sesión en la consola FunctionGraph y seleccione **Functions** > **Function List** en el panel de navegación.

Paso 2 Haga clic en **Create Function**.

Paso 3 Establezca los siguientes parámetros:

- **Function Name:** Ingrese un nombre de función, por ejemplo, **apig**.
- **Agency:** Seleccionar **Use no agency**.
- **Enterprise Project:** seleccione **default**.
- **Runtime:** Selecciona **Python 2.7**.

Paso 4 Haga clic en **Create Function**.

Paso 5 En la página de ficha **Code**, copie el siguiente código en la ventana de código y haga clic en **Deploy**.

```
# -*- 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
    }
```

Paso 6 Seleccione **Configuration > Trigger** y haga clic en **Create Trigger**.

Figura 4-12 Creación de un activador



Paso 7 Configuración de la información del activador.

Tabla 4-10 Información de activador

Parámetro	Descripción
Trigger Type	Seleccione API Gateway (APIG) .
API Name	Introduce un nombre de API, por ejemplo, API_apig .
API Group	Un grupo de API es una colección de API. Puede administrar las API por grupo de API. Seleccione APIGroup_test .
Environment	Una API se puede llamar en diferentes entornos, como entornos de producción, prueba y desarrollo. API Gateway admite la gestión del entorno, lo que le permite definir diferentes rutas de solicitud para una API en diferentes entornos. Para asegurarse de que se puede llamar a la API, seleccione RELEASE .

Parámetro	Descripción
Security Authentication	Hay tres modos de autenticación: <ul style="list-style-type: none"> ● App: Autenticación de AppKey y AppSecret. Este modo es de alta seguridad y se recomienda. Para obtener más información, consulte Autenticación de aplicaciones. ● IAM: Autenticación de IAM. Este modo solo otorga permisos de acceso a los usuarios de IAM y es de seguridad media. Para obtener más información, consulte Autenticación de IAM. ● None: No hay autenticación. Este modo otorga permisos de acceso a todos los usuarios. Seleccione None .
Protocol	Hay dos tipos de protocolos: <ul style="list-style-type: none"> ● HTTP ● HTTPS Seleccione HTTPS .
Timeout (ms)	Ingrese 5000 .

Paso 8 Haga clic en **OK**.

 **NOTA**

1. La URL del activador APIG es de **https://0ed9f61512d34982917a4f3cfe8ddd5d.apig.xxx.xxx.com/apig**.
2. Después de crear el activador APIG, se genera una API denominada **API_apig** en la consola API Gateway. Puede hacer clic en el nombre de la API en la lista de activadores para ir a la consola API Gateway.

----Fin

Invocar la función

Paso 1 Introduzca la URL del activador APIG en la barra de direcciones de un navegador y pulse **Enter**.

Paso 2 Después de ejecutar la función, se devuelve un resultado.

----Fin

4.13 Uso de un activador de APIC

Esta sección describe cómo crear un activador APIC y llamar a una API para activar una función.

Para obtener más información sobre el origen de eventos APIC, consulte [Orígenes de eventos compatibles](#).

Prerrequisitos:

Ha creado un grupo de API, por ejemplo, **APIConnect_test**. Para obtener más información, consulte [Crear un grupo de API](#).

Creación de un activador APIC

Paso 1 Inicie sesión en la consola FunctionGraph y seleccione **Functions > Function List** en el panel de navegación.

Paso 2 Haga clic en **Create Function**.

Paso 3 Establezca los siguientes parámetros:

- **Function Name:** Ingrese un nombre de función, por ejemplo, **apig**.
- **Agency:** Seleccionar **Use no agency**.
- **Enterprise Project:** seleccione **default**.
- En **Runtime**, seleccione **Node.js 10.16**.

Paso 4 Haga clic en **Create Function**.

Paso 5 Seleccione **Configuration > Trigger** y haga clic en **Create Trigger**.

Figura 4-13 Creación de un activador



Paso 6 Configuración de la información del activador.

Tabla 4-11 Información de activador

Parámetro	Descripción
Trigger Type	Selecciona API Connect (APIC) .
Instance	Seleccione una instancia. Si no hay ninguna instancia disponible, haga clic en Create Instance .
API Name	Ingrese un nombre de API, por ejemplo, API_apic .
API Group	Un grupo de API es una colección de API. Puede administrar las API por grupo de API. Ejemplo: DEFAULT .
Environment	Una API se puede llamar en diferentes entornos, como entornos de producción, prueba y desarrollo. API Gateway admite la gestión del entorno, lo que le permite definir diferentes rutas de solicitud para una API en diferentes entornos. Para asegurarse de que se puede llamar a la API, seleccione RELEASE .

Parámetro	Descripción
Security Authentication	Hay tres modos de autenticación: <ul style="list-style-type: none"> ● App: Autenticación de AppKey y AppSecret. Este modo es de alta seguridad y se recomienda. Para obtener más información, consulte Autenticación de aplicaciones. ● IAM: Autenticación de IAM. Este modo solo otorga permisos de acceso a los usuarios de IAM y es de seguridad media. Para obtener más información, consulte Autenticación de IAM. ● None: No hay autenticación. Este modo otorga permisos de acceso a todos los usuarios. Seleccione None .
Protocol	Hay dos tipos de protocolos: <ul style="list-style-type: none"> ● HTTP ● HTTPS Seleccione HTTPS .
Timeout (ms)	Ingrese 5000 .

Paso 7 Haga clic en **OK**.

 **NOTA**

Una vez creado el activador, se genera una API denominada **API_apic** en la consola API Gateway. Puede hacer clic en el nombre de la API en la lista de activadores para ir a la consola API Gateway.

----Fin

Invocar la función

Paso 1 Inicie sesión en ROMA Connect, busque la instancia seleccionada (por ejemplo, **Ac6-instance-NoDelete**) y vea la dirección IP pública.

Paso 2 Introduzca la dirección IP pública en el cuadro de direcciones del navegador.

Paso 3 Después de ejecutar la función, se devuelve un resultado.

----Fin

Consulta del resultado de la ejecución

Paso 1 Inicie sesión en la consola FunctionGraph y seleccione **Functions > Function List** en el panel de navegación.

Paso 2 Haga clic en el nombre de la función **nodejs-test**.

Paso 3 En la página de detalles de la función mostrada, haga clic en la ficha **Logs** para consultar los registros de ejecución de funciones.

Paso 4 Haga clic en **View Context** en la misma fila que un registro para ver los detalles del registro.

----Fin

4.14 Uso de un activador DMS (for RabbitMQ)

Esta sección describe cómo crear un activador DMS (for RabbitMQ) para una función. Actualmente, solo se admite el modo de intercambio fanout. Después de usar un activador DMS (for RabbitMQ), el FunctionGraph sondea periódicamente nuevos mensajes en un tema específico vinculado al intercambio de una instancia RabbitMQ y pasa los mensajes como parámetros de entrada para invocar funciones. Para obtener más información sobre el origen de eventos de DMS (for RabbitMQ), consulte [Orígenes de eventos compatibles](#).

Prerrequisitos:

- Ha creado una función en la consola FunctionGraph. Para más detalles, consulte [Creación de una función desde cero](#).
- Ha habilitado el acceso de VPC. Para más detalles, consulte [Configuración de VPC](#).
- Se ha creado una instancia de RabbitMQ. Para obtener más información, consulte [Compra de una instancia](#).
- Las reglas del grupo de seguridad de la instancia se han configurado correctamente.
 - a. En la sección **Network** de la página de ficha **Basic Information**, haga clic en el nombre del grupo de seguridad.
 - i. SSL desactivado
Para el acceso intra-VPC, se debe permitir el acceso entrante a través del puerto 5672.
Para el acceso público, se debe permitir el acceso entrante a través del puerto 15672.
 - ii. SSL habilitado
Para el acceso intra-VPC, se debe permitir el acceso entrante a través del puerto 5671.
Para el acceso público, se debe permitir el acceso entrante a través del puerto 15671.

Creación de un activador de RabbitMQ

Paso 1 Inicie sesión en [la consola de FunctionGraph](#). En el panel de navegación, seleccione **Functions > Function List**.

Paso 2 Haga clic en la función que se va a configurar para ir a la página de detalles de la función.

Paso 3 Seleccione **Configuration > Trigger** y haga clic en **Create Trigger**.

Figura 4-14 Creación de un activador



Paso 4 Establezca los siguientes parámetros:

- **Trigger Type:** Seleccione **DMS (for RabbitMQ)**.

- ***Instance:** Seleccione una instancia de RabbitMQ.
- ***Password:** Introduzca la contraseña de la instancia RabbitMQ.
- ***Exchange:** Ingrese el nombre del intercambio a utilizar.
- **VM:** Ingrese un nombre de VM personalizado.
- ***Batch Size:** Establece el número de mensajes que se van a recuperar de un tema cada vez.

Paso 5 Haga clic en **OK**.

----Fin

 **NOTA**

Después de habilitar el acceso a la VPC, debe configurar los permisos de subred correspondientes para el grupo de seguridad RabbitMQ. Para obtener más información sobre cómo habilitar el acceso de VPC, consulte [Configuración de VPC](#).

Configuración de un evento DMS (for RabbitMQ) para activar la función

- Paso 1** Inicie sesión en [la consola de FunctionGraph](#). En el panel de navegación, seleccione **Functions > Function List**.
- Paso 2** Haga clic en la función que se va a configurar para ir a la página de detalles de la función.
- Paso 3** En la página de detalles de la función, seleccione una versión.
- Paso 4** En la página de la ficha **Code**, haga clic en **Test**. Aparece el cuadro de diálogo **Configure Test Event**.
- Paso 5** Defina los parámetros descritos en [Tabla 4-12](#) y haga clic en **Save**.

Tabla 4-12 Información del evento de prueba

Parámetro	Descripción
Configure Test Event	Puede elegir crear un evento de prueba o editar uno existente. Utilice la opción predeterminada Create new test event .
Event Template	Seleccione rabbitmq-event-template .
Event Name	El nombre del evento puede contener de 1 a 25 caracteres y debe comenzar con una letra y terminar con una letra o un dígito. Solo se permiten letras, dígitos, guiones bajos (_), y guiones (-). Por ejemplo, kafka-123test .
Event data	El sistema carga automáticamente la plantilla de eventos RabbitMQ integrada, que se utiliza en este ejemplo sin modificaciones.

NOTA

La plantilla de evento es la siguiente:

```
{
  "event_version": "v1.0",
  "event_time": 1576737962,
  "trigger_type": "RABBITMQ",
  "region": "xx-xxxx-1",
  "records": [
    {
      "messages": [
        "rabbitmq message1",
        "rabbitmq message2",
        "rabbitmq message3",
        "rabbitmq message4",
        "rabbitmq message5"
      ],
      "instance_id": "81335d56-b9fe-4679-ba95-7030949cc76b",
      "exchange": "exchange-test"
    }
  ]
}
```

Paso 6 Haga clic en **Test**. Se muestra el resultado de la prueba de función.

----Fin

4.15 Apéndice: Expresiones de Cron para un activador de temporizador de función

Puede configurar una expresión cron en los siguientes formatos para un activador de temporizador de función:

- **@every** format

El formato es "@every *N* **unit**". *N* es un entero positivo. **unit** puede ser ns, μ s, ms, s, m o h. Una expresión @every significa invocar una función cada *N* unidades de tiempo, como se muestra en [Tabla 4-13](#).

Tabla 4-13 Expresiones de ejemplo

Expresiones	Significado
@every 30m	Activa una función cada 30 minutos.
@every 1.5h	Activa una función cada 1.5 horas.
@every 2h30m	Activa una función cada 2.5 horas.

- Formato estándar

El formato es "seconds minutes hours day-of-month month day-of-week". *day-of-week* es opcional. Los campos deben estar separados entre sí mediante un espacio. [Tabla 4-14](#) describe los campos en una expresión cron estándar.

Tabla 4-14 Descripción del parámetro

Parámetro	Descripción	Rango de valores	Personajes especiales permitidos
Segundos	Sí	0-59	, - * /
Minuto/s	Sí	0-59	, - * /
Horas	Sí	0-23	, - * /
Día (Day-of-month)	Sí	1-31	, - * ? /
Mes	Sí	1-12 or Jan-Dec. El valor es insensible a mayúsculas y minúsculas, como se muestra en Tabla 4-15 .	, - * /
Día de la semana (Day-of-week)	No	0-6 o Sun-Sat. El valor es insensible a mayúsculas y minúsculas, como se muestra en Tabla 4-16 . 0 significa domingo.	, - * ? /

Tabla 4-15 Descripción del valor del campo de mes

Mes	Dígitos	Abreviatura
Enero	1	Jan
Febrero	2	Feb
Marzo	3	Mar
Abril	4	Apr
Mayo	5	May
Junio	6	Jun
Julio	7	Jul
Agosto	8	Aug
Septiembre	9	Sep
Octubre	10	Oct
Noviembre	11	Nov

Mes	Dígitos	Abreviatura
Diciembre	12	Dec

Tabla 4-16 Descripción del valor del campo día de la semana

Día de la semana	Dígitos	Abreviatura
Lunes	1	Mon
Martes	2	Tue
Miércoles	3	Wed
Jueves	4	Thu
Viernes	5	Fri
Sábado	6	Sat
Domingo	0	Sun

Tabla 4-17 describe los caracteres especiales que se pueden usar en una expresión cron.

Tabla 4-17 Descripción de carácter especial

Carácter especial	Significado	Descripción
*	Se utiliza para especificar todos los valores de un campo.	* en el campo minutos significa cada minuto.
,	Se utiliza para especificar varios valores, que pueden ser discontinuos.	Por ejemplo, "Jan,Apr,Jul,Oct" o "1,4,7,10" en el campo de mes y "Sat,Sun" o "6,0" en el campo día de la semana.
-	Se utiliza para especificar un rango.	Por ejemplo, "0-3" en el campo minutos.
?	Se utiliza para especificar algo en uno de los dos campos en los que se permite el carácter, pero no el otro.	Puede especificar algo solo en el campo día del mes o día de la semana. Por ejemplo, si desea que su función se ejecute en un día particular (como el 10º) del mes, pero no le importa qué día de la semana es, luego ponga "10" en el campo day-of-month y "?" en el campo day-of-week.

Carácter especial	Significado	Descripción
/	Se utiliza para especificar incrementos. El carácter delante de la barra inclinada indica cuándo empezar, y el que está detrás de la barra inclinada representa el incremento.	Por ejemplo, "1/3" en el campo minutos significa activar la función cada 3 minutos a partir de las 00:01:00 de la hora.

Tabla 4-18 describe varias expresiones cron de ejemplo.

Tabla 4-18 Ejemplo de expresiones cron

Expresión de Cron	Ejemplo
0 15 2 * * ?	Ejecuta una función a las 02:15:00 todos los días.
0 30 8 ? * Mon-Fri	Ejecuta una función a las 08:30:00 todos los lunes a viernes.
0 45 7 1-3 * ?	Ejecuta una función a las 07:45:00 los tres primeros días de cada mes.
0 0/3 * * * Mon,Wed,Fri,Sun	Ejecuta una función cada 3 minutos todos los lunes, miércoles, viernes y domingos.
0 0/3 9-18 ? * Mon-Fri	Ejecuta una función cada 3 minutos durante las 09:00–18:00 todos los lunes a viernes.
0 0/30 * * * ?	Ejecuta una función cada 30 minutos.

5 Depuración de la función

5.1 Depuración en línea

Consideraciones

Los datos del evento se pasan al manejador de su función como una entrada. Después de la configuración, los datos del evento se conservan para su uso posterior. Cada función puede tener un máximo de 10 eventos de prueba.

Creación de un evento de prueba

- Paso 1** Inicie sesión en la consola FunctionGraph y seleccione **Functions > Function List** en el panel de navegación.
- Paso 2** Haga clic en el nombre de la función deseada.
- Paso 3** En la página de detalles de la función, seleccione una versión y haga clic en **Test**.
- Paso 4** En el cuadro de diálogo **Configure Test Event**, configure la información del evento de prueba de acuerdo con [Tabla 5-1](#). El parámetro marcado con un asterisco (*) es obligatorio.

Tabla 5-1 Información del evento de prueba

Parámetro	Descripción
Configure Test Event	Puede elegir crear un evento de prueba o editar uno existente. Utilice la opción predeterminada Create new test event .
Event Template	Si selecciona blank-template , puede crear un evento de prueba desde cero. Si selecciona una plantilla, el evento de prueba correspondiente en la plantilla se carga automáticamente. Para obtener más información sobre las plantillas de eventos, consulte Tabla 5-2 .

Parámetro	Descripción
*Event Name	El nombre del evento puede contener de 1 a 25 caracteres y debe comenzar con una letra y terminar con una letra o un dígito. Solo se permiten letras, dígitos, guiones bajos (_), y guiones (-). Por ejemplo, even-123test .
Event data	Introduzca un evento de prueba.

Tabla 5-2 Descripción de la plantilla de evento

Nombre de la plantilla	Descripción
blank-template	El evento de plantilla es {"key": "value"}, que se puede cambiar en función de los requisitos.
apig-event-template	Simula un evento API Gateway para activar tu función.
dms-event-template	Simula un evento DMS para activar su función.
dis-event-template	Simula un evento DIS para activar su función.
smn-event-template	Simula un evento SMN para activar su función.
obs-event-template	Simula un evento OBS para activar su función.
timer-event-template	Simula un evento de temporizador para activar su función.
lts-event-template	Simula un evento LTS para activar su función.
cts-event-template	Simula un evento CTS para activar su función.
dds-event-template	Simula un evento DDS para activar su función.
kafka-event-template	Simula un evento Kafka para activar su función.
rabbitmq-event-template	Simula un evento RabbitMQ para activar su función.
gaussmongo-event-template	Simula un evento GaussDB (para Mongo) para activar su función.

Nombre de la plantilla	Descripción
login-security-template	Sirve como entrada para la plantilla de función loginSecurity-realtime-analysis-python .
porn-image-analyse	Sirve como entrada para la plantilla de función de porn-image-analysis .
voice-analyse	Sirve como entrada para la plantilla de función de voice-analysis .
image-tag	Sirve como entrada para las plantillas de función image-tag y porn-image-analysis .

Paso 5 Haga clic en **Save**.

----Fin

Prueba de una función

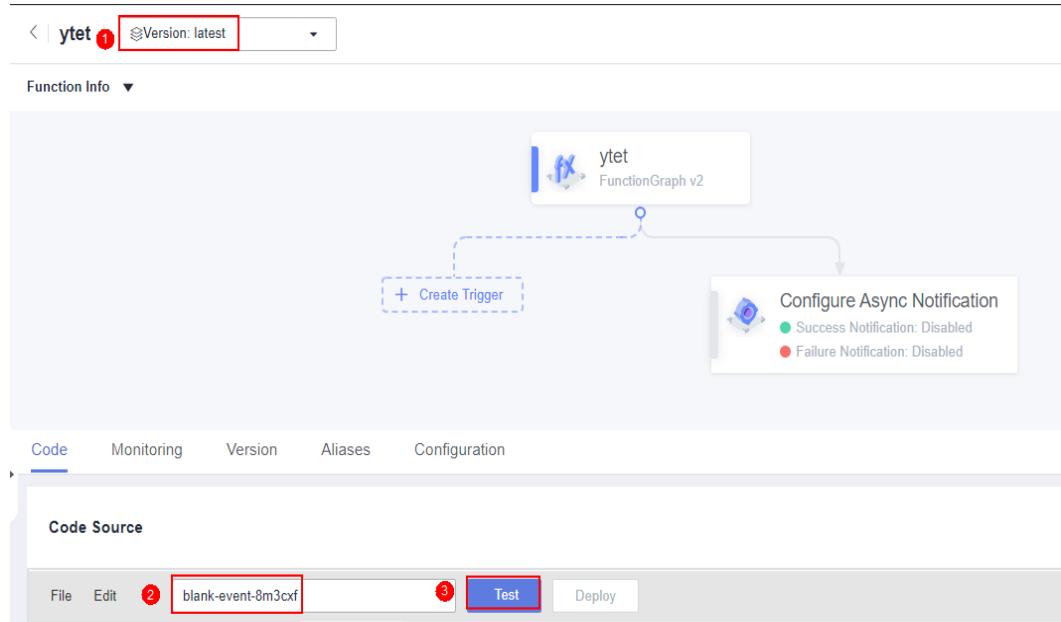
Después de crear una función, puede probarla en línea para comprobar si puede ejecutarse correctamente como se esperaba.

Paso 1 Inicie sesión en [la consola de FunctionGraph](#). En el panel de navegación, seleccione **Functions > Function List**.

Paso 2 Haga clic en el nombre de la función deseada.

Paso 3 En la página de detalles de función mostrada, seleccione una versión y un evento de prueba y haga clic en **Test**.

Figura 5-1 Selección de un evento de prueba



Paso 4 Haga clic en **Test**. Se muestra el resultado de la prueba de función.

 **NOTA**

El área **Log Output** muestra un máximo de 2 KB de registros. Para ver más registros, consulte [Gestión de registros de funciones](#).

----Fin

Modificación de un evento de prueba

Paso 1 Inicie sesión en la consola FunctionGraph y seleccione **Functions > Function List** en el panel de navegación.

Paso 2 Haga clic en un nombre de función.

Paso 3 En la página de detalles de función mostrada, seleccione una versión y haga clic en **Configure Test Event**. Aparece el cuadro de diálogo **Configure Test Event**.

Paso 4 En el cuadro de diálogo **Configure Test Event**, configure la información del evento de prueba de acuerdo con [Tabla 5-3](#).

Tabla 5-3 Información del evento de prueba

Parámetro	Descripción
Create new test event	Cree un evento de prueba.
Edit saved test event	Modificar un evento de prueba existente.
Event data	Modifique el código de evento de prueba.

Paso 5 Haga clic en **Save**.

----Fin

Eliminación de un evento de prueba

Paso 1 Inicie sesión en la consola FunctionGraph y seleccione **Functions > Function List** en el panel de navegación.

Paso 2 Haga clic en un nombre de función.

Paso 3 En la página de detalles de la función mostrada, seleccione una versión y elija **Select test event > Configure test event**.

Paso 4 En el cuadro de diálogo **Configure Test Event**, seleccione el evento de prueba que desea eliminar según [Tabla 5-4](#).

Tabla 5-4 Información del evento de prueba

Parámetro	Descripción
Configure Test Event	Seleccione Edit saved test event .
Saved Test Event	Seleccione el evento de prueba que desea eliminar.

Paso 5 Haga clic en **Delete**.

----Fin

5.2 Depuración local con VSCode

Información general

FunctionGraph de Huawei es un complemento de Visual Studio Code (VSCode) de productos sin servidor de Huawei Cloud. Con este complemento, usted puede:

- Crear rápidamente una función local.
- Ejecutar y depurar una función local e implementarla en la nube.
- Extraer la lista de funciones de la nube, llamar a funciones de la nube y cargar paquetes ZIP a la nube.

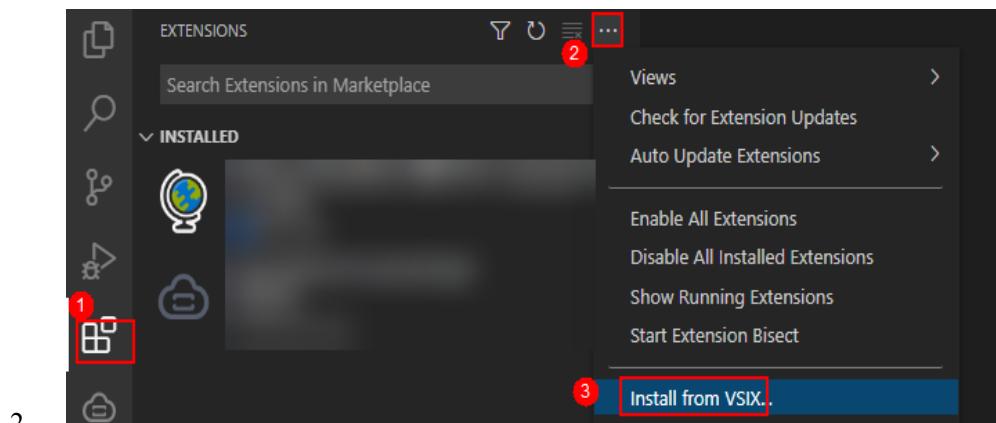
Prerrequisitos:

Usted ha descargado la herramienta **VSCode (superior de 1.54.0)** y la ha instalado.

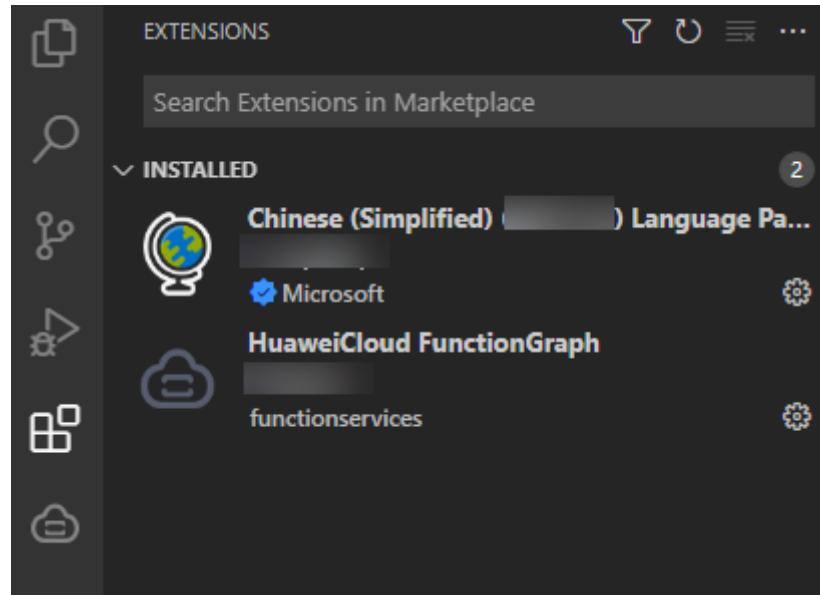
Ha descargado el complemento de **Huawei Cloud FunctionGraph**.

Instalación de complemento

1. Abra la herramienta de VSCode, elija **Extensions** en el árbol de navegación de la izquierda y elija... > **Install from VSIX** a la derecha del panel de extensión.

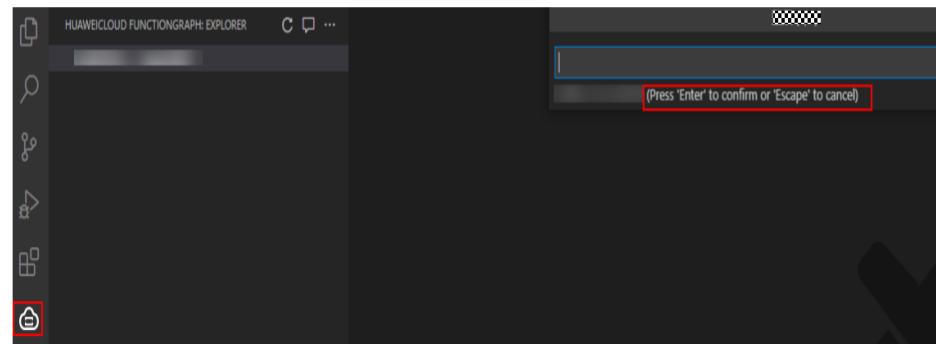


- 2.
3. En la página mostrada, importe el complemento de FunctionGraph de Huawei.
4. Una vez que la instalación se haya realizado correctamente, se mostrará el complemento.

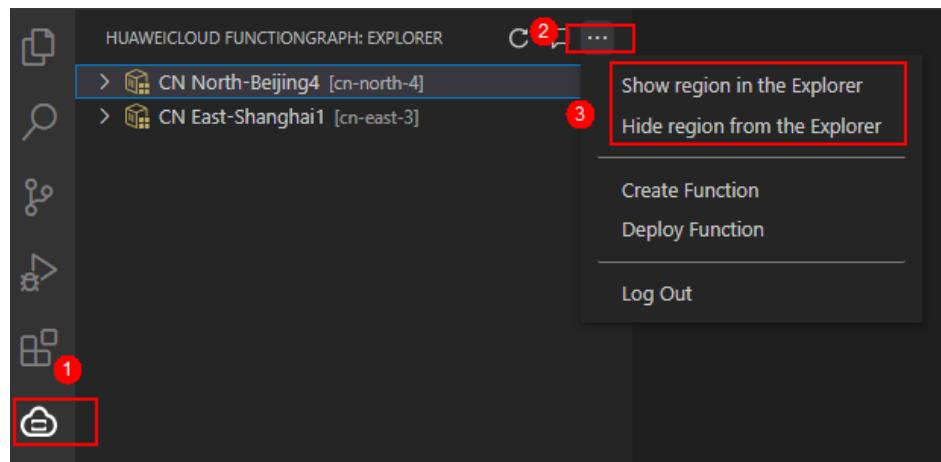


Inicio de sesión en el complemento de FunctionGraph

1. Abre el complemento FunctionGraph de Huawei a la izquierda, usa tu cuenta IAM de Huawei Cloud para iniciar sesión e ingresa la información de la cuenta según se te indique.

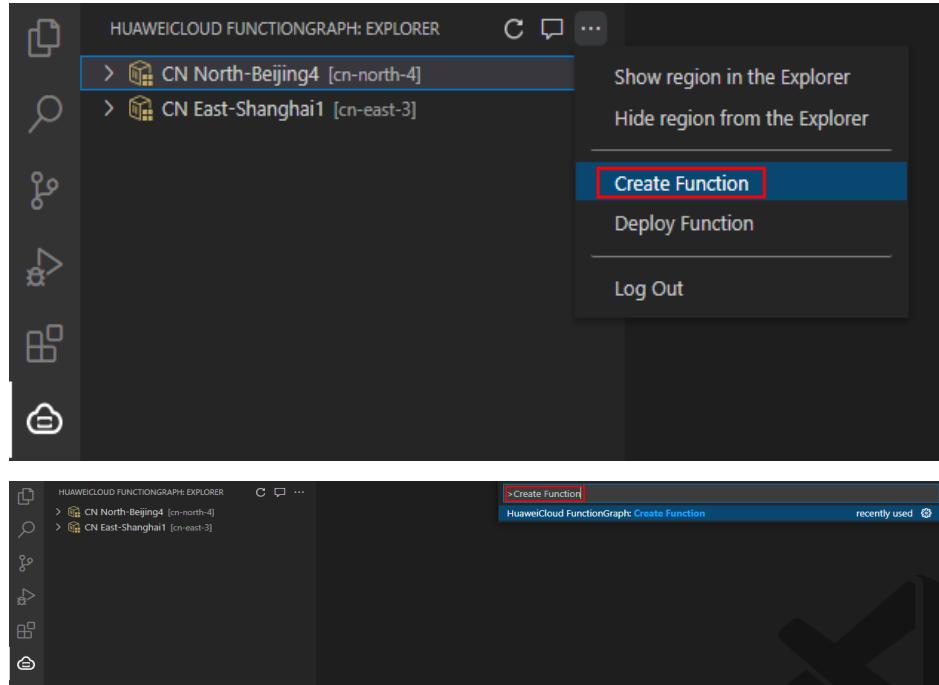


2. Seleccione una región para ver información de función.
 - **Show region in the Explorer:** Seleccione la región de destino.
 - **Hide region from the Explorer:** Ocultar regiones que no le interesan.



Creación de una función

1. En el panel de complementos, seleccione **Create Function** o pulse **Ctrl+Shift+p** para buscar el comando **Create Function**. A continuación, seleccione o introduzca el tiempo de ejecución, la plantilla, el nombre de la función y el archivo local según se le solicite.

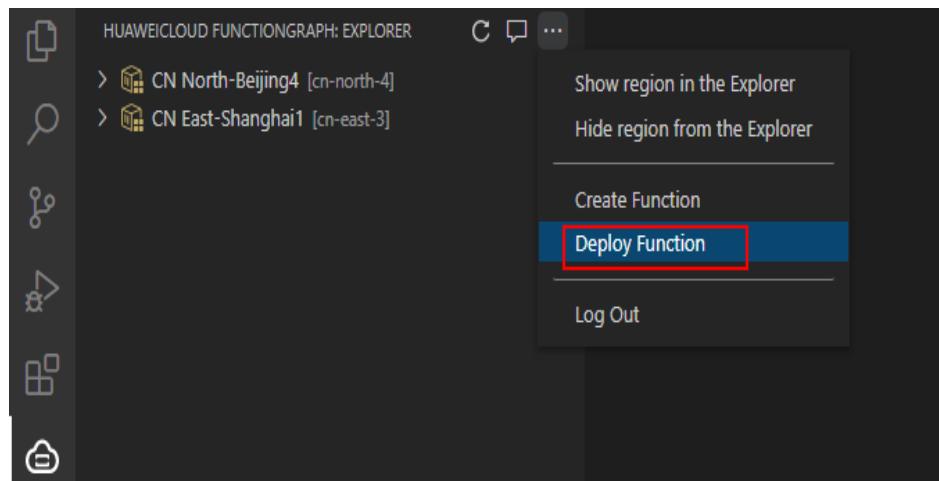


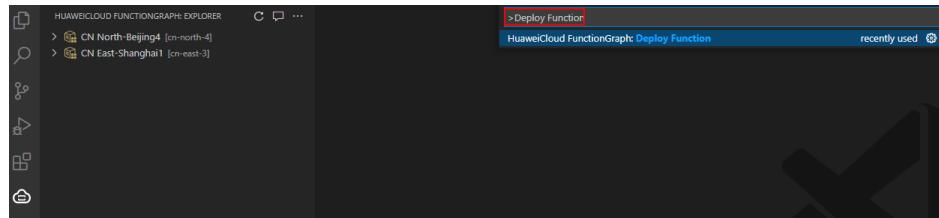
2. Despues de crear una función local, el archivo de controlador se abre automáticamente.

Despliegue de una función

En el panel de complementos, seleccione **Deploy Function** o presione **Ctrl+Shift+p** para buscar el comando **Deploy Function** y seleccione la función que se va a desplegar y la región según se le solicite.

- Si la implementación se realiza correctamente, se muestra un mensaje de éxito en la esquina inferior derecha de la página. Cambie a la región de destino para ver el resultado de la implementación.
- Si la implementación falla, vea el registro de errores en el área **Output** y rectifique el error.





Depuración local

Node.js

Cree el archivo **main.js** en la carpeta de funciones y copie el siguiente contenido en el archivo. Haga clic en **Run y Debug** a la izquierda. A continuación, haga clic en **Add Config**, seleccione Node.js y presione **F5** para depurar. (**Requisito: Se ha instalado Node.js.**)

```
const handler = require('./index'); //Path of the function handler file. Change it based on site requirements.
const event = { 'hello': 'world' }; //Test event. Modify it based on site requirements.
const context = {}; // Context Class
console.log(handler.handler(event, context));
```

Python

Cree el archivo **main.py** en la carpeta de funciones y copie el siguiente contenido en el archivo. Haga clic en **Run y Debug** a la izquierda. A continuación, haga clic en **Add Config**, seleccione Python y presione **F5** para depurar. (**Requisito: Se ha instalado Python.**)

```
import sys
import index #Path of the function handler file. Change it based on site requirements.
# The main method is used for debugging, and event is the selected debugging event.
if __name__ == '__main__':
....event = { 'hello': 'world' } # Test event. Modify it based on site requirements.
    context = ''
    content = index.handler(event, context)
....print('Returned value:')
    print(content)
```

Otras funciones

- Apertura en Portal

Haga clic derecho en una función y elija **Open in Portal** en el menú contextual. Se muestra la página de detalles de la función.

- Ejecución de una función de nube

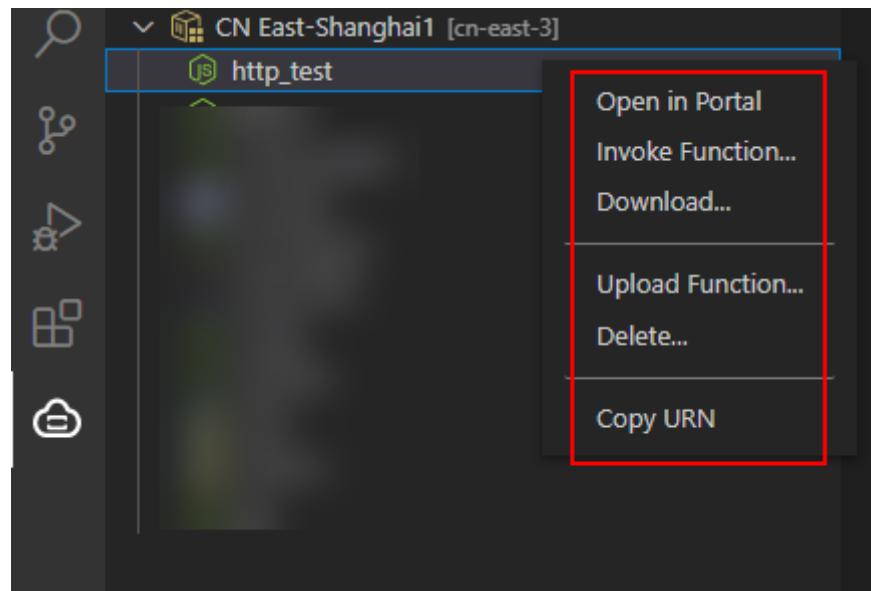
- Haga clic derecho en la función de destino y elija **Invoke Function...** desde el menú contextual.
- En el panel **Invoke Function**, seleccione el evento que desea transferir y haga clic en **Invoke**. El registro de funciones y el resultado se muestran en el área **Output**.

- Descargar una función en la nube

Haga clic derecho en la función que desea descargar y elija **Download...** desde el menú contextual. El código de función se descarga de la nube a la ruta local especificada y el archivo de controlador se abre automáticamente.

- Actualización de una función de nube
Haga clic derecho en la función de destino, elija **Cargar función...** en el menú contextual, y seleccione un paquete ZIP para cargar.
- Eliminación de una función de nube
 - Haga clic con el botón derecho en la función que desea eliminar y elija **Delete...** desde el menú contextual.
 - En el cuadro de diálogo de confirmación, haga clic en **Delete** para eliminar la función.
- Copiar URN

Haga clic con el botón derecho en la función cuyo URN necesita copiarse y elija **Copy URN** en el menú contextual.



6 Invocación de la función

6.1 Invocación síncrona

Los clientes esperan respuestas explícitas a sus solicitudes desde una función.

Las respuestas se devuelven solo después de invocar la función.

Para más detalles, consulte [Depuración en línea](#).

6.2 Invocación asincrónica

Los clientes no se preocupan por los resultados de invocación de funciones de sus solicitudes. Después de recibir una solicitud, el FunctionGraph la pone en una cola, devuelve una respuesta y procesa otras solicitudes cuando hay recursos inactivos.

Si una función es invocada asincrónicamente y es estrangulada, el FunctionGraph reintenta automáticamente el evento estrangulado, con un intervalo entre reintentos. Los eventos asincrónicos se ponen en cola antes de que se utilicen para invocar una función.

Tabla 6-1 enumera los orígenes de eventos y los modos de invocación admitidos. Para obtener más información sobre cómo gestionar activadores, consulte [Creación de activadores](#).

Tabla 6-1 Modos de invocación de función

Origen del evento	Modo de invocación
SMN	Invocación asincrónica
DMS	Invocación asincrónica
APIG	Invocación síncrona
OBS	Invocación asincrónica
DIS	Invocación asincrónica
Temporizador	Invocación asincrónica

Origen del evento	Modo de invocación
DMS for Kafka	Invocación asincrónica
GaussDB (for Mongo)	Invocación asincrónica

NOTA

Si la latencia de ejecución de la función E2E supera los 90s, se recomienda la invocación asincrónica. Si se utiliza una invocación síncrona, no se pueden recibir respuestas después de 90s debido a restricciones de pasarela.

6.3 Mecanismo de reintento

Si la invocación síncrona o asincrónica falla, haga lo siguiente:

- Invocación síncrona
Vuelva a intentarlo.
- Invocación asincrónica

Los métodos de reintento varían según los tipos de error. Para más detalles, consulte [Tabla 6-2](#).

Tabla 6-2 Métodos de reintento para diferentes tipos de error

Tipo de error	Mecanismo de reintento
El código de usuario es anormal.	No se admite el reintento.
Se produce un error interno en el servicio de función.	FunctionGraph admite reintentos para invocación asincrónica. Puede configurar el número máximo de reintentos y el período máximo de validez del mensaje haciendo referencia a Configuración de la notificación de ejecución asincrónica . Los reintentos se realizarán según el número máximo de reintentos preestablecidos y el período máximo de validez del mensaje (hasta 24 horas).

7 Monitoreo

7.1 Métricas

7.1.1 Monitoreo de funciones

FunctionGraph está interconectado con Cloud Eye, lo que le permite ver las métricas de funciones sin necesidad de ninguna configuración.

Consulta de métricas de función

FunctionGraph recopila métricas de función y muestra resultados agregados. Cambie a la versión de la función de destino antes de ver las métricas.

1. Inicie sesión en [la consola de FunctionGraph](#). En el panel de navegación, seleccione **Functions > Function List**.
2. Haga clic en la función que se va a configurar para ir a la página de detalles de la función.
3. Seleccione **Monitoring > Metrics**, seleccione un intervalo (5 minutos, 15 minutos o 1 hora) y compruebe el estado de ejecución de la función.

NOTA

Se muestran las siguientes métricas: invocaciones, errores, duración (incluidas las duraciones máxima, media y mínima) y aceleración.

Descripción de Métrica

[Tabla 7-1](#) describe las métricas de la función.

Tabla 7-1 Métricas de función

Métrica	Unidad	Descripción
Invocations	Vez	Número total de solicitudes de invocación, incluidos los errores de invocación y las invocaciones rechazadas. En caso de invocación asíncrona, el cuento comienza solo cuando se ejecuta una función en respuesta a una solicitud.
Duración	milsegundo	Duración máxima: la duración máxima que se ejecuta una función dentro de un período. Duración mínima: la duración mínima que se ejecuta una función dentro de un período. Duración media: la duración media que se ejecuta una función dentro de un período.
Errors	Vez	Número de veces que sus funciones fallan al devolver el código de error 200 . También se incluyen los errores causados por la sintaxis de la función o la ejecución.
Throttles	Vez	Número de veces que FunctionGraph rechaza sus funciones debido al límite de recursos.

7.1.2 Métricas de función

Introducción

En esta sección se describen los FunctionGraph espacios de nombres, las métricas de funciones y las dimensiones notificados a Cloud Eye. Puede ver las métricas de funciones y las alarmas usando la consola de Cloud Eye o llamando a las API.

Espacios de nombres

SYS.FunctionGraph

Métricas de función

Tabla 7-2 Monitoreo de métricas soportadas por FunctionGraph

ID de métrica	Nombre de la métrica	Descripción	Rango de valores	Objeto monitoreado	Período de monitoreo de datos brutos (minuto)
count	Invocations	Número de invocaciones de función Unidad: Vez	≥ 0 vez	Funciones	1
failcount	Errors	Número de errores de invocación Se incluyen los siguientes errores: <ul style="list-style-type: none">● Error de solicitud de función (que provoca un fallo de ejecución y devuelve el código de error 200)● Sintaxis de la función o error de ejecución Unidad: Vez	≥ 0 vez	Funciones	1
rejectcount	Throttles	Número de aceleradores de función Es decir, el número de veces que FunctionGraph acelera sus funciones debido al límite de recursos. Unidad: Vez	≥ 0 vez	Funciones	1
concurrency	Number of concurrent requests	Número de solicitudes simultáneas durante la invocación de la función. Unidad: Vez	≥ 0 vez	Funciones	1

ID de métrica	Nombre de la métrica	Descripción	Rango de valores	Objeto monitoreado	Período de monitoreo de datos brutos (minuto)
reservedinstancenum	Number of reserved instances	Número de instancias reservadas para la invocación de funciones. Unidad: Vez	≥ 0 vez	Funciones	1
duration	Average duration	Duración media de la invocación de la función Unidad: ms	≥ 0 ms	Funciones	1
maxDuration	Maximum duration	Duración máxima de la invocación de la función Unidad: ms	≥ 0 ms	Funciones	1
minDuration	Minimum duration	Duración mínima de la invocación de la función Unidad: ms	≥ 0 ms	Funciones	1

Dimensiones

Clave	Valor
package-functionname	<i>Nombre de la aplicación-nombre de la función</i> Ejemplo: default-myfunction_Python

7.1.3 Creación de una regla de alarma

Después de crear una función y un activador, puede supervisar los estados de invocación y ejecución de la función en tiempo real.

Consulta de métricas de función

FunctionGraph diferencia las métricas de una función por versión, lo que le permite consultar las métricas de una versión de función específica.

Procedimiento

Cree una regla de alarma para que una función informe de métricas a Cloud Eye para que pueda ver gráficos de supervisión y mensajes de alarma en la consola de Cloud Eye.

- Paso 1** Inicie sesión en [la consola de FunctionGraph](#). En el panel de navegación, seleccione **Functions > Function List**.
- Paso 2** Haga clic en el nombre de la función deseada.
- Paso 3** En la página de detalles de función mostrada, seleccione una versión de función o alias y elija **Monitoring > Metrics**.
- Paso 4** Haga clic en **Create Alarm Rule**.
- Paso 5** En el cuadro de diálogo **Create Alarm Rule**, establezca los parámetros de alarma y haga clic en **Siguiente**, como se muestra en [Figura 7-1](#).

Figura 7-1 Creación de una regla de alarma

The screenshot shows the 'Create Alarm Rule' dialog box. The 'Name' field is set to 'alarm-f6ew'. The 'Metric' field is set to 'Invocations'. The 'Alarm Policy' section shows 'Average' as the metric type, '5 minutes' as the period, and '1 period' as the count. A note below says: 'If you select Raw data, the monitoring period does not need to be specified.' The 'Severity' section has 'Major' selected. The 'Description' field is empty. The 'Alarm Notification' toggle is off. The bottom right corner shows '0/255'.

- Paso 6** Escriba un nombre de regla y haga clic en **OK**.

----Fin

Métricas de función

[Tabla 7-3](#) enumera las métricas de función que puede monitorear Cloud Eye.

Tabla 7-3 Métricas de función

Métrica	Mostrar nombre	Descripción	Unidad	Límite superior	Límite inferior	Umbral recomendado	Tipo de valor	Dimensión
count	Invocations	Número de invocaciones de función	Vez	-	0	-	int	package-functionname
failcount	Errors	Número de errores de invocación	Vez	-	0	-	int	package-functionname
rejectcount	Throttles	Número de aceleradores de función	Vez	-	0	-	int	package-functionname
duration	Average Duration	Duración media de la invocación de la función	ms	-	0	-	int	package-functionname
maxDuration	Maximum Duration	Duración máxima de la invocación de la función	ms	-	0	-	int	package-functionname
minDuration	Minimum Duration	Duración mínima de la invocación de la función	ms	-	0	-	int	package-functionname

7.2 Registros

7.2.1 Consulta de registros de funciones

FunctionGraph está interconectado con LTS, lo que le permite ver los registros de funciones sin la necesidad de ninguna configuración.

Consulta de registros de funciones

En la consola FunctionGraph, vea los registros de funciones de las siguientes maneras:

- Consulta de registros en la página de resultados de ejecución

Después de crear una función, pruébela y vea los registros de prueba en la página de resultados de ejecución. Para más detalles, consulte [Depuración en línea](#).

La página de resultados de ejecución muestra un máximo de 2 KB de registros. Para ver más registros de la función, vaya a la página de pestaña **Logs**.

- Consulta de registros en la página de ficha **Logs**

En la página de detalles de la función, elija **Monitoring > Logs** para consultar información de registro. Para más detalles, consulte [Gestión de registros de funciones](#).

Descarga de registros

Después de consultar los registros de una versión de función dentro de un intervalo de fechas especificado, puede descargar los registros para su análisis posterior.

NOTA

- Se puede descargar un máximo de 5000 registros a la vez. Al consultar registros, seleccione un intervalo de tiempo adecuado para evitar la pérdida de registros.
- La marca de tiempo de los registros está en UTC.

7.2.2 Gestión de registros de funciones

AVISO

FunctionGraph v1 admite la gestión de registros mediante AOM o LTS.

La versión 2 de FunctionGraph admite la gestión de registros solo con LTS.

- Uso de AOM para gestionar los registros de funciones

- Uso de LTS para gestionar los registros de funciones

Uso de AOM para gestionar los registros de funciones

Paso 1 Inicie sesión en **la consola de FunctionGraph**. En el panel de navegación, seleccione **Functions > Function List**.

Paso 2 Haga clic en el nombre de la función deseada.

Paso 3 Seleccione **Monitoring > Logs**. En la página mostrada, introduzca los criterios de búsqueda.

NOTA

- Usted puede:
 1. Introduzca una palabra clave para la búsqueda exacta. Una palabra clave es una palabra entre dos delimitadores adyacentes.
 2. Ingrese una palabra clave con comodines para la coincidencia difusa. Ejemplo: ***ROR***, **ERR***, o **ER*OR**.
 3. Introduzca una frase para la búsqueda exacta. Ejemplo: **Start to refresh alm Statistic**.
 4. Usa AND (&&) u OR (||) para combinar palabras clave para buscar. Ejemplo: **query&&logs** o **query||logs**.
- Puede seleccionar **Last 30 minutes**, **Last hour**, **Last day** y **Custom** (hasta un mes, por ejemplo, 1/4/2022 16:34:48 -1/5/2022 16:34:48).
- Puede consultar los registros por versión.



Paso 4 Haga clic en para buscar registros.

Figura 7-2 Consulta de registros

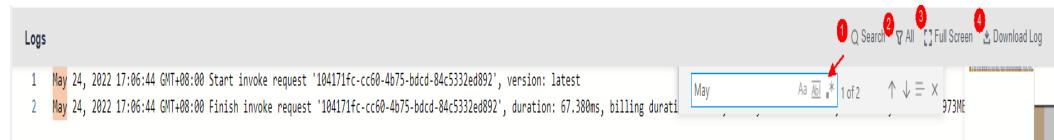


NOTA

El resultado de la consulta de registro contiene tiempo, ID de solicitud, resultado de invocación, duración, memoria y versión.

Paso 5 Realice las siguientes operaciones si es necesario:

1. Buscar registros por palabra clave.
2. Filtrar los registros por estado: **Error**, **Info**, **Error & Warning**, y **Error & Warning & Info**.
3. Ver los registros en pantalla completa.
4. **Descargar registros**.

Figura 7-3 Otras operaciones

----Fin

Uso de LTS para gestionar los registros de funciones

Puede habilitar LTS para administrar mejor los registros de funciones. Después de habilitar LTS, el FunctionGraph crea automáticamente un grupo de registros y 20 flujos de registros. Los registros de una función se almacenarán consistentemente en un flujo de registro aleatorio. Por ejemplo, si los registros de la función A se almacenan en el flujo de registro A durante su primera ejecución, sus registros posteriores siempre se almacenarán en el flujo de registro A. Un flujo de registro puede contener los registros de múltiples funciones.

NOTA

- De forma predeterminada, se crean 20 flujos de registro, que no se pueden personalizar. En la página de ficha **Logs** de la función, presione **F12** para averiguar el ID de flujo de registro de la API **query** y, a continuación, busque el ID de flujo de registro correspondiente en LTS.



- De forma predeterminada, el primer grupo de registros no se puede eliminar. Si un grupo de registro se elimina por error en la consola LTS, el FunctionGraph no detecta la eliminación y no creará automáticamente un grupo de registro aunque vuelva a crear una función. En ese caso, puede habilitar un grupo de registros llamando a la API de informes de registros de LTS. Después de que la depuración se haya realizado correctamente, vea la información en la consola LTS.

Paso 1 Habilitar LTS.

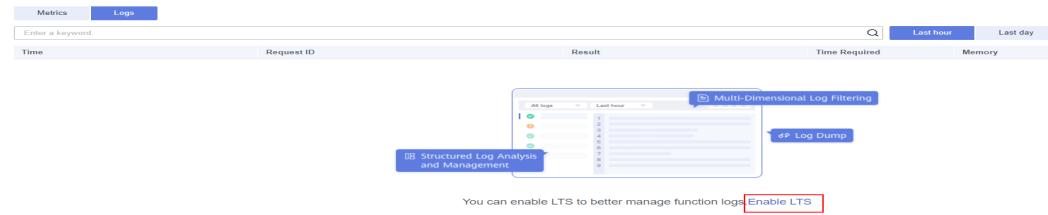
Habilitación de LTS en la versión 1 de FunctionGraph: En la página de ficha **Logs**, haga clic en **Enable LTS to Manage Function Logs**.

Figura 7-4 Habilitación de LTS para administrar registros de funciones

NOTA

En la versión 1 de FunctionGraph puede volver a AOM. En ese caso, AOM se hará cargo de LTS para gestionar los registros de funciones. Se le facturará por los registros generados sobre una base de pago por uso.

Habilitar LTS en la versión 2 de FunctionGraph: En la página de la ficha **Logs**, haga clic en **Enable LTS**. Haga clic en **OK**. LTS se ha habilitado correctamente.

Figura 7-5 Habilitación de LTS para administrar registros de funciones

NOTA

En la versión 2 de FunctionGraph solo se puede usar LTS para administrar registros de funciones.

Paso 2 Establecer criterios de filtro.

Figura 7-6 Establecer criterios de filtro

Request ID	Result	Time Required	Memory	Version
May 24, 2022 19:20:12 GMT+08:00	12884b5b-c90a-4155-968f-89a5dbc56b6	16.448ms	26.727MB	latest
May 24, 2022 17:35:23 GMT+08:00	78772a95-f4f8-407e-b022-867682f02f6c	65.924ms	26.840MB	latest
May 24, 2022 17:35:23 GMT+08:00	968196ec-9f13-47a4-a280-15b14154b3eb	22.059ms	26.840MB	latest
May 24, 2022 17:32:33 GMT+08:00	8042119b-cb1a-4f8e-9006-2b577bc76a01	103.574ms	22.406MB	latest
May 24, 2022 17:32:33 GMT+08:00	b27ee79a-4335-424e-a1f4-e0525389fa17	36.803ms	21.141MB	latest

Figura 7-7 Consultar resultados de invocación

Result	Request ID	Result	Time Required	Memory	Version
Execution successful	e011bce4-0eaa-436b-8465-b253e89a5d5d	Code error	67.244ms	26.727MB	latest
Execution failed	12884b5b-c90a-4155-968f-89a5dbc56b6	Load successful	16.448ms	26.727MB	latest
Initialization successful	78772a95-f4f8-407e-b022-867682f02f6c	Code error	65.924ms	26.840MB	latest
Initialization failed	968196ec-9f13-47a4-a280-15b14154b3eb	Load successful	22.059ms	26.840MB	latest
Load successful	8042119b-cb1a-4f8e-9006-2b577bc76a01	Code error	103.574ms	22.406MB	latest
Load failed	b27ee79a-4335-424e-a1f4-e0525389fa17	Load successful	36.803ms	21.141MB	latest
System Error					
Invocation timed out					
Memory threshold exceeded					
Disk threshold exceeded					
Code error					

Resultado de la invocación	Descripción
Ejecución exitosa	Registro impreso cuando una función se ejecuta correctamente.
Error al ejecutar	Registro impreso cuando una función no se puede ejecutar debido al tiempo de espera de la invocación, la memoria o el umbral del disco superado, o errores de código. Para ver los registros sobre el tiempo de espera de invocación, seleccione Invocation timed out en la lista desplegable. Los métodos para ver los otros tres tipos de registros son los mismos.

Resultado de la invocación	Descripción
Inicialización exitosa	Registro impreso cuando una función se inicializa correctamente.
Error de inicialización	Registro impreso cuando falla la inicialización de la función.
Se ha cargado correctamente	Registro generado cuando el tiempo de ejecución carga correctamente el archivo de función.
Error de carga	Registro generado cuando el tiempo de ejecución no puede cargar el archivo de función.
Error del sistema	error interno.
Se ha agotado el tiempo de espera de la invocación	Registro impreso cuando el período de invocación de función es más largo que el límite preestablecido.
Umbral de memoria excedido	Registro impreso cuando el tamaño de la memoria de la función excede el límite preestablecido.
Umbral de disco excedido	Registro impreso cuando el tamaño del disco excede el límite preestablecido.
Error de código	Registro impreso cuando se produce un error de código.

NOTA

- Puede buscar registros por ID de solicitud y resultado de invocación.
- Puede seleccionar **Last hour**, **Last day**, **Last 3 days**, y **Custom** (hasta un mes, por ejemplo, 1/4/2022 16:34:48–1/5/2022 16:34:48).
- Puede ir a la consola LTS para realizar más operaciones.

----Fin

Descarga de registros

AVISO

- Actualmente, los registros sólo se pueden descargar cuando se utiliza AOM para la administración de registros.
- FunctionGraph v1 le permite gestionar registros de funciones mediante AOM.
- FunctionGraph v2 le permite gestionar registros de funciones usando LTS, pero no es compatible con la descarga de registros.

Paso 1 Inicie sesión en la consola FunctionGraph y seleccione **Functions > Function List** en el panel de navegación.

Paso 2 Haga clic en el nombre de la función deseada.

Paso 3 Haga clic en **Logs** para ir a la página de registro.

Paso 4 Seleccione una versión y un intervalo de tiempo y haga clic en **Download Log**.

NOTA

Se puede descargar un máximo de 5000 registros a la vez. Al consultar registros, seleccione un intervalo de tiempo adecuado para evitar la pérdida de registros.

----Fin

8 Gestión de funciones

Información general

La función es una combinación de código, tiempo de ejecución, recursos y ajustes necesarios para lograr un propósito específico. Es la unidad mínima que puede funcionar independientemente. Los activadores pueden activar una función y programar automáticamente los recursos y entornos necesarios para lograr los resultados esperados.

Edición de código en línea

FunctionGraph le permite editar el código de función de la misma manera que administrar un proyecto. Puede crear y editar archivos y carpetas. Después de cargar un paquete ZIP de códigos, puede consultar y editar los códigos en la consola. **Tabla 8-1** describe los menús del editor en línea.

Tabla 8-1 Menús del editor en línea

Uso	Descripción
File	Crea archivos y carpetas. Puede crear archivos, crear carpetas basadas en plantillas y cerrar todos los archivos.
Edit	Edita el código de función. Puede realizar la operación de deshacer, editar o comentar.
Find	Busca y reemplaza el código.
Go to	Va a una ubicación determinada en el código. Puede ir a una línea especificada, corchetes o la siguiente línea problemática.
View	Proporciona funciones comunes. Puede ver la paleta de comandos y cambiar el tema de visualización.
Test	Prueba una función en línea y muestra el resultado de la prueba, el resumen y los registros.

 NOTA

- Java es un lenguaje compilado, que no admite la edición de código en línea. Si su función no utiliza ninguna dependencia de terceros, puede cargar un archivo JAR de función. Si su función utiliza dependencias de terceros, comprima las dependencias y el archivo JAR de función en un archivo ZIP y luego cargue el archivo ZIP.
- Si edita código en Go, comprima el archivo compilado en formato ZIP y asegúrese de que el nombre del archivo de biblioteca dinámica sea coherente con el nombre del complemento del controlador. Por ejemplo, si el nombre del archivo de biblioteca dinámica es **function.so**, establezca el controlador en **function.Handler** de acuerdo con [Tabla 8-2](#).
- Cuando cree un archivo ZIP, coloque el archivo de controlador bajo el directorio **root** para asegurarse de que su código puede ejecutarse normalmente después de ser descomprimido.
- Working directory: Una función puede leer archivos en el directorio de código. El directorio de trabajo de una función es el directorio de nivel superior del archivo de controlador. Para leer el archivo **test.conf** en el mismo nivel de directorio que el archivo manejador, utilice la ruta relativa **code/test.conf** o una ruta absoluta (valor de la variable de entorno **RUNTIME_CODE_ROOT**). Para escribir un archivo (por ejemplo, para crear o descargar un archivo), vaya al directorio **/tmp** o utilice la función de montaje del sistema de archivos proporcionada por FunctionGraph.

Modificación del código de función

Después de crear una función, la versión predeterminada es la última. Cada función tiene la última versión. Puede modificar una función basándose únicamente en su última versión.

- Paso 1** Inicie sesión en la consola FunctionGraph y seleccione **Functions > Function List** en el panel de navegación.
- Paso 2** Haga clic en un nombre de función.
- Paso 3** (Opcional) En la página de detalles de la función, seleccione la versión más reciente y haga clic en la ficha **Code**.
- Paso 4** En la página de ficha **Code**, modifique la información de código de acuerdo con [Tabla 8-2](#) y [Tabla 8-3](#).

Tabla 8-2 Información de código

Parámetro	Descripción
controlador	<ul style="list-style-type: none"> Para una función Node.js o Python, el controlador puede ser nombrado en el formato de <code>[file name].[function name]</code>, que debe contener un punto (.). Ingrese myfunction.handler, que indica que el nombre del archivo es myfunction.js (myfunction.py para una función de Python y myfunction.java para una función de Java) y el nombre de la función es manejador. Para una función Java, el controlador debe ser nombrado en el formato de <code>[nombre del paquete].[nombre del archivo].[nombre de la función]</code>, por ejemplo, com.xxxxx.exp.Myfunction.myHandler. Para una función Go, el controlador debe ser nombrado en el formato de <code>[nombre del complemento].[nombre de la función]</code>, por ejemplo, function.Handler. El nombre de la función debe comenzar con una letra mayúscula y el nombre del controlador puede contener un máximo de 128 caracteres. Para una función C#, el controlador debe ser nombrado en el formato de <code>[nombre de archivo de ensamblado .NET]::[namespace y clase de la función del controlador]::[nombre de la función del controlador]</code>, por ejemplo, HelloCsharp::Example.Hello::Handler.
Iniciador	<p>Puede habilitar la inicialización de funciones en la página de ficha Configuration. El inicializador debe ser nombrado del mismo modo que el controlador. Por ejemplo, para una función Node.js o Python, establezca un nombre de inicializador en el formato de <code>[file name].[initialization function name]</code>.</p> <p>NOTA Este parámetro no es necesario si la inicialización de la función está deshabilitada. Para obtener más información sobre cómo crear una función de inicialización, consulte Inicialización de una función.</p>
Dependencias	<p>Paquetes de software de terceros requeridos por la función. Puede administrar sus dependencias subiéndolas a FunctionGraph y administrándolas. Si las dependencias son demasiado grandes, súbelas a través de OBS. Para obtener más información, consulte Bibliotecas dependientes.</p> <p>NOTA Excepto sus dependencias privadas, el FunctionGraph proporciona algunas dependencias comunes, que puede elegir al crear una función.</p>
Modos de entrada de código	Método para introducir el código de función. Para obtener más información sobre los modos de entrada de código admitidos, consulte Tabla 8-3 .

Tabla 8-3 Modos de entrada de código

Tiempo de ejecución	Modos de entrada de código	Descripción
Node.js 6.10/ Node.js 8.10/ Node.js 10.16/ Node.js 12.13/ Node.js 14.18	Editar código en línea	Editar código en el cuadro de código. Para obtener más información, consulte Tabla 8-1 .
	Subir archivo ZIP	Haga clic en Select File y cargue un paquete de código local a FunctionGraph. El tamaño del archivo ZIP que se va a cargar no puede exceder los 50 MB. Si supera los 50 MB, cargue el archivo ZIP con un bucket OBS.
	Subir archivo desde OBS	Pegue la URL de enlace del bucket OBS que almacena un archivo ZIP de código.
Python 2.7/Python 3.6/Python 3.9	Editar código en línea	Editar código en el cuadro de código. Para obtener más información, consulte Tabla 8-1 .
	Subir archivo ZIP	Haga clic en Select File y cargue un paquete de código local a FunctionGraph. El tamaño del archivo ZIP que se va a cargar no puede exceder los 50 MB. Si supera los 50 MB, cargue el archivo ZIP con un bucket OBS.
	Subir archivo desde OBS	Pegue la URL de enlace del bucket OBS que almacena un archivo ZIP de código.
Java 8/Java 11	Subir archivo ZIP	Haga clic en Select File y cargue un paquete de código local a FunctionGraph. El tamaño del archivo ZIP que se va a cargar no puede exceder los 50 MB. Si supera los 50 MB, cargue el archivo ZIP con un bucket OBS.
	Subir archivo JAR	Haga clic en Select File y cargue un archivo JAR local a FunctionGraph. El tamaño del archivo JAR que se va a cargar no puede exceder los 50 MB. Si supera los 50 MB, conviértelo en un archivo ZIP y suba el archivo ZIP a OBS.
	Subir archivo desde OBS	Pegue la URL de enlace del bucket OBS que almacena un archivo ZIP de código.
Go 1.8/Go 1.x	Subir archivo ZIP	Haga clic en Select File y cargue un paquete de código local a FunctionGraph. El tamaño del archivo ZIP que se va a cargar no puede exceder los 50 MB. Si supera los 50 MB, cargue el archivo ZIP con un bucket OBS.
	Subir archivo desde OBS	Pegue la URL de enlace del bucket OBS que almacena un archivo ZIP de código.

Tiempo de ejecución	Modos de entrada de código	Descripción
C#(.NET Core 2.0)/ C#(.NET Core 2.1)/ C#(.NET Core 3.1)	Subir archivo ZIP	Haga clic en Select File y cargue un paquete de código local a FunctionGraph. El tamaño del archivo ZIP que se va a cargar no puede exceder los 50 MB. Si supera los 50 MB, cargue el archivo ZIP con un bucket OBS.
	Subir archivo desde OBS	Pegue la URL de enlace del bucket OBS que almacena un archivo ZIP de código.
PHP 7.3	Subir archivo ZIP	Haga clic en Select File y cargue un paquete de código local a FunctionGraph. El tamaño del archivo ZIP que se va a cargar no puede exceder los 50 MB. Si supera los 50 MB, cargue el archivo ZIP con un bucket OBS.
	Subir archivo desde OBS	Pegue la URL de enlace del bucket OBS que almacena un archivo ZIP de código.

AVISO

Si el código que se va a cargar contiene informaciones confidenciales (como contraseñas de cuenta), cífrelas para evitar fugas.

Paso 5 Haga clic en **Save**.

----Fin

Modificación de configuraciones de función

Puede definir variables para su función y modificar las configuraciones de la función.

Paso 1 Inicie sesión en la consola FunctionGraph y seleccione **Functions > Function List** en el panel de navegación.

Paso 2 Haga clic en un nombre de función.

Paso 3 (Opcional) En la página de detalles de la función, seleccione la versión más reciente y haga clic en la ficha **Configuration**.

Paso 4 Establezca los parámetros descritos en **Tabla 8-4**.

Tabla 8-4 Parámetros de configuración

Parámetro	Descripción
Description	Descripción de la función, que no puede superar los 512 caracteres. Modifique la descripción según sea necesario.

Parámetro	Descripción
Agency	<p>Se utiliza para delegar FunctionGraph para acceder a otros servicios en la nube. Por ejemplo, se requiere una delegación cuando FunctionGraph accede a servicios en la nube como OBS o SMN.</p> <p>Seleccione una delegación de la lista desplegable. Si no hay ninguna delegación en la lista desplegable, haga clic en Create Agency para crear una en la consola de IAM.</p> <p>Para más detalles, consulte Configuración de permisos de delegación.</p>
Memory (MB)	Valor: 128, 256, 512, 768, 1024, 1280, 1536, 1792, 2048, 2560, 3072, 3584 o 4096. Modifique la descripción según sea necesario.
Initialization Timeout (s)	<p>Duración máxima de la función puede ser inicializada. Establezca este parámetro si habilita la inicialización de funciones.</p> <p>El valor oscila entre 1s y 300s.</p>
Execution Timeout (s)	<p>Duración máxima de la función puede ser ejecutada.</p> <p>El valor oscila entre 3s y 900s.</p>
Max. Requests per Instance	<p>Número de solicitudes simultáneas admitidas por una sola instancia. Rango de valores: 1–100. Para más detalles, consulte Configuración de Single-Instance Multi-Concurrency.</p>
Max. Instances per Function	<p>Número máximo de instancias en las que se puede ejecutar una función. -1 (predeterminado): La función puede ejecutarse en cualquier número de instancias. 0: La función está deshabilitada.</p> <p>NOTA</p> <p>Las solicitudes que excedan la capacidad de procesamiento de las instancias serán descartadas.</p> <p>Los errores causados por solicitudes excesivas no se mostrarán en los registros de funciones. Puede obtener detalles de error haciendo referencia a Configuración de la notificación de ejecución asincrónica.</p>
VPC Access	Si el acceso de VPC está habilitado para la función, especifique una delegación con permisos de administración de VPC.
Encryption Settings	<p>Pares clave-valor definidos para almacenar información confidencial. Las teclas pueden contener letras, dígitos y guiones bajos (_), y deben comenzar con una letra. Los valores se cifrarán para su almacenamiento y no se mostrarán en texto plano.</p> <p>Haga clic en Add now? y establecer claves y valores.</p> <p>Para eliminar un par clave-valor, haga clic en Delete junto a él.</p>
Environment Variables	<p>Pares clave-valor definidos para la función. Las teclas pueden contener letras, dígitos y guiones bajos (_), y deben comenzar con una letra.</p> <p>Haga clic en Add now? y establecer claves y valores.</p> <p>Para eliminar un par clave-valor, haga clic en Delete junto a él.</p>

 NOTA

- La duración máxima para devolver un resultado es de 90s. Si tanto el tiempo de espera establecido como la duración de ejecución real de una función superan los 90s, se visualiza un mensaje que indica que la función ha expirado el tiempo de espera. Sin embargo, el backend todavía está en ejecución. Puede ver el resultado devuelto en la página de ficha **Logs**.
- Al definir variables de entorno, el FunctionGraph muestra toda la información de entrada en texto sin formato. Para garantizar la seguridad, no incluya información confidencial.

Paso 5 Haga clic en **Save**.

----Fin

Exportación de una función

Puede exportar las funciones que ha creado.

Paso 1 Inicie sesión en la consola FunctionGraph y seleccione **Functions** > **Function List** en el panel de navegación.

Paso 2 Haga clic en un nombre de función.

Paso 3 En la página de detalles de la función mostrada, seleccione **Operation** > **Export function** en la esquina superior derecha.

 NOTA

- Un usuario solo puede exportar una función a la vez.
- El paquete de recursos de función exportado no puede exceder los 50 MB.
- El nombre del paquete de recursos de función exportado tiene el formato *nombre de función + valor MD5 de código de función.zip*.
- El paquete de recursos de función exportado no incluye información de alias.

----Fin

Desactivación de una función

Las funciones deshabilitadas ya no se pueden ejecutar.

Paso 1 Inicie sesión en la consola FunctionGraph y seleccione **Functions** > **Function List** en el panel de navegación.

Paso 2 Haga clic en el nombre de la función que desea deshabilitar.

Paso 3 En la página de detalles de la función mostrada, haga clic en **Disable** en la esquina superior derecha.

Paso 4 En la página mostrada, haga clic en **Yes**. La función está desactivada.

 NOTA

- Solo se pueden desactivar las funciones de la última versión.
- Las versiones publicadas basadas en la última versión deshabilitada de una función también están deshabilitadas y nunca se pueden habilitar.
- Despues de deshabilitar una función, puede modificar su código pero no puede ejecutar la función.

----Fin

Habilitación de una función

Las funciones deshabilitadas se pueden activar de nuevo según sea necesario.

- Paso 1** Inicie sesión en la consola FunctionGraph y seleccione **Functions > Function List** en el panel de navegación.
- Paso 2** Haga clic en el nombre de la función que desea habilitar.
- Paso 3** En la página de detalles de la función mostrada, haga clic en **Enable** en la esquina superior derecha.

----Fin

Eliminación de una función

Puede eliminar las funciones no utilizadas para liberar recursos.

- Paso 1** Inicie sesión en la consola FunctionGraph y seleccione **Functions > Function List** en el panel de navegación.
- Paso 2** Haga clic en el nombre de la función que desea eliminar.
- Paso 3** Seleccione un criterio de búsqueda (nombre de función, tiempo de ejecución o nombre de aplicación) en la esquina superior derecha, introduzca una palabra clave y haga clic en  para buscar la función que desea eliminar.
- Paso 4** Haga clic en **Delete** en la misma fila que la función.
- Paso 5** En el cuadro de diálogo **Delete Function**, haga clic en **Yes** para confirmar la eliminación.

----Fin

9 Gestión de dependencias

FunctionGraph le permite gestionar la dependencias de manera unificada. Puede cargar dependencias desde una ruta local o a través de OBS si son demasiado grandes, y especificar nombres para ellas.

Para obtener más información, consulte [¿Cómo puedo crear dependencias de función?](#)

📖 NOTA

- El nombre de cada archivo en el paquete de dependencias no puede terminar con una tilde (~).
- Un paquete de dependencias puede contener hasta archivos 30,000.
- Si su dependencia privada es grande, se recomienda establecer el tiempo de ejecución de la función en un valor grande.

Creación de una dependencia

Paso 1 Inicie sesión en la consola FunctionGraph y elija **Functions > Dependencies** en el panel de navegación.

Paso 2 Haga clic en **Create Dependency**.

Paso 3 Establezca los siguientes parámetros:

- **Name:** Introduzca un nombre de dependencia.
- **Runtime:** seleccione un tiempo de ejecución.
- **Description:** Introduzca una descripción para la dependencia. Este parámetro es por elección.
- **Upload Mode:** Sube un archivo ZIP o sube un archivo desde OBS.

Paso 4 Haga clic en **OK**.

----Fin

Configuración de dependencias para una función

Paso 1 Inicie sesión en la consola FunctionGraph y seleccione **Functions > Function List** en el panel de navegación.

Paso 2 Haga clic en el nombre de la función deseada.

Paso 3 En la página de detalles de función mostrada, haga clic en la ficha **Code** y haga clic en **Add** en el área **Dependencies**.

Paso 4 En el cuadro de diálogo **Select Dependency** que se muestra, seleccione dependencias y haga clic en **OK**.

 **NOTA**

- Puede agregar un máximo de 20 dependencias para una función.
- Excepto sus dependencias privadas, el FunctionGraph proporciona algunas dependencias comunes, que puede elegir al crear una función.

Paso 5 Después de configurar las dependencias, haga clic en **Save** en la esquina superior derecha.

----Fin

Eliminación de una dependencia

Paso 1 Inicie sesión en la consola FunctionGraph y elija **Functions > Dependencies** en el panel de navegación.

Paso 2 Haga clic en **Delete** en la columna **Operation** de la dependencia de destino.

Paso 3 En el cuadro de diálogo **Delete Dependency**, haga clic en **Yes**.

 **NOTA**

Las dependencias a las que hacen referencia las funciones no se pueden eliminar.

----Fin

Bibliotecas dependientes

Bibliotecas dependientes admitidas

FunctionGraph es compatible con bibliotecas estándar y de terceros.

● Bibliotecas estándar

Si utiliza bibliotecas estándar, puede importarlas a su código o paquete en línea y subirlas a FunctionGraph.

● Bibliotecas no estándar admitidas

FunctionGraph ofrece componentes integrados de terceros enumerados en **Tabla 9-1** y **Tabla 9-2**. Puede importar estas bibliotecas al código en línea de la misma manera que importa bibliotecas estándar.

Tabla 9-1 Componentes de terceros integrados con el tiempo de ejecución de Node.js

Nombre	Uso	Versión
q	Encapsulación de método asincrónico	1.5.1
co	Control de proceso asincrónico	4.6.0
lodash	Biblioteca común de herramientas y métodos	4.17.10

Nombre	Uso	Versión
esdk-obs-nodejs	SDK de OBS	2.1.5
express	Marco de desarrollo de aplicaciones basado en web simplificado	4.16.4
fgs-express	Proporciona un marco de aplicación Node.js para FunctionGraph y API Gateway para ejecutar aplicaciones sin servidor y las REST API. Este componente proporciona un ejemplo del uso del framework Express para crear aplicaciones o servicios web sin servidor y API RESTful.	1.0.1
request	Simplifica la invocación de HTTP y admite HTTPS y redirección.	2.88.0

Tabla 9-2 Bibliotecas no estándar soportadas por el tiempo de ejecución de Python

Módulo	Uso	Versión
dateutil	Fecha y hora de procesamiento	2.6.0
requests	Biblioteca de HTTP	2.7.0
httplib2	Cliente de HTTP	0.10.3
numpy	Cómputo matemática	1.13.1
redis	Cliente de Redis	2.10.5
obsclient	Cliente de OBS	-
smnsdk	Acceso de SMN	1.0.1

- Otras bibliotecas de terceros (FunctionGraph no tiene bibliotecas integradas de terceros no estándar, excepto las enumeradas en la tabla anterior.)

Para usar funciones de bibliotecas de terceros, empaquete estas bibliotecas y cárguelas en un bucket OBS especificado, y pegue la URL de enlace OBS de estas bibliotecas al crear una función.

Importación de bibliotecas dependientes

El código para procesar imágenes es el siguiente:

```
# -*- coding: utf-8 -*-
from PIL import Image, ImageEnhance
```

```
from com.obs.client.obs_client import ObsClient

import sys
import os

current_file_path = os.path.dirname(os.path.realpath(__file__))
# append current path to search paths, so that we can import some third
party libraries.
sys.path.append(current_file_path)
region = 'china'
obs_server = 'obs.xxxxxxxxcloud.com'
def newObsClient(context):
    ak = context.getAccessKey()
    sk = context.getSecretKey()
    return ObsClient(access_key_id=ak, secret_access_key=sk,
server=obs_server,
                  path_style=True, region=region, ssl_verify=False,
max_retry_count=5, timeout=20)
def downloadFile(obsClient, bucket, objName, localFile):
    resp = obsClient.getObject(bucket, objName, localFile)
    if resp.status < 300:
        print 'download file', file, 'succeed'
    else:
        print('download failed, errorCode: %s, errorMessage: %s,
requestId: %s' % resp.errorCode, resp.errorMessage,
                  resp.requestId)
def uploadFileToObs(client, bucket, objName, file):
    resp = client.putFile(bucket, objName, file)
    if resp.status < 300:
        print 'upload file', file, 'succeed'
    else:
        print('upload failed, errorCode: %s, errorMessage: %s,
requestId: %s' % resp.errorCode, resp.errorMessage,
                  resp.requestId)
def getObjInfoFromObsEvent(event):
    s3 = event['Records'][0]['s3']
    eventName = event['Records'][0]['eventName']
    bucket = s3['bucket']['name']
    objName = s3['object']['key']
    print "*** obsEventName: %s, srcBucketName: %s, objName: %s",
eventName, bucket, objName
    return bucket, objName
def set_opacity(im, opacity):
    """Set the transparency."""
    if im.mode != "RGBA":
        im = im.convert('RGBA')
    else:
        im = im.copy()
        alpha = im.split()[3]
        alpha = ImageEnhance.Brightness(alpha).enhance(opacity)
        im.putalpha(alpha)
    return im
def watermark(im, mark, opacity=0.6):
    """Add a watermark."""
    try:
        if opacity < 1:
            mark = set_opacity(mark, opacity)
        if im.mode != 'RGBA':
            im = im.convert('RGBA')
        if im.size[0] < mark.size[0] or im.size[1] < mark.size[1]:
            print "The mark image size is larger size than original
image file."
```

```
        return False
    x = (im.size[0] - mark.size[0]) / 2
    y = (im.size[1] - mark.size[1]) / 2
    layer = Image.new('RGBA', im.size, )
    layer.paste(mark, (x, y))
    return Image.composite(layer, im, layer)
except Exception as e:
    print ">>>>>>>>> WaterMark EXCEPTION: " + str(e)
    return False
def watermark_image(localFile, fileName):
    im = Image.open(localFile)
    watermark_image_path = os.path.join(current_file_path,
"watermark.png")
    mark = Image.open(watermark_image_path)
    out = watermark(im, mark)
    print "*****finish water mark"
    name = fileName.split('.')
    outFileName = name[0] + '-watermark.' + name[1]
    outFilePath = "/tmp/" + outFileName
    if out:
        out = out.convert('RGB')
        out.save(outFilePath)
    else:
        print "Sorry, Save watermarked file Failed."
    return outFileName, outFilePath
def handler(event, context):
    srcBucket, srcObjName = getObjInfoFromObsEvent(event)
    outputBucket = context.getUserData('obs_output_bucket')
    client = newObsClient(context)
    # download file uploaded by user from obs
    localFile = "/tmp/" + srcObjName
    downloadFile(client, srcBucket, srcObjName, localFile)
    outFileName, outFile = watermark_image(localFile, srcObjName)
    # Upload converted files to a new OBS bucket.
    uploadFileToObs(client, outputBucket, outFileName, outFile)
    return 'OK'
```

Para las bibliotecas estándar y las bibliotecas no estándar compatibles, puede usarlas directamente en su función.

Para las bibliotecas de terceros no estándar que no son proporcionadas por FunctionGraph puede usarlas siguiendo los siguientes pasos:

1. Empaque las bibliotecas dependientes en un archivo ZIP, cargue el archivo ZIP en un bucket OBS y obtenga la URL de enlace OBS.
2. Inicie sesión en la consola FunctionGraph y elija **Functions > Dependencies** en el panel de navegación.
3. Haga clic en **Create Dependency**.
4. Establezca el nombre de la dependencia y el tiempo de ejecución, especifique el vínculo OBS y haga clic en **OK**.

NOTA

Para obtener más información sobre cómo obtener la URL de enlace OBS, consulte [Acceso a un objeto mediante su URL](#). (La siguiente figura se incluye solo para referencia. Por favor, utilice la URL real del paquete de archivo subido.)

Figura 9-1 Obtención de la URL del enlace OBS

Object Name	test	Version ID	null
Storage Class	Standard	Change Storage Class	Last Modified
Size	5.55 MB	Encrypted	No
Link	https://ysx-fgs-test.obs.cn-north-6.myhuaweicloud.com/test		

Figura 9-2 Establecer la dependencia

Create Dependency

Name: test

Runtime: Python 2.7

Description: Enter a maximum of 512 characters.

Upload Mode: [Upload ZIP file](#) [Upload file from OBS](#)

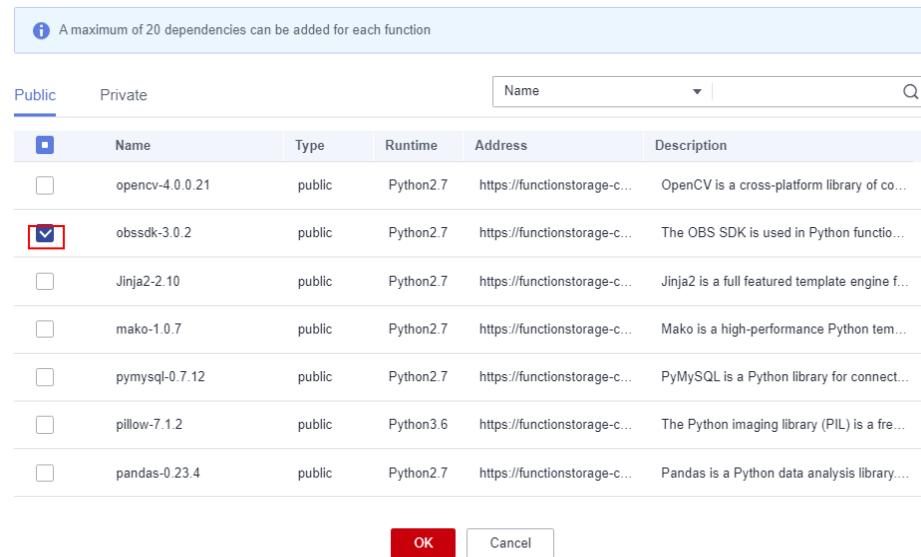
<https://obs-code.obs.myhwclouds.com/depends.zip>

Paste the OBS link URL of a ZIP file. [View OBS console](#)

OK Cancel

5. En la página de detalles de la función, haga clic en la ficha **Code**, haga clic en **Select** junto a **Dependencies**, seleccione la dependencia creada en 4(ver [Figura 9-3](#)), y haga clic en **OK**.

Figura 9-3 Selección de una dependencia
Select Dependency



6. Haga clic en **Save**.

⚠️ ADVERTENCIA

Cada paquete de dependencias no puede contener un archivo con el mismo nombre que un archivo de código. De lo contrario, los dos archivos pueden fusionarse o sobrescribirse incorrectamente. Por ejemplo, si el paquete de dependencias **depends.zip** contiene un archivo llamado **index.py**, el controlador de una función no se puede establecer en **index.handler**. De lo contrario, se generará un archivo de código también llamado **index.py**.

10 Gestión de instancias reservadas

¿Qué es una instancia reservada?

FunctionGraph proporciona instancias reservadas y bajo demanda.

- Las instancias bajo demanda son creadas y liberadas por FunctionGraph según el uso real de la función. Al recibir solicitudes para llamar a funciones, el FunctionGraph asigna automáticamente recursos de ejecución a las solicitudes.
- Las instancias reservadas pueden ser creadas y liberadas por usted según sea necesario. Después de crear instancias reservadas para una función, el FunctionGraph reenvía preferentemente las solicitudes para invocar la función a las instancias reservadas. Si el número de solicitudes excede la capacidad de procesamiento de las instancias reservadas, el FunctionGraph reenviará las solicitudes excesivas a instancias bajo demanda y asignará automáticamente recursos de ejecución a estas solicitudes.

Después de crear instancias reservadas para una función, el código, las dependencias y el inicializador de la función se cargan automáticamente. Las instancias reservadas siempre están activas en el entorno de ejecución y eliminan la influencia de los arranques en frío en sus servicios.

NOTA

De forma predeterminada, no tiene permisos para usar instancias reservadas. Para usar esta función, [envíe un ticket de servicio](#) para agregar su cuenta a la lista blanca.

Puede crear instancias reservadas directamente o crearlas en la página de configuración de funciones. Las diferencias entre los dos modos se describen en la siguiente tabla.

Tabla 10-1 Diferencias entre los dos modos

Modo	Ventajas	Desventaja
Creación directa de instancias reservadas	Puede crear una instancia reservada con solo unos pocos clics.	Solo se puede crear un número fijo de instancias reservadas. Las instancias reservadas pueden ser insuficientes durante las horas pico y quedarse inactivas durante las horas fuera de pico.

Modo	Ventajas	Desventaja
Creación de instancias reservadas en la página de configuración de funciones	Puede crear diferentes números de instancias reservadas para diferentes períodos, evitando el desperdicio de instancias reservadas durante las horas valle y asegurando suficientes instancias reservadas durante las horas pico.	El procedimiento es complejo.

Creación directa de un número fijo de instancias reservadas

Antes de crear instancias reservadas, asegúrese de que la función, por ejemplo, **Objective-func**, para la que desea crear instancias reservadas ya existe en la consola FunctionGraph.

1. Inicie sesión en la consola FunctionGraph y elija **Functions** > **Reserved Instances** en el panel de navegación.
2. Haga clic en **Configure Reserved Instance**.
3. Establezca los siguientes parámetros:

Tabla 10-2 Información de instancia reservada

Parámetro	Descripción
App	Selecciona la aplicación a la que pertenece la función Objective-func .
Function	Seleccione Objective-func .
Version	Seleccione una versión de la función Objective-func .
RI Quantity	Introduzca el número de instancias reservadas que se van a crear. Para determinar cuántas instancias reservadas crear, vea el número de instancias usadas en el área Instance Statistics o compruebe el uso de la función Objective-func .

4. Haga clic en **OK**.

 **NOTA**

Después de crear instancias reservadas para una función, sólo puede cambiar la cantidad de instancia reservada.

Creación de instancias reservadas en la página de configuración de funciones

El número de instancias de función varía entre diferentes períodos. Para garantizar suficientes instancias reservadas durante las horas pico y evitar el desperdicio de instancias reservadas durante las horas valle, puede crear un temporizador para invocar la función correspondiente con diferente número de instancias reservadas en diferentes períodos.

Antes de crear instancias reservadas, asegúrese de que la función, por ejemplo, **Objective-func**, para la que desea crear instancias reservadas ya existe en la consola FunctionGraph.

1. Inicie sesión en la consola FunctionGraph y seleccione **Functions > Function List** en el panel de navegación.
2. Haga clic en **Create Function**.
3. Establezca la siguiente información.

Parámetro	Descripción
Template	Selecciona Create from scratch .
Function Name	Introduzca un nombre para identificar la función.
App	Seleccione default .
Agency	Selecciona Use no agency .
Description	Introduzca una descripción para la función. Este parámetro es por elección.
Runtime	Seleccione Python 2.7 .
Handler	Ingrese index.handler .
Code Entry Mode	Al crear una función, seleccione Default code . En la página de ficha Configuration , seleccione Edit code inline e introduzca el siguiente código:

```
# -*- coding:utf-8 -*-
import json
import requests
def handler(event, context):
    domainId = "https://{{Endpoint}}"
    url = "/v2/{{project_id}}/fgs/functions/{{func-urn}}/reservedinstances"
    token = context getToken()
    requrl = domainId + url
    headerdata = {"Content-Type": "application/json", "x-auth-token": token}
    r = requests.put(requrl, data=event["user_event"],
    headers=headerdata, verify=False)
    return r.json
```

Reemplace los siguientes parámetros por los valores reales:

- *Endpoint*: Endpoint de la función **Objective-func**. Para obtener más información, consulte [Regiones y Endpoints](#).
- *project_id*: ID del proyecto al que pertenece la función **Objective-func**. Para obtener más detalles, consulte [Obtención de un ID del proyecto](#).
- *func-urn*: URN de la función **Objective-func**.

4. Haga clic en **Create Now**.
5. En la página de la ficha **Configuration**, haga clic en **Create Agency**.
6. Crear una delegación y concederle el permiso de **FunctionGraph User**. Para más detalles, consulte [Configuración de permisos de delegación](#).
7. En la página de ficha **Configuration**, seleccione la delegación creada en 6 y haga clic en **Save**.
8. En la página de la ficha **Triggers**, haga clic en **Create Trigger**.
9. Establezca la siguiente información.

Parámetro	Descripción
Trigger Type	Seleccione Timer .
Timer Name	Introduzca un nombre para identificar el temporizador.
Rule	Seleccione Cron expression e introduzca una expresión cron según sea necesario.
Enable Trigger	De forma predeterminada, esta función está habilitada. Consérve la configuración predeterminada.
Additional Information	Introduzca el número de instancias reservadas necesarias en diferentes períodos según la regla del activador.

Figura 10-1 Creación de un activador de temporizador

Create Trigger

Trigger Type [?](#) Timer

The total number of DDS, GAUSSMONGO, DIS, DMS, LTS, Kafka and timer triggers cannot exceed 10. You have created 0 such triggers.

* Timer Name Timer-k2nn

Enter 1 to 64 characters, starting with a letter. Only letters, digits, hyphens (-), and underscores (_) are allowed.

* Rule Fixed rate Cron expression 0 */3 * * ?

[Cron Expressions](#)

* Enable Trigger

Additional Information [?](#) {"count":2}

112,048

Figura 10-1 muestra un activador de temporizador que solicita a FunctionGraph que cree dos instancias reservadas cada 3 minutos.

10. Haga clic en **OK**.

Después de la creación del activador, se crearán diferentes números de instancias reservadas para la función **Objective-func** en diferentes períodos basados en la regla del activador.

11 Aumento de la cuota de recursos

Información general

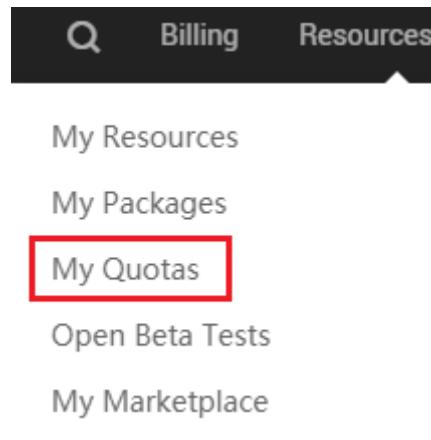
Una cuota es un límite en la cantidad o capacidad de un determinado tipo de recursos de servicio que puede usar. Por ejemplo, el número máximo de ECS o discos EVS que se pueden crear.

Si una cuota no puede satisfacer sus necesidades, solicite una cuota más alta.

Consulta de cuotas

1. Inicie sesión en la consola de gestión.
2. En la esquina superior derecha de la página, seleccione **Resources > My Quotas**.

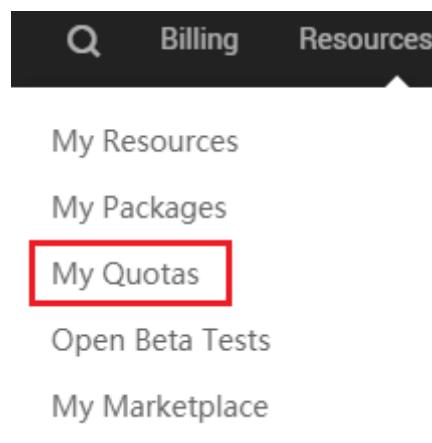
Figura 11-1 Ir a la página **Service Quota**



3. En la página **Service Quota**, vea las cuotas usadas y totales de recursos. Si una cuota no puede satisfacer sus necesidades, solicite una cuota más alta realizando las siguientes operaciones.

Aumento de las cuotas

1. Inicie sesión en la consola de gestión.
2. En la esquina superior derecha de la página, seleccione **Resources > My Quotas**. Se muestra la página **Service Quota**.

Figura 11-2 Ir a la página Service Quota

3. Haga clic en **Increase Quota**.
4. En la página **Create Service Ticket**, establezca los parámetros.
En el área **Problem Description**, introduzca la cuota requerida y el motivo del ajuste.
5. Lea los acuerdos y confirme que está de acuerdo con ellos y, a continuación, haga clic en **Submit**.

12 Auditoría

12.1 Operaciones registradas por CTS

Para obtener más información, consulte [Consultar trazas en tiempo real](#).

Tabla 12-1 enumera las operaciones FunctionGraph que pueden ser registradas por CTS.

Tabla 12-1 Operaciones registradas por CTS

Operación	Tipo de recurso	Nombre del evento
Creación de una función	Function	CreateFunction
Eliminación de una función	Function	DeleteFunction
Modificación de información de función	Function	ModifyFunctionMetadata
Publicación de una versión de función	Function version	PublishFunctionVersion
Eliminación de un alias de versión de función	Function version alias	DeleteVersionAlias
Eliminación de un activador de función	Trigger	DeleteTrigger
Creación de un activador de función	Trigger	CreateTrigger
Deshabilitación de un activador de función	Trigger	DisabledTrigger
Habilitación de un activador de función	Trigger	enabledTrigger
Ejecución de una función	Function instance	invokeFunction