

FunctionGraph

FAQs

Issue	01
Date	2026-08-24



Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2026. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Cloud Computing Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei Cloud and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Cloud Computing Technologies Co., Ltd.

Address: Huawei Cloud Data Center Jiaoxinggong Road
Qianzhong Avenue
Gui'an New District
Gui Zhou 550029
People's Republic of China

Website: <https://www.huaweicloud.com/intl/en-us/>

Contents

1 Product Consulting.....	1
1.1 Do I Need to Apply for Any Compute, Storage, or Networking Services When Using FunctionGraph?.....	1
1.2 Do I Need to Deploy My Code After Programming?.....	1
1.3 How Much Disk Space Is Allocated to Each FunctionGraph Function?.....	1
1.4 Can I Enable a Listening Port in a Function to Receive External TCP Requests via EIP?.....	2
1.5 How Do I Obtain the Source IP Address of an HTTP Request Initiated by a Function?.....	2
1.6 Can FunctionGraph Decompile the Uploaded ZIP File?.....	3
1.7 Does FunctionGraph Support Function Extension?.....	4
1.8 How Is Code Isolated in FunctionGraph?.....	4
1.9 What Is an App in FunctionGraph?.....	4
1.10 Do I Need to Pay for Cold Start Time?.....	4
1.11 Will the Requests of All My Functions in Different Regions Be Billed?.....	4
1.12 What Compatibility Issues Exist During the Migration?.....	4
1.13 What Runtimes Does FunctionGraph Support?.....	5
2 Function Creation FAQs.....	7
2.1 Can I Change the Name of a Function After It Is Created?.....	7
2.2 Can I Change the Runtime of a Function After It Is Created?.....	7
2.3 Why Am I Seeing a Message Indicating that My Account Was Suspended When Creating a Function?.....	7
2.4 How Do I Export Code of a Function?.....	7
2.5 When Updating An Image with the Same Name, Why Do Reserved Instances Use the Old Image?.....	7
3 Configuring Functions.....	8
3.1 Can I Enter Sensitive Information in Environment Variables?.....	8
3.2 How Does a FunctionGraph Function Read and Write an Uploaded File?.....	8
3.3 Why "failed to mount exist system path" Is Displayed When a File System Is Mounted to a Function.....	9
3.4 How Does FunctionGraph Implement Domain Name Resolution?.....	9
3.5 How Do I Use a Domain Name to Access an API Registered with API Gateway (Dedicated)?.....	12
3.6 Why Is the Domain Name Unresolvable When a Function Accesses an API Registered with APIG Using a Domain Name?.....	13
3.7 Which Directories Can Be Accessed When a Custom Runtime Is Used?.....	13
3.8 What Chinese Fonts Does FunctionGraph Support?.....	13

3.9 Can I Add Threads and Processes in Function Code?.....	13
3.10 How Does a Function Access the MySQL Database?.....	14
3.11 What Do I Do If My Function Cannot Connect to Redis Through a VPC?.....	15
3.12 How Do I Read the Request Header of a Function?.....	15
3.13 How Do I Use the Versions and Aliases of an HTTP Function with an APIG Trigger for Gray Upgrade?	16
4 Invoking Functions.....	19
4.1 How Long Does It Take to Execute a FunctionGraph Function?.....	19
4.2 Which Steps Are Included in Function Execution?.....	19
4.3 What Will Happen to Instances If a FunctionGraph Function Is Not Executed for a Long Time?.....	19
4.4 How Do I Obtain the Memory Usage During Function Execution?.....	19
4.5 Why Is It Slow to Invoke a Function for the First Time?.....	20
4.6 How Do I Check the Used Memory If the Function Execution Fails and "runtime memory limit exceeded" Is Returned?.....	20
4.7 Why "CrashLoopBackOff" Is Displayed When a Custom Image Function Fails to Be Executed?.....	20
4.8 Why Can't I Receive Responses for Synchronous Function Invocation?.....	21
4.9 What Should I Do If the os.system("command &") Execution Logs Are Not Collected?.....	21
4.10 What Are the Possible Causes for Function Timeout?.....	21
4.11 What If Error Code 500 Is Reported When Functions that Use APIG Triggers Return Strings?.....	21
4.12 Why Can't I Print Logs of a Python 2.7 Function After the Execution of reload(sys)?.....	22
5 Configuring Triggers.....	23
5.1 How Do I Obtain the Request Path or Request Parameters in an APIG Trigger?.....	23
5.2 Can I Configure a Kafka Trigger in a Different Subnet from My Function?.....	24
6 Configuring Dependencies.....	25
6.1 How Do I Create an ODBC-based Python Function Dependency?.....	25
6.2 How Do I Create Function Dependencies?.....	25

1 Product Consulting

1.1 Do I Need to Apply for Any Compute, Storage, or Networking Services When Using FunctionGraph?

When using FunctionGraph, you do not need to apply for or pre-configure any compute, storage, or networking services, but need to upload and run code in supported runtimes. FunctionGraph provides and manages underlying compute resources, including server CPUs, memory, and networks. It performs configuration and resource maintenance, code deployment, automatic scaling, load balancing, secure upgrade, and resource monitoring.

1.2 Do I Need to Deploy My Code After Programming?

After programming, you only need to package your code into a ZIP file (Java, Node.js, Python, and Go) or JAR file (Java), and upload the file to FunctionGraph for execution.

When creating a ZIP file, place the handler file under the **root** directory to ensure that your code can be run normally after being decompressed.

If you edit code in Go, zip the compiled file, and ensure that the name of the dynamic library file is consistent with the plugin name of the handler. For example, if the name of the dynamic library file is **testplugin.so**, set the handler to **testplugin.Handler**.

1.3 How Much Disk Space Is Allocated to Each FunctionGraph Function?

Each FunctionGraph function is allocated 512 MB ephemeral disk space. You can upload deployment packages up to 10 GB in size. For more information, see [Notes and Constraints](#).

1.4 Can I Enable a Listening Port in a Function to Receive External TCP Requests via EIP?

No. FunctionGraph does not support this feature currently. Functions are about serverless computing, and compute resources are allocated while functions are running. Customizing a listening port is not suitable.

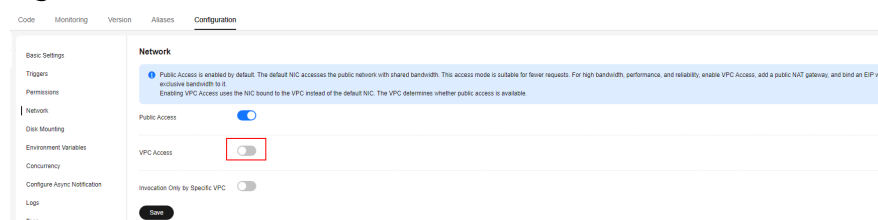
1.5 How Do I Obtain the Source IP Address of an HTTP Request Initiated by a Function?

Public Access

- **VPC Access** disabled for the function

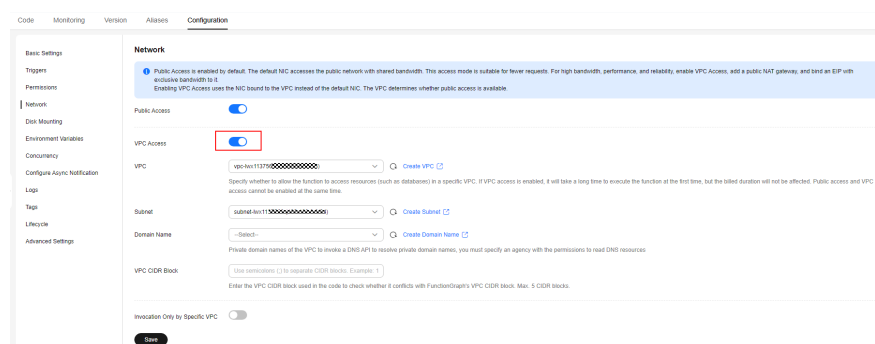
The SNAT address of FunctionGraph is used to access the public network. It is **fixed**. To obtain it, contact technical support.

Figure 1-1 VPC access disabled



- **VPC Access** enabled for the function (For details about the VPC configuration, see [Configuring Networks](#))

Figure 1-2 VPC access enabled

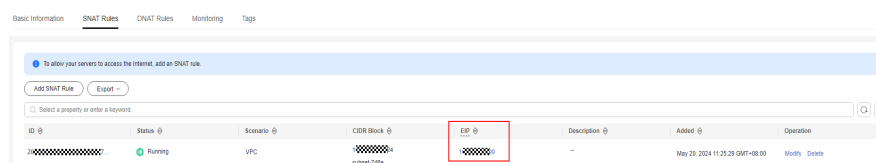


The SNAT address configured in the user VPC is used to access the public network. It is **fixed**. To obtain the public IP address, perform the following steps:

- a. Log in to the NAT Gateway console, click  in the upper left corner, and select a region.
- b. In the navigation pane on the left, choose **NAT Gateway** > **Public NAT Gateway**. In the list on the right, click the target gateway name.

- c. On the **SNAT Rules** tab, obtain the public IP address in the rule list.

Figure 1-3 Obtaining the public IP address



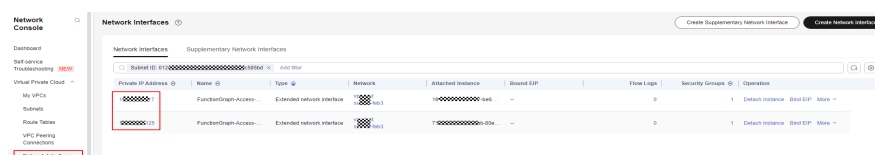
Intra-VPC Access

When **VPC Access** is enabled, functions can access resources in a VPC. (For details about the VPC configuration, see [Configuring Networks](#))

The user VPC address mounted in the PAT is used to access resources in the VPC. The address changes dynamically. To view the private IP address, perform the following steps:

1. Log in to the VPC console, click in the upper left corner, and select a region.
2. In the navigation pane on the left, choose **Virtual Private Cloud > Network Interfaces**. On the displayed page on the right, obtain the private IP address whose name is **FunctionGraph-Access-VPC-Port** and **Type** is **Extended network interface**. For details about elastic network interfaces, see [Elastic Network Interface](#).

Figure 1-4 Obtaining the private IP address



NOTE

- When configuring the whitelist or inbound/outbound rules of a security group, ensure that the configured IP address is within the VPC CIDR block. For details about how to create a security group, see [Creating a Security Group](#).
- Make sure to test the VPC function to display the **Extended network interface** in the **Type** column.
- The two private IP addresses obtained above are the active and standby addresses. For details, contact technical support.

1.6 Can FunctionGraph Decompile the Uploaded ZIP File?

No. Please decompile your files before uploading them.

1.7 Does FunctionGraph Support Function Extension?

FunctionGraph has integrated non-standard libraries such as redis, http, and obs_client. You can directly use these libraries when developing functions. For more information, see [Developer Guide](#).

Alternatively, use your own dependencies. For more information, see [Dependency Management](#).

1.8 How Is Code Isolated in FunctionGraph?

Each FunctionGraph function runs in its own environment and has its own resources and file system.

1.9 What Is an App in FunctionGraph?

An app acts like a folder. After a function is created, it is automatically categorized into the **default** app and cannot be switched to other apps. In the future, functions will be managed by tags for better experience.

1.10 Do I Need to Pay for Cold Start Time?

No. Cold start time will not be metered and you do not have to pay for it.

1.11 Will the Requests of All My Functions in Different Regions Be Billed?

Yes.

1.12 What Compatibility Issues Exist During the Migration?

1. **args** difference

V1:

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

After migration to V2, it will be:

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

The **sys.argv** of Python runtime varies between the two versions.

For V2, it is `['/home/snuser/runtime/python3.6/server.py', '127.0.0.1:31536', '/opt/function/code']`, whose last two parameters are not available for V1.

2. **asyncio** difference

V1:

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

After migration to V2, it will be:

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

In V1, **asyncio.get_event_loop()** obtains the current event loop from the OS thread (main thread). It will throw **RuntimeError** in V2, because Python runtime of V2 does not run functions in the main thread.

To use **asyncio** in V2, create an event loop.

1.13 What Runtimes Does FunctionGraph Support?

Table 1-1 lists the runtimes supported by FunctionGraph.

Table 1-1 Supported runtimes

Runtime	Supported Version	Development Instructions
Node.js	6.10, 8.10, 10.16, 12.13, 14.18, 16.17, 18.15, 20.15	For details about the function syntax, SDK APIs, and function development, see Developing Functions in Node.js .
Python	2.7, 3.6, 3.9, 3.10, 3.12	For details about the function syntax, SDK APIs, and function development, see Developing Functions in Python .
Java	8, 11, 17, 21 (only available in ME-Riyadh and TR-Istanbul)	For details about the function syntax, SDK APIs, and function development, see Developing Functions in Java .
Go	1.x	For details about the function syntax, SDK APIs, and function development, see Developing Functions in Go .

Runtime	Supported Version	Development Instructions
C#	.NET Core 2.1, .NET Core 3.1, .NET Core 6.0, .NET Core 8.0 (only in ME-Riyadh and TR-Istanbul)	For details about the function syntax, SDK APIs, and function development, see Developing Functions in C# .
PHP	7.3 and 8.3	For details about the function syntax, SDK APIs, and function development, see Developing Functions in PHP .
Custom	-	-
Cangjie	1.0	-

2 Function Creation FAQs

2.1 Can I Change the Name of a Function After It Is Created?

No. A function's name cannot be changed once the function is created.

2.2 Can I Change the Runtime of a Function After It Is Created?

No. Once a function is created, its runtime cannot be changed.

2.3 Why Am I Seeing a Message Indicating that My Account Was Suspended When Creating a Function?

Your account is in arrears.

2.4 How Do I Export Code of a Function?

1. Log in to the FunctionGraph console, and click the name of the target function to go to the details page. Choose **Operation** > **Export function** in the upper right, and click **Export Code**.
2. Alternatively, call the function export API.

2.5 When Updating An Image with the Same Name, Why Do Reserved Instances Use the Old Image?

Use a non-latest tag to manage image updates, and do not use the same image name.

3 Configuring Functions

3.1 Can I Enter Sensitive Information in Environment Variables?

FunctionGraph displays all the information you enter in plain text. Therefore, do not enter sensitive information, such as passwords, when you define environment variables. For details, see [Configuring Environment Variables](#).

3.2 How Does a FunctionGraph Function Read and Write an Uploaded File?

Background

A function can read files in the code directory. The working directory of a function is the upper-level directory of the handler file. Assume that you have uploaded a folder named **backend**. To read its **test.conf** file in the same level of directory as the handler file, use relative path **code/backend/test.conf** or use a full path (that is, the value of the **RUNTIME_CODE_ROOT** environment variable). To write a file (for example, to create or download a file), go to the **/tmp** directory or use the file system mounting feature provided by FunctionGraph.

NOTE

- If containers are reclaimed, file read/written content will become invalid.
- Currently, FunctionGraph does not support instance persistence.

Typical Scenarios

- Download files stored in Object Storage Service (OBS) to the **/tmp** directory for processing.
- To store function execution data in OBS, create a file in the **/tmp** directory, write the data into the file, and then upload the file to OBS.

Obtaining Uploaded Files

Take Python as an example. If you use `os.getcwd()` to query the current directory, the directory will be `/opt/function`. However, code has actually been uploaded to `/opt/function/code`.

You can use either of the following methods to obtain uploaded files:

1. Run the `cd` command to switch to `/opt/function/code`.
2. Access the full path (value of the `RUNTIME_CODE_ROOT` environment variable).

NOTE

You can obtain uploaded files by referring to the preceding methods when other languages are used.

3.3 Why "failed to mount exist system path" Is Displayed When a File System Is Mounted to a Function

When you see this message, mount the file to a new path.

User ID/user group ID: Can be any number except 1000. The value `-1` will be automatically converted to `1003`. The two IDs control the directory permissions for accessing a remote file system.

File system/ECS name: Name of the file system or ECS to create. Ensure that you have specified a VPC and agency that you have been authorized to access.

Shared directory: To configure a remote shared directory for the mounted ECS, see [Creating an NFS Shared Directory on an ECS](#).

Access path: Location where the file system is to be mounted in the function. Set a new two-level directory that starts with `/mnt`. For example, `/mnt/test`.

3.4 How Does FunctionGraph Implement Domain Name Resolution?

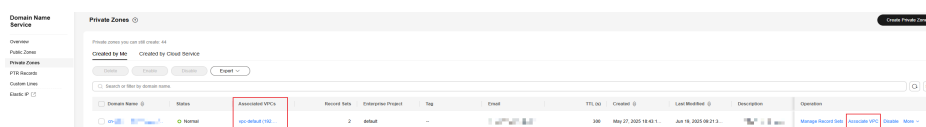
FunctionGraph cannot directly resolve private Huawei Cloud DNS domain names. To resolve them, call DNS APIs and perform the following steps.

Resolving a Private DNS Domain Name with an Event Function

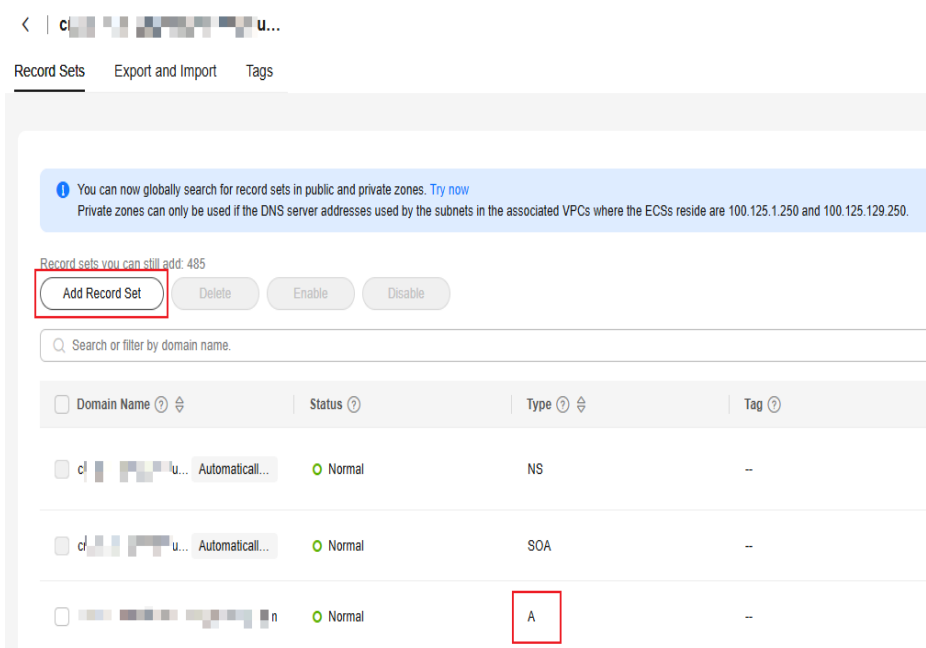
Ensure that a VPC and private DNS domain name have been created before performing the following steps:

Step 1 Associate a VPC with the private domain name and add record sets.

Log in to the DNS console and associate a VPC with the private domain name.

Figure 3-1 Associating a VPC with the private domain name

Click the domain name, and add a type A record set.

Figure 3-2 Adding a record set

Step 2 Create a function.

Create a function whose runtime is Python 2.7. The following is sample code.

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

Step 3 Configure an agency with DNS and VPC permissions for the function.

Log in to the [IAM console](#), and configure an agency with the **DNS ReadOnlyAccess** and **VPC Administrator** permissions for FunctionGraph. For details, see [Configuring Agency Permissions](#).

NOTE

You need to configure the permission to read DNS resource data because the function needs to obtain such data when resolving a domain name. Otherwise, the following error message is displayed, indicating that the DNS resource data failed to be obtained.

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

Step 4 Configure the function.

On the details page of the function created in [Step 2](#), click the **Configuration** tab and configure the following settings:

1. For **Permissions**, select the agency created in [Step 3](#).
2. Enable VPC access and select the created VPC, subnet, and domain name. For details, see [Figure 3-3](#).

Figure 3-3 Configuring the function

The screenshot shows the 'Configuration' tab of a function in FunctionGraph. The left sidebar lists various settings, with 'Network' highlighted. The main content area is titled 'Network' and contains the following settings:

- Public Access:** A toggle switch is currently turned off. A note states: 'Public Access is enabled by default. The default NIC accesses the public network with shared bandwidth. This access mode is bandwidth to it. Enabling VPC Access uses the NIC bound to the VPC instead of the default NIC. The VPC determines whether public access'.
- VPC Access:** A toggle switch is turned on and highlighted with a red box. A note states: 'Specify whether to allow the function to access resources (such as databases) in a specific VPC. cannot be enabled at the same time.'
- VPC:** A dropdown menu shows 'vpc-xxxxxx' with a search icon and a 'Create VPC' link.
- Subnet:** A dropdown menu shows 'subnet-xxxxxx24)' with a search icon and a 'Create Subnet' link.
- Domain Name:** A dropdown menu shows 'dns-xxxxxx' and 'test-xxxxxx.com' with a search icon and a 'Create Domain Name' link. A note states: 'Private domain names of the VPC to invoke a DNS API to resolve private domain names, you mu'.
- VPC CIDR Block:** A text input field with a placeholder: 'Use semicolons (;) to separate CIDR blocks. Example: 1'. A note states: 'Enter the VPC CIDR block used in the code to check whether it conflicts with FunctionGraph's VF'.
- Invocation Only by Specific VPC:** A toggle switch is currently turned off.
- Save:** A black button at the bottom.

Step 5 Check the execution result.

Then all configured IPv4 domain names can be parsed.

----End

NOTE

Changes to the IP addresses corresponding to the VPC domain names you configure will take effect in 10 minutes.

Resolving a Private DNS Domain Name with a Container Image-based Function

1. Obtain a private domain name and zone ID.
This procedure uses a domain name with a record set as an example.
 - a. Log in to the DNS console.
 - b. Obtain a zone ID.



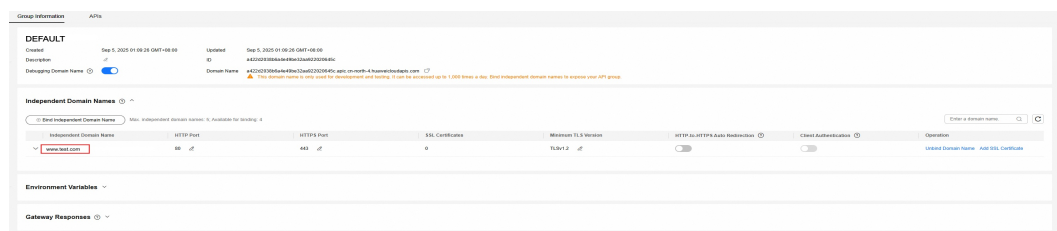
Click , and select **Domain Name** in the search box to obtain a zone ID.

- c. Obtain the private domain name corresponding to a recording set.
Click the domain name to go to the record set list, and select a record set.
2. Compile the resolution logic.
Use [API Explorer](#) to debug the API used to [query record sets in a zone](#).
 - Set **zone_id** to the zone ID obtained in the preceding step, and click **Debug**. The IP address of the private domain name is displayed in the response body.
 - Switch to the **Sample Code** tab to obtain the complete code. For details about the dependencies, click **View SDK Details**.

3.5 How Do I Use a Domain Name to Access an API Registered with API Gateway (Dedicated)?

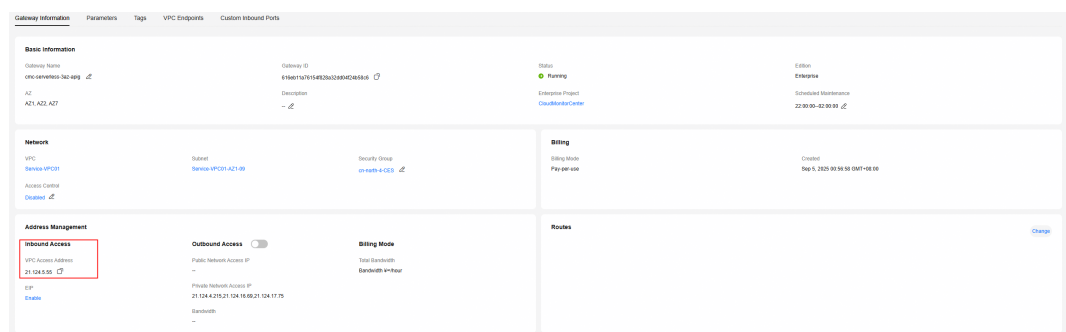
The domain name **www.test.com** is used as an example. The procedure is as follows:

Figure 3-4 Example domain name



- Step 1** Log in to the API Gateway console, choose **Dedicated Gateways** in the navigation pane, and click the target gateway name. On the **Gateway Information** page, view **EIP** in the **Inbound Access** area to obtain the IP address of the API gateway.

Figure 3-5 Obtaining the IP address of the API gateway



- Step 2** On the DNS console, configure an IPv4 rule for mapping **www.test.com** to an API gateway address.

Figure 3-6 Configuring an IPv4 rule

Domain Name	Status	Type	Tag	TTL	Value	Created	Last Modified	Description	Operation
node-aria-console-test-402.com	Normal	A	test_name = node-aria-console-test-402.com	300	21.138.25.42	Nov 26, 2025 22:12:58 GMT+08:00	Nov 26, 2025 22:12:58 GMT+08:00	--	Modify Disable Delete
node-aria-console-test-402.com	Normal	A	test_name = node-aria-console-test-402.com	300	21.138.26.114	Nov 26, 2025 22:12:58 GMT+08:00	Nov 26, 2025 22:12:58 GMT+08:00	--	Modify Disable Delete
www.test.com	Normal	A	test_name = node-aria-console-test-402.com	300	21.124.5.55	Nov 26, 2025 22:12:58 GMT+08:00	Nov 26, 2025 22:12:58 GMT+08:00	--	Modify Disable Delete
node-aria-console-test-402.com	Normal	A	test_name = node-aria-console-test-402.com	300	21.138.25.7	Nov 26, 2025 22:12:58 GMT+08:00	Nov 26, 2025 22:12:58 GMT+08:00	--	Modify Disable Delete
node-aria-console-test-402.com	Normal	A	test_name = node-aria-console-test-402.com	300	21.138.27.208	Nov 26, 2025 22:12:57 GMT+08:00	Nov 26, 2025 22:12:57 GMT+08:00	--	Modify Disable Delete

Step 3 Configure domain name resolution by referring to [How Does FunctionGraph Implement Domain Name Resolution?](#). In this way, you can access the API registered with the API gateway by using domain name **www.test.com**.

----End

3.6 Why Is the Domain Name Unresolvable When a Function Accesses an API Registered with APIG Using a Domain Name?

Currently, FunctionGraph resolves only Huawei Cloud DNS domain names and POD domain names.

3.7 Which Directories Can Be Accessed When a Custom Runtime Is Used?

By default, only the **/tmp** directory can be accessed, for example, for creating or downloading files.

3.8 What Chinese Fonts Does FunctionGraph Support?

FunctionGraph supports the following Chinese fonts:

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

3.9 Can I Add Threads and Processes in Function Code?

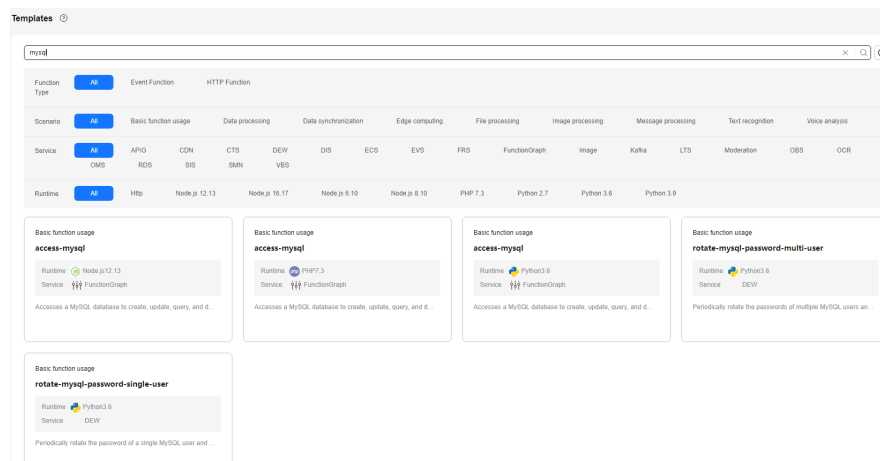
You can create additional threads and processes in your function by using runtime and OS features.

3.10 How Does a Function Access the MySQL Database?

Perform the following operations:

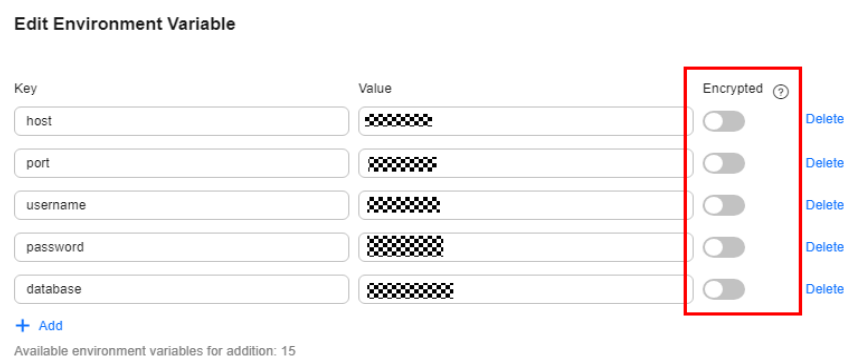
1. Check whether the MySQL database is deployed in a VPC.
 - Yes: Configure the same VPC and subnet as the MySQL database for the function by referring to [Configuring VPC Access](#).
 - No.
2. Search for MySQL templates and select the one with the desired runtime, as shown in [Figure 3-7](#). Set the parameters as required and click **Create Function**.

Figure 3-7 Selecting a function template



3. After the MySQL function is created, choose **Configuration > Environment Variables**, enable encryption as required (see [Figure 3-8](#)), and click **Save**.

Figure 3-8 Enabling encryption



NOTE

If the function needs to access RDS APIs, [create an agency](#) and grant required permissions.

3.11 What Do I Do If My Function Cannot Connect to Redis Through a VPC?

Symptom

With the same code, the function can connect to Redis 1 through VPC 1, but cannot connect to Redis 2 through VPC 2.

Solution

1. Check the current configuration: Redis1 is in VPC1, Redis2 is in VPC2, and the Redis client uses the same set of code.
2. Check the log, which shows that the Redis IP address appears as garbled characters when the client tries to connect Redis2 through VPC2.



3. Analyze Redis1 and Redis2. They are the same except for their IP addresses and passwords. The Redis2 password was found to contain an at sign (@). As a result, the IP address was incorrectly intercepted and garbled when the Redis was requested.
4. Change the password of Redis2 to rectify the fault.

3.12 How Do I Read the Request Header of a Function?

The first parameter in the function handler contains the request header. You can print the function execution result to obtain required fields.

As shown in the following figure, **event** is the first parameter in the function handler, and **headers** is the request header.

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

3.13 How Do I Use the Versions and Aliases of an HTTP Function with an APIG Trigger for Gray Upgrade?

1. Create versions V1 and V2 based on **latest**, create an alias named **alias** for V1 with a 100% weight, and create an APIG trigger for **alias**.

Figure 3-9 Publishing versions V1 and V2

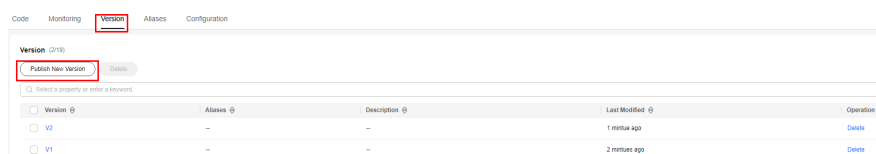


Figure 3-10 Creating alias and binding V1 to it

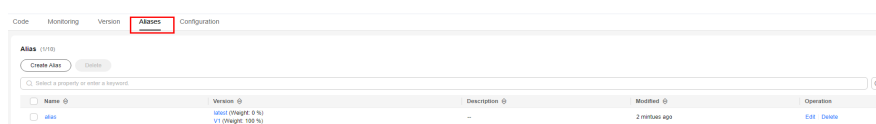
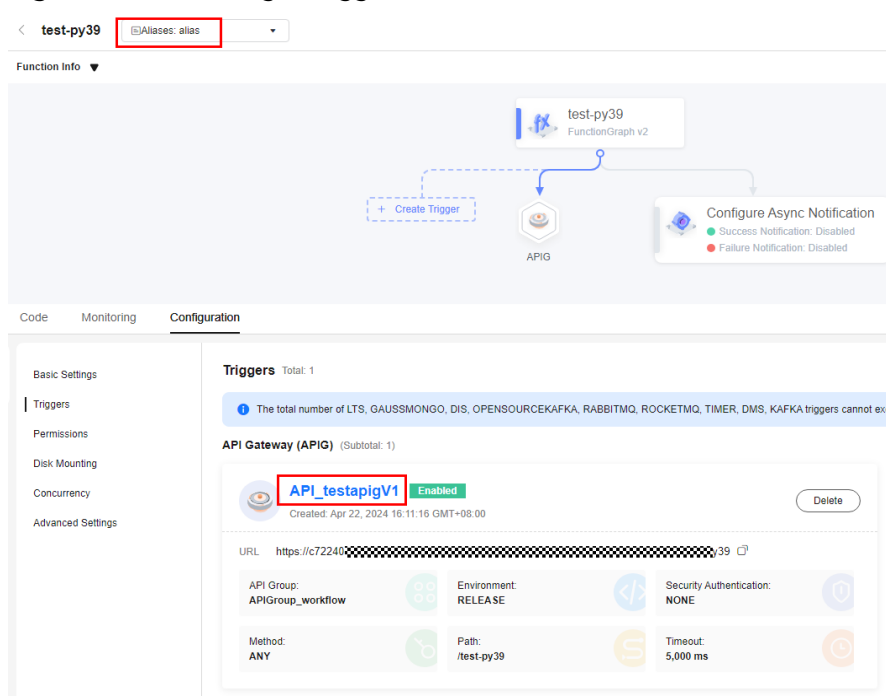


Figure 3-11 Creating a trigger for alias



NOTE

You can create APIG triggers for a function alias or version. By default, a trigger name is **API_{Function name}{Version}**, and the request path is **/_{Function name}-_{Version}**.

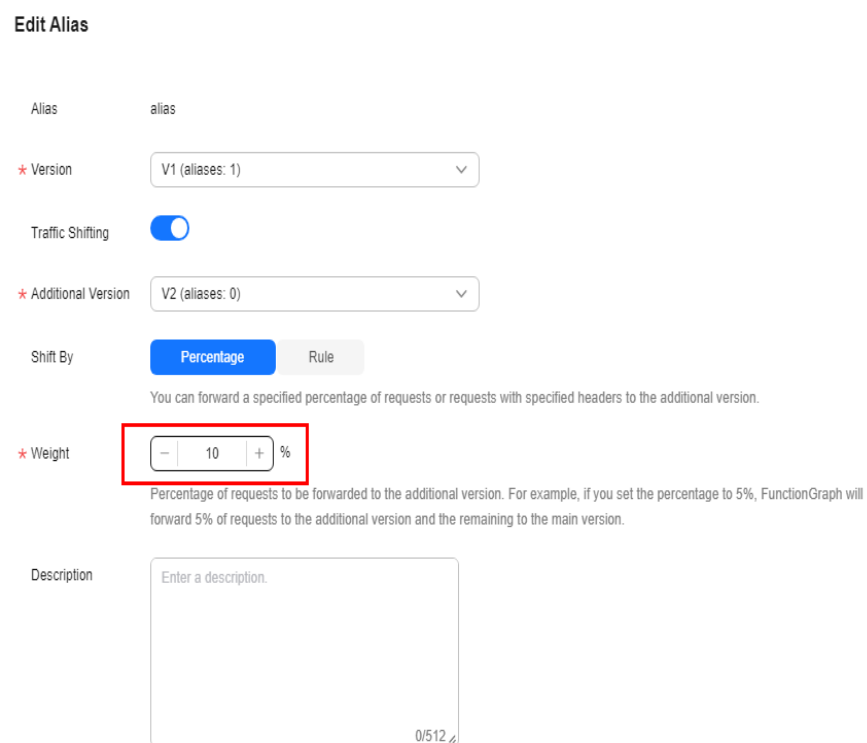
An APIG trigger URL will only be used either for an alias or its matched version. For example, assume that V1 has been bound to **alias**. If you create a trigger with the default name **API_testapigV1** for V1 and then create another trigger with the same name and URL for **alias**, the **API_testapigV1** trigger will not be displayed in the trigger list of V1.

- Go back to the **Aliases** tab page of **latest**, click **Edit** in the row of **alias**, and set **Additional Version** to **V2** with a custom weight. This is for gray upgrade from V1 to V2.

NOTE

Weights indicate the percentage of received data that will be allocated to corresponding versions. You can set a weight that meets your service requirements.

Figure 3-12 Editing an alias



Alias alias

* Version V1 (aliases: 1)

Traffic Shifting ☒

* Additional Version V2 (aliases: 0)

Shift By Percentage Rule

You can forward a specified percentage of requests or requests with specified headers to the additional version.

* Weight %

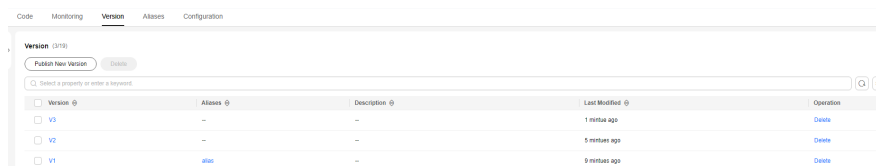
Percentage of requests to be forwarded to the additional version. For example, if you set the percentage to 5%, FunctionGraph will forward 5% of requests to the additional version and the remaining to the main version.

Description

0/512

- Create version V3, click **Edit** in the row of **alias** on the **Aliases** tab page, and change **Version** to **V2** and **Additional Version** to **V3** with a custom weight. This is for gray upgrade from V2 to V3.

Figure 3-13 Creating V3



Version	Aliases	Description	Last Modified	Operation
V3	--	--	1 minutes ago	Delete
V2	--	--	5 minutes ago	Delete
V1	alias	--	9 minutes ago	Delete

Figure 3-14 Editing an alias

Edit Alias

Alias

alias

★ Version

V2 (aliases: 0)

Traffic Shifting

☒

★ Additional Version

V3 (aliases: 0)

Shift By

Percentage

Rule

You can forward a specified percentage of requests or requests with specified headers to the additional version.

★ Weight

-

90

+

%

Percentage of requests to be forwarded to the additional version. For example, if you set the percentage to 5%, FunctionGraph will forward 5% of requests to the additional version and the remaining to the main version.

Description

Enter a description.

0/512

NOTICE

After publishing a version based on **latest**, you can publish more versions only after making a configuration or code change on the function.
Aliases bound with triggers cannot be deleted.

4 Invoking Functions

4.1 How Long Does It Take to Execute a FunctionGraph Function?

Within 900s for synchronous execution and 72 hours for asynchronous execution.

The default execution timeout is 3s. You can set the timeout (unit: s) to an integer from 3 to 259,200. If you set the timeout of a function to 3s, it will be terminated after 3s.

4.2 Which Steps Are Included in Function Execution?

Function execution includes two steps:

1. Select an idle instance with required memory.
2. Run specified code.

4.3 What Will Happen to Instances If a FunctionGraph Function Is Not Executed for a Long Time?

If a function has not been executed for a period of time, all instances related to the function will be released.

4.4 How Do I Obtain the Memory Usage During Function Execution?

The returned information about a function contains the maximum memory consumed. For more information, see [SDK APIs](#) in the *FunctionGraph Developer Guide*. Alternatively, check the memory usage in the execution result.

4.5 Why Is It Slow to Invoke a Function for the First Time?

The function experiences a cold start. Initialization or first-time operations (such as loading code and dependencies) make the first request slower. Subsequent requests are faster as long as the container remains active. If there is no request within one minute, the container will be deleted.

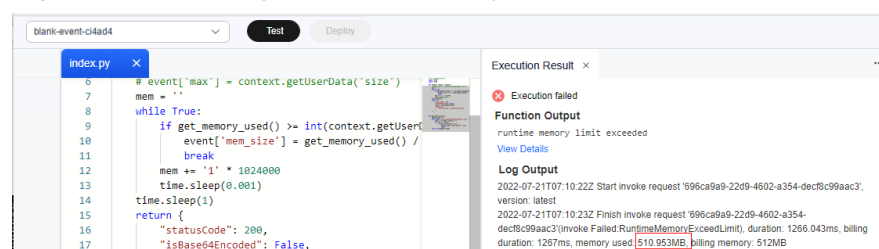
For C# or Go, startup is slower due to runtime mechanisms. To improve performance:

- Allocate more memory to the function.
- Simplify function code, for example, delete unnecessary dependency packages.
- When using C# in non-concurrent scenarios, you can also:
Create a one-minute timer trigger to ensure that there is at least one active instance.

4.6 How Do I Check the Used Memory If the Function Execution Fails and "runtime memory limit exceeded" Is Returned?

Check the used memory in the response.

Figure 4-1 Checking the used memory



4.7 Why "CrashLoopBackOff" Is Displayed When a Custom Image Function Fails to Be Executed?

The message "CrashLoopBackOff: The application inside the container keeps crashing" is displayed when a custom image execution failure occurs. In this case, perform the following operations:

1. Analyze the causes.

Figure 4-2 Viewing the execution result

```
function invocation exception, error: CrashLoopBackOff: The application inside the container keeps crashing:  
Traceback (most recent call last):  
  File "app.py", line 1, in <module>  
    from flask import Flask, request, g  
ModuleNotFoundError: No module named 'flask'
```

2. Verify the container image by referring to [Deploying a Function Using a Container Image](#).
3. Check whether the image uses the Linux x86 architecture. Currently, only Linux x86 images are supported.

4.8 Why Can't I Receive Responses for Synchronous Function Invocation?

If the E2E function execution latency exceeds 90s, asynchronous invocation is recommended. If synchronous invocation is used, no responses can be received after 90s due to gateway restrictions.

4.9 What Should I Do If the `os.system("command &")` Execution Logs Are Not Collected?

Do not use `os.system("command &")`. The background command output will not be collected. To obtain the command output, use `subprocess.Popen` instead.

4.10 What Are the Possible Causes for Function Timeout?

- The code logic timed out. In this case, optimize the code or increase the timeout.
- The network timed out. To fix this issue, increase the timeout.
- It took a long time to load Java classes during cold start. In this case, increase the timeout or memory.

4.11 What If Error Code 500 Is Reported When Functions that Use APIG Triggers Return Strings?

Ensure that the function response for an invocation by API Gateway has been encapsulated and contains `body(String)`, `statusCode(int)`, `headers(Map)`, and `isBase64Encoded(boolean)`.

The following is an example response returned by a Node.js function that uses an APIG trigger:

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

The following is an example response returned by a Java function that uses an APIG trigger:

```
import java.util.Map;  
  
public HttpTriggerResponse index(String event, Context context){  
  String body = "<html><title>FunctionStage</title>"  
    + "<h1>This is a simple APIG trigger test</h1><br>"  
    + "<h2>This is a simple APIG trigger test</h2><br>"  
    + "<h3>This is a simple APIG trigger test</h3>"  
    + "</html>";  
  int code = 200;  
  boolean isBase64 = false;  
  Map<String, String> headers = new HashMap<String, String>();  
  headers.put("Content-Type", "text/html; charset=utf-8");  
  return new HttpTriggerResponse(body, headers, code, isBase64);  
}  
  
class HttpTriggerResponse {  
  private String body;  
  private Map<String, String> headers;  
  private int statusCode;  
  private boolean isBase64Encoded;  
  public HttpTriggerResponse(String body, Map<String,String> headers, int statusCode,  
boolean isBase64Encoded){  
    this.body = body;  
    this.headers = headers;  
    this.statusCode = statusCode;  
    this.isBase64Encoded = isBase64Encoded;  
  }  
}
```

4.12 Why Can't I Print Logs of a Python 2.7 Function After the Execution of reload(sys)?

Try printing logs with `context.getLogger()`.

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

5 Configuring Triggers

5.1 How Do I Obtain the Request Path or Request Parameters in an APIG Trigger?

The request path or request parameters are carried in the input parameters of **event** by default. The input value of an APIG trigger is the event template of the function. For details about the event template, see [APIG Event Source](#).

How to obtain

The following are the formats for obtaining the request path and request parameters from the **event** object:

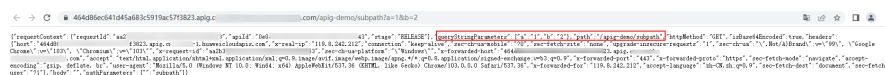
Format of the request path: `event['path']`

Format of the request parameter: `event['queryStringParameters']['Parameter name']`

How to call

You can call an API using its request path. Example: **`https://464d86ec641d45a683c5919ac57f3823.apig.projectID.huaweicloudapis.com/apig-demo/subpath`**

You can also call the API by adding request parameters: **`https://464d86ec641d45a683c5919ac57f3823.apig.projectID.huaweicloudapis.com/apig-demo/subpath?a=1&b=2`**



```
{
  "requestContext": {
    "requestId": "a2f3023.apig.o",
    "apiId": "bd",
    "stage": "RELEASE",
    "requestUri": "/apig-demo/subpath? a=1&b=2",
    "httpMethod": "GET",
    "extendedRequestId": "a2f3023.apig.o",
    "accountId": "464d86ec641d45a683c5919ac57f3823",
    "resourcePath": "/apig-demo/subpath",
    "httpVersion": "1.1",
    "ip": "192.168.1.1",
    "caller": "APIG"
  },
  "queryStringParameters": {
    "a": "1",
    "b": "2"
  }
}
```

Table 5-1 Request path and parameters

Parameter	Description
queryStringParameters	Request parameters. Parameters following the URL in a GET request. When a GET request is initiated, the parameters are transferred in the form of a URL string. That is, the character string after the question mark (?) are the request parameters and are separated by &.
path	Request path. Access address of the API.

5.2 Can I Configure a Kafka Trigger in a Different Subnet from My Function?

No. You can only create a Kafka trigger in the same VPC subnet as your function.

6 Configuring Dependencies

6.1 How Do I Create an ODBC-based Python Function Dependency?

For OS-dependent packages (for example, unixODBC), download the source code to compile dependency packages.

1. Log in to your ECS on the ECS console (ensure that the GCC and Make tools have been installed), and run the following command to download the source code package:

```
wget source code path
```

If you downloaded a **.zip** file, run the following command to decompress it:

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

If you downloaded a **tar.gz** file, run the following command to decompress it:

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

2. Run the following command to create the **/opt/function/code** directory:

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

3. Go to the destination directory and run the following command:

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

4. Go to **/opt/function/code/lib/pkgconfig** and check whether the prefix directory is **/opt/function/code**.

```
cd /opt/function/code/lib/pkgconfig
```

5. Copy all files in **/opt/function/code/lib** to **/opt/function/code**.

```
cp -r /opt/function/code/lib/* /opt/function/code
```

6. Switch to **/opt/function/code** and compress all files in it to a **.zip** package.

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

6.2 How Do I Create Function Dependencies?

This section describes how to create a private dependency of a function on your local PC.

Notes and Constraints

- The maximum size of a dependency is 300 MB, and the maximum number of files in a dependency is 30,000.
- The name of a file in a dependency cannot end with a tilde (~), for example, **module~**.
- If the modules to be installed need dependencies such as **.dll**, **.so**, and **.a**, archive them to a **.zip** package.
- **You are advised to create function dependencies in Huawei Cloud EulerOS 2.0.** If other OSs are used, the dynamic link library may not be found due to the differences between underlying dependency libraries.

Setting Up the EulerOS Environment

You are advised to create function dependencies in EulerOS. EulerOS is an enterprise-grade Linux OS based on open-source technology. It features high security, scalability, and performance, meeting customers' requirements for IT infrastructure and cloud computing services.

You can set up the [Huawei Cloud EulerOS](#) environment using the following methods:

- Buy a EulerOS ECS on Huawei Cloud by referring to [Purchasing and Logging In to a Linux ECS](#). On the **Configure Basic Settings** page, select **Public Image**, and select **Huawei Cloud EulerOS** and an image version.
- Download the EulerOS image, and use virtualization software to set up the EulerOS VM on a local PC.

Creating a Function Dependency

For details about how to create a dependency for a function, see .

Helpful Links

- After creating a dependency locally, create and configure the dependency on the FunctionGraph console by referring to .
- Manage function dependencies through APIs. For details, see [Dependency APIs](#).