08 Best Practices (Official)

 Issue
 01

 Date
 2025-02-28





Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2025. 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

NUAWEI 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 FunctionGraph Best Practices	1
2 Processing DIS Data	3
2.1 Introduction	3
2.2 Preparation	3
2.3 Building a Program	5
2.4 Adding an Event Source	11
2.5 Processing Data	12
3 Integrating with LTS to Analyze Logs in Real Time	14
3.1 Introduction	14
3.2 Preparation	15
3.3 Building a Program	17
3.4 Adding an Event Source	18
3.5 Processing Log Data	18
3.6 Other Application Scenarios	19
4 Integrating with CTS to Analyze Login/Logout Security	20
4.1 Introduction	20
4.2 Preparation	21
4.3 Building a Program	22
4.4 Adding an Event Source	22
4.5 Processing Operation Records	23
5 Periodically Starting or Stopping Huawei Cloud ECSs	24
6 Building an HTTP Function with Spring Boot	28
7 Creating a FunctionGraph Backend API That Uses a Custom Authorizer	33
7.1 Introduction	33
7.2 Resource Planning	34
7.3 Building a Program	34
7.4 Adding an Event Source	40
7.5 Debugging and Calling the API	41
8 Uploading Files with FunctionGraph and APIG	43
8.1 Introduction	43

8.2 Resource Planning	43
8.3 Procedure	44
8.3.1 Node.is	
8.3.2 Python	46
9 Processing IoT Data	49
91 Introduction	49
9.2 Preparation	50
9.3 Building a Program	
10 Workflow + Function: Automatically Processing Data in OBS	
10.1 Introduction	
10.2 Preparation	
10.3 Building a Program	57
10.4 Processing Images	62
11 Filtering Logs in Real Time by Using FunctionGraph and ITS	64
111 Introduction	64
11.2 Preparation	
11.3 Building a Program	67
11.4 Adding an Event Source	69
11.5 Processing Results	69
11.6 Extended Applications	70
12 Deploying AI Drawing Stable-Diffusion in the Application Center	72
12 Deploying AI Drawing Stable-Diffusion in the Application Center 12.1 Introduction	72
 12 Deploying AI Drawing Stable-Diffusion in the Application Center 12.1 Introduction	72
 12 Deploying Al Drawing Stable-Diffusion in the Application Center 12.1 Introduction	72 72 72 72
 12 Deploying Al Drawing Stable-Diffusion in the Application Center	
 12 Deploying Al Drawing Stable-Diffusion in the Application Center	
 12 Deploying Al Drawing Stable-Diffusion in the Application Center	
 12 Deploying Al Drawing Stable-Diffusion in the Application Center	72 72 72 72 72 73 73 73 73 74 74
12 Deploying Al Drawing Stable-Diffusion in the Application Center. 12.1 Introduction. 12.2 Preparation. 12.2.1 Overview. 12.2.2 Creating a VPC and Subnet. 12.2.3 Creating an SFS Turbo File System. 12.2.4 Creating an Agency. 12.2.5 Configuring Domain Name Resolution. 12.3 Application Creation and Deployment.	72 72 72 72 72 73 73 73 73 74 76 78
12 Deploying Al Drawing Stable-Diffusion in the Application Center. 12.1 Introduction. 12.2 Preparation. 12.2.1 Overview. 12.2.2 Creating a VPC and Subnet. 12.2.3 Creating an SFS Turbo File System. 12.2.4 Creating an Agency. 12.2.5 Configuring Domain Name Resolution. 12.3 Application Creation and Deployment. 12.4 Application Use.	72 72 72 72 73 73 73 73 74 74 76 80
12 Deploying Al Drawing Stable-Diffusion in the Application Center	72 72 72 72 73 73 73 73 74 76 76 80 80 81
12 Deploying Al Drawing Stable-Diffusion in the Application Center. 12.1 Introduction. 12.2 Preparation. 12.2.1 Overview. 12.2.2 Creating a VPC and Subnet. 12.2.3 Creating an SFS Turbo File System. 12.2.4 Creating an Agency. 12.2.5 Configuring Domain Name Resolution. 12.3 Application Creation and Deployment. 12.4 Application Use. 12.5 Custom Models. 12.5.1 Initializing a Model.	72 72 72 72 73 73 73 74 74 76 78 80 81 81
12 Deploying Al Drawing Stable-Diffusion in the Application Center. 12.1 Introduction. 12.2 Preparation. 12.2.1 Overview. 12.2.2 Creating a VPC and Subnet. 12.2.3 Creating an SFS Turbo File System. 12.2.4 Creating an Agency. 12.2.5 Configuring Domain Name Resolution. 12.3 Application Creation and Deployment. 12.4 Application Use. 12.5 Custom Models. 12.5.1 Initializing a Model. 12.5.2 Importing and Loading Models.	72 72 72 72 73 73 73 73 74 76 78 80 80 81 81 81 83
12 Deploying AI Drawing Stable-Diffusion in the Application Center	72 72 72 72 73 73 73 73 74 76 78 80 80 81 81 81 83 84
12 Deploying AI Drawing Stable-Diffusion in the Application Center	72 72 72 72 73 73 73 73 74 76 78 80 81 81 81 81 83 84 84
12 Deploying AI Drawing Stable-Diffusion in the Application Center	72 72 72 72 73 73 73 74 74 76 78 80 80 81 81 81 81 83 84 84 84 82 80 80 80 80 80 80 80 80 80 80 80 80 80
12 Deploying Al Drawing Stable-Diffusion in the Application Center	72 72 72 72 73 73 73 74 74 76 78 80 81 81 81 81 83 84 84 83 90 90
12 Deploying AI Drawing Stable-Diffusion in the Application Center	72 72 72 72 73 73 73 73 74 76 78 80 81 81 81 81 81 81 81 83 84 84 84 90 90
12 Deploying AI Drawing Stable-Diffusion in the Application Center	72 72 72 72 73 73 73 74 74 76 78 80 81 81 81 81 81 83 84 84 84 89 90 90 90
12 Deploying AI Drawing Stable-Diffusion in the Application Center	72 72 72 72 73 73 73 74 74 76 78 80 81 81 81 81 81 83 84 84 89 90 90 90 90 90

14 Using FunctionGraph HTTP Functions to Process gRPC Requests	97
15 Cold Start Optimization Practices	100

FunctionGraph Best Practices

This document summarizes practices in common application scenarios of FunctionGraph. Each practice case is given detailed solution description and operation guidance, helping you easily build your services based on FunctionGraph.

Practice	Description
Processing DIS Data	When using the Data Ingestion Service (DIS) to collect real-time Internet of Things (IoT) data streams, process the collected data, for example, convert its format, and then store the processed data into the CloudTable Service (CloudTable).
Integrating with LTS to Analyze Logs in Real Time	Analyze and process key log data of servers such as ECS collected and converted by Log Tank Service (LTS), filter out alarm logs, store processed log data in a specified OBS bucket, and use SMN to push alarm messages to service personnel by SMS message or email.
Integrating with CTS to Analyze Login/Logout Security	Obtain operation records of subscribed cloud resources from Cloud Trace Service (CTS), analyze and process the records, report alarms, and use SMN to push alarm messages to service personnel by SMS message or email.
Periodically Starting or Stopping Huawei Cloud ECSs	Use FunctionGraph to call the corresponding ECS APIs to start or stop ECSs at specified time.

Table 1-1 FunctionGraph best practices

Practice	Description
Building an HTTP Function with Spring Boot	Deploy services on FunctionGraph using Spring Boot.
Creating a FunctionGraph Backend API That Uses a Custom Authorizer	APIG supports IAM, app, and custom authentication. Create a FunctionGraph API that uses a custom authorizer.
Uploading Files with FunctionGraph and APIG	Uses Node.js and Python as examples to describe how to develop a backend parsing function for obtaining uploaded files, such as run logs and web application images, from devices to cloud servers based on FunctionGraph and APIG.
Processing IoT Data	Use FunctionGraph functions to process data reported by IoT devices and device status changes through IoTDA triggers.
Workflow + Function: Automatically Processing Data in OBS	Use a function flow to automatically process data in OBS, such as video analysis, image transcoding, and video frame capturing (available only in CN East-Shanghai1 and AP-Singapore).
Filtering Logs in Real Time by Using FunctionGraph and LTS	Obtain log data using an LTS trigger created on FunctionGraph, analyze and process key information in the logs by using a customized function, and then transfer the filtered logs to another log stream.
Building an HTTP Function with Go	Deploy services on FunctionGraph using Go.
Using FunctionGraph HTTP Functions to Process gRPC Requests	Process gRPC requests in FunctionGraph (only supported in LA- Santiago).
Cold Start Optimization Practices	Optimize FunctionGraph cold start.

2 Processing DIS Data

Introduction Preparation Building a Program Adding an Event Source Processing Data

2.1 Introduction

The best practice for FunctionGraph guides you through DIS data processing based on a function.

Scenarios

When using the Data Ingestion Service (DIS) to collect real-time Internet of Things (IoT) data streams, process the collected data, for example, convert its format, and then store the processed data into the CloudTable Service (CloudTable).

Procedure

- Create a Virtual Private Cloud (VPC) and cluster.
- Build a data processing program and package the code.
- Create a function on the FunctionGraph console.
- Configure a DIS event to test the data processing function.

2.2 Preparation

This tutorial demonstrates how to convert the format of DIS data and store the converted data into CloudTable. To achieve this purpose, you need to create a VPC and then create a cluster on the CloudTable console.

Before creating a function, you must create an agency to delegate FunctionGraph to access DIS and CloudTable resources.

Creating a VPC

- Step 1 Log in to the VPC console and click Create VPC.
- **Step 2** Set the private cloud information.

In the **Basic Information** area, enter a name, for example, **vpc-cloudtable**. Use the default values for other parameters.

For **Default Subnet**, use the default settings.

Step 3 Confirm the configuration information and click **Create Now**.

----End

Creating a Cluster

- Step 1 In the left navigation pane of the management console, choose Analytics > CloudTable Service to go to the CloudTable console. On the Cluster Mode page, click Buy Cluster.
- **Step 2** Set the cluster information.
 - **Region**: Use the default region.
 - Name: Enter a name, for example, cloudtable-dis.
 - VPC: Select vpc-cloudtable created in Creating a VPC.
 - Retain the default values for other parameters.

Figure 2-1 Buying a cluster

Buy Cluster									
Configure				(2) Co	onfirm				(3) Finish
* Region	• Second	Tunicate with each other or	er an in	tranet. For low net	vork latency :	and quick resour	rce access, select th	ie nearest region.	
* AZ 🕜	AZ1	AZ2		AZ3		AZ7			
* Billing Mode	Pay-per-use	Yearly/Monthly							
* Name 🕐	cloudtable-dis								
* VPC ⑦	vpc-6413		• C	View VPC					
* Subnet 🕜	subnet	87.281	•	Create Subnet					
* Security Group	cluster and a cluster of the cluster	13/49/3 ¹	• c	View Security Gr	oup				
* Database Engine	HBase	ClickHouse							
* HBase Version	2.4.14		•						

Step 3 Confirm the configuration information and click **Submit**.

Figure 2-2 Creating a cluster

CloudTable Service	Cluster Management									No.	Buy Cluster
Cluster Management											Feedback
Help									All projects 🔹 📿	Enter a cluster name.	С
	Cluster Name 💠	Cluster Status 💠	Task Status 💠	0 ¢	V \$	Enterprise Pr 💠	Created \$	Billing Mode 💠	Access Address (Intranet) 💠	Operation	
	cloudlable-ece3	© Creating0%	-	HBase	2.4.14	default	-	Pay-per-use	-	View Metric Scale Out	More +

D NOTE

Creating a cluster takes a long time. You can check the creation progress according to Figure 2-2.

----End

Creating an Agency

- Step 1 In the left navigation pane of the management console, choose Management & Governance > Identity and Access Management to go to the IAM console. Then choose Agencies in the navigation pane.
- **Step 2** On the **Agencies** page, click **Create Agency**.
- **Step 3** Set the agency information.
 - For Agency Name: Enter an agency name, for example, DISDemo.
 - For Agency Type, select Cloud service.
 - For **Cloud Service**, select **FunctionGraph**.
 - For Validity Period, select Unlimited.
- Step 4 Click Next. On the Select Policy/Role page, select DIS Administrator and Cloudtable Administrator.

NOTE

Cloudtable Administrator depends on **Tenant Guest** and **Server Administrator**. When you select the former, the latter will also be selected.

Step 5 Click **Next**, select an authorization scope that meets your service requirements, and click **OK**.

----End

2.3 Building a Program

Download the **source code** and **program package** (including function dependencies) to create a function from scratch for converting DIS stream data formats.

Creating a Project

This example uses a Java function to convert the format of DIS stream data. For details about function development, see **Developing Functions in Java**. The service code is not described.

Download the sample source code package **fss_examples_dis_cloudtable_src.zip**, decompress the file, and import it to Eclipse, as shown in **Figure 2-3**.

Figure 2-3 Sample code



In the sample code, modify **proID** (project ID), **clusID** (cluster ID), and **hostName** (CloudTable endpoint), and save the modification.

To obtain the project ID, perform the following steps:

- 1. Under the current login account in the upper right corner, choose **My Credentials**, as shown in **Figure 2-4**.
- 2. Obtain the project ID in the project list, as shown in Figure 2-5.

Figure 2-4 My Credentials

Resources	Billing	Enterprise	Tools	Service Tickets	D Q 🖓)en
					Copy URN	Basic Information Authenticated
						Security Settings
						My Credentials
					Description:	Identity and Access Management
					-	Switch Role

Figure 2-5 Project ID

My Credentials	API Credentials ③						
API Credentials	Learn more about Huawei Cloud accounts, IAM users, and projects.						
Access Keys	IAM User Name						
	Projects	Droject Name					

To obtain the cluster ID, perform the following steps:

- 1. Log in to the **CloudTable console**.
- 2. In the navigation pane, choose **Cluster Management**. Click cluster **cloudtable-dis** created in **Creating a Cluster**.

3. On the **cloudtable-dis** page that is displayed, find the cluster ID, as shown in **Figure 2-6**.

Figure 2-6 Cluster ID

IAM	Agencies ()					Create Agency	
Users							
User Groups	Diele Approies anilaties for creation: 407						
Permissions v	Al v Q Enter an agency name.						
Projects	🗌 Agency Namel D 🖗	Delegated Party 🕀	Validity Period 😣	Created 🖨	Description ()	Operation	
Agencies Identity Providers	ore_admin_trust	Cloud senice Industrial Digital Thread (IDT)	Unimited	Apr 24, 2024 09:27:51 GWT+08:00	-	Authorize Modify Delete	

When creating a function on the FunctionGraph console, set a handler in the format of [*package name*].[*file name*].[*function name*], for example, **com.huawei.cff.TableTester.MyHandler** for the preceding code.

Packaging the Code

Use Eclipse to package the code into a JAR file named **Table Tester.jar** according to the following figures.





⊜ Export	- • ×
Select Export resources into a JAR file on the local file system.	ß
Select an export wizard:	
type filter text	
 □ File System □ Preferences □ Install □ Installed Software Items to File □ Java □ JAR file □ Javadoc □ Runnable JAR file ▷ Exun/Debug ▷ Tasks ▷ Team ▷ XML 	E
? < Back Next > Finish	Cancel

Figure 2-8 Selecting a file type

JAR Export						
JAR File Specification						
Define which resources should be exported i	nto the JAR.					
Select the resources to export:						
DISTest	▼ X .classpath ▼ X .project					
 Export generated class files and resources Export all output folders for checked projects Export Java source files and resources Export refactorings for checked projects. <u>Select refactorings</u> 						
Select the <u>export destination</u> : JAR file: C:\Users!						
Options: Compress the contents of the JAR file Add directory entries Overwrite existing files without warning						
? sack	Next > Finish Cancel					

Figure 2-9 Publishing the code file

Package the function dependencies by performing the following steps:

- 1. **Download program package fss_examples_dis_cloudtable.zip**, and decompress it, as shown in **Figure 2-10**.
- 2. Use Table Tester.jar to replace DIS Test.jar, as shown in Figure 2-11.
- 3. Package all of the files into **disdemo.zip**, as shown in **Figure 2-12**.

😋 🔍 🗢 📕 🕨 Computer 🕨 DataDisk (D:) 🕨 Code 🕨 fss_examples_dis_cloudtable 🕨 File Edit View Tools Help Organize

Include in library

Share with

New folder
 Date modified
 Type

 2015/12/9 15:01
 Executable Jar File

 2015/92/4 19:59
 Executable Jar File

 2016/12/16 10:56
 Executable Jar File

 2016/12/16 10:56
 Executable Jar File

 2018/2/2 8:39
 Executable Jar File

 2017/4/15:25
 Executable Jar File

 2017/4/18 19:15
 Executable Jar File
 🔆 Favorites 🔲 Desktop Name Date modified Туре Size a commons-beanutils-1.9.1.jar a commons-collections-3.2.1.jar a commons-langr-2.6.jar a commons-lagging-1.1.1.jar a DISTest.jar a ezeroph-1.0.6.jar a json-lib-2.4.jar 228 KB 228 KB 562 KB 278 KB 60 KB 5 KB 85 KB 156 KB 🗼 Downloads 🔛 Recent Places ➢ Libraries
 ➢ Documents
 ➢ Music
 ➢ Pictures
 ☑ Videos File to be replaced 💻 Computer Local Disk (C:) 🗣 Network

Figure 2-10 File directory before replacement



😋 🔵 🗢 💄 🕨 Comput	er 🕨 DataDisk (D:) 🕨 Code 🕨 fss_examples_d	is_cloudtable 🕨			
File Edit View Tools	Help				
Organize 👻 Include i	n library 👻 Share with 👻 New folder				
🔶 Favorites	Name	Date modified	Туре	Size	
🥅 Desktop	commons-beanutils-1.9.1.jar	2015/12/9 15:01	Executable Jar File	228 KB	
🚺 Downloads	commons-collections-3.2.1.jar	2015/9/24 19:59	Executable Jar File	562 KB	
Recent Places	🔟 commons-lang-2.6.jar	2016/12/16 10:56	Executable Jar File	278 KB	
	commons-logging-1.1.1.jar	2017/4/10 11:30	Executable Jar File	60 KB	
Calibraries	ezmorph-1.0.6.jar	2017/4/7 15:25	Executable Jar File	85 KB	
Documents	🔟 json-lib-2.4.jar	2017/8/18 19:15	Executable Jar File	156 KB	
Music	🔟 TableTester.jar	2018/2/23 15:35	Executable Jar File	5 KB	
Videos					
💻 Computer 🏝 Local Disk (C:)		File a	fter replace	ment	
🕞 DataDisk (D:)					
🙀 Network					

Figure 2-12 Packaging the files in ZIP format

ile Edit View Tools	Help			
Organize 🔻 🛛 Include i	n library 🔻 Share with 👻 New folder	(8	Archive name and narameters	2
🔆 Favorites	^		General Adversal Options Ril	un Rushun Ting Currunt
🧮 Desktop	commons-beanutils-1.9.1.jar	2015/12	Advanced options FII	es backup inne comment
〕 Downloads	commons-collections-3.2.1.jar	2015/9/	Archive name	Browse
🗐 Recent Places	📓 commons-lang-2.6.jar	2016/12	disdemo.zip	•
	🔳 commons-logging-1.1.1.jar	2017/4/		Update mode
🥞 Libraries	📓 ezmorph-1.0.6.jar	2017/4/	Profiles	Add and replace files
Documents	🔟 json-lib-2.4.jar	2017/8/	Andrew Ground	Audio dan antinan
J Music	🔟 TableTester.jar	2018/2/		Poloto filos after archiving
Pictures			CRAK CRAKJ CIF	Create SEX archive
🛃 Videos			Compression method	Create solid archive
			Normal	Add recovery record
🖳 Computer			Dictionary size	Test archived files
🕌 Local Disk (C:)			32 KB •	Lock archive
👝 DataDisk (D:)				
			Split to volumes, size	

Creating a Function

When creating a function, specify an agency to delegate FunctionGraph to access DIS and CloudTable resources.

- Step 1 Log in to the FunctionGraph console, and choose Functions > Function List in the navigation pane.
- Step 2 Click Create Function.
- **Step 3** Select **Create from scratch**, set the function information, and click **Create Function**.

- For Function Type, select Event Function.
- For **Function Name**: Enter a function name, for example, **DISDemo**.
- For Agency, select DISDemo created in Preparation.
- For **Runtime**, select **Java 8**.
- **Step 4** On the function details page, configure the following information:
 - Choose Configuration > Basic Settings, change the handler to com.huawei.cff.TableTester.MyHandler, and click Save.
 - On the Code tab, choose Upload > Local ZIP, upload the disdemo.zip package generated in Packaging the Code.

----End

Modifying Function Configurations

After the function is created, the default memory is 128 MB, and the default timeout is 3s, which are insufficient for the data processing. Perform the following steps to modify the configurations.

- **Step 1** On the **DISDemo** details page, choose **Configuration** > **Basic Settings**, and modify the following information as required:
 - For Memory, select 512.
 - For Execution Timeout, enter 15.
 - Keep other parameters unchanged.
- Step 2 Click Save.

----End

2.4 Adding an Event Source

After creating the function, you can add an event source by creating a DIS trigger. Perform the following procedure:

Step 1 On the DISDemo page, select Configure Test Event on the Code tab, as shown in Figure 2-13.

Figure 2-13 Configuring a test event

Coue	wonitoring	VEISION	AlldSES	Conngurau	UII	
Code S	ource					
📀 Fil	e Edit Set	tings				
🤹 Pr	oject	Configure Test Ev	rent	^	Test	Deploy
Project		Configure Test Ev	ent			

Step 2 In the **Configure Test Event** dialog box, set the test event information, as shown in **Figure 2-14**.

- Select Create new test event.
- Event Template: Select Data Ingestion Service (DIS).
- **Event Name**: Enter an event name, for example, **dis-test**.

Figure 2-14 Test event

Event Templates (19)	* Event Na	me dis-z7hr4m
learch	Q 1 {	
loud Service Event	2	"ShardID": "ShardId-000000000", "Message": {
API Gateway (APIG)	4	"next_partition_cursor": "eyJnZXRJdGVyYXRvclBhcmFtIjp7InN
API Gateway (Dedicated Gateway)	6	{
Cloud Trace Service (CTS)	7	<pre>"partition_key": "shardId_0000000000", "data": "d2VsY29t70==".</pre>
Document Database Service (DDS)	9	"sequence number": "0"
GeminiDB Mongo	10	};;
Data Injection Service (DIS)	11	i "nartition key": "shardId 00000000"
Log Tank Service (LTS)	13	"data": "dXNpbmc=",
Simple Message Notification (SMN)	14 15	"sequence_number": "1" },
Timer	16	{
DMS (for Kafka)	17	"partition_key": "shardId_000000000", "data": "RnVuV3Rnb25IdGEn70==".

Step 3 Click Create.

----End

2.5 Processing Data

Perform the following procedure to process simulated stream data:

Step 1 On the **DISDemo** page, select test event **dis-test**, and click **Test** to test the function, as shown in **Figure 2-15**.

Figure 2-15 Configuring a test event

Code Source	
🥪 File Edit Settings	
I dis-test	✓ Test Deploy

Step 2 After the function is executed successfully, check the logs shown in **Figure 2-16**. For all logs of the function, go to the **Logs** tab page.

Figure 2-16 Function execution result

amp":1520234900307, "\$":"c2hhcmRJZF8wMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDA	313
2018-03-05 07:26:25.212+00:00 - request id: 27cba082-f68e-40ff-a575-803021e6457b	
2018-03-05 07:26:25.212+00:00 - partition_key : shardId_0000000000 sequence_number : 3 data : c2VydmljZQ==	Inserted data
_2018-03-05 07:26:25.212+00:00 - request id: 27cba082-f68e-40ff-a575-803021e6457b	moortou uutu
2018-03-05 07:26:25.213+00:00 - Insert data : {"Row": [{"key": "cm93Mu==","Cell": [{"column": "ZjE6c2VxdWVuY2VfbnVtYmVy","\$":	"Mw=="},{"column": "ZjE6cGFydGl0aW9uX2tleQ==","\$": "c2hhcmRJZF8
wHDAwHDAwHDAw"},{"column": "ZjE6ZGF8YQ==","\$": "c2VydmljZQ=="}]}]}	
2018-03-05 07:26:25.213+00:00 - request id: 27cba082-f68e-40ff-a575-803021e6457b	
2018-03-05 07:26:25.213+00:00 - Insert url : http://100.125.1.131:8080/v1.0/7aaef7e2cebf4287a13d1a3ae5a9e789/clusters/66f383bd	-dd8a-4020-b613-61e57c361568/hbase/cff_cloud_table/row3
2018-03-05 07:26:25.226+00:00 - request id: 27cba082-f68e-40ff-a575-803021e6457b	
2018-03-05 07:26:25.227+00:00 - HTTP/1.1 200 OK	
2018-03-05 07:26:25.228+00:00 - log an empty string	Data query address
2018-03-05 07:26:25.228+00:00 - request id: 27cba082-f68e-40ff-a575-803021e6457b	Data query audiene
2018-03-05 07:26:25.229+00:00 - URL: http://100.125.1.131:8080/v1.0/7aaef7e2cebf4287a13d1a3ae5a9e789/clusters/66f383bd-dd8a-40	20-b613-61e57c361568/hbase/cff_cloud_table/row3
2018-03-05 07:26:25.238+00:00 - request id: 27cba082-f68e-40ff-a575-803021e6457b	
2018-03-05 07:26:25.239+00:00 - HTTP/1.1 200 0K	Data guaried an ClaudTable
2018-03-05 07:26:25.239+00:00 - request id: 27cba082-f68e-40ff-a575-803021e6457b	Data queried on Cloud rable
2018-03-05 07:26:25.239+00:00 - {"Row":[{"key":"cm03Mu==","Cell":[{"column":"ZjE6ZGF0YQ==","timestamp":1520234900335,"\$":"c2Vy	dmljZQ=="},{"column":"ZjE6cGFydGl0aW9uX2tleQ==","timestamp":152
0234900335,"\$":"c2hhcmRJZF8wHDAwHDAwHDAwHDAwHDAwHAw"},{"column":"ZjE6c2VxdWVuY2VfbnVtYmVy","timestamp":1520234900335,"\$":"Hw=="}]}]}	
2010 02 05 15:25:25 247:00:00 5inich populat '27cha002 660a 4066 a575 00202106457h' duration: 6340 052mc billing duration: 6	AGGer momony used: 160.203MP

----End

3 Integrating with LTS to Analyze Logs in Real Time

Introduction Preparation Building a Program Adding an Event Source Processing Log Data Other Application Scenarios

3.1 Introduction

Scenarios

Quickly collect, process, and convert task logs of servers, such as ECSs, through Log Tank Service (LTS).

Obtain log data based on an LTS trigger created on FunctionGraph, analyze and process key information in the logs by using a customized function, and then filter alarm logs.

Use SMN to push alarm messages to service personnel by SMS message or email.

Store processed log data in a specified OBS bucket for subsequent processing. The processing workflow is shown in **Figure 3-1**.

Figure 3-1 Processing workflow



Values

- Quickly collects and converts logs through LTS.
- Processes and analyzes data in response to log events in a serverless architecture, which features automatic scaling, no operation and maintenance, and pay-per-use billing.
- Sends alarm notifications through SMN.

3.2 Preparation

Collecting and Storing Logs

- Create a log group, for example, **polo.guoying** on the LTS console. For details, see **Creating a Log Group**.
- Create a log stream, for example, lts-topic-gfz3 on the LTS console. For details, see Creating a Log Stream.
- Configure an agent to collect logs from servers, such as ECSs, to a specified log group. For details, see **Installing the ICAgent**.

Pushing Alarm Messages

- Create a topic named fss_test on the SMN console. For details, see Creating a Topic.
- Add subscriptions to the **fss_test** topic to push alarm messages. For details, see **Adding a Subscription**.
- Define an environment variable named **SMN_Topic** with value **fss_test** to push alarm messages to the subscription endpoints under the **fss_test** topic.

D NOTE

Alarm messages of a subscribed topic can be pushed through email, SMS messages, and HTTP/HTTPS.

In this example, when log events trigger the specified function through an LTS trigger, the function filters alarm logs and pushes alarm message to the subscription endpoints.

Processing Cloud Data

Create an OBS bucket and object, and configure event notifications.

1. Create a bucket and an object on the OBS console, as shown in **Figure 3-2**. For details, see **Creating a Bucket**.

Figure 3-2 Creating a bucket

< 🏐 obs-swx5322	98 👌 Standard Single-AZ storage 8238238238238238238238238238238238238238		🕑 Task Center
Overview Objects Metrics NEW	Libbage Badistics <u>A</u> View Toge Analyse Storage Topul v Thatfic Topul Covinas/Turlic v Requests Topul v	Alarms • Critical	Ø Alarm Details
Basic Configurations Domain Name Ngmt Teoping	D types Much-over-north productions 20.4 kg Much-over-north productions 135 Much-over-north productions	Major Minor	0
Permissions Bucket Policies		Warning	0
Data Security	Domain Name Details	Basic Information	
CORS Rules	Type Domain Name Protocol Operation	Objects 0	
URL Validation	Ensport O SEVERIZENZENZENZENZENZENZEN	Access Private	
Disaster	Access Domain Name () obd isStatis	Cluster Type Public cluster	
Recovery & Backup	Static website hosting domain name HTTPSHTTP 🖨 Configure	Enterprise Project default	
Cross-Region Replication		Account ID 15+	NANAN
Data Management Lifecycle Rules	Basic Configurations		

D NOTE

Name the bucket as **logstore** and the object as **log.txt** to store log data.

Creating an Agency

- 1. Log in to the Identity and Access Management (IAM) console.
- 2. On the IAM console, choose **Agencies** from the navigation pane, and click **Create Agency** in the upper right corner.

Figure 3-3 Creating an agency

IAM	Agencies ③					[Create Agency
Users						_	
User Groups	Delete Agencies available for creation: 4	Dents Apercies available for conditor: 438					
Permissions ~	Al v Q Entr	All v] [Q. Erktran agostoy name.					
Projects	Agency NameID ()	Delegated Party ()	Validity Period 😑	Created 🖨	Description ()	Operation	
Agencies Identity Providers	swt_ee_trast	Cloud service SoftWare Repository for Container (SWR)	Unlimited 	Apr 17, 2024 15:38:52 GMT+08:00	Create by SWR Team. To ensure services	Authorize Modify Delete	
Security Settings		Account	1 day				

- 3. Configure the agency.
 - For **Agency Name**: Enter an agency name, for example, **LtsOperation**.
 - For Agency Type, select Cloud service.
 - For Cloud Service, select FunctionGraph.
 - For Validity Period, select Unlimited.
 - For **Description**: Enter the description.

4. Click **Next**. On the displayed page, search for **LTS Administrator** and **SMN Administrator** in the search box on the right and select them.

NOTE

LTS Administrator depends on **Tenant Guest**. When you select the former, the latter will also be selected.

5. Click **Next** and select the application scope of the permissions based on service requirements.

3.3 Building a Program

Download fss_examples_logstore_warning.zip to create an alarm log extraction function from scratch.

Creating a Function

Create a function by uploading the **sample code package** to extract logs. Select the Python 2.7 runtime and the agency **LtsOperation** created in **Creating an Agency**. For details about how to create a function, see **Creating an Event Function**.

This function performs Base64 decoding on received log event data, extracts alarm logs containing keyword **WRN**, **WARN**, **ERR**, or **ERROR**, and then stores the extracted logs in the specified OBS bucket. Set log extraction conditions based on the content of your service logs.

Setting Environment Variables

On the **Configuration** tab page of the preceding function, set environment variables to pass the bucket address, bucket name, and object name, as shown in **Table 3-1**.

Environment Variable	Description
obs_address	OBS endpoint. To obtain the OBS endpoint, see Regions and Endpoints .
obs_store_bucket	Name of the destination bucket for storing logs.
obs_store_objName	Name of the target file for storing logs.
SMN_Topic	SMN topic.
region	Name of your region. To obtain the region name, see Regions and Endpoints .

Table 3-1 Environment variables

)

Set the environment variables by following the procedure in **Environment Variables**.

3.4 Adding an Event Source

Create an LTS trigger by using the log group and log topic created in **Preparation**, and configure the trigger information according to **Figure 3-4**.

Figure 3-4 Creating an LTS trigger

lts-topic-afz3

Create Trigg	er	
Trigger Type	Log Tank Service (LTS)	~
	The total number of OPENSOURCEKA KAFKA triggers cannot exceed 10 for a	FKA, RABBITMQ, DIS, LTS, DDS, TIMER, DMS, GAUSSMON function version, you have created 0 such triggers.
★ Log Group	polo.guoying	✓ Q. Create Log Group [2]

When the accumulated log size or log retention period meets a specified threshold, LTS log data is consumed, which triggers the function associated with the log group.

✓ Q Create Log Stream [2]

3.5 Processing Log Data

★ Log Stream

Email notifications will be received from SMN if alarm logs containing keyword **WRN**, **WARN**, **ERR**, or **ERROR** are generated, as shown in Figure 3-5. You can also view details of the alarm logs by opening the **log.txt** file in the specified bucket, as shown in Figure 3-6.

Figure 3-5 Email notification

Get warning message. The content of message is: ["(\"p\':\'192.168.1.98\', \'line_no\': 616, \'host_name\': \'ecs-testagent.novaloca(\', \'time\': 1530009653059, \'path\': \'/usr/local/telescope/log/common.log\', \'message\': \'2018-06-23/184053 [WRN] [config.go:82] The projectid or instanceld of config.json is not consistent with metadata, use metadata.\\\(\\',\'\'gu_u\d':\'\'esd30530-7361.1e88b05-2864486270\'']

Figure 3-6 Alarm log details

"\"{\\\"message\\\";\\\"2018-06-26/18:40:53 [WRN] [config.go:82] The projectId or instanceId of config.json is not consistent with metadata, use metadata.\\\\n\\",""time\\\":1530009653059,\\\"host_name\\\";\\\"ecs-testagent.novalocal\\\",\\\"ip\\\":\\\"192.168.1.98\\\",\\\"path\\\": \\\"/usr/local/telescope/log/common.log\\", \\\"log_uid\\\":\\\"663d6930-792d-1188-8b09-286ed488ce70\\\",\\\"ine_no\\\":fi6}\""

On the **Monitoring** tab page of the function, check the number of invocations, as shown in **Figure 3-7**.



Figure 3-7 Function metrics

3.6 Other Application Scenarios

FunctionGraph and Log Tank Service (LTS) can be used to process cloud logs, push alarm messages, and store logs in a specified Object Storage Service (OBS) bucket. You can use FunctionGraph and LTS in multiple scenarios. For example, you can create a timer trigger to periodically analyze and process log data in an OBS bucket.

4 Integrating with CTS to Analyze Login/ Logout Security

Introduction Preparation Building a Program Adding an Event Source Processing Operation Records

4.1 Introduction

Scenarios

Collect real-time records of operations on cloud resources.

Create a Cloud Trace Service (CTS) trigger to obtain records of subscribed cloud resource operations; analyze and process the operation records, and report alarms.

Use SMN to push alarm messages to service personnel by SMS message or email. The processing workflow is shown in **Figure 4-1**.



Figure 4-1 Processing workflow

Values

- Quickly analyzes operation records collected by CTS and filters out operations from specified IP addresses.
- Processes and analyzes data in response to log events in a serverless architecture, which features automatic scaling, no operation and maintenance, and pay-per-use billing.
- Sends alarm notifications through SMN.

4.2 Preparation

Enabling CTS

Configure a tracker on CTS, as shown in **Figure 4-2**. For details, see **Configuring a Tracker**.

Figure 4-2 Configuring a tracker

< Create Tracker		1 Basi
Basic CTS functions are for	or free. However, you will be charged for LTS, OBS, DEW, or SMIN services you use if you enable the trace analysis, trace transferring, or key event notification function. Learn more	C
Basic Information		
* Tracker Name	lest	
	The name must start with a letter or digit. Only letters, digits, hyphens (-), and underscores (_) are allowed. The name of the data tracker cannot be system or system-trace.	
Enterprise Project (?)	default v Q View Projects [2]	

Creating an Agency

- **Step 1** Log in to the **IAM console**, and choose **Agencies** in the navigation pane.
- **Step 2** On the **Agencies** page, click **Create Agency**.
- **Step 3** Set the agency information.
 - For Agency Name: Enter an agency name, for example, serverless_trust.
 - For Agency Type, select Cloud service.
 - For Cloud Service, select FunctionGraph.
 - For Validity Period, select Unlimited.
 - For **Description**, enter a description.
- Step 4 Click Next. On the Select Policy/Role page, select CTS Administrator and SMN Administrator.

NOTE

- **SMN Administrator**: Users with this permission can perform any operation on SMN resources.
- **CTS Administrator** depends on **Tenant Guest**. When you select the former, the latter will also be selected.
- **Step 5** Click **Next**, select an authorization scope that meets your service requirements, and click **OK**.

----End

Pushing Alarm Messages

- Create a topic named **cts_test** on the SMN console. For details, see **Creating a Topic**.
- Add subscriptions to the cts_test topic to push alarm messages. For details, see Adding a Subscription.

NOTE

Alarm messages of a subscribed topic can be pushed through emails, SMS messages, and HTTP/HTTPS.

In this example, when operation log events trigger the specified function, the function filters operations that are performed by users not in the IP address whitelist, and pushes alarm messages to the subscription endpoints.

4.3 Building a Program

Download index.zip to create an alarm log analysis function from scratch.

Creating a Function

Create a function by uploading the **sample code package** to extract logs. Select the Python 2.7 runtime and the agency **serverless_trust** created in **Creating an Agency**. For details about how to create a function, see **Creating an Event Function**.

This function analyzes received operation records, filters logins or logouts from unauthorized IP addresses using a whitelist, and sends alarms under a specified SMN topic. This function can be used to build an account security monitoring service.

Setting Environment Variables

On the **Configuration** tab page of the function details page, set the environment variables listed in **Table 4-1**.

Environment Variable	Description
SMN_Topic	SMN topic.
RegionName	Region name.
IP	IP address whitelist.

 Table 4-1 Environment variables

Set the environment variables by following the procedure in **Environment Variables**.

4.4 Adding an Event Source

Create a CTS trigger, as shown in Figure 4-3.

 \sim

Figure 4	4-3 Crea	ating a	CTS	trigger
----------	-----------------	---------	-----	---------

Create Trigger				,		
Trigger Type	Cloud Trace Service (CTS)	~				
	CTS collects operation records of subscribed cloud resources. After you create a CTS trigger for a function, operation records of the subscribed cloud services are passed as an input parameter to invoke the function.					
	You have created 0 CTS trigger	s (max. CTS triggers: 10).				
	CTS has been enabled	and you can create CTS trig	gers.			
★ Event Notification Name	cts_test Enter 1 to 64 characters. Only le	etters, digits, and underscore) es (_) are allowed.			
* Custom Operations	A maximum of 10 services and 100 op	perations can be selected.				
	Service Type	Resource Type	Trace Name	Op		
	IAM V	user X V	login × ✓ logout ×	Delete		
	Add Operation					

CTS records the logins and logouts of users on IAM.

4.5 Processing Operation Records

The function runs in response to account logins and logouts to filter those not from the IP address whitelist, and sends a message or email through SMN, as shown in **Figure 4-4**.

Figure 4-4 Email notification

```
Illegal operation[ IP:10.65.56.139, Action:login]
```

The email contains the unauthorized IP address and user operation (login or logout).

On the **Monitoring** tab page of the function, check the number of invocations, as shown in **Figure 4-5**.

Figure 4-5 Function metrics

Code Monitoring Version Aliases Configuration		
Metros Logs		
You can query the monitoring information of different versions that have been published for a function.		
	Feedbed	View Alarm Rule View CES Matrice Last Gap Last 3 days Custom (Hitlenai: 1 hour
Invocations	13 Q I	Duration Do I
Count 15 times		Duration 0 ms
Unit: times		Units ms
21		1
18		
12 19		
3		
0 14.00 16.00 18.00 20.00 22.00 00.00 02.00 04.00 06.00 08.00 10	00 12:00 14:00	0

5 Periodically Starting or Stopping Huawei Cloud ECSs

Application Scenario

If you need to start or stop your ECSs at specified time, you can use FunctionGraph to call the corresponding ECS APIs.

- Startup node: VM that needs to be started periodically.
- Shutdown node: VM that needs to be stopped periodically.

Prerequisites

- 1. Obtain the program package for periodically starting or stopping ECSs.
- 2. Create the **EcsOperation** agency and grant it the **ECS FullAccess** permission. For details, see **Creating an Agency**.

Creating an Agency

- 1. Log in to the IAM console.
- 2. On the IAM console, choose **Agencies** from the navigation pane, and click **Create Agency** in the upper right corner.

Figure 5-1 Creating an agency

IAM	Agencies ()							Create Agency
Users								
User Groups	Danta Agencies analitable for creation: 433							
Permissions ~	All v) (O, Entra an agency name.							
Projects		Agency Name10 (e)	Delegated Party ()	Validity Period \varTheta	Created 🖨	Description ()	Operation	
Agencies Identity Providers		without	Cloud service SoftWare Repository for Container (SWR)	Unimited 	Apr 17, 2024 15:30 52 GMT+00:00	Create by SWR Team. To ensure services	Authorize Modily Del	**

- 3. Configure the agency.
 - For **Agency Name**: Enter an agency name, for example, **EcsOperation**.
 - For Agency Type, select Cloud service.
 - For Cloud Service, select FunctionGraph.
 - For Validity Period, select Unlimited.
 - **Description**: Enter the description.
- 4. Click **Next**. On the displayed page, search for **ECS FullAccess** in the search box on the right and select it.

Figure 5-2 Selecting the permission

andle contrates be unapped a second state.	01000 1010
Vew Selected (1) Copy Permissions from Another Project	All policies/toles v Fuzzy search v ecs full X : Q
Policy/Role Name	Type
CS FullAcces All permissions of ECB service.	System-defined policy

5. Click **Next**, select an authorization scope that meets your service requirements, and click **OK**.

Building a Program

Step 1 Create a function.

To create a function for periodically starting or stopping ECSs, upload the program package (for **starting** or **stopping** ECSs) and select the **EcsOperation** agency. For details, see **Creating a Function**.

Select the Python 3.6 runtime and the agency **EcsOperation** created in the previous step.

Step 2 Set environment variables.

On the **Configuration** tab page, set environment variables according to **Table 5-1**.

Environment Variable	Description
region	Region where your ECSs are located, for example, ap- southeast-1.
projectId	ID of the project to which the ECS belongs. For details about how to obtain the project ID, see Obtaining a Project ID .
whiteLists	 If you want to periodically start certain ECSs, specify the IDs of the ECSs that need to be started and separate them with commas (,).
	 If you want to periodically stop certain ECSs, specify the IDs of the ECSs that need to be stopped and separate them with commas (,).
type	Stop type, which needs to be configured when you want to periodically start ECSs. Options:
	SOFT: normal ECS stop (default) HARD: forcible ECS stop
type	 If you want to periodically start certain ECSs, specify the IDs of the ECSs that need to be started and separate them with commas (,). If you want to periodically stop certain ECSs, specify the IDs of the ECSs that need to be stopped and separate them with commas (,). Stop type, which needs to be configured when you want to periodically start ECSs. Options: SOFT: normal ECS stop (default) HARD: forcible ECS stop

 Table 5-1
 Environment variables

Set the environment variables by following the procedure in **Configuring Environment Variables**.

D NOTE

- In the current example, there are no requirements on the region where the function is executed. If the function and the ECS to be started or stopped are in the same region, perform the preceding operations. If they are in different regions, for example, the function is running in CN-Hong Kong and the ECS to be started or stopped is located in AP-Bangkok, change the values of **projectId** and **region** to those of AP-Bangkok, **obtain an** AK/SK and add them to the environment variables, and then delete the configured agency.
 - In AK/SK-based authentication, AK/SK is used to sign requests and the signature is then added to the request headers for authentication.
 - AK: access key ID, which is a unique identifier used in conjunction with a secret access key to sign requests cryptographically.
 - SK: secret access key used together with an AK to sign requests cryptographically. It identifies a request sender and prevents the request from being modified.

Edit Environment Variable Key Value Encrypted (?) **388888** Delete region projectId whiteLists Delete type SOFT Delete ak Delete sk Delete + Add

Figure 5-3 Setting environment variables

Available environment variables for addition: 14

- If a large number of ECSs need to be started or stopped, increase the execution timeout for your function.
- Environment variables in **Table 5-1** except **whiteLists** are mandatory. **whiteLists** indicates the comma-separated IDs of the ECSs to be started or stopped.
- Endpoint of the ECS service in the format of "{region}.{domain}", for example, apsoutheast-1.myhuaweicloud.com. To obtain the endpoint information, see Regions and Endpoints.

Step 3 Add a dependency.

On the Code tab, add dependency huaweicloudsdk_ecs_core_py3.6.

For more information, see **Configuring Dependencies for a Function**.

NOTE

If **huaweicloudsdk_ecs_core_py3.6** is not available in your region, contact customer service.

----End

Adding an Event Source

Create a timer trigger and set the trigger parameters according to **Figure 5-4**.

Figure 5-4 Cre	ating a timer	trigger
----------------	---------------	---------

Create Trigger

Trigger Type	Timer ~
★ Timer Name	Timer-u7q9
	Enter 1 to 64 characters, starting with a letter. Only letters, digits, hyphens (-), and underscores (_) are allowed.
* Rule	Fixed rate
	0 0 0,8,12,18 ** ?
* Enable Trigger	
Additional Information ③	
	0/2,048 2

6 Building an HTTP Function with Spring Boot

Introduction

This chapter describes how to deploy services on FunctionGraph using Spring Boot.

Usually, you may build Spring Boot applications using **SpringInitializr** or IntelliJ IDEA. This chapter uses the Spring.io project in **https://spring.io/guides/gs/rest-service/** as an example to deploy an HTTP function on FunctionGraph.

Procedure

To deploy an existing project to FunctionGraph, change the listening port of the project to **8000**, and create a file named **bootstrap** in the same directory as the JAR file to include the command for executing the JAR file.

In this example, a Maven project created using IntelliJ IDEA is used.

Building a Code Package

1. Open the Spring Boot project and click **package** in the **Maven** area to generate a JAR file.





2. Set the web port to **8000 (do not change this port)** using the **application.properties** file or specify the port during startup. HTTP functions only support this port.

Figure 6-2 Configuring port 8000



3. Create a file named **bootstrap** in the same directory as the JAR file, and enter the startup parameters.

/opt/function/runtime/java11/rtsp/jre/bin/java -jar -Dfile.encoding=utf-8 /opt/function/code/rest-service-complete-0.0.1-SNAPSHOT.jar

NOTE

The Java runtime environment can be directly invoked in the function, and no additional installation is required.

4. Compress the JAR file and **bootstrap** file into a ZIP package.

Creating an HTTP Function and Uploading Code

Create an HTTP function and upload the ZIP file. For details, see **Creating an HTTP Function**.

Verifying the Result

- Using a test event
 - a. On the function details page, select a version and click **Configure Test Event**.
 - b. On the **Configure Test Event** page, select the event template, and modify the **path** and **pathParameters** parameters in the template to construct a simple GET request.

Figure 6-3 Configuring a test event



- c. Click Create.
- d. Click **Test** to obtain the response.

When debugging a function, increase the memory size and timeout, for example, increase them to 512 MB and 5s.

Figure 6-4 Viewing the returned result

Code Montoning Version	Alases Conguration	
Code Source		S Feedback Uptase -
👻 File Edit Settings		View Dependencies Open in CodeVrts IDE Online 🍾
Project apig-773zp	V Test Deplay	
Project B position B metanoscorptine 633 (pol.)	ionny 1 1 //st/faction/nation/jeall/tip/jea/lio/jea-jar-dtile.ecolog-otf-4	Decision Result *
		2022-011-02 3130, 72 Prol 115 — [no4:00:ext] to kits any ADB pathwell Privet. Completed Initiation in 1 ms 2022-011103 2021 Princh Index append 1556053-046-64 61 61 #-05686660507, Junitori 612 227ms, billing Junitori 605m, mimory aude 123 23908, billing memory 51208 Summary

- Using an APIG trigger
 - a. Create an APIG trigger by referring to **Using an APIG Trigger**. Set the authentication mode to **None** for debugging.
 - b. Copy the generated URL, add the request parameter greeting? name=fg_user to the end of the URL (see Figure 6-5), and access the URL using a browser. The response shown in the following figure is displayed.

Figure 6-5 Invoking the function



The default APIG trigger URL is in the format "*Domain name*/*Function name*". In this example, the URL is **https://your_host.com/ springboot_demo**, where the function name **springboot_demo** is the first part of the path. If you send a GET request for **https://** *your_host.com*/**springboot_demo/greeting**, the request address received by Spring Boot contains **springboot_demo/greeting**. If you have uploaded an existing project, you cannot access your own services because the path contains a function name. To prevent this from happening, use either of the following methods to annotate or remove the function name:

 Method 1: Modify the mapping address in the code. For example, add the first part of the default path to the GetMapping or class annotation.

Figure 6-6 Modifying the mapping address



 Method 2: Click the trigger name to go to the APIG console, and delete the function name in the path.

FAQ

1. What Directories Are Accessible to My Code?

An uploaded code package is stored in the **/opt/function/code/** directory of the function (runtime environments, compute resources, or containers). However, the directory can only be read and cannot be written. If some data must be written to the function during code running and logged locally, or your dependency is written by default to the directory where the JAR file is located, use the **/tmp** directory.

2. How Are My Logs Collected and Output?

Function instances that have not received any requests during a specific period of time will be deleted together with their local logs. You will be unable to view the function logs during function running. Therefore, in addition to writing logs to your local host, output logs to the console by setting the output target of Log4j to **System.out** or by using the **print** function.

Logs output to the console will be collected. If you have enabled LTS, the logs will also be stored in LTS for near real-time analysis.

Suggestion: **Enable LTS**, and click **Go to LTS** to view and analyze logs on the **Real-Time Logs** tab page.

Figure 6-7 Accessing LTS for log analysis

Code	Monitoring	Version	Aliases	Configuration	
Metrics	Logs				
1 Funct	lionGraph records a	Il requests proce	essed by your fur	inction and automatically stores the logs in LTS. Verify code by inserting custom logging statements. The following table lists the function logs. View more c	an LTS. 🕑

3. What Permissions Does My Code Have?

Similar to common event functions, code does not have the **root** permission. Code or commands requiring this permission cannot be executed in HTTP functions.

4. How Do I Package Spring Boot Projects of Multiple Modules?

Configure the following to package these Spring Boot projects.

				-
<build></build>				
<plugins></plugins>				
<plugin></plugin>				
<groupid></groupid>	org.springfra	mework	.boot <td>old></td>	old>
<artifactId>spring-boot-maven-plugin</artifactId> <configuration> <mainClass>com.example.YourServiceMainClass</mainClass> </configuration> <executions> <executions> <goals> <goal>repackage</goal> </goals> </execution> </executions> </plugin> </plugins> </build>

Creating a FunctionGraph Backend API That Uses a Custom Authorizer

Introduction Resource Planning Building a Program Adding an Event Source Debugging and Calling the API

7.1 Introduction

In addition to IAM and app authentication, APIG also supports custom authentication with your own system, which can better adapt to your business capabilities.

This chapter guides you through the process of creating a FunctionGraph API that uses a custom authorizer.

Solution

- Log in to the FunctionGraph console, and create a function for custom authentication.
- Create a service function.
- Create an API group on the APIG console.
- Create an API and configure a custom authorizer and a FunctionGraph backend for it.
- Debug the API.

D NOTE

After you complete the operations in this tutorial, your Huawei Cloud account will have the following resources:

- 1. An API group storing APIs
- 2. A custom authentication function
- 3. A service function
- 4. An API with a custom authorizer and a FunctionGraph backend

7.2 Resource Planning

Ensure that the following resources are in the same region.

Table	7-1	Resource	р	lanning
-------	-----	----------	---	---------

Resource	Quantity
API group	1
Custom authentication function	1
Service function	1
API	1

7.3 Building a Program

Creating an API group

Before creating a function and adding an event source, create an API group to store and manage APIs.

NOTE

Before enabling APIG functions, buy a gateway by referring to **Buying a Gateway**.

- **Step 1** Log in to the APIG console, choose **API Management** > **API Groups** in the navigation pane, and click **Create API Group** in the upper right.
- Step 2 Select Create Directly, set the group information, and click OK.
 - **Name**: Enter a group name, for example, **APIGroup_test**.
 - **Description**: Enter a description about the group.

----End

Creating a Custom Authentication Function

Frontend custom authentication means APIG uses a function to authenticate received API requests. To authenticate API requests by using your own system,

create a frontend custom authorizer in APIG. Create a FunctionGraph function with the required authentication information. Then use it to authenticate APIs in APIG.

This section uses the header parameter **event["headers"]** as an example. For the description about request parameters, see **Request Parameter Code Example**.

- Step 1 In the left navigation pane of the management console, choose Compute > FunctionGraph to go to the FunctionGraph console. Then choose Functions > Function List in the navigation pane.
- Step 2 Click Create Function.
- **Step 3** Set the function information, and click **Create Function**.
 - Template: Select Create from scratch.
 - Function Type: Select Event Function.
 - Function Name: Enter a function name, for example, apig-test.
 - Agency: Select Use no agency.
 - Runtime: Select Python 2.7.
- **Step 4** On the function details page that is displayed, click the **Code** tab and copy the **example request parameter code** to the online editor, and click **Deploy**.
- **Step 5** Click **Configure Test Event**, and select an event template. Modify the template as required, and click **Create**. In this example, add "**auth**":"**abc**" to "**headers**".

Figure 7-1 Configuring a test event



Step 6 Click **Test**. If the result is **Execution successful**, the function is successfully created.

Figure 7-2 Viewing the execution result

```
Execution successful
Function Output
{
    "body": "{\"status\": \"allow\", \"context\": {\"user\": \"success\"}}",
    "statusCode": 200
}
```

----End

Creating a Custom Authorizer

Create a custom authorizer in APIG and connect it to the frontend custom authentication function.

- Step 1 In the left navigation pane of the management console, choose Middleware > API Gateway to go to the APIG console. In the navigation pane, choose API Management > API Policies. On the Custom Authorizers tab, click Create Custom Authorizer.
- **Step 2** Configure basic information about the custom authorizer according to the following figure.
 - Name: Enter a name, for example, Authorizer_test.
 - **Type**: Select **Frontend**.
 - Function URN: Select apig-test.

Create Custom Author	izer	
* Name	Authorizer_test	
★ Type	Frontend Backend	
* Function URN	um:fss:d Select	
* Version/Alias	Version • LATEST	•
* Max. Cache Age (s) ?	- 0 +	
Identity Sources (?)	Parameter Location Parameter Name	Operation
	Add Identity Source	
Send Request Body		
User Data	Enter the user data.	
		0/2,048
	The user data will be stored as plaintext. Be careful with the information that you include here.	ne

Figure 7-3 Creating a custom authorizer

Step 3 Click OK.

----End

Creating a Backend Service Function

APIG supports FunctionGraph backends. After you create a FunctionGraph backend API, APIG will trigger the relevant function, and the function execution result will be returned to APIG.

- **Step 1** Create a service function by referring to **Creating a Custom Authentication Function**. The function name must be unique.
- **Step 2** On the **Code** tab of the function details page, copy the following code to the online editor, and click **Deploy**.

-*- coding:utf-8 -*-
import json
def handler (event, context):
body = " <html><title>Functiongraph Demo</title><body>Hello, FunctionGraph!</body></html> '
print(body)
return {
"statusCode":200,
"body":body,
"headers": {
"Content-Type": "text/html",
},

"isBase64Encoded": False
}

----End

Request Parameter Code Example

The following are the requirements you must meet when developing FunctionGraph functions. Python 2.7 is used as an example.

The function must have a clear API definition. Example:

def handler (event, context)

- **handler**: name of the entry point function. The name must be consistent with that you define when creating a function.
- event: event parameter defined in JSON format for the function.
- context: runtime information provided for executing the function. For details, see SDK APIs.

event supports three types of request parameters in the following formats:

- Header parameter: event["headers"]["Parameter name"]
- Query string: event["queryStringParameters"]["Parameter name"]
- Custom user data: event["user_data"]

The three types of request parameters obtained by the function are mapped to the custom authentication parameters defined in APIG.

- Header parameter: Corresponds to the identity source specified in Header for custom authentication. The parameter value is transferred when the API that uses custom authentication is called.
- Query string: Corresponds to the identity source specified in **Query** for custom authentication. The parameter value is transferred when the API that uses custom authentication is called.
- Custom user data: Corresponds to the user data for custom authentication. The parameter value is specified when the custom authorizer is created.
- The function response cannot be greater than 1 MB and must be in the following format:

```
{ "statusCode":200,
```

```
"body": "{\"status\": \"allow\", \"context\": {\"user\": \"abc\"}}"
}
```

The **body** field is a character string, which is JSON-decoded as follows:

```
"status": "allow/deny",
"context": {
    "user": "abc"
}
```

The **status** field is mandatory and is used to identify the authentication result. The authentication result can only be **allow** or **deny**. **allow** indicates that the authentication is successful, and **deny** indicates that the authentication fails.

The **context** field is optional and can only be key-value pairs. The key value cannot be a JSON object or an array.

The **context** field contains custom user data. After successful authentication, the user data is mapped to the backend parameters. The parameter name in **context** is case-sensitive and must be the same as the system parameter name. The parameter name must start with a letter and can contain 1 to 32 characters, including letters, digits, hyphens (-), and underscores (_).

Example Header Parameter

```
# -*- coding:utf-8 -*-
import json
def handler(event, context):
  if event["headers"].get("auth")=='abc':
     resp = {
        'statusCode': 200,
        'body': json.dumps({
           "status":"allow",
           "context":{
              "user":"success"
           }
        })
     }
  else:
     resp = {
        'statusCode': 200,
        'body': json.dumps({
           "status":"deny",
        })
     }
  return json.dumps(resp)
```

Example Query String

```
# -*- coding:utf-8 -*-
import json
def handler(event, context):
  if event["queryStringParameters"].get("test")=='abc':
     resp = {
        'statusCode': 200,
        'body': json.dumps({
           "status":"allow",
           "context":{
              "user":"abcd"
           }
        })
     }
  else:
     resp = {
        'statusCode': 200,
        'body': json.dumps({
           "status":"deny",
        })
     }
  return json.dumps(resp)
```

Example User Data

```
# -*- coding:utf-8 -*-
import json
def handler(event, context):
    if event.get("user_data")=='abc':
        resp = {
            'statusCode': 200,
            'body': json.dumps({
                "status":"allow",
                "context":{
                "user":"abcd"
            }
        })
```

```
}
else:
  resp = {
    'statusCode': 200,
    'body': json.dumps({
        "status":"deny",
     })
  }
return json.dumps(resp)
```

7.4 Adding an Event Source

Creating an API

After creating an API group, custom authentication function, and backend function, create a FunctionGraph backend API that uses a custom authorizer by performing the following steps:

- **Step 1** Log in to the APIG console, choose **API Management** > **APIs** in the navigation pane, and click **Create API** in the upper right.
- **Step 2** Configure the basic information according to Figure 7-4 and Figure 7-5.
 - API Name: Enter a name, for example, API_test.
 - Group: Select API group APIGroup_test.
 - URL: Set Method to ANY, Protocol to HTTPS, and Path to /testAPI.
 - Gateway Response: Select default.
 - Authentication Mode: Select Custom.
 - Custom Authorizer: Select Authorizer_test.

Figure 7-4 Configuring frontend definition

Frontend Definition					
* API Name	API_test				
	Enter a string of 3 to 255 cl	haracters starting with a letter. Only le	etters, digits, hyphens (-), under	rscores (_), periods (.), slash (/), colons	(:), and parentheses (()) are allow
* Group	APIGroup_test				
* URL	Method Protoc	col Subdomain Name		Path	
	ANY HTT	PS ▼ a5e	f71db8b4f9f4e.apic.cn-e	/testAPI	0
* Gateway Response	default	•			
Matching	Exact match	Prefix match			
5	API requests will be forwar	rded to the specified path.			
Tags	Select or enter tags.				
Description	Enter a description.				
			0/255		
Content Format Type					

Security Configuration	
Visibility 🕐	Public Private
Authentication Mode	App IAM Custom None
	You can create a custom authorizer to control API access using your own authentication system. Security Level:
Custom Authorizer	Authorizer_test C Create Custom Authorizer
CORS	Enable this option to allow restricted resources on a web page to be requested from other domains.

Figure 7-5 Configuring security settings

For more parameters, see Creating an API.

Step 3 Click Next to configure the backend service according to Figure 7-6.

- Backend Type: Select FunctionGraph.
- **Function URN**: Select the created service function.
- Version/Alias: Select the latest version.
- Invocation Mode: Select Synchronous.

Figure 7-6 Configuring the backend service

Backend Configure	ation		
Backend Type	НТТ	P&HTTPS FunctionGraph	Mook
Default Backend		Basic Information	
Backend Policies	(+)	Function Name	apig-test1
		* Function URN	um.fss.c 3:9a Select
		* Version/Alias	Version • latest •
		* Invocation Mode	Synchronous v
		Timeout (ms)	5000
		Backend Authentication (?)	Use custom authorizer for authentication

- Step 4 Click Finish.
- **Step 5** Click **Publish** to publish the API in the RELEASE environment.

Figure 7-7 Publishing an API

API test
OC Create Policy

Create Po

7.5 Debugging and Calling the API

APIG provides online debugging, enabling you to check an API after configuring it.

- Step 1 Log in to the APIG console. In the navigation pane, choose API Management > APIs. Then click API_test, and click Debug.
- **Step 2** Add a header parameter and click **Debug**.

- Parameter Name: Enter auth.
- Parameter Value: Enter abc.

Figure 7-8 Adding a header

Method GET	×		
GET https://a5e1c	fc79e164152b3cf71db8b4f914e.apic.c 3.hu , s.c	mytestAPI	Debug
Parameters	Headers (1)		
Parameter Name		Parameter Value	
auth		abc	

Step 3 Check whether the API response contains the content you have defined in the service function. See Figure 7-9.

Figure 7-9 API response

```
HTTP/1.1 200 OK
Content-Length: 87
Connection: keep-alive
Content-Type: text/html; charset=UTF-8
Date: Tue, 07 Feb 2023 06:39:18 GMT
Server: api-gateway
Strict-Transport-Security: max-age=31536000; includeSubdomains;
X-Apig-Latency: 2140
X-Apig-Latelimit-Api: remain:95,limit:100,time:1 minute
X-Apig-Ratelimit-Api: remain:950,limit:16000,time:1 second
X-Apig-Ratelimit-Api-Allenv: remain:5509,limit:16000,time:1 second
X-Apig-Ratelimit-Api-Allenv: remain:5601,limit:16000,time:1 second
X-Apig-Ratelimit-Lapi-Allenv: remain:5601,limit:16000,time:1 second
X-Apig-Ratelimit-Lapi-Allenv: remain:5601,limit:16000,time:1 second
X-Apig-Ratelimit-Joi-Allenv: remain:5601,limit:1000,time:1 second
X-Apig-Vpstream-Latency: 153
X-Cff-Billing-Duration: 5
X-Cff-Billing-Duration: 5
X-Cff-Request-Id: 4550cdf5-d74-4aa5-ba04-c79f84d436cz
X-Content-Type-Options: nospif
X-Download-Options: nospen
X-Frame-Options: SMCRGIGIN
X-Request-Id: dfa7d5925751f31f12221f45459a1312
X-Reguest-Id: dfa7d5925751f31f12221f45459a1312
X-Respectedion: 1; mode=block;
```

<html><title>Functiongraph Demo</title><body>Hello, FunctionGraph!</body></html>

----End

8 Uploading Files with FunctionGraph and APIG

Introduction Resource Planning Procedure

8.1 Introduction

Scenario

Uploading files, such as run logs and web application images, from devices to cloud servers is a type of common scenarios for websites and applications. These scenarios can be implemented by using function backends and APIG. This chapter uses Node.js and Python as examples to describe how to develop a backend parsing function for obtaining uploaded files.

Constraints

- The file uploaded in a request cannot exceed 6 MB.
- Function logic processing must be within 15 minutes.

8.2 Resource Planning

Table 8-1 Resource planning

Product	Configuration Example
APIG	Region: AP-SingaporeSpecifications: shared gateway or dedicated gateway
FunctionGraph	Region: AP-SingaporeBilling mode: pay-per-use

8.3 Procedure

This solution includes the following steps:

- 1. Create a function to receive and parse uploaded files.
- 2. Bind an APIG trigger to the function for E2E testing.

8.3.1 Node.js

Prerequisites

- You have a Huawei Cloud account and have completed real-name authentication.
- Your Huawei Cloud account is not in arrears and has sufficient balance for the resources involved in this example.

Procedure

- **Step 1** Create a function.
 - Log in to the FunctionGraph console, choose Functions > Function List in the navigation pane, and click Create Function.
 - 2. Select **Create from scratch**, set the function information, and click **Create Function**.
 - Function Type: Select Event Function.
 - **Region**: Select **AP-Singapore**.
 - **Function Name**: Enter a function name, for example, **upload-file-1**.
 - Agency: Select Use no agency.
 - Runtime: Select Node.js 14.18.
 - On the Code tab of the function details page, copy the following code to replace the default code, and click Deploy. const stream = require("stream"); const Busboy = require("busboy");

```
exports.handler = async (event, context) => {
  const logger = context.getLogger()
  logger.info("Function start run.");
  if (!("content-type" in event.headers) ||
     !event.headers["content-type"].includes("multipart/form-data")) {
     return {
        'statusCode': 200,
        'headers': {
           'Content-Type': 'application/json'
        },
         'body': 'The request is not in multipart/form-data format.',
     };
  }
  const busboy = Busboy({ headers: event.headers });
  let buf = Buffer.alloc(0);
  busboy.on('file', function (fieldname, file, filename, encoding, mimetype) {
     logger.info('filename:' + JSON.stringify(filename))
     file.on('data', function (data) {
        logger.info('Obtains ' + data.length + ' bytes of data.')
        buf = Buffer.concat([buf, data]);
```

```
}):
  file.on('end', function () {
     logger.info('End data reception');
  });
});
busboy.on('finish', function () {
  // Data is processed here.
  logger.info(buf.toString());
   return {
     'statusCode': 200.
     'headers': {
        'Content-Type': 'application/json'
      'body': 'ok',
  };
});
// The APIG trigger encodes data using Base64 by default. The data is decoded here.
const body = Buffer.from(event.body, "base64");
var bodyStream = new stream.PassThrough();
bodyStream.end(body);
bodyStream.pipe(busboy);
```

Step 2 Configure a dependency.

}

- 1. Make dependency: To parse uploaded files with busboy, generate dependency **busboy.zip** for Node.js 14.18. If you use another Node.js version, create the corresponding dependency by referring to **Creating a Dependency**.
- Create dependency: In the navigation pane, choose Functions > Dependencies. Then click Create Dependency, configure the dependency information, and click OK.
 - Name: Enter a dependency name, for example, busboy.
 - Code Entry Mode: Select Upload ZIP.
 - Runtime: Select Node.js 14.18.
 - Upload File: Upload the dependency you made.
- 3. Add dependency: On the details page of function **upload-file-1**, click **Add** at the bottom of the **Code** tab. On the **Select Dependency** page, set **Type** to **Private**, select the **busboy** dependency, and click **OK**.
- **Step 3** Create an APIG trigger.
 - On the details page of function upload-file-1, choose Configuration > Triggers.
 - 2. Click Create Trigger, and set Trigger Type to API Gateway (APIG) or API Gateway (Dedicated Gateway). In this example, select API Gateway (APIG).
 - API Name: Retain the default name.
 - API Group: If no API group is available, click Create API Group to create one.
 - Environment: Select RELEASE.
 - Security Authentication: In this example, select None for testing. You
 can select a more secure authentication mode, such as IAM, for your own
 services.
 - Protocol: Select HTTPS.
 - **Timeout (ms)**: Retain the default value **5000**.

Step 4 Perform E2E testing.

The curl tool is used as an example (**curl -F** is mainly used in Linux). You can also use other tools such as Postman. Create a file named **app.log** with any content on your local host. Example:

start something run stop all

Run the following command:

curl -iv {APIG trigger URL} -F upload=@/{Local file path}/app.log

Figure 8-1 Example

On the **Monitoring** tab of the **upload-file-1** function details page, view the file content in the logs. If needed, you can modify the code to save data to OBS or LTS or to directly process the data.

----End

8.3.2 Python

Prerequisites

- You have a Huawei Cloud account and have completed real-name authentication.
- Your Huawei Cloud account is not in arrears and has sufficient balance for the resources involved in this example.

Procedure

Step 1 Create a function.

- Log in to the FunctionGraph console, choose Functions > Function List in the navigation pane, and click Create Function.
- 2. Select **Create from scratch**, set the function information, and click **Create Function**.
 - Function Type: Select Event Function.
 - **Region**: Select **AP-Singapore**.
 - **Function Name**: Enter a function name, for example, **upload-file-1**.
 - Agency: Select Use no agency.
 - Runtime: Select Python 3.6.
- On the Code tab of the function details page, copy the following code to replace the default code, and click Deploy.
 # -*- coding: utf-8 -*-

from requests_toolbelt.multipart import decoder import base64

def handler(event, context): context.getLogger().info("Function start run.")

```
content_type = "
if "content-type" in event['headers']:
  content_type = event['headers']['content-type']
if "multipart/form-data" not in content_type:
  return {
     "statusCode": 200,
     "body": "The request is not in multipart/form-data format.",
     "headers": {
        "Content-Type": "application/json"
     }
  }
body = event['body']
# The APIG trigger encodes data using Base64 by default. The data is decoded here.
raw_data = base64.b64decode(body)
for part in decoder.MultipartDecoder(raw_data, content_type).parts:
  # Data is processed here.
  context.getLogger().info(part.content)
return {
  "statusCode": 200,
  "body": "ok",
   "headers": {
      "Content-Type": "application/json"
  }
}
```

Step 2 Create an APIG trigger.

- On the details page of function upload-file-1, choose Configuration > Triggers.
- 2. Click **Create Trigger**, and set **Trigger Type** to **API Gateway (APIG)** or **API Gateway (Dedicated Gateway)**. The shared gateway is used as an example.
 - API Name: Retain the default name.
 - API Group: If no API group is available, click Create API Group to create one.
 - Environment: Select RELEASE.
 - Security Authentication: In this example, select None for testing. You can select a more secure authentication mode, such as IAM, for your own services.
 - Protocol: Select HTTPS.
 - Timeout (ms): Retain the default value 5000.
- **Step 3** Perform E2E testing.

Create a file named **app.log** with any content on your local host. Example: start something run

stop all

 Take the curl tool as an example (curl -F is mainly used in the Linux environment). Run the following command: curl -iv {APIG trigger URL} -F upload=@/{Local file path}app.log

Figure 8-2 Example

• Take the Postman tool as an example. Set the following parameters and click **Send**.

Figure 8-3 Example



Name: Select upload.

Type: Select file.

Value: Click Upload to upload the created app.log file.

On the **Monitoring** tab of the **upload-file-1** function details page, view the file content in the logs. If needed, you can modify the code to save data to OBS or LTS or to directly process the data.

Figure 8-4 View logs

	_								
Metrics	Logs								
FunctionG	iraph records all requests processed by your function and automatically stores the logs in LTS.	Verify code by inserting custom logging	statements. The following table lists the function i	logs. View more on LTS. 🕑					×
		Las filason	INFORMATION INCOME	un .					
og oroop	San	Log Contain	DADADADADA						
how we all the	Descent land								
vertuear crai	Request Logs								
O Estreal	Interquest Logs			Now All Varriage and Allowar	Lastheer	Last day	Last 2 days	Custom	
Q Enteral	keyword.		8	Ihow All Versions and Aliases	Last hour	Last day	Last 3 days	Custom	Q
Q. Enter a l	keyword.		8	Inow All Versions and Allases	Last hour	Last day Q. Search II V	Last 3 days	Custom	().
C. Enter a l	kiyward.		8	Itow All Versions and Allases	Lasthour	Lastday Q Search ♥/	Last 3 days	Custom	() d Log
Q Enteral Logs 282	keyverd 2022-11-14T07:41:30Z Finish invoke re	equest '3b7298b6-	° 0a8a-4559-b306-4b20a	New AI Versions and Allases	Last hour : 2.601ms,	Q Search W/	Last3 days	Custom	d Log
C Enteral Logs 282 283	199001.000 199002. 2022-11-14T07:41:30Z Finish invoke re 2022-11-14T07:41:30Z Start invoke re	equest '3b7298b6- quest '04d1b2c0-0	° 0a8a-4559-b306-4b20a 0da5-4082-84e9-913aea	Row Al Versions and Alacco	Lasthour : 2.601ms, latest	Q Search V / billing du	Last3 days	Custom & Download 3ms, n	d Log
Q Enteral Logs 282 283 284	negeen.uge 197001 2022-11-14707.41:302 Finish invoke re 2022-11-14707.41:302 Start invoke reg 2022-11-14707.41:302 Gddib2c0-dds-4	equest '3b7298b6- guest '04d1b2c0-0 082-84e9-913aea58	° 0a8a-4559-b306-4b20a dda5-4082-84e9-913aea 33ae9 Function start	how Al Versions and Aliasos 701fedf', duration 5833e9', version: run.	Lasthour : 2.601ms, latest	Q Search V A	Last3days	Custom Dourrioan 3ms, m	d Log memor
Q Enteral Logs 282 283 284 285	1999 1999 2022-11-14107:41:302 Finish invoke re 2022-11-14107:41:302 Start invoke re 2022-11-14107:41:302 6415/20-045-40 2022-11-14107:41:302 6415/20-045-40	equest '3b7298b6- quest '04d1b2c0-0 852-84e9-913aea58 882-84e9-913aea58	0a8a-4559-b306-4b20a da5-4822-8449-913aea 33ae9 Function start 33ae9 b'this is app 1	<pre>https://www.arg.ang.ang.ang.ang.ang.ang.ang.ang.ang.an</pre>	Lasthour : 2.601ms, latest g\nrun\nsto	Q Search V / Dilling du	Last3 days	Custom & Doumloan 3ms, m	(Log nemor

----End

9 Processing IoT Data

Introduction Preparation Building a Program

9.1 Introduction

Scenarios

This section demonstrates how to combine FunctionGraph and IoT Device Access (IoTDA) to process status data reported by IoT devices. IoT devices are managed on the IoTDA platform. Data generated by the devices is transferred from IoTDA to trigger the FunctionGraph functions you have compiled for processing.

This combination is suitable for processing device data and storing them to OBS, structuring and cleansing data and storing them to a database, and sending event notifications for device status changes.

This best practice focuses on how to combine IoTDA and FunctionGraph. For details about how to manage devices and report data using IoTDA, see the documentation of IoTDA. In this chapter, we use IoTDA and FunctionGraph to convert WGS84 coordinates to GCJ02.



Procedure

- Create an IoTDA instance in IoTDA. (The standard edition is free of charge. You can use it for testing purposes.)
- Create a function in FunctionGraph.
- Set forwarding rules in IoTDA or create an IoTDA trigger in FunctionGraph.
- Send test messages using forwarding rules.

9.2 Preparation

Before creating a forwarding rule, create an IoTDA instance as well as products and devices. In this best practice, we only create an instance for testing.

Creating an IoTDA Instance

- Step 1 Log in to the IoTDA console. In the navigation pane, choose IoTDA Instances.
- **Step 2** On the right of the **IoTDA Instances** page, click **Buy Instance**. The parameter configuration page is displayed. Set the parameters based on service requirements.

Figure 9-1 Enabling free standard edition

< IoTDA Standa	ard Edition				
Region Instance Version Billing Mode	O 100-1026-026/0241 V Standard Entropolicy Yearly-blockby Pag-pall-code				
Specifications Instance Type Specifications	Standard Recommend Specifications				
	Instances must be used together with service units. Select Unit Type	the unit type and quantity (200 at most) as required. Max Concurrent Online Devices	Maximum Messages per Day	Max Message TPS ③	Required Quantity
	Free unit 90	1,000	10,000	10	- t +
	Micro-frequency unit ST	10,000	400,000	10	
	 Low-frequency unit S1 	10,000	4.001.000	200	
	 Intermediate-frequency unit S2 	10,000	40,000,000	1,000	
	High-frequency unit 53	10,000	150,000,000	3,000	
	Selected Specifications Tipos Free unit S0 Max Concu	ment Online Devices1,000 Maximum Messages per Der	10,000 Max Message TPS 10TPS		
num Teet etands	ard recourse already exists				(Canval) Bay

Step 3 Click Create.

----End

Creating a Function

- Step 1 In the left navigation pane of the management console, choose Compute > FunctionGraph. On the FunctionGraph console, click Create Function.
- Step 2 Select Create from scratch. Set Function Type to Event Function, enter a name (for example, iotdemo) for Function Name, select a runtime (for example, Python 3.9), and click Create Function.

----End

Creating a Forwarding Rule

Forwarding rules are used to transfer data from IoTDA to trigger specified functions. For this purpose, you can create forwarding rules in IoTDA or create an

IoTDA trigger in FunctionGraph. Perform the following procedure to create a forwarding rule:

Step 1 In the navigation pane on the left, choose IoT > IoT Device Access. On the IoTDA console, click the instance name. On the displayed page, choose Rules > Data Forwarding, and click Create Rule.

< 📝 freeStanda	rdInstance-test Standard V O Running
Overview	Data Forwarding
Products	
Devices ~	Rule List AMQP Queues
Rules 1	You can add and manage rules that trigger corresponding actions on connected devices when specifi
Data Forwarding 2	Create Rule Configure Policy Delete Enable
Server Certificates	Q Select a property or enter a keyword.
Device Linkage	Rule Name 🕀 Rule ID 🕀
O&M V	transfer-test dc7b9eba-8e04-4cce-ad9c-bc01fe9d9c57
Resource Spaces	

Figure 9-2 Creating a rule

Step 2 Enter basic information and click **Create Rule**.

NOTE

- Rule Name: Enter a custom rule name.
- Data Source: select Device message.
- Trigger: select device message reporting.
- **Resource Space**: Retain the default value.
- **Step 3** To set the forwarding target, click **Add**, and select **FunctionGraph**.
- Step 4 If this is the first time you select FunctionGraph, authorize access to IoTDA.
- **Step 5** Select function **iotdemo**.

Figure 9-3 Adding a forwarding target

Add Forwarding Tax	nat	Х
* Forwarding Target	Function Graph	
Region	InclicationGraph hostic event-driver functions in a serverises centeal while ensuring high availability, high scalability, and zero maintenance. All yr need to do is write your code and set the execution conditions.	.u
* Target FunctionGraph	initiations v Q A	
	Add Forwarding Target * Ferwarding Target Region * Tanget FunctionGraph	Add Forwarding Target * Foreading Target * Foreading Target * Foreading Target * Exceeding Target * Target *

Step 6 Start the rule.

----End

9.3 Building a Program

Editing a Function Program

Open function **iotdemo**, copy the following coordinate conversion code to the function. This code is for testing purposes only and can be modified if needed.

```
# -*- coding:utf-8 -*-
import json
import math
from math import pi
def handler(event, context):
  data = event["notify_data"]["body"]
  lat = data["lat"]
  lng = data["lng"]
  print(f" WGS84: ({lng},{lat})")
  gcj_lng, gcj_lat = transform(lng, lat)
  print(f" GCJ02: ({gcj_lng},{gcj_lat})")
  body = \{
      "gcj_lng": gcj_lng,
      "gcj_lat": gcj_lat
  }
  return {
     "statusCode": 200.
     "isBase64Encoded": False,
     "body": json.dumps(body),
     "headers": {
        "Content-Type": "application/json"
     }
  }
def transform(lon, lat):
  a = 6378245.0
  ee = 0.00669342162296594323
  dlat = transform_lat(lon - 105.0, lat - 35.0)
  dlon = transform_lon(lon - 105.0, lat - 35.0)
  rad_lat = lat / 180.0 * pi
  magic = math.sin(rad_lat)
  magic = 1 - ee * magic * magic
  sqrt_magic = math.sqrt(magic)
  dlat = (dlat * 180.0) / ((a * (1 - ee)) / (magic * sqrt_magic) * pi)
  dlon = (dlon * 180.0) / (a / sqrt_magic * math.cos(rad_lat) * pi)
  mg_lon = lon + dlon
  mg_lat = lat + dlat
  return mg_lon, mg_lat
def transform_lon(x, y):
  ret = 300.0 + x + 2.0 * y + 0.1 * x * x + \
     0.1 * x * y + 0.1 * math.sqrt(math.fabs(x))
  ret += (20.0 * math.sin(6.0 * pi * x) +
        20.0 * math.sin(2.0 * pi * x)) * 2.0 / 3.0
  ret += (20.0 * math.sin(pi * x) +
        40.0 * math.sin(pi / 3.0 * x)) * 2.0 / 3.0
  ret += (150.0 * math.sin(pi / 12.0 * x) +
        300.0 * math.sin(pi / 30.0 * x)) * 2.0 / 3.0
return ret
def transform_lat(x, y):
ret = -100.0 + 2.0 * x + 3.0 * y + 0.2 * y * y +
```

```
0.1 * x * y + 0.2 * math.sqrt(math.fabs(x))

ret += (20.0 * math.sin(6.0 * pi * x) +

20.0 * math.sin(2.0 * pi * x)) * 2.0 / 3.0

ret += (20.0 * math.sin(pi * y) +

40.0 * math.sin(pi / 3.0 * y)) * 2.0 / 3.0

ret += (160.0 * math.sin(pi / 12.0 * y) +

320 * math.sin(pi / 30.0 * y)) * 2.0 / 3.0

return ret
```

Online Joint Commissioning with IoTDA

- Step 1 Log in to the IoTDA console and click an instance name. In the navigation pane, choose Rules & > Data Forwarding. In the Rule List, click View on the right of the target rule name. The Data Forwarding Rule Details page is displayed.
- **Step 2** Select **Set Forwarding Target** and click **Test** on the right of the forwarding target to edit the test data.

Figure 9-4 Testing the forwarding rule



Step 3 Enter the test data and click Connectivity Test.

```
"resource": "device.message",
  "event": "report",
  "event_time": "string",
  "notify_data": {
     "header": {
        "app_id": "d4922d8a-6c8e-4396-852c-164aefa6638f",
        "device_id": "d4922d8a-6c8e-4396-852c-164aefa6638f",
"node_id": "ABC123456789",
        "product_id": "ABC123456789",
        "gateway_id": "d4922d8a-6c8e-4396-852c-164aefa6638f",
        "tags": [
           {
              "tag_key": "testTagName",
              "tag_value": "testTagValue"
           }
        ]
    },
"body": {
        "lat": 92.64763932844794,
        "lng": 35.25202546134364
     }
  }
}
```



< Data Forwarding Rule De	Compatibility Test	×
🗊 Stef Farwarding Data ——— 🕘 Bet Finanading Targat ———— 3 Enable Rule	You can enter the data you want to test in the input box below, as be forwarded to the set larget	rd test whether the data can
Specify the largeds (Marker Claud server or phone server) to finance the olde to. (Mar) Up to 10 threading largest can be added.	Test Data 'product_jd': 'ABC123466789', 'galwaway_dt': '4822208a-8c8a-4396-852c-164aata683 'fag5'; [Analog Input Template 81°,
Function Group In O SENTERSTERSTERSTER	Tap_key="ballbackground- } Tap_key="ballbackground- } background- background-background-background- Tap="background-background-background- Tap="background-backg	
	Test Result (Buccess) (Aug 08, 2024 16:13:13 GMT+08:00) SUCCESS	Clear

Step 4 Go to the FunctionGraph console, choose **Monitoring** > **Logs**, and click the request ID in blue to view logs.

Figure 9-6 Viewing logs

Code Monito	aring Version Alia	ses Configuration								
Metrics	Logs									
FunctionGraph	h records all requests processed I	ay your function and automatically stores the logs in LTS. Verify co	de by inserting custom logging	statements. The following table lists	the function logs. View more on LTS.	c				×
Log Group	function	5-20 Sec. 0	Log Stream	- IPARAMANANANANAN	NATION OF C					
Request List	Request Logs									
O, Enter a key	word.				Show All Versions and Alla	IS85 🕜 👘	Last hour Last day	Last 3 days	Custom	0
Time		Request ID	Duration		Memory Used		Version	Cause		
Aug 08, 202	4 16:15:31 GMT+06:00	c464.6728.728.728.728.728.728.728.3		37.098ms	_	39.695MB	latest	Cold start successful		

Figure 9-7 Viewing request details

13cb1df1-6eeb-4343-b471-4abaf5138240

Result • Cold sta	art successful	Duration 75.474m	s Memory Used	33.902MB		
Logs			QS	earch 🛛 😨 All	E Full Screen	📥 Download Log
1 2023-€ 2 2023-€ 3 ∨ 2023-€ 4 WGS84 5 GCJ02 6 2023-€	4-25T14:32:57 4-25T14:32:57 :(35.2520254 :(35.2520254 :(35.1853171 4-25T14:32:57	Z Start load reque Z Finish load requ Z Start invoke rec 6134364,92.6476393 39674076,92.64928 Z Finish invoke re	est '4f8e0710-2d1 lest '4f8e0710-2d juest '12cbldfl-6 l2844794) l65064839) equest '13cbldfl-	7-4275-9a4 17-4275-9a eeb-4343-b 6eeb-4343-	4-5150431f156 44-5150431f15 471-4abaf5138 b471-4abaf513	7', RDB6000000000000000000000000000000000000

To invoke other systems, persist data in OBS, or achieve other purposes, modify the program.

----End

10 Workflow + Function: Automatically Processing Data in OBS

Introduction Preparation Building a Program Processing Images

10.1 Introduction

This best practice guides you through OBS data processing by using FunctionGraph. (The function flow feature is available in CN East-Shanghai1 and AP-Singapore.)

Scenarios

Use a function flow to automatically process data in OBS, such as video analysis, image transcoding, and video frame capturing.

- Upload images to a specified OBS bucket.
- Orchestrate functions to download images from OBS for transcoding and return the transcoded images to the client using a stream.

D NOTE

The function you create must be in the same region (default region recommended) as the OBS bucket.

Procedure

- Create a bucket on the OBS console.
- Upload images to the bucket.
- Create a function.
- Create a function flow and orchestrate functions.

• Trigger the function to transcode images.

NOTE

After you complete the operations in this tutorial, your account will have the following resources:

- 1. One OBS bucket (for storing uploaded images)
- 2. One image processing function (test-rotate)
- 3. One function flow (test-rotate-workflow)

10.2 Preparation

Create an OBS bucket to store uploaded images.

Then create an agency to delegate FunctionGraph to access OBS resources.

Creating an OBS bucket

The bucket and function must be in the same region.

Procedure

Step 1 In the left navigation pane of the management console, choose Storage > Object Storage Service to go to the OBS console, and click Create Bucket.

On the **Create Bucket** page, set the bucket information.

- For **Region**, select a region.
- For **Bucket Name**: Enter a custom bucket name, for example, **your-bucket**-**input**.
- For Data Redundancy Policy, select Single-AZ storage.
- For **Default Storage Class**, select **Standard**.
- For **Bucket Policies**, select **Private**.
- For Default Encryption, select Disable.
- For **Direct Reading**, select **Disable**.

Retain the default values for other parameters and click **Create Now**.

View your-bucket-input in the bucket list.

----End

Creating an Agency

Step 1 In the left navigation pane of the management console, choose Management & Governance > Identity and Access Management to go to the IAM console. Then choose Agencies in the navigation pane.

On the Agencies page, click Create Agency.

Set the agency information.

- For Agency Name: Enter an agency name, for example, serverless_trust.
- For Agency Type, select Cloud service.
- For Cloud Service, select FunctionGraph.
- For Validity Period, select Unlimited.
- For Description, enter a description.
 Click Next. On the Select Policy/Role page, select OBS Administrator.
- **Step 2** Click **Next**, select an authorization scope that meets your service requirements, and click **OK**.

----End

10.3 Building a Program

This section provides the sample code for image rotation.

Creating a Deployment Package

This example uses a Go function to rotate images. For details about function development, see the *FunctionGraph Developer Guide*. Figure 10-1 shows the sample code directory. The service code is not described.

Figure 10-1 Sample code directory



Creating a Function

When creating a function, specify an agency with OBS access permissions so that FunctionGraph can invoke the OBS service.

Step 1 Log in to the **FunctionGraph console**, and choose **Functions > Function List** in the navigation pane.

Click Create Function.

Set the function information.

After setting the basic information, click **Create**.

- For Function Type, select Event Function.
- For **Function Name**: Enter a function name, for example, **test-rotate**.
- For Agency, select serverless_trust.

• For **Runtime**, select **Go 1.x**.

On the details page of function **test-rotate**, configure the following information:

- a. On the **Code** tab, choose **Upload** > **Local ZIP**, upload the binary file **go**-**test.zip** of the sample code.
- b. Choose **Configuration** > **Basic Settings**, set the following parameters, and click **Save**.
 - For Memory, select 256.
 - For **Execution Timeout**, enter **40**.
 - For **Handler**, retain the default value **index.handler**.
 - For **App**, retain the default value **default**.
 - For **Description**, enter **Image rotation**.
- c. Choose **Configuration** > **Environment Variables**, set environment variables, and click **Save**.

bucket: the bucket parameter defined in **handler.go** for pulling images. Set the value to **your-bucket-output**, the bucket created for storing images.

object: the image name parameter defined in **handler.go**. Set the value to **your-picture-name**.

obsAddress: the bucket address parameter defined in **handler.go** for pulling images. Set the value to **obs.region.myhuaweicloud.com**.

----End

Environment Variable	Description
bucket	OBS bucket parameters defined in the handler.go file for pulling images.
object	The image name parameter defined in handler.go .
obsAddress	The bucket address parameter defined in handler.go for pulling images. The value of obsAddress is in the format of obs.{region}.myhuaweicloud.com . For details about the value of region , see Regions and Endpoints .

Table 10-1 Environment variable description

---- End

Creating a Flow

Step 1 Return to the FunctionGraph console. Then choose Flows in the navigation pane.Click Create next to Express Flow.

Create Flow											Û	
All 🔻	Q		(+	+	+	+					
			Start				E	incti	on			
∧ Components			otart						011			
😥 EG												
Runction												
∧ Processors		•										
Conditional Branch												
Error Handling												
Loop												
ϕ_{00}^{00} Parallel Branch												
Service												
Start												

Figure 10-2 Creating an express flow

Step 2 Drag a function node and click it to configure parameters.

- **App**: Retain the default value **default**.
- **Function**: Select the **test-rotate** function created in the previous step.
- Version: Retain the default value latest.
- Retain the default values for other parameters.

Figure 10-3	Configuring	function node
-------------	-------------	---------------

zyljava8 🖉		
* App	default	~ C
★ Function	test-rotate	~) C
* Version	latest	 ✓ C View Function
Function Parameters	Key Value	DefaultValue Operation
	⊕ Add	
Input Filter Expression (?)		
Output Filter Expression ③		
Turn on StandbyOper		
	When enabled, the current node name function node names	e cannot be the same as other

Click **OK** after the parameters are configured.

- **Step 3** After the function flow node is created, click **Save** in the upper right corner, configure the following basic information, and click **OK**.
 - Name: test-rotate-workflow.
 - Enterprise Project: Retain the default value default.
 - Logs: Retain the default value ALL.

Retain the default values for other parameters.

Figure 10-4 Saving a flow

* * * * *	ta t	Format 🖬 Process	Cons	tants	ŝ	Ċ	Sav	/e (
Create Flow		×					0		
+ Name	test rotate-workflow								
A Humo									
	Enter 1 to 64 characters, starting with a letter and ending with a letter or digits, hyphens (-), and underscores (_) are allowed.	digit. Only letters,							
ŧ.	default view Enter	rprise Project							
k Logs	ALL ~								
Streaming Response									
Description	Please enter content								
	0/200 🏑								
		_							
	2 ок	Cancel							

10.4 Processing Images

Upload an image to bucket **your-bucket-input**, and use a tool to simulate a client and trigger the function flow. The image is rotated by 180°, and then returned to the client as stream data.

Uploading an Image

Log in to the **OBS console**, go to the object page of the **your-bucket-input** bucket, and upload the **image.jpeg** image, as shown in **Figure 10-5**. **Figure 10-6** shows the uploaded image.

Figure 10-5 Example



Figure 10-6 Uploaded image

Name	Storage Class	Size 🕈	Last Modified	Operation
image (peg	Standard	1.86 MB	Apr 23, 2024 17:32:53 GMT+08:00	Download Share More ~

Using Postman to Trigger the Function Flow

POS	 A state of the second state of th	Page Sand v	l
Poran	is Authorization Headers (12) Body Pro-request Script Tests Settings	Cookies	•
Head	lers. 🐵 9 hidden	Vite - Bat (d. Press	
	Key	Volue Buik Edit Presets	
	X-Auth-Token	(laser_toker()	
	Content-Type	application/joon	
2	X-Stream-Enable	the	
	Key	Value	
Body	Cookies (2) Headers (11) Test Results	Contract 200 DK Times 1 m 3155 a Size: 555 MD Save Response v	
Pre	tty Raw Preview Visualize 180N v 📅	9	
2	0.01 0.01 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000		
1			
- 2	States () Mistra (SHI) 13 WWW (SHI) 11 WWW (SHI)		
6	20022020202020-0002020 1 20-405-00 * 20020-00 * 24000-0		
	B40billing ()+5794-001691253049721445grijstungs 200		
- 18	10101/-000100-00100100000		
- 11			
12	\$1000 (_:000007:0100 000001/000007/////00007/00000000000		
13	+++ +01/404/404/404/404/4000000t		
- 14	0-0+0000+100010000000100000000000000000		
- 13		iese hereen here	
12	THE LAST DE DEDUCT, DEBUG DE DESERVOIR DE STATE DE LE DE		
18	0000/00_00 **0007_0007.00/0_000		
19	- 012+00000000000000000000000000000000000	00000000000000000000000000000000000000	۶
	If 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0	H2/1-010-2_0-0-0-0-0-00000-0-0-0-0-0-0-0-0-0-0-	
20	We'enteesesteryeesesterteetetteesterteesesterteeteesesterteeteeteesterteetteet	8/02/18/001(00;000:000:000:000:0000:000000000000	
21			
- 22			
24	00000001000-000.001010000 0m000000000000000000000000	KKOTOT*(COOT)000001/0021-00000/TE*0TXTET/WTEVE 1102K00*00/COOC0000	
- 25	- 0000110(000-01000010000-10000)/000_00 11 01100100-10100 010001010101000100-110:0001-0100000100-010-01	0001010/000000000000000000000000000000	
26	\$		
29			
30	61161611166666601)0-66811666226161-66566100-622161-6006		
31) (IDEADED (DEADED CONTRACTORISED AND CONTRACTORISE AND CONTRACTORISE AND CONTRACTORISE AND CONTRACTORISE AND CONTRACTORISE AND CONTRACTORISTICA AND CONTRACTORISTICA AND CONTRACTORISTICA AND CONTRACTORIS		
22	<00101010101010101010101010101010101010		
- 23			
	12.00 C (100 C (
35			
- 32)12200000000-10000210)-1000000(c000000000000000000000000000000		
35)_ 0:0000 0		
- 29	reserves exerces instead and control and the state is a second state in the second state is a seco	neessesse nes sessingesenennentenensesnes reamsig(s)s(s)s(s, 00)s(r),s(m0000)s(s)(s)(s)(s)(s)(s)(s)(s)(s)(s)(s)(s)(s	ŕ
41			
42	(v48 2008 8 · · · · · · · · · · · · · · · · ·		
43	101001001001001001001001001001000001110000		
	1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1		
	1/2 s-manufiltum atom		
- 6		ken 🔶 men, e	
45	(); (kes∳ mijit	

The following figure shows the image saved from the byte stream.

08 Best Practices (Official)



11 Filtering Logs in Real Time by Using FunctionGraph and LTS

Introduction Preparation Building a Program Adding an Event Source Processing Results Extended Applications

11.1 Introduction

This chapter elaborates the following aspects of the practice:

- Scenario and benefits
- Preparation
- Building a program
- Event source
- Processing results
- Extended applications

Scenario

Quickly collect, process, and convert task logs of servers, such as ECSs, through Log Tank Service (LTS).

Obtain log data using an LTS trigger created on FunctionGraph, analyze and process key information in the logs by using a customized function, and then transfer the filtered logs to another log stream. **Figure 11-1** shows this process.

Figure 11-1 Processing workflow



Benefits

- Quickly collect and convert logs with LTS.
- Process and analyze data by using the event triggering and auto scaling features of serverless function computing. No O&M is involved, and resources are pay-per-use.
- Transfer filtered logs to another log stream. The original log stream is automatically deleted at the expiration time you set, reducing log storage costs.

11.2 Preparation

Download **lts_cleanse.zip** (including code file **write_log.py** of function A, code file **lts_cleanse.py** of function B, and dependency **huaweicloudsdklts**) and **lts_cleanse.zip.sha256** to filter logs in real time.

Collecting and Storing Logs

- Create two log groups, for example, **test1206** and **test-1121**, on the LTS console. For details, see **Creating a Log Group**.
- Create two log streams, for example, test-206 and test-1121, on the LTS console. For details, see Creating a Log Stream.
- Create function **A** to write logs to **test-206**. For the sample code of this function, see the **write_log.py** file.
- Create function **B** with an LTS trigger to receive logs from **test-206**, process the logs, and write the result to **test-1121**. For the sample code of this function, see the **lts_cleanse.py** file.
- Configure an agent to collect logs from servers, such as ECSs, to a specified log group. For details, see **Installing the ICAgent**.

Figure 11-2 Flowchart



Creating an Agency

- **Step 1** Log in to the IAM console.
- **Step 2** Choose **Agencies** from the navigation pane, and click **Create Agency** in the upper right corner, as shown in **Figure 11-3**.

Figure 11-3 Creating an agency

IAM	Agencies ①					Create Agency
Users						
User Groups	Delete Agencies available for creation	437				
Permissions ~	AL V Q.B	ter an agency name.				
Projects	Agency Name10 B	Delegated Party B	Validity Period 🛞	Created 🖨	Description G	Operation
Agencies		Claudiservice	Unimbed			
Martin President	dme_edmin_trust	Industrial Digital Thread (IDT)		Apr 24, 2024 09:27:51 GMT+08:00	-	Authorize Modify Delete

- **Step 3** Configure the agency.
 - Agency Name: Enter an agency name, for example, LtsOperation.
 - Agency Type: Select Cloud service.
 - Cloud Service: Select FunctionGraph.
 - Validity Period: Select Unlimited.
 - **Description**: Describe the agency.
- **Step 4** Click **Next**. On the displayed page, search for **LTS Administrator** in the search box on the right and select it.

NOTE

LTS Administrator depends on Tenant Guest. When you select the former, the latter will also be selected.

Step 5 Click **Next**, select an authorization scope that meets your service requirements, and click **OK**.

----End

11.3 Building a Program

Prerequisites

(1) The IP address in the two functions is an access point of LTS. To obtain this IP address, perform the following steps:

- Log in to the LTS console. In the navigation pane on the left, choose Host Management > Hosts.
- 2. In the upper right corner of the page, click **Install ICAgent**.
- 3. Obtain the access point IP address in the Install ICAgent window.

Figure 11-4 Access point IP address

Install ICAge	nt 🕐							
Host	Intra-Region Ho	osts	Extra-Region Ho	ests beta				
os	Linux	Windows	?					
installation Mode	Obtain AK/SK	Creat	te an agency					
	To install ICAgent on multiple intra-region hosts with one click, see Inherited Batch Installation. (Skip this step for ServiceStane or CCE) To install it on a single intra-region host:							эг
			in a bingio ina a r	Sgion nost.				
 Step 1: Enter 	the AK/SK to generate	the installation	command. How	Do I Obtain ar	n AK/SK Pair?			
3 Sten 2: Conv	the ICAgent installation	n command						
		r command.						
Turn of	command history to	o prevent the	AK/SK from b	eing stored.				
Copy Comma	and 🥏							
10 er	iter the AK/SK, either:							×
1. Copy th	e command and replac	e {input_your_a	ak} and {input_y	our_sk} without	the braces {},	or		
2. Run the	e copied command and	enter the AK ar	nd SK when "En	er the AK" and	"Enter the SK	are displayed.		
								_
set +o his	tory;curl https://icag	ent 💦 🖓		<u>.</u>				
4.myhuav	veicloud.com/ICAger	nt_linux/apm	_agent_install	sh > apm_a	gent_install.s	h && REGIO	N影響運動運動	ž.
bash apm	_agent_install.sh -a	k {input_your	r_ak} -sk {inpu	t_your_sk} -r	egic	ection (Construction)	t	
Of1d	2477-267-267-2a4	a1d35ea -aco	cessip 10000	教授者(初50 -ob)	sdomain obs	Construites		
4.myhuav	veicloud.com -acces	sdomain Its-a	access	a si	eicloud.com	set -o history	Γ;	
3 Step 3: Log in ICAgent is su	n to the host as user roo iccessfully installed. You	ot and run the c u can then view	copied command v the ICAgent sta	. When the me tus on the Hos	ssage "ICAgen t Management	t install success page.	s" is displayed	Ļ
A If the inst								

2. Obtain the values of **log_group_id** and **log_stream_id** in the functions. For details, see **Obtaining the Account ID, Project ID, Log Group ID, and Log Stream ID**.

3. Create the LTS dependency required by function **B**. For details, see **How Do I Create a Dependency on the FunctionGraph Console?** and **How Do I Add a Dependency to a Function?** You can run the **pip install huaweicloudsdklts** command to create the dependency. The sample code contains the **huaweicloudsdklts** dependency for Python 3.9.
Creating a Function

Create a log extraction function by uploading the sample code package. Select the Python 3.9 runtime and the agency **LtsOperation** created in **Creating an Agency**. For details about how to create a function, see **Creating an Event Function**.

Create function **A**. For the sample code of this function, see the **write_log.py** file. In the code of function **A**, replace **host**, **log_group_id**, and **log_stream_id** with the access point and the IDs of log group **test-1206** and log stream **test-206**, as shown in **Figure 11-5**.





Create function **B**. For the sample code of this function, see the **lts_cleanse.py** file. In the code of function **B**, replace **host**, **log_group_id**, and **log_stream_id** with the access point and the IDs of log group **test-1121** and log stream **test-1121**, and add the **huaweicloudsdklts** dependency to this function, as shown in **Figure 11-6** and **Figure 11-7**.

Figure 11-6 lts_cleanse.py



Figure 11-7 Adding a dependency for function B

Add Delete							
Q. Select a property or enter a keyword.						Q (
□ Name ⊖	Type Θ	Version Θ	Runtime O	Address 0	Description Θ	Operation	
huaweicloudsdkts	Private	4	Python3.9	https://unctions/2000000000000000000000000000000000000	-	Delete	

This function performs Base64 decoding on received log event data, extracts alarm logs containing keyword **WRN**, **WARN**, **ERR**, or **ERROR**, and then stores the extracted logs to a specified LTS log stream. Set log extraction conditions based on the content of your service logs.

11.4 Adding an Event Source

Create an LTS trigger by using the log group and log stream created in **Preparation**, and configure the trigger information according to **Figure 11-8**.

Figure 11-8 Creating an LTS trigger

Create Trigger

Trigger Type	Log Tank Service (LTS)	~		
★ Log Group	test-1206	~	Q	Create Log Group 🛽
★ Log Stream	test-206	~	Q	Create Log Stream [

When the accumulated log size or log retention period meets a specified threshold, LTS log data will be consumed, which will trigger the function associated with the log group.

11.5 Processing Results

Filter alarm logs containing keyword **WRN**, **WARN**, **ERR**, or **ERROR**, and transfer them to a specified log stream. Figure 11-9 and Figure 11-10 show the real-time logs before and after filtering, respectively.



Figure 11-9 Logs before filtering

Figure 11-10 Logs after filtering



Check the function invocation by viewing the metrics, as shown in the following figures.

Figure 11-11 Function metrics (1)

Invocations Count 45 times	11 C i	Duration Ouration 0 ms	C C I
Unit times 18			
15		~	
9		No Data.	
0 10.50 1205 1320 1435 1550 1705 1820 1835 2050 2205 2320 0035 0150	03.05 04.20 05:35 06:50 08:05 09:20 10:50		

Figure 11-12 Function metrics (2)

Errors / Error rate	0 C i	Throttles	0 C I
Count 4 times		Count 0 times	
🗢 System error count 🔶 Function error count 🔶 Error rate			
Unit: times	Unit: %	Unit: times	
2	100	1	
	80		
1	60		
	40		
	20		
0	0 05 06:50 08:05 09:20 10:50	0 1050 12:05 13:20 14:35 15:50 17:05 18:20 19:35 20:50 22:05 23:20 00:35 01:50 /	03:05 04:20 05:35 06:50 08:05 09:20 10:50

Figure 11-13 Function metrics (3)



11.6 Extended Applications

In addition to log processing and transfer to LTS, the combination of FunctionGraph and LTS can apply to more scenarios, for example, analyzing and

processing log data with a timer trigger to delete redundant logs and save space and costs.

12 Deploying AI Drawing Stable-Diffusion in the Application Center

Introduction Preparation Application Creation and Deployment Application Use Custom Models Advanced Use Disclaimer

12.1 Introduction

You can deploy AI drawing Stable-Diffusion in the application center of FunctionGraph and upload custom models. Also, you can quickly access the Stable-Diffusion WebUI to perform text-to-image and image-to-image transformation (Only available in **CN East-Shanghai1**).

12.2 Preparation

12.2.1 Overview

Custom Models

To use custom models, the Virtual Private Cloud (VPC) and Scalable File Service (SFS) provided by Huawei Cloud are required. You can perform the following steps:

- 1. Creating a VPC and Subnet
- 2. Creating an SFS Turbo File System

3. Creating an Agency

Custom Domain Names

To use a custom domain name, Huawei Cloud Domains service is required. You can perform the following steps:

- 1. Create an information template and perform identity authentication.
- 2. Wait for the real-name authentication result.
- 3. Query and purchase a domain name.
- 4. File the domain name. (Select FunctionGraph as the filing service.)
- 5. Configuring Domain Name Resolution.

12.2.2 Creating a VPC and Subnet

- **Step 1** Log in to **Huawei Cloud Network Console** and click **Create VPC**. The **Create VPC** page is displayed.
- **Step 2** On the **Create VPC** page, configure the parameters as follows:
 - Basic Information
 - Region: Select CN East-Shanghai1. Currently, the Stable-Diffusion application can be deployed only in this region.
 - Name: Enter a custom name.
 - **IPv4 CIDR Block**: Set this parameter based on site requirements.
 - Enterprise Project: Use default.
 - Default Subnet
 - **AZ**: Retain the default value. Ensure that the AZs of the file systems created later are the same.
 - Name: Enter a custom name.
 - **IPv4 CIDR Block**: Set this parameter based on the site requirements.
 - Retain the default values for other parameters.
- Step 3 Click Create Now.

----End

12.2.3 Creating an SFS Turbo File System

- Step 1 Log in to Huawei Cloud SFS Console, select SFS Turbo, and click Create File System. The Create File System page is displayed.
- **Step 2** Set the following parameters:
 - **Billing Mode**: Select a billing mode based on the site requirements. **Pay-per-use** is recommended.

NOTE

For details about SFS billing, see **Billing Overview**. For details about the pricing, see **Price Calculator**.

• **Region**: Select **CN East-Shanghai1**. Currently, the Stable-Diffusion application can be deployed only in this region.

- **Project**: Use default.
- **AZ**: The option must be the same as the subnet AZ.
- File System Type: Select one based on the site requirements.
- **Storage Class**: Select one based on the site requirements.
- Capacity (TB): Select one based on the site requirements.
- VPC: Select the VPC and subnet created in Creating a VPC and Subnet.
- **Security Group**: Set this parameter as prompted.
- Enterprise Project: Use default.
- Retain the default values for other parameters.
- **Step 3** After the configuration, click **Create Now** and wait until the SFS Turbo file system is created.

----End

12.2.4 Creating an Agency

To deploy the Stable-Diffusion WebUI through Huawei Cloud serverless application center, you need to use FunctionGraph with other cloud services and authorize operation permissions on cloud service resources to FunctionGraph by configuring an agency. It is strongly recommended that you create the agency in advance as the agency takes effect 15 to 30 minutes later.

- **Step 1** Log in to the **IAM console**. In the navigation pane, choose **Agencies**. On the **Agencies** page that is displayed, click **Create Agency** in the upper right corner.
- **Step 2** Set the following parameters:
 - Agency Name: Enter a custom agency name.
 - Agency Type: Select Cloud service.
 - Cloud Service: Select FunctionGraph.
 - Validity Period: Select Unlimited.
 - **Description (Optional)**: Enter a description.

★ Agency Name	fg-sd
★ 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	FunctionGraph ~
* Validity Period	Unlimited V
Description	Enter a brief description.
	0/255
	OK Cancel

Figure 12-1 Creating an agency

- **Step 3** After setting the parameters, click **Next**. Search for and select the following five system-defined policies or roles from the list, and click **Next**.
 - OBS Administrator
 - SFS FullAccess
 - SFS Turbo FullAccess
 - VPC Administrator (If you select this policy, SWR Admin is automatically selected.)
 - SWR Admin

Figure 12-2 Selecting a policy

0	Precyllate 2 Interd Scope 3 I Train					
Assign se	voted permissions to fp-ed.					Create Policy
0.8	etile gemissions: 8, which include the dependency gemissions of VPC Administrator. Clori View Selected or expand the details area to view the dependency permissions.					
Ver	Needed (8) Copy Premissions how Acobier Project		All policies/toles	v Alsenios	 Разун	Enter a policy name, role nQ
	Palicy Rate Name	- Tope				
2~	089 Aministrator Object Storage Senico Administrator	System-defined policy				
•	275 PuNAcoss Al permissions of 975 service.	System-defined policy				
•	5PS Turbi FAll-Access All premissions of Scatable File Service (SPS Tates).	System-defined policy				
•	VPC Administrator VPC Administrator	Bystem-defined role				
•	Serve Aministrary Serve Aministrary	System-defined role				
• ~	Style Admin Debuter Topostory Admin	System-defined sole				

Step 4 Select a scope for the permissions as required. If you are not sure about the scope, select **All resources** or **CN East-Shanghai1** in **Region-specific projects**.

Figure 12-3 Region-specific projects

Suecificaçãos 20 Suecificaçãos 3 Fran
The following are recommended scopes for the permissions you selected. Select the desired scope nequiring minimum authorization. Learn toor to select a proper scope.
Stope
A resurces
O Entrypts regists Whi sees with a data for an expansion in the solubid entropies project later of an assigned spenticions. For sample, are entropies project and are the adopted in different hugines. Many sea associate the entropies project with the Will sees, they can associate the entropies project with the Will sees.
 Reproduced, project Will cases will be able to an encourse: In the solected region specific project based on ansigned permissions.
Some permissions (VOSS Administrator) will be applied to all resources by default.
Total projects: 10 Select the decined projects.
Project[Region] Description
C (*****) [OK Exit Strangton]

NOTE

OBS Administrator does not support this authorization scope. By default, the permissions apply to all resources.

Step 5 Click OK. The agency is created. Wait until the agency takes effect.

----End

12.2.5 Configuring Domain Name Resolution

Step 1 After the steps in **Application Creation and Deployment** are complete, click **Bind now**. Click the **Summary** tab and copy the subdomain name for backup.

Figure 12-4 Copying the subdomain name

,	API Groups / sd_a	bigw_groupTheThe		
	Summary	APIs Variables Domain Names Gateway Responses		
ľ				
	Name	sd_apigw_group_	ID	c8 35
	Created	Feb 27, 2025 20:04:52 GMT+08:00	Status	Not listed
	Subdomain Nam	estadoren a merinaria metanaria urtuarreat. 🗇	Description	sd_apigw_group
		The subdomain can be used only for development and testing and can be accessed 1,000 times a day. Bind independent domain names to the API group so that the group's APIs are accessible to user	š	

Step 2 Log in to the DNS console, choose **Public Zones**, and click **Manage Record Set** on the right of the purchased domain name.

Figure 12-5 Clicking Manage Record Sets

Domain Name Service	Public Zones ()								Create Public Zone
Cveniew Public Zones	Public cones you can still cester. 49 Delete Eatch Operations v Export)							
Private Zimes	Q. Search or filter by domain name.								0.0
PTR Records	Domain Name 🔒 Status	Record Set	s Enterprise Project	Tag Er	nal T	FL(6) Created (6)	Last Modified 🖗	Description	Operation
Eastor P (*	C Same Otoma	;	2 defaut	-	a hada qili qara ka	300 Od 25, 2023 1942 30 G	Oct 25, 2023 19:42:38 G	-	Manage Record Sets Check Comain Name Disable More +
	Total Records: 1								11 v (1)



Figure 12-6 Adding a record set

Record Sets	Export and Import	Tags	DNSSEC	Traces
1 We Hua Pub Cha	have added new DNS servers t wei Cloud provides the Check I lic zones take effect only after y inge the effective time of the do	o improve gl Domain Nam 'ou update th main name [lobal DNS resolution the function to quick the name servers of DNS server accord	n. View details y verify whether the domain name resolution takes effect. your domains with the registrar to ns1.huaweicloud-dns.org, ns1.hua ng to the description provided by the domain name service provider.
Record sets	s you can still add: 488			
Add Re	ecord Set Delete	Enable		

Step 4 In the displayed **Add Record Set** dialog box, configure the information.

- 1. **Name**: The value can be customized.
- 2. Type: Select CNAME Map one domain to another.
- 3. Value: Enter the subdomain name copied in step 1.

Retain the default values of other parameters.

CNAME – Map one domain to another	×
Name	
Example: www	com
ine 🕥	
Default	~
TL (s) 🕥	
300	
/alue ⑦	
and the second sec	
	4
 Advanced Settings (Optional) 	
Alias: No Weight: 1 Tag: Description:	

Figure 12-7	Configuring	the record set



----End

12.3 Application Creation and Deployment

Step 1 Log in to the FunctionGraph console and select CN East-Shanghai1 region. In the navigation pane on the left, choose Application Center. In the upper right corner, click Create. The page for selecting a template is displayed.

Figure 12-8 Creating an application



Step 2 Click Create Application. The application introduction page is displayed. Read it carefully. If this option is unavailable, check whether your current region is CN East-Shanghai1.

Figure 12-9 Selecting an application template

Create With			
Template AL, video processing, and many other templates			
Template			
Enter a template name.			0
Runtime All Node Pyth	on HTTP		
Category All Web Al			
() 17 ()	O 15 Å	27 👌	④ 4,073 ♦
AI Conversations ChatGLM3	Al Drawing ComfyUI	AI Drawing ComfyUI+FLUX	Al Drawing Stable-Diffusion
AI Conversations	Al Drawing with ComfyUI	AI Drawing with ComfyUI+FLUX	Al Drawing
AI HTTP	А НТТР	А НТР	AI HTTP

Step 3 Click **Learn More** to read the application usage document carefully and click **Create Application** in the lower right corner. The configuration page is displayed.

Applications / Greate Application			Al Drawing Stable-Diffusion
Create With			State Diffusion Ar Drawing
			Application Description
Al, video processing, and many other templates			Deploy Stable-Olflusion-WebUI on HUWWEI CLOUD Function@raph to support custom models.
			Pre-preparation
Tempiate			Note: This app can be used only in Shanghai 1.
Enter a template norm.	6-m WTTD		To use this app, you need to subscribe to the following services. Fees may be incurred during the use of this app:
Category Al Web	4		Service Description
			FunctionGraph Provides inference computing power for Stable Diffusion
Ø	S *	O 27 A	5F5 Used to store model files, plugins, generated results, etc.
Al Conversations ChatGLM3 Al Conversations	Al brawing com/yuli Al brawing with Com/yUl	Al Drawing ComtyUH+FLUX	You also need to be ansare of
м нтр	N HTTP	8 HTP	 In the diplocation big/diplocation involves and concess with pre-transmort singer increases. C-point models needs to be diplocation to the relevant disk using the elevation inadapprinter that the application is subconsistivity displayed in the selections in the usging efficiency and staff after the models hadded the model is displayed in the selections in the usging efficiency and staff after there receive. The whith screen takes a leng time during the initial istance of the apps. This is the cold ward staffs of the areases visit adverts.
Carter Generation	CO 45 A	django 44 🔶	Application deployment
Animize the uphoaded availar file.	Create a function application based on DIS service triggers to quicity process real-time data. After the application is deployed, processing functions are automatically created, DIS services are	Django is an open source web application framework written in Pythoe, adopted MTV's framing model, I. e., model M, view V, and template T.	Click Application Configuration at the bottom of this page to quickly deploy the application. The deployment takes about one minute to create related resources. Wall will the creation is complete.
N Python 3.8	Enabled, and Unit inggers are cleaned.	Weg HTTP	Choose service: Logical same Punction
			Punction service stable_605ese Al dearing function; which can access the Stable 205ees WebUI Result its APIG larger
			Cancel Create Applicator

Figure 12-10 Template description

Step 4 On the **Configure Application** page, set the **Application Name**, **Agency**, and **Description** (optional). Select the agency created in **Creating an Agency** and click **Create Now** in the lower right corner.

Figure 12-11 Configuring an application

Applications / Creat	te Application
< Create App	lication
Create With	
AI, vide	are or processing, and many other templates
Application Con	figuration
S Al Draw	ring Stable-Diffusion mg
Basic Info	
Region	• of Landau ·
	Regions are geographic areas isolated from each other. Resources are region-specific and cannot be used across regions through internai network connections. For low network latency and quick resource access, select the nearest region.
Project	CN East (Previous Vingle-git) V
Application Name	stable-diffusion_20250227195428
	Enter 2 to 60 characters starting with a letter and ending with a letter or digit. Only letters, digits, underscores (_, and hyphens (·) are allowed.
Runtime	Http
Agency ①	Use no agency
	Specify an agency if you want to delegate FunctionGraph to access other cloud services, such as LTS and VPC.

Step 5 Wait until the application is created. The created application contains resources such as functions, agencies, and triggers, as shown in **Figure 12-12**.

Figure 12-12 Application created

stable-dif	stable-diffusion_20241226163749							Uplied Model Access Application	Usage Description Decem
Click Access App	ication to go to the application page. The page	may take up to 30 seconds to load for the first time. Rafesh the p	page if it takes a long time.						
A The temporary d	omain name sill expire in 30 days. Delete it and	bind another one Bind If a custom damain name has been bound	A click here to refresh.						
ID Modified	496%) 9869-976690 and in a Dec 26, 2024 16:38:43 GMT+06:00	in d	Stack Name Description	5006-01.501_207755411754-01-44		Slack ID	er nationalist ne d		
Resources Code									
Only resources a	ssociated with the application template are disp	ayed.							
Cloud Service		Physical Resource Name/D 🖯			Logic Name $ \theta$		Type Θ	Status ()	
FunctionGraph		$da_1^{\prime}=(1-a_1^{\prime})a_1^{\prime}+(1-a_2^{\prime})a_1^{\prime}+(1$			createT1gger		hasveickud_tp:_tanctan_tigger	Created	
FunctionGraph		A Print Prin			obs_model_higger		hasveicloud_tp:_tanction_trigger	Created	
API Gateway (St	and APIG)	Provide a series of the series			fietrancer_apigv_	graup	hasveidoud_api_gateway_graup	Created	
API Gateway (St	and APIG(a gang ang ang ang ang ang ang ang ang a			sd_apipv_grap		hasveidoud_api_gatemay_graup	Created	
FunctionGraph		-	don default custom-models	tool 30341226083813 latest	custom_models_te	si i	haaveidout_tp:_turction	Created	
FunctionGraph		محدث المتجوب والإوالين المراجع ويوسد	tor adapt state_dfusion	_24244226083813 Maat	stabe_difusion		haaveidout_tp:_tunction	Created	

Step 6 The output of the text-to-image application is determined by the open-source models and user input. To use it legally, you need to bind a licensed domain name for open access. Click **Bind now** to bind the resolved custom domain name. Then return to the application page and click **Refresh** to start using Stable-Diffusion.

Figure 12-13 Binding a custom domain name

API Groups / sd_apigw.group.		
Summary APIs Variables Domain Names Gateway Responses		
Ensure that the domain names you want to bind to the API group meet the following conditions: • You have an independent domain name that is accessible. • The domain names much have laready been CMMAEI to be subdomain name of the API group. • UP thing much base compatible for the onion name. • UP the API group contains HTTPS- compatible APIs, add an SSL conflicate for the independent domain name bound to the group. Available domain names for binding . • Bind independent Domain Name	Bind Independent Domain Name * Domain Name state: gitting ok Cancel	×



12.4 Application Use

The output of the text-to-image application is determined by the open-source models and user input. To use it legally, you need to bind a licensed domain name

for open access. Click **Bind now** on the application details page to bind the custom domain name. For details, see **Configuring the Domain Name for Calling APIs**. Open the bound domain name in the browser or click **Access Application** to access the Stable Diffusion WebUI, it takes a long time to start the system for the first time.

Figure 12-14 Getting started with Stable-Diffusion

stable-dit	fusion_20241226163749	Ubbert Middel Access Approximent Charge Decorptions Contex				
Click Access Ap	sication to go to the application page. The page may take up to 30 seconds to load for the first time. Rathooh the	age if it takes a long time.				
A The temporary of	omain name will expire in 30 days. Delete it and bind another one Sint If a custom domain name has been bour	Lolochere ta refresh-				
ID	AUGU HOBINGHINGHINGH (Slack Name	state-dfusion_20041226103743-ajalle x_	Stack ID	Contractions and a state of the	
Modified	Dec 26, 2024 10 38 43 GMT+08.00	Description				

Figure 12-15 Stable-Diffusion WebUI

dvilloutnie, Villenneef 506H sudietsmaan (1994) • 🕖 Automatic • 😥			
Marken Janes Alexa Marken Calebrahaman Tala Antonio Antonio			
uang mang cius menio ukupinungin nan selatip banana	6/75		
materpiese, best quility, ney detailed, externely detailed beautiful, super detailed, busiter has, likatorion, dynamic angles, phy, bestorschring, standing, mannepain, looking at viewe; interview, beautiful detailed yees, expluidary beautiful face, fasting, high-standards, beautiful and tetailed light and stadow			Generate
	-		
Silvaluegister, Indunde Lapplegninger, geoprepring, u.J. 1921-sel badpompt umphadmagenegning, u.J. Lappging, Jand Segmantatel kinds and fragen party draws floce, etc. limit, mosing limit, discovered and using limit, discovered	0/75	<u> </u>	
			ו /
Generation TestalTherative Representation Oristypains Loss			
Singling method Savepling tases 20			
Lórs ·			
Here fa 🔹 4 Refer			
Nah 312 Bachaust 1			
ngt 12 Batsia 1			
05sak [1]			
Sed .			
-1. Q 🌢 🗆 Gra			
12202402			
Sola			
104			

NOTE

If reserved instances are not used, the first cold start takes about 30 seconds. If the loading times out, refresh the page.

12.5 Custom Models

The Stable-Diffusion application supports custom models. You can upload your own model to the corresponding path of a specified OBS bucket. The auxiliary function will forward the model to the shared path of SFS Turbo. And you can use it after reloading the page.

12.5.1 Initializing a Model

- Step 1 Log in to the FunctionGraph console and select CN East-Shanghai1 region. In the navigation pane on the left, choose Application Center. Click the name of the target application in the list. The Summary page is displayed.
- Step 2 On the Summary page, click Initialize Custom Model to start the configuration.

Figure 12-16 Initializing a custom model

stable-diff	usion_20250227200427				Vitable Caston Model From Application Usage Description Cores
Click Access Acel	cution to go to the application paper. The paper may take up to 30 seconds to load for the first time. Refeats the p	ege II it teles e long time.			
A The temporary da	main name will expine in 30 days. Delete it and bind another one Bind if a custom domain name has been bound	diachere la refrest.			
ID Motified	10800 (Frankinski alb. 1480 audieta (f 140 alf. 2010 120 122 (MT+05.00	Stack Name Description	stable-diffusion_20050027200127Tile	Stack ID	$a \in M$ where the theorem $\mathcal O$

Step 3 Select the created **VPC**, **subnet**, and **file system**. The default function access path is **/mnt/auto**. Set other parameters based on service requirements. After the configuration is complete, click **OK** and wait until the initialization is complete.

Initialize Custo	om Model	\times
SFS Turbo	Manually configure	
* VPC	vpc-II VPC []	
* Subnet	subnet V Q Create Subnet	
File System Type	SFS Turbo SFS ECS Scalable File Service (SFS) provides SFS and SFS Turbo file systems. The SFS Turbo file system supports the following storage classes: Standard, Standard-Enhanced, Performance, and Performance-Enhanced.	
★ File System	-Select- V Q Create File System [2]	
* Shared Directory	Enter a path.	
* Access Path	/mnt/auto	
	Location where the file system is mounted in the function. Set a new two-level directory that starts with /mnt or /home, or a single-level directory.	
	Cancel	

Figure 12-17 Configuring custom model initialization

Step 4 After **step 3** is complete, click **Upload Model**. The file management page is displayed. The default username and password are **admin**. Change the password after login to ensure data security.

Figure 12-18 File management

🕒 Q Search			○ III ± ± @ Ø
My files	ń		
In New folder	Name ψ	Size .	Last modified
New file	at the set		3 days ago
Settings	estpets		18 days ago
	top log		a month aga
17.1 0/5 of 500-0/8 used	E detabase.cb	64 KB	a few seconds ago
Fin Brawner (antracied) Help	configian	176.8	15 daya aga

Step 5 Table 12-1 lists some key directories.

Table 12-1 Key directory path

Path	Description
sd/models/Stable-diffusion	Stores Checkpoint model files.
sd/models/VAE	Stores VAE files.
sd/models/Lora	Stores LoRA models.
sd/extensions	Stores plug-ins.
sd/outputs	Stores outputs.

----End

12.5.2 Importing and Loading Models

Step 1 Upload the custom models, such as Checkpoint, VAE, and LoRA models, to the designated path.

Figure 12-19 Checkpoint models

k > od > models > Stable-diffusion						
Name \downarrow	9120	Last modified				
≥ readme.md	invalid link	3 days ago				
MR 3DQ _SDXL V0.2.safetensors	6.62 GIB	6 months ago				
MR 3DQ_1.************************************	5.28 GIB	7 months ago				
chilloutmix_NIPrunedFp16Fix.safetensors	invalid link	3 days ago				
atomixAnime_v20.safetensors	1.99 GiB	3 daya ago				

Figure 12-20 VAE models

♠ > od > models > VAE		
Name U	529	Last modified
vae ft-mse-840000-ema-pruned safetenoors	319.14 MB	3 days ago
vae 84m pruned, 2.0. salitemeors	319.14 MB	3 daya ago
by Put your Will release project folder here to:	invalid link	3 days ago

Figure 12-21 LoRA models

t>	ad > models > Lora		
Nam	\checkmark	Size	Last modified
	V_1.0.safetensors	144.11 MiB	3 days ago
	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	144.11 MiB	3 daya ago
		144.11 MiB	3 days ago
69	Put your Lora release project folder here.txt	invalid link	3 days ago

Step 2 After the upload is complete, return to the Stable-Diffusion WebUI and reload the model. The loading may take a long time and after that the new models will be displayed.

Figure 12-22 Checkpoint models displayed

Stable Diffusion checkpoint	SD VAE			
atomixAnime_v20.safetensors [680f76631a] -	Automatic		•	
 atomixAnime_v20.safetensors [680f76631a] chilloutmix_NiPrunedFp16Fix.safetensors [59ffe2243a] MR 3DO 1.5 XXX V2.safetensors [5025a4d3ed] 	Checkpoint Merger	Train	Python Module Installer	Setti
MR 3DQ _SDXL V0.2.safetensors [be7b2c345e] (ai	led beautiful, super deta	ailed, tousled h	nair, illustration, dynamic angle	s. girly. fasl

Figure 12-23 VAE models displayed

Stable Diffusion checkpoint		SD VAE			
atomixAnime_v20.safetensors [680f76631a] 🔹		Automatic 🗸		0	
		Automatic	́Т)		
txt2img img2img Extras PNG Info	Check	None	er	Settings	Extensions
masterpiece, best quality, very detailed, extremely det	ailed beau	vae-84n-pruned_z.o.saletensors	gles	, girly, fashion clo	othing, standing

Figure 12-24 LoRA models displayed



----End

12.6 Advanced Use

12.6.1 Using ECS as an NFS Server to Isolate Resources of Multiple Users

Using ECS as an NFS Server to Isolate Resources of Multiple Users

In addition to SFS file systems, NFS shared paths on ECS can also be mounted to FunctionGraph functions. Using ECS makes it easier to manage resources for multiple tenants.

- **Step 1 Buy an ECS**. Pay attention to the following aspects:
 - Select CN East-Shanghai1 for Region. You can select the ECS specification and image version based on service requirements. EulerOS 2.5 64bit(40 GiB) is used as an example. Some Linux commands vary depending on the image version.

Figure 12-25 Basic information



• The size of many model files ranges from 1 GB to more than 10 GB. It is advised to select a system disk or mount a data disk as required. Click **Next: Configure Network**.

Figure 12-26 Configuring a system disk

System Disk	General Purpose SSD	v	-	40	+	GiB IOPS limit: 2,280, IOPS <u>burst limit</u> : 8,000 ③	Show ✓ Enabled SCSI
[Add Data Disk You can a	attach 59) more S	SCSI disk	S.		

• Select the VPC and subnet created in the previous step.

Figure 12-27 Configuring the VPC and subnet

Configure Basic Settings	Configure Network	Configure Advanced Settings	(4) Confirm
Network	vpc	Q subnet-	Q Automatically assign IP address v Available private IP addresses: 239
	Create VPC 🕑		
Extension NIC	Add NIC NICs you can still add: 1		
Source/Destination Check	0		

• For details about how to configure a security group, see the following figure. For inbound rules, enable ports 111, 2049, 2051, 2052, and 20048 for the IP address segment in the subnet for the NFS service. Configure other ports, such as port 22 for SSH and SFTP and port 21 for FTP.

Figure 12-28 Configuring a security group

Summary	Inbound Rules Outbound P	lates Associated Instances	Tag						
0	Some security group rules nill not take effect If the source is set to 0.0.0.00 pr.:0, then a security group rule.	t for ECSs with certain specifications. Laws I enternal IP addresses are either allowed or	nove 🗹 denied to access your instances, depending-	on if the action is Allow or Dany. If the access	is allowed, exposing high-risk ports, such as	pot 22, 3338, or 6846, to the public network will have your instances nutreeable to network	i htrusions, service interuptions, date leaf-ag	e, or tansomnane affacts. You should only co	rigure known IP addresses for the
	di Rult Del Hall Rult	Allow Comman Parts (Batch Operations Mare ~	Inbound Rules: 9 Vew Security Group Co	ntiguration Examples 🕑				
	Select a property or entire a keyword.								0.0
	Ptority	Sidus	Action	Type	Protocol & Port	Source	Description	Last Modified	Operation
	1	O Enabled	Allow	IN .	UDP: 20048	reduces.	NF8	Dec 84, 2024 15 31 40 GMT-38 08	Modly Replicate More
	1	O Enabled	Abov	ini	ube: 111	1.4.7.7	NFS	Dec 84, 2024 15 01 31 6MT+9E 00	Modly Replicate More v
	1	O Enabled	Allow	ins.	TOP. 2851		N°5	Dec 84, 2024 15 01 23 OMT-98 00	Modly Replicate More ~
	1	O Enabled	Allow	154	TOP 2852	NAME:	N°5	Dec 84, 2024 15 01 05 0MT-08 08	Modly Replicate More -
	1	O Enabled	Allow	154	TOP: 111	44 -	N°5	Dec 84, 2024 15 00 55 OMT-08 08	Modly Replicate More -
	1	O Enabled	Allow	154	TCP: 2949	Annuals.	N°5	Dec 84, 2024 15 00 10 OMT-08 08	Modly Replicate More -
	1	O Enabled	Allow	ins.	TCP. 20048	1000	NF5	Dec 84, 2024 15 00 01 DMT+08.00	Modly Replicate More
	1	O Enabled	Alex	м	н	1-4.0	constant and parts	Aug 81, 2024 10 53 55 CMT-08 08	Modly Replicate More ~
	4	O Enabled	Allow	154		eres is	.00-00.00000000000000000000000000000000	Aug 01, 2024 10 53 55 DMT-08 08	Modly Replicate More -

• Purchase an EIP based on service requirements.

Figure 12-29 Purchasing an EIP

EIP	Auto assign Use existing Not required ⑦
EIP Type	Dynamic BGP Static BGP
	⊘ Greater than or equal to 99.95% service availability rate
Billed By	Bandwidth 🝁 🗠 Traffic 🗠 For light/sharply fluctuating traffic 🗠 For staggered peak hours 0
	Billed based on total traffic irrespective of usage duration; configurable maximum bandwidth size.
Bandwidth Size	5 10 20 50 100 Custom - 5 + The bandwidth can be from 1 to 300 Mbit/s.
	Anti-DDoS protection ③ Free
Release Option	Release with ECS
	If you select this option, the EIP will be released when the ECS is deleted.

Step 2 Configure NFS sharing.

After purchasing an ECS, you can configure NFS sharing. The following uses user1 and user2 as an example.

1. Add users user1 and user2 and create the **home** directory.

useradd -m user1 && useradd -m user1

- 2. Change the password. passwd user1 passwd user2
- 3. Create a shared directory for users and change the operation permission on the shared directory to 777.

mkdir /home/user1/share && chmod 777 /home/user1/share mkdir /home/user2/share && chmod 777 /home/user2/share

NOTE

The shared directory is a subdirectory of the **home** directory and inaccessible to other users. This ensures that the function has the operation permission after mounting the directory. Therefore, permission 777 will not cause unauthorized operations.

- 4. Install the NFS service. yum install rpcbind nfs-utils // Run the corresponding command for images that use apt or other package management tools.
- 5. Add the following content to the **/etc/exports** file: /home/user1/share xx.xx.xx/xx(rw) // Fill xx.xx.xx/xx with the network segment of the created

subnet. /home/user2/share xx.xx.xx/xx(rw) // Fill xx.xx.xx/xx with the network segment of the created subnet.

- 6. Start the NFS service. systemctl start rpcbind nfs
- 7. Enable the NFS service to automatically start upon system startup.

echo "xx.xx.xx.xx:/home/user1/share /nfs nfs4 defaults 0 0" >> /etc/fstab // Enter the IP address of the ECS in the subnet.

echo "xx.xx.xx./home/user2/share /nfs nfs4 defaults 0 0" >> /etc/fstab // Enter the IP address of the ECS in the subnet. mount -av

8. Check the sharing information. If the following is displayed, the NFS sharing is successfully configured.

showmount -e xx.xx.xx. (IP address of the server)

Figure 12-30 Checking sharing information

[root@ecs-nfs-share ~]# showmount -e
Export list for 2 239.
/home/user2/share0/24
/home/user1/share :
/nome/snare 0/24
[root@ecs-nfs-share ~]#

----End

Mounting an Application

In the FunctionGraph application center, create a Stable-Diffusion application for user1 and user2, respectively. The following uses user1 as an example. The procedure for user2 is the same.

Step 1 Go to the Summary page of user1 application, find the function whose logical name is stable_diffusion or custom_models_tool in Resources, and click the link to go to the function details page. The operations of the two functions are the same. The stable_diffusion function is used as an example.

Figure 12-31 Functions of user1

stable-diffusion_2025022	7200427				Initiative Cuntom	n Model Access Application Usage Description	Deep
Click Access Application to go to the application page. The p	age may take up to 38 seconds to load for the first time. Mehesh t	he page # it blies a long tim					
A The temporary domain name sell ergins in 38 days. Delete it	and bind another one $\operatorname{Bind} \mathbb{X}$ a custom domain name has been by	nord, click have to referch.					
D teltup/vie + silenieu profes wodied Pet 27, 2005 20.05 22 GWT+00.0	0 40 C	Stack Name Description	state-offsee_2005027200427 Tille I -	Stack D	and the of the sile of a		
Resources Code							
Only resources associated with the application template are	feelined.						
Cloud Service	Physical Resource Name/D 0			Logic Name ()	Type 0	Status 0	
Functiondrage	endowed particular definition of			Custors_nodels_tax	hameclast jgs_function	Created	
API Gatevay (Shand APIG)	Distance on the Walker Units			febraroer_apipv_group	humeichad api galevay group	Created	
API Galevily (Shared APIG)	cl-s. cover. 2006. Andres et. chelses 5			sd_apips_prop	humeichad api galevay groe	Created	
FunctionGraph	and the set of the state of the state of the		policy postation	stable_diffusion	hummiciaed_tps_function	Created	
FunctionDraph	(4) module Window associated			create Tripper	hummeiched_lps_function_tripper	Created	
FunctionGraph	2011 Children Children (dm 1073)			ote, model, traper	hummiched Ap. Amoton, byper	Created	

Step 2 On the function details page, choose Configuration > Network, enable VPC Access, configure the VPC and subnet, select the same VPC and subnet as those of ECS, and click Save.

Figure 12-32 Configuring network

Network Network Togos • •	Code Monitoring Version	Alases Configuration	
Type: • Dis Add costs: website addition to the back the Casses the site addition. The back the the states. The back the the addition to the back the distance and the back the distance addition. The back the states the back the distance addition. The distance addition to the back the distance addition. The distance addistance addition. The distance additistance addited addit	Basic Settings	Network	
unitability PARL/CODE PARL/CODE binder: 0 0 binder: 0 0 conserver VPC / Acces Image:	Triggers	Public Access is enabled by Enabling VPC Access user	y default. The default NC accesses the pablic network with shared bandwidth. This access model is subble for lever aquests. For high bandwidth, performance, and reliability, enable VPC Access, add a public NAT galaxies, and bind an DP with evolutive bandwidth to it. The NC bound to the VPC instead of the default NC. The VPC determines withine public access is available.
Def Angel VC Angel Defaunce Volume VC Demander Volume VC Demander Volume VC Defaunce Volume VC Defaunce Volume Defaunce Volume	Network	Public Access	
Ownery V/C Testing and testing an	Disk Nounting Environment Variables	VPC Access	
United processing Description Description <thdescription< th=""></thdescription<>	Concurrency	VPC	
Likeols Dotatil hare	Logs	Subret	
	Lifecycle Advanced Settings	Domain Name	dends Q Overla Down have (C Prade down reads of her VPC to lowed a DME APTs receive private downer haven, yes mad peoply an agency with the permissioned to read DME recourses
VPC CORI Block Une sensative (1) in suparts CORI Block Example Enter the VPC CORI Block and in the code to deals with FundamOrgen's VPC CORI Block. No. 5 CORI Block.		VPC CIDR Block	Une servicine () is separate ODB tools, Europe. Earl the VPC COR tools used in the oak is chose where it centrals with Readoningers VPC COR tools. Not 5 COR tools.
Invester Only in Stands VPC 🔊		Invocation Only by Specific VPC	

Step 3 In the navigation pane, choose **Disk Mounting > Mount File System**.

- File System Type: Select ECS.
- **ECS**: Select the ECS used to configure the NFS sharing.
- Shared Directory: Enter /home/user1/share (e/home/user2/share for user2).
- Access Path: Enter /mnt/auto.

Figure 12-33 Mounting a file system

Mount File Syste	em X
File System Type	SFS Turbo SFS ECS
	You need to enable the NFS service for ECS to share a specified directory, which can be mounted to functions in the same VPC.
* ECS	C-Select V Q Create ECS 🖸
* Shared Directory	Enter a path.
	Enter an existing path.
* Access Path	This field cannot be left blank.
	Location where the file system is mounted in the function. Set a new two-level directory that starts with /mnt or /home, or a single-level directory.
Cancel	

Step 4	After the	configuration i	is complete,	click OK.

----End

Accessing the WebUI Program to Create Directories and Files

- **Step 1** Go to the **Summary** page of user1, click **Access Application** in the upper right corner, and wait until the function is started. The function automatically creates a directory required by the application in the mounting directory.
- **Step 2** Go back to the **Summary** page of user1 application and click **Upload Model** to open the file management tool.

Figure 12-34 File management tool

th⇒ ad		
Name ψ	Size	Last modified
textual_inversion_templates		3 days ago
scripta		3 daya ago
root root		3 days ago
repositories		3 days ago
cutputs		3 days ago
models		3 days ago
localizations		3 days ago
extensions-builtin		3 days ago
extensions		3 days ago
embeddings		3 days ago
Configs		3 days ago
<> uFoorfig (son	86.52.108	a few seconds ago
III atylan.cav	0.8	3 daya ago
configuron	10.68 KB	2 minutes ago
conteijon	166.15 K0B	3 daya ago

- **Step 3** Save your model and plug-in files to the corresponding directories. The following is an introduction of some main directories.
 - **sd/models/Stable-diffusion**: Path for storing Stable-Diffusion Checkpoint model files.
 - **sd/models/VAE**: Path for storing VAE files.
 - **sd/models/Lora**: Path for storing LoRA models.
 - **sd/extensions**: Path for storing plug-ins.
- **Step 4** Reload the WebUI. The newly imported models are displayed.

Figure 12-35 Checkpoint models

e officiale checkpoint 52/VK			
1ješlozenic j. NPosedlj Ližije Lijnočno 🔹 🚺 dlškoline 2. Kolcépt 🔹 🔯			
töring langsting Datas Miljinfo Chedipalitilitinger Tanin Settings Datessions	470		
namepies, best publy, wy datalied, externels datalied boundhil, super datalied, tassied hais, litanistics, dynamic angles, gifs, tubion clothing, namequies, boiling at viewet interview, board, baardhil datalied yes, expediately boardhil hea, floating, high saturation, boardhil and datalied light and shadow		Gen	wate
	4/75		
0.m/s/opport.hathandirLog/ingetivezg.despenytive_v/_17/zmc2/adopting/out/out/magnegative_v1_2.negative_hand-regunsioned hands and fingers_pointy down locustos links, insing links disconnected links, and levense hands, and y			8+ 1
Georation TodualInversion Hypernetworks Oreclapoints Lon			
NO: NO: NO: NO:			
DREVTEN DREVTEN DREVTEN DREVTEN			
TREVEN TREVEN TREVEN			
0.7(chilloutnix, NPrun olehotes): 0.7 (chilloutnix, Diffruentis, Diffruentis,			
oV4_v4) p16Fix mixProV4_v4 sd v1.5-inpainting v1.5-pruned emaonly			





Step 5 Click **Generate** in the upper right corner. The image is automatically saved to the **/home/user1/share/sd/outputs/txt2img/202x-xx-xx** directory.

Figure 12-37 Directory for saving images

5 5 outputs 5 bt2ing 5 2024-02-20		
Name 🗸	Size	Last modified
60002-538029379007233 prg	554.66 108	2 days ago
🚷 00001-3869566415.png	388.49 108	2 days ago
00000 2211850919.png	390.37 108	2 days ago

----End

12.6.2 API Access

By default, API access is disabled for the deployed Stable-Diffusion application. To enable it, configure and save the following environment variable. For details, see **Configuring Environment Variables**.

Table 12-2 Environment varia

Кеу	Value
EXTRA_ARGS	apiapi-auth username1:password1,username2:password2nowebui

Configure concurrency parameters for the Stable-Diffusion function by referring to **Configuring Multiple Concurrency for a Single Instance**. The recommended parameters are as follows:

- WebUI mode
 - Requests per instance \geq 100. According to the test result, the number of requests for a single instance is about 15, and you are advised to set the number for multiple users to greater than 100.
 - Max. instances per function: 1. When generating graphs in WebUI mode, the progress is checked continuously. However, if there are multiple instances, the requests may be sent in a disorderly manner, which prevents the progress from being displayed and the final result from being viewed. To avoid this, it is advised to set the maximum number of instances for a single function to 1.
- API mode
 - Requests per instance: 1–5. This ensures that an instance does not have too many queuing requests. When the number of concurrent requests

reaches the threshold, a new instance is displayed to ensure the graph generation speed.

- Maximum instances per function: 400 (default). You can change the value as required.

Using Moderation to Review the Generation Result

Stable-Diffusion is an AIGC inference model. The final result of image generation may be uncertain due to different prompts and models, which may cause pornographic and violent violation risks. You are advised to use Stable-Diffusion with Huawei Cloud Moderation to review the generated result. For details, see **Image Moderation (V3)**.

12.6.3 Enabling WebUI Authentication

The WebUI authentication is disabled by default. To prevent your functions from being stolen due to domain name leakage, you can configure environment variables for Stable-Diffusion functions to enable WebUI authentication and for **API Access** at the same time. For details, see **Environment Variables**. After the setting is complete, refresh the WebUI. Enter the username and password to start drawing.

Table 12-3 Environment variable

Кеу	Value
EXTRA_ARGS	gradio-auth user1:password1

12.6.4 Sharing Models and Plugins

Using ECS as an NFS Server to Isolate Resources of Multiple Users describes using an ECS as the NFS server to isolate resources when multiple users use the Stable-Diffusion WebUI. This solution is applicable to scenarios where strong isolation is required between users.

In some scenarios, if you want multiple users to share some resources, such as model files and plug-ins, the storage space is wasted if each user copies a file. This problem can be solved by mounting different applications to the same SFS file system. However, user operations and configurations may interfere with each other, perform the following operations to solve it:

The following operations take user1 and user2 in Using ECS as an NFS Server to Isolate Resources of Multiple Users as an example. You need to create a Stable-Diffusion application for each of the two users and use the same SFS file system to initialize the custom model. For details, see Application Creation and Deployment and Custom Models. In this case, the two users have shared the model and plug-in. The subsequent steps will solve the operation and configuration isolation problem.

Creating a Configuration File

Open the file management page of any user by referring to **step 4** in **Initializing a Model**. After login, go to the **sd** directory. If the **sd** directory does not exist or the **config.json** file does not exist in the directory, start the WebUI of any user by referring to **Application Use** and perform operations in this document again.

Figure 12-38 Entering the sd directory



Locate the **config.json** file, copy it to the **sd** directory, and rename it to **config_user1.json**. Repeat for user2, naming the copied file to **config_user2.json**.

Figure 12-39 Copying the config.json File

						< 🖌 🗋 + 🗉	
☆ > sd						<u> </u>	
textual_inversi	scripts 2 months ago	root 	repositories 2 months ago	python T4 days ago	outputs 4 days ago	models 2 months ago	localizations 7 months ago
extensions-buil	extensions 	embeddings 2 months ago	configs 				
B6.92 KB an hour ago	styles.csv 0.8 2 months ago	sd-webui-new 14.33 GB a month ago	SB 381 B 2 months ago	<pre>config_1.json 10.71 KB 3 hours ago</pre>	Config json 10.71 KB 2 hours ago	Cache.json 166.66 KiB a month ago	

Figure 12-40 Creating the config_user1.json and config_user2.json files

							↔ # ± ± 0 Ø
♠ > ed							
textual_inversi 2 months ago	2 months ago	root 2 months ago	repositories 2 monthe ago	python 14 days ago	outputs 4 days ago	models 2 months ago	localizations 2 months ago
extensions-buil 	extensions 	embeddings 	configs - 2 months ago	1	1		
billion disconting ison 86.92 108 an hour ago	B 2 months ago	sd-webui-new 14.33 GB a month ago	<pre>index.py 381 B 2 months ago</pre>	config_user2.js 10.71 KB a few seconds ago	<pre>config_user1.js 10.71 KB 3 hours ago</pre>	<pre>config.json 10.71 KB 2 hours ago</pre>	cache.json 166.66 Ki8 a month ago

Modifying Environment Variables and Using the New Configuration File

In **Creating a Configuration File**, you have created a configuration file for each user. Now, modify environment variables to enable different users to use their own configuration files. Configure the following environment variables for the Stable-Diffusion function by referring to **Environment Variables** and save the settings (you can set the environment variables together with **API Access** and **Enabling WebUI Authentication**). The following are the environment variables of user1 and user2.

	Table 12-4	Environment	variables	of user1	and user2
--	------------	-------------	-----------	----------	-----------

Кеу	Value
EXTRA_ARGS	ui-settings-file=/mnt/auto/sd/config_user1.json
EXTRA_ARGS	ui-settings-file=/mnt/auto/sd/config_user2.json

Changing the Path for Saving Images

After the preceding configurations are complete, the two users can share models and plug-ins without affecting each other. To further isolate the inference results of different users, you can change the result save path in the settings.

12.6.5 Using a Dedicated APIG Trigger

The Stable-Diffusion application created in the application center uses the shared APIG trigger. Resources such as bandwidth are shared by all shared APIG users. Therefore, if your service has higher requirements, you can use the dedicated APIG trigger. The procedure is as follows:

Buying a Dedicated Gateway

Purchase a dedicated APIG gateway based on your service requirements. For details, see **Buying a Gateway**. Note that you need to enable public access to the WebUI and set the public access bandwidth as required.

Configuring a Dedicated APIG Trigger

- Step 1 Go to the details pages of the two functions by referring to step 1 in Mounting an Application. The operations for the two functions are the same. This section uses the stable-diffusion function as an example.
- Step 2 On the function details page, choose Configuration > Triggers > Create Trigger. Select API Gateway (Dedicated Gateway) for Trigger Type, the purchased instance for API Instance, different groups for the two functions, None for Security Authentication, and HTTPS for Protocol. Set Timeout to 60,000 and retain the default values for other parameters, and click OK.
- **Step 3** Delete the shared APIG trigger created by the system by default.

----End

Modifying the Trigger Backend Timeout

- Step 1 Modify the value of **backend_timeout** to 600,000 by referring to Configuring Gateway Parameters.
- Step 2 Go to the details pages of the two functions by referring to step 1 in Mounting an Application. The operations for the two functions are the same. This section uses the stable-diffusion function as an example.
- **Step 3** On the function details page, choose **Configuration** > **Triggers**, and click the trigger name to go to the API management page.
- Step 4 Click Edit > Next to go to the Define Backend Request step, change the value of Timeout (ms) to the required value (the maximum value is 600,000, that is, 600 seconds), and click Finish.
- **Step 5** After the modification is complete, return to the API management page, click **Publish** in the upper right corner, and then click **Publish**.

----End

Binding a Domain Name

Bind domain names to the APIG groups of your two functions by referring to **Configuring the Domain Name for Calling APIs**.

12.7 Disclaimer

- All projects used in this application, such as Stable-Diffusion, Stable-Diffusion-WebUI, and the Image Building Project, are open-source community projects. Huawei Cloud only provides computing power support.
- 2. This application is only for reference and learning purposes. To use it in a production environment, optimize it by referring to the image building project. If you encounter any issues while using FunctionGraph, submit a service ticket. For questions related to open source projects, seek help from the open source community.
- 3. A gateway will be created on APIG upon application deployment. Bind a custom domain name as prompted and use it to access the WebUI.

$13_{\text{Building an HTTP Function with Go}}$

Introduction

This chapter describes how to deploy services on FunctionGraph using Go.

HTTP functions do not support direct code deployment using Go. This section uses binary conversion as an example to describe how to deploy Go programs on FunctionGraph.

Procedure

Building a code package

Create the source file **main.go**. The code is as follows:

```
// main.go
package main
import (
  "fmt"
  "net/http"
  "github.com/emicklei/go-restful"
func registerServer() {
  fmt.Println("Running a Go Http server at localhost:8000/")
  ws := new(restful.WebService)
  ws.Path("/")
  ws.Route(ws.GET("/hello").To(Hello))
  c := restful.DefaultContainer
  c.Add(ws)
  fmt.Println(http.ListenAndServe(":8000", c))
}
func Hello(reg *restful.Request, resp *restful.Response) {
  resp.Write([]byte("nice to meet you"))
func main() {
  registerServer()
# bootstrap
/opt/function/code/go-http-demo
```

In **main.go**, an HTTP server is started using port **8000**, and an API whose path is / **hello** is registered. When the API is invoked, "nice to meet you" is returned.

Compiling and packaging

1. On the Linux server, compile the preceding code using the **go build -o go**http-demo main.go command. Then, compress **go-http-demo** and **bootstrap** into a ZIP package named **xxx.zip**.

```
2. To use the Golang compiler to complete packaging on a Windows host, perform the following steps:
# Switch the compilation environment
# Check the previous Golang compilation environment
go env
# Set the following parameters to the corresponding value of Linux
set GOARCH=amd64
go env -w GOARCH=amd64
set GOOS=linux
go env -w GOOS=linux
# go build -o [target executable program] [source program]
# Example
go build -o go-http-demo main.go
# Restore the compilation environment
```

set GOARCH=amd64 go env -w GOARCH=amd64 set GOOS=windows go env -w GOOS=windows

Creating an HTTP function and uploading code

Create an HTTP function and upload the **xxx.zip** package. For details, see **Creating an HTTP Function**.

Creating an APIG trigger

Create an APIG trigger by referring to **Using an APIG Trigger**. Set the authentication mode to **None** for debugging.

Figure 13-1 APIG trigger

API Gateway (APIG) (Subto	tal: 1)			
Created: Apr 25, 2	Enabled 2024 14:53:40 GI	MT+08:00		Delete
URL https://fa508db2d			com/test_tactics_12 🗇	
API Group: functiongraph		Environment: RELEASE	Security Authentication: NONE	
Method: ANY		Path: /	Timeout: 5,000 ms	

Invocation test

Copy the URL of the APIG trigger and the **/hello** path registered in the code to the address box of the browser. The following information is displayed.



← C ∩ ∩ https:// .apig .apig .apig .com/hello

14 Using FunctionGraph HTTP Functions to Process gRPC Requests

Introduction

This chapter describes how to use gRPC and process gRPC requests in FunctionGraph.

This chapter uses **example/helloworld** in the **gRPC example code** project as an example to describe how to process gRPC requests with HTTP functions in FunctionGraph. Since HTTP functions do not directly support Go code deployment, this chapter provides an example of using binary conversion to deploy a Go program on FunctionGraph.

NOTE

- This feature is supported only in the LA-Santiago region.
- By default, you do not have gRPC permissions. To use gRPC, **submit a service ticket** to add your account to the whitelist.

Procedure

1. Building a code package

Create the source file **main.go**. The code is as follows: // Package main implements a grpc_server for Greeter service. package main

```
import (
    "context"
    "flag"
    "fmt"
    "log"
    "net"
    pb "helloworld/helloworld"
    "google.golang.org/grpc"
)
var (
    port = flag.Int("port", 8000, "The grpc_server port")
)
// server is used to implement helloworld.GreeterServer.
```

```
type server struct {
   pb.UnimplementedGreeterServer
// SayHello implements helloworld.GreeterServer
func (s *server) SayHello(ctx context.Context, in *pb.HelloRequest) (*pb.HelloReply, error) {
   log.Printf("Received: %v", in.GetName())
   return &pb.HelloReply{Message: "Hello " + in.GetName()}, nil
}
func main() {
   flag.Parse()
   lis, err := net.Listen("tcp", fmt.Sprintf("127.0.0.1:%d", *port))
   if err != nil {
     log.Fatalf("failed to listen: %v", err)
   }
   s := grpc.NewServer()
   pb.RegisterGreeterServer(s, &server{})
   log.Printf("grpc_server listening at %v", lis.Addr())
   if err := s.Serve(lis); err != nil {
     log.Fatalf("failed to serve: %v", err)
   }
# bootstrap
$RUNTIME_CODE_ROOT/grpc-server
```

In **main.go**, a gRPC server is started using port **8000** and **helloworld.GreeterServer** is registered. When the service is invoked, **Hello XXX** will be returned.

2. Compiling and packaging

- a. On the Linux server, compile the preceding code using the **go build -o grpc-server main.go** command. Then, compress **grpc-server** and **bootstrap** into a ZIP package named **xxx.zip**.
- b. To use the Golang compiler to complete packaging on a **Windows** host, perform the following steps:

```
# Switch the compilation environment
# Check the previous Golang compilation environment
go env
# Set the following parameters to the corresponding value of Linux
set GOARCH=amd64
go env -w GOARCH=amd64
set GOOS=linux
go env -w GOOS=linux
```

go build -o [target executable program] [source program] # Example go build -o grpc-server main.go

Restore the compilation environment set GOARCH=amd64 go env -w GOARCH=amd64 set GOOS=windows go env -w GOOS=windows

3. Creating an HTTP function and uploading code

Create an HTTP function and upload the **xxx.zip** package. For details, see **Creating an HTTP Function**.

4. Creating an APIG trigger

Create an APIG trigger. For details, see **Using an APIG Trigger**. You are advised to set **Request Protocol** to **gRPC** and **Security Authentication** to **None** to simplify the debugging process.

Figure 14-1 APIG trigger

API_lys_grpc_test001 Enabled WebSocket Created: Jul 08, 2024 10.38:15 GMT+08:00								
URL grpcs://4feb417b3fad404			7					
API Group: comfyui_apig_group_20240702082047	SS R	Environment: RELEASE		Security Authentication: IAM				
Method: POST	P	Path:		Timeout: 5,000 ms				

5. Invocation test

Use Postman to debug gRPC requests.

Figure 14-2 gRPC request result

A 2015da601b704219a35be0 [*] C ↑↓ Greeter / SayHello								
Mess	age	Authorization	Metadata (4)	Service definition	Scripts	Settings		
1 2 3	-2 -2	"name": "fgte	st"					

La es Use Example Message ∨					
Respo	nse	Metadata (10) Trailers Test results			
-0					
1	Ę				
2		"message": "Hello fgtest"			
3	}				

15 Cold Start Optimization Practices

The serverless architecture features pay-per-use, auto scaling, and complexity shielding, making it a new paradigm of next-generation cloud computing. However, in real-time scenarios, cold start poses a significant challenge. When building web services with serverless, if the cold start and initialization time exceed 5 seconds, it can greatly impact user experience. Therefore, accelerating cold start and improving the user experience is a pressing issue when constructing serverless architectures.

The lifecycle of a serverless instance consists of three phases:

- Initialization: FunctionGraph attempts to reuse a previous execution environment, or if none is available, it creates resources, downloads function code, initializes extensions and runtime, and runs initialization code (non-main program code).
- **Execution**: In this phase, the instance starts to execute the function after receiving an event, and waits for the next event after the function completes.
- **Shutdown**: This phase starts if the function does not receive any calls within a period of time. In this phase, FunctionGraph closes the runtime, then sends a shutdown event to each extension, and deletes the environment.

When FunctionGraph is triggered, if no activated function instance is available, the function code is downloaded and an execution environment is created. The period from the time when a function is triggered to the time when a new FunctionGraph environment is created is called cold start. In the serverless architecture, the cold start cannot be avoided.

Currently, FunctionGraph has optimized the cold start on the system side. For details about the cold start on the user side, see the following solutions.

Memory

Given a fixed level of request concurrency, higher function memory leads to better cold start performance with more CPU resources allocated.

Cold Start with Snapshot

Cold start is quite prominent in Java applications. Huawei Cloud FunctionGraph has proposed a process snapshot-based cold start acceleration solution, which

helps you break through the performance bottleneck while involving zero or few code changes. In this solution, the execution environment is restored from a snapshot captured after initialization, avoiding complex framework startup and service initialization. The startup latency of Java applications is significantly reduced, and the performance is improved by over 90%.

You can use a Java function to enable snapshot-based cold start. For details, see **Configuring Snapshot-based Cold Start**. FunctionGraph executes the initialization code of the function, captures a snapshot of the initialization context, and then caches the snapshot after encryption. When the function is invoked and a cold start is triggered for scale-out, the execution environment is restored from the snapshot instead of an initialization process.

Code Simplification and Image Downsizing

FunctionGraph downloads function code during cold start, but larger code packages or custom images can prolong download and cold start time. Optimize your application by removing unnecessary code (using commands like **npm prune** in Node.js and **autoflake** in Python) and third-party library dependencies (test case source code, useless binary files, and data files) to speed up the download and decompression process.

Public Dependencies

When developing applications, especially in Python, third-party libraries are often included. Large dependencies can slow down cold start as they need to be downloaded. FunctionGraph offers both public and private dependency modes. Public dependencies are recommended because they are pre-downloaded to execution nodes to save time.

Warming

When an event triggers a function, if a function instance in an active state can be called, cold start can be avoided, and a response time can be reduced. You can use either of the following warming methods:

- Use timer triggers. For details, see Using a Timer Trigger.
- Use reserved instances. For details, see **Reserved Instance Management**.