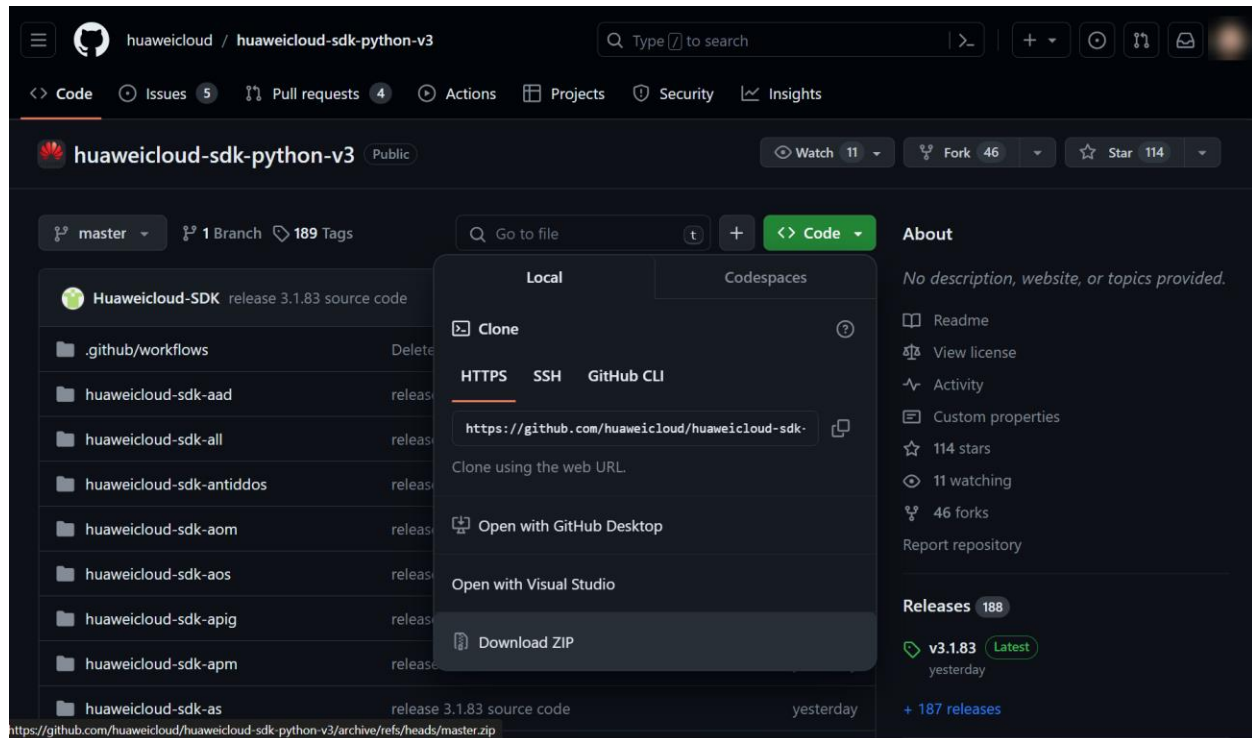


Procedimento para ativar e hibernar o CCE Cluster com o FunctionGraph

Primeiramente, acesse o repositório de SDK de Python da Huawei em [huaweicloud/huaweicloud-sdk-python-v3 \(github.com\)](https://github.com/huaweicloud/huaweicloud-sdk-python-v3) . A documentação está disponível em [huaweicloud-sdk-python-v3/README.md at master · huaweicloud/huaweicloud-sdk-python-v3 \(github.com\)](https://github.com/huaweicloud/huaweicloud-sdk-python-v3/blob/master/README.md)

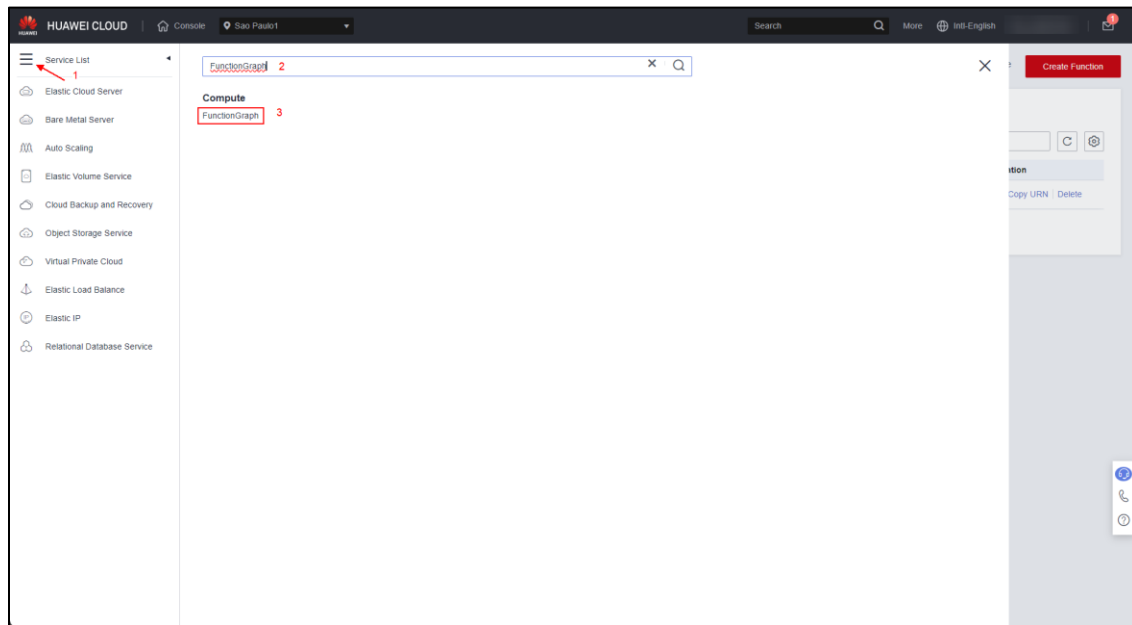
Baixe o ZIP do projeto clicando em Code e então Download ZIP



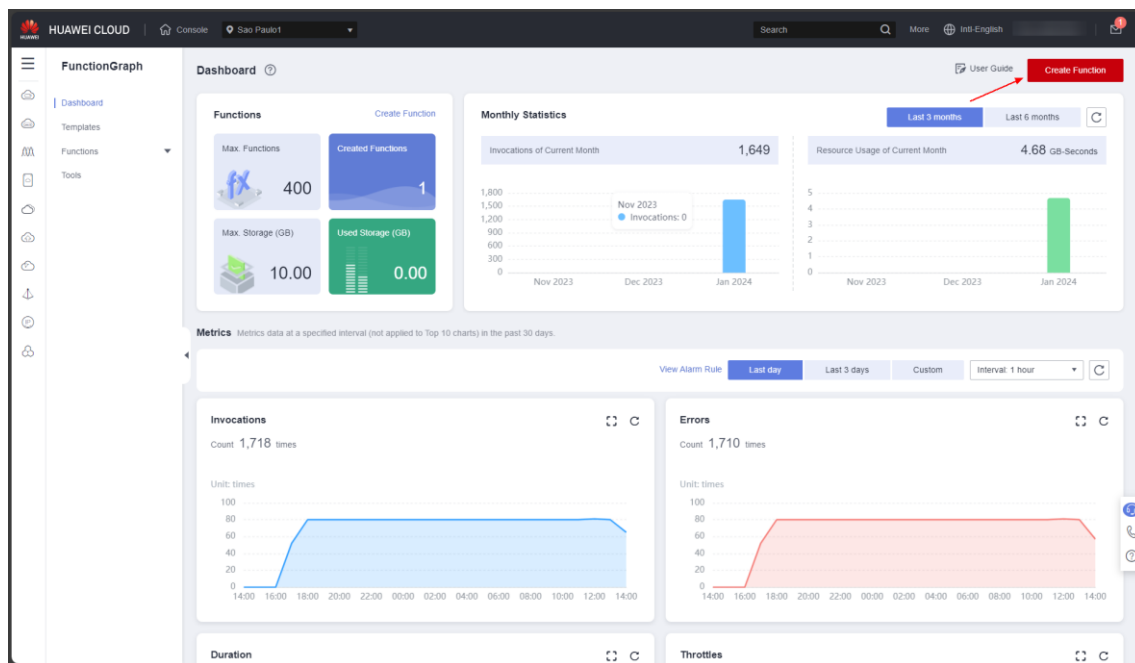
Abra a pasta e acesse o diretório **huaweicloud-sdk-cce**. Dentro deste diretório, faça a compactação da pasta **huaweicloudsdkcce** (dentro dessa pasta haverá o diretório **v3** e **__init__.py**) e salve-a em um local de fácil acesso. Essa pasta zipada será enviada posteriormente para as dependências do FunctionGraph.

huaweicloudsdkcce	2/23/2024 1:27 PM	File folder
huaweicloudsdkcce	2/23/2024 1:30 PM	Compressed (zipped)...
LICENSE	2/23/2024 1:27 PM	File
README_PYPI	2/23/2024 1:27 PM	Markdown Source File
requirements	2/23/2024 1:27 PM	Text Document
setup	2/23/2024 1:27 PM	Python Source File

No console da Huawei Cloud, clique em **Service List** no canto superior esquerdo da tela. Depois, digite **FunctionGraph** na barra de busca e então clique em **FunctionGraph** abaixo de **Compute**.



Na página do FunctionGraph, clique em **Create Function** no lado superior direito da tela.




Na página de criação da função, selecione as seguintes opções:


- Create from scratch
- Function Type: Event Function
- Region: LA-SaoPaulo1
- Project: LA-SaoPaulo1
- Function Name: (insira o nome da função)
- Agency: Use no agency
- Runtime: Python 3.9

<


Create Function



Create from scratch
Create a function with your own code.



Container Image
Select an image to deploy your function.



Select template
Create a function using the sample code.

Basic Information

* Function Type ?

Event Function

HTTP Function

Processes event requests and can be triggered by APIG, OBS, and DIS events.

* Region

LA-Sao Paulo1

Regions are geographic areas isolated from each other. Resources are region-specific and cannot be used across regions through internal network connections. For low network latency and quick resource access, select the nearest region.

* Project

LA-Sao Paulo1(Default)

* Function Name

Enter a function name.

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

Agency ?

Use no agency

Create Agency

Agencies delegate FunctionGraph to access other cloud services. For example, an agency is required when FunctionGraph accesses services such as OBS, DMS, and DIS.

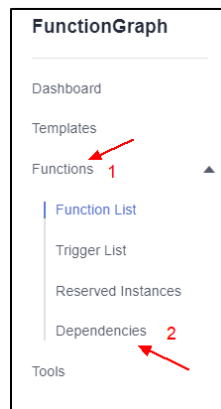
Runtime ?

Python 3.9

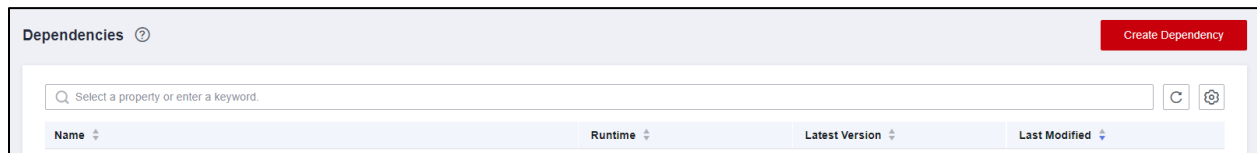
[Learn how to develop functions in Python.](#)

Select a language to compile the function. CodeArts IDE Online supports Node.js, Python, and PHP.

Na barra lateral do FunctionGraph, clique em **Functions** e então em **Dependencies**.



Na página seguinte, clique em **Create Dependency**.



Insira as seguintes opções:

- Name: (insira o nome da dependência)
- Code Entry Mode: Upload ZIP
- Upload File: (selecione o arquivo zip)
- Runtime: Python 3.9

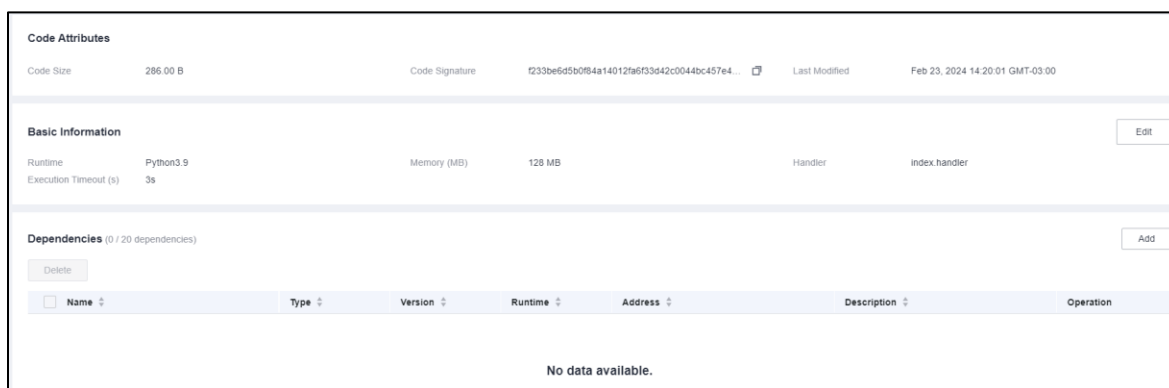
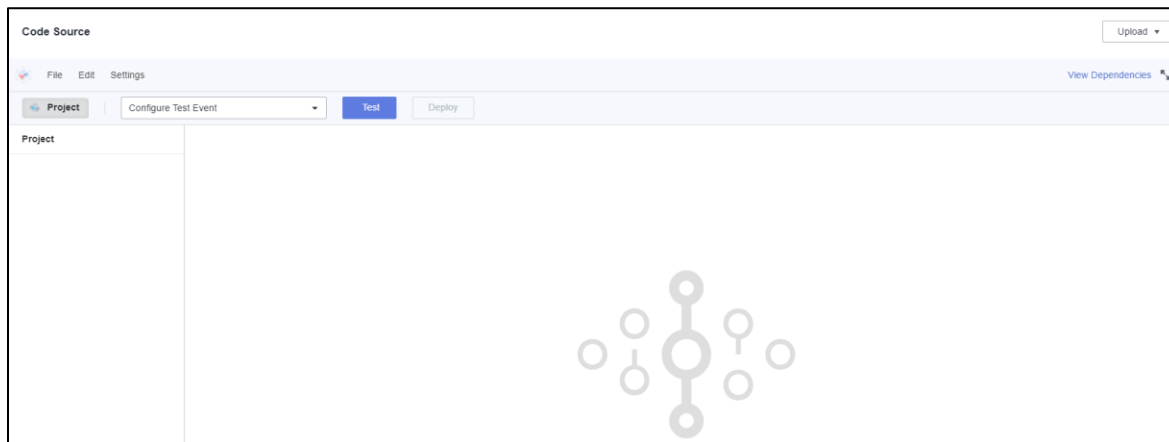
Clique em **OK**.

A screenshot of the 'Create Dependency' form. The form has the following fields:

- Name**: A text input field with the placeholder 'Enter a dependency name.' Below it, a note says: 'Enter 1 to 96 characters, starting with a letter and ending with a letter or digit. Only letters, digits, hyphens (-), periods (.), and underscores (_) are allowed.'
- Code Entry Mode**: Two buttons: 'Upload ZIP' (selected) and 'Upload from OBS'.
- Upload File**: A text input field with 'huaweicloudsdkcce.zip' and an 'X' icon to remove it. To the right is a 'Select File' button. Below it, a note says: 'For a file larger than 10 MB, upload it from OBS.'
- Runtime**: A dropdown menu with 'Python 3.9' selected.
- Description**: A text area with the placeholder 'Enter a description.'

At the bottom right, there is a character count '0/512'.

Acesse a função criada navegando novamente em **Functions, Function List** e então selecione a função. O console da huawei disponibilizará todas as opções para configurar e monitorar sua função. Existem quatro secções principais: **Code Source**, **Code Attributes**, **Basic Information** e **Dependencies**.



Na área de **Dependencies**, clique em **Add**. Uma secção chamada **Select Dependency** será aberta na lateral direita da página e as seguintes informações devem ser inseridas:

- Type: Private
- Name: huaweicloudsdkcce
- Version: 1

Select Dependency

A maximum of 20 dependencies can be added for each function

★ Runtime

Python3.9

★ Type

PublicPrivate

★ Name

huaweicloudsdkcce

★ Version

1

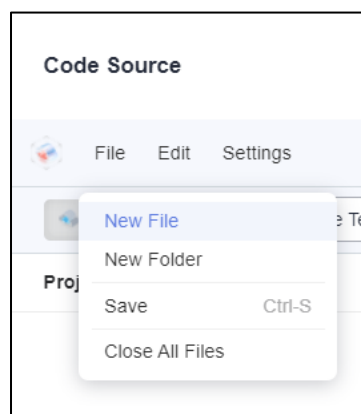
Dependencies (1 / 20 dependencies)							Add
Delete							
<input type="checkbox"/> Name	Type	Version	Runtime	Address	Description	Operation	
<input type="checkbox"/> huaweicloudsdk_smn_core_py3.9	Public	1	Python3.9	https://functionstorage-sa-brazil-1.obs.sa-brazi...	--	Delete	

Mude de aba para **Configuration**, clique em **Environment Variables** e adicione duas variáveis de ambiente **ak** e **sk** que corresponderão a **Access Key** e **Secret Key** de sua conta.

Key	Value	Encrypted	Operation
<input type="checkbox"/> projectid		<input type="checkbox"/>	Delete
<input type="checkbox"/> endpoint	cce-sa-brazil-1.myhuaweicloud.com	<input type="checkbox"/>	Delete
<input type="checkbox"/> region	sa-brazil1	<input type="checkbox"/>	Delete
<input type="checkbox"/> whiteLists		<input type="checkbox"/>	Delete
<input type="checkbox"/> ak		<input type="checkbox"/>	Delete
<input type="checkbox"/> sk		<input type="checkbox"/>	Delete

[Save](#) [Add](#) [Delete](#)

Em **Code Source**, clique em **File** e então em **New File** e renomeie o arquivo para **index.py**. Neste arquivo, deve-se inserir o código disponibilizado a seguir.



```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcore.exceptions import exceptions

from huaweicloudsdkcce.v3 import *
from huaweicloudsdkcce.v3.region.cce_region import CceRegion

import ssl
```

```

ssl._create_default_https_context = ssl._create_unverified_context

def handler(event, context):

    project_id = context.getUserData('projectId', '').strip()
    endpoint = context.getUserData('endpoint', '').strip()
    region = context.getUserData('region', '').strip()
    ak = context.getAccessKey().strip()
    sk = context.getSecretKey().strip()
    instruct_type = context.getUserData('type', '').strip()
    white_list = context.getUserData('whiteLists', '').strip().split(',')

    if not project_id:
        raise Exception("'project_id' not configured")

    if not region:
        raise Exception("'region' not configured")

    if not white_list:
        raise Exception("'whiteLists' not configured")

    if not instruct_type:
        instruct_type = "SOFT"

    if not ak or not sk:
        ak = context.getUserData('ak', '').strip()
        sk = context.getUserData('sk', '').strip()
        if not ak or not sk:
            raise Exception("ak/sk empty")

    logger = context.getLogger()

    credentials = BasicCredentials(ak, sk).with_project_id(project_id)

    if endpoint:
        client = CceClient.new_builder() \
            .with_credentials(credentials) \
            .with_endpoint(endpoint) \
            .build()

    else:
        client = CceClient.new_builder() \
            .with_credentials(credentials) \
            .with_region(CceRegion.value_of(region)) \
            .build()

    try:

        # Request to Hibernate Cluster (the parameter of the request is the
cluster's id)
        request = HibernateClusterRequest(cluster_id='insert_cluster_id_here!!!')
        response = client.hibernate_cluster(request)
        print(response)

    except exceptions.ClientRequestException as e:
        logger.error(e.status_code)
        logger.error(e.request_id)
        logger.error(e.error_code)
        logger.error(e.error_msg)

```

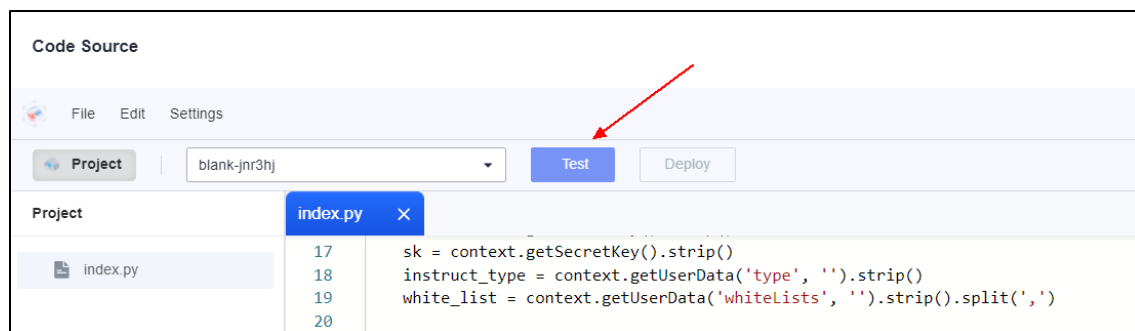
Nesta opção, o código irá fazer com que o cluster hiberne com a execução do código em:

```
# Request to Hibernate Cluster (the parameter of the request is the cluster's id)
request = HibernateClusterRequest(cluster_id='insert_cluster_id_here!!!')
response = client.hibernate_cluster(request)
print(response)
```

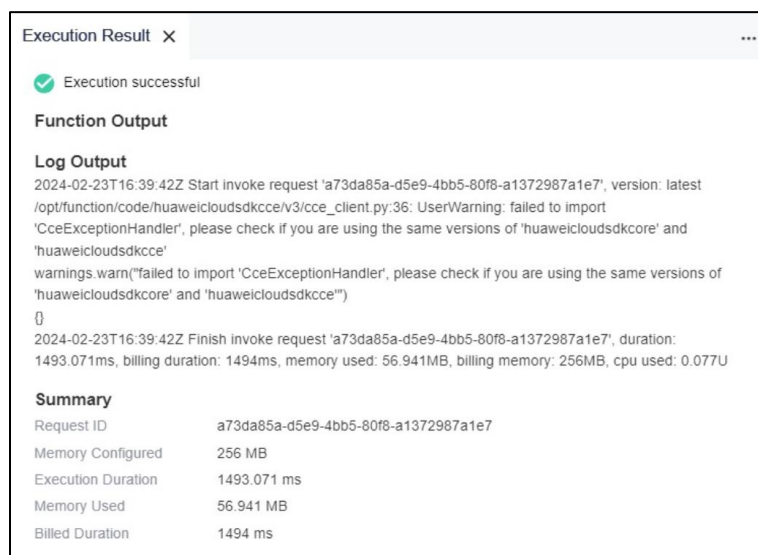
Para que o cluster seja ativado, essa parte do código deve ser alterada para a seguinte:

```
# Request to Awake Cluster (the parameter of the request is the cluster's id)
request = AwakeClusterRequest(cluster_id='insert_cluster_id_here!!!')
response = client.awake_cluster(request)
print(response)
```

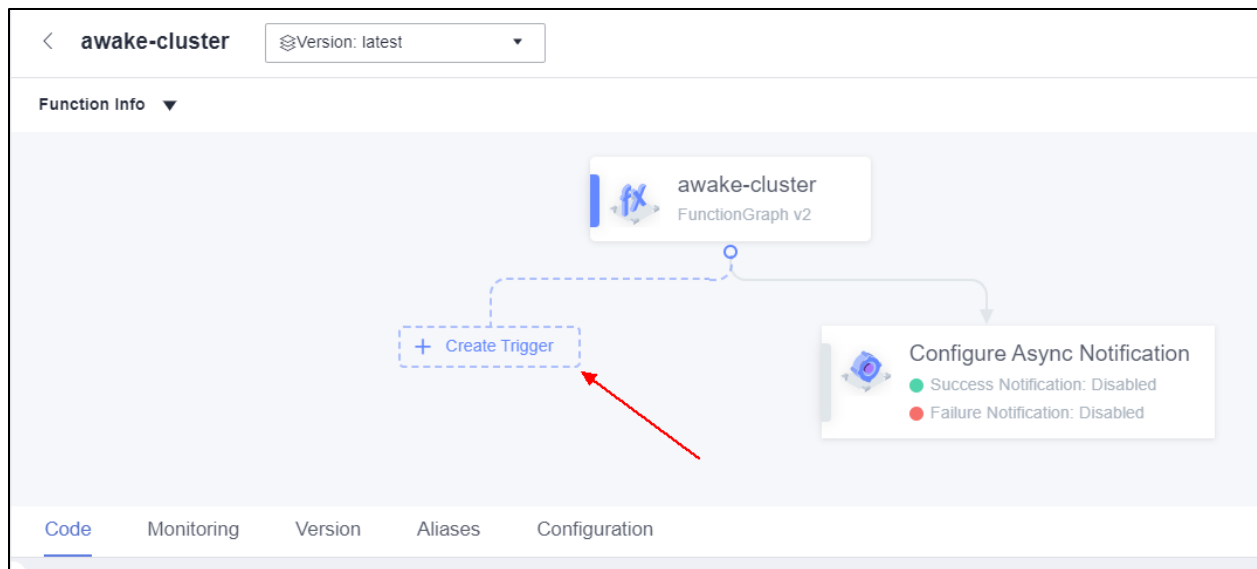
Clique em **Test** para testar a função.



A seguinte mensagem será exibida:



Na topologia da função, clique em **Create Trigger**.




A aba **Create Trigger** será disponibilizada na lateral direita da página. Escolha as seguintes opções:

- Trigger Type: Timer
- Timer Name: Timer-hibernate
- Rule: Cron expression: 0 0 21 * * * (Isso significa que será acionado às 21:00:00 de todos os dias de todos os meses de todos os anos; Outra possível opção é: 40 15 * * * ?, que irá acionar o trigger às 15:40:00 de todos os dias de todos os meses de todos os anos.)
- Enable Trigger: Deixar ligado

Clique em **OK**.

The screenshot shows the 'Create Trigger' configuration page. It has a title 'Create Trigger' and a subtitle 'Trigger Type ?'. The 'Trigger Type' is set to 'Timer'. Below this, a note states: 'The total number of DDS, GAUSSMONGO, DIS, LTS, Kafka and timer triggers cannot exceed 10. You have created 0 such triggers.' The 'Timer Name' is 'Timer-hybermate' (note the typo). Below this, a note states: 'Enter 1 to 64 characters, starting with a letter. Only letters, digits, hyphens (-), and underscores (_) are allowed.' The 'Rule' is set to 'Cron expression' with the value '0 0 21 * * *'. Below this, a link 'Cron Expressions' is visible. The 'Enable Trigger' toggle is turned on. At the bottom, there is an 'Additional Information ?' section with a text area and a character count '0/2,048'.

Timer (Subtotal: 1)



Timer-hibernate

Created: Feb 23, 2024 15:56:37 GMT-03:00

Enabled

Disable

Delete

Rule

Cron expression0 0 21 * * *

Additional Information >

Acessando a página do cluster após as 21 horas, poderemos ver que ele está hibernado.

ClustersBack to Old Clusters

Abnormal Clusters/Total Clusters 0 / 1

Abnormal Nodes/Total Nodes 0 / 0

Recommended Upgrade Clusters 0


All Clusters (1)

CCE Standard Cluster (1) ⓘ

CCE Turbo Cluster (0)

Q Search or filter by keyword.

C




cce-test


● Hibernated


0 / 0

--

--

 View Event

 Wake Up

 Delete Cluster

CCE Standard

v1.27

| Nodes: 50


Available/...

CPU Usage

Memory Usage

Pay-per-use

| Created on Feb 15, 2024 14:34:42 GMT-03:00

 No node in the cluster. Create or accept nodes.

-- End (1/1) --