

FunctionGraph

Perguntas frequentes

Edição 01
Data 2024-04-07



Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2024. Todos os direitos reservados.

Nenhuma parte deste documento pode ser reproduzida ou transmitida em qualquer forma ou por qualquer meio sem consentimento prévio por escrito da Huawei Cloud Computing Technologies Co., Ltd.

Marcas registadas e permissões



HUAWEI e outras marcas registadas da Huawei são marcas registadas da Huawei Technologies Co., Ltd.

Todas as outras marcas registadas e os nomes registados mencionados neste documento são propriedade dos seus respectivos detentores.

Aviso

Os produtos, os serviços e as funcionalidades adquiridos são estipulados pelo contrato estabelecido entre a Huawei Cloud e o cliente. Os produtos, os serviços e as funcionalidades descritos neste documento, no todo ou em parte, podem não estar dentro do âmbito de aquisição ou do âmbito de uso. Salvo especificação em contrário no contrato, todas as declarações, informações e recomendações neste documento são fornecidas "TAL COMO ESTÃO" sem garantias ou representações de qualquer tipo, sejam expressas ou implícitas.

As informações contidas neste documento estão sujeitas a alterações sem aviso prévio. Foram feitos todos os esforços na preparação deste documento para assegurar a exatidão do conteúdo, mas todas as declarações, informações e recomendações contidas neste documento não constituem uma garantia de qualquer tipo, expressa ou implícita.

Índice

1 Perguntas frequentes gerais.....	1
1.1 O que é o FunctionGraph?.....	1
1.2 Preciso solicitar serviços de computação, armazenamento ou rede ao usar o FunctionGraph?.....	1
1.3 Preciso implementar meu código após a programação?.....	1
1.4 Como obter um token?.....	2
1.5 Quais são os tempos de execução compatíveis com o FunctionGraph?.....	3
1.6 Quanto espaço em disco é atribuído a cada função do FunctionGraph?.....	3
1.7 O FunctionGraph oferece suporte ao controle de versão de função?.....	3
1.8 Como uma função lê ou grava arquivos?.....	3
1.9 Como definir um proxy ao usar a CLI?.....	4
1.10 O FunctionGraph oferece suporte à extensão de função?.....	4
1.11 Quais permissões são necessárias para um usuário do IAM usar o FunctionGraph?.....	4
1.12 Como criar um pacote de dependência Python baseado em unidade ODBC para consulta de banco de dados?.....	5
1.13 O que é a cota do FunctionGraph?.....	5
1.14 Quais fontes chinesas são compatíveis com o FunctionGraph?.....	5
1.15 Como o FunctionGraph resolve um nome de domínio do DNS privado?.....	6
1.16 Como uma função baseada em imagem de contêiner resolve um nome de domínio do DNS privado?.....	7
1.17 Como usar um nome de domínio para acessar uma API registrada com o API Gateway (dedicado)?.....	8
1.18 Quais são os cenários comuns de aplicações do FunctionGraph?.....	8
1.19 Por que o nome de domínio do API Gateway vinculado a um serviço não pode ser resolvido durante a invocação de função?.....	8
1.20 O FunctionGraph oferece suporte à transmissão síncrona na largura de banda máxima da intranet?.....	9
1.21 O que acontece se a cota da VPC for usada?.....	9
1.22 Como imprimir logs de Info, Error ou Warn?.....	9
1.23 Posso definir o nome de domínio de uma API para o meu próprio nome de domínio?.....	9
1.24 Posso alterar o tempo de execução?.....	9
1.25 Posso mudar o nome de uma função?.....	9
1.26 Por que a mensagem "failed to mount exist system path" é exibida?.....	9
1.27 Como obter arquivos carregados?.....	10
1.28 Por que não consigo receber respostas para invocação síncrona?.....	10
1.29 O que fazer se os logs de execução de <code>os.system("command &")</code> não forem coletados?.....	10
1.30 Quais diretórios podem ser acessados quando um tempo de execução personalizado é usado?.....	10
1.31 Quais versões secundárias do Python 3.6 e 3.9 são compatíveis?.....	10

1.32	Quais ações podem ser usadas em vez de uma agência de administrador de VPC para acesso à VPC?.....	11
1.33	Quais são as possíveis causas para o tempo limite de função?.....	11
1.34	Como obter o código de uma função?.....	11
1.35	Tem código de amostra para inicializadores?.....	11
1.36	Como ativar a consulta de log estruturada?.....	12
2	Perguntas frequentes sobre criação de função.....	16
2.1	Posso adicionar threads e processos no código de função?.....	16
2.2	Quais são as regras para empacotar um projeto da função?.....	16
2.3	Como o FunctionGraph isola o código?.....	16
2.4	Como criar o arquivo de inicialização para uma função HTTP?.....	16
3	Perguntas frequentes sobre gerenciamento de gatilhos.....	17
3.1	Quais eventos podem acionar uma função do FunctionGraph?.....	17
3.2	E se o código de erro 500 for reportado quando funções que usam gatilhos de APIG retornam cadeias?.....	17
3.3	O que significam LATEST e TRIM_HORIZON na configuração de gatilho do DIS?.....	18
3.4	Por que não consigo ativar ou desativar os gatilhos do OBS chamando APIs?.....	18
3.5	Como usar um gatilho de APIG para chamar uma função?.....	18
3.6	Como uma função obtém o caminho ou os parâmetros da solicitação ao usar um gatilho de APIG?.....	19
3.7	Posso criar um gatilho do OBS com um bucket existente?.....	19
4	Perguntas frequentes sobre gerenciamento de dependências.....	20
4.1	O que é uma dependência?.....	20
4.2	Quando preciso de dependência?.....	20
4.3	Quais são as precauções para usar uma dependência?.....	20
4.4	Quais dependências são compatíveis com o FunctionGraph?.....	20
4.5	O FunctionGraph oferece suporte a bibliotecas de classe?.....	22
4.6	Como usar dependências de terceiros no FunctionGraph?.....	22
4.7	Como criar dependências de função?.....	22
4.8	Como criar uma dependência no console do FunctionGraph?.....	25
4.9	Como adicionar uma dependência a uma função?.....	25
5	Perguntas frequentes sobre execução de funções.....	26
5.1	Quanto tempo leva para executar uma função do FunctionGraph?.....	26
5.2	Quais etapas estão incluídas na execução da função?.....	26
5.3	Como o FunctionGraph processa solicitações simultâneas?.....	26
5.4	O que acontece se as instâncias de função não forem executadas por muito tempo?.....	27
5.5	Como acelerar o acesso inicial a uma função?.....	27
5.6	Como saber a memória atual usada para a execução da função?.....	27
5.7	Por que a minha primeira solicitação é lenta?.....	27
5.8	O que fazer se ocorrer um erro ao chamar uma API?.....	27
5.9	Como ler o cabeçalho da solicitação de uma função?.....	28
5.10	Uma função pode invocar outra função?.....	28
5.11	A interface de execução síncrona pode ser invocada em uma rede privada?.....	30
5.12	Por que uma função usa mais memória do que estimada e até aciona o alarme de falta de memória?.....	30

5.13 Como verificar o uso da memória ao ver "runtime memory limit exceeded"?	30
5.14 Como resolver problema de "CrashLoopBackOff"?	31
5.15 Depois que atualizar uma imagem com o mesmo nome, as instâncias reservadas ainda usam a imagem antiga. O que devo fazer?	31
6 Perguntas frequentes sobre configuração da função	32
6.1 Posso definir variáveis de ambiente ao criar funções?	32
6.2 Posso inserir informações sensíveis em variáveis de ambiente?	32
6.3 Como colocar marca d'água em imagens?	32
6.4 Como usar uma função para chamar uma sub-função?	32
6.5 Como usar as versões e aliases de uma função HTTP com um gatilho de APIG para atualização cinza?	32
7 Perguntas frequentes sobre acesso a recursos externos	36
7.1 Como uma função acessa o banco de dados MySQL?	36
7.2 Como uma função acessa o Redis?	37
7.3 O que fazer se minha função não conseguir se conectar ao Redis por meio de uma VPC?	38
7.4 Como configurar o acesso à rede externa?	38
8 Outras perguntas frequentes	39
8.1 Como visualizar as regras de alarme configuradas para uma função?	39
8.2 O FunctionGraph oferece suporte à descompilação de ZIP durante a transcodificação de vídeo?	39
8.3 Os recursos criados durante o FunctionGraph OBT 2.0 serão liberados automaticamente quando expirarem? Serão cobrados?	39
8.4 O que é uma aplicação no FunctionGraph?	39
8.5 Preciso pagar pelo tempo de inicialização a frio?	39
8.6 Por que uma mensagem é exibida indicando que minha conta foi suspensa ao criar uma função?	40
8.7 As solicitações de todas as minhas funções em diferentes regiões serão cobradas?	40
9 Migração do FunctionGraph V1 para V2	41
9.1 Quais problemas de compatibilidade existem durante a migração?	41
9.2 Por que não posso imprimir logs de uma função do Python 2.7 após a execução de reload(sys)?	41

1 Perguntas frequentes gerais

1.1 O que é o FunctionGraph?

FunctionGraph permite que você execute seu código sem provisionar nem gerenciar servidores, garantindo alta disponibilidade e escalabilidade. Tudo o que você precisa fazer é enviar seu código e definir as condições de execução, e o FunctionGraph cuidará do resto. Você paga apenas pelo que usa e não é cobrado quando o seu código não está em execução.

1.2 Preciso solicitar serviços de computação, armazenamento ou rede ao usar o FunctionGraph?

Ao usar o FunctionGraph não é necessário solicitar ou pré-configurar nenhum serviço de computação, armazenamento ou rede, mas fazer upload e executar código em tempos de execução suportados. O FunctionGraph fornece e gerencia recursos computacionais subjacentes, incluindo CPUs de servidor, memória e redes. Ele executa a configuração e manutenção de recursos, implementação de código, dimensionamento automático, balanceamento de carga, atualização segura e monitoramento de recursos.

1.3 Preciso implementar meu código após a programação?

Após a programação, você só precisa empacotar seu código em um arquivo ZIP (Java, Node.js, Python e Go) ou arquivo JAR (Java) e enviar o arquivo para FunctionGraph para execução.

Ao criar um arquivo ZIP, coloque o arquivo do manipulador sob o diretório **root** para garantir que seu código possa ser executado normalmente após ser descompactado.

Se você editar código em Go, compacte o arquivo compilado e verifique se o nome do arquivo de biblioteca dinâmica é consistente com o nome do plug-in do manipulador. Por exemplo, se o nome do arquivo de biblioteca dinâmica for **testplugin.so**, defina o manipulador como **testplugin.Handler**.

1.4 Como obter um token?

Você pode usar um token para autenticação ao chamar APIs. Para obter um token, use a API padrão do Identity and Access Management (IAM).

- Execute o seguinte comando para obter o token na região **CN South-Guangzhou**:

```
curl -k -i -X POST https://iam.cn-south-1.myhuaweicloud.com/v3/auth/tokens -H 'Content-Type: application/json' -d '{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "name": "HUAWEI CLOUD account",
          "password": "Login password",
          "domain": {
            "name": "HUAWEI CLOUD account"
          }
        }
      }
    },
    "scope": {
      "project": {
        "name": "cn-south-1"
      }
    }
  }
}'
```

- Execute o seguinte comando para obter o token na região **CN North-Beijing1**:

```
curl -k -i -X POST https://iam.cn-north-1.myhuaweicloud.com/v3/auth/tokens -H 'Content-Type: application/json' -d '{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "name": "HUAWEI CLOUD account",
          "password": "Login password",
          "domain": {
            "name": "HUAWEI CLOUD account"
          }
        }
      }
    },
    "scope": {
      "project": {
        "name": "cn-north-1"
      }
    }
  }
}'
```

 **NOTA**

O valor de **X-Subject-Token** no cabeçalho da resposta é um token. Um token obtido em uma região só pode ser usado para chamar FunctionGraph nessa região.

Para obter detalhes, consulte [Obtenção de um token de usuário](#).

1.5 Quais são os tempos de execução compatíveis com o FunctionGraph?

Tabela 1-1 lista os tempos de execução compatíveis pelo FunctionGraph.

Tabela 1-1 Tempos de execução e versões compatíveis

Tempo de execução	Versão
Python	2.7, 3.6, 3.9
Node.js	6.10, 8.10, 10.16, 12.13, 14.18
Java	8 e 11
Go	1.x
C#.NET Core	2,0, 2,1 e 3,1
PHP	7.3

1.6 Quanto espaço em disco é atribuído a cada função do FunctionGraph?

Cada função do FunctionGraph é alocada 512 MB de espaço em disco efêmero. Você pode carregar pacotes de implementação de até 10 GB de tamanho. Para obter mais informações, consulte [Notas e restrições](#).

1.7 O FunctionGraph oferece suporte ao controle de versão de função?

Sim. Para obter detalhes, consulte [Gerenciamento de versões](#).

1.8 Como uma função lê ou grava arquivos?

Conhecimento de fundo

Uma função pode ler arquivos no diretório de código. O diretório de trabalho de uma função é o diretório de nível superior do arquivo do manipulador. Suponha que você carregou uma pasta chamada **backend**. Para ler seu arquivo **test.conf** no mesmo nível de diretório que o arquivo do manipulador, use código de caminho **code/backend/test.conf** ou use um caminho

completo (ou seja, o valor da variável de ambiente **RUNTIME_CODE_ROOT**). Para gravar um arquivo (por exemplo, para criar ou transferir um ficheiro), aceda ao diretório **/tmp** ou use o recurso de montagem do sistema de arquivos fornecido pelo FunctionGraph.

NOTA

- Se os contêineres forem recuperados, o conteúdo lido/gravado do arquivo se tornará inválido.
- Atualmente, o FunctionGraph não oferece suporte à persistência de instância.

Cenários típicos

- Baixe arquivos armazenados no Object Storage Service (OBS) para o diretório **/tmp** para processamento.
- Para armazenar dados de execução de função no OBS, crie um arquivo no diretório **/tmp**, escreva os dados no arquivo e, em seguida, carregue o arquivo no OBS.

1.9 Como definir um proxy ao usar a CLI?

Pergunta

Ao usar a CLI para carregar um pacote de código postal, como faço para definir um servidor de proxy e informações de identidade para concluir a autenticação por meio do gateway de proxy na rede interna?

Resposta

Execute o seguinte comando para definir um proxy:

```
export HTTP_PROXY="http://user:password@proxyIp:proxyPort"
```

Para obter mais informações, consulte https://www.cyberciti.biz/faq/unix-linux-export-variable-http_proxy-with-special-characters/.

1.10 O FunctionGraph oferece suporte à extensão de função?

O FunctionGraph integrou bibliotecas não-padrão como redis, http e obs_client. Você pode usar diretamente essas bibliotecas ao desenvolver funções. Para obter mais informações, consulte [Guia de desenvolvedor](#).

Alternativamente, use suas próprias dependências. Para obter mais informações, consulte [Gerenciamento de dependência](#).

1.11 Quais permissões são necessárias para um usuário do IAM usar o FunctionGraph?

Ao fazer logon no FunctionGraph e adicionar, excluir, modificar e consultar uma função e seus acionadores como um usuário do IAM, conceda permissões ao grupo de usuários ao qual o usuário do IAM pertence, conforme necessário. Por exemplo, para criar um bucket e um gatilho do OBS, conceda a permissão de **Tenant Administrator** para o OBS. Outro exemplo,

para carregar dependências do OBS, conceda as permissões **FunctionGraph FullAccess** e **OBS Administrator** ao usuário do IAM e defina o escopo como serviços globais. Gerencie permissões com base no princípio de menos permissões (PoLP). Para obter detalhes, consulte [Descrição de permissões](#).

1.12 Como criar um pacote de dependência Python baseado em unidade ODBC para consulta de banco de dados?

Para pacotes dependentes do sistema operacional (por exemplo, unixODBC), baixe o código fonte para compilar pacotes de dependência.

1. Efetue logon no seu ECS no console do ECS (garantir que as ferramentas GCC e Make foram instaladas) e execute o seguinte comando para baixar o pacote de código-fonte:

```
wget source code path
```

Se você baixou um arquivo **.zip**, execute o seguinte comando para descompactá-lo:

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

Se você baixou um arquivo **tar.gz**, execute o seguinte comando para descompactá-lo:

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

2. Execute o seguinte comando para criar o diretório **/opt/function/code**:

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

3. Vá para o diretório de destino e execute o seguinte comando:

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

4. Vá para **/opt/function/code/lib/pkgconfig** e verifique se o diretório de prefixos é **/opt/function/code**.

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

5. Copie todos os arquivos em **/opt/function/code/lib** para **/opt/function/code**.

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

6. Mude para **/opt/function/code** e comprima todos os arquivos nele para um pacote **.zip**.

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

1.13 O que é a cota do FunctionGraph?

Para obter detalhes sobre a cota de recursos do FunctionGraph, consulte [Notas e Constraints](#). Para obter detalhes sobre como aumentar a cota, consulte [Como solicitar uma cota mais alta?](#)

1.14 Quais fontes chinesas são compatíveis com o FunctionGraph?

O FunctionGraph é compatível com as seguintes fontes chinesas:

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

- NotoSerifSC-Regular.otf

1.15 Como o FunctionGraph resolve um nome de domínio do DNS privado?

O FunctionGraph não pode analisar diretamente nomes de domínio do DNS privados da Huawei Cloud. Para analisá-los, chame APIs de DNS e execute as etapas a seguir.

Resolver um nome de domínio do DNS privado

Verifique se uma VPC e um nome de domínio do DNS privado foram criados antes de executar as seguintes etapas:

Passo 1 Associe uma VPC ao nome de domínio privado e adicione conjuntos de registros.

Faça logon no console do DNS e associe uma VPC ao nome de domínio privado.

Clique no nome de domínio e adicione um conjunto de registros do tipo A.

Passo 2 Crie uma função.

Crie uma função cujo tempo de execução é o Python 2.7. O seguinte é o código de exemplo.

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

Passo 3 Configure uma agência com permissões de DNS e VPC para a função.

No console do IAM, crie uma agência com permissões de **DNS ReadOnlyAccess** e **VPC Administrator** para o FunctionGraph.

NOTA

Você precisa configurar a permissão para ler dados de recursos do DNS porque a função precisa obter esses dados ao analisar um nome de domínio. Caso contrário, a seguinte mensagem de erro é exibida, indicando que os dados de recursos do DNS não foram obtidos.

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

Passo 4 Configure a função.

Na página de detalhes da função criada em [Passo 2](#), clique na guia **Configuration** e defina as seguintes configurações:

1. Para **Permissions**, selecione a agência criada em [Passo 3](#).
2. Ative o acesso à VPC e selecione a VPC, a sub-rede e o nome de domínio criados. Para mais detalhes, consulte [Figura 1-1](#).

Figura 1-1 Configuração da função

VPC Access ?

Status

VPC [Create VPC](#)

Subnet [Create Subnet](#)

Domain Name ? [config.vpc.dns.aEle](#)

[Save](#)

Passo 5 Verifique o resultado da execução.

Em seguida, todos os nomes de domínio IPv4 configurados podem ser analisados.

----Fim

 **NOTA**

As alterações nos endereços IP correspondentes aos nomes de domínio da VPC que você configurar entrarão em vigor em 10 minutos.

1.16 Como uma função baseada em imagem de contêiner resolve um nome de domínio do DNS privado?

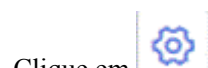
As funções do FunctionGraph criadas com uma imagem de contêiner não podem analisar diretamente nomes de domínio do DNS privados. No entanto, você pode chamar APIs de DNS para atingir esse objetivo.


Resolver um nome de domínio do DNS privado

1. Obtenha um nome de domínio privado e ID de zona.

Esse procedimento usa um nome de domínio com um conjunto de registros como exemplo.

- a. Efetue logon no console do DNS.
- b. Obtenha um ID de zona.



Clique em  e selecione **Domain Name** na caixa de pesquisa para obter um ID de zona.

- c. Obtenha o nome de domínio privado correspondente a um conjunto de gravação.

Clique no nome de domínio para ir para a lista de conjuntos de registros e selecione um conjunto de registros.

2. Compile a lógica de resolução.

Depurar a API usada para [consultar conjuntos de registros em uma zona](#).

- Defina **zone_id** como o ID de zona obtido na etapa anterior e clique em **Debug**. O endereço IP do nome de domínio privado é exibido no corpo da resposta.
- Alterne para a guia **Sample Code** para obter o código completo. Para obter detalhes sobre as dependências, clique em **View SDK Details**.

1.17 Como usar um nome de domínio para acessar uma API registrada com o API Gateway (dedicado)?

O nome de domínio **www.test.com** é usado como exemplo. O procedimento é o seguinte:

Passo 1 Faça logon no console do API Gateway, escolha **Dedicated Gateways** no painel de navegação e clique no nome do gateway de destino. Na página **Gateway Information**, visualize o **EIP** na área **Inbound Access** para obter o endereço IP do gateway da API.

Passo 2 No console do DNS, configure uma regra IPv4 para mapear **www.test.com** para um endereço de gateway de API.

Passo 3 Configure a resolução de nome de domínio referindo-se a [Como o FunctionGraph resolve um nome de domínio do DNS privado?](#). Dessa forma, você pode acessar a API registrada no gateway da API usando o nome de domínio **www.test.com**.

----Fim

1.18 Quais são os cenários comuns de aplicações do FunctionGraph?

1. Aplicações Web: miniprogramas, páginas Web/apps, chatbots e Backends for Frontends (BFF).
2. Aplicações orientadas a eventos: processamento de arquivos, processamento de imagens, transmissão/transcodificação de vídeo ao vivo, processamento de fluxo de dados em tempo real e processamento de regras/eventos da IoT.
3. Aplicações de IA: integração de serviços de terceiros, inferência de IA e reconhecimento de placas de veículos.

Para obter detalhes, consulte [Cenários de aplicação](#).

1.19 Por que o nome de domínio do API Gateway vinculado a um serviço não pode ser resolvido durante a invocação de função?

Atualmente, o FunctionGraph resolve apenas nomes de domínio do DNS da Huawei Cloud e nomes de domínio POD.

1.20 O FunctionGraph oferece suporte à transmissão síncrona na largura de banda máxima da intranet?

No momento, não.

1.21 O que acontece se a cota da VPC for usada?

Um locatário pode criar até 4 VPCs. Para criar mais VPCs, [envie um tíquete de serviço](#).

1.22 Como imprimir logs de Info, Error ou Warn?

Tomemos o Java como exemplo. Você pode usar [esta demonstração](#) para imprimir logs.

1.23 Posso definir o nome de domínio de uma API para o meu próprio nome de domínio?

Sim. O procedimento é o seguinte:

Passo 1 Faça logon no console do APIG e vincule um nome de domínio consultando [Vinculação de um nome de domínio](#).

Passo 2 Na página de guia **Domain Names** do grupo de APIs criado, clique em **Bind Independent Domain Name**. Por exemplo, defina `xxxx.apig.x` como `test.com/user/get`.

----Fim

1.24 Posso alterar o tempo de execução?

Não. Uma vez que uma função é criada, seu tempo de execução não pode ser alterado.

1.25 Posso mudar o nome de uma função?

Não. O nome de uma função não pode ser alterado uma vez que a função seja criada.

1.26 Por que a mensagem "failed to mount exist system path" é exibida?

Quando você vir essa mensagem, monte o arquivo em um novo caminho.

User ID/user group ID: pode ser qualquer número, exceto 1000. O valor `-1` será automaticamente convertido para `1003`. Os dois IDs controlam as permissões de diretório para acessar um sistema de arquivos remoto.

File system/ECS name: nome do sistema de arquivos ou ECS a ser criado. Verifique se você especificou uma VPC e uma agência que você foi autorizado a acessar.

Shared directory: para configurar um diretório compartilhado remoto para o ECS montado, consulte [Criação de um diretório compartilhado do NFS em um ECS](#).

Access path: local onde o sistema de arquivos deve ser montado na função. Defina um novo diretório de dois níveis que comece com `/mnt`. Por exemplo, `/mnt/test`.

1.27 Como obter arquivos carregados?

Tomemos o Python como exemplo. Se você usar `os.getcwd()` para consultar o diretório atual, o diretório será `/opt/function`. No entanto, o código foi realmente carregado para `/opt/function/code`.

Você pode usar um dos seguintes métodos para obter arquivos carregados:

1. Execute o comando `cd` para mudar para `/opt/function/code`.
2. Acesse o caminho completo (valor da variável de ambiente `RUNTIME_CODE_ROOT`).

NOTA

Você pode obter arquivos carregados consultando os métodos anteriores quando outras linguagens são usadas.

1.28 Por que não consigo receber respostas para invocação síncrona?

Se a latência de execução da função E2E exceder 90s, a invocação assíncrona é recomendada. Se a invocação síncrona for usada, nenhuma resposta poderá ser recebida após os 90s devido a restrições de gateway.

1.29 O que fazer se os logs de execução de `os.system("command &")` não forem coletados?

Não use `os.system("command &")`. A saída do comando em segundo plano não será coletada. Para obter a saída do comando, use `subprocess.Popen` em vez disso.

1.30 Quais diretórios podem ser acessados quando um tempo de execução personalizado é usado?

Por padrão, somente o diretório `/tmp` pode ser acessado, por exemplo, para criar ou baixar arquivos.

1.31 Quais versões secundárias do Python 3.6 e 3.9 são compatíveis?

3.6.8 e 3.9.2.

1.32 Quais ações podem ser usadas em vez de uma agência de administrador de VPC para acesso à VPC?

As ações listadas em [Tabela 1-2](#) podem ser usadas.

Tabela 1-2 Ações

Permissão	Ação
Exclusão de uma porta	vpc:ports:delete
Consulta de uma porta	vpc:ports:get
Criação de uma porta	vpc:ports:create
Consulta de uma VPC	vpc:vpcs:get
Consulta de uma sub-rede	vpc:subnets:get

1.33 Quais são as possíveis causas para o tempo limite de função?

- A lógica do código expirou o tempo limite. Nesse caso, otimize o código ou aumente o tempo limite.
- A rede expirou o tempo limite. Para corrigir esse problema, aumente o tempo limite.
- Demorou muito tempo para carregar as classes de Java durante a inicialização a frio. Nesse caso, aumente o tempo limite ou a memória.

1.34 Como obter o código de uma função?

1. Faça login no console do FunctionGraph e clique no nome da função de destino para acessar a página de detalhes. Escolha **Operation** > **Export function** no canto superior direito e clique em **Export Code**.
2. Como alternativa, chame a API de exportação de função.

1.35 Tem código de amostra para inicializadores?

Sim. Veja os seguintes exemplos:

- Node.js ([Introdução ao inicializador](#))


```
exports.initializer = function(context, callback) {  
    callback(null, '');  
};
```

- Python ([Introdução ao inicializador](#))

```
def my_initializer(context):  
    print("hello world!")
```

- Java ([Introdução ao inicializador](#))

```
public void my_initializer(Context context)  
{  
    RuntimeLogger log = context.getLogger();  
    log.log(String.format("ak:%s", context.getAccessKey()));  
}
```

- PHP ([Introdução ao inicializador](#))

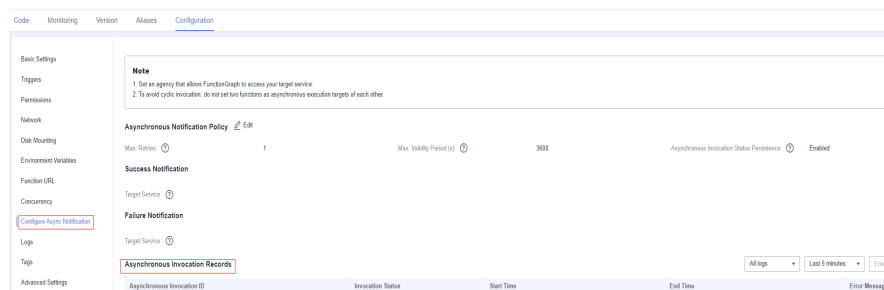
```
<?php  
Function my_initializer($context) {  
    echo 'hello world' . PHP_EOL;  
}  
?>
```

1.36 Como ativar a consulta de log estruturada?

Cenário

Para verificar o status das solicitações de chamada assíncronas, visualize os registros escolhendo **Configuration > Configure Async Notification** na página de detalhes da função, conforme mostrado na [Figura 1-2](#).

Figura 1-2 Registros de invocação assíncrona



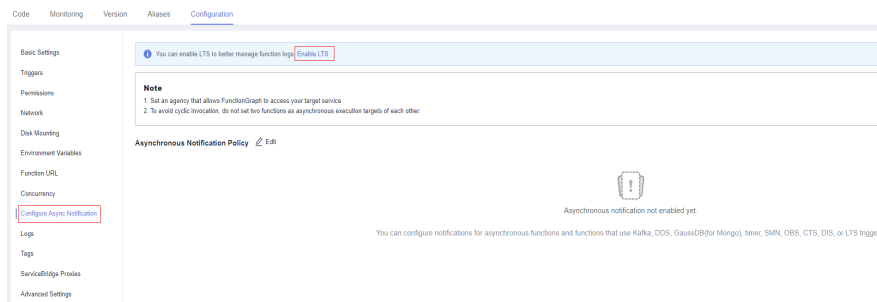
Pré-requisitos

Você ativou a persistência do status de invocação assíncrona.

Procedimento

- Passo 1** Entre em contato com o atendimento ao cliente para adicionar sua conta à lista branca desse recurso.
- Passo 2** Na página **Configure Async Notification**, clique em **Enable LTS**, como mostrado na [Figura 1-3](#).

Figura 1-3 Ativar o LTS



Passo 3 Clique em **Edit** ao lado de **Asynchronous Notification Policy** e ative **Asynchronous Invocation Status Persistence**, conforme mostrado na **Figura 1-4** e **Figura 1-5**.

Figura 1-4 Configurar a política de notificação assíncrona

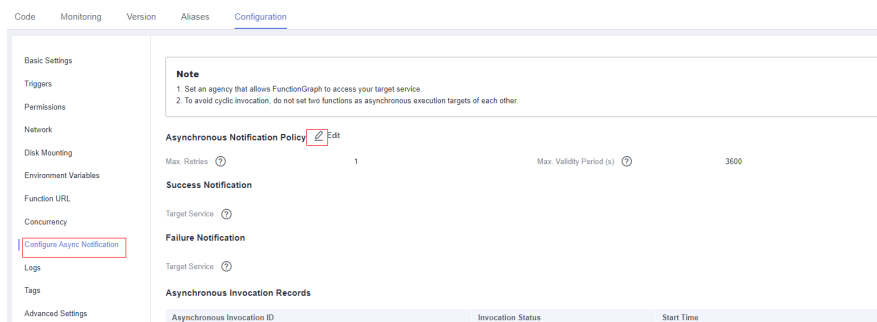
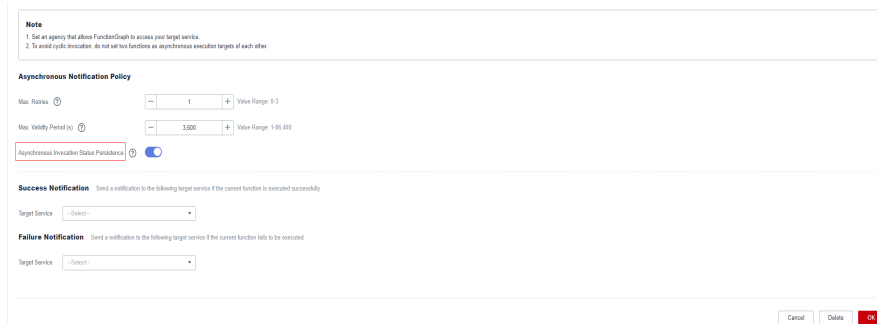


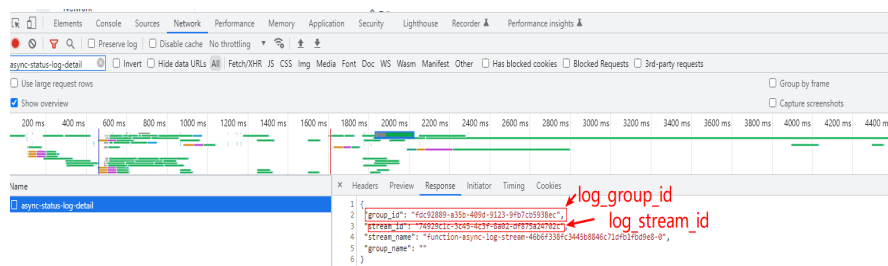
Figura 1-5 Ativar a persistência do status de invocação assíncrona



Passo 4 Configure a consulta estruturada no console do LTS.

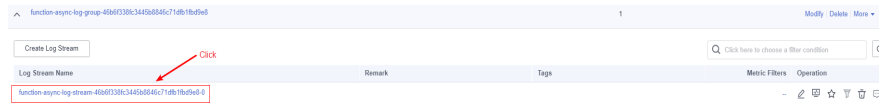
1. Na página de detalhes da função, visualize o grupo de logs e o fluxo de logs. Pressione **F12**, escolha **Network**, incorpore o filtro **async-status-log-detail**, e obtenha o ID do grupo de log e o ID do fluxo de log, como mostrado em **Figura 1-6**.

Figura 1-6 Obter o ID do grupo de logs e o ID do fluxo de logs



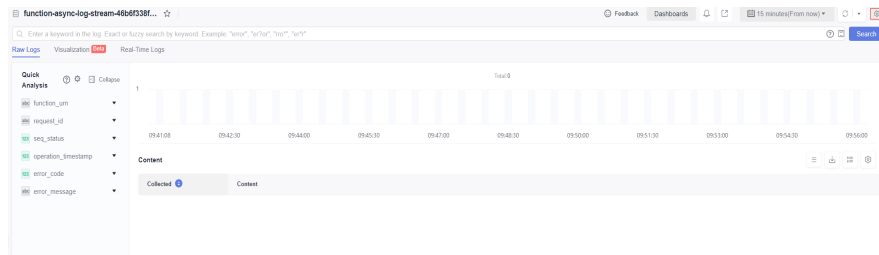
- No console do LTS, localize o grupo de logs e o fluxo de logs por seus IDs, conforme mostrado na [Figura 1-7](#).

Figura 1-7 Visualizar fluxo de log



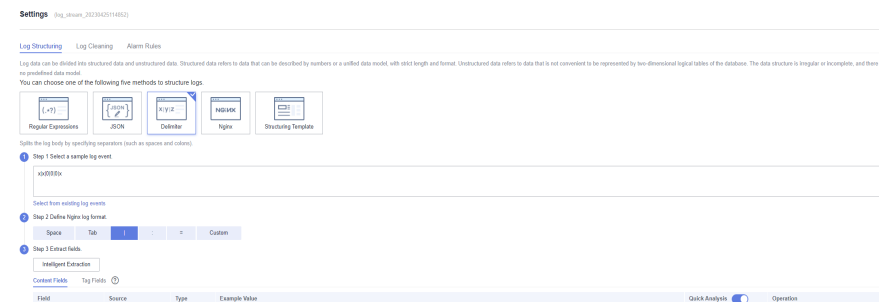
- Na página de detalhes do fluxo de log, clique no ícone de engrenagem no canto superior direito, como mostrado na [Figura 1-8](#).

Figura 1-8 Clicar no ícone de engrenagem



- Configure a estruturação de log, conforme mostrado na [Figura 1-9](#).


Figura 1-9 Configurar a estruturação de log



- Clique em **Intelligent Extraction**, como mostrado na [Figura 1-10](#).

Figura 1-10 Extração inteligente



- Clique em  para modificar a definição de campo da seguinte forma:
 - Altere **field1** para **function_urn** e seu tipo para **string**.
 - Altere **field2** para **request_id** e seu tipo para **string**.
 - Altere **field3** para **seq_status** e seu tipo para **long**.
 - Altere **field4** para **operation_timestamp** e seu tipo para **long**.

- e. Altere **field5** para **error_code** e seu tipo para **long**.
- f. Altere **field6** para **error_message** e seu tipo para **string**.

Ative **Quick Analysis**, conforme mostrado na **Figura 1-11**.

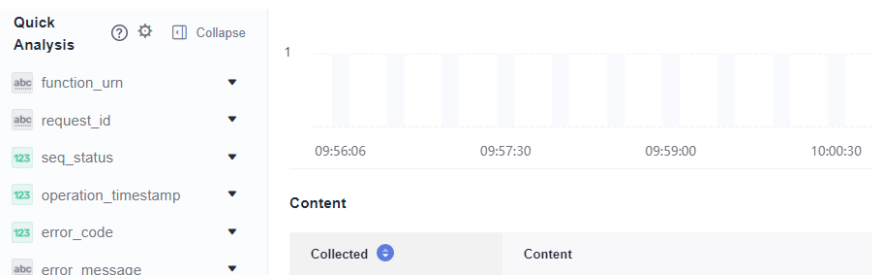
Figura 1-11 Ativar a análise rápida



Field	Source	Type	Example Value	Quick Analysis
field1	Content Fields	string	x	<input checked="" type="checkbox"/>
field2	Content Fields	string	x	<input checked="" type="checkbox"/>
field3	Content Fields	long	0	<input checked="" type="checkbox"/>
field4	Content Fields	long	0	<input checked="" type="checkbox"/>
field5	Content Fields	long	0	<input checked="" type="checkbox"/>
field6	Content Fields	string	x	<input checked="" type="checkbox"/>

- 7. Clique em **Save**. **Figura 1-12** mostra a configuração.

Figura 1-12 Configuração salva



----Fim

2 Perguntas frequentes sobre criação de função

2.1 Posso adicionar threads e processos no código de função?

Você pode criar threads e processos adicionais em sua função usando os recursos do tempo de execução e do SO.

2.2 Quais são as regras para empacotar um projeto da função?

Além da edição de código embutido, é possível criar uma função carregando um arquivo ZIP ou JAR ou carregando um arquivo ZIP do OBS. Para obter mais detalhes.

2.3 Como o FunctionGraph isola o código?

Cada função do FunctionGraph é executada em seu próprio ambiente e tem seus próprios recursos e sistema de arquivos.

2.4 Como criar o arquivo de inicialização para uma função HTTP?

Para criar uma função HTTP, crie um arquivo de inicialização. Para detalhes, consulte [Criação de um arquivo de inicialização](#).

3 Perguntas frequentes sobre gerenciamento de gatilhos

3.1 Quais eventos podem acionar uma função do FunctionGraph?

Para obter detalhes, consulte [Fontes de eventos suportadas](#).

3.2 E se o código de erro 500 for reportado quando funções que usam gatilhos de APIG retornam cadeias?

Verifique se a resposta da função para uma invocação pelo API Gateway foi encapsulada e contém **body(String)**, **statusCode(int)**, **headers(Map)** e **isBase64Encoded(boolean)**.

Veja a seguir um exemplo de resposta retornada por uma função de Node.js que usa um gatilho de APIG:

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

Veja a seguir um exemplo de resposta retornada por uma função de Java que usa um gatilho de APIG:

```
import java.util.Map;

public HttpTriggerResponse index(String event, Context context){
  String body = "<html><title>FunctionStage</title>"
    + "<h1>This is a simple APIG trigger test</h1><br>"
    + "<h2>This is a simple APIG trigger test</h2><br>"
}
```

```
        + "<h3>This is a simple APIG trigger test</h3>"
        + "</html>";
    int code = 200;
    boolean isBase64 = false;
    Map<String, String> headers = new HashMap<String, String>();
    headers.put("Content-Type", "text/html; charset=utf-8");
    return new HttpTriggerResponse(body, headers, code, isBase64);
}

class HttpTriggerResponse {
    private String body;
    private Map<String, String> headers;
    private int statusCode;
    private boolean isBase64Encoded;
    public HttpTriggerResponse(String body, Map<String,String> headers,
int statusCode, boolean isBase64Encoded){
        this.body = body;
        this.headers = headers;
        this.statusCode = statusCode;
        this.isBase64Encoded = isBase64Encoded;
    }
}
```

3.3 O que significam LATEST e TRIM_HORIZON na configuração de gatilho do DIS?

Os cursores **LATEST** e **TRIM_HORIZON** especificam os pontos iniciais para a leitura de dados nos fluxos do Data Ingestion Service (DIS).

- **TRIM_HORIZON**: os dados são lidos a partir do registro válido mais antigo armazenado na partição.
Por exemplo, um locatário usou um fluxo do DIS para carregar três partes de dados A1, A2 e A3. Supondo que A1 expire, mas A2 e A3 ainda sejam válidos após um período de tempo, se o locatário especificar **TRIM_HORIZON** para download de dados, apenas A2 e A3 poderão ser baixados.
- **LATEST**: os dados são lidos a partir do último registro na partição. Essa opção garante que os dados mais recentes na partição sejam lidos.

Para obter detalhes, consulte [Obtenção de um cursor](#).

3.4 Por que não consigo ativar ou desativar os gatilhos do OBS chamando APIs?

O OBS não oferece suporte a gatilhos de recepção. Portanto, os gatilhos do OBS não podem ser ativados ou desativados.

3.5 Como usar um gatilho de APIG para chamar uma função?

Para obter detalhes, consulte [Uso de gatilho de APIG \(dedicado\)](#).

3.6 Como uma função obtém o caminho ou os parâmetros da solicitação ao usar um gatilho de APIG?

Por padrão, o caminho ou os parâmetros da solicitação são incluídos em **event**. Uma função invoca APIG usando seu modelo de evento. Você pode obter o caminho ou os parâmetros da solicitação a partir do resultado da execução da função.

Exemplo:

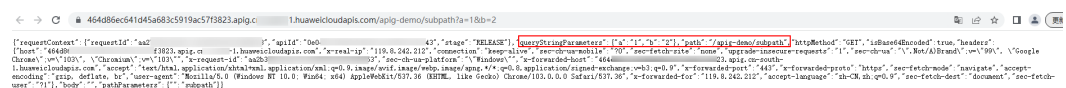
```
{ requestContext:
  { requestId: [REDACTED],
    apiId: [REDACTED],
    stage: 'RELEASE' },
  queryStringParameters: { a: '1', b: '2' },
  path: '/apig-demo/subpath',
  httpMethod: 'GET',
  isBase64Encoded: true,
```

- **queryStringParameters**: parâmetros que seguem o ponto de interrogação (?), separados por E comercial (&). Esses parâmetros são adicionados na URL de uma solicitação GET e serão transferidos no formato de cadeia URL quando uma solicitação GET for iniciada.
- **path**: URL da API.

Você pode chamar uma API usando seu caminho de solicitação. Exemplo: **https://464d86ec641d45a683c5919ac57f3823.apig.projectID.huaweicloudapis.com/apig-demo/subpath**

Como alternativa, você pode chamar uma API adicionando parâmetros de solicitação. Exemplo:

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



3.7 Posso criar um gatilho do OBS com um bucket existente?

Sim. Se uma mensagem for exibida indicando que a configuração do gatilho atual entra em conflito com a de outro, os dois gatilhos terão o mesmo bucket, prefixo e sufixo. Se você ainda quiser usar esse bucket para o gatilho atual, modifique o prefixo ou sufixo.

4 Perguntas frequentes sobre gerenciamento de dependências

4.1 O que é uma dependência?

Uma dependência é um pacote de programa e também um ambiente necessário para executar um pacote de software. O pacote de software depende e só pode ser executado no ambiente.

4.2 Quando preciso de dependência?

Quando você instala um programa ou desenvolve código que depende de um ambiente para ser executado, é necessário introduzir a dependência.

4.3 Quais são as precauções para usar uma dependência?

- O nome de cada arquivo em uma dependência não pode terminar com um til (~).
- Não deve haver mais do que 30.000 arquivos em uma dependência.
- Você pode carregar um arquivo de dependência ZIP dentro de 10 MB na página de detalhes da função. Para uma dependência maior (máx. 300 MB), faça o upload usando o OBS.
- Se sua função usar uma grande dependência privada, aumente o tempo limite escolhendo **Configuration > Basic Settings** na página de detalhes da função.

4.4 Quais dependências são compatíveis com o FunctionGraph?

Dependências compatíveis

O FunctionGraph oferece suporte a bibliotecas padrão e dependências de terceiros.

- Bibliotecas padrão

Ao usar bibliotecas padrão, você pode importá-las para o código embutido ou empacotá-las e enviá-las para FunctionGraph.

- Bibliotecas não padrão compatíveis
O FunctionGraph fornece componentes internos de terceiros, conforme descrito em [Tabela 4-1](#) e [Tabela 4-2](#). Você pode importar esses componentes para o código embutido da mesma forma que importa bibliotecas padrão.

Tabela 4-1 Componentes de terceiros integrados com o tempo de execução do Node.js

Nome	Descrição	Versão
q	Encapsulamento de método assíncrono	1.5.1
co	Controle assíncrono do processo	4.6.0
lodash	Biblioteca de ferramentas e métodos comuns	4.17.10
esdk-obs-nodejs	SDK do OBS	2.1.5
express	Estrutura simplificada de desenvolvimento de aplicações baseado na Web	4.16.4
fgs-express	Fornece uma estrutura de aplicações de Node.js para FunctionGraph e APIG para executar aplicações sem servidor e APIs REST. Este componente fornece um exemplo do uso da estrutura Express para criar aplicações ou serviços da Web sem servidor e APIs RESTful.	1.0.1
request	Simplifica a invocação HTTP e oferece suporte a HTTPS e redirecionamento.	2.88.0

Tabela 4-2 Bibliotecas não-padrão compatíveis pelo tempo de execução do Python

Módulo	Descrição	Versão
dateutil	Processamento de data e hora	2.6.0
requests	Biblioteca de HTTP	2.7.0
httplib2	httpclient	0.10.3
numpy	Computação matemática	1.13.1
redis	Cliente de Redis	2.10.5

Módulo	Descrição	Versão
obscliente	Cliente do OBS	-
smnsdk	Acesso da SMN (nuvem pública)	1.0.1

- Outras bibliotecas de terceiros

Para outras bibliotecas de terceiros não listadas nas tabelas anteriores, empacote e carregue-as em um bucket do OBS ou na página de detalhes da função. Para obter detalhes, consulte [Como criar uma dependência no console do FunctionGraph?](#) Essas bibliotecas serão usadas em seu código de função.

4.5 O FunctionGraph oferece suporte a bibliotecas de classe?

Sim. O FunctionGraph oferece suporte a bibliotecas padrão e bibliotecas de terceiros não padrão. Para mais detalhes, consulte [Quais dependências são compatíveis com o FunctionGraph?](#)

4.6 Como usar dependências de terceiros no FunctionGraph?

1. Empacote bibliotecas de terceiros em um pacote ZIP, referindo-se a [Como criar dependências de função?](#)
2. Crie uma dependência no console do FunctionGraph referindo-se a [Como criar uma dependência no console do FunctionGraph?](#)
3. Na página de detalhes da função, clique na guia **Code** e adicione a dependência consultando [Como adicionar uma dependência a uma função?](#) Em seguida, você pode usar a dependência no código da função.

4.7 Como criar dependências de função?

É aconselhável criar dependências de função no EulerOS. Se outros sistemas operacionais forem usados, um erro pode ocorrer devido a bibliotecas dependentes subjacentes. Por exemplo, a biblioteca de vínculo dinâmico não pode ser encontrada.

NOTA

Se os módulos a serem instalados precisarem de dependências como .dll, .so e .a, archive-os em um pacote .zip.

Configurar o ambiente do EulerOS

EulerOS é um sistema operacional Linux de código aberto para empresas. Ele apresenta alta segurança, escalabilidade e desempenho, atendendo aos requisitos dos clientes para infraestrutura de TI e serviços de computação em nuvem. [EulerOS da Huawei Cloud](#) é recomendado.

1. Compre um EulerOS do ECS na Huawei Cloud, referindo-se a [Compra e logon em um ECS do Linux](#). Na página **Configure Basic Settings**, selecione **Public Image** e selecione **Huawei Cloud EulerOS** e uma versão de imagem.
2. Baixe a imagem do EulerOS e use o software de virtualização para configurar a VM do EulerOS em um PC local.

Criar uma dependência para uma função de Python

Certifique-se de que a versão do Python do ambiente de empacotamento seja a mesma da função. Para o Python 2.7, recomenda-se o Python 2.7.12 ou posterior. Para o Python 3.6, recomenda-se o Python 3.6.3 ou posterior.

Para instalar a dependência do PyMySQL para uma função do Python 2.7 no diretório **/tmp/pymysql**, execute o seguinte comando:

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

Depois que o comando for executado com sucesso, vá para o diretório **/tmp/pymysql**:

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

Vá para o diretório **site-packages** (geralmente, **usr/lib64/python2.7/site-packages/**) e execute o seguinte comando:

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

A dependência necessária é gerada.

NOTA

Para instalar o pacote de instalação local de rodas, execute o seguinte comando:

```
pip install piexif-1.1.0b0-py2.py3-none-any.whl --root /tmp/piexif  
//Replace piexif-1.1.0b0-py2.py3-none-any.whl with the actual installation  
package name.
```

Criando uma dependência para uma função de Node.js

Verifique se a versão correspondente do Node.js foi instalada no ambiente.

Para instalar a dependência de MySQL para uma função de Node.js 8.10, execute o seguinte comando:

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

A pasta **node_modules** é gerada sob o diretório atual.

- Sistema operacional Linux

Execute o seguinte comando para gerar um pacote ZIP.

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

A dependência necessária é gerada.

- Sistema operacional Windows

Compacte **node_modules** em um arquivo ZIP.

Para instalar múltiplas dependências, crie um arquivo **package.json** primeiro. Por exemplo, insira o seguinte conteúdo no arquivo **package.json** e execute o seguinte comando:

```
{  
  "name": "test",
```

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

NOTA

Não execute o comando **CNPM** para gerar dependências de Node.js.

Compacte **node_modules** em um pacote ZIP. Isso gera uma dependência que contém tanto o MySQL quanto o Redis.

Para outras versões do Node.js, você pode criar dependências da maneira indicada acima.

Criar uma dependência para uma função de Java

Quando você compila uma função usando Java, as dependências precisam ser compiladas localmente. Para obter detalhes sobre como adicionar dependências, consulte [Desenvolvimento de funções em Java \(usando um projeto IDEA de Java\)](#).

Criar uma dependência para uma função de PHP

EulerOS 2.9.6 é recomendado.

Por padrão, o Composer e o PHP 7.3 foram instalados no ambiente. Instale o Protobuf 3.19 usando o Composer.

Crie o arquivo **composer.json** com o seguinte conteúdo:

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

Execute o seguinte comando:

```
Composer install
```

A pasta **vendor** é gerada com as subpastas **autoload.php**, **composer** e **google** no diretório atual.

- Linux

Execute o seguinte comando para gerar um pacote ZIP.

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

- Windows

Compacte **vendor** em um arquivo ZIP.

Se várias dependências precisarem ser instaladas, especifique-as no arquivo **composer.json**, compacte a pasta do **vendor** em um arquivo ZIP e faça o upload.

NOTA

Para usar dependências de terceiros baixadas usando o Composer no código do projeto de PHP, carregue as dependências através de **require "./vendor/autoload.php"**. Por padrão, os arquivos descompactados do pacote ZIP carregado são colocados em um diretório no mesmo nível do código do projeto.

4.8 Como criar uma dependência no console do FunctionGraph?

1. Entre ao console do FunctionGraph, e escolha **Functions > Dependencies** no painel de navegação.
2. Clique em **Create Dependency**.
3. Defina os seguintes parâmetros.

Tabela 4-3 Parâmetros de configuração de dependência

Parâmetro	Descrição
Name	Nome da dependência.
Code Entry Mode	Carregue um arquivo ZIP diretamente ou através do OBS. <ul style="list-style-type: none">● Upload ZIP file: clique em Select File para carregar um arquivo ZIP.● Upload from OBS: especifique um URL de link do OBS. Para obter detalhes sobre como obter o URL, consulte Acesso de um objeto usando seu URL.
Runtime	Selecione um tempo de execução.
Description	Descrição da dependência. Este parâmetro é opcional.

4. Clique em **OK**. Por padrão, uma nova dependência é a versão **1**.
5. Clique no nome da dependência e exiba todas as versões e informações relacionadas na página exibida. Cada dependência pode ter várias versões.
 - Para criar uma versão de dependência, clique em **Create Version** no canto superior direito da página.
 - Para exibir o endereço de uma versão, clique na versão.
 - Para excluir uma versão, clique no ícone de exclusão na mesma linha.



4.9 Como adicionar uma dependência a uma função?

1. Na página de detalhes da função, clique na guia **Code** e clique em **Add** na área **Dependencies**.
 - **Public**: dependências públicas são fornecidas pelo FunctionGraph e podem ser adicionadas diretamente.
 - **Private**: dependências privadas são aquelas que você criou e enviou.
2. Clique em **OK**.

5 Perguntas frequentes sobre execução de funções

5.1 Quanto tempo leva para executar uma função do FunctionGraph?

Dentro de 900s para execução síncrona e 72 horas para execução assíncrona.

O tempo limite de execução padrão é 3s. Você pode definir o tempo limite (unidade: s) para um número inteiro de 3 a 259.200. Se você definir o tempo limite de uma função para 3s, ela será encerrada após 3s.

5.2 Quais etapas estão incluídas na execução da função?

A execução da função inclui duas etapas:

1. Selecione uma instância ociosa com a memória necessária.
2. Execute o código especificado.

5.3 Como o FunctionGraph processa solicitações simultâneas?

O FunctionGraph aumenta ou diminui automaticamente instâncias de função com base no número de solicitações. Se o número de solicitações simultâneas aumentar, o FunctionGraph alocará mais instâncias de função para processar as solicitações. Se esse número diminuir, o FunctionGraph alocará menos instâncias de função em conformidade.

Número de instâncias de função = simultaneidade de função/simultaneidade por instância

- Simultaneidade de função: o número de solicitações executadas simultaneamente por uma função em um determinado ponto de tempo.
- Simultaneidade por instância: o número máximo de solicitações simultâneas permitidas por uma única instância. Isso é equivalente ao parâmetro **Max. Requests per Instance** na página **Concurrency**.

 **NOTA**

Para obter detalhes sobre serviços relacionados que podem estar envolvidos em solicitações simultâneas, consulte [Fontes de eventos suportadas](#).

5.4 O que acontece se as instâncias de função não forem executadas por muito tempo?

Se uma função não foi executada por um período de tempo, todas as instâncias relacionadas à função serão liberadas.

5.5 Como acelerar o acesso inicial a uma função?

C# e Go suportam uma velocidade de inicialização menor do que outras linguagens devido a problemas de mecanismo. Você pode usar os seguintes métodos para acelerar o acesso inicial a uma função:

- Aloque mais memória para a função.
- Simplifique o código da função, por exemplo, exclua pacotes de dependências desnecessárias.
- Ao usar C# em cenários não simultâneos, você também pode:
Crie um gatilho de temporizador de um minuto para garantir que haja pelo menos uma instância ativa.

5.6 Como saber a memória atual usada para a execução da função?

As informações retornadas sobre uma função contêm o máximo de memória consumida. Para obter mais informações, consulte [APIs do SDK](#) no *Guia de desenvolvedor do FunctionGraph*. Como alternativa, verifique o uso de memória no resultado da execução.

5.7 Por que a minha primeira solicitação é lenta?

As funções são iniciadas a frio. Se a inicialização ou uma operação longa for realizada durante a execução da primeira função, a primeira solicitação será atrasada. No entanto, as solicitações subsequentes antes da exclusão do contêiner serão mais rápidas. Se não houver nenhuma solicitação dentro de um minuto, o contêiner será excluído.

5.8 O que fazer se ocorrer um erro ao chamar uma API?

Corrija a falha consultando a seção [Códigos de erro](#). Se a falha persistir, entre em contato com o suporte técnico.

5.9 Como ler o cabeçalho da solicitação de uma função?

O primeiro parâmetro no manipulador da função contém o cabeçalho da solicitação. Você pode imprimir o resultado da execução da função para obter os campos obrigatórios.

Como mostrado na figura a seguir, **event** é o primeiro parâmetro no manipulador de função, e **headers** é o cabeçalho da solicitação.

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

5.10 Uma função pode invocar outra função?

Sim. Uma função pode chamar outra função usando uma API ou SDK na mesma região. Para invocação entre regiões, o chamador deve ter acesso público.

Usar uma API

Para obter detalhes, consulte [Exemplo 4: uso de uma função para chamar uma subfunção](#).

Usar um SDK (Python)

1. Crie uma função chamada **FuncB**, que será invocada.

Crie a função do zero com o tempo de execução do Python 3.6 e o seguinte código:

```
# -*- coding:utf-8 -*-
import json
def handler (event, context):
    # Prints input parameters to verify the invocation result.
    logger = context.getLogger()
    logger.info(event)
    return {
        "statusCode": 200,
        "isBase64Encoded": False,
        "body": json.dumps(event),
        "headers": {
            "Content-Type": "application/json"
        }
    }
```

2. Crie uma função chamada **FuncA**.

1. Crie a função do zero com o tempo de execução do Python 3.6 e o código a seguir. Substitua *<Your Function Region>* e *<Your Function URN>* pela região e URN do **FuncB**.

```
# -*- coding:utf-8 -*-
from huaweicloudsdkcore.auth.credentials import BasicCredentials
```

```
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import
FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *
def handler (event, context):
    logger = context.getLogger()
    ak = context.getAccessKey()
    sk = context.getSecretKey()

    credentials = BasicCredentials(ak, sk)
    client = FunctionGraphClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(FunctionGraphRegion.value_of("<Your Function Region>")) \
        .build()

    try:
        request = InvokeFunctionRequest()
        request.function_urn = "<Your Function URN>"
        # Input parameters.
        request.body = '{"input_key':'input_value'}"
        # Response body format. v0: text; v1: JSON.
        request.x_cff_request_version = "v1"
        response = client.invoke_function(request)
        logger.info(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)
    return {
        "statusCode": 200,
        "isBase64Encoded": False,
        "body": "",
        "headers": {
            "Content-Type": "application/json"
        }
    }
```

- a. Configure dependências para a função.

Configure dependências **huawei-cloud-sdk-core-python-3.x** e **huaweicloud-sdk-python-v3**, que podem ser baixadas de [Visão geral do SDK](#). Para obter detalhes sobre como criar uma dependência, consulte [Como criar dependências de função?](#)

- b. Conceda permissões para a função.

Conceda a permissão **FunctionGraph CommonOperation** à função criando uma agência para que a função possa chamar outras funções. Para obter detalhes, consulte [Criação de uma agência](#).

3. Verifique o resultado da execução.

Configure um evento de teste em branco para **FuncA** e clique em **Test**. Um resultado semelhante ao seguinte é exibido.

Figura 5-1 Visualização do resultado da execução

```
2022-07-12T12:11:01Z Start invoke request '1b588617-ab5e-4bbc-bb9f-913b4eb21b97', version: latest
2022-07-12T12:11:02Z 1b588617-ab5e-4bbc-bb9f-913b4eb21b97 INFO {"request_id": "ee52f806-fad2-443a-9527-f5451106510b", "result": {"statusCode": 200, "isBase64Encoded": false, "body": ""}, {"input_key": "input_value"}}, {"headers": {"Content-Type": "application/json"}, "status": 200}
2022-07-12T12:11:02Z Finish invoke request '1b588617-ab5e-4bbc-bb9f-913b4eb21b97', duration: 1558.074ms, billing duration: 1559ms, memory used: 35.035MB, billing memory: 128MB
```

Visualize os logs de execução de **FuncB's** gerados no mesmo período.

```
2022-07-12T12:11:02Z Start invoke request 'ee52f806-fad2-443a-9527-f5451106510b', version: latest
2022-07-12T12:11:02Z ee52f806-fad2-443a-9527-f5451106510b {"input_key": "input_value"}
2022-07-12T12:11:02Z Finish invoke request 'ee52f806-fad2-443a-9527-f5451106510b', duration: 36.095ms, billing duration: 37ms, memory used: 20.754MB, billing memory: 128MB
```

5.11 A interface de execução síncrona pode ser invocada em uma rede privada?

Sim, a interface é invocada em uma rede privada por padrão. Para invocação entre regiões, ative o acesso à rede pública.

5.12 Por que uma função usa mais memória do que estimada e até aciona o alarme de falta de memória?

1. A análise de eventos e o cache consomem memória extra durante a invocação da função.
2. Após a conclusão da invocação, a memória recuperada geralmente é colocada no pool interno em vez de voltar ao sistema operacional, resultando em alto uso de memória. Isso é mais óbvio no caso de alta simultaneidade.

5.13 Como verificar o uso da memória ao ver "runtime memory limit exceeded"?

Verifique a memória usada na resposta.

Figura 5-2 Verificar a memória utilizada

```
2022-07-21T07:10:22Z Start invoke request '696ca9a9-22d9-4602-a354-decf8c99aac3',  
version: latest  
2022-07-21T07:10:23Z Finish invoke request '696ca9a9-22d9-4602-a354-  
decf8c99aac3'(invoke Failed:RuntimeMemoryExceedLimit), duration: 1266.043ms, billing  
duration: 1267ms, memory used: 510.953MB, billing memory: 512MB
```

5.14 Como resolver problema de "CrashLoopBackOff"?

A mensagem "CrashLoopBackOff: a aplicação dentro do contêiner continua travando" é exibida quando ocorre uma falha de execução de imagem personalizada. Neste caso, execute as seguintes operações:

1. Analise as causas.

Figura 5-3 Visualização do resultado da execução

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

2. Verifique a imagem de contêiner consultando [Implementação de uma função usando uma imagem de contêiner](#).
3. Verifique se a imagem usa a arquitetura x86 do Linux. Atualmente, apenas as imagens x86 do Linux são compatíveis.

5.15 Depois que atualizar uma imagem com o mesmo nome, as instâncias reservadas ainda usam a imagem antiga. O que devo fazer?

Use uma tag não mais recente para gerenciar atualizações de imagem e não use o mesmo nome de imagem.

6 Perguntas frequentes sobre configuração da função

6.1 Posso definir variáveis de ambiente ao criar funções?

Sim. Defina variáveis para passar configurações dinamicamente para o código de função e bibliotecas sem alterar o código. Para obter mais informações, consulte [Configuração de variáveis de ambiente](#).

6.2 Posso inserir informações sensíveis em variáveis de ambiente?

O FunctionGraph exibe todas as informações inseridas em texto simples. Portanto, não insira informações insensíveis, como senhas, ao definir variáveis de ambiente.

6.3 Como colocar marca d'água em imagens?

Veja [Marca d'água em imagens](#).

6.4 Como usar uma função para chamar uma sub-função?

Para obter detalhes, consulte "Uso de uma função para chamar uma sub-função".

6.5 Como usar as versões e aliases de uma função HTTP com um gatilho de APIG para atualização cinza?

1. Crie as versões V1 e V2 com base em **latest**, crie um alias chamado **alias** para V1 com um peso de 100% e crie um gatilho de APIG para **alias**.

Figura 6-1 Publicação das versões V1 e V2

Version	Aliases	Description	Last Modified
V2	-	-	just now
V1	-	-	just now

Figura 6-2 Criar alias e vincular V1 a ele

Name	Version	Description	Modified
alias	V1 (Weight: 100 %)	-	just now

Figura 6-3 Criar um gatilho para o alias

NOTA

Você pode criar gatilhos de APIG para um alias de função ou versão. Por padrão, um nome de trigger é **API_{Nome da função}{Versão}** e o caminho de solicitação é **/{Nome da função}-{Versão}**.

Um URL de gatilho de APIG só será usado para um alias ou sua versão correspondente. Por exemplo, suponha que V1 tenha sido vinculado ao **alias**. Se você criar um gatilho com o nome padrão **API_testapigV1** para V1 e, em seguida, criar outro gatilho com o mesmo nome e URL para **alias**, o gatilho **API_testapigV1** não será exibido na lista de gatilhos de V1.

- Volte para a página de guia **Aliases** de **latest**, clique em **Edit** na linha de **alias** e defina **Additional Version** como **V2** com um peso personalizado. Isto é para atualização cinza de V1 para V2.

NOTA

Os pesos indicam a porcentagem de dados recebidos que serão alocados às versões correspondentes. Você pode definir um peso que atenda aos seus requisitos de serviço.

Figura 6-4 Editar um alias

★ Alias ⓘ

Enter 1 to 63 characters, starting with a letter and ending with a letter or digit. Only letters, digits, hyphens (-), and underscores (_) are allowed.

★ Version Weight 90 %

Traffic Shifting

Requests of the major version will be forwarded to an additional version according to the specified weight. [Learn more](#)

Additional Version ★ Weight %

Description

0/512

3. Criar versão V3, clique em **Edit** na linha de **alias** na página de guia **Aliases** e altere **Version** para **V2** e **Additional Version** para **V3** com um peso personalizado. Isto é para atualização cinza de V2 para V3.

Figura 6-5 Criação da V3

Version	Aliases	Description	Last Modified
V3	-	-	just now
V2	-	-	3 minutes ago
V1	-	-	3 minutes ago

Figura 6-6 Editar um alias

★ Alias ⓘ

Enter 1 to 63 characters, starting with a letter and ending with a letter or digit. Only letters, digits, hyphens (-), and underscores (_) are allowed.

★ Version Weight 10 %

Traffic Shifting

Requests of the major version will be forwarded to an additional version according to the specified weight. [Learn more](#)

Additional Version ★ Weight %

Description

0/512

AVISO

Depois de publicar uma versão com base em **latest**, você pode publicar mais versões somente depois de fazer uma configuração ou alteração de código na função.

Aliases vinculados a gatilhos não podem ser excluídos.

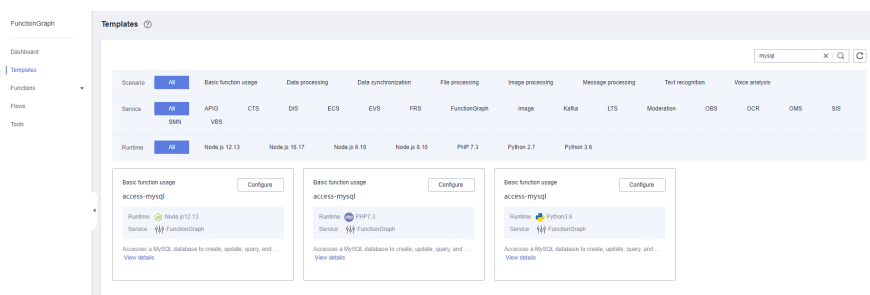
7 Perguntas frequentes sobre acesso a recursos externos

7.1 Como uma função acessa o banco de dados MySQL?

Realize as operações a seguir:

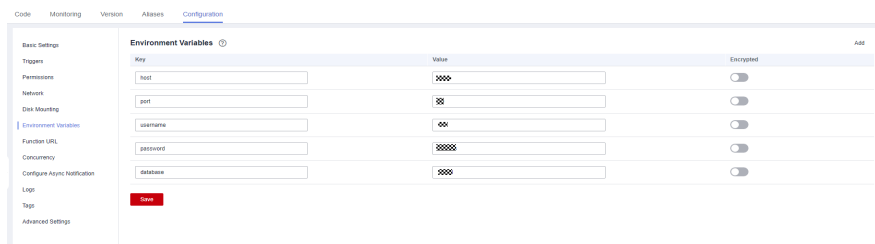
1. Verifique se o banco de dados MySQL está implantado em uma VPC.
 - Sim: configure a mesma VPC e a mesma sub-rede que o banco de dados MySQL para a função consultando [Configuração do acesso à VPC](#).
 - Não: veja [Como configurar o acesso à rede externa?](#)
2. Procure por modelos de MySQL e selecione aquele com o tempo de execução desejado, como mostrado na [Figura 7-1](#). Defina os parâmetros conforme necessário e clique em **Create Function**.

Figura 7-1 Selecionar um modelo de função



3. Depois que a função de MySQL for criada, escolha **Configuration > Environment Variables**, ative a criptografia conforme necessário (veja [Figura 7-2](#)), e clique em **Save**.

Figura 7-2 Ativar a criptografia



NOTA

Se a função precisar acessar as APIs do RDS, [crie uma agência](#) e conceder as permissões necessárias.

7.2 Como uma função acessa o Redis?

Realize as operações a seguir:

1. Verifique se a instância do Redis está implementada em uma VPC.
 - Se a instância do Redis for implementada em uma VPC, configure a mesma VPC e a mesma sub-rede que a instância do Redis para a função consultando [Configuração do acesso da VPC](#).
 - Se a instância do Redis for construída em uma rede pública, obtenha seu endereço IP público.
2. Compile código para conectar uma função à instância do Redis.

O FunctionGraph integrou a biblioteca de terceiros [redis-py](#) em seus tempos de execução de Python 2.7 e Python 3.6. Portanto, você não precisa baixar nenhuma outra biblioteca do Redis.

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

NOTA

- Se a função não conseguir acessar a instância do Redis em uma rede pública, execute as seguintes operações:
 - Modifique o arquivo **redis.conf** para permitir o acesso de qualquer endereço IP.
 - Defina uma senha para acessar a instância do Redis no arquivo **redis.conf**.
 - Desative o firewall.
- Se a função precisar acessar as APIs do DCS, [crie uma agência](#) e conceder as permissões necessárias.

7.3 O que fazer se minha função não conseguir se conectar ao Redis por meio de uma VPC?

Sintoma

Com o mesmo código, a função pode se conectar ao Redis 1 através da VPC 1, mas não pode se conectar ao Redis 2 através da VPC 2.

Solução

1. Verificar a configuração atual: o Redis1 está na VPC1, o Redis2 está na VPC2, e o cliente do Redis usa o mesmo conjunto de códigos.
2. Verifique o log, que mostra que o endereço IP do Redis aparece como caracteres ilegíveis quando o cliente tenta conectar o Redis2 por meio da VPC2.



3. Analise o Redis1 e o Redis2. Eles são os mesmos, exceto por seus endereços IP e senhas. A senha do Redis2 foi encontrada para conter um sinal de arroba (@). Como resultado, o endereço IP foi incorretamente interceptado e distorcido quando o Redis foi solicitado.
4. Altere a senha do Redis2 para corrigir a falha.

7.4 Como configurar o acesso à rede externa?

Por padrão, as funções implementadas em uma VPC são isoladas da Internet. Se uma função precisar acessar redes internas e externas, adicione um gateway NAT público para a VPC.

Pré-requisitos

1. Você criou uma VPC e uma sub-rede de acordo com [Criação de uma VPC](#).
2. Você obteve um endereço IP elástico de acordo com [Atribuição de um EIP](#).

Procedimento de criação de um gateway NAT público

Passo 1 Faça logon no console do NAT Gateway e clique em **Buy Public NAT Gateway**.

Passo 2 Na página exibida, insira as informações do gateway, selecione uma VPC e uma sub-rede (por exemplo, **vpc-01**) e confirme e envie as configurações para comprar um NAT gateway. Para obter detalhes, consulte [Compra de um gateway NAT público](#).

Passo 3 Clique no nome do gateway NAT público. Na página de detalhes exibida, clique em **Add an SNAT Rule** e clique em **OK**.

----Fim

8 Outras perguntas frequentes

8.1 Como visualizar as regras de alarme configuradas para uma função?

Faça login no console do Cloud Eye e visualize as regras de alarme.

8.2 O FunctionGraph oferece suporte à descompilação de ZIP durante a transcodificação de vídeo?

Não. Por favor, descompile seus arquivos antes de enviá-los.

8.3 Os recursos criados durante o FunctionGraph OBT 2.0 serão liberados automaticamente quando expirarem? Serão cobrados?

O FunctionGraph libera recursos automaticamente quando eles expiram e você não será cobrado por eles.

8.4 O que é uma aplicação no FunctionGraph?

Uma aplicação funciona como uma pasta. Depois que uma função é criada, ela é automaticamente categorizada na aplicação **default** e não pode ser alternada para outras aplicações. No futuro, as funções serão gerenciadas pelo rótulo para uma melhor experiência.

8.5 Preciso pagar pelo tempo de inicialização a frio?

Não. O tempo de inicialização a frio não será medido e você não precisa pagar por isso.

8.6 Por que uma mensagem é exibida indicando que minha conta foi suspensa ao criar uma função?

A sua conta está em atraso.

8.7 As solicitações de todas as minhas funções em diferentes regiões serão cobradas?

Sim.

9 Migração do FunctionGraph V1 para V2

9.1 Quais problemas de compatibilidade existem durante a migração?

1. Diferença de `args`

V1:

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

Após a migração para V2, será:

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

O `sys.argv` do tempo de execução do Python varia entre as duas versões.

Para V2, é `['/home/snuser/runtime/python3.6/server.py', '127.0.0.1:31536', '/opt/function/code']`, cujos dois últimos parâmetros não estão disponíveis para V1.

2. Diferença de `asyncio`

V1:

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

Após a migração para V2, será:

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

Na V1, `asyncio.get_event_loop()` obtém o loop de eventos atual de thread do SO (thread principal). Ele lançará `RuntimeError` na V2, porque o tempo de execução do Python da V2 não executa funções em thread principal.

Para usar `asyncio` na V2, crie um loop de eventos.

9.2 Por que não posso imprimir logs de uma função do Python 2.7 após a execução de `reload(sys)`?

Tente imprimir logs com `context.getLogger()`.

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