

FunctionGraph

Visão geral de serviço

Edição 01
Data 2022-11-16



Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2022. 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 Infográficos do FunctionGraph.....	1
2 O que é o FunctionGraph?.....	3
3 Funções do produto.....	6
4 Vantagens do produto.....	9
5 Cenários de aplicação.....	11
6 Tipos de função.....	13
6.1 Funções do evento.....	13
6.2 Funções de HTTP.....	14
7 Cotas e restrições de uso.....	15
8 Cobrança.....	18
9 Gerenciamento de permissões.....	21
10 Conceitos.....	24
11 Relações entre FunctionGraph e outros serviços.....	26

1 Infográficos do FunctionGraph

FunctionGraph 2.0
Next-Generation Function Computing

FunctionGraph allows you to run your code without provisioning or managing servers while ensuring **high availability and scalability**.

Simply upload your code and set execution conditions, and FunctionGraph will take care of the rest: the way for what you use, not when your code is not running.

FunctionGraph 2.0 is a compute service powered by Huawei YunRong (Huawei YunRong is a **distributed kernel** for cloud-native serverless technologies) **enables** function programming, multi-provider full function runtime, and efficient load service distribution. Developers now build cloud-native applications more easily.

1. Multiple Development Languages and Triggers

Six Languages

1) Develop functions using the four mainstream languages.

- Python 3.8
- Node.js 12
- Java 8
- Go 1.12

2) Use custom programming languages or image packages.

- ARM Linux/Windows
- ARM Linux/Windows

10+ Triggers for Function Invocation

Synchronous Invocation: The client needs to wait until it receives a response. The requested function must be executed and return a response as the client can proceed to the next step.

Asynchronous Invocation: The client does not need to wait after it sends requests to the server. The server jobs required requests first, it queues, and processes them one by one. The server then returns the processed results to the client.

2. Easier Development with Function Flows

Drag and drop to create flow charts and orchestrate components.

Commonall:

- Component
- Flowchart
- Function
- Event
- Condition
- Parallel
- Sequence
- Loop
- Timeout
- Wait
- Condition
- Function
- Event

3. On- and Off-Cloud Development and Debugging

(On-Cloud) CloudIDE

Cloud IDE (On-Cloud) → Code Repository (On-Cloud) → Function Graph → Push Request to Server

(Off-Cloud) VSCode Plug-In

Cloud IDE (Off-Cloud) → Code Repository (Off-Cloud) → Function Graph → Push Request to Server

4. Serverless Reconstruction with HTTP Functions

Package the **original web service code** into an HTTP function using API Gateway and FunctionGraph for serverless reconstruction.

Traditional Architecture: Client → Web Service → API Gateway → Function Graph → Backend

Serverless Reconstruction: Client → API Gateway → Function Graph → Backend

5. Flexible Resource Scheduling

To take maximum compute resources under any strategy in the case of varying traffic, and maintain optimal performance needs to be balanced with usage efficiency, and **dynamic resource assignment to the server**.

Optimal usage → API Gateway → Function Graph (Dynamic Resource Assignment) → Backend

6. Cold Start in Hundreds of Milliseconds

Cold start:

- 300 ms
- 50 ms

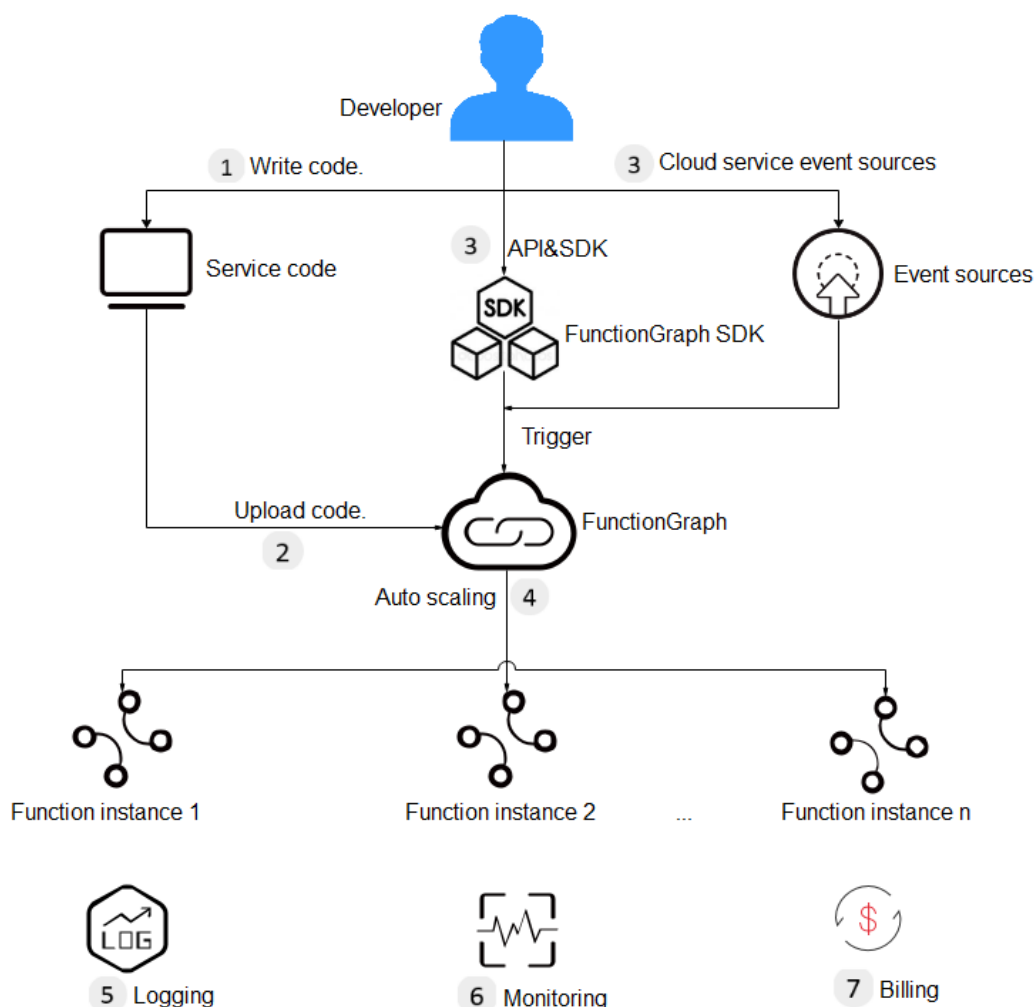
2 O que é o FunctionGraph?

Processo de utilização

O FunctionGraph hospeda e calcula funções baseadas em eventos em um contexto sem servidor, garantindo alta disponibilidade, alta escalabilidade e manutenção zero. Tudo o que você precisa fazer é escrever seu código e definir condições. Você paga apenas pelo que usa e não é cobrado quando o código não está em execução.

Figura 2-1 mostra o processo de utilização do FunctionGraph.

Figura 2-1 Processo de utilização



1. Escrever código.

Escreva código em Node.js, Python, Java, Go, C# ou PHP. Para obter detalhes, consulte [Guia do desenvolvedor do FunctionGraph](#).

2. Carregar o código.

Atualmente, você pode editar o código em linha, carregar um arquivo ZIP ou JAR ou obter um arquivo ZIP do OBS. Para mais detalhes, consulte [Tabela 3-2](#).

3. Disparar funções por chamadas de API ou eventos de serviço de nuvem.

Chame as API de RESTful ou use fontes de eventos de serviço de nuvem para acionar a execução de funções e gerar instâncias para implementar funções de serviço.

4. Escala automática é implementada.

O FunctionGraph implementa o escala automática com base no número de solicitações recebidas. Para obter detalhes, consulte [Cotas e restrições de uso](#).

5. Exibir registros.

Visualize registros de execução de funções, pois o FunctionGraph está interconectado com o Serviço de Tanque de Registros (LTS). Para obter detalhes, consulte [Consultando registros de função](#).

6. Visualizar informações de monitoramento.

Visualize informações gráficas de monitoramento à medida que o FunctionGraph é interconectado com o Cloud Eye. Para obter detalhes, consulte [Visualização de métricas de função](#).

7. Modo de cobrança

Depois que uma função é executada, você será cobrado com base no número de solicitações de execução da função e na duração da execução. (v1: faturado a cada 100 ms; v2: faturado a cada 1 ms)

3 Funções do produto

Gerenciamento de funções

O FunctionGraph fornece gerenciamento de funções baseado em console.

- Os runtimes Node.js, Java, Python, Go, C# e personalizados são suportados. [Tabela 3-1](#) fornece os detalhes.

NOTA

É aconselhável usar a versão de tempo de execução mais recente.

Tabela 3-1 Tempos de execução

Tempo de execução	Versão suportada
Node.js	6.10, 8.10, 10.16, 12.13, e 14.18
Python	2.7, 3.6, e 3.9
Java	8.0 and 11
Go	1.8 and 1.x
C#	.NET Core 2.0, .NET Core 2.1, e .NET Core 3.1
PHP	7.3
Custom	-

- Vários modos de entrada de código
O FunctionGraph permite editar código em linha, carregar um arquivo ZIP do OBS ou carregar diretamente um arquivo ZIP ou JAR. [Tabela 3-2](#) lista os modos de entrada de código suportados para cada tempo de execução.

Tabela 3-2 Modos de entrada de código

Tempo de execução	Editando código em linha	Carregando um arquivo ZIP	Carregando um arquivo JAR	Carregando um arquivo ZIP do OBS
Node.js	Compatível	Compatível	Incompatível	Compatível
Python	Compatível	Compatível	Incompatível	Compatível
Java	Incompatível	Compatível	Compatível	Compatível
Go	Incompatível	Compatível	Incompatível	Compatível
C#	Incompatível	Compatível	Incompatível	Compatível
PHP	Compatível	Compatível	Incompatível	Compatível
Custom	Compatível	Compatível	Incompatível	Compatível

Gatilho

O FunctionGraph oferece suporte a vários tipos de gatilho, como Notificação de Mensagem Simples (SMN), API Gateway (APIG) e tipos de gatilho OBS. [Tabela 3-3](#) lista os tipos de gatilho suportados e o modo de chamada de função de cada tipo de gatilho.

Tabela 3-3 Modos de invocação de função

Gatilho	Modo de invocação da função
Gatilho SMN	Invocação assíncrona
gatilho do Serviço de Mensagens Distribuídas (DMS)	Invocação assíncrona
Gatilho APIG	Invocação síncrona
Gatilho OBS	Invocação assíncrona
gatilho do Serviço de Ingestão de Dados (DIS)	Invocação assíncrona
Gatilho Timer	Invocação assíncrona
Gatilho de Kafka	Invocação assíncrona

Registrações e métricas

O FunctionGraph exibe métricas de monitoramento de funções e coleta registros de execução de funções, permitindo visualizar status de funções e localizar problemas consultando registros.

Para obter detalhes sobre como consultar registros, consulte [Gerenciamento de registros de função](#).

Para obter detalhes sobre como exibir métricas de função, consulte [Visualização de métricas de função](#).

Para obter detalhes sobre informações de monitoramento em nível de locatário, consulte [Introdução ao Dashboard](#).

Inicialização da função

A interface do inicializador é introduzida em:

- Isole a inicialização da função e o processamento de solicitações para permitir uma lógica de programa mais clara e um código melhor estruturado e de maior desempenho.
- Assegure a atualização suave da função para evitar perda de desempenho durante a inicialização a frio da camada de aplicação. Habilite novas instâncias de função para executar automaticamente a lógica de inicialização antes de processar as solicitações.
- Identifique a sobrecarga da inicialização da camada de aplicativo e determine com precisão o tempo para o dimensionamento de recursos e a quantidade de recursos necessários. Esse recurso torna a latência de solicitação mais estável quando a carga do aplicativo aumenta e mais instâncias de função são necessárias.

Funções HTTP

Você pode definir **Function Type** como **HTTP Function** na página de criação da função. As funções HTTP são projetadas para otimizar os serviços da Web. Você pode enviar solicitações HTTP para os URL para acionar a execução da função. As funções HTTP suportam apenas gatilhos APIG e API Connect (APIC).

NOTA

Este recurso é suportado apenas pelo FunctionGraph v2.

Rastreamento

Você pode ativar o rastreamento para funções. Em seguida, você pode acessar o console do Application Performance Management (APM) para exibir a JVM e as informações de rastreamento. Atualmente, apenas funções Java podem ser rastreadas.

Imagens personalizadas

Você pode empacotar e fazer upload diretamente de imagens de contêiner. As imagens são carregadas e iniciadas pela plataforma e podem ser chamadas de maneira semelhante às funções HTTP. Ao contrário do modo de upload de código anterior, você pode usar um pacote de código personalizado, que é flexível e reduz os custos de migração.

NOTA

Este recurso é suportado apenas pelo FunctionGraph v2.

4 Vantagens do produto

Sem servidores para gerenciar

O FunctionGraph executa seu código automaticamente e libera você do provisionamento e do gerenciamento de servidores, permitindo que você se concentre na inovação dos negócios.

Escala automática

O FunctionGraph é dimensionado automaticamente para atender às flutuações nas demandas de recursos e garante que o serviço permaneça acessível mesmo durante picos e picos.

Ele dimensiona automaticamente os recursos de entrada/saída com base no número de solicitações de serviço e distribui solicitações para instâncias de função por meio de balanceamento de carga automático.

Gatilho baseado em eventos

O FunctionGraph se integra a vários serviços em nuvem, como SMN e OBS, em um mecanismo de acionamento baseado em eventos para atender aos requisitos do serviço.

Ele é interconectado com os serviços LTS e Cloud Eye, permitindo que você visualize logs e métricas de funções sem a necessidade de configurações.

Alta disponibilidade

Se uma instância falhar, o FunctionGraph iniciará outra instância para processar novas solicitações e liberará recursos da instância não saudável.

Pagamento por uso

Você será cobrado com base no número de solicitações de função e na duração da execução e não será cobrado quando o código não estiver em execução.

Faturamento de instância reservada

As instâncias reservadas podem ser criadas para inicializar funções e eliminar a influência do cold start nos serviços. As instâncias reservadas estão sempre vivas no ambiente de execução.

Para o uso de instâncias reservadas, você será cobrado com base no número de solicitações e na duração de execução das instâncias reservadas. A duração mínima da corrida é de 60s.

5 Cenários de aplicação

O FunctionGraph é adequado para vários cenários, como processamento de arquivos em tempo real, processamento de fluxo de dados em tempo real, back-ends de aplicativos web e móveis e aplicativos de IA.

Cenário 1: Aplicações orientadas a eventos

Os serviços são executados no modo orientado a eventos e os recursos são provisionados com base nas demandas. Os desenvolvedores não precisam se preocupar com picos de serviço ou depressões. Os recursos ociosos não são faturados, reduzindo os custos de O&M. Os aplicativos orientados a eventos incluem processamento de arquivos, processamento de imagens, transmissão/transcodificação ao vivo, processamento de fluxo de dados em tempo real e processamento de regras/eventos da IoT.

- **Processamento de arquivos em tempo real**

Quando os arquivos são carregados de um cliente para o OBS, as funções podem ser acionadas para criar miniaturas de imagens em tempo real, converter formatos de vídeo, agregar e filtrar arquivos de dados ou implementar outras operações de arquivos.

Vantagens:

- O FunctionGraph aloca automaticamente recursos para executar mais instâncias de função à medida que o número de solicitações recebidas aumenta.
- Os arquivos são carregados no OBS para acionar as funções de processamento de arquivos. O FunctionGraph aloca automaticamente recursos para executar mais instâncias de função à medida que o número de solicitações recebidas aumenta.
- Você será cobrado somente pelos recursos usados para processar arquivos conforme necessário. (você não é cobrado por recursos ociosos durante baixas na demanda).

Por exemplo, crie dois intervalos no OBS e crie uma função para compactar imagens carregadas no intervalo especificado. Para obter detalhes, consulte [Compactação de imagens](#).

- **Processamento de fluxo de dados em tempo real**

O FunctionGraph trabalha com o DIS para processar fluxos de dados em tempo real. O FunctionGraph é compatível com rastreamento de atividade de aplicativos, processamento de transações sequenciais, análise de fluxo de dados, classificação de dados, geração de métricas, filtragem de logs, indexação, análise de mídia social e telemetria e medição de dados de dispositivos IoT.

Vantagens:

- Os dados são coletados por meio de fluxos DIS para acionar funções de processamento de dados.
- O FunctionGraph aloca automaticamente recursos para executar mais instâncias de função à medida que o número de solicitações recebidas aumenta.
- Você será cobrado somente pelos recursos usados para processar arquivos conforme necessário . (você não é cobrado por recursos ociosos durante baixas na demanda).

Cenário 2: Aplicações da web

Conecte o FunctionGraph com outros serviços em nuvem ou suas MV para criar rapidamente back-ends móveis e web altamente disponíveis e escaláveis. Os aplicativos da Web incluem miniprogramas, páginas da Web/aplicativos, chatbots e Backends para Frontends (BFF).

Vantagens:

- O FunctionGraph garante alta confiabilidade dos dados do site usando OBS e CloudTable e alta disponibilidade da lógica do site usando o API Gateway.
- O FunctionGraph aloca automaticamente recursos para executar mais instâncias de função à medida que o número de solicitações recebidas aumenta.
- Você será cobrado somente pelos recursos usados para processar arquivos conforme necessário . (você não é cobrado por recursos ociosos durante baixas na demanda).

Cenário 3: Aplicações de IA

A evolução da inteligência requer que vários serviços sejam integrados para uma implementação rápida. Os aplicativos de IA incluem integração de serviços de terceiros, inferência de IA, reconhecimento facial e reconhecimento de placas de veículos.

Vantagens:

- O FunctionGraph trabalha com serviços de IE para reconhecimento de texto e moderação de conteúdo para atender a uma ampla variedade de cenários.
- Você só precisa solicitar serviços relacionados e escrever código de serviço sem ter que provisionar ou gerenciar servidores.
- Você será cobrado apenas pela execução de funções e pelos serviços de IE usados sem ter que pagar por recursos ociosos quando as demandas de serviço forem baixas.

6 Tipos de função

6.1 Funções do evento

NOTA

Você precisa escolher entre **Event Function** e **HTTP Function** ao criar uma função no FunctionGraph v2.

Visão geral

O FunctionGraph oferece suporte a funções de eventos. Um evento pode acionar a execução da função. Geralmente, ele está no formato JSON. Você pode criar um evento para acionar sua função por meio da plataforma de serviço em nuvem ou do CloudIDE. Todos os tipos de gatilhos suportados pelo FunctionGraph podem acionar funções de eventos.

NOTA

1. Na página de criação de função, **Function Type** é definido como **Event Function** por padrão.
2. Durante o teste, uma função pode ser acionada simplesmente inserindo o evento especificado no formato JSON.
3. Você também pode usar gatilhos para acionar funções de evento.

Vantagens

- Fácil programação de nó único
Você pode editar funções de evento ou fazer upload de pacotes de código no FunctionGraph ou CloudIDE e implantá-las com apenas alguns cliques. Não há necessidade de você se preocupar com a simultaneidade da função ou a retificação de falhas.
- Tempos de execução de alto desempenho e alta velocidade
As funções de evento podem ser iniciadas, dimensionadas e chamadas em milissegundos. Falhas podem ser detectadas e corrigidas em segundos.
- Cadeia de ferramenta completa
O FunctionGraph fornece registro, rastreamento, depuração e monitoramento abrangentes, permitindo que os desenvolvedores implementem funções em apenas três etapas.

Restrições

As funções de evento enfrentam restrições de origem de evento. Você precisa cumprir as regras de desenvolvimento de funções da plataforma de funções.

6.2 Funções de HTTP

NOTA

Este recurso é suportado apenas pelo FunctionGraph v2.

Visão geral

O FunctionGraph oferece suporte a funções de eventos e funções de HTTP. As funções de HTTP são projetadas para otimizar os serviços da Web. Você pode enviar solicitações de HTTP para os URL para acionar a execução da função. As funções de HTTP suportam apenas gatilhos de APIG e APIC.

NOTA

1. As funções de HTTP suportam o protocolo de HTTP/1.1.
2. Na página de criação de função, **HTTP Function** é adicionada recentemente.
3. A função de HTTP deve ser definida como **bootstrap**. Você pode escrever diretamente o comando **startup** e **allow access over port 8000**.

Vantagens

- Suporte para vários frameworks
Você pode usar estruturas da Web comuns, como Node.js Express e Koa, para escrever funções da Web e migrar seus serviços de estrutura da Web locais para a nuvem com o mínimo de modificações.
- Menos etapas de processamento de solicitações
As funções podem receber e processar diretamente solicitações de HTTP, eliminando a necessidade do API Gateway para converter o formato JSON. Isso acelera o processamento de solicitações e melhora o desempenho do serviço da Web.
- Experiência de escrita premium
Escrever funções de HTTP é semelhante a escrever serviços web nativos. Você também pode usar as API nativas do Node.js para aproveitar a experiência de desenvolvimento local.

Restrições

- As funções de HTTP suportam apenas gatilhos de APIG (compartilhados), APIG (dedicados) e APIC.
- Vários gatilhos de API podem ser vinculados à mesma função, mas todas as API devem pertencer ao mesmo serviço de APIG.
- Para funções de HTTP, o tamanho do corpo da resposta de HTTP não pode exceder 6 MB.

7 Cotas e restrições de uso

Restrições de recursos da conta

A tabela a seguir fornece as cotas para recursos da conta. Para obter detalhes sobre como consultar e modificar cotas, consulte [Cotas](#).

Tabela 7-1 Restrições de recursos da conta

Recursos	Restrições
Número máximo de funções que podem ser criadas em uma conta	400
Número máximo de versões permitidas para uma função	10
Número máximo de aliases permitidos para uma função	10
Número máximo de gatilhos (DIS/DMS/Kafka/Timer) permitidos para uma versão de função	10
Tamanho de um pacote de implantação de código (em formato ZIP ou JAR) que pode ser exibido no console do FunctionGraph	3 MB
Tamanho de um pacote de implantação de código (em formato ZIP ou JAR) que pode ser carregado no console do FunctionGraph	10 MB
Tamanho de um pacote de implantação de código (em formato ZIP ou JAR) que pode ser editado inline durante a invocação da API da função	50 MB
Tamanho de um pacote de implantação de código original permitido durante a invocação da API de função	<ul style="list-style-type: none">● Pacote ZIP que é carregado diretamente: 1500 MB● Carregamento do pacote através do OBS: 300 MB

Recursos	Restrições
Tamanho máximo de pacotes de implantação permitidos para uma conta	10 GB
Número de execuções simultâneas por conta	100
Número máximo de instâncias reservadas que uma conta pode criar	90 (Número de execuções simultâneas por conta x 90%)
Tamanho de todas as variáveis de ambiente de uma função	4096 caracteres

Restrições de recurso de execução de função

Tabela 7-2 Restrições de recurso de execução de função

Recursos	Restrições
Espaço em disco efêmero (/tmp de espaço)	512 MB
Número de descritores de arquivos	1024
Número total de processos e threads	1024
Duração máxima de execução por solicitação	900s
Tamanho válido da carga útil do corpo da solicitação de chamada (invocação síncrona)	6 MB
Tamanho da carga válida do corpo da resposta de chamada (invocação síncrona)	6 MB
Tamanho válido da carga do corpo da solicitação de chamada (invocação assíncrona)	256 KB
Tamanho dos recursos importados	Arquivo de ZIP ≤ 50 MB
Tamanho dos recursos exportados	≤ 50 MB

 **NOTA**

- Tamanho da carga válida do corpo de resposta de chamada (invocação síncrona): A cadeia de caracteres retornada ou a cadeia de caracteres JSON do corpo da resposta serializada é menor ou igual a 6 MB por padrão. O tamanho real dos dados varia dependendo das configurações de backend do FunctionGraph. O backend determina o tamanho dos dados serializados com um desvio de nível de byte. O tamanho real da carga válida é de 6 MB ± 100 bytes.
- Não é aconselhável chamar uma função cujo tempo de execução exceda 90s no console do FunctionGraph. Para invocar tal função, use invocação assíncrona.
- O tamanho da carga válida de um corpo de solicitação é de 6 MB quando um gatilho Kafka, DDS, DIS ou GaussDB (para Mongo) é usado e é de 4 MB quando um gatilho APIG é usado.

8 Cobrança

O FunctionGraph utiliza o pagamento por uso e o preço é calculado com base no número de solicitações e na duração da execução. Não há cobranças mínimas necessárias.

Preço total = Preço para pedidos Preço para duração da execução

Cobrança de Instância Comum

Preço para pedidos

- Você será cobrado com base no número de solicitações, que é o número total de vezes que todas as funções são chamadas.
- Os primeiros 1 milhão de pedidos por mês são gratuitos. Após o uso de 1 milhão de solicitações gratuitas, você será cobrado com base nas taxas especificadas no [Detalhes de Preços do FunctionGraph](#).

Preço da duração da execução

O FunctionGraph fornece instâncias reservadas e sob demanda, que são mensuradas de maneira diferente.

- **Instâncias reservadas:** Consulte [Cobrança de instância reservada](#).
- **Instâncias sob demanda:** Esse tipo de instância é criado e liberado pelo FunctionGraph e você será cobrado com base na duração real da execução. O tempo de execução é a partir do momento em que seu código de função começa a ser executado até o momento em que ele retorna uma resposta ou termina.

Encargos adicionais

Cobranças adicionais serão geradas se outros serviços em nuvem forem usados com o FunctionGraph. Para obter mais informações, consulte [Detalhes de Preços do FunctionGraph](#).

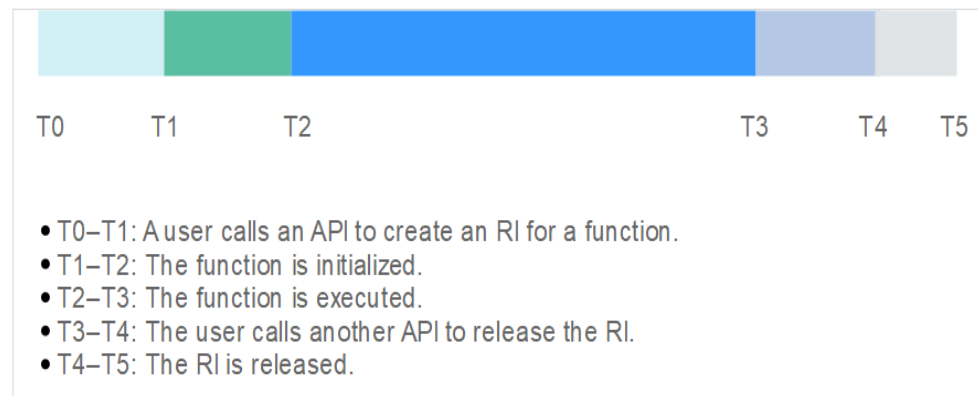
Cobrança de instância reservada

Você pode criar e liberar instâncias reservadas e será cobrado com base em sua duração de execução. As instâncias reservadas permanecem vivas no ambiente de execução, eliminando a influência de cold starts na latência.

- Se você chamar uma API para criar uma instância reservada, o faturamento começará assim que a criação da instância reservada for concluída.

- Se você chamar outra API para liberar uma instância reservada, as novas solicitações não serão roteadas para a instância reservada. A instância reservada será liberada dentro de um determinado período e o faturamento será interrompido.

Figura 8-1 Ciclo de vida de uma instância reservada



Como mostrado em [Figura 8-1](#), o faturamento dura de T1 a T4.

As instâncias reservadas são medidas em uma granularidade de segundo. Se uma instância reservada for executada por qualquer fração de minuto, você será cobrado pelo minuto completo. Caso contrário, você será cobrado com base na duração real da execução.

Por exemplo, se uma instância reservada for executada por 51s, você será cobrado por 1 minuto. Se a instância reservada for executada por 61s, você será cobrado por 61s.

- A unidade de duração de execução é GB-segundo, o que significa 1 GB por segundo.

Nível gratuito

O FunctionGraph oferece um nível gratuito, que você pode compartilhar com seus usuários do IAM.

- **Pedidos:** 1 milhão de solicitações gratuitas por mês.
- **Duração da execução:** 400.000 GB-segundos de tempo de execução gratuito por mês. Por exemplo, se a memória para uma função é de 1 GB, a duração da execução livre a cada mês é de 400.000s. Se a memória for de 512 MB, a duração da execução livre será de 800.000s.

AVISO

O nível gratuito é redefinido no início de cada mês.

[Tabela 8-1](#) mostra a duração de execução livre para diferentes especificações de memória.

Tabela 8-1 Duração da execução livre

Memória (MB)	Duração da execução gratuita (s) por mês
128	3.200.000
256	1.600.000
512	800.000
768	533.333
1024	400.000
1280	320.000
1536	266.667
Não	1024 x 400.000/N

Renovação

Para renovar a assinatura, escolha [Renewals](#) no console de gerenciamento. Para obter detalhes, consulte [Gerenciamento de renovação](#).

Expiração e pagamento em atraso

Se a sua conta estiver em atraso, pode ver os detalhes dos pagamentos em atraso no Centro de Cobrança. Para evitar que os recursos relacionados sejam interrompidos ou liberados, recarregue sua conta o mais cedo possível. Para obter detalhes, consulte [Execução de pagamentos pendentes \(clientes diretos pós-pagos\)](#).

9 Gerenciamento de permissões

Se você precisar atribuir permissões diferentes aos funcionários da sua empresa para acessar seus recursos do FunctionGraph, o IAM é uma boa opção para o gerenciamento de permissões refinado. O IAM fornece autenticação de identidade, gerenciamento de permissões e controle de acesso, ajudando você a proteger o acesso aos seus recursos de nuvem.

Com o IAM, você pode usar sua conta para criar usuários do IAM para seus funcionários e atribuir permissões aos usuários para controlar seu acesso a tipos de recursos específicos. Por exemplo, alguns desenvolvedores de software em sua empresa precisam usar recursos do FunctionGraph, mas não devem excluí-los ou executar operações de alto risco. Para alcançar esse resultado, você pode criar usuários do IAM para os desenvolvedores de software e conceder a eles apenas as permissões necessárias para usar os recursos do FunctionGraph.

Se sua conta não precisar de usuários individuais do IAM para o gerenciamento de permissões, você pode pular este capítulo.

O IAM pode ser usado gratuitamente. Você paga apenas pelos recursos em sua conta. Para obter mais informações sobre o IAM, consulte [Visão geral do serviço do IAM](#).

Permissões do FunctionGraph

Por padrão, os novos usuários de IAM não têm nenhuma permissão atribuída. Você precisa adicionar um usuário a um ou mais grupos e atribuir políticas de permissões a esses grupos. O usuário então herda permissões dos grupos dos quais é membro. Esse processo é chamado de autorização. Após a autorização, o usuário pode executar operações especificadas no FunctionGraph com base nas permissões.

O FunctionGraph é um serviço de nível de projeto implantado e acessado em regiões físicas específicas. Para atribuir permissões FunctionGraph a um grupo de usuários, especifique o escopo como projetos específicos da região e selecione os projetos para que as permissões entrem em vigor. Se **All projects** estiver selecionado, as permissões entrarão em vigor para o grupo de usuários em todos os projetos específicos da região. Ao acessar o FunctionGraph os usuários precisam mudar para uma região onde foram autorizados a usar o serviço FunctionGraph.

Você pode conceder permissões aos usuários usando funções e políticas.

- **Funções:** Um tipo de mecanismo de autorização de granulação grosseira que define permissões relacionadas às responsabilidades do usuário. Esse mecanismo fornece apenas um número limitado de funções de nível de serviço para autorização. Ao usar funções para conceder permissões, talvez você também precise atribuir outras funções

das quais as permissões dependem. No entanto, as funções não são uma escolha ideal para autorização refinada e controle de acesso seguro.

- Políticas: Um tipo de mecanismo de autorização refinado que define as permissões necessárias para realizar operações em recursos de nuvem específicos sob determinadas condições. Esse mecanismo permite uma autorização baseada em políticas mais flexível, atendendo aos requisitos de controle de acesso seguro.

Tabela 9-1 lista todas as políticas do sistema suportadas pelo FunctionGraph.

Tabela 9-1 Descrição de permissões

Nome da função/ política	Descrição	Categoria	Dependência
Administrador do FunctionGraph	Essa função tem as permissões para gerenciar funções, fluxos e gatilhos e invocar funções.	Função definida pelo sistema	Convidado do locatário
Invocador FunctionGraph	Essa função tem permissões para consultar funções, fluxos, gatilhos e invocar funções.	Função definida pelo sistema	N/A
FunctionGraph FullAccess	Esta política concede todas as permissões para FunctionGraph.	Política definida pelo sistema	N/A
FunctionGraph ReadOnlyAccess	Esta política concede permissões de somente leitura para o FunctionGraph.	Política definida pelo sistema	N/A
FunctionGraph CommonOperations	Essa política concede permissões para consultar funções e gatilhos e invocar funções.	Política definida pelo sistema	N/A

lista as operações comuns suportadas por cada política de FunctionGraph definida pelo sistema. Escolha as políticas apropriadas definidas pelo sistema de acordo com esta tabela.

Tabela 9-2 Operações comuns suportadas por cada política definida pelo sistema

Operação	Invocador FunctionGraph	Administrador do FunctionGraph	FunctionGraph ReadOnly Access	FunctionGraph CommonOperations	FunctionGraph FullAccess
Criação de funções	×	✓	×	×	✓
Consulta de funções	✓	✓	✓	✓	✓
Modificação de funções	×	✓	×	×	✓
Apagamento de funções	×	✓	×	×	✓
Invocação de funções	✓	✓	×	✓	✓
Consulta de registros de função	✓	✓	✓	✓	✓
Visualização de métricas de função	✓	✓	✓	✓	✓

Links úteis

- [Visão geral do serviço de IAM](#)
- [Criando um Usuário e Concedendo Permissões](#)
- [Políticas de permissões e ações suportadas](#)

10 Conceitos

Função

Funções são códigos definidos para lidar com eventos.

Origem do evento

Uma origem de evento é um serviço de nuvem pública ou aplicativo personalizado que publica eventos.

Invocação síncrona

Os clientes esperam por respostas explícitas às suas solicitações de uma função. As respostas são retornadas somente depois que a função é chamada.

Invocação assíncrona

Os clientes não se importam com os resultados de invocação de função de suas solicitações. Depois de receber uma solicitação, o FunctionGraph a coloca em uma fila, retorna uma resposta e processa outras solicitações quando há recursos ociosos.

gatilho

Um gatilho é um evento que dispara a execução de uma função.

Multi-concorrência de instância única

O número de solicitações que podem ser processadas simultaneamente por uma instância.

Imagens personalizadas

Você pode empacotar e fazer upload diretamente de imagens de contêiner. A plataforma então carrega e inicia essas imagens para criar funções.

Execução de função personalizada

Você pode personalizar scripts e arquivos para executar funções.

Registrações de função

registrações gerados durante a invocação da função.

Monitoramento da função

Monitoramento de informações geradas durante a execução da função.

Versão da função

O FunctionGraph permite publicar uma ou mais versões nos processos de desenvolvimento, teste e produção para gerenciar seu código de função. As variáveis de código e ambiente de cada versão são salvas como um instantâneo. Depois que o código da função for publicado, modifique as configurações quando necessário.

Alias da função

Você pode criar um alias para uma versão de função específica. Para reverter para uma versão anterior, use o alias correspondente para representar a versão em vez de modificar o código da função.

Cada alias de função pode ser vinculado a uma versão principal e a uma versão adicional para deslocamento de tráfego.

Pacote de dependência

O FunctionGraph permite gerenciar dependências de maneira unificada. Você pode carregar dependências de um caminho local ou por meio do OBS, se elas forem muito grandes, e especificar nomes para elas.

Para obter detalhes sobre como gerar dependências de função, consulte [Como criar dependências de função?](#)

Arquivo bootstrap

O arquivo **bootstrap** é o arquivo de inicialização de uma função HTTP. A função HTTP só pode ler **bootstrap** como o nome do arquivo de bootstrap. Se o nome do arquivo não for **bootstrap**, o serviço não poderá ser iniciado.

11 Relações entre FunctionGraph e outros serviços

Tabela 11-1 descreve os serviços de nuvem que foram interconectados com o FunctionGraph.

Tabela 11-1 Serviços interligados

Serviço	Função
SMN	As funções do FunctionGraph são construídas para processar notificações SMN. Para obter detalhes, consulte Guia de usuário do SMN .
DMS	As funções do FunctionGraph são configuradas para pesquisar automaticamente filas DMS para mensagens e processar novas mensagens. Para obter detalhes, consulte Guia de usuário do DMS .
API Gateway	As funções do FunctionGraph são invocadas via HTTPS definindo APIs REST com serviços de back-end especificados. Para obter detalhes, consulte Guia de usuário do API Gateway .
OBS	As funções do FunctionGraph são criadas para processar eventos de intervalo do OBS, como eventos de criação ou exclusão de objetos. Por exemplo, quando uma imagem é carregada no intervalo especificado, o OBS invoca a função para ler a imagem e criar uma miniatura. Para obter detalhes, consulte Guia de usuário do OBS .
DIS	As funções do FunctionGraph são criadas para pesquisar periodicamente fluxos DIS para novos registros, como fluxos de cliques, transações financeiras, fluxos de mídia social, registros de TI e eventos de rastreamento de localização. Para obter detalhes, consulte Guia do Usuário do DIS .

Serviço	Função
CTS	<p>As funções FunctionGraph são definidas analisar e processar a informação chave nos logs de acordo com as notificações de evento do tipo de serviço especificado e das operações configuradas no CTS.</p> <ul style="list-style-type: none">● Com o CTS, você pode registrar operações associadas ao FunctionGraph para consulta, auditoria e backtracking posteriores.● O CTS começa a gravar operações em recursos da nuvem que estão sendo permitidos uma vez. Veja traços dos últimos sete dias no console CTS.
Cloud Eye	<p>O FunctionGraph é interconectado ao Cloud Eye para relatar métricas de monitoramento, permitindo visualizar métricas de função e mensagens de alarme através do Cloud Eye. Para obter detalhes, consulte Guia do Usuário do Cloud Eye.</p> <ul style="list-style-type: none">● Para obter detalhes sobre métricas de função compatíveis com o Cloud Eye, consulte Configuração do monitoramento.
VPC	<p>As funções podem ser configuradas para acessar recursos em Virtual Private Clouds (as VPC) ou para acessar a Internet por meio da tradução de endereços de rede (SNAT) de origem, vinculando endereços IP elásticos. Para obter detalhes, consulte Guia de usuário da VPC.</p>