

FunctionGraph

Service Overview

Issue 01
Date 2019-06-27



Copyright © Huawei Technologies Co., Ltd. 2019. All rights reserved.

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

Trademarks and Permissions



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

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

Notice

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

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

Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: <http://e.huawei.com>

Contents

1 Overview.....	1
2 Product Features.....	4
2.1 Function-related Features.....	4
2.2 Workflow-related Features.....	6
2.3 Product Advantages.....	7
3 Application Scenarios.....	8
4 Permissions Description.....	10
5 Related Services.....	13
6 Feature Updates.....	16
7 Quotas and Usage Restrictions.....	22

1 Overview

FunctionGraph hosts and computes functions in a serverless context. It automatically scales to suit fluctuations in resource demands during peaks and spikes while requiring no reservation of dedicated servers or capacities. You will be billed based on actual usage. FunctionGraph consists of the function and workflow modules that respectively implement function computing and orchestration.

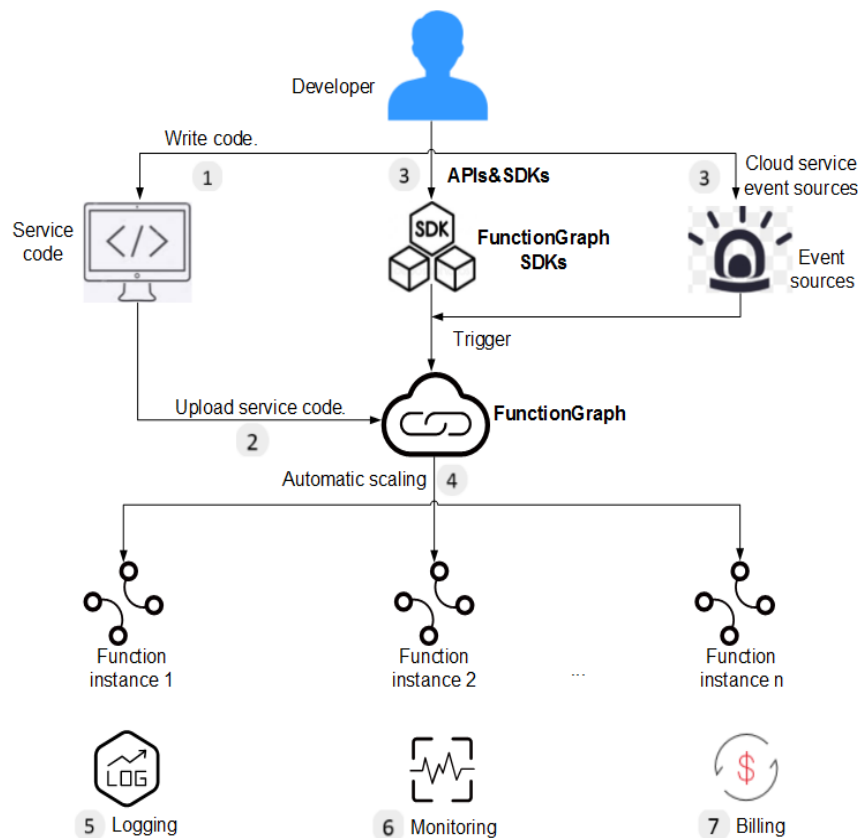
Function

FunctionGraph hosts event-driven functions while ensuring high availability, high scalability, and zero maintenance. All you need to do is write your code and set the conditions. You pay only for what you use and you are not charged when your code is not running.

Function Application Process

[Figure 1-1](#) shows the process of using functions.

Figure 1-1 Function application process



The process is explained as follows:

1. Write code in Node.js, Python, Java, Go, C#, or PHP. For details, see the [FunctionGraph Developer Guide](#).
2. Alternatively, edit code inline, directly upload a ZIP file or JAR file, or upload a ZIP file from OBS. For details, see [Table 2-2](#).
3. Create an API or set a cloud service event source to trigger the function. For details, see [Using an SMN Trigger](#), [Using a DMS Trigger](#), [Using an APIG Trigger](#), [Using an OBS Trigger](#), [Using a DIS Trigger](#), [Using an LTS Trigger](#), [Using a CTS Trigger](#), and [Using a Timer Trigger](#).
4. During function execution, FunctionGraph scales automatically based on the number of requests without the need for configurations. For details about the maximum number of function instances that can be run concurrently, see [Usage Restrictions](#).
5. FunctionGraph works with Log Tank Service (LTS), allowing you to query run logs of your function without the need for configurations. For details, see [Querying Logs](#).
6. FunctionGraph works with Cloud Eye, allowing you to view graphical monitoring information about your function without the need for configurations. For details, see [Function Monitoring](#).
7. After function execution, FunctionGraph bills based on the compute time you consume. The billing is down to the 100 ms range.

Workflow

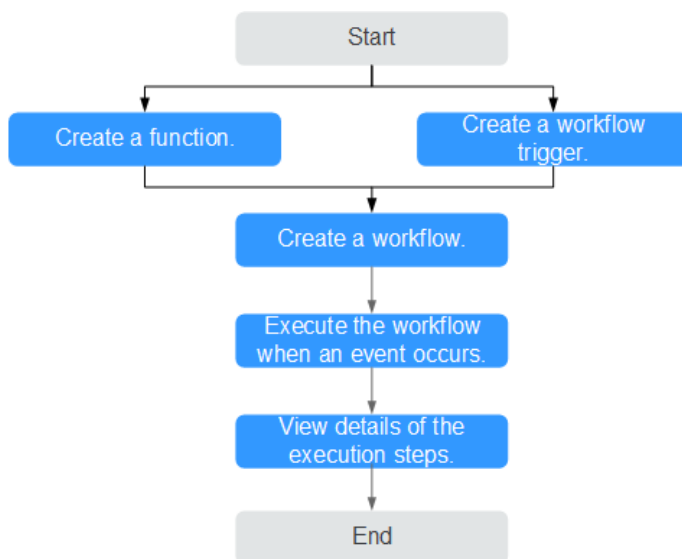
FunctionGraph coordinates distributed application functions and microservice components using visual workflows. With FunctionGraph, you can use independently executable, discrete

functions to build applications, which can be easily scaled up or modified to meet fast-growing and complex service requirements. FunctionGraph has the following features:

Workflow application process

Figure 1-2 shows the process of using workflows. Typically, you can orchestrate an internal trigger in an event workflow or bind external triggers to a workflow.

Figure 1-2 Workflow application process



2 Product Features

2.1 Function-related Features

Function Management

FunctionGraph provides console-based function management.

- The Node.js, Java, Python, Go, C#, and PHP runtimes are supported. You can also use a custom runtime. [Table 2-1](#) lists the supported runtimes.

Table 2-1 Runtime description

Runtime	Supported Version	Reference
Node.js	6.10 and 8.10	Developing Functions in Node.js
Python	2.7 and 3.6	Developing Functions in Python
Java	8.0	Developing Functions in Java
Go	1.8	Developing Functions in Go
C#	.NET Core 2.0 and .NET Core 2.1	Developing Functions in C#
PHP	7.3	Developing Functions in PHP
Custom	-	Using a Custom Runtime

- Multiple code entry modes
FunctionGraph allows you to edit code inline, upload a ZIP file from Object Storage Service (OBS), or directly upload a ZIP file or JAR file. [Table 2-2](#) lists the code entry modes supported for each runtime.

Table 2-2 Code entry modes

Runtime	Editing Code Inline	Uploading a ZIP File	Uploading a JAR File	Uploading a ZIP File from OBS
Node.js	Supported	Supported	Not supported	Supported
Python	Supported	Supported	Not supported	Supported
Java	Not supported	Supported	Supported	Supported
Go	Not supported	Supported	Not supported	Supported
C#	Not supported	Supported	Not supported	Supported
PHP	Supported	Supported	Not supported	Supported
Custom	Supported	Supported	Not supported	Supported

Triggers

FunctionGraph supports the following trigger types: SMN, DMS, APIG, OBS, DIS, LTS, CTS, and timer. [Table 2-3](#) lists the supported trigger types and the function invocation mode of each trigger type.

With OBS triggers, functions are invoked asynchronously. For details, see [Using an OBS Trigger](#).

Table 2-3 Function invocation modes

Trigger Type	Function Invocation Mode	Reference
SMN trigger	Asynchronous invocation	Using an SMN Trigger
DMS trigger	Asynchronous invocation	Using a DMS Trigger
APIG trigger	Synchronous invocation	Using an APIG Trigger
OBS trigger	Asynchronous invocation	Using an OBS Trigger
DIS trigger	Asynchronous invocation	Using a DIS Trigger
Timer trigger	Asynchronous invocation	Using a Timer Trigger
LTS trigger	Asynchronous invocation	Using an LTS Trigger
CTS trigger	Asynchronous invocation	Using a CTS Trigger

Logs and Metrics

FunctionGraph graphically displays function monitoring metrics and collects function run logs, enabling you to view function statuses, and locate problems by querying logs.

For details on how to query logs, see [Querying Function Logs](#).

For details on how to view the monitoring information about a function, see [Viewing Function Metrics](#).

For details about tenant-level monitoring information, see [Dashboard](#).

Function Initialization

The initializer interface is introduced to:

- Isolate function initialization and request processing to enable clearer program logic and better structured and higher-performance code.
- Ensure smooth function upgrade to prevent performance loss during the application layer's cold start initialization. Enable new function instances to automatically execute initialization logic before processing requests.
- Identify the overhead of application layer initialization, and accurately determine when resource scaling is performed and how many resources are required. This feature makes request latency more stable when the application load increases and more function instances are required.

2.2 Workflow-related Features

Graphical Console

FunctionGraph provides a graphical console, which can orchestrate the components of distributed applications by using visual workflows. You can use simple commands to define each step of an application, and FunctionGraph automatically generates a graphical workflow based on the defined steps.

After the application is started, FunctionGraph displays the execution status of each step graphically. You can quickly check whether each step is executed in the expected sequence. The console displays error information, helping you quickly locate and rectify faults.

Built-In Error Handling Mechanism

FunctionGraph can track the status of each step, and automatically handle errors by using the built-in retry and rollback functions.

FunctionGraph can automatically retry failed or timed-out tasks, capture specific errors, and restore the tasks. If all operations fail, FunctionGraph can roll back to the specified code.

Automatic Scaling

FunctionGraph automatically scales operation and underlying computing resources based on application steps to respond to changing workloads. This ensures that application workflows can always maintain high availability as the number of requests increases.

Execution Logs

FunctionGraph provides a real-time diagnosis and control panel, which records the details of each execution. When an error occurs, you can quickly locate and analyze the fault.

All execution records, including the overall status, faulty step, input, and output, are displayed in a visualized manner, providing information required for quick fault locating and rectification.

Management Security

FunctionGraph is interconnected with the Identity and Access Management (IAM) service of HUAWEI CLOUD. IAM policies can be used to control access to FunctionGraph.

2.3 Product Advantages

No Servers to Manage

FunctionGraph automatically runs your code and frees you from provisioning and managing servers, allowing you to focus on business innovation.

Auto-scaling

FunctionGraph automatically scales to suit fluctuations in resource demands and ensures that the service remains accessible even during peaks and spikes.

It automatically scales in/out resources based on the number of service requests, and distributes requests to function instances through automatic load balancing.

Event-based Triggering

FunctionGraph integrates with multiple cloud services (such as SMN, OBS, DIS, and DMS) in an event-based triggering mechanism to meet service requirements.

It is interconnected with the LTS and Cloud Eye services, allowing you to view function logs and metrics without the need for any configurations.

High Availability

If an instance becomes faulty, FunctionGraph starts another instance to process new requests and releases resources from the unhealthy instance.

Pay per Use

You will be billed based on the number of requests, execution duration, and number of state transitions (precise down to the 100 ms range).

3 Application Scenarios

FunctionGraph is suitable for various scenarios, such as real-time file processing, real-time data stream processing, web & mobile application backends, and artificial intelligence (AI) application.

Real-time File Processing

Uploading files from a client to OBS triggers functions that create image thumbnails in real time, convert video formats, aggregate and filter data files, or implement other file operations.

Advantages:

- FunctionGraph automatically scales out resources to run more function instances as the number of requests increases.
- Files are uploaded to OBS to trigger file processing functions.
- You will be billed only for resources used to process files as needed (you are not billed for idle resources during lows in demand).

For example, create two buckets in OBS, and build a function for compressing images uploaded to the specified bucket. For details, see [Compressing Images](#).

Real-time Stream Processing

Combine FunctionGraph with DIS to process real-time streaming data for such purposes as application activity tracking, transaction order processing, click stream analysis, data cleansing, metrics generation, log filtering, indexing, social media analysis, and IoT device data telemetry and metering.

Advantages:

- Data is collected by means of DIS streams to trigger data processing functions.
- FunctionGraph automatically scales out resources to run more function instances as the number of requests increases.
- You will be billed only for resources used to process files as needed (you are not billed for idle resources during lows in demand).

For example, build a function to process (such as to convert format) the real-time IoT data streams collected by DIS, and store the processed data into CloudTable Service (CloudTable). For details, see [Processing DIS Data](#).

Web & Mobile Backends

Interconnect FunctionGraph with other HUAWEI CLOUD services or your virtual machines to quickly build highly available and scalable web & mobile backends.

Advantages:

- FunctionGraph ensures high reliability of website data using OBS and CloudTable, and high-availability of website logic using API Gateway.
- FunctionGraph automatically scales out resources to run more function instances as the number of requests increases.
- You will be billed only for resources used to process files as needed (you are not billed for idle resources during lows in demand).

Artificial Intelligence

Integrate FunctionGraph with HUAWEI CLOUD EI services for text recognition and illicit image identification.

Advantages:

- FunctionGraph works with EI services for text recognition and content moderation to suit a wide range of scenarios - make adjustments whenever you need as demands change.
- You only need to apply for related services and write service code without having to provision or manage servers.
- You will be billed only for function execution and used EI services without having to pay for idle resources when service demands are low.

For example, build a function with an APIG trigger to provide pornographic image identification API services. For details, see [Building an AI Application for Identifying Pornographic Images](#).

4 Permissions Description

If you need to assign different permissions to employees in your enterprise to access your FunctionGraph resources, IAM is a good choice for fine-grained permissions management. IAM provides identity authentication, permissions management, and access control, helping you secure access to your HUAWEI CLOUD resources.

With IAM, you can use your HUAWEI CLOUD account to create IAM users for your employees, and assign permissions to the users to control their access to specific resource types. For example, some function developers in your enterprise need to use FunctionGraph resources but must not delete them or perform any high-risk operations. To achieve this result, you can create IAM users for the function developers and grant them only the permissions required for using FunctionGraph resources.

If your HUAWEI CLOUD account does not need individual IAM users for permissions management, you may skip over this chapter.

IAM can be used free of charge. You pay only for the resources in your account. For more information about IAM, see the [IAM Service Overview](#).

Supported System Policies

A policy is a set of permissions defined in JSON format. By default, new IAM users do not have any permissions assigned. You need to add a user to one or more groups, and assign permissions policies to these groups. The user then inherits permissions from the groups it is a member of. This process is called authorization. After authorization, the user can perform specified operations on FunctionGraph based on the permissions. IAM provides system policies that define the common permissions for different services, such as administrator and read-only permissions. You can directly use these system policies to assign permissions.

FunctionGraph is a project-level service deployed in specific physical regions. Therefore, FunctionGraph permissions are assigned to users in specific regions (such as **CN North-Beijing1**) and only take effect for these regions. If you want the permissions to take effect for all regions, you need to assign the permissions to users in each region. When accessing FunctionGraph, the users need to switch to a region where they have been authorized to use the FunctionGraph service.

Table 4-1 lists all the system policies supported by FunctionGraph. Dependencies are policies on which a system policy depends to take effect. For example, some FunctionGraph policies are dependent on the policies of other services. When assigning FunctionGraph permissions to users, you need to also assign dependent policies for the FunctionGraph permissions to take effect.

Table 4-1 System policy summary

Policy Name	Description
FunctionGraph Administrator	Administrator permissions for FunctionGraph
FunctionGraph Invoker	Permissions for querying and invoking FunctionGraph functions
Tenant Administrator	Administrator permissions for all cloud services in the region where the policy is assigned
Server Administrator	Administrator permissions for all server services, including Elastic Cloud Server (ECS), Elastic Volume Service (EVS), and Virtual Private Cloud (VPC), in the region where the policy is assigned
Tenant Guest	Permissions for viewing all cloud services in the region where the policy is assigned

Table 4-2 lists the common operations supported by each system policy of FunctionGraph. Please choose proper system policies according to this table.

Table 4-2 Common operations supported by each system policy

Operation	Tenant Administrator	FunctionGraph Administrator	FunctionGraph Invoker	Tenant Guest
Creating functions	√	√	×	×
Querying functions	√	√	√	√
Modifying functions	√	√	×	×
Deleting functions	√	√	×	×
Invoking functions	√	√	√	×
Querying function logs	√	×	×	√
Viewing function metrics	√	×	×	√

Permissions for configuring triggers

OBS triggers: Can be configured by tenants or subusers granted permissions of the **OBS Administrator** policy in the **Global service** region.

- DMS triggers
 - a. Can be configured by tenants or subusers granted permissions of the **Tenant Guest** policy or a higher-level policy in a specific region.
 - b. To enable FunctionGraph to access DMS, an agency must be created by tenants or subusers granted permissions of the **Security Administrator** policy in the **Global service** region, and then specified for required functions.
- DIS triggers
 - a. Can be configured by tenants or subusers granted permissions of the **DIS Administrator** policy or a higher-level policy in a specific region.
 - b. To enable FunctionGraph to access DIS, an agency must be created by tenants or subusers granted permissions of the **Security Administrator** policy in the **Global service** region, and then specified for required functions.
- OBS triggers
Can be configured by tenants or subusers granted permissions of the **OBS Administrator** policy in the **Global service** region.
- SMN triggers
Can be configured by tenants or subusers granted permissions of the **SMN Administrator** policy or a higher-level policy in a specific region.
- APIG triggers
Can be configured by tenants or subusers granted permissions of the **Tenant Administrator** policy in a specific region.
- Timer triggers
Can be configured by tenants or subusers granted permissions of the **Tenant Administrator** policy in a specific region.

Helpful Links

- [IAM Service Overview](#)
- [Creating a User Group and User and Granting FunctionGraph Permissions](#)
- [Policy Syntax](#)

5 Related Services

Table 5-1 describes the HUAWEI CLOUD services that have been interconnected with FunctionGraph.

Table 5-1 Interconnected services

Service	Function
SMN	FunctionGraph functions are constructed to process SMN notifications. For details, see SMN User Guide .
DMS	FunctionGraph functions are configured to automatically poll DMS queues for messages and process any new messages. For details, see DMS User Guide .
API Gateway	FunctionGraph functions are invoked over HTTPS by defining REST APIs with specified backend services. For details, see API Gateway User Guide .
OBS	FunctionGraph functions are created to process OBS bucket events, such as object creation or deletion events. For example, when an image is uploaded to the specified bucket, OBS invokes the function to read the image and create a thumbnail. For details, see OBS User Guide .
DIS	FunctionGraph functions are created to periodically poll DIS streams for new records, such as website click streams, financial transactions, social media streams, IT logs, and location-tracking events. For details, see the DIS User Guide .
LTS	FunctionGraph functions are built to process logs subscribed to in LTS. When LTS collects subscribed logs, the function is triggered to process or analyze the logs or to load the logs to other systems. For details, see LTS User Guide .

Service	Function
CTS	<p>FunctionGraph functions are defined to analyze and process key information in logs according to the event notifications of specified service type and operations configured in CTS. For details, see CTS User Guide.</p> <ul style="list-style-type: none"> ● With CTS, you can record operations associated with FunctionGraph for later query, audit, and backtrack operations. Table 5-2 lists the FunctionGraph operations that can be recorded by CTS. ● CTS starts recording operations on cloud resources once being enabled. Operation records of the last 7 days can be viewed on the CTS console. For details on how to query operation records, see Querying Real-Time Traces.
Cloud Eye	<p>FunctionGraph is interconnected with Cloud Eye to report monitoring metrics, allowing you to view function metrics and alarm messages through Cloud Eye. For details, see Cloud Eye User Guide.</p>
VPC	<p>Functions can be configured to access resources in your Virtual Private Clouds (VPCs) or to access the Internet through source network address translation (SNAT) by binding elastic IP addresses. For details, see VPC User Guide.</p>
AOS	<p>FunctionGraph works with Application Orchestration Service (AOS) to deploy cloud resources on HUAWEI CLOUD with a few clicks, automating service processes. For details, see AOS User Guide.</p>

Table 5-2 FunctionGraph operations that can be recorded by CTS

Operation	Resource Type	Trace Name
Creating a function	Functions	CreateFunction
Deleting a function	Functions	DeleteFunction
Modifying function information	Functions	ModifyFunctionMetadata
Publishing a function version	FunctionVersions	PublishFunctionVersion
Deleting a function alias	FunctionVersionsAlias	DeleteVersionAlias
Deleting a function trigger	Trigger	DeleteTrigger
Creating a function trigger	Trigger	CreateTrigger
Disabling a function trigger	Trigger	DisabledTrigger
Enabling a function trigger	Trigger	enabledTrigger
Executing a function	FunctionInstance	invokeFunction

Operation	Resource Type	Trace Name
Creating a workflow	graph	createGraph
Deleting a workflow	graph	deleteGraph
Creating a workflow trigger	event	createEvent
Deleting a workflow trigger	event	deleteEvent
Updating a workflow trigger	event	updateEvent
Starting a workflow instance	execution	startExecution
Testing a workflow instance	execution	testExecution
Stopping a workflow instance	execution	stopExecution
Testing a workflow in response to event messages	event	eventTest

6 Feature Updates

Date	Description
2019-08-16	Added support for the following: <ul style="list-style-type: none">● Test event persistence● Custom runtimes● Mounting file systems to functions
2019-5-27	<ol style="list-style-type: none">1. Optimized the inline code editor by adding the functions of testing functions, viewing testing results, and creating multi-level Python modules.2. Added support for integrating function templates with related cloud services.3. Added support for PHP 7.3. Code can be edited inline in PHP 7.3.
2019-3-15	<ol style="list-style-type: none">1. Added support for binding a major version and an additional version to each alias.2. Added support for configuring functions to access resources in Virtual Private Clouds (VPCs) or to access the Internet through source network address translation (SNAT) by binding elastic IP addresses.3. Added the JSON serialization and deserialization interfaces in the C# SDK.
2019-3-4	<ol style="list-style-type: none">1. Added support for function initialization.2. Launched FunctionGraph API 2.0.
2019-1-26	<ol style="list-style-type: none">1. Combined the package and deploy commands for remote deployment using AOS.2. Added support for the following memory sizes: 1792 MB, 2048 MB, 2560 MB, 3072 MB, 3584, and 4096 MB.3. Optimized the implementation of function execution and configuration agencies.4. Added support for querying logs by function version.

Date	Description
2018-12-25	<ol style="list-style-type: none">1. Added support for automatically generating functions with Huawei Serverless Sandbox (HSS).2. Added support for running HSS in Windows.
2018-11-30	<ol style="list-style-type: none">1. Added support for importing and exporting function apps.2. Added support for specifying different agencies for function execution and trigger creation.
2018-11-16	<ol style="list-style-type: none">1. Added descriptions about FunctionGraph metrics monitored on Cloud Eye.2. Added the dependency management module for managing dependency packages.3. Added support for downloading function logs.
2018-11-2	<ol style="list-style-type: none">1. Added support for C# (.NET Core 2.0) and C# (.NET Core 2.1).2. Added the Tools page for downloading SDKs, CLI, and HSS and calculating monthly price based on estimated usage.
2018-10-18	<ol style="list-style-type: none">1. Connected to Cloud Eye to report metrics to it. You can view monitoring graphs and alarm messages on the Cloud Eye console.2. Added third-party components for Node.js, including smnsdk, express, fgs-express, and request.
2018-09-29	<ol style="list-style-type: none">1. Added support for function apps.2. Added support for editing multiple function code directories.
2018-08-29	<ol style="list-style-type: none">1. Added confirmation for deleting functions, workflows, triggers, versions, and aliases.2. Added support for disabling functions.3. Provided use cases to guide you through the whole process of creating serverless applications.4. Enhanced CLI to support the interaction mode.5. Enhanced HSS to support loading of cloud-based events.6. Added new function templates.

Date	Description
2018-07-28	<ol style="list-style-type: none">1. Added support for Node.js 8.10.2. Optimized the monitoring of invocation duration, and provides maximum, average, and minimum invocation durations.3. Added support for configuring a backend timeout for an APIG trigger.4. Integrated common third-party components for Node.js.5. Added support for CTS triggers. Functions will be triggered when specified operations are performed on cloud resources subscribed on CTS.6. Added support for exporting function code and configurations and importing functions.7. Added event structure definition to the Java SDK. Currently, DMS, DIS, SMN, LTS, timer, and APIG triggers are supported. The definition makes coding much simpler when triggers are required.
2018-07-13	<ol style="list-style-type: none">1. Added support for LTS triggers to connect FunctionGraph to LTS. FunctionGraph functions can be invoked to analyze logs subscribed on LTS. For details, see Using an LTS Trigger.2. Fixed known issues.
2018-06-30	<ol style="list-style-type: none">1. Merged the function and workflow products as well as their materials into FunctionGraph.2. Put FunctionGraph into commercial use.

Function Feature Updates

Table 6-1 Function feature updates

Version	Date	Description
1.0.25	2018-05-18	<ol style="list-style-type: none">1. Added support for timer triggers to execute functions as scheduled. For details, see Using a Timer Trigger.2. Resolved known issues.
1.0.20	2018-03-16	<ol style="list-style-type: none">1. Provided the Huawei Serverless Sandbox (HSS) tool to test and debug functions locally and to generate sample payload for different even sources. For details, see HSS User Guide.2. Improved service performance and reliability.

Version	Date	Description
1.0.18	2018-02-06	<ol style="list-style-type: none"> Added support for version definition. For details, see Version Management. Added support for querying function metrics by version. For details, see Viewing Function Metrics. Provided the command line interface (CLI) to manage functions, triggers, and aliases and to invoke functions. For details, see Introduction to CLI.
1.0.17	2018-01-24	<ol style="list-style-type: none"> Added support for configuring test events. For details, see Test Management. Provided trigger templates. For details, see Creating a Test Event.
1.0.12	2018-01-06	Added new features. For details, see Table 6-2 .
1.0.7	2017-11-15	<ol style="list-style-type: none"> Added the Dashboard page. Integrated the FunctionGraph menu in the navigation pane.

Table 6-2 New features in version 1.0.12

Category	Function	Description
Function	Function template	<ol style="list-style-type: none"> Provided the function template list. Added support for searching templates by runtime or template name, and displaying templates on different pages by type. Added support for automatically loading the code, triggers, and running environment of each template. For details, see Function Templates .
	Environment variable	Added support for configuring environment variables. For details, see Environment Variables .
Runtime	Dependency	Added support for configuring dependent libraries (uploaded from OBS). For details, see Dependency Management .
	Python	<ol style="list-style-type: none"> Added support for Python 3.6. Integrated common third-party Python libraries with cloud service SDKs. For details, see Dependency Management.
Trigger	APIG trigger	<ol style="list-style-type: none"> Interconnected FunctionGraph with API Gateway. Added support for APIG triggers. For details, see Using an APIG Trigger .

Category	Function	Description
Monitoring	Tenant-level metrics	Added support for viewing function metrics. For details, see Dashboard .
Permission	Sub-user permission control	Added support for creating user groups and sub-users, and granting FunctionGraph permissions to sub-users of specific user groups. For details, see User Permissions .
Frontend	Dashboard page	<ol style="list-style-type: none">Graphically displayed tenant resources.Graphically displayed tenant-level function metrics. For details, see Dashboard .

Workflow Feature Updates

Table 6-3 Workflow feature updates

Version	Date	Description
1.0.7	2018-03-06	<ul style="list-style-type: none">Removed the countdown from the workflow creation completion page.Simplified the display of colons behind field labels.Added support for creating a workflow without opening a new window.Added support for dragging state components to the visual workflow pane.Optimized the log list of workflow instances.Limited the header length to 32 KB.
1.0.5	2018-02-06	<ul style="list-style-type: none">Optimized the date and time display.Removed the countdown from the workflow creation completion page.Simplified the display of colons behind field labels.Optimized the switching between templates for different service scenarios.Upgraded the console framework.Improved backend performance and O&M capabilities.
1.0.4	2018-01-15	<ul style="list-style-type: none">Added support for timers to trigger workflows as schedule.For details, see Workflow Timer Management.

Version	Date	Description
1.0.3	2018-01-06	Changes to the Dashboard page: <ul style="list-style-type: none">● Graphically displayed tenant resources.● Graphically displayed tenant-level function metrics. For details, see Dashboard .
1.0.2	2017-12-29	<ul style="list-style-type: none">● Added support for modifying workflows.● Added support for managing the mapping relationships between workflows, events, and functions.● Added support for repeatedly executing workflows with triggers.● Added support for switching instance details in the code view and the visual workflow view.● Added support for creating and editing workflows in graphical mode, and displaying the workflow editing page in vertical or horizontal split mode.● Added zoom-in and zoom-out buttons and prompts on the visual workflow page.● Added support for creating a workflow by copying an existing one.● Displayed log information on the workflow test page.
1.0.1	2017-11-30	Added support for the OBS event source. Workflows can be triggered by OBS events, and do not need to be triggered by the SMN event source when OBS events generate SMN messages.
1.0.0	2017-11-15	This is the first official release.

7 Quotas and Usage Restrictions

Tenant Resource Restrictions

The following table provides the quotas for tenant resources. For details on how to query and modify quotas, see [Quotas](#).

Table 7-1 Tenant resource restrictions

Resource	Restriction
Maximum number of functions that a tenant can create	400
Maximum number of versions allowed for a function	10
Maximum number of aliases allowed for a function	10
Maximum number of DIS and DMS triggers allowed for a function version	10
Size of code edited inline for a function	10 KB
Size of a code deployment package (in ZIP or JAR format)	50 MB
Size of the original code in a deployment package	250 MB
Maximum size of deployment packages allowed for a tenant	10 GB
Number of concurrent executions per tenant	100
Number of environment variables allowed for a function	20 (2 KB)
Default workflow quota	512
Maximum workflow quota	1000

Resource	Restriction
Default trigger quota	200
Maximum trigger quota	10,000

Function Running Resource Restrictions

Table 7-2 Function running resource restrictions

Resource	Restriction
Ephemeral disk space (/tmp space)	512 MB
Number of file descriptors	1024
Total number of processes and threads	1024
Maximum execution duration per request	900s
Valid payload size of invocation request body (synchronous invocation)	6 MB
Valid payload size of invocation response body (synchronous invocation)	6 MB
Valid payload size of invocation request body (asynchronous invocation)	128 KB
Format and maximum size of imported resources	ZIP file not more than 50 MB
Size of exported resources	≤ 50 MB

Workflow Restrictions

Table 7-3 Workflow restrictions

Resource	Restriction
Workflow name	<p>Contains 1 to 64 characters except special characters, For example:</p> <ul style="list-style-type: none"> ● Space ● Brackets <>{}[] ● ? * ● Special characters "#%^\ ~`\$&,;:/ ● Control characters (U+0000 - 001F, U+007F-009F)
Workflow definition	Contains 1 to 65536 characters.

Resource	Restriction
Maximum number of states allowed for a workflow	50
Maximum number of actions allowed for a state	10

Workflow Instance Restrictions

Table 7-4 Workflow instance restrictions

Resource	Restriction
Maximum execution duration	1 year
Maximum duration for retaining execution history	90 days, after which you cannot retrieve or view execution records.
Maximum execution duration of all actions in a single state	20 minutes
Instance name	Contains 1 to 80 characters except special characters, For example: <ul style="list-style-type: none"> ● Space ● Brackets <> {} [] ● ? * ● Special characters "#%^\ ~`\$&,;:/ ● Control characters (U+0000 - 001F, U+007F-009F)
Input size and format of each state	1 - 32768 characters in JSON format
Maximum number of timers bound to a workflow	5

Workflow Trigger Restrictions

Table 7-5 Workflow trigger restrictions

Resource	Restriction
Use of internal triggers	A workflow cannot use cloud service triggers (such as OBS and SMN) and worker triggers at the same time.
Maximum number of EVENT states with a cloud service internal trigger	1

Resource	Restriction
Maximum number of cloud service internal triggers allowed for an EVENT state	1
Binding of external triggers	A workflow cannot be bound with external and internal triggers at the same time.