ALIBABA CLOUD

Alibaba Cloud

FunctionFlow

Product Introduction

Document Version: 20220511

Alibaba Cloud

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.

2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company or individual in any form or by any means without the prior written consent of Alibaba Cloud.

3. The content of this document may be changed because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.

4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides this document based on the "status quo", "being defective", and "existing functions" of its products and services. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, special, or punitive damages, including lost profits arising from the use or trust in this document (even if Alibaba Cloud has been notified of the possibility of such a loss).

5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.

6. Please directly contact Alibaba Cloud for any errors of this document.

Document conventions

Style	Description	Example
⚠ Danger	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	⚠ **Danger:** Resetting will result in the loss of user configuration data.
🔔 Warning	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	🔔 **Warning:** Restarting will cause business interruption. About 10 minutes are required to restart an instance.
📢 Notice	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	📢 **Notice:** If the weight is set to 0, the server no longer receives new requests.
? Note	A note indicates supplemental instructions, best practices, tips, and other content.	? **Note:** You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click **Settings**> **Network**> **Set network type**.
Bold	Bold formatting is used for buttons , menus, page names, and other UI elements.	Click **OK**.
`Courier font`	Courier font is used for commands	Run the `cd /d C:/window` command to enter the Windows system folder.
Italic	Italic formatting is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	This format is used for an optional value, where only one item can be selected.	`ipconfig [-all|-t]`
{} or {a|b}	This format is used for a required value, where only one item can be selected.	`switch {active|stand}`

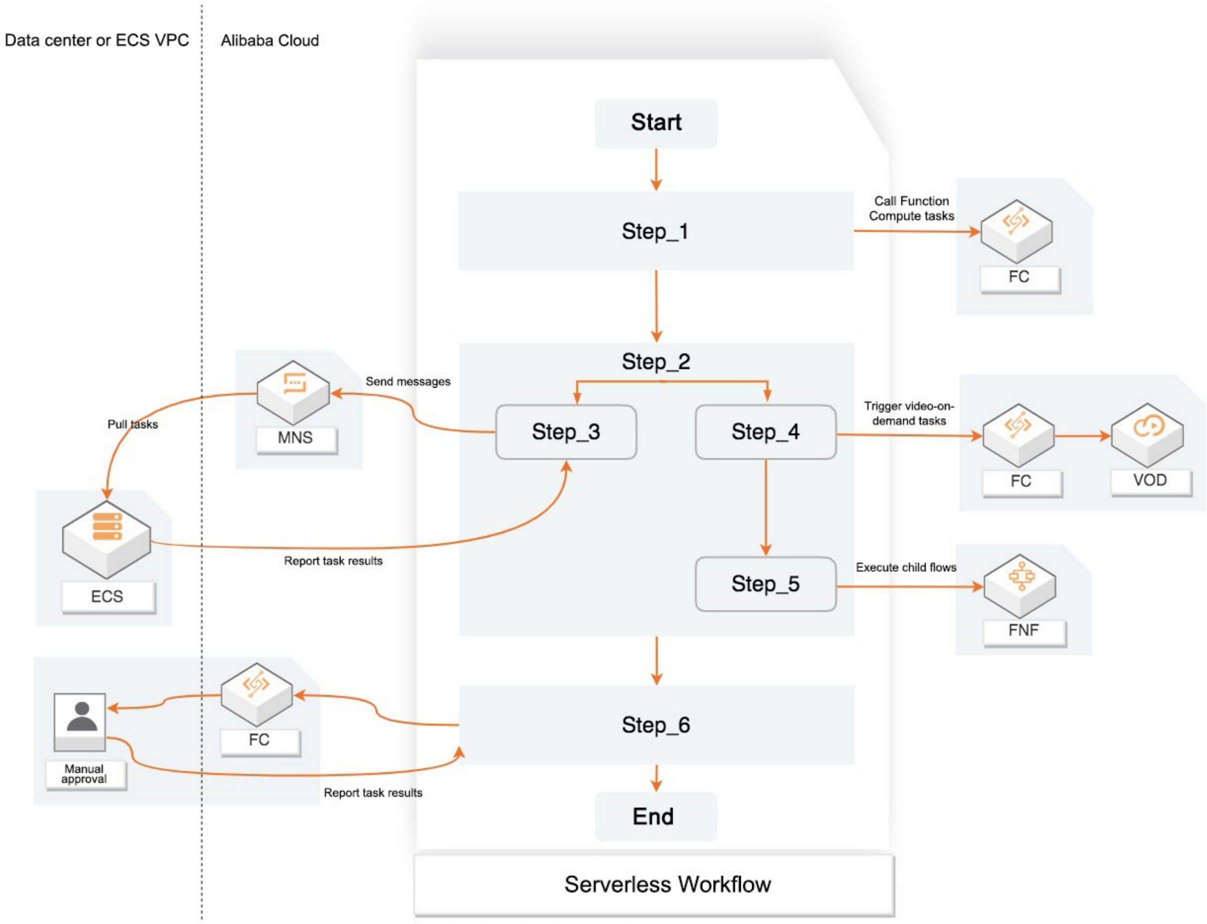
Table of Contents

1.What is Serverless Workflow?

is a fully managed cloud service that coordinates distributed tasks.

In , you can orchestrate distributed tasks in sequence, in branches, or in parallel. Based on the defined steps, reliably coordinates task executions, tracks the state transition of each task, and executes the user-defined retry logic when necessary to ensure the smooth completion of your flow. provides the logging and auditing features to monitor flow executions, allowing you to diagnose and debug applications. simplifies complex tasks such as task coordination, state management, and error handling required for application development and execution of business flows. In this way, developers can focus on business logic development.

The following figure illustrates how coordinates distributed tasks, which can be functions, integrated cloud service APIs, and programs that run on virtual machines (VMs) or containers.



Benefits

- Coordination of distributed components

  can orchestrate applications that are developed in different architectures, different networks, and different languages. It can also eliminate the gaps in the transition from Apsara Stack to the hybrid cloud (Apsara Stack and Alibaba Cloud) or Alibaba Cloud, or from a monolithic architecture to a microservices architecture.

- Less flow code

provides rich control logic such as sequence, choice, and parallelism, so that you can implement complex business logic with less code.

- Improved fault tolerance of applications

  manages flow states and provides built-in checkpoints and playback capabilities to ensure that your application is executed step by step as expected. Error retries and catch allow you to handle errors flexibly.

- Serverless

  When you use , you are charged based on the number of step transitions required to execute your application. After the execution, you will not be charged. supports automatic scaling and therefore frees you from managing hardware budgets and extensions.

Features

- Service orchestration capabilities

  can separate flow logic from task execution and save you time in writing and orchestrating code. For example, after you call a face recognition function to detect the face in an image, the image is cropped based on the face position, and a message is sent to notify the user at last. provides a serverless solution to reduce your orchestration and O&M costs.

- Coordination of distributed components

  can coordinate applications that are developed in different languages, different architectures, and different networks. serves as a coordinator when applications transit from Apsara Stack to the hybrid cloud (Apsara Stack and Alibaba Cloud) or Alibaba Cloud or evolve from a monolithic architecture to a microservices architecture.

- Built-in error handling

  By using built-in error retry and catch capabilities, you can automatically retry tasks that failed or timed out, respond differently to different types of errors, and define fallback logic.

- Visual monitoring

  provides visual interfaces for you to define flows and view the execution status. The state includes input and output. This allows you to identify and troubleshoot problems.

- Support for long-running flows

  can track the entire flow and allow long-time flow execution to ensure the completion of the flow. Some flows may take several hours, days, or even months, for example, O&M-related pipelines and email promotion flows.

- Flow state management

  manages all states in a flow execution, including tracking its execution steps and the data transfer between steps. You do not need to manage the flow states yourself or build complex state management into tasks.

2.Scenarios

coordinates distributed applications and microservices to build complex, multi-step, stateful, and long-running flows.

Transactional flow orchestration

In complex scenarios that involve order management, such as e-commerce websites, hotel booking, and flight reservations, applications need to access multiple remote services, and have high requirements for the operational semantics of transactions. In other words, all steps must succeed or fail without intermediate states. In applications with small traffic and centralized data storage, the atomicity, consistency, isolation, durability (ACID) properties of relational databases can guarantee that transactions are reliably processed. However, in large-traffic scenarios, distributed microservices are usually used for high availability and scalability. To guarantee reliable processing of multi-step transactions, the service providers usually need to introduce queues and persistent messages and display the flow status to the distributed architecture. This brings additional development and O&M costs. ensures reliable processing of distributed transactions in complex flows, and therefore helps users focus on their own business logic.

For more information about how to use Serverless workflow to orchestrate transactional flows, see Reliably process distributed multi-step transactions.

Multimedia file processing

helps you orchestrate multiple tasks, such as transcoding, frame capture, face recognition, voice recognition, and review and upload of multimedia files, into a complete flow. You can use Function Compute to submit an Intelligent Media Management (IMM) task or a user-created processor to generate an output that meets your business requirements. Tasks that encounter errors and exceptions can be reliably retried, which significantly improves the multimedia task processing throughput.

Genetic data processing

can orchestrate multiple distributed batch computing jobs in sequence or in parallel and reliably supports large-scale computing tasks that require long execution time and high concurrency. For example, in genetic data analysis, gene sequences are aligned, variation analysis is performed on all chromosomes in parallel, and finally all chromosome data is aggregated to produce the results. Based on specified dependencies, submits batch computing jobs with different CPU, memory, and bandwidth specifications to improve execution reliability and resource utilization and reduce costs.

Data pipelines

You can use to build highly available data pipelines. For example, measurement data from different data sources is collected into Log Service. A time-based trigger of Function Compute triggers each hour. uses Function Compute to process the measurement data of multiple shards in parallel and write the results back to Log Service. Then, all measurement data of the shards is aggregated and written into Tablestore. Finally, bills are generated for each user. allows you to retry a failed step in a flow to reduce the failure probability. supports dynamic parallel execution of tasks to achieve high scalability of data processing capabilities.

Automated O&M

Common challenges in automated O&M include cumbersome steps, varying lengths of execution time, low reliability of standalone scripts, complex dependencies, and inability to visualize the progress. The combination of and Function Compute can perfectly handle these challenges. For example, during automated software deployment, you need to build Docker containers, upload container images, start and track nodes, track images on all nodes, and start the containers with new images. Logs generated by functions in each step are stored in Log Service, so that you can query and use the logs. Compared with standalone O&M scripts, automated tools based on provides higher availability, built-in error handling, and graphical progress.

3.Limits

You must follow some limits on the use of . This topic introduces two types of limits that you must follow when you use : naming-related limits and account-related limits.

Naming-related limits

The flow and execution naming must comply with the following rules:

- A name can contain uppercase letters (A to Z), lowercase letters (a to z), digits (0 to 9), underscores (_), and hyphens (-).
- The name must start with an uppercase letter (A to Z), a lowercase letter (a to z), or an underscore (_).
- The name is case-sensitive.
- The name must be 1 to 128 characters in length.

Account-related limits

Resource limits

Item	Default value	Description
Maximum number of executions in progress *	10,000	Specifies the total number of executions that are in the running state under the account. If you need to increase the limit, .
Maximum step entry rate *	Initial number of tokens: 100. Recovery rate: 100/s	Specifies the total number of step entries per second under the account. If you need to increase the limit, .
Maximum number of flows that can be created	10,000	Specifies the sum of the number of flows that can be created under the account.

Limits on API call rates

Operation	Maximum number of tokens	Token recovery speed (tokens/s)	Description
StartExecution *	100	100	If you need to increase the limit, .
DescribeExecution	50	5	N/A
GetExecutionHisotry	50	5	N/A
Other operations	10	2	N/A

Limits on flows, executions, and historical events

Item	Dimension	Default value	Description
Maximum number of events per execution	Execution	5,000	If you need to increase the limit, .
Data size of the biggest event	Event	32 KiB	N/A
Maximum request size	Request	1024 KiB	N/A

4.Terms

This topic describes and defines terms used in Serverless Workflow to help you better understand and use Serverless Workflow.

Serverless Workflow

A fully managed serverless cloud service that coordinates the execution of multiple distributed tasks. Serverless Workflow allows you to orchestrate distributed tasks in sequence, branch, or parallel. This way, Serverless Workflow can coordinate multiple tasks based on the preset sequence.

distributed task

A distributed task in Serverless Workflow can be a function, an API integrated with a cloud service, and a program that runs on a virtual machine or a container.

flow

A series of sequential steps, which define the business logic and contain all the information of the sequential steps.

step

The work unit in a flow. A step can be a simple atomic step, such as task, succeed, fail, wait, or pass. A step can also be a complex control step, such as choice, parallel, or foreach. Multiple steps can be used together to build complex business logic.

parent step

If Step A contains Step B, Step A is the parent step of Step B.

child step

If Step A contains Step B, Step B is a child step of Step A.

task step

One of the step types, which is used to define the information about function invocation of Function Compute. When a task step is executed, functions are invoked based on this step.

pass step

One of the step types, which is used to show a constant and convert the input to the expected output. Pass steps can be used to debug the flow logic of functions for which no task steps are created.

wait step

One of the step types, which is used to suspend the execution of a flow. You can select a relative duration or specify an absolute end time in the timestamp format for a wait step.

choice step

One of the step types, which allows Serverless Workflow to execute different steps based on selected conditions.

parallel step

One of step types, which allows Serverless Workflow to execute multiple different steps in parallel.

foreach step

One of the step types, which allows Serverless Workflow to execute multiple identical steps in parallel.

succeed step

One of the step types, which is used to end a series of sequential steps in advance. A succeed step is typically used together with a choice step. When the conditions of a choice step are met, the flow goes to a succeed step, and no other steps are executed.

fail step

One of the step types, which is used to end a series of sequential steps in advance. After a fail step is executed, the steps following the fail step are not executed, which causes the parent step of the fail step to fail. Then, the steps following the failed parent step are not executed, which causes the parent step of this failed parent step to fail. This knock-on effect continues until the flow execution fails.

Flow Definition Language (FDL)

The language that is used to describe and define business logic. When a flow is executed, Serverless Workflow executes steps based on the flow definition.

time-based schedule

A schedule that is performed by Serverless Workflow at a specified interval.