Understand Workflows

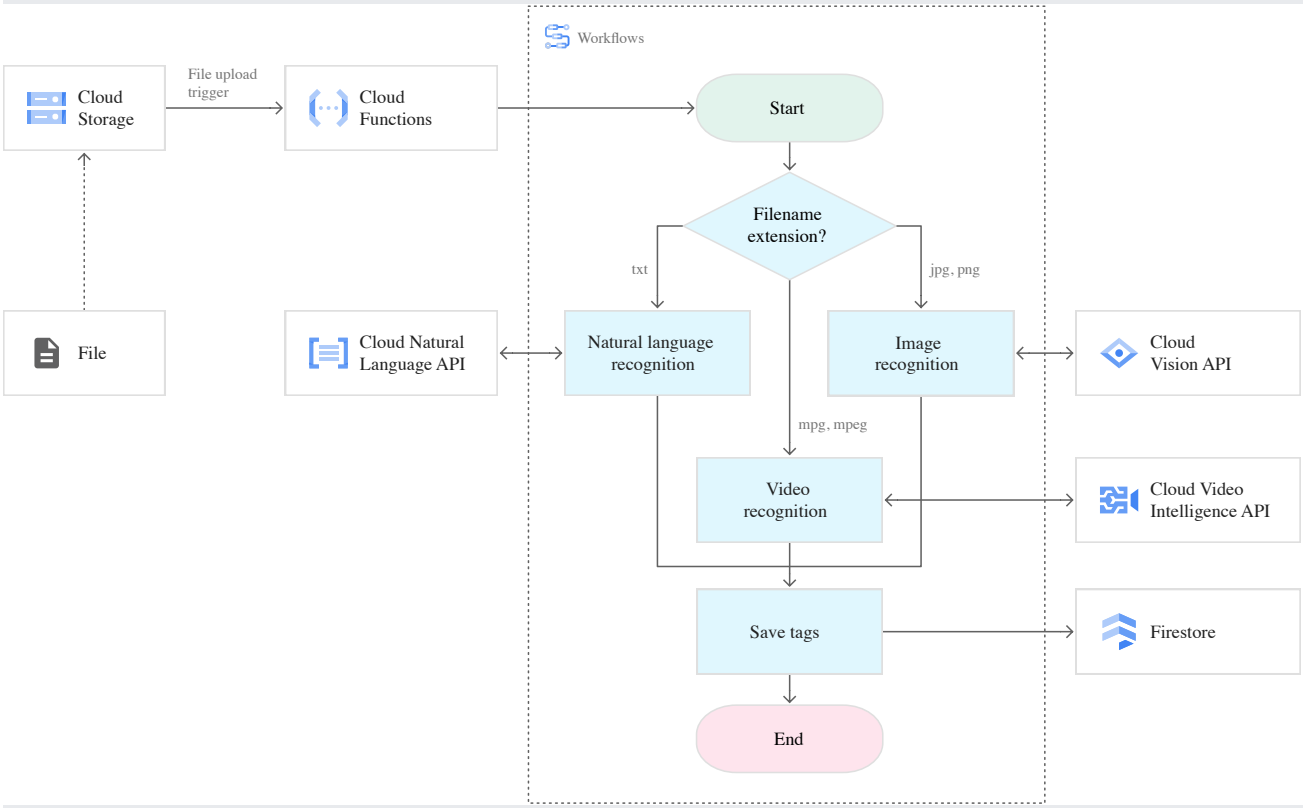
Workflows is a fully-managed orchestration platform that executes services in an order that you define: a *workflow*. These workflows can combine services including custom services hosted on Cloud Run or Cloud Functions, Google Cloud services such as Cloud Vision AI and BigQuery, and any HTTP-based API.

By incorporating Workflows into solutions, you can make service dependencies explicit and observable end-to-end. A workflow that specifies an application, operational, or business process provides a source-of-truth or canonical narrative for the process.

Workflows is serverless, scaling up as needed, and no charges are incurred while idle. Because a workflow contains no code or library dependencies, it does not require security patches. Once you deploy a workflow, you can expect it to reliably execute without maintenance. A workflow can hold state, retry, poll, or wait for up to a year.

Workflows is compliant with these certifications and standards (/workflows/compliance).

The following diagram shows an example of using Workflows to orchestrate services:



Key use cases

Workflows supports many use cases. Some examples are:

Service orchestration	*Create solutions by combining services*—Perform a sequence of operations across multiple systems, waiting for all operations to complete. Can be event-driven. For example: • Send newly uploaded files to Cloud Vision AI, then write tags into Firestore • Call a Cloud Function and send the results to a Cloud Run service
Batch jobs	*Operate on multiple items*—Perform operations on a set of items or batch data. Often scheduled. For example: • Send daily customer emails • Prepare and run BigQuery or machine learning jobs • Generate reports
Business processes	*Automate line-of-business workflows*—Encode the steps in a business process, including conditions, actions, and human-in-the-loop events. For example: • Track an order from request to fulfillment • Automate resource requests with approvals
IT process automation	*Managed execution of service operations*—Easily script sequences of Google Cloud service operations. For example: • Provision new tenant projects or infrastructure • Turn down resources on a schedule or through event triggers

Core concepts

A workflow consists of a series of steps described using the Workflows syntax, and can be written in either YAML or JSON. This is the workflow's **definition**. For a detailed explanation of the Workflows syntax, see the Syntax reference (/workflows/docs/reference/syntax).

After a workflow is created, it is **deployed**, which makes the workflow ready for execution. Learn how to create and update a workflow (/workflows/docs/creating-updating-workflow) in the Google Cloud console or by using the Google Cloud CLI.

An **execution** is a single run of the logic contained in a workflow's definition. A workflow that hasn't been executed generates no charges. All workflow executions are independent, and the product's rapid scaling allows for a high number of concurrent executions. You can

execute a workflow using the client libraries, in the Google Cloud console, using the Google Cloud CLI, or by sending a request to the Workflows REST API. For details, see Execute a workflow (/workflows/docs/executing-workflow).

Key capabilities

Execution control

Steps

To create a workflow, you define the desired steps and order of execution using the Workflows syntax. Every workflow must have at least one step. By default, Workflows treats steps as if they are in an ordered list and executes them one at a time until all the steps have run. For details, see Steps (/workflows/docs/reference/syntax/steps).

Conditions

You can use a `switch` block as a selection mechanism that allows the value of an expression to control the flow of a workflow's execution. For details, see Conditions (/workflows/docs/reference/syntax/conditions).

Iteration

You can use a `for` loop to iterate over a sequence of numbers or through a collection of data, such as a list or map. For details, see Iteration (/workflows/docs/reference/syntax/iteration).

Subworkflows

A subworkflow works similarly to a routine or function in a programming language, allowing you to encapsulate a step or set of steps that your workflow will repeat multiple times. For details, see Subworkflows (/workflows/docs/reference/syntax/subworkflows).

Triggering executions

Manual

You can manage workflows from either the Google Cloud console or from the command line using the Google Cloud CLI. Visualization support while editing the Workflows syntax is also available through the Google Cloud console.

Programmatic

The Cloud Client Libraries for the Workflows API, or the REST API, can be used to manage workflows. For details, see Workflows APIs and reference (/workflows/docs/apis).

Scheduled

You can use Cloud Scheduler to run a workflow on a particular schedule, such as every Monday at 9 AM or every 15 minutes. For details, see Schedule a workflow using Cloud Scheduler (/workflows/docs/schedule-workflow).

Runtime arguments

Data passed at runtime can be accessed by adding a `params` field to your main workflow (placed in a `main` block). The `main` block accepts a single argument that is any valid JSON data type. The `params` field names the variable that the workflow uses to store the data you pass in. For details, see Runtime arguments (/workflows/docs/reference/syntax/runtime-args).

Connecting services

When to call a service

How do you know when to create steps in YAML or JSON using the Workflows syntax or when to create a service—for example, a Cloud Run service or a Cloud Function—to do the work instead?

Use Workflows to call services from the workflow itself and handle the results, and to execute simple tasks like making an HTTP call. Workflows can invoke services (/workflows/docs/calling-run-functions), parse responses, and construct inputs for other connected services. Calling a service allows you to avoid the complications of extra invocations, additional dependencies, and services calling services.

Create services to do any work that is too complex for Workflows; for example, implementing reusable business logic, complex computations, or transformations that are not supported by Workflows expressions and its standard library. A complicated case is typically easier to implement in code, instead of using YAML or JSON and the Workflows syntax.

HTTP APIs

You can define a workflow step that makes an HTTP call and assign the response from the call to a variable. For example, you can invoke a Google Cloud service such

as Cloud Functions or Cloud Run through an HTTP request. Both HTTP and HTTPS requests are supported. For details, see Make an HTTP request (/workflows/docs/http-requests) and Invoke Cloud Functions or Cloud Run (/workflows/docs/calling-run-functions).

You can also invoke private on-premises, Compute Engine, Google Kubernetes Engine (GKE), or other Google Cloud endpoints. For details, see Invoke private on-prem, Compute Engine, GKE, or other endpoint (/workflows/docs/call-private-endpoints).

Connectors

Workflows publishes connectors that can be used to connect to other Google Cloud APIs within a workflow, and to integrate your workflows with those Google Cloud products. They simplify calling services because they handle the formatting of requests for you, and provide methods and arguments so that you don't need to know the details of a Google Cloud API. For details, see Understand connectors (/workflows/docs/connectors).

Standard library and environment variables

The Workflows standard library and built-in environment variables allow you to easily construct arguments for services and process responses.

The standard library includes modules and frequently used functions, such as for data type and format conversions. There is no need to import or load libraries in a workflow—library functions work out of the box. For details, see the Standard library overview (/workflows/docs/reference/stdlib/overview).

You can access a workflow's environment information (such as its location or project identifier) using built-in environment variables. Built-in environment variables require no declaration and are available in every workflow execution. For details, see Built-in environment variables (/workflows/docs/reference/environment-variables).

Error handling

You can make your workflows resilient and customize their behavior when a failure occurs by using Workflows' exception handling, including automated HTTP call retries with exponential back-offs, custom error handlers, and other advanced features. For details, see Workflow errors (/workflows/docs/reference/syntax/error-types).

Waiting

Callbacks allow workflow executions to wait for another service to make a request to the callback endpoint; that request resumes the execution of the workflow. With callbacks, you can signal to your workflow that a specified event has occurred, and wait on that event without polling. For details, see Wait using callbacks (/workflows/docs/creating-callback-endpoints).

You can pause the execution of a workflow by adding a sleep step to your workflow's definition. You can then use `sys.sleep` to poll for data over a given interval. For details, see Wait using polling (/workflows/docs/sleeping).

Authentication and access control

Since every workflow execution requires an authenticated call, you can mitigate the risk of accidental or malicious calls by using Workflows. You can also simplify interactions with other Google Cloud APIs by using IAM-based service accounts (/docs/authentication/production), and you can securely store keys and passwords for authenticating to external APIs by using Secret Manager (/secret-manager/docs). For details, see Authentication (/workflows/docs/authentication) and Access control (/workflows/docs/access-control).

Observability

Workflows automatically generates execution logs for workflow executions in Cloud Logging. You can also control when logs are sent to Logging during a workflow execution through call logging or custom logs. For details, see Send logs to Cloud Logging (/workflows/docs/log-workflow).

Audit logging information is available through Cloud Audit Logs. For details, see Workflows audit logging information (/workflows/docs/audit-logging).

Code samples

You can find many useful Workflows code samples on the samples page (/workflows/docs/samples).

What's next

- Get started using Workflows from the Google Cloud console (/workflows/docs/create-workflow-console) or using the gcloud CLI (/workflows/docs/create-workflow-gcloud).
- Learn more about creating and updating workflows (/workflows/docs/creating-updating-workflow).
- Learn how to control the order in which a workflow's steps are run (/workflows/docs/controlling-execution-order).

Last updated 2022-09-30 UTC.