Cloud Functions Overview

cloud.google.com/functions/docs/concepts/overview

What are Google Cloud Functions?

Google Cloud Functions is a serverless execution environment for building and connecting cloud services. With Cloud Functions you write simple, single-purpose functions that are attached to events emitted from your cloud infrastructure and services. Your function is triggered when an event being watched is fired. Your code executes in a fully managed environment. There is no need to provision any infrastructure or worry about managing any servers.

Cloud Functions can be written using JavaScript, Python 3, Go, or Java runtimes on Google Cloud Platform. You can take your function and run it in any standard Node.js (Node.js 10 or 12), Python 3 (Python 3.7 or 3.8), Go (Go 1.11 or 1.13) or Java (Java 11) environment, which makes both portability and local testing a breeze.

Connect and extend cloud services

Cloud Functions provides a connective layer of logic that lets you write code to connect and extend cloud services. Listen and respond to a file upload to Cloud Storage, a log change, or an incoming message on a Pub/Sub topic. Cloud Functions augments existing cloud services and allows you to address an increasing number of use cases with arbitrary programming logic. Cloud Functions have access to the Google Service Account credential and are thus seamlessly authenticated with the majority of Google Cloud services, including Cloud Vision, as well as many others. In addition, Cloud Functions are supported by numerous Google Cloud client libraries, which further simplify these integrations.

Events and triggers

Cloud events are *things* that happen in your cloud environment. These might be things like changes to data in a database, files added to a storage system, or a new virtual machine instance being created.

Events occur whether or not you choose to respond to them. You create a response to an event with a *trigger*. A trigger is a declaration that you are interested in a certain event or set of events. Binding a function to a trigger allows you to capture and act on events. For more information on creating triggers and associating them with your functions, see Events and Triggers.

Serverless

Cloud Functions removes the work of managing servers, configuring software, updating frameworks, and patching operating systems. The software and infrastructure are fully managed by Google so that you just add code. Furthermore, provisioning of resources happens automatically in response to events. This means that a function can scale from a few invocations a day to many millions of invocations without any work from you.

Use cases

Asynchronous workloads like lightweight ETL, or cloud automations such as triggering application builds now no longer need their own server and a developer to wire it up. You simply deploy a function bound to the event you want and you're done.

The fine-grained, on-demand nature of Cloud Functions also makes it a perfect candidate for lightweight APIs and webhooks. In addition, the automatic provisioning of HTTP endpoints when you deploy an HTTP function means there is no complicated configuration required as there is with some other services. See the following table for additional common Cloud Functions use cases:

Use case	Description
Data processing / ETL	Listen and respond to Cloud Storage events such as when a file is created, changed, or removed. Process images, perform video transcoding, validate and transform data, and invoke any service on the internet from your Cloud Functions.
Webhooks	Via a simple HTTP trigger, respond to events originating from 3rd party systems like GitHub, Slack, Stripe, or from anywhere that can send HTTP requests.
Lightweight APIs	Compose applications from lightweight, loosely coupled bits of logic that are quick to build and that scale instantly. Your functions can be event-driven or invoked directly over HTTP/S.
Mobile backend	Use Google's mobile platform for app developers, Firebase, and write your mobile backend in Cloud Functions. Listen and respond to events from Firebase Analytics, Realtime Database, Authentication, and Storage.
IoT	Imagine tens or hundreds of thousands of devices streaming data into Pub/Sub, thereby launching Cloud Functions to process, transform and store data. Cloud Functions lets you do it in a way that's completely serverless.

What's next

Was this helpful