

GOOGLE CLOUD FUNCTIONS: AN INTRODUCTION



The buzz is booming around [serverless functions](#), also known as cloud functions. A sure sign of this? The big players, like Google and AWS, have entered the arena.

Google has introduced its own serverless function on GCP. Since Google joined the [Functions as a Service \(FaaS\)](#) world, the new serverless execution option offers many benefits. Let's take a look.

What are Google Cloud Functions?

The execution environment of Google Cloud Functions enables serverless programming on top of [Google Cloud Platforms](#). The introduction of Google Cloud Functions, which are scalable, pay-as-you-go FaaS, offers many options in a simpler environment, making it easier for developers to run and scale your code in the cloud. The functions can be coded in Python, node.js, and Go—so developers likely won't need to learn anything new to use them.

Google Cloud Functions easily solves the many worries you may traditionally experience, like:

- Multiple services needing extra managing
- Predicting traffic to each service
- Risk of service going down during a spike or load
- Overpaying for unused capacity

You may be thinking "Why is Google Cloud Functions important?" Well, there are a number of key features we will explain, [below](#), that makes using Google Cloud easier, faster, and seamless of the

developer.

Types of Google Cloud Functions

There are two types of Google Cloud Functions:

- HTTP functions
- Background functions

HTTP Functions

The HTTP functions are primarily used when you want to invoke your function via an HTTP(S) request. This type of function can be used for webhooks and creating APIs. The Cloud Function abstracts away the information infrastructure, so you do not need much code.

Executing your Google Cloud Function has to signal your function in order to actually run. This is known as a trigger. The [HTTP](#) trigger comes from a request sent to your function or can be an event from Google Pub/Sub events, stack driver, or a Firebase Event.

The following command shows how to call the function and pass it as a parameter using curl:

```
curl -X POST HTTP_TRIGGER_ENDPOINT -H "Content-Type:application/json" '{"name":"Jane"}'where  
HTTP_TRIGGER_ENDPOINT(https://YOUR_REGION-YOUR_PROJECT_ID.cloudfunctions.net/  
FUNCTION_NAME)
```

[\(Source\)](#)

Background Functions

Background functions are triggered when specific cloud events occur. [For example](#), when a file is uploaded into the cloud, a function can automatically run to process the upload. These background functions currently respond to:

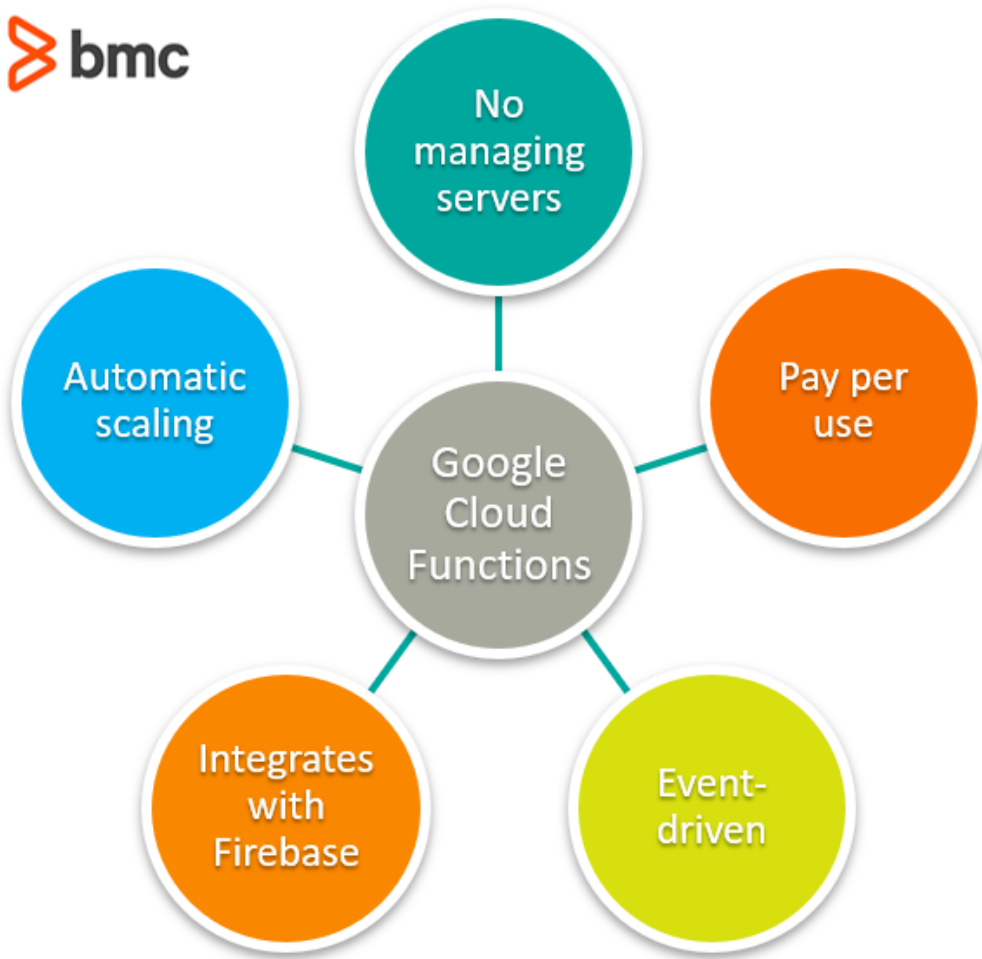
- Pub/Sub events
- Cloud storage
- Firebase events

This creates multiple use cases such as:

- Transforming data
- Processing IoT messages
- Watching project logs and reacting to specified changes within the infrastructure

Benefits of Google Cloud Functions

Google Cloud Functions make the experience simple and intuitive for the developer. Simply write your code and Google Cloud can handle the rest of the operational infrastructure. Here are the unique key features of Cloud Functions that make it so beneficial for many use cases:



No server management

First off, it's completely serverless, meaning that you can simply deploy your code and Google will automatically scale it for you. Google Cloud Functions has no server management which makes the overall architecture for a developer more flexible, in addition, a quicker time to release.

Pay per use

Another benefit of Google Cloud Function is you only pay for what you use. As a user, you pay nothing when your function is idle. Your function's execution time is only billed metered to the nearest 100 milliseconds. The [pricing breakdown](#) of Google Cloud Functions is based on

- The length of time your function runs
- The number of times the function is invoked
- The number of resources you provision for your functions.

This function is especially beneficial if you offer a service that has unpredictable traffic patterns—you can optimize your spending.

Event-driven architecture

Cloud functions are event-driven, which means they are executed when cloud events occur. An event-driven architecture allows for greater operational efficiency, creating a seamless, cost-effective operation. This type of architecture leads to forecasting functionality which gives real-time analytics so you can identify patterns and forecast and prepare for future responses. Developers

especially value the increased versatility of this event-driven system.

Automatic scaling

GCP automatically scales servers up or down as needed—no need for your developers to try to manage requests.

Integrated services

Another great benefit of Google Cloud function being event-driven is that it is seamlessly compatible with the [Firebase platform](#) in addition to GCP and Google Assistance. Firebase is Google's version of a mobile development platform that works to empower users to quickly craft and grow an app. Firebase and Google Cloud Platform share common integrations, including products, projects, billing, access control, terms of service, and accounts for streamlined usability.

More features

The simplicity of Google Cloud includes many other great features. There is also no need to provision virtual machines, no need to install new operating systems, or OS security patches.

Logging & monitoring

In a serverless world, it is vital to know how your app is performing. The cloud function feeds its logs into stack driver [logging](#) and sends telemetry data, like memory and execution time, into stack driver monitoring. This will allow you to review how your application is doing using familiar diagnostic tools across platforms.

Start building functions

On the Google Cloud Platform, you can create functions through the cloud console web interface. You can deploy functions from a gear depository or package them from your local development workstation. Google offers an open-source emulator that allows you to develop and run functions locally. It even connects to many popular IDEs to help prevent bugs in the system. Once you start building, you naturally start implementing other native cloud tools like:

- Big data
- Storage cues
- Scalable file storage
- Machine learning

Google Cloud Functions makes it possible to make your infrastructure invisible. This means developers can focus on more important parts of development—resulting in incredible apps shipping faster and more simply than ever. To get started, take advantage of [\\$300 in free credits](#) when you sign up with GCP.

Additional resources

For more on this topic and GCP, check out the [BMC DevOps Blog](#) and the [BMC Multi-Cloud Blog](#) or

explore these articles:

- [Availability Regions and Zones for AWS, Azure & GCP](#)
- [Google Cloud GPU: The Benefits of Using GPUs in the Cloud](#)
- [What Is Google Cloud Run?](#)
- [Introduction to Google Cloud Machine Learning Engine](#)
- [What Is AWS Lambda?](#)