

Cloud Functions

Product Overview

Documentation

Quickstart

How-to Guides

All How-to Guides

Writing Cloud Functions

Deploying Cloud Functions

Calling Cloud Functions

Monitoring Cloud Functions

Local Emulator

APIs & Reference

All APIs & References

Command Reference

REST Reference

RPC Reference

Concepts

All Concepts

Overview of Cloud Functions

Events and Triggers

Quickstart

★ Beta

This is a Beta release of Google Cloud Functions. This API might be changed in backward-incompatible ways and is not subject to any SLA or deprecation policy.

This page shows you how to create and deploy a Cloud Function using the `gcloud` command-line tool.

Before you begin

1. Sign in to your Google account.
- If you don't already have one, [sign up for a new account](#).
2. Select or create a Cloud Platform project.
- GO TO THE PROJECTS PAGE
3. Enable billing for your project.
- ENABLE BILLING
4. Enable the Cloud Functions API.
- ENABLE THE API
5. Install and initialize the Cloud SDK.
6. Update and install `gcloud` components:
- gcloud components update &&
gcloud components install beta
7. Prepare your environment for Node.js development.
- GO TO THE SETUP GUIDE

Create a function

1. Create a directory on your local system for the function code:
- LINUX OR MAC OS X

WINDOWS

Create the directory:

mkdir ~/gcf_hello_world

Move into the directory:

cd ~/gcf_hello_world
2. Create an `index.js` file in the `gcf_hello_world` directory with the following contents:
- NODE.JS

VIEW ON GITHUB

FEEDBACK

/**
 * Cloud Function.
 *
 * @param {object} event The Cloud Functions event.
 * @param {function} The callback function.
 */
exports.helloWorld = function helloWorld (event, callback) {
 console.log(`My Cloud Function: \${event.data.message}`);
 callback();
};
- This is a simple function named `helloWorld` that writes a message to the Cloud Functions logs.

Create a Cloud Storage bucket

- Create a new Cloud Storage bucket for your function:
- gsutil mb -p [PROJECT_ID] gs://[BUCKET_NAME]
- where:
- [PROJECT_ID] is the ID of your project
 - [BUCKET_NAME] is a globally-unique bucket name.

Deploy a function

1. Deploy the function with a Pub/Sub topic named `hello_world` :
- gcloud beta functions deploy helloWorld --stage-bucket [BUCKET_NAME] --trigger-topic hello_world
- It might take a few minutes to deploy the function the first time.
- Note:** [BUCKET_NAME] is the name of the bucket without the `gs://` protocol identifier before the name.
2. Verify the status of the function:
- gcloud beta functions describe helloWorld
- A **READY** status indicates that the function has been deployed:
- status: READY
triggers:
- pubsubTopic: projects/[PROJECT_ID]/topics/hello_world

Test the function

- Test the function from the command line:
- LINUX OR MAC OS X

WINDOWS

gcloud beta functions call helloWorld --data '{"message":"Hello World!"}'
- The `gcloud` tool returns the execution ID for the function, which looks something like this:
- executionId: eTe2m2-RVmJU-0

View logs

- Check the logs to see your actions in the log history:
- gcloud beta functions logs read helloWorld
- If the function executed successfully, messages in the log appear as follows:
- | LEVEL | NAME | EXECUTION_ID | TIME_UTC | LOG |
|-------|------------|----------------|-------------------------|---|
| D | helloWorld | - | 2016-03-17 20:35:41.043 | User function helloWorld loaded |
| D | helloWorld | eTe2m2-RVmJU-0 | 2016-03-17 20:45:18.334 | User function triggered, starting executi |
| I | helloWorld | eTe2m2-RVmJU-0 | 2016-03-17 20:45:18.335 | My Cloud Function: Hello World! |
| D | helloWorld | eTe2m2-RVmJU-0 | 2016-03-17 20:45:18.335 | Execution took 2 ms, user function comple |

What's next

- Learn more about writing Cloud Functions.
- Learn more about deploying Cloud Functions.
- Learn more about calling Cloud Functions.
- Learn more about monitoring Cloud Functions.
- Learn more about running Cloud Functions locally.
- Explore the more advanced Cloud Functions tutorials.

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](#), and code samples are licensed under the [Apache 2.0 License](#). For details, see our [Site Policies](#). Java is a registered trademark of Oracle and/or its affiliates.

Zuletzt aktualisiert am Mai 16, 2017