

Cloud Functions: Qwik Start - Console

GSP081



Overview

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 Cloud 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 are written in Javascript and execute in a Node.js environment on Google Cloud. You can take your Cloud Function and run it in any standard Node.js runtime 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 Cloud 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 such as Datastore, Cloud Spanner, Cloud Translation API, Cloud Vision API, as well as many others. In addition, Cloud Functions are supported by numerous [Node.js 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 like triggering application builds now no longer need their own server and a developer to wire it up. You simply deploy a Cloud 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 Function.
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 Cloud Pub/Sub, thereby launching Cloud Functions to process, transform and store data. Cloud Functions lets you do in a way that's completely serverless.

This hands-on lab shows you how to create, deploy, and test a cloud function using the Google Cloud console.

What you'll do

- Create a cloud function
- Deploy and test the function
- View logs

Setup

Before you click the Start Lab button

Read these instructions. Labs are timed and you cannot pause them. The timer, which starts when you click **Start Lab**, shows how long Google Cloud resources will be made available to you.

This Qwiklabs hands-on lab lets you do the lab activities yourself in a real cloud environment, not in a simulation or demo environment. It does so by giving you new, temporary credentials that you use to sign in and access Google Cloud for the duration of the lab.

What you need

To complete this lab, you need:

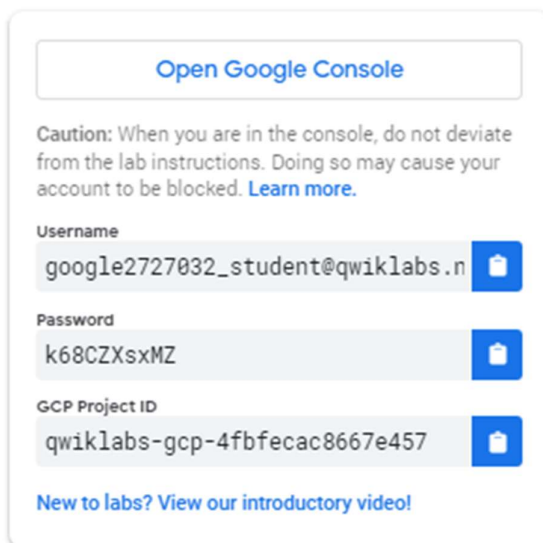
- Access to a standard internet browser (Chrome browser recommended).
- Time to complete the lab.

Note: If you already have your own personal Google Cloud account or project, do not use it for this lab.

Note: If you are using a Pixelbook, open an Incognito window to run this lab.

How to start your lab and sign in to the Google Cloud Console

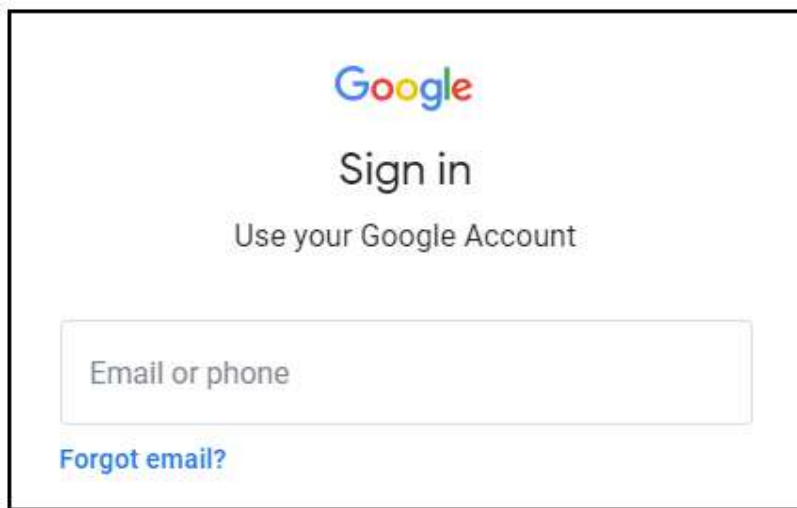
1. Click the **Start Lab** button. If you need to pay for the lab, a pop-up opens for you to select your payment method. On the left is a panel populated with the temporary credentials that you must use for this lab.



The screenshot shows a sign-in panel with the following elements:

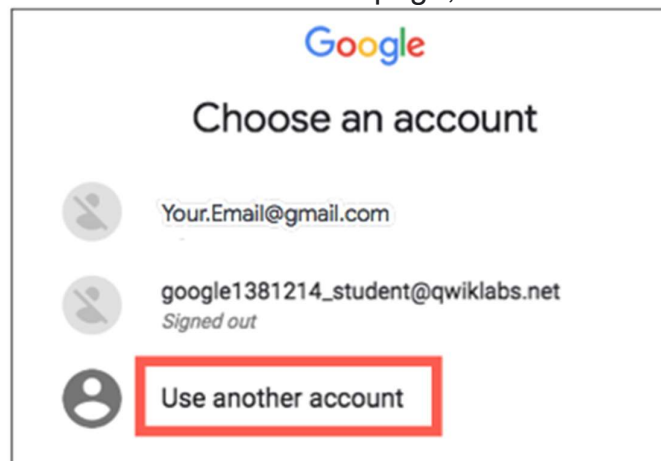
- A button at the top labeled "Open Google Console".
- A caution message: "Caution: When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked. [Learn more.](#)"
- Three input fields, each with a copy icon to its right:
 - Username:** google2727032_student@qwiklabs.n
 - Password:** k68CZXsxMZ
 - GCP Project ID:** qwiklabs-gcp-4fbfecac8667e457
- A link at the bottom: "New to labs? [View our introductory video!](#)"

2. Copy the username, and then click **Open Google Console**. The lab spins up resources, and then opens another tab that shows the **Sign in** page.



Tip: Open the tabs in separate windows, side-by-side.

If you see the **Choose an account** page, click **Use Another**



Account.

3. In the **Sign in** page, paste the username that you copied from the Connection Details panel. Then copy and paste the password.

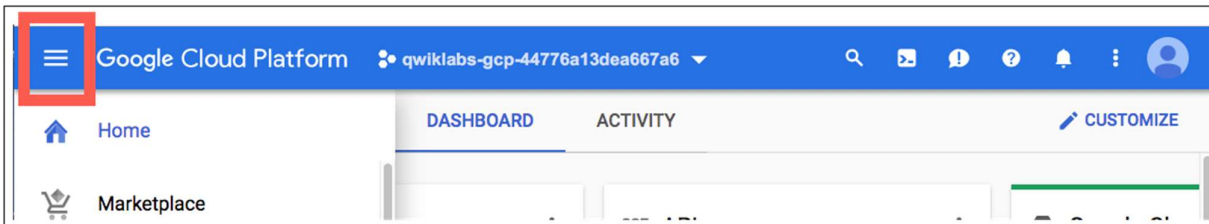
Important: You must use the credentials from the Connection Details panel. Do not use your Qwiklabs credentials. If you have your own Google Cloud account, do not use it for this lab (avoids incurring charges).

4. Click through the subsequent pages:

- Accept the terms and conditions.
- Do not add recovery options or two-factor authentication (because this is a temporary account).
- Do not sign up for free trials.

After a few moments, the Cloud Console opens in this tab.

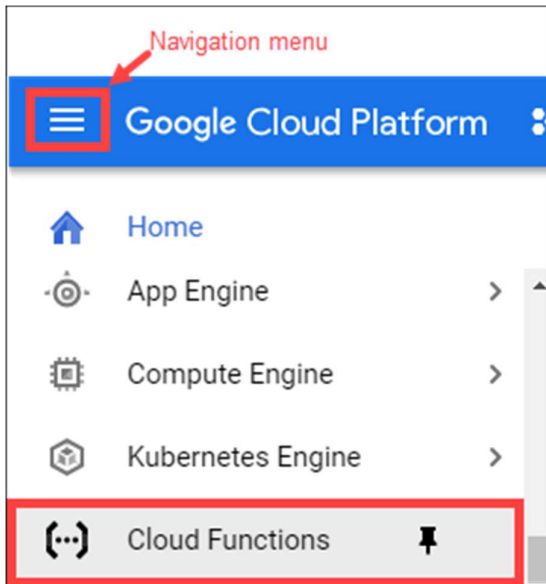
Note: You can view the menu with a list of Google Cloud Products and Services by clicking the **Navigation menu** at the top-left.



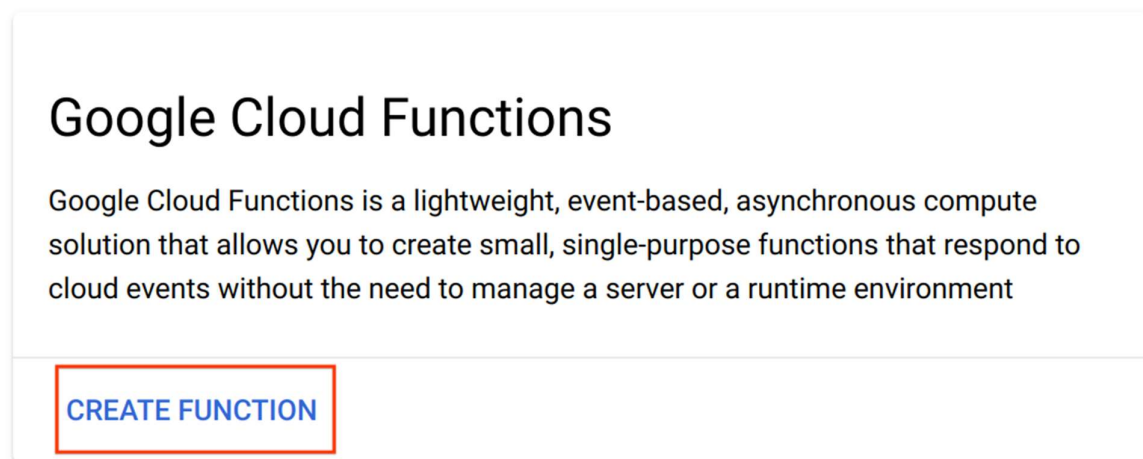
Create a function

In this step, you're going to create a cloud function using the console.

1. In the console, click the **Navigation menu** > **Cloud Functions**.



2. Click **Create function**.



3. In the **Create function** dialog, enter the following values:

Field	Value
Function name	GCFFunction
Trigger	Select HTTP and click Save
Memory allocated (In Runtime, Build and Connections Settings)	Keep it default and click Next



1

Configuration

2

Code

Basics

Function name *

GCFunction



Region

us-central1



Trigger

HTTP

Trigger URL

`https://us-central1-qwiklabs-gcp-03-9d8890a3ef84.cloudfunctions.net/GCFunction`

Authentication

Require authentication

[EDIT](#)

RUNTIME, BUILD AND CONNECTIONS SETTINGS



RUNTIME

BUILD

CONNECTIONS

Memory allocated *

256 MiB



Timeout *

60

seconds

[NEXT](#)[CANCEL](#)

You deploy the function in the next section.

Deploy the function

Still in the **Create function** dialog, in Source code for **Inline editor** use the default `helloWorld` function implementation already provided for `index.js`. At the bottom, click **Deploy** to deploy the function.

After you click **Deploy**, the console redirects to the **Cloud Functions Overview** page.

While the function is being deployed, the icon next to it is a small spinner. When it's deployed, the spinner is a green check mark.



<input type="checkbox"/>	Name ^	Region	Trig
<input type="checkbox"/>	 GCFUNCTION	us-central1	HTT

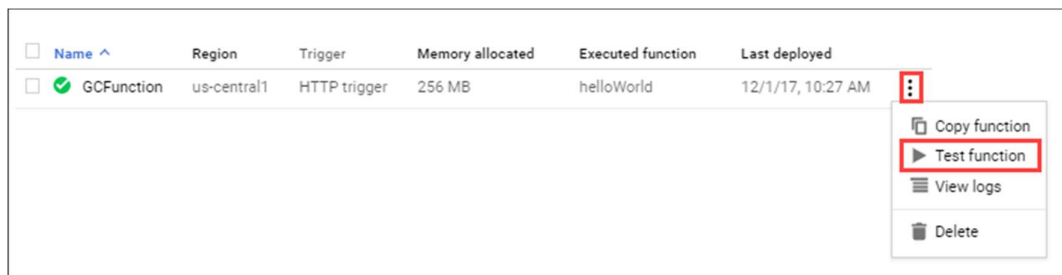
Test Completed Task


Click **Check my progress** to verify your performed task. If you have completed the task successfully you will be granted with an assessment score.

Test the function

Test the deployed function.

1. In the **Cloud Functions Overview** page, display the menu for your function, and click **Test function**.



<input type="checkbox"/>	Name ^	Region	Trigger	Memory allocated	Executed function	Last deployed	
<input type="checkbox"/>	 GCFUNCTION	us-central1	HTTP trigger	256 MB	helloWorld	12/1/17, 10:27 AM	<div><div></div><div>Copy function</div><div>Test function</div><div>View logs</div><div>Delete</div></div>

2. In the Triggering event field, enter the following text between the brackets `{ }` and click **Test the function**.

```
3. "message": "Hello World!"  
   content_copy
```

In the **Output** field, you should see the message `Success: Hello World!`

In the **Logs** field, a status code of **200** indicates success. (It may take a minute for the logs to appear.)

Cloud Functions

[Function details](#) [EDIT](#) [DELETE](#) [COPY](#)

GCFFunction

Version 1, deployed at Nov 26, 2020, 5:03:44 P...

METRICSDETAILSSOURCESOURCEVARIABLESTRIGGERPERMISSIONSLOGSTESTING

Triggering event ?

1

`{"message": "Hello World!"}`

← Enter this text

2

TEST THE FUNCTION

Output

Complete

\$

Hello World!

Logs

Fetched (up to 100 entries). [View all logs](#)

Scanned up to 11/18/20, 5:29 AM. Scanned 1.8 GB.

▶ 2020-11-26T11:35:04.404840722Z GCFFunction p4o1rnyzt7cu Function execution started

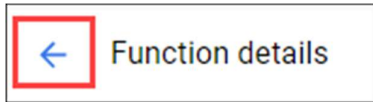
▶ 2020-11-26T11:35:04.602617691Z GCFFunction p4o1rnyzt7cu Function execution took 198 ms, finished with status code: 200

Success!!

View logs

View logs from the Cloud Functions Overview page.

1. Click the blue arrow to go back to the **Cloud Functions Overview** page.



2. Display the menu for your function, and click **View logs**.

<input type="checkbox"/> Name ^	Region	Trigger	Memory allocated	Executed function	Last deployed	
<input type="checkbox"/> GCFunction	us-central1	HTTP trigger	256 MB	helloWorld	12/5/17, 12:40 PM	<div><div></div><div>Copy function</div><div>▶ Test function</div><div>≡ View logs</div><div>🗑 Delete</div></div>

Example of the log history that displays in **Query results**:

Query results

Jump to Now

Actions

Configure

SEVERITY

TIMESTAMP

IST

SUMMARY

Showing logs for last 1 hour starting at 11/26/20, 4:12 PM.

Extend time by: 1 hour

Edit time

>

2020-11-26 17:02:21.124 IST

cloudfunctions.googleapis.com

google.cloud.functions.v1.CloudFunctionsService.CreateFunction

>

2020-11-26 17:03:44.888 IST

cloudfunctions.googleapis.com

google.cloud.functions.v1.CloudFunctionsService.CreateFunction

>

2020-11-26 17:05:04.404 IST

GCFunction

p4o1rnyzt7cu

Function execution started

>

2020-11-26 17:05:04.602 IST

GCFunction

p4o1rnyzt7cu

Function execution took 198 ms, finished with status code: 200

Your application is deployed, tested, and you can view the logs.

Test your Understanding

Below are multiple-choice questions to reinforce your understanding of this lab's concepts. Answer them to the best of your abilities.

Cloud Functions is a serverless execution environment for building and connecting cloud services.

True

Which type of trigger is bound while creating Cloud Function in the lab?

HTTP

Congratulations!

This concludes the hands-on lab, [Cloud Functions: Qwik Start - Console](#). You used the Google Cloud console to create, deploy, and test a cloud function.



Finish Your Quest

Continue your Quest with [Baseline: Deploy & Develop](#) and [Baseline: Infrastructure](#) Quests. A Quest is a series of related labs that form a learning path. Completing this Quest earns you the badge above, to recognize your achievement. You can make your badge (or badges) public and link to them in your online resume or social media account. Enroll in this Quest and get immediate completion credit if you've taken this lab. [See other available Qwiklabs Quests](#).

Take Your Next Lab

- This lab is also part of a series of labs called Qwik Starts. These labs are designed to give you a little taste of the many features available with Google Cloud. Search for "Qwik Starts" in the [lab catalog](#) to find the next lab you'd like to take!
- Now that you used the console to start a Google Cloud Function, try and compare starting a Cloud Function using the command line. See [Cloud Functions: Qwik Start - Using the Command Line](#).
- Check out our [Quests](#) for a series of labs on one topic.

Next Steps / Learn More

- Learn more about an App Engine with [An Overview Of App Engine](#)
- Try something else with an App Engine with [Getting Started with Flask on App Engine Standard Environment](#)

Google Cloud Training & Certification

...helps you make the most of Google Cloud technologies. [Our classes](#) include technical skills and best practices to help you get up to speed quickly and continue your learning journey. We offer fundamental to advanced level training, with on-demand, live, and virtual options to suit your busy schedule. [Certifications](#) help you validate and prove your skill and expertise in Google Cloud technologies.

Manual Last Updated March 23, 2021

Lab Last Tested March 23, 2021

Copyright 2021 Google LLC All rights reserved. Google and the Google logo are trademarks of Google LLC. All other company and product names may be trademarks of the respective companies with which they are associated.