**Cloud Functions: Qwik Start - Command Line**

experimentLabschedule30 minutesuniversal\_currency\_alt1 Creditshow\_chartIntroductory

Rate Lab

**GSP080**



**Overview**

A cloud function is a piece of code that runs in response to an event, such as an HTTP request, a message from a messaging service, or a file upload. 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.

Since cloud functions are event-driven, they only run when something happens. This makes them a good choice for tasks that need to be done quickly or that don't need to be running all the time.

For example, you can use a cloud function to:

* automatically generate thumbnails for images that are uploaded to Cloud Storage.
* send a notification to a user's phone when a new message is received in Cloud Pub/Sub.
* process data from a Cloud Firestore database and generate a report.

You can write your code in any language that supports Node.js, and you can deploy your code to the cloud with a few clicks. Once your cloud function is deployed, it will automatically start running in response to events.

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

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

What you'll do

* Create a simple 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 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.

To complete this lab, you need:

* Access to a standard internet browser (Chrome browser recommended).

**Note:** Use an Incognito or private browser window to run this lab. This prevents any conflicts between your personal account and the Student account, which may cause extra charges incurred to your personal account.

* Time to complete the lab---remember, once you start, you cannot pause a lab.

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

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 the **Lab Details** panel with the following:
   * The **Open Google Cloud console** button
   * Time remaining
   * The temporary credentials that you must use for this lab
   * Other information, if needed, to step through this lab
2. Click **Open Google Cloud console** (or right-click and select **Open Link in Incognito Window** if you are running the Chrome browser).

The lab spins up resources, and then opens another tab that shows the **Sign in** page.

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

**Note:**If you see the **Choose an account** dialog, click **Use Another Account**.

1. If necessary, copy the **Username** below and paste it into the **Sign in** dialog.

"Username"

Copied!

content\_copy

You can also find the **Username** in the **Lab Details** panel.

1. Click **Next**.
2. Copy the **Password** below and paste it into the **Welcome** dialog.

"Password"

Copied!

content\_copy

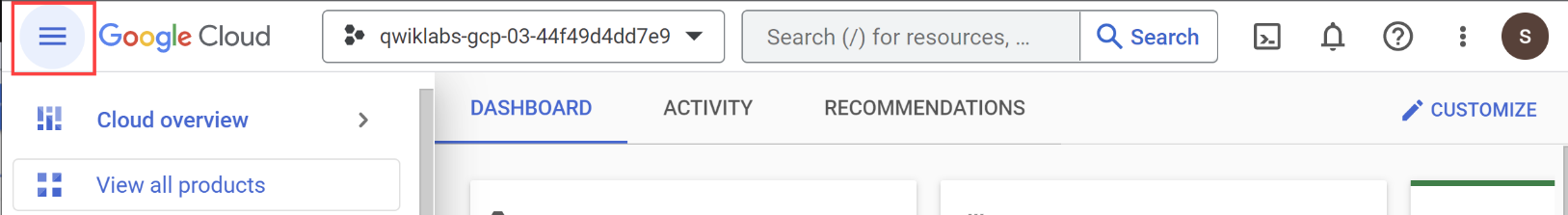
You can also find the **Password** in the **Lab Details** panel.

1. Click **Next**.

**Important:**You must use the credentials the lab provides you. Do not use your Google Cloud account credentials.**Note:**Using your own Google Cloud account for this lab may incur extra charges.

1. 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 Google Cloud console opens in this tab.

**Note:** To view a menu with a list of Google Cloud products and services, click the **Navigation menu** at the top-left. 

Activate Cloud Shell

Cloud Shell is a virtual machine that is loaded with development tools. It offers a persistent 5GB home directory and runs on the Google Cloud. Cloud Shell provides command-line access to your Google Cloud resources.

1. Click **Activate Cloud Shell** Activate Cloud Shell icon at the top of the Google Cloud console.

When you are connected, you are already authenticated, and the project is set to your **Project\_ID**, PROJECT\_ID. The output contains a line that declares the **Project\_ID** for this session:

Your Cloud Platform project in this session is set to "PROJECT\_ID"

gcloud is the command-line tool for Google Cloud. It comes pre-installed on Cloud Shell and supports tab-completion.

1. (Optional) You can list the active account name with this command:

gcloud auth list

Copied!

content\_copy

1. Click **Authorize**.

**Output:**

ACTIVE: \*

ACCOUNT: "ACCOUNT"

To set the active account, run:

$ gcloud config set account `ACCOUNT`

1. (Optional) You can list the project ID with this command:

gcloud config list project

Copied!

content\_copy

**Output:**

[core]

project = "PROJECT\_ID"

**Note:**For full documentation of gcloud, in Google Cloud, refer to [the gcloud CLI overview guide](https://cloud.google.com/sdk/gcloud).

**Task 1. Create a function**

First, you're going to create a simple function named helloWorld. This function writes a message to the Cloud Functions logs. It is triggered by cloud function events and accepts a callback function used to signal completion of the function.

For this lab the cloud function event is a cloud pub/sub topic event. A pub/sub is a messaging service where the senders of messages are decoupled from the receivers of messages. When a message is sent or posted, a subscription is required for a receiver to be alerted and receive the message. To learn more about pub/subs, in Cloud Pub/Sub Guides, see [Google Cloud Pub/Sub: A Google-Scale Messaging Service](https://cloud.google.com/pubsub/architecture).

To learn more about the event parameter and the callback parameter, in Cloud Functions Documentation, see [Background Functions](https://cloud.google.com/functions/docs/writing/background).

To create a cloud function:

1. In Cloud Shell, run the following command to set the default region:

gcloud config set compute/region REGION

Copied!

content\_copy

1. Create a directory for the function code:

mkdir gcf\_hello\_world

Copied!

content\_copy

1. Move to the gcf\_hello\_world directory:

cd gcf\_hello\_world

Copied!

content\_copy

1. Create and open index.js to edit:

nano index.js

Copied!

content\_copy

1. Copy the following into the index.js file:

/\*\*

\* Background Cloud Function to be triggered by Pub/Sub.

\* This function is exported by index.js, and executed when

\* the trigger topic receives a message.

\*

\* @param {object} data The event payload.

\* @param {object} context The event metadata.

\*/

exports.helloWorld = (data, context) => {

const pubSubMessage = data;

const name = pubSubMessage.data

? Buffer.from(pubSubMessage.data, 'base64').toString() : "Hello World";

console.log(`My Cloud Function: ${name}`);

};

Copied!

content\_copy

1. Exit nano (Ctrl+x) and save (Y) the file.

**Task 2. Create a Cloud Storage bucket**

* Use the following command to create a new Cloud Storage bucket for your function:

gsutil mb -p [PROJECT\_ID] gs://[BUCKET\_NAME]

Copied!

content\_copy

**PROJECT\_ID** is the Project ID in the lab details panel on the left of this lab: <filled in at lab start>

**BUCKET\_NAME** is the name you give to the bucket. You can use the Project ID as the bucket name to ensure a globally unique name: <filled in at lab start>

To learn more about naming buckets, in Cloud Storage documentation, see [Bucket naming guidelines](https://cloud.google.com/storage/docs/naming-buckets).

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.

Create a cloud storage bucket.

Check my progress

**Task 3. Deploy your function**

When deploying a new function, you must specify --trigger-topic, --trigger-bucket, or --trigger-http. When deploying an update to an existing function, the function keeps the existing trigger unless otherwise specified.

For this lab, you'll set the --trigger-topic as hello\_world.

1. Disable the Cloud Functions API:

gcloud services disable cloudfunctions.googleapis.com

Copied!

content\_copy

1. Re-enable the Cloud Functions API:

gcloud services enable cloudfunctions.googleapis.com

Copied!

content\_copy

1. Add the artifactregistry.reader permission for your appspot service account. Replace **[PROJECT\_ID]** with your qwiklabs Project ID.

gcloud projects add-iam-policy-binding [PROJECT\_ID] \

--member="serviceAccount:[PROJECT\_ID]@appspot.gserviceaccount.com" \

--role="roles/artifactregistry.reader"

Copied!

content\_copy

1. Deploy the function to a pub/sub topic named **hello\_world**, replacing [BUCKET\_NAME] with the name of your bucket:

gcloud functions deploy helloWorld \

--stage-bucket [BUCKET\_NAME] \

--trigger-topic hello\_world \

--runtime nodejs20

Copied!

content\_copy

**Note:**If you get OperationError, ignore the warning and re-run the command.

If prompted, enter Y to allow unauthenticated invocations of a new function.

1. Verify the status of the function:

gcloud functions describe helloWorld

Copied!

content\_copy

An ACTIVE status indicates that the function has been deployed.

entryPoint: helloWorld

eventTrigger:

eventType: providers/cloud.pubsub/eventTypes/topic.publish

failurePolicy: {}

resource:

...

status: ACTIVE

...

Every message published in the topic triggers function execution, the message contents are passed as input data.

Test Completed Task

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

Deploy the function.

Check my progress

**Task 4. Test the function**

After you deploy the function and know that it's active, test that the function writes a message to the cloud log after detecting an event.

* Enter this command to create a message test of the function:

DATA=$(printf 'Hello World!'|base64) && gcloud functions call helloWorld --data '{"data":"'$DATA'"}'

Copied!

content\_copy

The cloud tool returns the execution ID for the function, which means a message has been written in the log.

Example output:

executionId: 3zmhpf7l6j5b

View logs to confirm that there are log messages with that execution ID.

**Task 5. View logs**

* Check the logs to see your messages in the log history:

gcloud functions logs read helloWorld

Copied!

content\_copy

If the function executed successfully, messages in the log appear as follows:

LEVEL: D

NAME: helloWorld

EXECUTION\_ID: 4bgl3jw2a9i3

TIME\_UTC: 2023-03-23 13:45:31.545

LOG: Function execution took 912 ms, finished with status: 'ok'

LEVEL: I

NAME: helloWorld

EXECUTION\_ID: 4bgl3jw2a9i3

TIME\_UTC: 2023-03-23 13:45:31.533

LOG: My Cloud Function: Hello World!

LEVEL: D

NAME: helloWorld

EXECUTION\_ID: 4bgl3jw2a9i3

TIME\_UTC: 2023-03-23 13:45:30.633

LOG: Function execution started

**Note:** The logs will take around 10 mins to appear. Also, the alternative way to view the logs is, go to **Logging** > **Logs Explorer**.

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

**Task 6. 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.

Serverless lets you write and deploy code without the hassle of managing the underlying infrastructure.



True



False

**Congratulations!**

You used the Google Cloud console to create, deploy, and test a cloud function using the command line.

Take your next lab

This lab is 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](https://google.qwiklabs.com/catalog) to find the next lab you'd like to take!

Next steps / Learn more

* Now that you used the command line to start a Cloud Function, try the [Cloud Functions: Qwik Start - Console](https://google.qwiklabs.com/catalog_lab/704) lab to start a Cloud Function using the Cloud Console.
* For more information on creating triggers and associating them with your functions, see [the Events and Triggers section of the Cloud Functions Guides](https://cloud.google.com/functions/docs/concepts/events-triggers).

Google Cloud training and certification

...helps you make the most of Google Cloud technologies. [Our classes](https://cloud.google.com/training/courses) 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](https://cloud.google.com/certification/) help you validate and prove your skill and expertise in Google Cloud technologies.

**Manual Last Updated April 08, 2024**

**Lab Last Tested April 08, 2024**

Copyright 2024 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.

[navigate\_beforePrevious](https://www.cloudskillsboost.google/course_sessions/14297640/labs/464354)[Nextnavigate\_next](https://www.cloudskillsboost.google/course_sessions/14297640/labs/464356)