Cloud Run Functions: Qwik Start - Console

Overview

A Cloud Run 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 Run 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 Run 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 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 Run 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 Run function using the Google Cloud console.

What you'll do

- Create a Cloud Run function
- Deploy and test the function
- View logs

Setup and requirements

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 are made available to you.

This hands-on lab lets you do the lab activities in a real cloud environment, not in a simulation or demo environment. It does so by giving you new, temporary credentials 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 (recommended) or private browser window to run this lab. This prevents 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: Use only the student account for this lab. If you use a different Google Cloud account, you may incur charges to that 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 dialog opens for you to select your payment method. On the left is the Lab Details pane 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**.

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

"Username"

You can also find the Username in the Lab Details pane.

4. Click **Next**.
5. Copy the **Password** below and paste it into the **Welcome** dialog.

"Password"

You can also find the Password in the Lab Details pane.

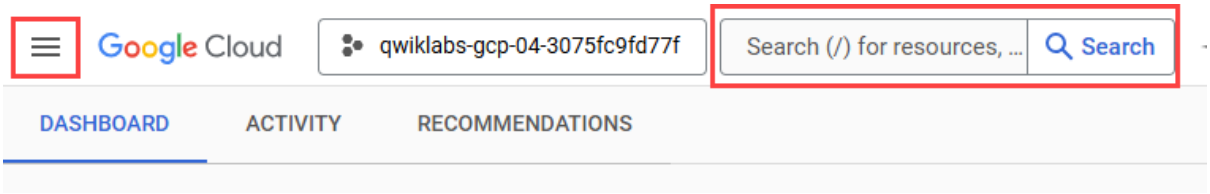
6. Click **Next**.

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.

7. 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 access Google Cloud products and services, click the **Navigation menu** or type the service or product name in the **Search** field.



Task 1. Create a function

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

1. In the console, on the **Navigation menu** (☰) click **Cloud Run**.
2. Click **WRITE A FUNCTION**.
3. In the **function** dialog, enter the following values:

Field	Value
Service name	gcfunction
Region	REGION
Authentication	Allow unauthenticated invocations
Memory allocated (In Container(s), Volumes and Security Settings)	Keep default
Execution environment (In Container(s), Volumes and Security Settings)	Second generation
Revision scaling (In Container(s), Volumes and Security Settings)	Set the **Maximum number of instance** to **5**, and then click **Create**

Note: A helpful popup may appear to validate the required APIs are enabled in the project. Click the **ENABLE** button when requested.

You deploy the function in the next section.

Task 2. Deploy the function

1. Still in the **Create function** dialog, in Source code for **Inline editor** use the default helloHttp function implementation already provided for index.js.
2. Click **SAVE and REDEPLOY** to deploy the function.

Note: 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.

Task 3. Test the function

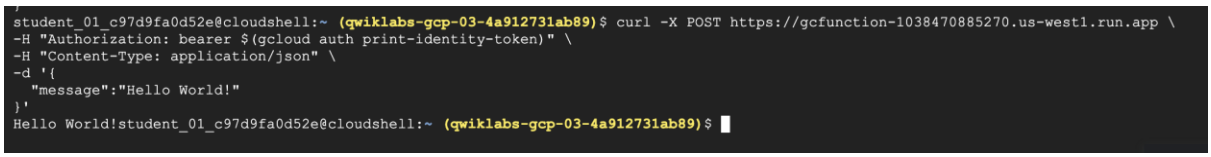
Test the deployed function.

1. On the function details dashboard, to test the function click **TEST**.

2. In the Triggering event field, enter the following text between the brackets {} .

"message":"Hello World!"

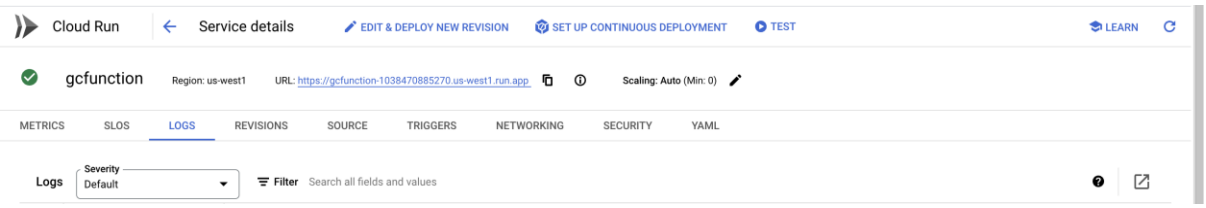
3. Copy the **CLI test command** and run it in the cloud shell.
4. You will see the "Hello World!" message as the output.



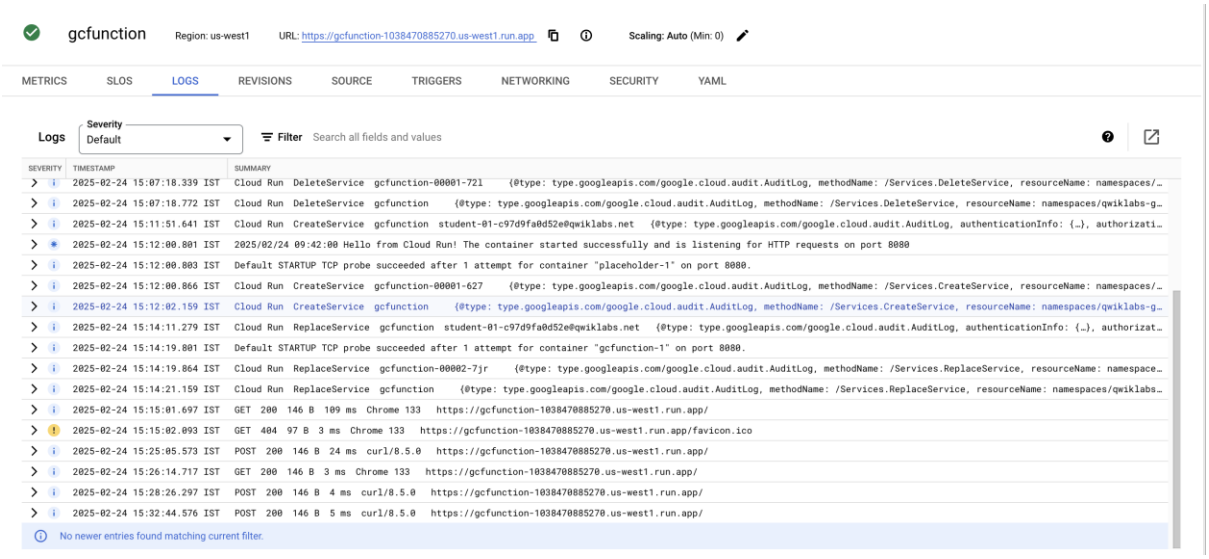
Task 4. View logs

View logs from the service details page.

1. On the **Service Details Overview** page click **Logs** tab



Example of the log history that displays in **Results**:



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

Congratulations!