**Google Cloud Functions**

**Overview**

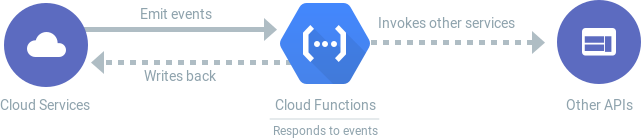
With Cloud Functions you write simple, single-purpose functions that are attached to events emitted from your cloud infrastructure and services. Your Cloud Function is triggered when an event being watched is fired. Your code executes in a fully managed environment. There is no need to provision any infrastructure or worry about managing any servers.

Cloud Functions can currently be written in JavaScript, Python, or Go. In the case of JavaScript, they execute in a Node.js environment on Google Cloud Platform. You can take your Cloud Function and run it in any standard Node.js runtime which makes both portability and local testing a breeze.

**Features**

**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 Platform services such as Data store, Cloud Spanner, Cloud Translation API, Cloud Vision API, as well as many others.



**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.

**Server less**

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.

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

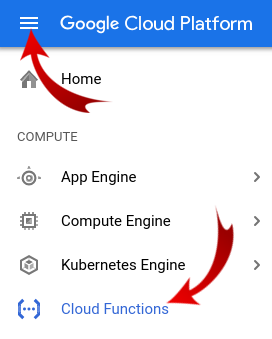
* Create a cloud function
* Deploy and test the function
* View logs

**LAB**

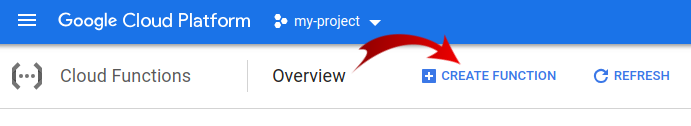
**Create a function**

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

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

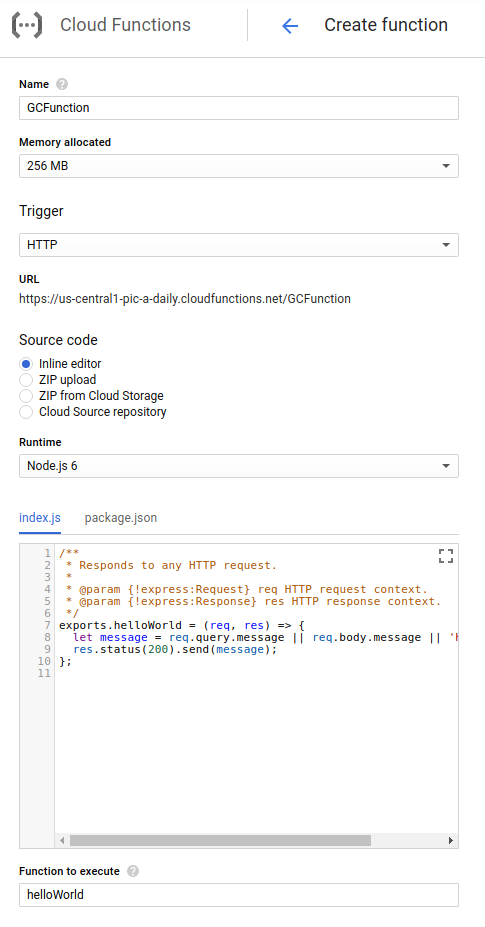


**Click "CREATE FUNCTION”:**



**Name your function GC Function and keep the other defaults:**

Memory allocation, HTTP as a trigger, inline editor, default runtime, and default sample code.



While still in the "create function" dialog, at the bottom, click "Create" to deploy the function:

b68c3647b771e6f9.png

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

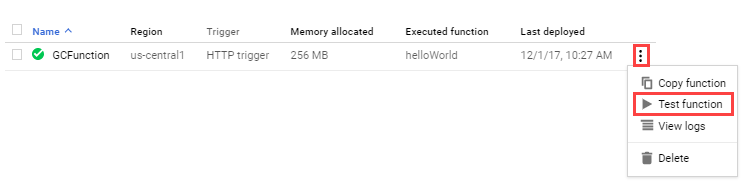
While the function is being deployed, the icon next to it is a small spinner. When it's deployed, the spinner will be a green check mark. If there are any issues, you will be notified with a red icon and a link to the logs to understand what went wrong (likely an issue in your code).

3ec684b1b4906657.png

That's it! Your function is now live and ready to be triggered via HTTP.

**Test the function**

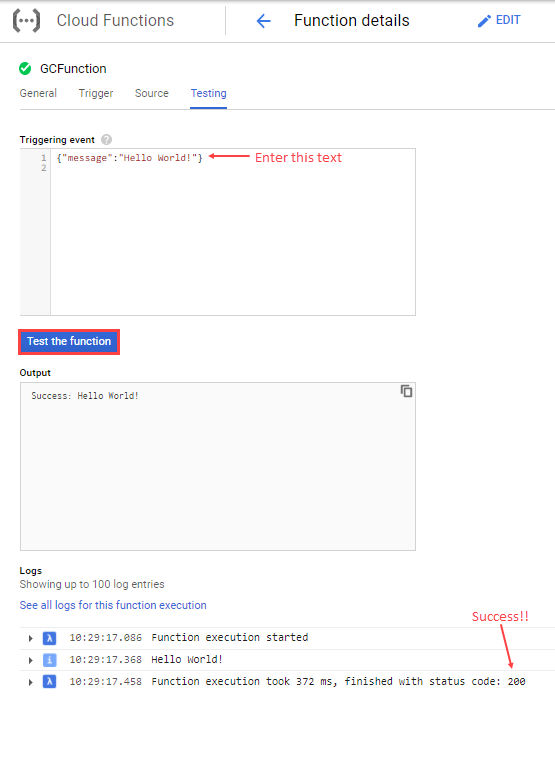
In the Cloud Functions Overview page, display the menu for your function, and click Test function:



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

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

In the Logs field, a status code of 200 indicates success. Note that it may take a minute for the logs to appear.



**View the function logs**

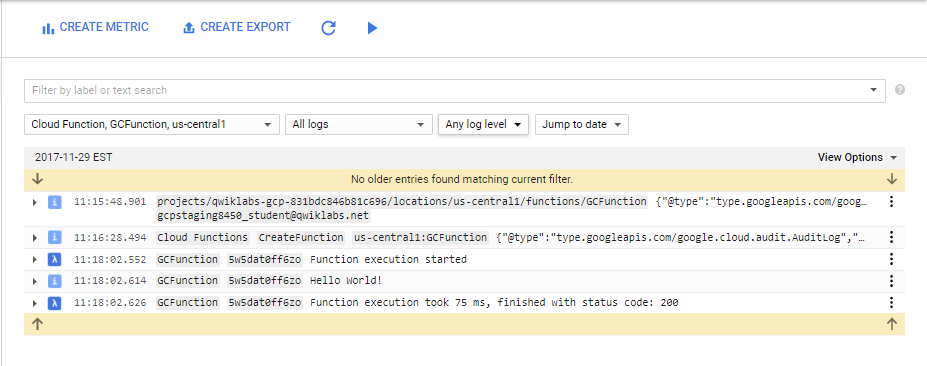
Click the blue arrow to go back to the Cloud Functions Overview page:

8917a2bfa4fb9502.png

Display the menu for your function, and click View logs:



Here is an example of the log history:



Note that you can filter logs per function, per severity, and search via a label or some free-form text.

Your application is deployed, tested, and you were able to view the logs.

**Cleanup resources**

Well you don't really need to clean up resources since this is server less computing: scaling to zero meaning also scaling cost to zero. So if your function is getting no traffic, there will be no cost incurred. Also, the first 2 million Cloud Function invocations each month are free.

If you'd like to delete the function, simply head over to the overview page, select the function and click DELETE:

