

Serverless Applications

Google Cloud Functions

This content is protected and may not be shared, uploaded, or distributed.

Outline

- Google Cloud Functions + API Management
- Documentation available at:
<https://cloud.google.com/functions>
- Scalable pay as you go Functions-as-a-Service (FaaS) to run your code with zero server management.
 - No servers to provision, manage, or upgrade
 - Automatically scale based on the load
 - Integrated monitoring, logging, and debugging capability
 - Built-in security at role and per function level based on the principle of least privilege
 - Key networking capabilities for hybrid and multi-cloud scenarios

Google Cloud Functions

Why Google Cloud Functions?

- **Serverless Applications on Google's Infrastructure.** Construct applications from bite-sized business logic **billed** to the nearest **100 milliseconds**, only while your code is running. Serve users from zero to planet-scale, all without managing any infrastructure.

Microservices Over Monoliths

- Developer agility comes from building systems composed of small, independent units of functionality focused on doing one thing well. Cloud Functions lets you build and deploy services at the level of a single function, not at the level of entire applications, containers, or VMs.

Connect & 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 events such as a file upload to Cloud Storage, an incoming message on a Cloud Pub/Subtopic, a log change in Stackdriver Logging, or a mobile-related event from Firebase.

Serverless Economics

- Cloud Functions are **ephemeral**, spinning up on-demand and back down in response to events in the environment. Pay only while your function is executing, metered to the nearest 100 milliseconds, and pay nothing after your function finishes.

Google Cloud Functions (cont'd)

Mobile Ready

- Mobile app developers can use Cloud Functions directly from **Firebase**, Google Cloud's mobile platform. Firebase natively emits events to which Cloud Functions can respond, including from Firebase Analytics, Realtime Database, Authentication, and Storage.

Just Add Code

- Run in a fully-managed, serverless environment where Google handles servers, operating systems, and runtime environments completely on your behalf. Each Cloud Function runs in its own isolated secure execution context, scales automatically, and has a lifecycle independent from other functions.

Open and Familiar

- **Cloud Functions are written in JavaScript and execute in a standard Node.js runtime environment.** .NET, Go, Java, PHP, Python and Ruby are also supported. We don't assume anything proprietary all the way down to the operating system, which means your functions will just work—including native libraries you bring to the platform. Discover a superior, open developer experience that comes from working hand-in-hand with the Node.js Foundation, with our Google colleagues and with the community through the **open source V8 engine**.

Google Cloud Functions Use Cases

Mobile Backend

- Use Google's mobile platform for app developers, Firebase, and extend your mobile backend with Cloud Functions. Listen and respond to events from Firebase Analytics, Realtime Database, Authentication, and Storage.

APIs & Microservices

- Compose applications from lightweight, loosely coupled bits of logic that are quick to build and scale automatically. Your functions can be event-driven or invoked directly over HTTP/S.

Data Processing / ETL

- Listen and respond to Cloud Storage events such as when a file is created, changed, or removed. Process images, do video transcoding, validate or 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/S requests.

IoT

- Imagine tens or hundreds of thousands of devices streaming data into Cloud Pub/Sub automatically launching Cloud Functions to process, transform and store data. Cloud Functions lets you do this in a way that's completely serverless.

Google Cloud Functions Features

Cloud Pub/Sub Triggers

- Cloud Functions can be triggered by messages on a Cloud Pub/Sub topic, and multiple functions can subscribe to the same topic.

Cloud Storage Triggers

- You can associate a Cloud Function to mutation events on a Cloud Storage bucket. Every time a file in your bucket is created, deleted or modified, your function will execute.

Firebase Triggers

- Mobile developers will find first-class integration between Firebase and Cloud Functions.

HTTP/S Invocation

- Functions deployed with an HTTP trigger are given a fully qualified domain together with a dynamically generated TLS certificate for secure communication.

GitHub/Bitbucket

- Using Cloud Source Repositories you can deploy Cloud Functions directly from your Github or Bitbucket repository without needing to upload code or manage versions yourself.

Logging, Monitoring & Debugging

- Logs emitted from your Cloud Functions are automatically written to Stackdriver Logging and performance telemetry is recorded in Stackdriver Monitoring. Stackdriver Debugger lets you investigate your code's behavior in production.

Google Cloud Functions Pricing

	FREE MONTH	LIMIT PER UNIT	PRICE UNIT
	ABOVE FREE LIMIT (PER UNIT)		
Invocations *	2 million invocations	\$0.40	per million invocations
Compute Time	400,000 GB-seconds	\$0.0000025	per GB-Second
	200,000 GHz seconds	\$0.0000100	per GHz-Second
Outbound Data (Egress)	5GB	\$0.12	per GB
Inbound Data (Ingress)	Unlimited	Free	per GB
Outbound Data to Google APIs in same region	Unlimited	Free	per GB

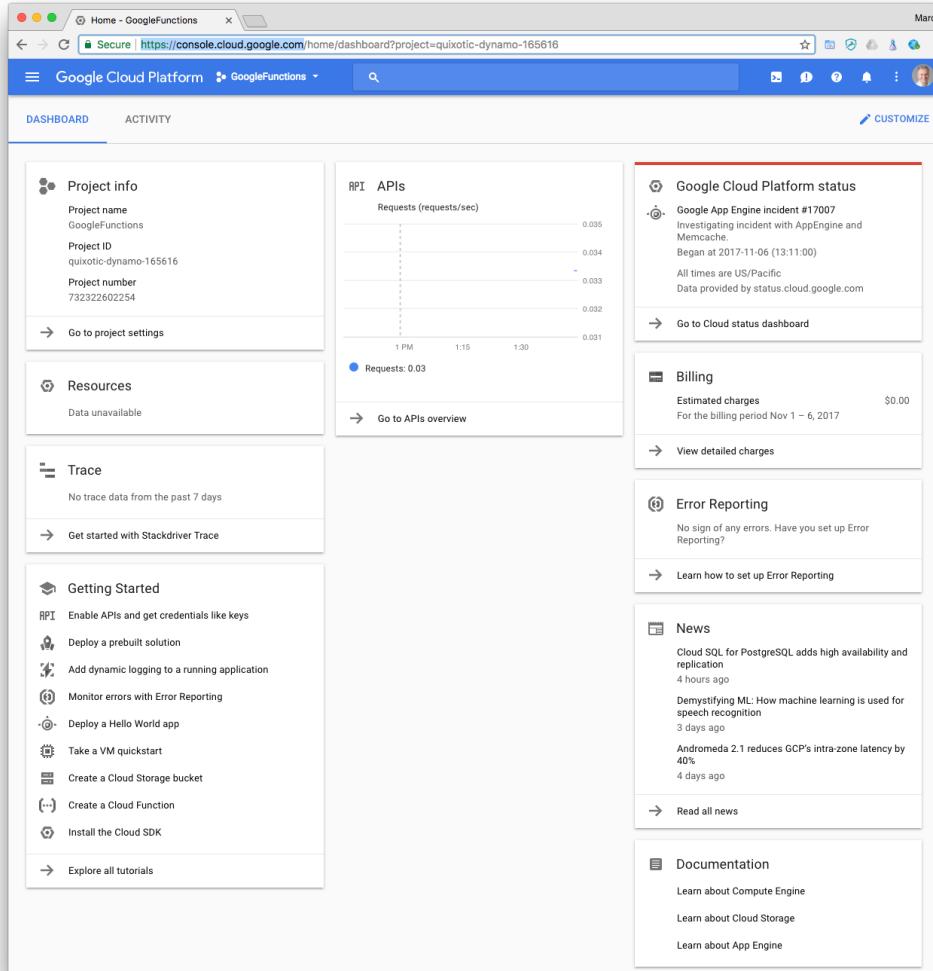
* Includes both Background and HTTP Functions.

See: <https://cloud.google.com/functions/pricing>

Create a Simple HTTP service using Google Cloud Functions

In this exercise you will demonstrate writing, deploying, and triggering an HTTP Cloud Function. The Cloud Function is triggered by an HTTP request and outputs a “Hello World” in our browser. This tutorial uses billable components of Cloud Platform, including Google Cloud Functions.

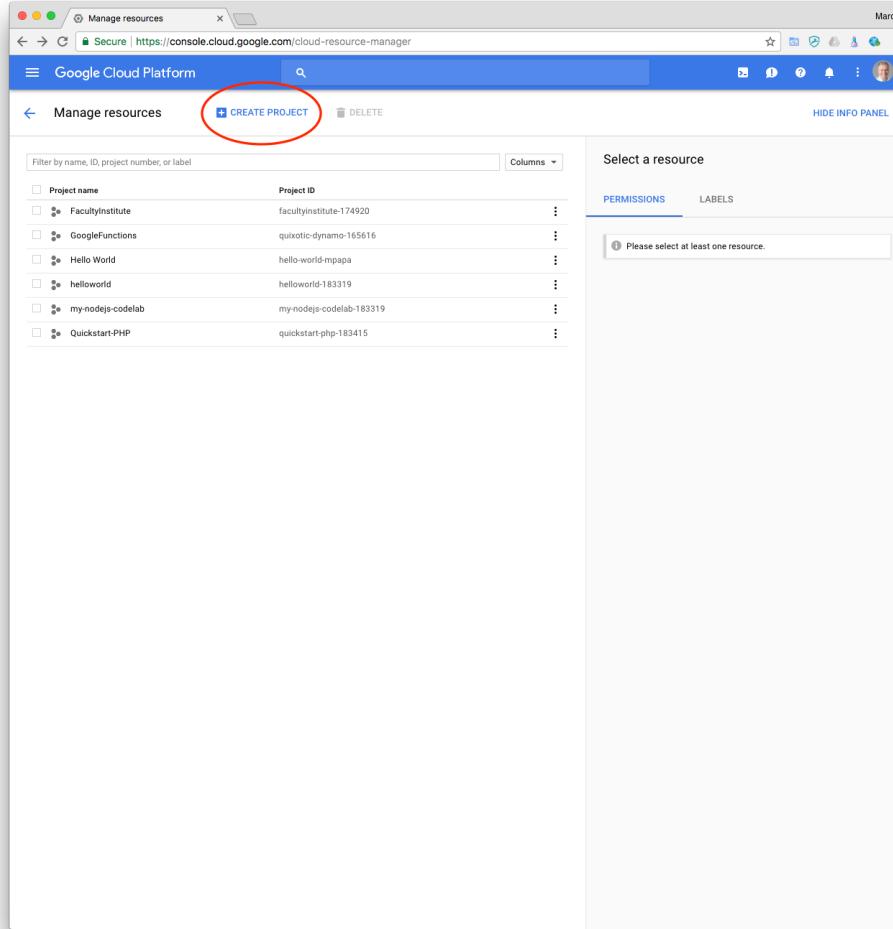
Google Cloud Functions (cont'd)



Follow the steps in this section to create a new Google Cloud function and an API endpoint to trigger it:

1. **Sign-in to the Google Cloud Platform at:**
<https://console.cloud.google.com>

Google Cloud Functions (cont'd)



2. Select or create a Cloud Platform project. Go to the Projects page at:

<https://console.cloud.google.com/project>

3. Select **CREATE PROJECT**.

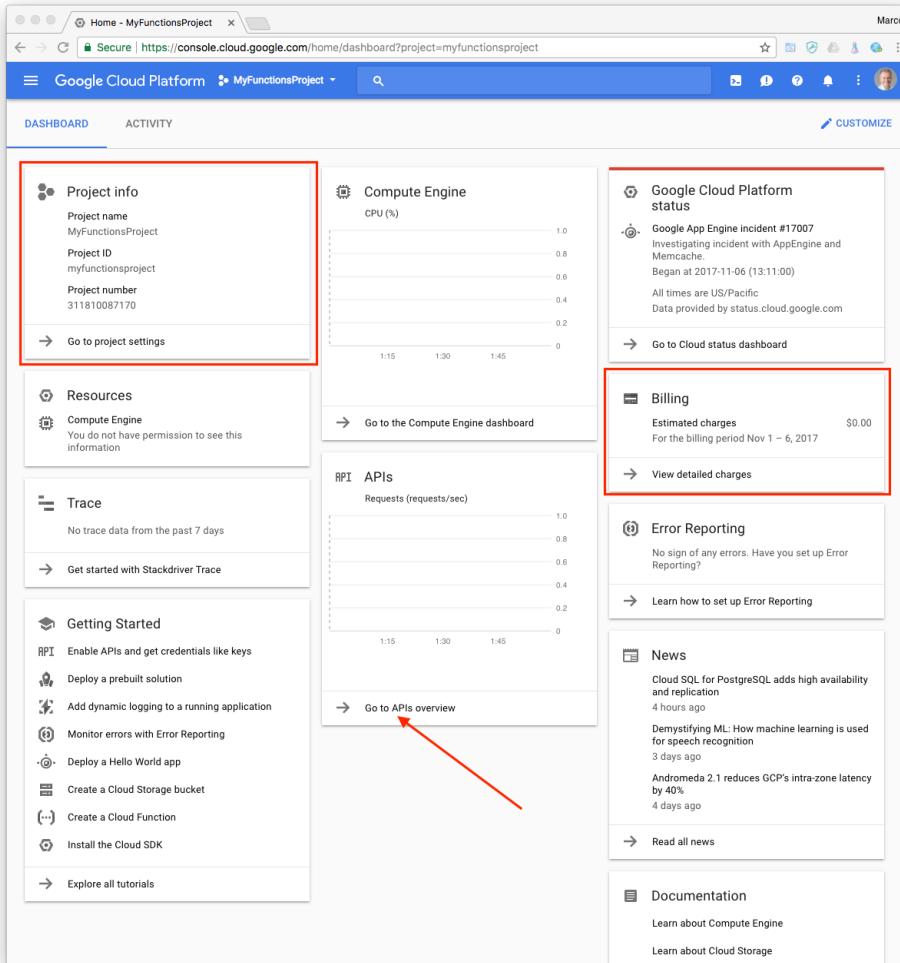
Google Cloud Functions (cont'd)

The image contains two screenshots of the Google Cloud Platform interface. The top screenshot shows the 'New Project' creation page. It has a 'Project name' field containing 'MyFunctionsProject'. A red arrow points to the 'Create' button. The bottom screenshot shows the 'Manage resources' page, displaying a list of projects. The 'MyFunctionsProject' row is circled in red.

Project Name	Project ID
FacultyInstitute	facultyinstitute-174920
GoogleFunctions	quixotic-dynamo-165616
Hello World	hello-world-mpapa
helloworld	helloworld-183319
my-nodejs-codelab	mynodejs-codelab-183319
MyFunctionsProject	myfunctionsproject
Quickstart PHP	quickstart-php-183415

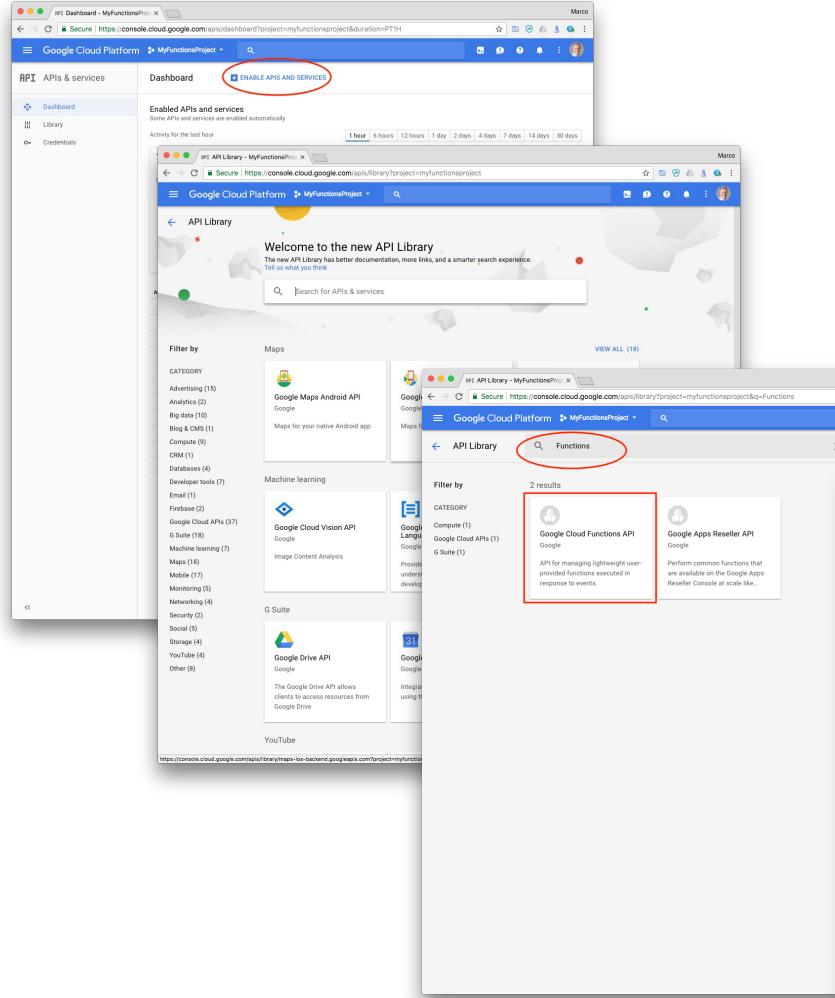
4. Enter your project name, such as *MyFunctionsProject*. Notice the project ID.
5. Click **Create**.
6. You may have to refresh the page to see your new project.
7. Click on the project name, *MyFunctionsProject*, in this example.
8. Click the **Products and Services** “3 bars” icon on top left and select **Home**.

Google Cloud Functions (cont'd)



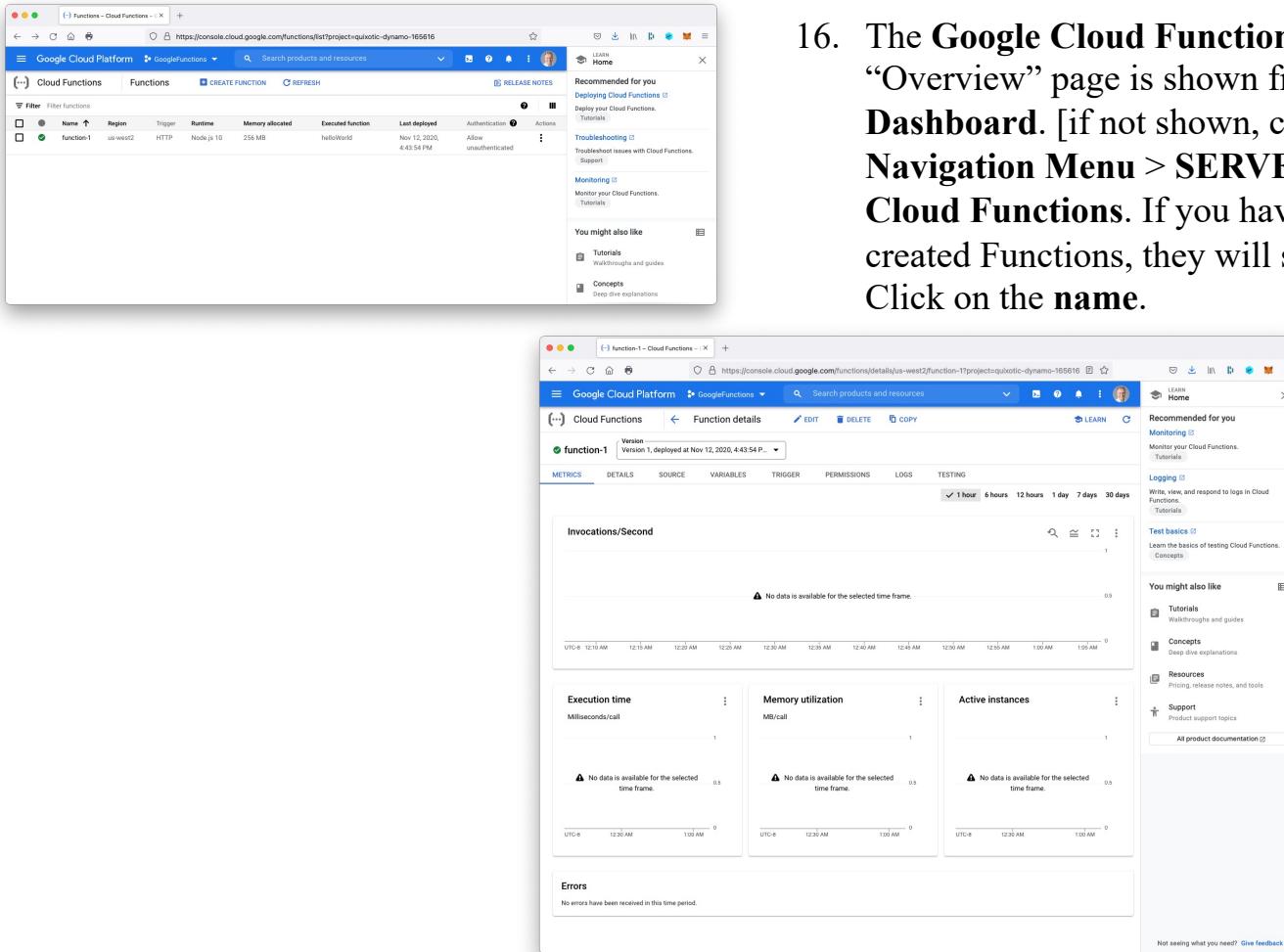
9. **Enable billing** for a project
10. How you enable billing depends on whether you're creating a new project or you're re-enabling billing for an existing project.
11. When you create a new project, you're prompted to choose which of your billing accounts you want to link to the project. If you have only one billing account, that account is automatically linked to your project.
12. Click on **Go to APIs overview**.

Google Cloud Functions (cont'd)



13. Enable the Cloud Functions API. In the **API & Services Dashboard**, click **ENABLE APIS AND SERVICES**. The page titled “Welcome to the new API Library” appears.
14. Enter “Functions” in the edit box. Then click **Cloud Functions API**.
15. Click **ENABLE**. Wait while “Enabling API.” repeat the same to enable the **Cloud Build API**.

Google Cloud Functions (cont'd)



The image contains two screenshots of the Google Cloud Functions interface. The top screenshot shows the 'Cloud Functions' list page, where a single function named 'function-1' is listed. The bottom screenshot shows the 'Function details' page for 'function-1', which includes tabs for Metrics, Details, Source, Variables, Trigger, Permissions, Logs, and Testing. The Metrics section displays various performance metrics over a selected time frame, with most values being zero or showing no data available.

16. The **Google Cloud Functions API** “Overview” page is shown from the **Dashboard**. [if not shown, click on the **Navigation Menu > SERVERLESS > Cloud Functions**. If you have previously created Functions, they will show on the list. Click on the **name**.

Google Cloud Functions (cont'd)

The screenshot shows the Google Cloud SDK Documentation page. The left sidebar contains navigation links for Cloud SDK, Quickstarts, How-to Guides, APIs & Reference, Concepts, and Resources. The main content area is titled "Google Cloud SDK Documentation". It features a "Contents" sidebar with links to "Install the latest Cloud Tools version (178.0.0)", "Install the latest Google Cloud Client Libraries", and "More information". Below this is a detailed description of the Google Cloud SDK and its command-line tools. A section titled "Install the latest Cloud Tools version (178.0.0)" provides instructions for different platforms: LINUX, DEBIAN/UBUNTU, RED HAT/CENTOS, MAC OS X (selected), and WINDOWS. Step 1 instructs to ensure Python 2.7.9 or later is installed, with a command-line example: "python -V". Step 2 provides download links for two packages:

PLATFORM	PACKAGE	SIZE	SHA256 CHECKSUM
Mac OS X (x86_64)	google-cloud-sdk-178.0.0-darwin-x86_64.tar.gz	14.9 MB	8ed778bf3627af9504d9ec1cac7e5e4ea1394edc9da306454b6bf12c499a4db2
Mac OS X (x86)	google-cloud-sdk-178.0.0-darwin-x86.tar.gz	14.9 MB	312f1b027ba0a4344db3d5797f76df4e56dcdb9107eee8e1fa262cef267d1f4

17. Install and initialize the Cloud SDK at:

<https://cloud.google.com/sdk/docs/>

18. In the section titled *Install the latest Cloud Tools version 270.0.0*, select your platform (several versions of Linux, Mac OSX or Windows)

19. Make sure that Python 3.7 or later is installed on your system:

```
python -V
```

20. Download your selected package. (google-cloud-sdk-270.0.0-darwin-x86_64.tar on macOS)
21. Extract the file to any location on your file system.

Google Cloud Functions (cont'd)

The screenshot shows a terminal window titled "Desktop -- bash -- 114x62". The user is running the command `./google-cloud-sdk-3/install.sh`. The output includes a usage statistics opt-out message, configuration of usage reporting, and a confirmation of the current Cloud SDK version (178.0.0). A table titled "Components" lists installed components with their IDs and sizes. The terminal also provides instructions for installing or removing components, updating the SDK, and sourcing completion scripts. The session ends with a prompt for the user's \$PATH.

```
Last login: Mon Nov  6 14:29:53 on ttys002
Marco-Papas-Mac-mini:~ marcopapa$ cd Desktop
Marco-Papas-Mac-mini:Desktop marcopapa$ ./google-cloud-sdk-3/install.sh
Welcome to the Google Cloud SDK!

To help improve the quality of this product, we collect anonymized usage data
and anonymized stacktraces when crashes are encountered; additional information
is available at <https://cloud.google.com/sdk/usage-statistics>. You may choose
to opt out of this collection now (by choosing 'N' at the below prompt), or at
any time in the future by running the following command:

  gcloud config set disable_usage_reporting true

Do you want to help improve the Google Cloud SDK (Y/n)? n

Your current Cloud SDK version is: 178.0.0
The latest available version is: 178.0.0

Components
+-----+-----+-----+-----+
| Status | Name | ID   | Size  |
+-----+-----+-----+-----+
| Not Installed | App Engine Go Extensions | app-engine-go | 97.7 MiB |
| Not Installed | Cloud Bigtable Command Line Tool | cbt | 4.0 MiB |
| Not Installed | Cloud Bigtable Emulator | bigtable | 3.5 MiB |
| Not Installed | Cloud Datalab Command Line Tool | datalab | < 1 MiB |
| Not Installed | Cloud Datastore Emulator | cloud-datastore-emulator | 17.7 MiB |
| Not Installed | Cloud Datastore Emulator (Legacy) | gcd-emulator | 38.1 MiB |
| Not Installed | Cloud Pub/Sub Emulator | pubsub-emulator | 33.2 MiB |
| Not Installed | Emulator Reverse Proxy | emulator-reverse-proxy | 14.5 MiB |
| Not Installed | Google Container Local Builder | container-builder-local | 3.7 MiB |
| Not Installed | Google Container Registry's Docker credential helper | docker-credential-gcr | 2.2 MiB |
| Not Installed | gcloud Alpha Commands | alpha | < 1 MiB |
| Not Installed | gcloud Beta Commands | beta | < 1 MiB |
| Not Installed | gcloud app Java Extensions | app-engine-java | 116.0 MiB |
| Not Installed | gcloud app PHP Extensions | app-engine-php | 21.9 MiB |
| Not Installed | gcloud app Python Extensions | app-engine-python | 6.2 MiB |
| Not Installed | kubectl | kubectl | 12.2 MiB |
| Installed | BigQuery Command Line Tool | bq | < 1 MiB |
| Installed | Cloud SDK Core Libraries | core | 7.4 MiB |
| Installed | Cloud Storage Command Line Tool | gsutil | 3.3 MiB |

To install or remove components at your current SDK version [178.0.0], run:
$ gcloud components install COMPONENT_ID
$ gcloud components remove COMPONENT_ID

To update your SDK installation to the latest version [178.0.0], run:
$ gcloud components update

==> Source [/Users/marcopapa/Desktop/google-cloud-sdk-3/completion.bash.inc] in your profile to enable shell command completion for gcloud.
==> Source [/Users/marcopapa/Desktop/google-cloud-sdk-3/path.bash.inc] in your profile to add the Google Cloud SDK command line tools to your $PATH.

For more information on how to get started, please visit:
https://cloud.google.com/sdk/docs/quickstarts

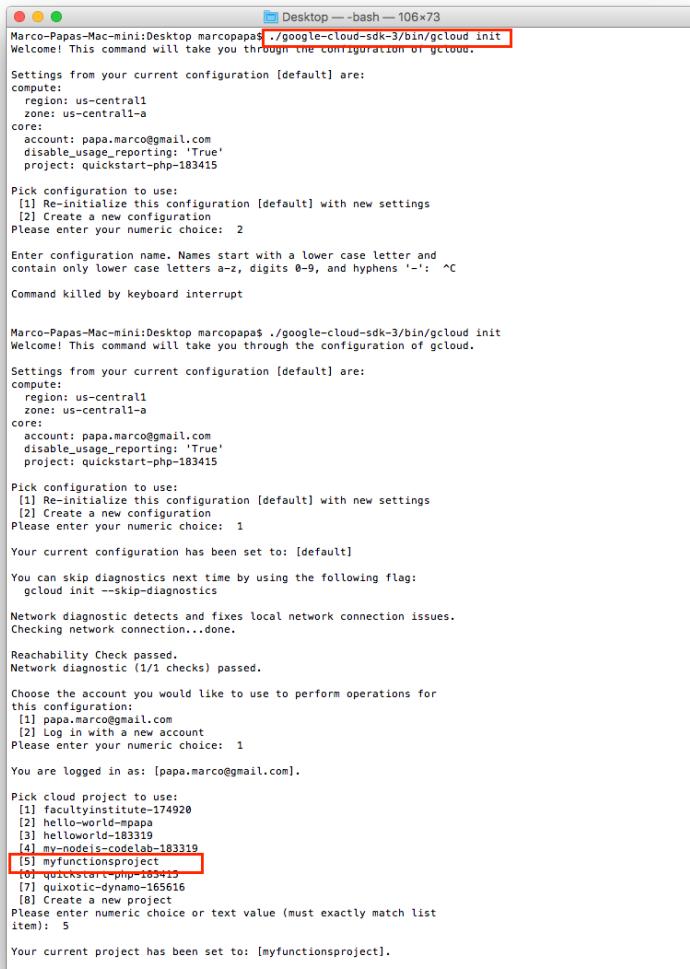
Marco-Papas-Mac-mini:Desktop marcopapa$
```

22. Run the **install script** to add SDK tools to your path, enable command completion in your bash shell, and/or and enable usage reporting.
`./google-cloud-sdk-3/install.sh`

Note: you may have to rename the SDK folder `google-cloud-sdk-3` from “`google-cloud-sdk 3`”.

18. Open a new terminal so that the changes take effect.

Google Cloud Functions (cont'd)



```
Marco-Papas-Mac-mini:Desktop marcopapa$ ./google-cloud-sdk-3/bin/gcloud init
Welcome! This command will take you through the configuration of gcloud.

Settings from your current configuration [default] are:
compute:
  region: us-central1
  zone: us-central1-a
core:
  account: papa.marco@gmail.com
  disable_usage_reporting: 'True'
  project: quickstart-php-183415

Pick configuration to use:
[1] Re-initialize this configuration [default] with new settings
[2] Create a new configuration
Please enter your numeric choice: 2

Enter configuration name. Names start with a lower case letter and
contain only lower case letters a-z, digits 0-9, and hyphens '-': ^C
Command killed by keyboard interrupt

Marco-Papas-Mac-mini:Desktop marcopapa$ ./google-cloud-sdk-3/bin/gcloud init
Welcome! This command will take you through the configuration of gcloud.

Settings from your current configuration [default] are:
compute:
  region: us-central1
  zone: us-central1-a
core:
  account: papa.marco@gmail.com
  disable_usage_reporting: 'True'
  project: quickstart-php-183415

Pick configuration to use:
[1] Re-initialize this configuration [default] with new settings
[2] Create a new configuration
Please enter your numeric choice: 1

Your current configuration has been set to: [default]

You can skip diagnostics next time by using the following flag:
  gcloud init --skip-diagnostics

Network diagnostic detects and fixes local network connection issues.
Checking network connection...done.

Reachability Check passed.
Network diagnostic (1/1 checks) passed.

Choose the account you would like to use to perform operations for
this configuration:
[1] papa.marco@gmail.com
[2] Log in with a new account
Please enter your numeric choice: 1

You are logged in as: [papa.marco@gmail.com].

Pick cloud project to use:
[1] facultyinstitute-174928
[2] hello-world-mcpapa
[3] helloworld-183319
[4] myfunctionsprojelab-183319
[5] myfunctionsproject
[6] quickstart-php-183415
[7] quixotic-dynamo-165616
[8] Create a new project
Please enter numeric choice or text value (must exactly match list
item): 5

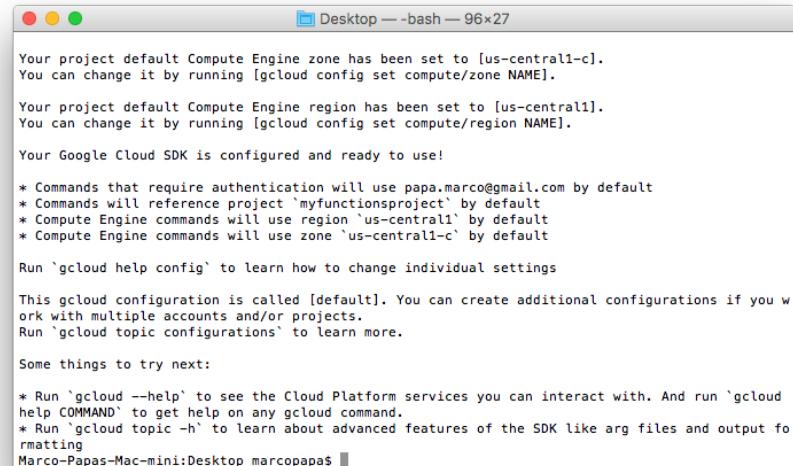
Your current project has been set to: [myfunctionsproject].
```

23. Run **gcloud init** to initialize the SDK:

```
./google-cloud-sdk/bin/gcloud init
```

24. You will be asked to select the project.

25. You maybe asked to “enable” API
[compute.googleapis.com] and “configure”
Google Compute Engine. Answer **Y** to both.



```
Your project default Compute Engine zone has been set to [us-central1-c].
You can change it by running [gcloud config set compute/zone NAME].
Your project default Compute Engine region has been set to [us-central1].
You can change it by running [gcloud config set compute/region NAME].
Your Google Cloud SDK is configured and ready to use!

* Commands that require authentication will use papa.marco@gmail.com by default
* Commands will reference project 'myfunctionsproject' by default
* Compute Engine commands will use region 'us-central1' by default
* Compute Engine commands will use zone 'us-central1-c' by default

Run `gcloud help config` to learn how to change individual settings

This gcloud configuration is called [default]. You can create additional configurations if you w
ork with multiple accounts and/or projects.
Run `gcloud topic configurations` to learn more.

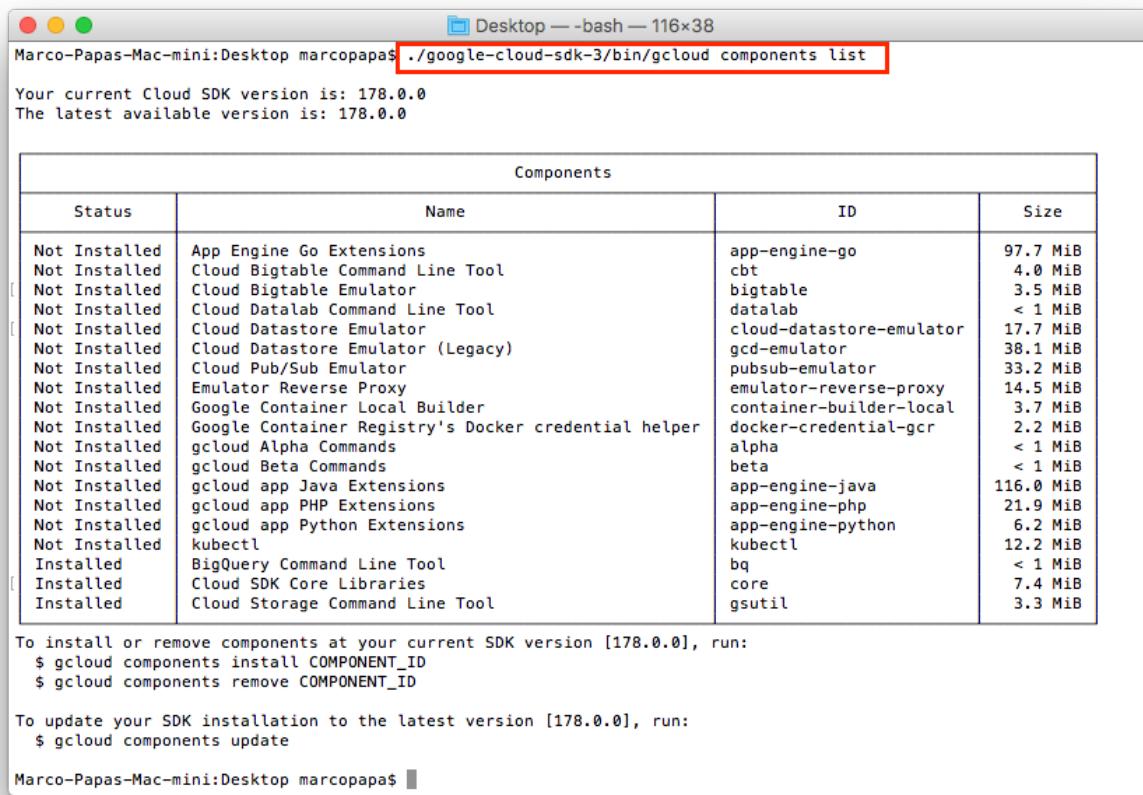
Some things to try next:

* Run `gcloud --help` to see the Cloud Platform services you can interact with. And run `gcloud
help COMMAND` to get help on any gcloud command.
* Run `gcloud topic -h` to learn about advanced features of the SDK like arg files and output fo
rmatting
Marco-Papas-Mac-mini:Desktop marcopapa$
```

Google Cloud Functions (cont'd)

30. Verify all gcloud installed components:

```
./google-cloud-sdk-3/bin/gcloud components list
```



The screenshot shows a terminal window titled "Marco-Papas-Mac-mini:Desktop marcopapas". The command `./google-cloud-sdk-3/bin/gcloud components list` is highlighted with a red box. The output displays the current Cloud SDK version (178.0.0) and the latest available version (178.0.0). A table lists the components, their status, names, IDs, and sizes.

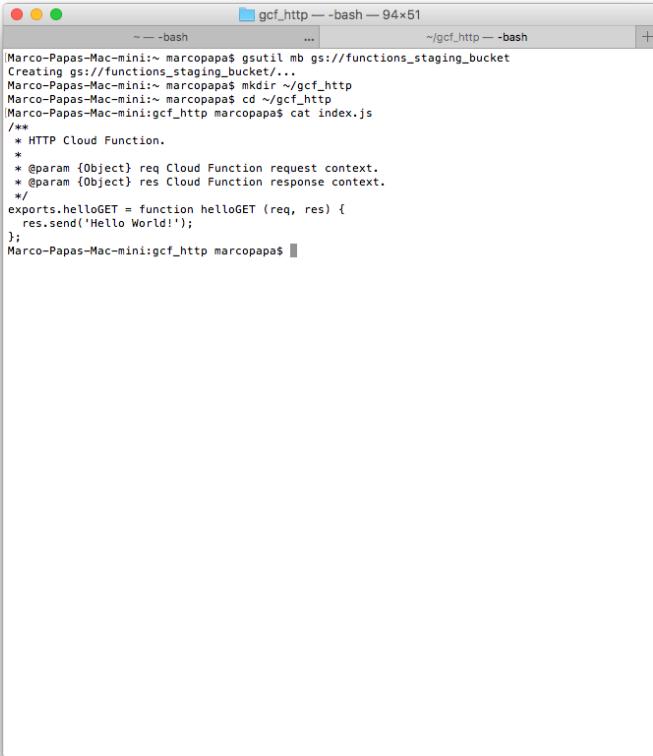
Components			
Status	Name	ID	Size
Not Installed	App Engine Go Extensions	app-engine-go	97.7 MiB
Not Installed	Cloud Bigtable Command Line Tool	cbt	4.0 MiB
Not Installed	Cloud Bigtable Emulator	bigtable	3.5 MiB
Not Installed	Cloud Databab Command Line Tool	dataab	< 1 MiB
Not Installed	Cloud Datastore Emulator	cloud-datastore-emulator	17.7 MiB
Not Installed	Cloud Datastore Emulator (Legacy)	gcd-emulator	38.1 MiB
Not Installed	Cloud Pub/Sub Emulator	pubsub-emulator	33.2 MiB
Not Installed	Emulator Reverse Proxy	emulator-reverse-proxy	14.5 MiB
Not Installed	Google Container Local Builder	container-builder-local	3.7 MiB
Not Installed	Google Container Registry's Docker credential helper	docker-credential-gcr	2.2 MiB
Not Installed	gcloud Alpha Commands	alpha	< 1 MiB
Not Installed	gcloud Beta Commands	beta	< 1 MiB
Not Installed	gcloud app Java Extensions	app-engine-java	116.0 MiB
Not Installed	gcloud app PHP Extensions	app-engine-php	21.9 MiB
Not Installed	gcloud app Python Extensions	app-engine-python	6.2 MiB
Not Installed	kubectl	kubectl	12.2 MiB
Installed	BigQuery Command Line Tool	bq	< 1 MiB
Installed	Cloud SDK Core Libraries	core	7.4 MiB
Installed	Cloud Storage Command Line Tool	gsutil	3.3 MiB

To install or remove components at your current SDK version [178.0.0], run:
\$ gcloud components install COMPONENT_ID
\$ gcloud components remove COMPONENT_ID

To update your SDK installation to the latest version [178.0.0], run:
\$ gcloud components update

Marco-Papas-Mac-mini:Desktop marcopapas\$

Google Cloud Functions (cont'd)



A screenshot of a macOS terminal window titled "gcf_http -- bash". The window shows a command-line session where a user is creating a Cloud Storage bucket for Cloud Functions staging. The terminal output is as follows:

```
Marco-Papas-Mac-mini:~ marcopapa$ gsutil mb gs://functions_staging_bucket
Creating gs://functions_staging_bucket/...
Marco-Papas-Mac-mini:~ marcopapa$ mkdir ~/gcf_http
Marco-Papas-Mac-mini:~ marcopapa$ cd ~/gcf_http
Marco-Papas-Mac-mini:gcf_http marcopapa$ cat index.js
/**
 * HTTP Cloud Function.
 *
 * @param {Object} req Cloud Function request context.
 * @param {Object} res Cloud Function response context.
 */
exports.helloGET = function helloGET (req, res) {
  res.send('Hello World!');
};
Marco-Papas-Mac-mini:gcf_http marcopapa$
```

31. Now prepare the application. Create a Cloud Storage bucket to stage your Cloud Functions files, where [YOUR_STAGING_BUCKET_NAME] is a globally-unique bucket name:

```
gsutil mb gs://[YOUR_STAGING_BUCKET_NAME]
```

As in:

```
gsutil mb gs://functions_staging_bucket
```

32. Create a directory on your local system for the application code:

Linux or Mac OS X:

```
mkdir ~/gcf_http
cd ~/gcf_http
```

Windows:

```
mkdir %HOMEPATH%\gcf_http
cd %HOMEPATH%\gcf_http
```

Google Cloud Functions (cont'd)

The screenshot shows a code editor window titled "index.js". The file content is as follows:

```
1 /**
2  * HTTP Cloud Function.
3  *
4  * @param {Object} req Cloud Function request context.
5  * @param {Object} res Cloud Function response context.
6  */
7 exports.helloGET = function helloGET (req, res) {
8     res.send('Hello World!');
9 };
10
```

The code defines a single function named `helloGET` which sends a "Hello World!" response. The code editor interface includes standard elements like tabs, status bar, and scroll bars.

33. Create an `index.js` file in the `gcf_http` directory with the following contents:

```
/**
 * HTTP Cloud Function.
 *
 * @param {Object} req Cloud
Function request context.
 * @param {Object} res Cloud
Function response context.
 */
exports.helloGET = function
helloGET (req, res) {
    res.send('Hello World!');
}
```

34. The `helloGET` function is exported by the module and is executed when you make an HTTP request to the function's endpoint.

Google Cloud Functions (cont'd)

- 35. Deploying the Function.** To deploy the helloGET function with an HTTP trigger, run the following command in the gcf_http directory:

```
gcloud beta functions deploy helloGET --stage-bucket [YOUR_STAGING_BUCKET_NAME]  
--trigger-http
```

where [YOUR_STAGING_BUCKET_NAME] is the name of your staging Cloud Storage Bucket, as in:

```
gcloud beta functions deploy helloGET --stage-bucket functions_staging_bucket --  
trigger-http  
(you may be asked to install the 'gcloud Beta commands' component. Answer Y.)
```



```
Marco-Papas-Mac-mini:gcf_http marcopapa$ gcloud beta functions deploy helloGET --stage-bucket functions_staging_bucket --trigger-http  
Copying file:///var/folders/zg/9vvzd3p14j71bc1wmn2_bwcr0000gn/T/tmpge0FnE/fun.zip [Content-Type=application/zip]...  
/ [1 files] [ 258.0 B/ 258.0 B]  
Operation completed over 1 objects/258.0 B.  
Waiting for operation to finish...done.  
Deploying function (may take a while - up to 2 minutes)...done.  
availableMemoryMb: 256  
entryPoint: helloGET  
httpsTrigger:  
  url: https://us-central1-myfunctionproject-163116.cloudfunctions.net/helloGET  
latestOperation: operations/bXlmdW5jdGlvbnByb2plY3QtMTYzMTE2L3VzLWNlbmRyYWwxL2hlbgxvR0VUL3puNDVZQkI0Y2hr  
name: projects/myfunctionproject-163116/locations/us-central1/functions/helloGET  
sourceArchiveUrl: gs://functions_staging_bucket/us-central1-helloGET-pdqtwznacin.zip  
status: READY  
timeout: 60s  
updateTime: '2017-03-30T18:26:13Z'  
Marco-Papas-Mac-mini:gcf_http marcopapa$
```

Google Cloud Functions (cont'd)

The screenshot shows the Google Cloud Platform Function details page for a function named "function-1". The "Trigger" tab is selected. Under the "HTTP" section, the "Trigger URL" is listed as <https://us-west2-quixotic-dynamo-165616.cloudfunctions.net/function-1>.

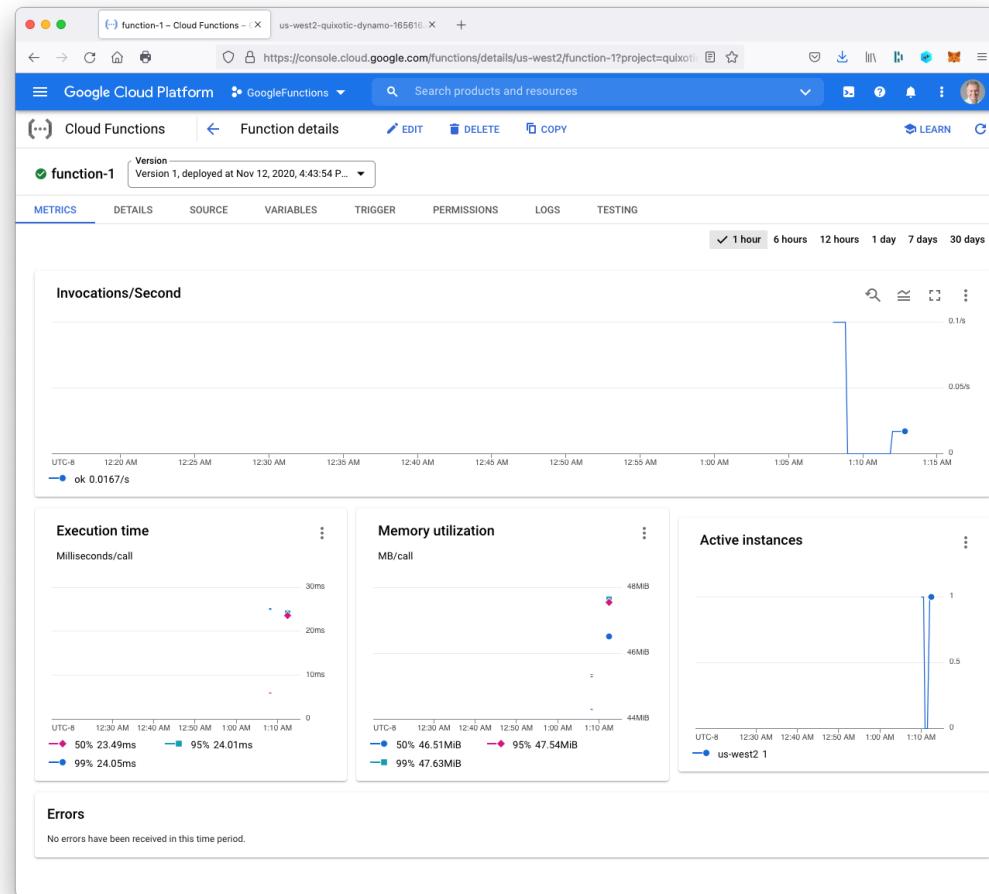
36. **Triggering the function.** Notice of the value of the **Trigger URL** in **TRIGGER** tab.

`https://us-west2-quixotic-dynamo-165616.cloudfunctions.net/function-1`

37. Make an **HTTP request** to your function, using curl or visit the function's endpoint in your browser to see the "Hello World!" message.

The screenshot shows a browser window displaying the response from the function endpoint. The URL is `https://us-west2-quixotic-dynamo-165616.cloudfunctions.net/function-1`. The page content is "Hello World!".

Google Cloud Functions (cont'd)



38. **Monitor the function.** From the Google Cloud Platform menu, select **Cloud Functions** and the **Metrics** tab.

Google Cloud Functions (cont'd)

A screenshot of the Google Cloud Functions console. The URL is https://console.cloud.google.com/functions/details/us-west2/function-1?project=quixotic-dynamo-165616. The page shows a function named 'function-1' with a single version deployed at Nov 12, 2020, 4:43:54 P. The tabs at the top are METRICS, DETAILS, SOURCE, VARIABLES, TRIGGER (which is highlighted with a red box), PERMISSIONS, LOGS, and TESTING. Below the tabs, it says 'HTTP' and shows a 'Trigger URL' of https://us-west2-quixotic-dynamo-165616.cloudfunctions.net/function-1.

A screenshot of the Google Cloud Functions console showing the 'Source' tab. The URL is https://console.cloud.google.com/functions/details/us-west2/function-1?project=quixotic-dynamo-165616. The tabs at the top are METRICS, DETAILS (highlighted with a red box), SOURCE, VARIABLES, TRIGGER, PERMISSIONS, LOGS, and TESTING. It shows the runtime as Node.js 10 and the entry point as helloWorld. The code editor displays index.js and package.json files. The index.js file contains the following code:

```
1 /**
2 * Responds to any HTTP request.
3 *
4 * @param {Object} req HTTP request context.
5 * @param {Object} res HTTP response context.
6 */
7 exports.helloWorld = (req, res) => {
8   let message = req.query.message || req.body.message || 'Hello World!';
9   res.status(200).send(message);
10};
```

A screenshot of the Google Cloud Functions console showing the 'Testing' tab. The URL is https://console.cloud.google.com/functions/details/us-west2/function-1?project=quixotic-dynamo-165616. The tabs at the top are METRICS, DETAILS, SOURCE, VARIABLES, TRIGGER, PERMISSIONS, LOGS, and TESTING (highlighted with a red box). It shows a 'Triggering event' section with a dropdown menu. At the bottom, there is a 'TEST THE FUNCTION' button and a log output area. The log output shows 'Output Complete' with the message '\$ Hello World!' and 'Logs Fetching...'.

Google Cloud Functions (cont'd)

41. Quickstart: “Using the Console” available at: <https://cloud.google.com/functions/docs/quickstart-console>

