Cloud Profiler: Qwik Start

45 minutesFree

GSP209

Overview

Cloud Profiler is a statistical, low-overhead profiler that continuously gathers CPU usage and memory-allocation information from your production

applications. It attributes that information to the application's source code, helping you identify the parts of the application consuming the most resources, and otherwise illuminating the performance characteristics of the code.

In this lab you will learn how to set up and use Stackdriver Profiler. First you'll download a sample Go program and run it with profiling enabled. Then you'll use the Cloud Profiler interface to explore the captured data.

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 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.
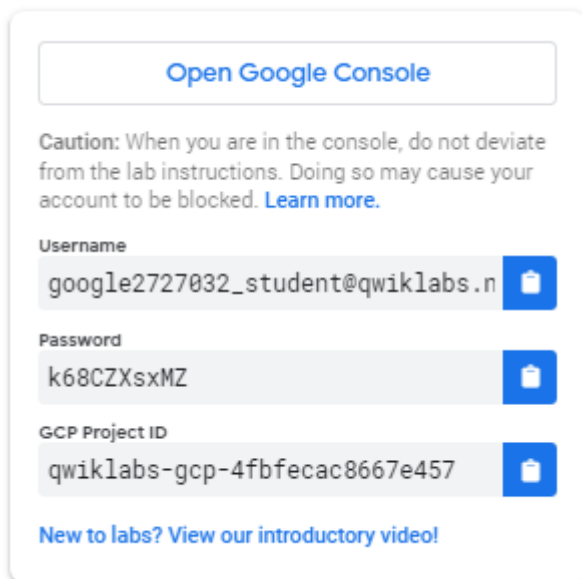
What you need
To complete this lab, you need:

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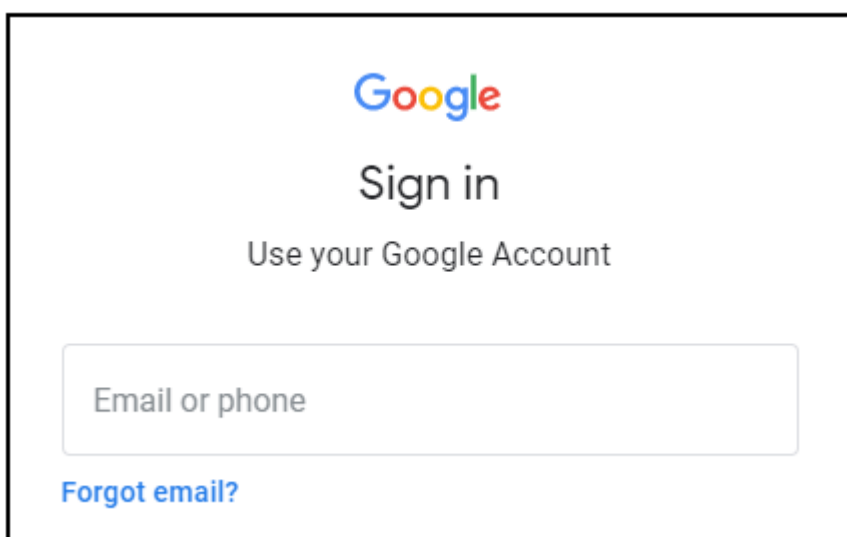
2. Copy the username, and then click **Open Google Console**. The lab spins up resources, and then opens another tab that shows the **Sign in** page.

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

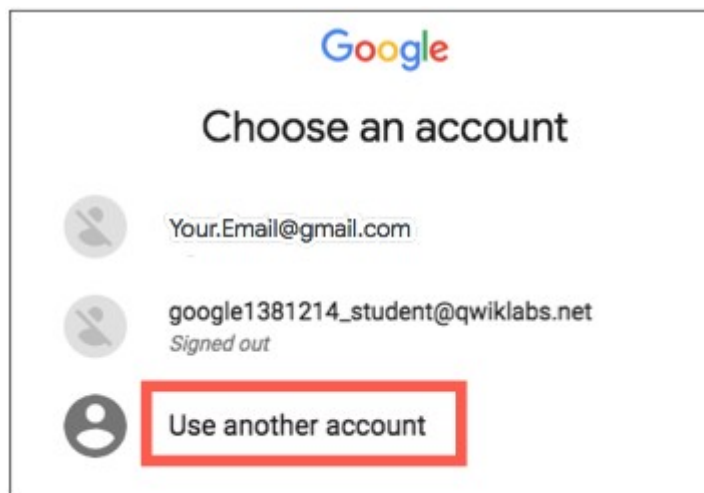**Choose an account** page, click **Use Another**

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-
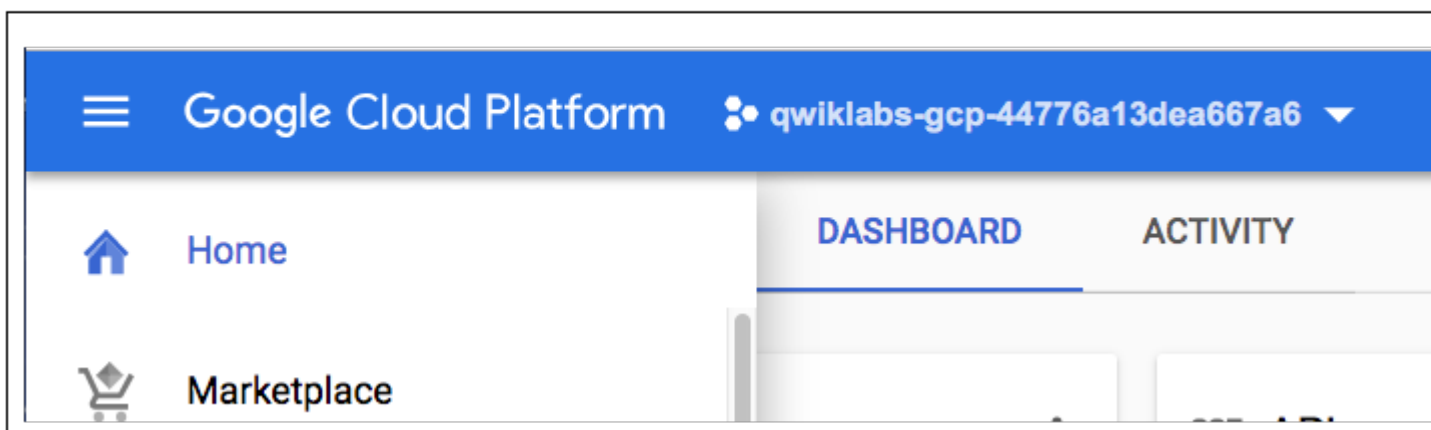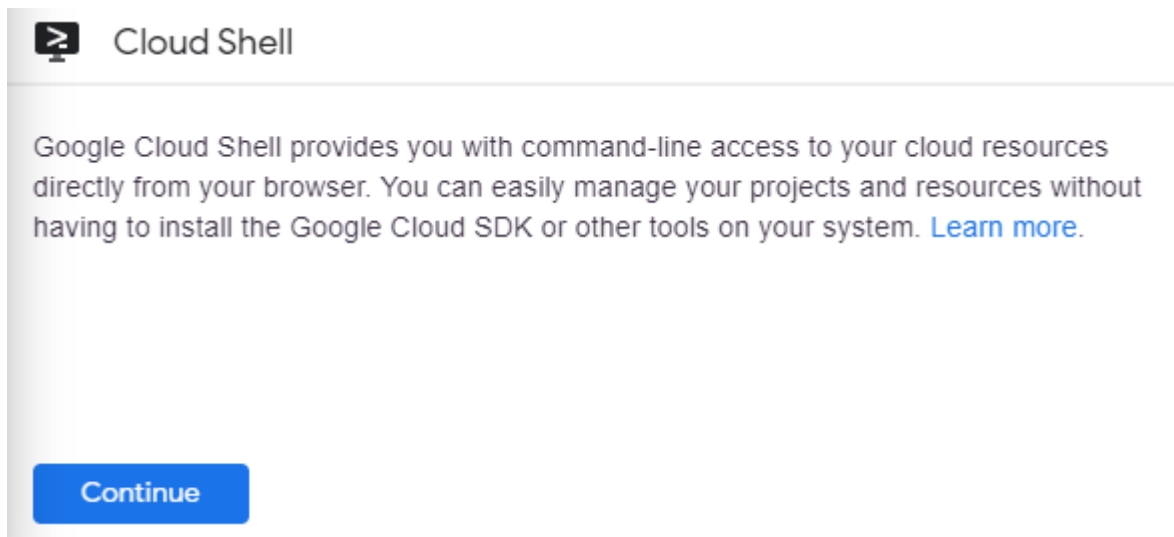

left.
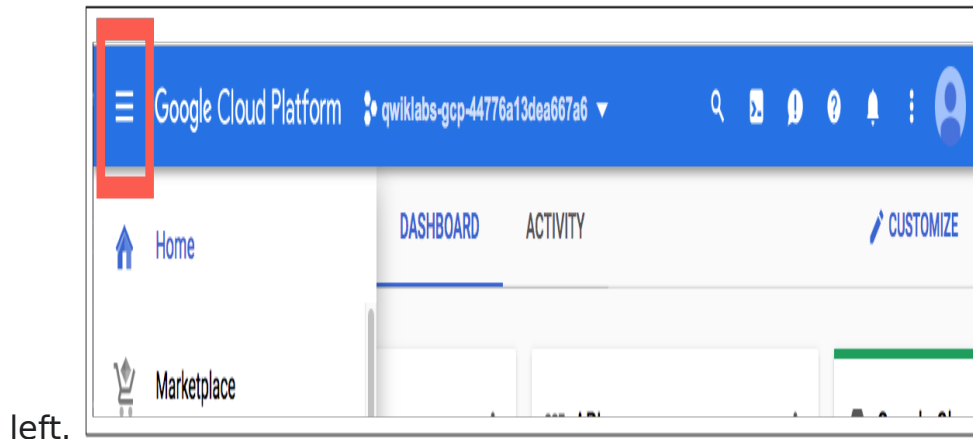
Activate Cloud Shell

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.

In the Cloud Console, in the top right toolbar, click the **Activate Cloud Shell** button.

Click **Continue**.

It takes a few moments to provision and connect to the environment. When you are connected, you are already authenticated, and the project is set to your PROJECT_ID. For example:

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

You can list the active account name with this command:

```
gcloud auth list
content_copy
```
(Output)

```
Credentialed accounts:
 - <myaccount>@<mydomain>.com (active)content_copy
```
(Example output)

```
Credentialed accounts:
 - google1623327_student@qwiklabs.netcontent_copy
```
You can list the project ID with this command:

```
gcloud config list project
content_copy
```
(Output)

```
[core]
project = <project_ID>content_copy
```
(Example output)

```
[core]
project = qwiklabs-gcp-44776a13dea667a6content_copy
```
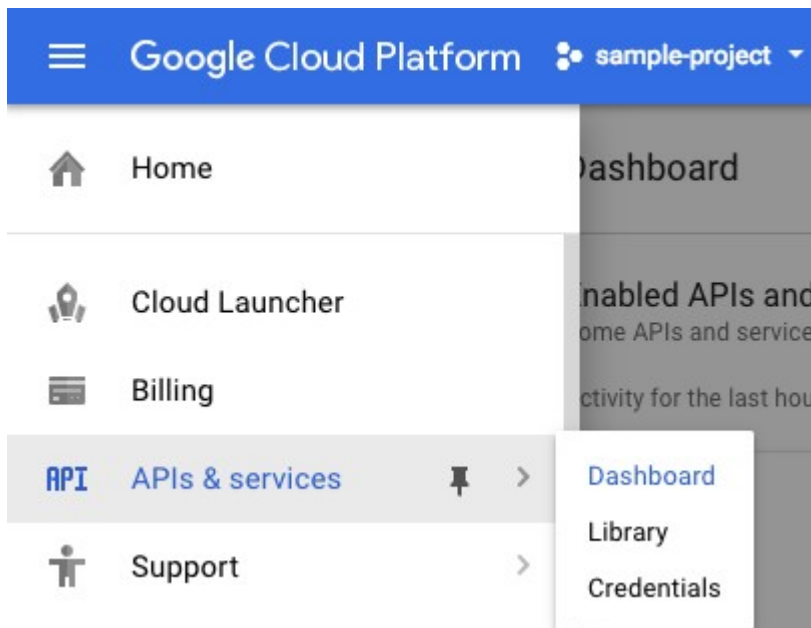 gcloud see the [gcloud command-line tool overview](#).
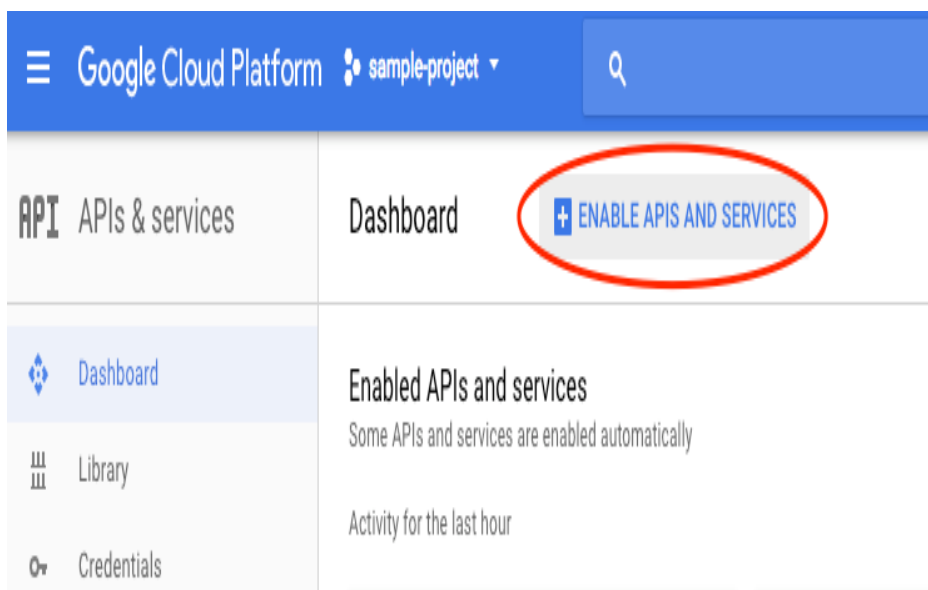
Enable Cloud Profiler API

Click on the menu icon in the top left of the screen.

Select **APIs & services** from the drop down and click on **Dashboard**.

Click on **Enable APIs and services**.

Then, search for **Stackdriver Profiler API** in the search box.

Click on **Stackdriver Profiler API**, then click **Enable** if it is not **Enabled**.

Get a program to profile

The sample program, profiler_quickstart, is in the [golang-samples](#) repository on GitHub. To get it, retrieve the package of Go samples:

```
go get -u github.com/GoogleCloudPlatform/golang-samples/profiler/profiler_quickstart
content_copy
```

Profile the code

The profiler_quickstart program creates a CPU-intensive workload to provide data to the profiler.

Your previous command imported the profiler_quickstart program as a Go module and you are now able to directly run the program.

Run the following to start the program. You can leave it running:

```
profiler_quickstart
content_copy
```

This program is designed to to load the CPU as it runs, and configured to use Cloud Profiler. Cloud Profiler collects profiling data from the program as it runs and periodically saves it. Progress is indicated with a pair of messages:

```
2018/06/19 20:38:18 profiler has started
2018/06/19 20:39:00 successfully created profile CPU
2018/06/19 20:39:11 start uploading profile
2018/06/19 20:40:23 successfully created profile CPU
2018/06/19 20:40:33 start uploading profile
2018/06/19 20:41:15 successfully created profile CPU
2018/06/19 20:41:25 start uploading profile
2018/06/19 20:41:45 successfully created profile CPU
...
content_copy
```
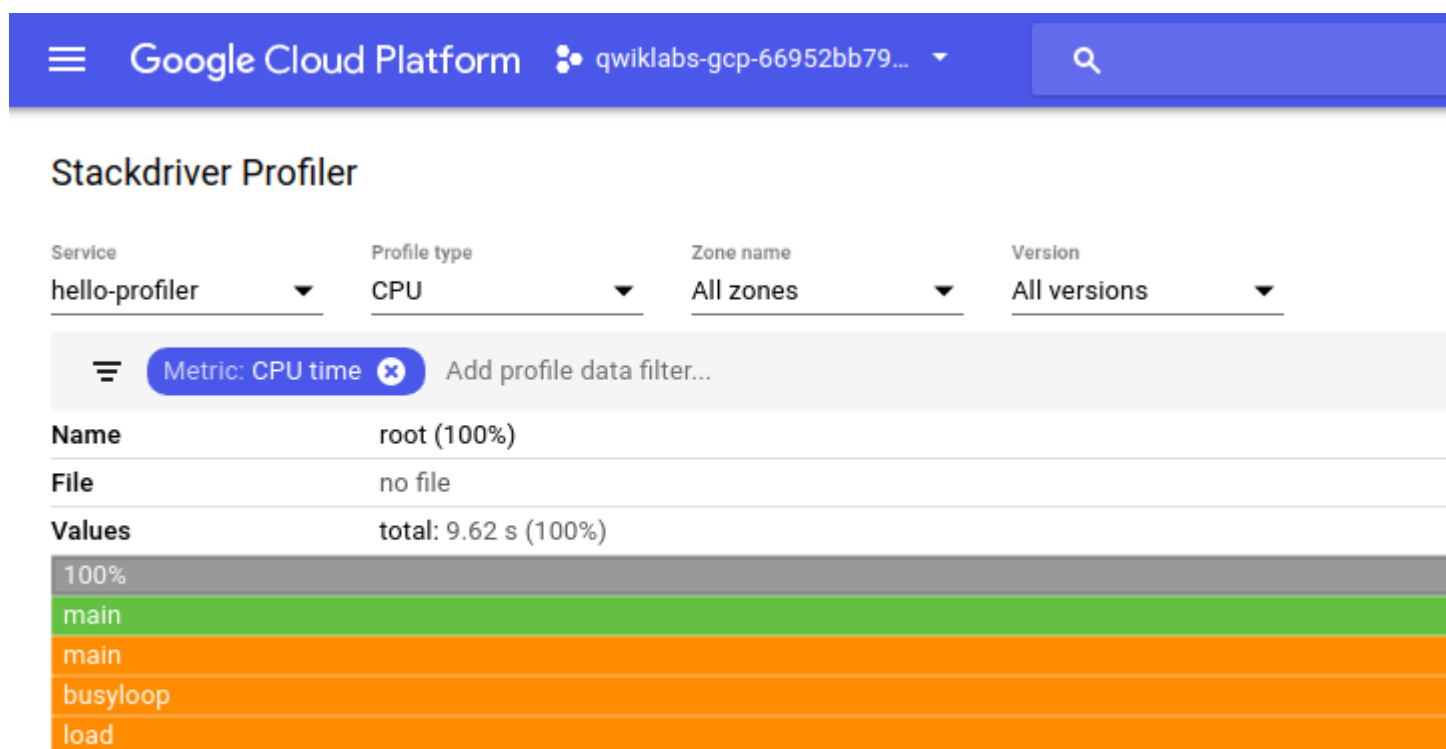
The program will continue to emit these messages while it runs.

Let 2-3 profiles get created, then continue with the lab.

Start the Profiler interface

In the Console, go to Profiler: from the Navigation menu, in the Cloud section, click on **Profiler**. You will be taken to the Profiler interface:

The interface is divided into two general areas:

- A control area for selecting the data to visualize.

- A flame-graph representation of the selected data.

Selecting Profiles

The interface offers an array of controls for exploring the profiling data. At the top of the interface, there are time controls, so you can examine data for the time range you choose.

Below that are options choosing the set of profile data to use:

- **Service** is for selecting the origin of the profiled data, useful if you are profiling several different applications.
- **Profile** type lets you choose the kind of profile data to display.
- **Zone name** and **Version** let you restrict display to data from Compute Engine zones or versions of the application.
 Just below the selectors for Service, Profile, etc. is the filter selector. Filters allow you to refine how the graph displays data. In the screenshot above, the **CPU time** filter is on, so all the CPU time data is displayed.

Exploring the data

Below the selection controls, the selected data is displayed as a flame graph. This type of chart shows you the call stacks in the program. Each function is represented by a frame in the graph, and its relative size shows the proportion of resource consumption that function is responsible for.

The top frame represents the entire program. This frame always shows 100% of the resource consumption, and it indicates how many profiles are averaged together in this graph.

The sample program does not appear to have a complicated set of call stacks; in the preceding screenshot, you see 5 frames:
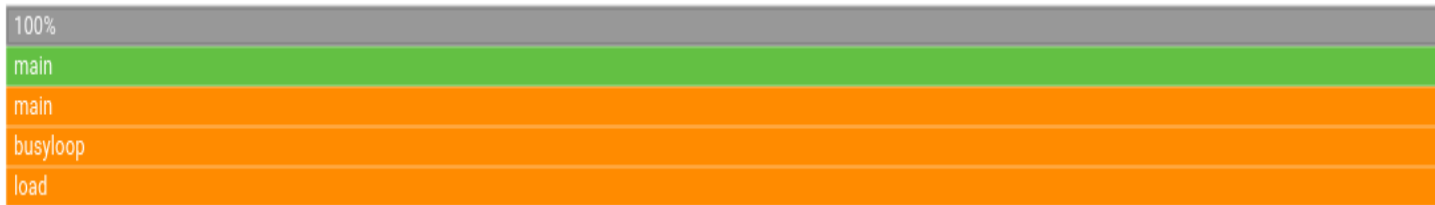

- The gray frame represents the entire executable, which accounts for 100% of the resources being consumed.
- The green main frame is the Go runtime.main.
- The orange main frame is the main routine of the sample program.
- The orange busyloop frame is a routine called from the sample's main.
- The orange main.load frame is a routine called from the sample's main.
  The filter selector lets you do things like filter out functions. For example, if there is a standard library of utility functions, you can remove them from the graph. You can also remove call stacks originating at a certain method, and simplify the graph in other ways.

The profiler_quickstart application is very simple, so there's not much to filter out, but in a complex application, being able to remove elements from the graph is very useful.

Use a filter to hide the call stack from the main routine to let you see what's happening outside main. This extra work accounts for a tiny 0.29% of the resource consumption, but it makes a much more interesting flame graph.

Click next to the CPU time filter to see other available filters.

Select **Hide stacks** then type in "main" as the value. Your flame graph will look something like this:
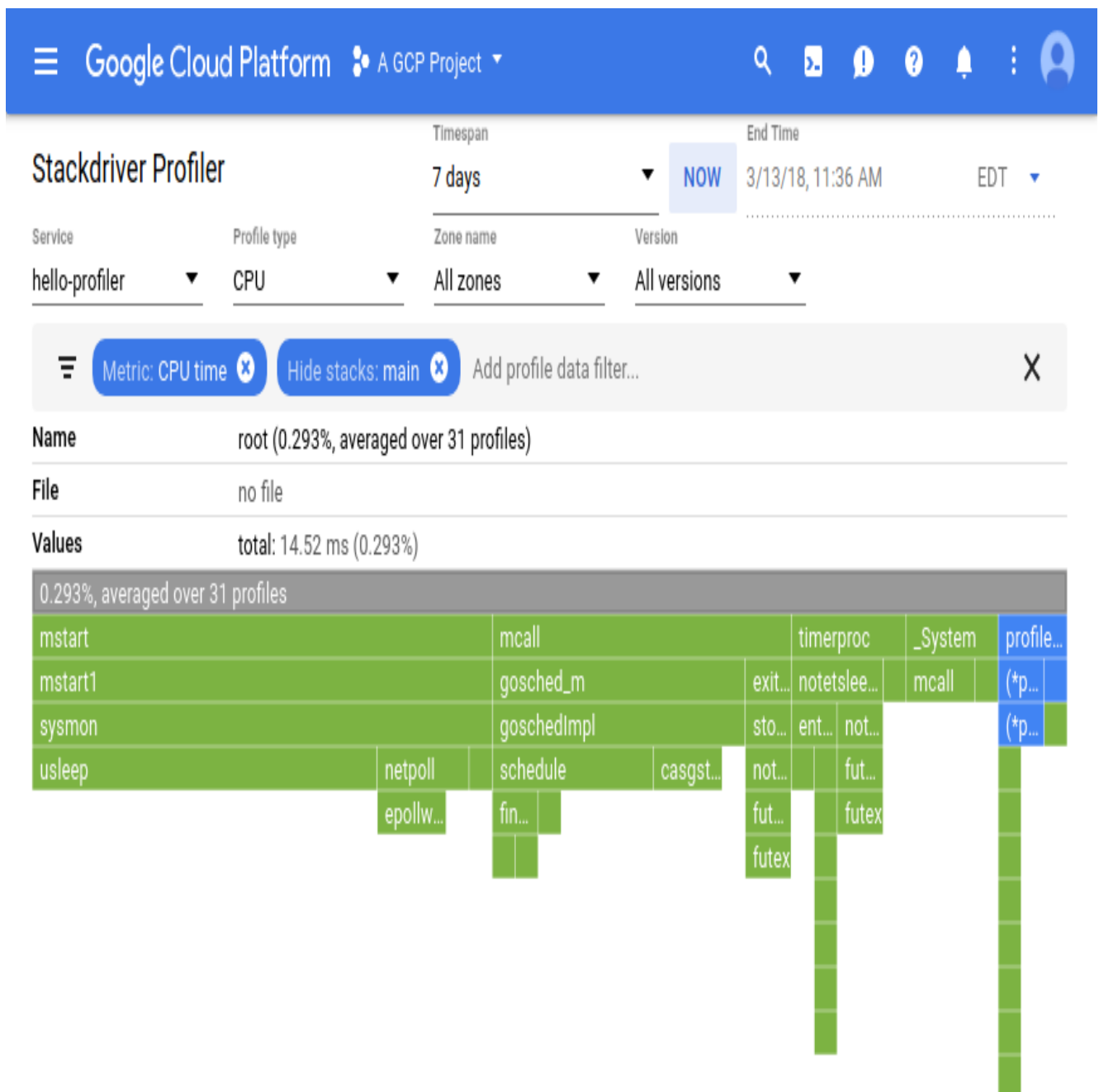
The more profiles that get generated, the more interesting your flame graph gets. In a few minutes refresh the Stackdriver Profiler console to see the graph develop. Wait a few more minutes and do it again.

Congratulations!

You learned how to use Cloud Profiler on Google Cloud.

Finish Your Quest

This self-paced lab is part of the Qwiklabs [Cloud Development](#) Quest. 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 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](#).

Next steps / learn more

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!

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 3, 2021

Lab last tested March 3, 2021