Using Pub/Sub to trigger a Cloud Function

This tutorial shows you how to use Cloud Scheduler and Pub/Sub to trigger a Cloud Function. Being able to schedule the execution of a Cloud Function is a common use case for Cloud Scheduler. You will:

- Create a simple Cloud Function that subscribes to a Pub/Sub topic.
- Create a Pub/Sub topic to trigger that function.
- Create a Cloud Scheduler job that invokes the Pub/Sub trigger.
- Run the Cloud Scheduler job.
- Verify success.

Before you begin

This tutorial assumes you have already worked through the Cloud Scheduler Quickstart (/scheduler/docs/quickstart). As a result, you have the following tools and resources available to you:

- A Google Cloud project with billing enabled.
- An App Engine app located in one of the App Engine supported locations (/appengine/docs/locations).
- The Cloud Scheduler API and the Pub/Sub API both enabled on the project. You enable the Cloud Functions API during the tutorial.

Costs

This tutorial uses billable components of Cloud Platform, including:

- Google Cloud Functions
- Google Cloud Pub/Sub
- Google Cloud Scheduler

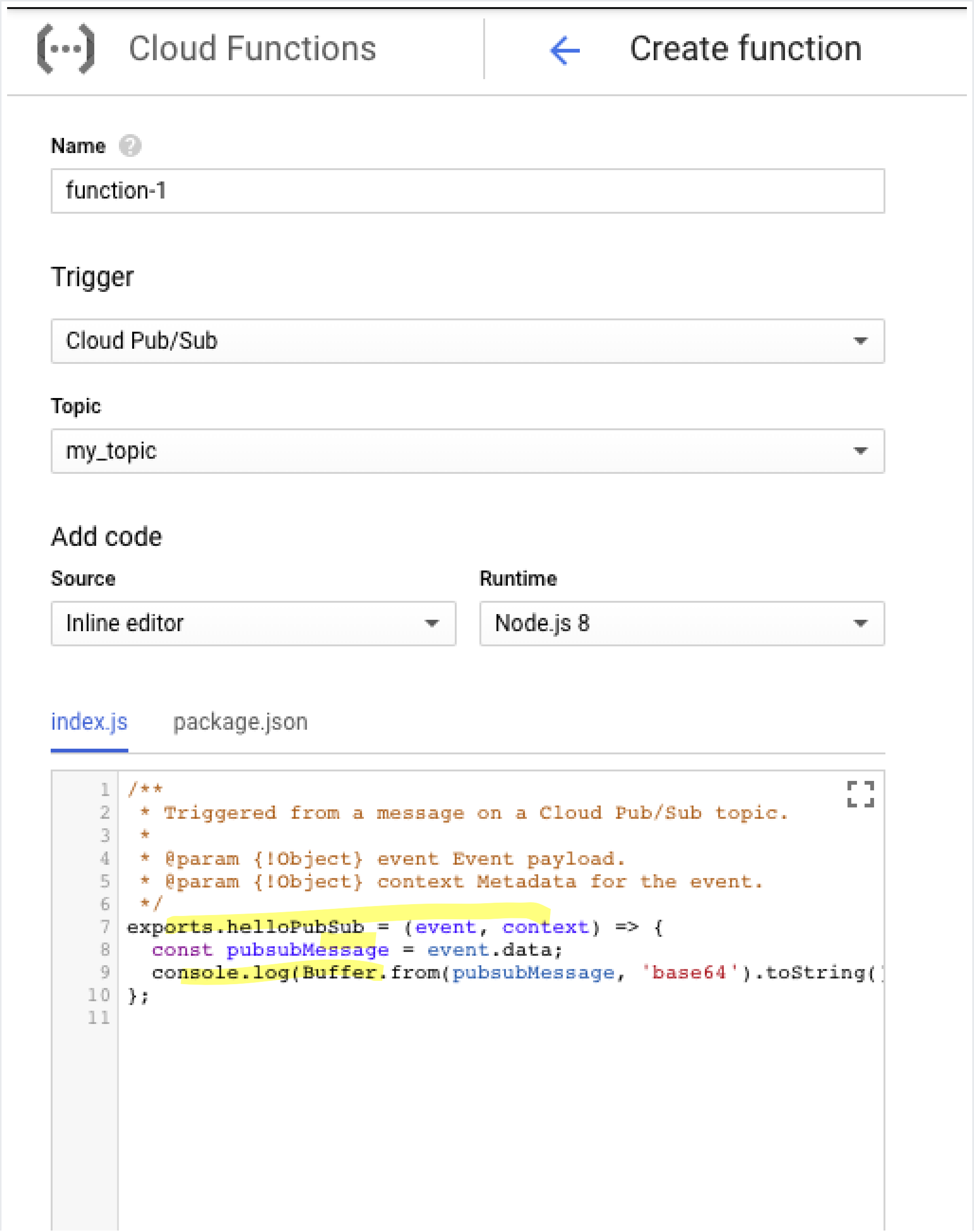
Use the Pricing Calculator (/products/calculator) to generate a cost estimate based on your projected usage.

Create a Cloud Function:

1. Visit the **Cloud Functions** page in the console.

   Go to the Cloud Functions page (https://console.cloud.google.com/functions)

2. Click the **Enable API** button.
3. Click the **Create function** button.



Name

function-1

Trigger

Cloud Pub/Sub

Topic

my_topic

Add code

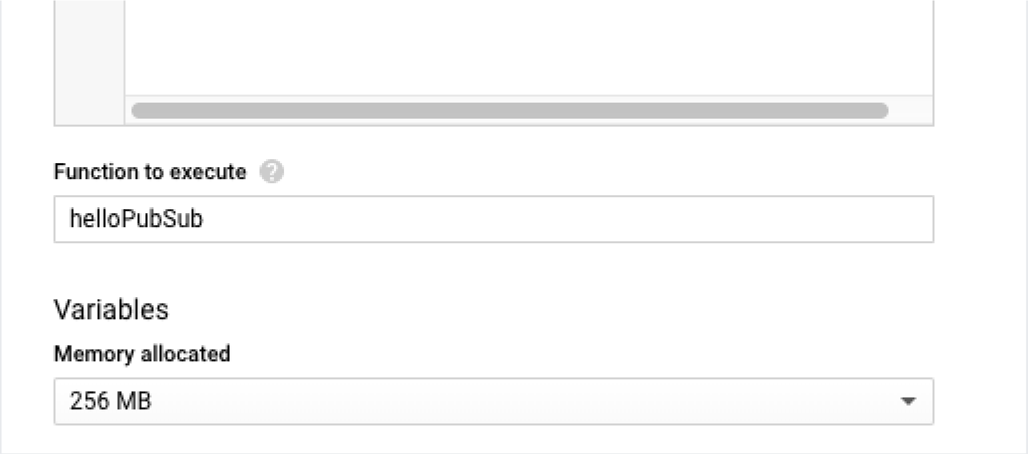
Source

Inline editor

Runtime

Node.js 8

```
/**
 * Triggered from a message on a Cloud Pub/Sub topic.
 *
 * @param {!Object} event Event payload.
 * @param {!Object} context Metadata for the event.
 */
exports.helloPubSub = (event, context) => {
  const pubsubMessage = event.data;
  console.log(Buffer.from(pubsubMessage, 'base64').toString()
};
```



Function to execute

helloPubSub

Variables

Memory allocated

256 MB

4. Give your function a name. Accept the default memory allocation.
5. Select **Cloud Pub/Sub** from the **Trigger** dropdown.
6. Select an existing topic from the dropdown, or use the **Create new topic** pop-up to create a new one. Your function is now subscribed to the topic. Make a note of the topic's name - you will need it later.
7. Accept the defaults for everything else, including the sample `index.js` and `package.json` code.
8. Click **Create**.

Create a Cloud Scheduler job

1. Visit the **Cloud Scheduler** page in the console.

   Go to the Cloud Scheduler page (https://console.cloud.google.com/cloudscheduler)

2. Click the **Create job** button.

3. If necessary, select a region for your job to run in.
4. Give your job a name and optionally add a description.
5. Specify the **frequency** for your job, using the unix-cron (http://man7.org/linux/man-pages/man5/crontab.5.html) format.

   ```
   * * * * *
   ```

   See Configuring Cron Job Schedules (/scheduler/docs/configuring/cron-job-schedules) for more information.
6. Select your timezone.
7. Select **Pub/Sub** from the **Target** dropdown.
8. Enter the **Topic** name you selected above.
9. Add a brief **Payload** string to be sent to your target.
10. Click **Create**.

You now have a cron job that sends a message to your Pub/Sub topic every minute. Your Cloud Function is subscribed to that topic.

Let's run the job you just created.

Run your job

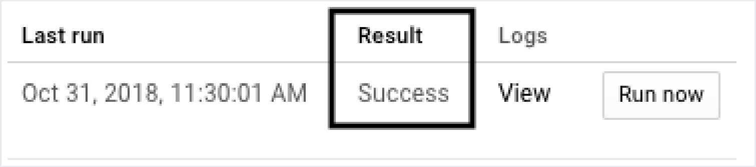
1. Open the Cloud Scheduler console page.

Go to the Cloud Scheduler page (https://console.cloud.google.com/cloudscheduler)

2. Click the **Run now** button.

   The first job created in a project can take a few minutes to run the first time it is invoked, because of required configuration. So you might need to wait a bit.

3. Notice the **Result** column.



Last run	Result	Logs	
Oct 31, 2018, 11:30:01 AM	Success	View	Run now

Congratulations! You have just used the console to see the results of running a cron job that sends a message to Pub/Sub. Next, you'll learn how to verify that your Cloud Function was actually executed.

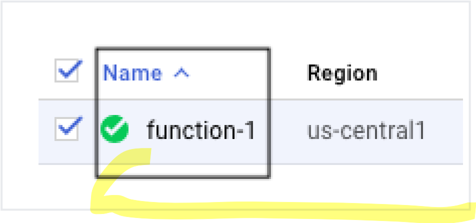
Verify the results in Cloud Functions:

To verify that your Cloud Function is being successfully triggered by your cron job:

1. Visit the **Cloud Functions** page in the console.

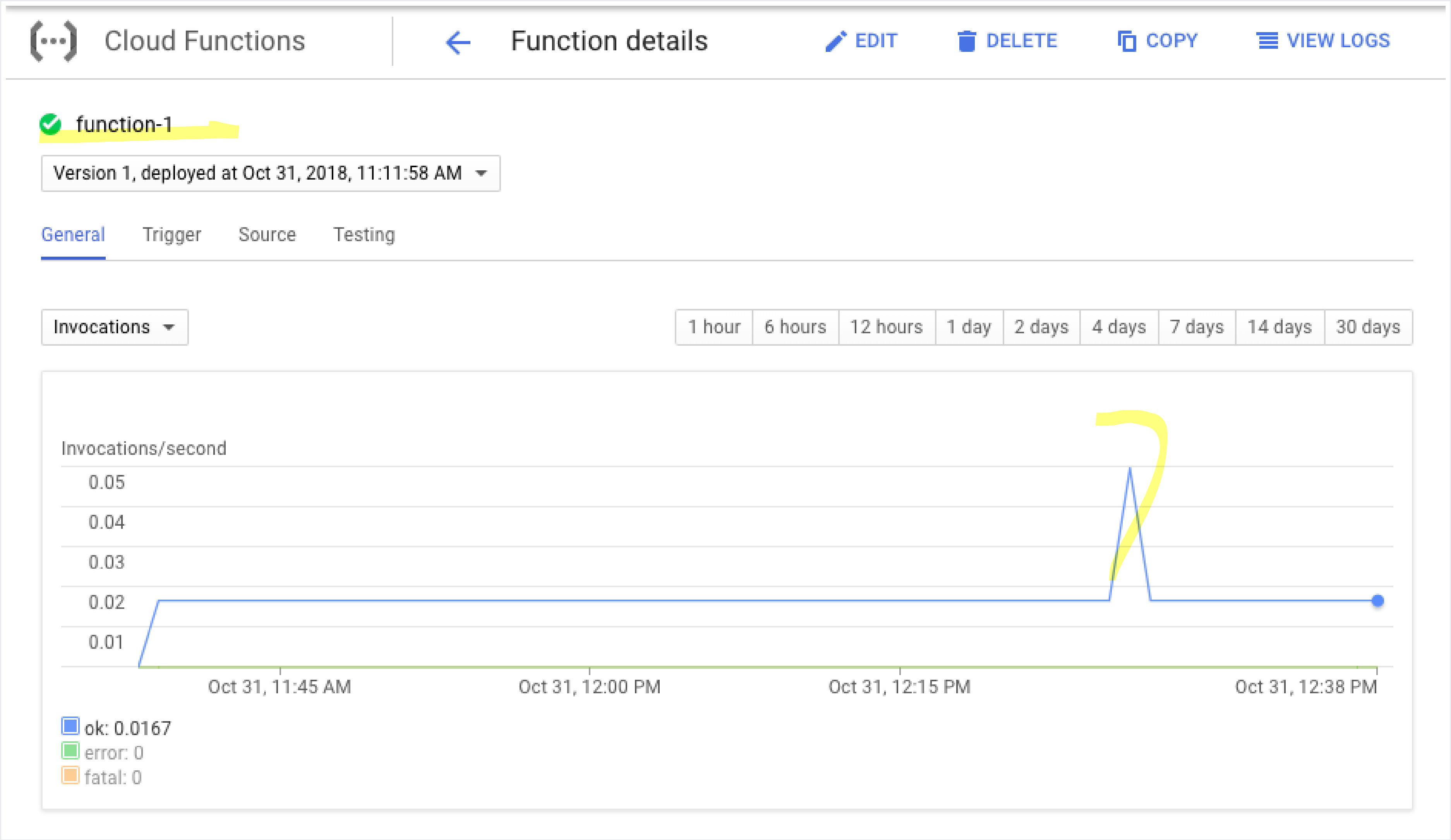
   Go to the Cloud Functions page (https://console.cloud.google.com/functions)

2. Click the function name.



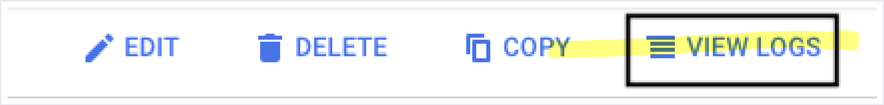
☑	Name ^	Region
☑	✓ function-1	us-central1

3. The **Function details** page opens. Check invocations.

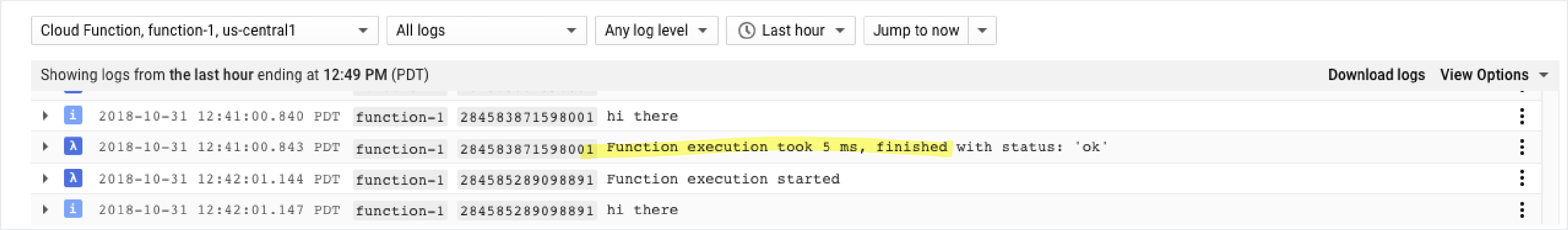


Congratulations! Your function has executed.

4. Click **View Logs**.



5. See the details about the invocation.



Showing logs from **the last hour** ending at **12:49 PM** (PDT)

```
2018-10-31 12:41:00.840 PDT function-1 284583871598001 hi there
2018-10-31 12:41:00.843 PDT function-1 284583871598001 Function execution took 5 ms, finished with status: 'ok'
2018-10-31 12:42:01.144 PDT function-1 284585289098891 Function execution started
2018-10-31 12:42:01.147 PDT function-1 284585289098891 hi there
```

Clean up

To avoid incurring charges to your Google Cloud account for the resources used in this tutorial:

Delete the Cloud Scheduler job

1. Go to the **Cloud Scheduler** page in the Cloud Console.

   Go to the Scheduler page (https://console.cloud.google.com/cloudscheduler)

2. Click the checkbox next to your job.

3. Click the **Delete** button at the top of the page and confirm your delete.

Delete the Pub/Sub topic

1. Go to the **Cloud Pub/Sub** page in the Cloud Console.

   Go to the Pub/Sub page (https://console.cloud.google.com/cloudpubsub/)

2. Click the checkbox next to your topic.
3. Click **Delete** at the top of the page and confirm your delete.

Delete the Cloud Functions function

1. Go to the **Cloud Functions** page in the Cloud Console.

   Go to the Cloud Functions page (https://console.cloud.google.com/functions)

2. Click the checkbox next to your function.
3. Click the **Delete** button at the top of the page and confirm your delete.

Delete the project

If you created a project just for this tutorial.

> **Caution**: Deleting a project has the following effects:
>
> - **Everything in the project is deleted.** If you used an existing project for this tutorial, when you delete it, you also delete any other work you've done in the project.
> - **Custom project IDs are lost.** When you created this project, you might have created a custom project ID that you want to use in the future. To preserve the URLs that use the project ID, such as an `appspot.com` URL, delete selected resources inside the project instead of deleting the

whole project.

If you plan to explore multiple tutorials and quickstarts, reusing projects can help you avoid exceeding project quota limits.

1. In the Cloud Console, go to the **Manage resources** page.

   Go to Manage resources (https://console.cloud.google.com/iam-admin/projects)

2. In the project list, select the project that you want to delete, and then click **Delete**.

3. In the dialog, type the project ID, and then click **Shut down** to delete the project.

Last updated 2021-08-19 UTC.