Cloud Run Functions: Qwik Start - Command Line

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.

1. Click **Activate Cloud Shell** at the top of the Google Cloud console.

2. Click through the following windows:

   - Continue through the Cloud Shell information window.
   - Authorize Cloud Shell to use your credentials to make Google Cloud API calls.

When you are connected, you are already authenticated, and the project is set to your **Project_ID**, `PROJECT_ID`. The output contains a line that declares the **Project_ID** for this session:

```
Your Cloud Platform project in this session is set to "PROJECT_ID"
```

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

3. (Optional) You can list the active account name with this command:

```
gcloud auth list
```

Copied!

content_copy

4. Click **Authorize**.

Output:

```
ACTIVE: *
ACCOUNT: "ACCOUNT"

To set the active account, run:
    $ gcloud config set account `ACCOUNT`
```

5. (Optional) You can list the project ID with this command:

```
gcloud config list project
```

Copied!

content_copy

Output:

```
[core]
project = "PROJECT_ID"
```

Note: For full documentation of `gcloud`, in Google Cloud, refer to the gcloud CLI overview guide.

Task 1. Create a function

First, you're going to create a simple function named `helloWorld`. This function writes a message to the Cloud Run functions logs. It is triggered by Cloud Run function events and accepts a callback function used to signal completion of the function.

For this lab the Cloud Run function event is a pub/sub topic event. A pub/sub is a messaging service where the senders of messages are decoupled from the receivers of messages. When a message is sent or posted, a subscription is required for a receiver to be alerted and receive the message. To learn more about pub/subs, in Pub/Sub Guides, see Pub/Sub: A Google-Scale Messaging Service.
To learn more about the event parameter and the callback parameter, in Cloud Run functions Documentation, see Background Functions.
To create a Cloud Run function:

1. In Cloud Shell, run the following command to set the default region:

   ```
   gcloud config set run/region REGION
   ```

   Copied!

   content_copy

2. Create a directory for the function code:

   ```
   mkdir gcf_hello_world && cd $_
   ```

   Copied!

   content_copy

3. Create and open `index.js` to edit:

```
nano index.js
```

Copied!

content_copy

4. Copy the following into the `index.js` file:

```
const functions = require('@google-cloud/functions-framework');

// Register a CloudEvent callback with the Functions Framework that will
// be executed when the Pub/Sub trigger topic receives a message.
functions.cloudEvent('helloPubSub', cloudEvent => {
  // The Pub/Sub message is passed as the CloudEvent's data payload.
  const base64name = cloudEvent.data.message.data;

  const name = base64name
    ? Buffer.from(base64name, 'base64').toString()
    : 'World';

  console.log(`Hello, ${name}!`);
});
```

Copied!

content_copy

18. Exit nano (Ctrl+x) and save (Y) the file.

19. Create and open `package.json` to edit:

20. Copy the following into the `package.json` file:

```
{
  "name": "gcf_hello_world",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "start": "node index.js",
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "dependencies": {
    "@google-cloud/functions-framework": "^3.0.0"
  }
}
```

Copied!

content_copy

32. Exit nano (Ctrl+x) and save (Y) the file.

33. Install the package dependencies

```
npm install
```

Copied!

content_copy

Expected Output:

```
added 140 packages, and audited 141 packages in 9s

27 packages are looking for funding
run `npm fund` for details

found 0 vulnerabilities
```

Task 2. Deploy your function

For this lab, you'll set the `--trigger-topic` as `cf_demo`.

Note:
Cloud Run functions are event driven, meaning a trigger type must be specified. When deploying a new function, `--trigger-topic`, `--trigger-bucket`, or `--trigger-http` are common trigger events. When deploying an update to an existing function, the function keeps the existing trigger unless otherwise specified.

1. Deploy the **nodejs-pubsub-function** function to a pub/sub topic named **cf-demo**

```
2. gcloud functions deploy nodejs-pubsub-function \
3.   --gen2 \
4.   --runtime=nodejs20 \
5.   --region=REGION \
6.   --source=. \
7.   --entry-point=helloPubSub \
8.   --trigger-topic cf-demo \
9.   --stage-bucket PROJECT_ID-bucket \
10.    --service-account
   cloudfunctionsa@PROJECT_ID.iam.gserviceaccount.com \
     --allow-unauthenticated
```

Copied!

content_copy

Note:
If you get a service account **serviceAccountTokenCreator** notification select **"n"**.

11. Verify the status of the function:

```
12. gcloud functions describe nodejs-pubsub-function \
     --region=REGION
```

Copied!

content_copy

An ACTIVE status indicates that the function has been deployed.

Expected Output:

```
BuildConfig:
  automaticUpdatePolicy: {}
  build: projects/630521560493/locations/REGION/builds/7ff9d415-
50d9-4557-9bcd-5afad42a6390
  dockerRegistry: ARTIFACT_REGISTRY
  dockerRepository:
projects/PROJECT_ID/locations/REGION/repositories/gcf-artifacts
  entryPoint: helloPubSub
...
State: ACTIVE
...
UpdateTime: '2024-08-05T13:51:05.317298824Z'
Url: https://REGION-PROJECT_ID.cloudfunctions.net/nodejs-pubsub-
function
```

Every message published in the topic triggers function execution, the message contents are passed as input data.

Task 3. Test the function

After you deploy the function and know that it's active, test that the function writes a message to the cloud log after detecting an event.

1. Invoke the PubSub with some data.

```
gcloud pubsub topics publish cf-demo --message="Cloud Function
Gen2"
```

Copied!

content_copy

Example output:

```
messageIds:
- '11927162971409664'
```

View logs to confirm that there are log messages with that execution ID.

Task 4. View logs

1. Check the logs to see your messages in the log history:

2.
```
gcloud functions logs read nodejs-pubsub-function \
  --region=REGION
```

Copied!

content_copy

Note:
The logs can take around 10 mins to appear. Also, the alternative way to view the logs is, go to **Logging** > **Logs Explorer**.

The Cloud Run function will output information similar to below:

```
LEVEL:
NAME: nodejs-pubsub-function
EXECUTION_ID: h4v6akxf4sxt
TIME_UTC: 2024-08-05 15:15:25.723
LOG: Hello, Cloud Function Gen2!


LEVEL: I
NAME: nodejs-pubsub-function
EXECUTION_ID:
TIME_UTC: 2024-08-05 15:15:25.711
LOG:

LEVEL:
NAME: nodejs-pubsub-function
EXECUTION_ID: h4oxfjn7zlyu
TIME_UTC: 2024-08-05 15:10:34.303
```

```
LOG: Hello, Friend!


LEVEL: I
NAME: nodejs-pubsub-function
EXECUTION_ID:
TIME_UTC: 2024-08-05 15:10:34.291
LOG:

LEVEL:
NAME: nodejs-pubsub-function
EXECUTION_ID: h4fjhyfxua3k
TIME_UTC: 2024-08-05 15:03:16.342
LOG: Hello, "SGVsbG8gZnJvbSB0aGUgY29tbWFuZCBsaW5l"!
```

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