Streaming IoT Core Data to Dataprep

GSP279

Google Cloud Self-Paced Labs

Overview

In this lab, you configure Cloud IoT Core and Cloud Pub/Sub to create a Pub/Sub topic and registry on Google Cloud. Using a simulated device, you stream data to Cloud Storage, then design a Dataprep flow and use it to analyze data.

Objectives

In this lab, you learn how to perform the following tasks:

- Create Cloud Pub/Sub topics and subscriptions
- Use IoT Core to create a registry
- Start the MQTT Application on a simulator
- Stream data to Cloud Storage
- Use Dataprep to manipulate the 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

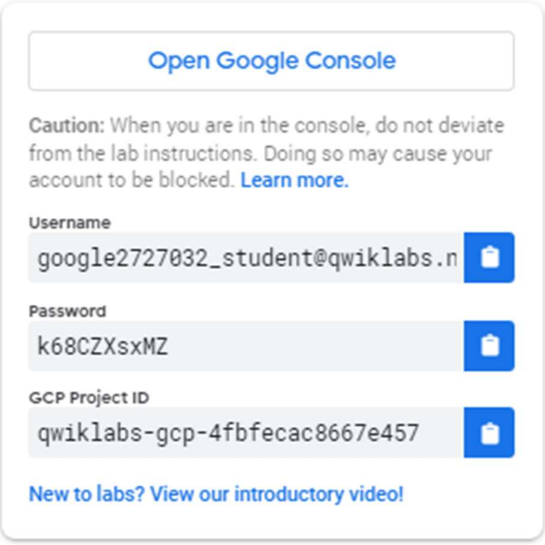
To complete this lab, you need:

- 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 Pixelbook, 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.



Caution: When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked. Learn more.

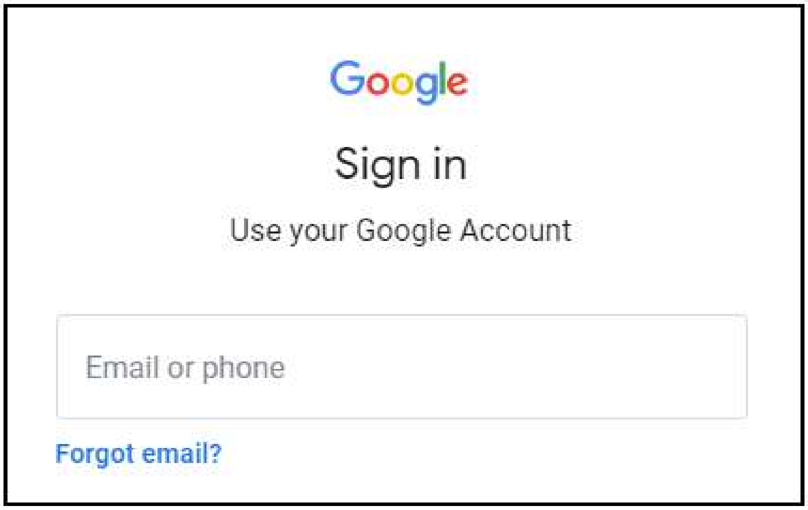
Username
google2727032_student@qwiklabs.n

Password
k68CZXsxMZ

GCP Project ID
qwiklabs-gcp-4fbfecac8667e457

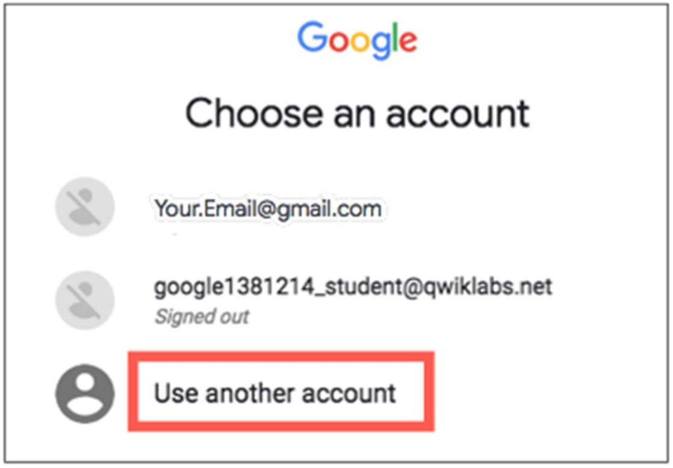
New to labs? View our introductory video!

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.

If you see the **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.

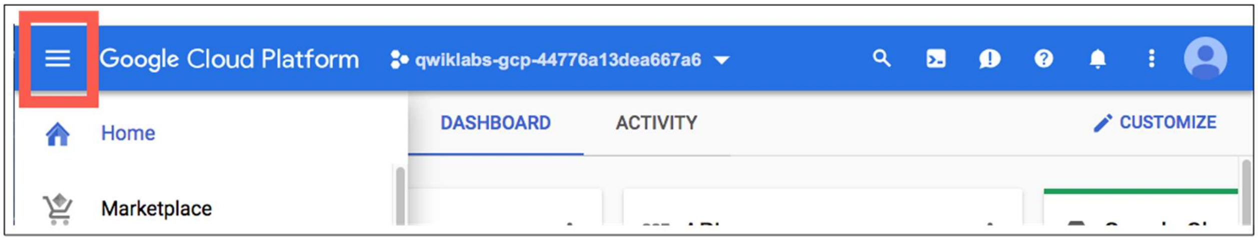
   Important: You must use the credentials from the Connection Details panel. Do not use your Qwiklabs credentials. If you have your own Google Cloud account, do not use it for this lab (avoids incurring charges).

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.

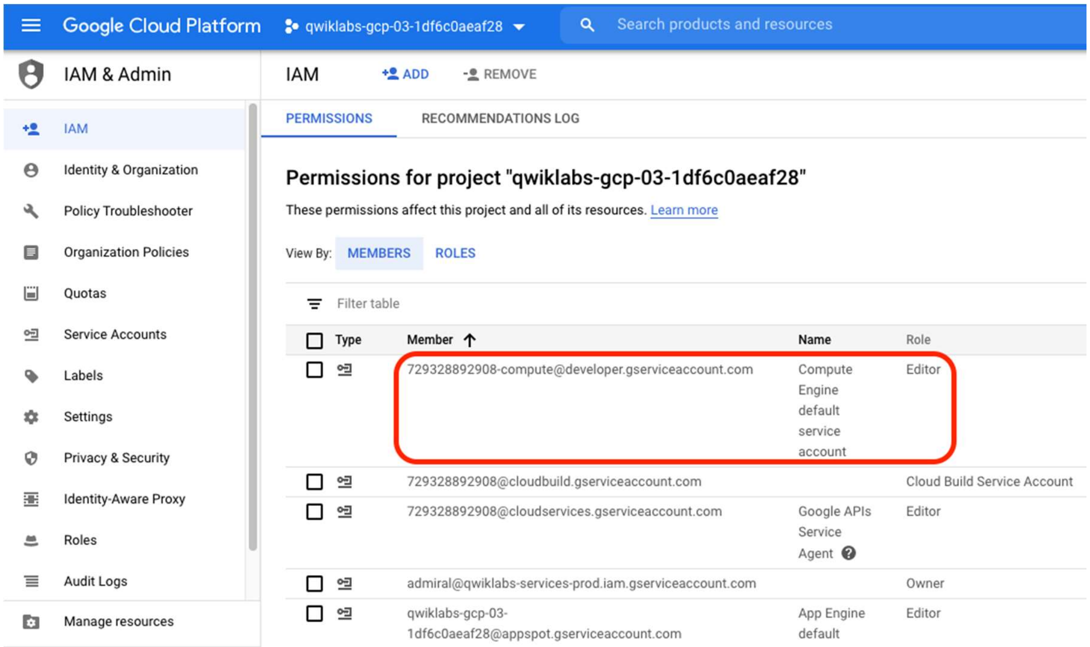


Check project permissions

Before you begin your work on Google Cloud, you need to ensure that your project has the correct permissions within Identity and Access Management (IAM).

1. In the Google Cloud console, on the **Navigation menu** (☰), click **IAM & Admin** > **IAM**.

2. Confirm that the default compute Service Account `{project-number}-compute@developer.gserviceaccount.com` is present and has the `editor` role assigned. The account prefix is the project number, which you can find on **Navigation menu** > **Home**.



Permissions for project "qwiklabs-gcp-03-1df6c0aeaf28"

These permissions affect this project and all of its resources. Learn more

□	Type	Member ↑	Name	Role
□		729328892908-compute@developer.gserviceaccount.com	Compute Engine default service account	Editor
□		729328892908@cloudbuild.gserviceaccount.com		Cloud Build Service Account
□		729328892908@cloudservices.gserviceaccount.com	Google APIs Service Agent	Editor
□		admiral@qwiklabs-services-prod.iam.gserviceaccount.com		Owner
□		qwiklabs-gcp-03-1df6c0aeaf28@appspot.gserviceaccount.com	App Engine default	Editor

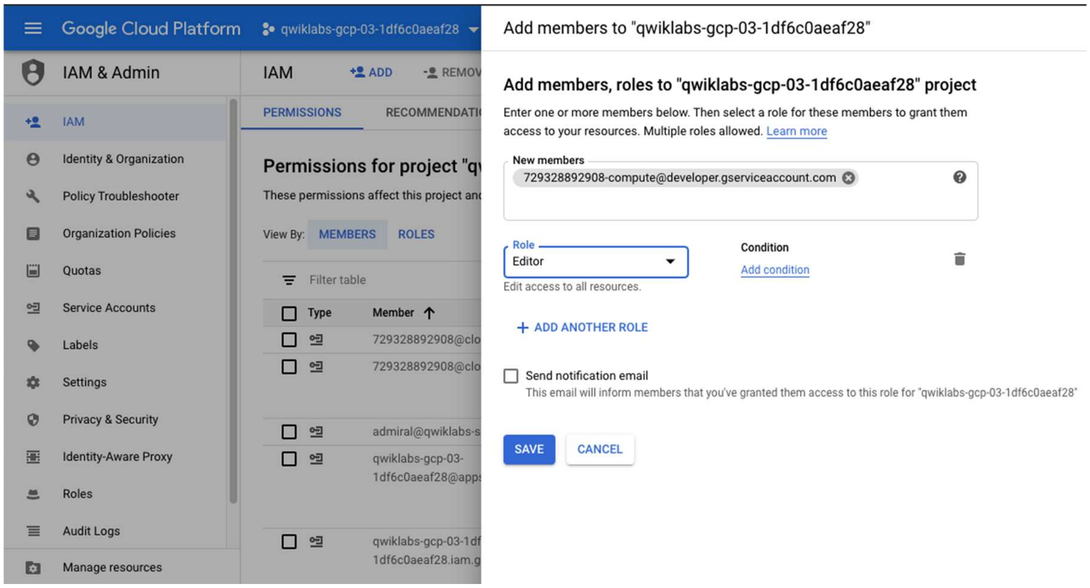
If the account is not present in IAM or does not have the `editor` role, follow the steps below to assign the required role.

- In the Google Cloud console, on the **Navigation menu**, click **Home**.
- Copy the project number (e.g. `729328892908`).
- On the **Navigation menu**, click **IAM & Admin** > **IAM**.
- At the top of the **IAM** page, click **Add**.
- For **New members**, type:

```
{project-number}-compute@developer.gserviceaccount.com
```

Replace `{project-number}` with your project number.

- For **Role**, select **Project** (or Basic) > **Editor**. Click **Save**.



Add members to "qwiklabs-gcp-03-1df6c0aeaf28"

Add members, roles to "qwiklabs-gcp-03-1df6c0aeaf28" project

Enter one or more members below. Then select a role for these members to grant them access to your resources. Multiple roles allowed. Learn more

New members: 729328892908-compute@developer.gserviceaccount.com

Role: Editor

Edit access to all resources.

Condition: Add condition

□ Send notification email

This email will inform members that you've granted them access to this role for "qwiklabs-gcp-03-1df6c0aeaf28"

Cloud Pub/Sub setup and topics

Create a pub/sub topic for your streaming data.

1. On the **Navigation menu**, click **Pub/Sub** > **Topics**.

2. Click **CREATE TOPIC**.

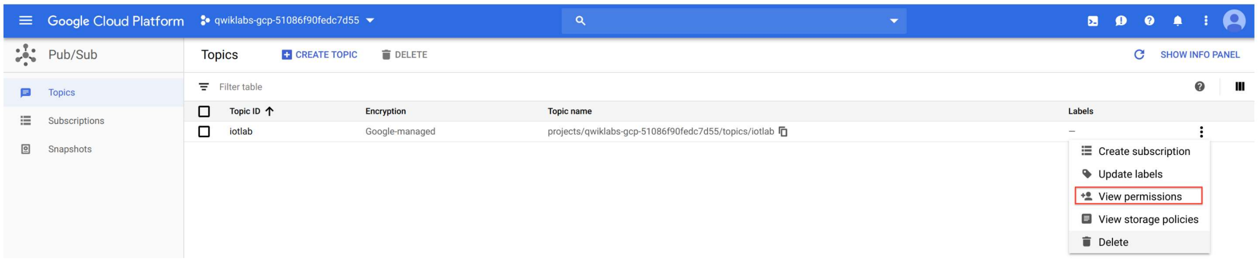
3. Type **iotlab** in Topic ID field.

4. Click **CREATE TOPIC**.

Setting topic permissions

You now have a pub/sub topic. To allow the project to publish this topic, add the project as a member/publisher.

1. Click the overflow menu next to the topic name, and then click **View permissions**.



□	Topic ID ↑	Encryption	Topic name	Labels
□	iotlab	Google-managed	projects/qwiklabs-gcp-51086f90fedc7d55/topics/iotlab	–

2. To add members, click **ADD MEMBER**.

3. Add the project as a member to the topic `cloud-iot@system.gserviceaccount.com`

4. Select the role of **Pub/Sub** > **Pub/Sub Publisher**, and then click **SAVE** to add the member.

Click **Check my progress** to verify the objective.

Create a pub/sub topic

Check my progress

Create a location for data storage

You need to create a storage folder to store the data streaming from the virtual device.

Create a storage bucket

1. On the **Navigation menu** click **Cloud Storage** > **Browser**.
2. Click **CREATE BUCKET**.
3. Enter a unique bucket name for your bucket, then click **CREATE**.

Click **Check my progress** to verify the objective.

Create a Cloud Storage bucket.

Check my progress

Create a folder in your bucket

1. Click **CREATE FOLDER** in the bucket you just created.
2. For folder name, type **Sensor-Data**, and then click **CREATE**.

Click **Check my progress** to verify the objective.

Create Sensor-Data folder in Cloud Storage bucket

Check my progress

Start a Dataflow job

You now have a device publishing data, and your Google Cloud Project is authorized to receive this data. Now you can start a Dataflow job to save the data to your bucket.

Create a Dataflow job from a template

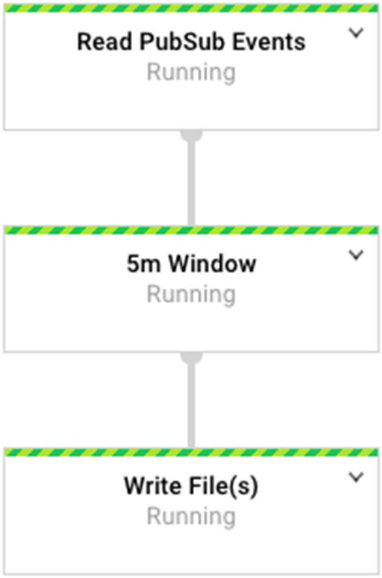
1. On the **Navigation menu** click **Dataflow**.
2. Click **CREATE JOB FROM TEMPLATE** at the top of the screen.
3. Enter the following values in the template.

Property	Value (type value or select option as specified)
Job name	sensor-data
Regional endpoint	us-central1
Dataflow template	Pub/Sub to Text Files on Cloud Storage

4. The template page will expand to display a series of textboxes. Some of the textboxes are optional and some are required. You will only modify the required textboxes. Be sure to replace `<project-id>` with your Google Cloud project ID and `<bucket-name>` with the name of the bucket you created.

Property	Value (type value or select option as specified)
Input Pub/Sub topic	projects/`project-id`/topics/iotlab
Output file directory in Cloud Storage	gs://`bucket-name`/Sensor-Data/ (note the slash at the end of the input text)
Output filename prefix	output-
Temporary Location	gs://bucket-name/tmp

5. Click **RUN JOB**. A Dataflow job will be kicked off—your console should now resemble the following:



Click **Check my progress** to verify the objective.

Set up a Cloud Dataflow Pipeline

Check my progress

Prepare your VM

In your project a pre-provisioned VM instance named **iot-device-simulator** will let you run instances of a Python script that emulate an MQTT-connected IoT device. Before you emulate the devices, you will also use this VM instance to populate your Cloud IoT Core device registry.

Connect to the iot-device-simulator VM instance

1. In the Cloud Console, go to **Navigation menu** > **Compute Engine** > **VM Instances**. You'll see your VM instance listed as **iot-device-simulator**.

2. To the right, click the **SSH** drop-down arrow and select **Open in browser window**.

3. In your SSH session on **iot-device-simulator**, enter this command to remove the default Google Cloud SDK installation. (In subsequent steps, you will install the latest version, including the beta component.)

```
sudo apt-get remove google-cloud-sdk -y
```

4. Now install the latest version of the Google Cloud SDK and accept all defaults:

```
curl https://sdk.cloud.google.com | bash
```

5. Press **Enter** to install your directory, then type "n" to skip reporting your data.

6. Type "Y" to continue, then **Enter** to update your path.

7. End your ssh session on the **iot-device-simulator** VM instance:

```
exit
```

8. Start another SSH session in the **iot-device-simulator** VM instance by clicking the **SSH** drop-down arrow and then selecting **Open in browser window**.

9. In your SSH session, run the following command to initialize the latest version of the gcloud SDK:

```
gcloud init
```

If you get the error message "Command not found", you might have forgotten to exit your previous SSH session and start a new one.

10. When you are asked whether to authenticate with an @developer.gserviceaccount.com account or to log in with a new account, type in the number **2** to **log in with a new account**.

11. When you are asked "Are you sure you want to authenticate with your personal account? Do you want to continue (Y/n)?" enter **Y**.

12. Copy the URL outputted and paste it in a new browser tab and hit **Enter**.

13. Select the account you logged into this lab with, then click **Allow**.

14. Copy the verification code and return to the SSH window where you last left off (There will be a prompt to "Enter verification code:".) Paste in the verification code and press **Enter**.

15. In response to "Pick cloud project to use", enter in the number that corresponds to the Google Cloud project that Qwiklabs created for you (do not pick "Qwiklabs Resources"!)

16. Make sure that the components of the SDK are up to date by running the following:

```
gcloud components update
```

You should receive the following output:

```
All components are up to date.
```

17. Enter the following command to install the beta components:

```
gcloud components install beta
```

Type **Y** to continue.

18. Enter this command to update the system's information about Debian Linux package repositories:

```
sudo apt-get update
```

19. Enter this command to make sure that various required software packages are installed:

```
sudo apt-get install python3-pip openssl git -y
```

20. Use **pip** to add needed Python components:

```
sudo pip3 install pyjwt paho-mqtt cryptography
```

21. Enter this command to add data to analyze during this lab:

```
git clone https://github.com/cagamboa123/training-data-analyst.git
```

22. Now run the following command to set your Project ID as an environment variable, replacing `<YOUR_PROJECT_ID>` with your Qwiklabs Project ID:

```
export PROJECT_ID=<YOUR_PROJECT_ID>
```

23. You must choose a region for your IoT registry. At the present time, these regions are supported:

- us-central1
- europe-west1
- Asia-east1

To set an environment variable containing your preferred region, enter the following command replacing `<YOUR_REGION>` with the region nearest to you:

```
export MY_REGION=<YOUR_REGION>
```

Leave this session open, you will return to it shortly.

Open IoT Core

1. Back in the Console, open the **Navigation menu** and click **IoT Core**.
2. Click **CREATE REGISTRY**.
3. On the **Create a registry** page, specify the following and leave the remaining settings as their defaults:

Field	Value
Registry ID	iotlab-registry
Region	The region closest to you
Select a Cloud Pub/Sub topic	projects/`project-id`/topics/iotlab

4. Click **CREATE**.

Click **Check my progress** to verify the objective.

Create the device registry

Check my progress

Create a Cryptographic KeyPair

To allow IoT devices to connect securely to Cloud IoT Core, you must create a cryptographic KeyPair.

Return to your SSH session in the **iot-device-simulator** VM instance and enter in the following commands to create a key-pair in the appropriate directory:

```
cd $HOME/training-data-analyst/quests/iotlab/
openssl req -x509 -newkey rsa:2048 -keyout rsa_private.pem \
    -nodes -out rsa_cert.pem -subj "/CN=unused"
```

This `openssl` command creates an RSA cryptographic keypair and writes it to a file called `rsa_private.pem`.

Create the device and add it to the registry

1. In your SSH session in the **iot-device-simulator** VM instance, type:

```
cat rsa_cert.pem
```

2. Select and copy the entire certificate. Including all dashes at the beginning and end of the certificate.

Output example (do not copy)

```
-----BEGIN CERTIFICATE-----
MIIC+DCCAeCgAwIBAgIJAOJikTScq9oPMA0GCSqGSIb3DQEBCwUAMBExDzANBgNV
BAMMBnVudXNlZDAeFw0xODA4MTMxNjQ2MTNaFw0xODA5MTIxNjQ2MTNaMBExD
zANBgNVBAMMBnVudXNlZDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggE
BAL+wLyITE5TjlH50I63ew3HdvoGty2aOpP04nMyOYZoooAw5o2rj5mkNb/hbkoMTkzo
6/5Jo0zgDYPVRpz2nGAhTfeQzPuvOfPZe7KPpZxYvmSN3pYT9kkiVo9pXwynG7q8kW
72Q9f0pffXS/VElPrC63Y9kcAgOyveZVX61qSokz4DVIj0Z6+1b1utxe2TnxR1q3Hce289
1re6qnxYp6Yuw0gVYtn8HdgEKKMqeSozqJP7dq8EvNkwY8BAUFU2NmuvwK2Z6hB1E
u0DImyhtKRxZ4pUbWuefC+P6GU2fB3rp4pR9Lc7xd5BuWXHgR6f0lV57elL9f1Q/iXippP
8RjhMCAwEAAaNTMFEwHQYDVR0OBBYEFF7808W+vP7vbgg6cS5Fky9xCstNMB8GA
1UdIwQYMBaAFF7808W+vP7vbgg6cS5Fky9xCstNMA8GA1UdEwEB/wQFMAMBAf8w
DQYJKoZIhvcNAQELBQADggEBAD9mSbWQRz8QHI947gGSMrsA+aO4dgWIujkypFw/p
7gSefleCCwGV4Wpfq6zoIjru9bnciWRLHZMKVbhptBDseyBnoPXxnJMgVYBAVzRRMhT
qPeo146Pv99dn3c310M2tkpQeQzP/wE9XFVqEud2sZCKXgXtydIsyTEX3wmG9s9m7f
6TJDknvJltOj1R7m+xO6GHPebK29x/r+LzPuYjIDYoG+mxLQUltDOM3v8QwZ4bneo+HI
BZX6FOBRb+x/fEE3EANCY3J5sKwRCxxXJ6l/Mts7aLUE6MrT8BM0n1fxnY7BX+6dvsJ
H/OeONG2tk3Y0ci/ly245NQyurqa3x35Ws=
-----END CERTIFICATE-----
```

3. Go back to the IoT Core tab click on **Devices** then **CREATE A DEVICE**.

4. In the **Device ID** field, enter **temp-sensor-buenos-aires**.

5. Expand **COMMUNICATION, CLOUD LOGGING, AUTHENTICATION** and in the Authentication section, set the fields as following.

Field	Value
Input method	Enter manually
Public key format	RS256_X509
Public key value	Paste the certificate number you copied

6. Click **CREATE**.
7. Now add a second device. Use the back arrow in IoT Core to return to the Devices details page.
8. Click **CREATE A DEVICE**.
9. In the **Device ID** field, enter **temp-sensor-istanbul**.
10. Expand **COMMUNICATION, CLOUD LOGGING, AUTHENTICATION** and in the Authentication section, set the fields as following.

Field	Value
Input method	Enter manually
Public key format	RS256_X509
Public key value	Paste the certifcate number you copied

11. Click **CREATE**.

Click **Check my progress** to verify the objective.

Run simulated devices

1. In your SSH session on the **iot-device-simulator** VM instance, enter these commands to download the CA root certificates from pki.google.com to the appropriate directory:

```
cd $HOME/training-data-analyst/quests/iotlab/
wget https://pki.google.com/roots.pem
```

2. Enter this command to run the first simulated device:

```
python3 cloudiot_mqtt_example_json.py \
   --project_id=$PROJECT_ID \
   --cloud_region=$MY_REGION \
   --registry_id=iotlab-registry \
   --device_id=temp-sensor-buenos-aires \
   --private_key_file=rsa_private.pem \
   --message_type=event \
   --algorithm=RS256 --num_messages=1000 > buenos-aires-log.txt 2>&1 &
```

It will continue to run in the background.

3. Enter this command to run the second simulated device:

```
python3 cloudiot_mqtt_example_json.py \
   --project_id=$PROJECT_ID \
   --cloud_region=$MY_REGION \
   --registry_id=iotlab-registry \
   --device_id=temp-sensor-istanbul \
   --private_key_file=rsa_private.pem \
   --message_type=event \
   --algorithm=RS256 --num_messages=1000
```

Telemetry data will flow from the simulated devices through Cloud IoT Core to your Cloud Pub/Sub topic. In turn, your Dataflow job will read messages from your Pub/Sub topic and write their contents to your BigQuery table.

Examine the stored data

Dataflow is collecting the data published by Pub/Sub and saving it in output files in the bucket and folder specified in the job template. The files are written every 5 minutes, and each begins with the prefix specified in the job template.

1. Back in the Console, open the **Navigation menu** and click **Cloud Storage**.

2. Click on the bucket you created.

3. Select the folder **Sensor-Data**. Dataflow is writing the data from the device to this folder. It writes a file every few minutes. If the folder is empty, wait a few minutes then click the "Refresh" link at the top of the Storage Browser page.

4. Open a file by clicking on its name. Click on the **Authenticated URL** link.

Your file contents should be similar to what is shown below.

Output (do not copy)

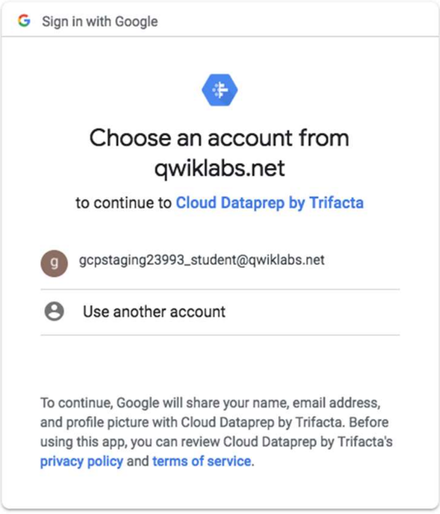
```
Publishing message 51 of 1000: '{'device': 'temp-sensor-istanbul', 'timestamp': 'Mon Aug
13 22:12:28 2018', 'temperature': 23.215100358910885}'on_publishPublishing message
52 of 1000: '{'device': 'temp-sensor-istanbul', 'timestamp': 'Mon Aug 13 22:12:29 2018',
'temperature': 23.22890311742878}'on_publishPublishing message 53 of 1000: '{'device':
'temp-sensor-istanbul', 'timestamp': 'Mon Aug 13 22:12:30 2018', 'temperature':
23.237443887313777}'on_publishPublishing message 54 of 1000: '{'device': 'temp-sensor-
istanbul', 'timestamp': 'Mon Aug 13 22:12:31 2018', 'temperature': 23.24313271883122}'
```

Create a Dataprep Flow

In this section you will create a Dataprep Flow. Open the **Navigation menu** and click on the **Dataprep** service.

Accept the "Google Dataprep Terms of Service". You will be prompted to share account information with Trifacta—check the box and click **Agree and Continue**. Then click **Allow** to let Trifacta access your project data.

You will be brought to a sign in page that looks similar to the following:



Sign in with Google

Choose an account from qwiklabs.net

to continue to Cloud Dataprep by Trifacta

gcpstaging23993_student@qwiklabs.net

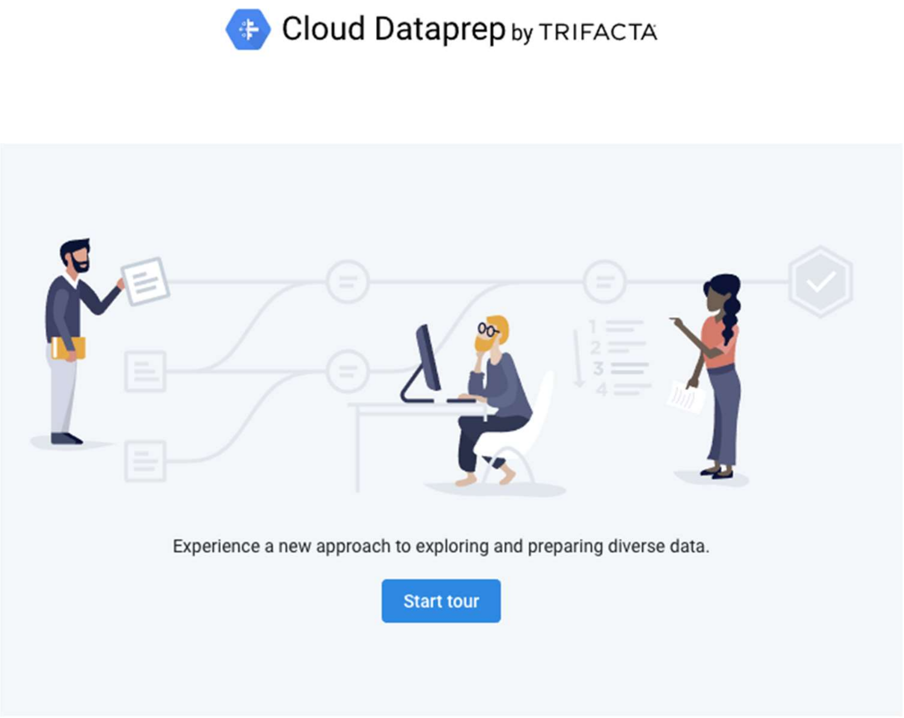
Use another account

To continue, Google will share your name, email address, and profile picture with Cloud Dataprep by Trifacta. Before using this app, you can review Cloud Dataprep by Trifacta's privacy policy and terms of service.

Click your Qwiklabs Google Cloud Username and then **Allow**. **Accept** the terms of service that follow.

Your storage location should be selected. Click **Continue** to finish setup.

You should now be on a home page that resembles the following:



Cloud Dataprep by TRIFACTA

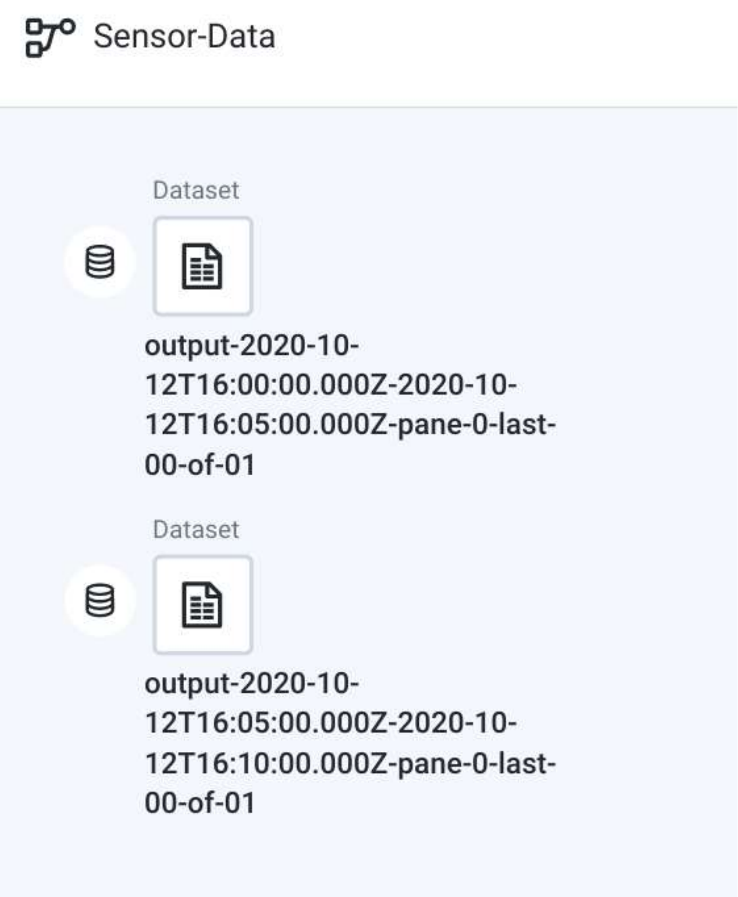
Experience a new approach to exploring and preparing diverse data.

You are now in the Cloud Dataprep UI. You wil now import the dataset from your Cloud Storage bucket.

1. Click **Import Data** from the top right corner.
2. On the left hand side, click **GCS**.
3. Click the name of your bucket, then click **Sensor-Data/**.
4. Click the **+** next to the files in the folder (should be ~2. Go back to Console and **Refresh** the bucket if you only see 1).
5. Check the **Add Datasets to a Flow** box in the bottom-right corner.
6. Then click **Continue**. Name the flow "Sensor-Data" and click **OK**.

Read and step through the information panels that define a Dataprep flow.

Your Dataprep page should now resemble the following:



Sensor-Data

Dataset

output-2020-10-12T16:00:00.000Z-2020-10-12T16:05:00.000Z-pane-0-last-00-of-01

Dataset

output-2020-10-12T16:05:00.000Z-2020-10-12T16:10:00.000Z-pane-0-last-00-of-01

Create a Dataprep recipe

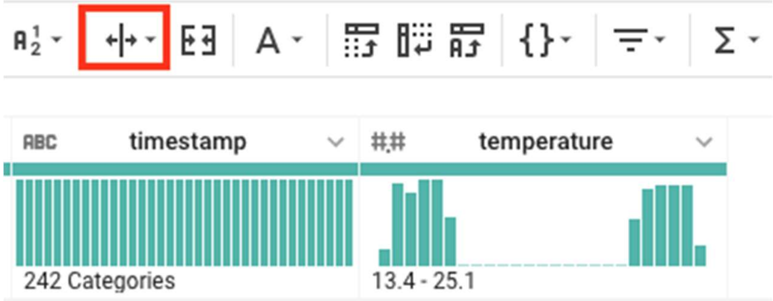
In this section you will create a Dataprep *recipe*. A recipe is a list of transformations and modifications of the data and data set. It will be applied to all the data presently in the dataset, as well as any new data.

1. Click on the first dataset and then click **Add** > **Recipe** in the right-hand side of the console.

2. Then click **Edit Recipe**.

Split a column

1. Read and step through the information panels that define a Transformer. Click the **Split Column** icon from the toolbar:



2. Click **On delimiter**
3. Click in the Column field, select **device**.
4. In the delimiter section type in `'temp-sensor-'`
5. Click **Add**.

You should have a column called **device2** with locations of the devices listed (`buenos-aires` and `istanbul`).

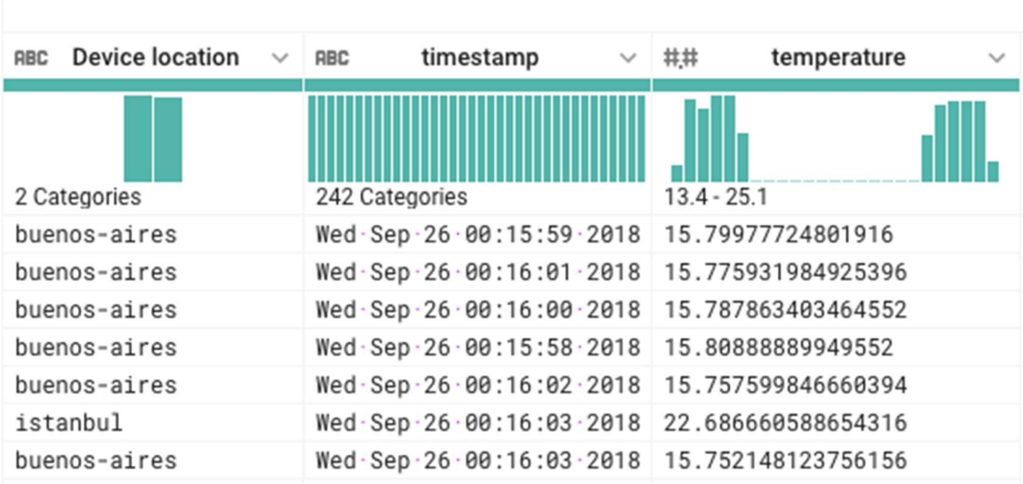
Delete column

1. Click the expansion arrow on the column titled **device1**.

2. Click **Delete**.

Rename column

1. Click the expansion arrow on the column titled **device2**.
2. Select **Rename**, and in the New name field, enter `'Device location'`.
3. Click **Add**.
4. The "device2" column should now be renamed to "Device location":



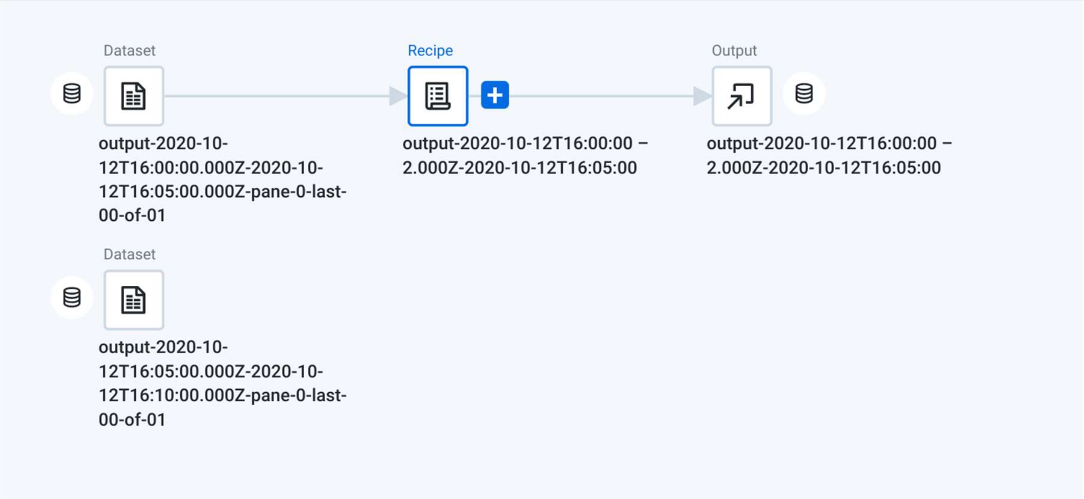
ABC Device location	ABC timestamp	## temperature
2 Categories	242 Categories	13.4 - 25.1
buenos-aires	Wed·Sep·26·00:15:59·2018	15.79977724801916
buenos-aires	Wed·Sep·26·00:16:01·2018	15.775931984925396
buenos-aires	Wed·Sep·26·00:16:00·2018	15.787863403464552
buenos-aires	Wed·Sep·26·00:15:58·2018	15.80888889949552
buenos-aires	Wed·Sep·26·00:16:02·2018	15.757599846660394
istanbul	Wed·Sep·26·00:16:03·2018	22.686660588654316
buenos-aires	Wed·Sep·26·00:16:03·2018	15.752148123756156

Round values in a column

1. Click the expansion arrow on the column titled **temperature**.
2. Click **Calculate** > **Round** > **Round**.
3. Click **Add**.

You now have a recipe for handling data coming from Cloud Storage.

4. Click **Run** in the far right corner, then **Run** again in the lower right corner. This can take several minutes to complete. Watch the job progress in the right-hand column.
5. When the job is done, **double click** the recipe icon in the flow and examine the data:


Dataset
output-2020-10-12T16:00:00.000Z-2020-10-12T16:05:00.000Z-pane-0-last-00-of-01
Recipe
output-2020-10-12T16:00:00 – 2.000Z-2020-10-12T16:05:00
Output
output-2020-10-12T16:00:00 – 2.000Z-2020-10-12T16:05:00
Dataset
output-2020-10-12T16:05:00.000Z-2020-10-12T16:10:00.000Z-pane-0-last-00-of-01

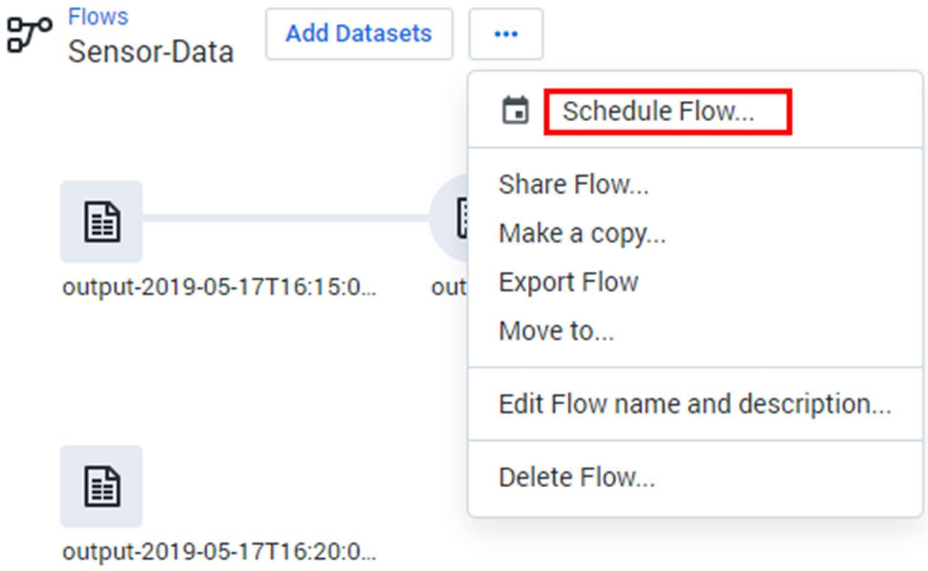
Schedule a Dataprep job

You now have a Dataprep dataset and a recipe for transforming the data. As you may have noticed, your devices are continuing to publish data and store it in Cloud Storage. You will now schedule a flow to download new data and execute the recipe.

1. Click on the Dataprep icon in the upper left corner:

2. Click on **Sensor-Data**.

3. At the top of the Flow page, click **Schedule**.

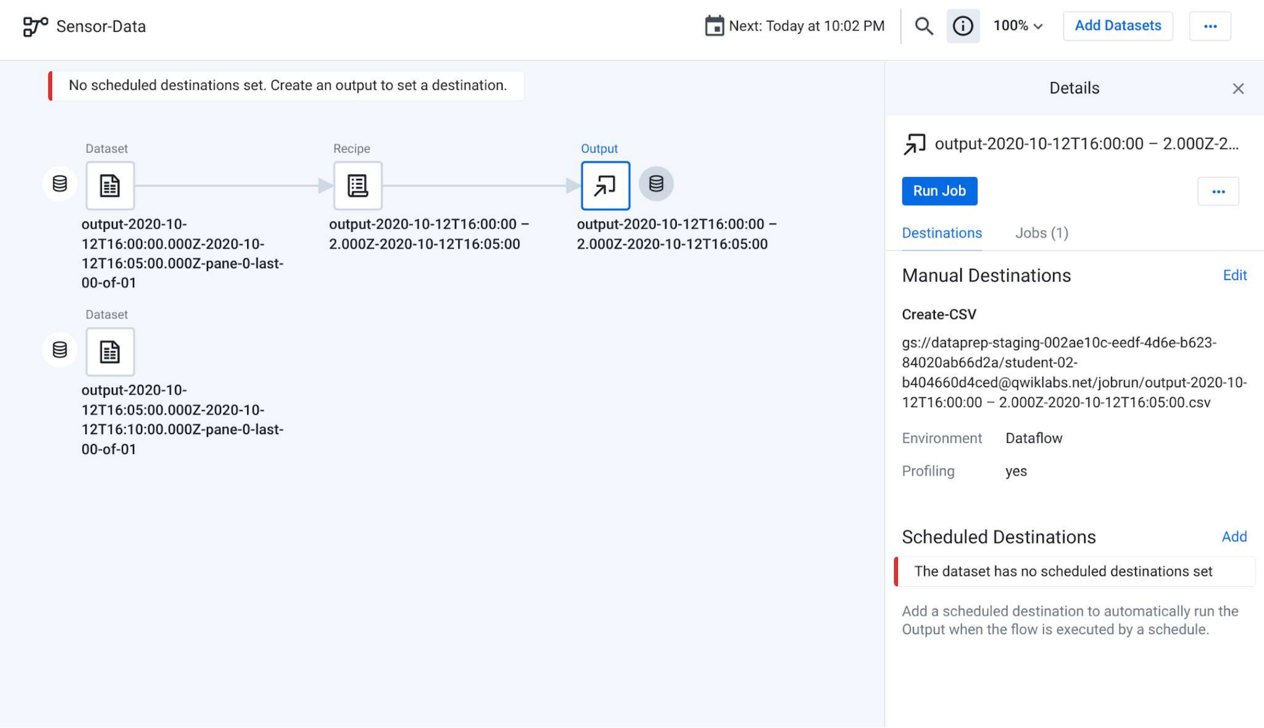


4. Specify the following:

Field	Value
Timezone	Select your timezone
Frequency	**hourly**
Minutes past the hour	Specify 2 minutes from your present time

5. Click **Save**. You will see a message that says ‘No scheduled destinations set...'. You will set the destination in the following steps.

6. Open the **Destinations** tab in the Details section of your recent recipe.



No scheduled destinations set. Create an output to set a destination.

output-2020-10-12T16:00:00.000Z-2020-10-12T16:05:00.000Z-pane-0-last-00-of-01

output-2020-10-12T16:00:00 – 2.000Z-2020-10-12T16:05:00

output-2020-10-12T16:00:00 – 2.000Z-2020-10-12T16:05:00

output-2020-10-12T16:05:00.000Z-2020-10-12T16:10:00.000Z-pane-0-last-00-of-01

output-2020-10-12T16:00:00 – 2.000Z-2...

Manual Destinations

Create-CSV

gs://dataprep-staging-002ae10c-eedf-4d6e-b623-84020ab66d2a/student-02-b404660d4ced@qwiklabs.net/jobrun/output-2020-10-12T16:00:00 – 2.000Z-2020-10-12T16:05:00.csv

Environment Dataflow

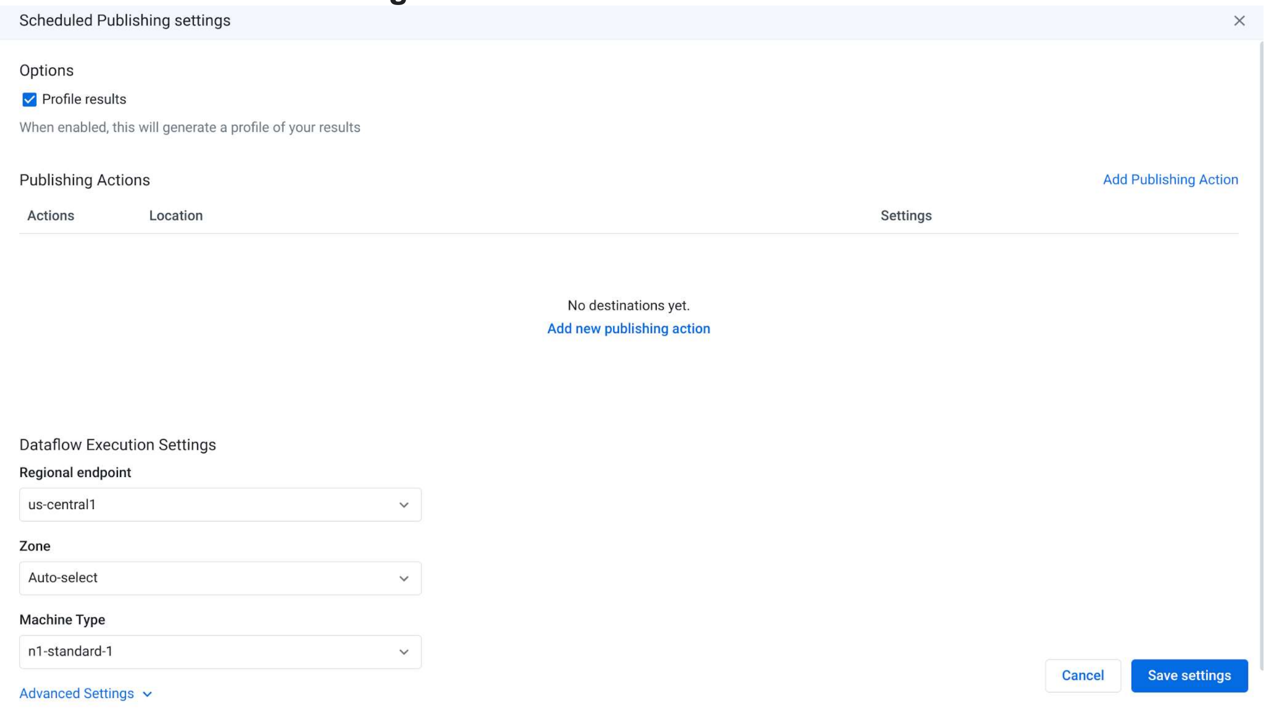
Profiling yes

Scheduled Destinations

The dataset has no scheduled destinations set

Add a scheduled destination to automatically run the Output when the flow is executed by a schedule.

7. For **Scheduled Destinations**, click **Add**.
8. Click **Add Publishing Action**.



Scheduled Publishing settings

Options

☑ Profile results

When enabled, this will generate a profile of your results

Publishing Actions

No destinations yet.

Dataflow Execution Settings

Regional endpoint: us-central1

Zone: Auto-select

Machine Type: n1-standard-1

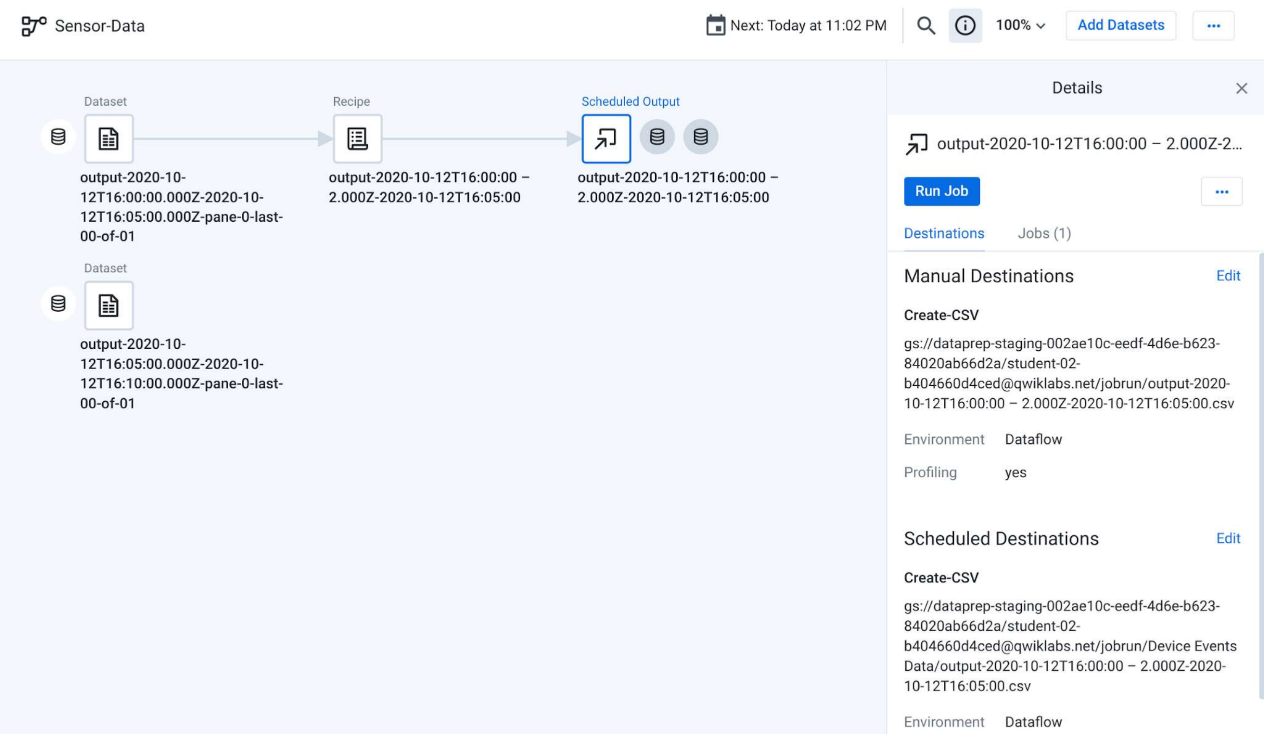
9. Click **GCS**.
10. Click **Create Folder**.
11. Specify the following:

Folder Name	**Device Events Data**

12. Click **Create Folder**.
13. Click **Create a new file**.
14. Click **Add**.

15. Click **Save Settings**.

Your flow should look like the image below. Note the calendar icon, this signifies the flow is scheduled.



Dataset: output-2020-10-12T16:00:00.000Z-2020-10-12T16:05:00.000Z-pane-0-last-00-of-01

Recipe: output-2020-10-12T16:00:00 – 2.000Z-2020-10-12T16:05:00

Scheduled Output: output-2020-10-12T16:00:00 – 2.000Z-2020-10-12T16:05:00

Dataset: output-2020-10-12T16:05:00.000Z-2020-10-12T16:10:00.000Z-pane-0-last-00-of-01

output-2020-10-12T16:00:00 – 2.000Z-2...

Manual Destinations

Create-CSV

gs://dataprep-staging-002ae10c-eedf-4d6e-b623-84020ab66d2a/student-02-b404660d4ced@qwiklabs.net/jobrun/output-2020-10-12T16:00:00 – 2.000Z-2020-10-12T16:05:00.csv

Environment Dataflow

Profiling yes

Scheduled Destinations

Create-CSV

gs://dataprep-staging-002ae10c-eedf-4d6e-b623-84020ab66d2a/student-02-b404660d4ced@qwiklabs.net/jobrun/Device Events Data/output-2020-10-12T16:00:00 – 2.000Z-2020-10-12T16:05:00.csv

Environment Dataflow

Examine the data

You have created a Dataprep flow and you have created a schedule for it to be regularly updated. Examine the data after a scheduled job execution.

1. Click the **Job history** icon from the left-hand menu:

You can run the job manually. If you run it manually, you won't see the auto update run.

To run a job manually, do the following:

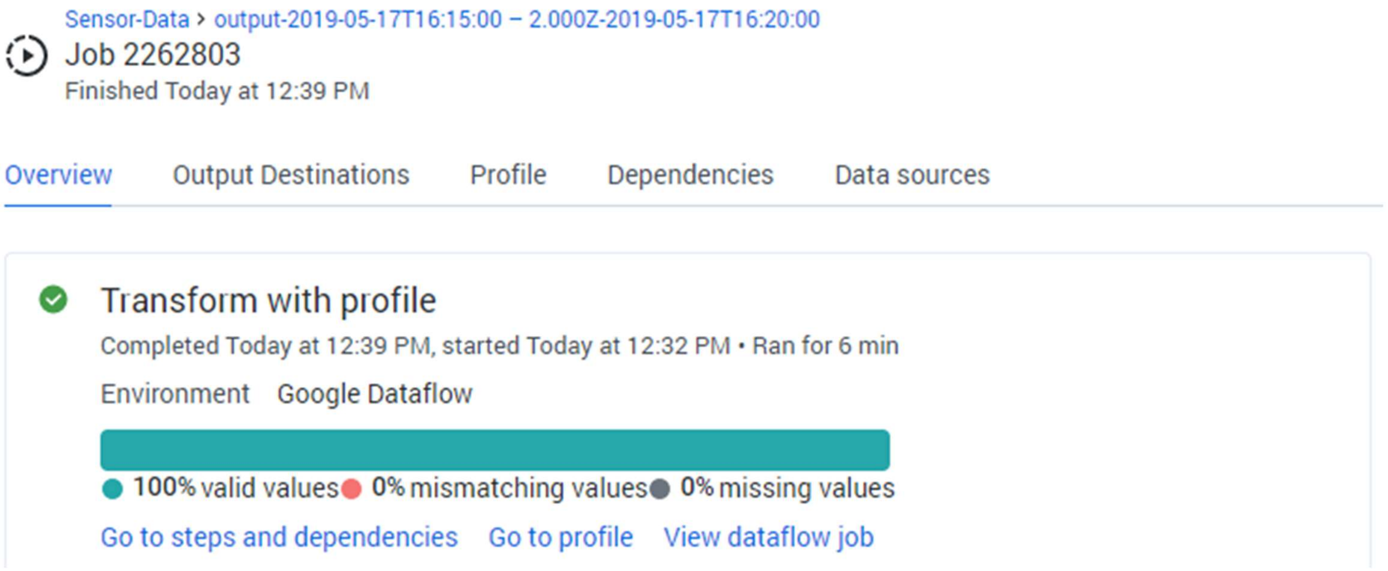
1. Click on the **Flows** page. Then click **Sensor-Data**.
2. Click on the second recipe icon and then click **Add** > **Recipe**.
3. Click on **Edit Recipe.**
4. Click on **Run**(upper right corner).
5. Click on **Run**(bottom right corner).
6. Click on the **Job history** icon from the left-hand menu.
2. The Jobs page lists the jobs that have been completed. **Click on the job number** as soon as it is completed. This time the job will take a little longer because it has more data to process.

Once you click on the finished job, the metadata for the dataset is shown. Moving your cursor over the bar graphs will give you details about each data point.

Dataprep assumes the dataset is large, therefore it does not show the entire dataset. The default is to show initial values. There are cases where this may not be acceptable.

Change dataset sample

1. Find the dataset name in the information bar at the top of the page. Click on the dataset name.



Sensor-Data > output-2019-05-17T16:15:00 – 2.000Z-2019-05-17T16:20:00

Job 2262803

Finished Today at 12:39 PM

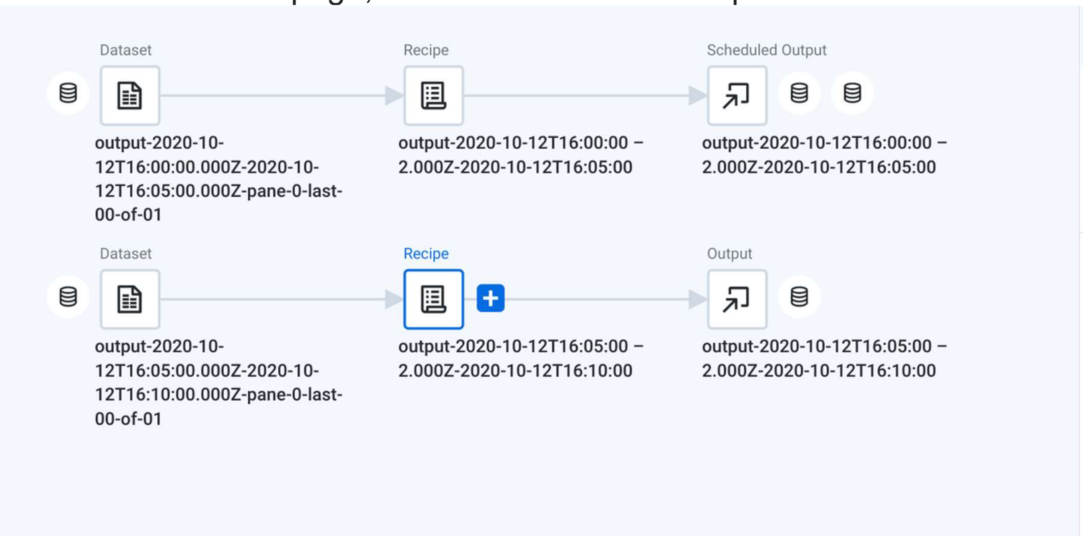
Transform with profile

Completed Today at 12:39 PM, started Today at 12:32 PM • Ran for 6 min

Environment Google Dataflow

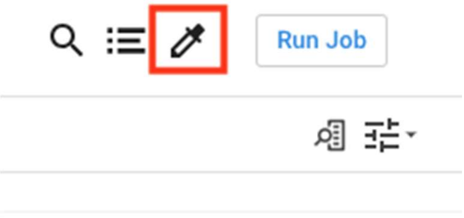
● 100% valid values ● 0% mismatching values ● 0% missing values

2. On the Flows page, **double click** on the recipe icon.



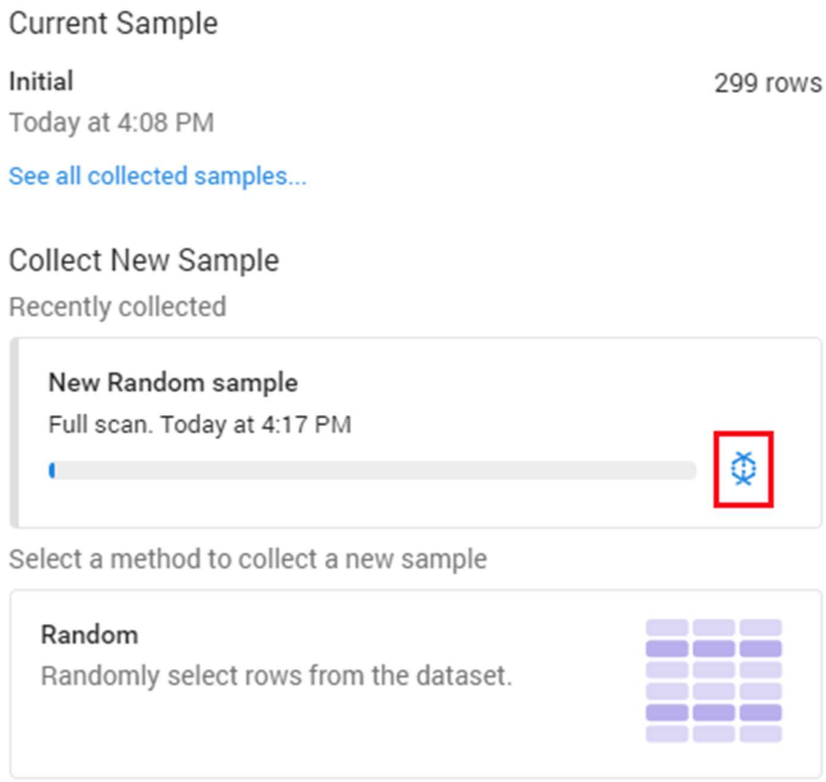
Dataset
output-2020-10-12T16:00:00.000Z-2020-10-12T16:05:00.000Z-pane-0-last-00-of-01
Recipe
output-2020-10-12T16:00:00 – 2.000Z-2020-10-12T16:05:00
Scheduled Output
output-2020-10-12T16:00:00 – 2.000Z-2020-10-12T16:05:00
Dataset
output-2020-10-12T16:05:00.000Z-2020-10-12T16:10:00.000Z-pane-0-last-00-of-01
Recipe
output-2020-10-12T16:05:00 – 2.000Z-2020-10-12T16:10:00
Output
output-2020-10-12T16:05:00 – 2.000Z-2020-10-12T16:10:00

3. In the upper right corner, click the **Samples** icon:



4. Click **GOT IT!** when the panel opens up in the right-hand column. This will display all the samples available.

5. Click **Random**.
6. Select **Full** to pull samples from the entire dataset.
7. Click **Collect**.
8. Click on the Dataflow symbol next to the progress bar (it might take a few seconds for this to appear; if taking too long, refresh your browser):



Current Sample

Initial — 299 rows

Today at 4:08 PM

See all collected samples...

Collect New Sample

Recently collected

New Random sample

Full scan. Today at 4:17 PM

Select a method to collect a new sample

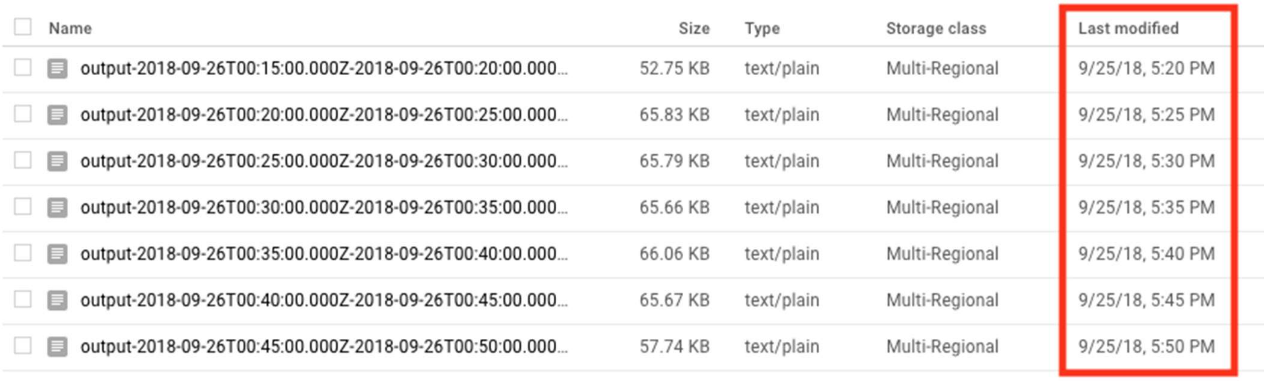
Random

Randomly select rows from the dataset.

Go back to the Cloud Console. The Dataflow page shows the job as running. The time to complete the job depends on how much data has accumulated in the **Sensor-Data** folder, but should only take a couple of minutes.

9. Click **Navigation menu** > **Cloud Storage** > **Browser**.
10. Click on the bucket you created.
11. Click **Sensor-Data**.

You should now see a list of files, each written five minutes apart. These files contain all the data published by the devices:

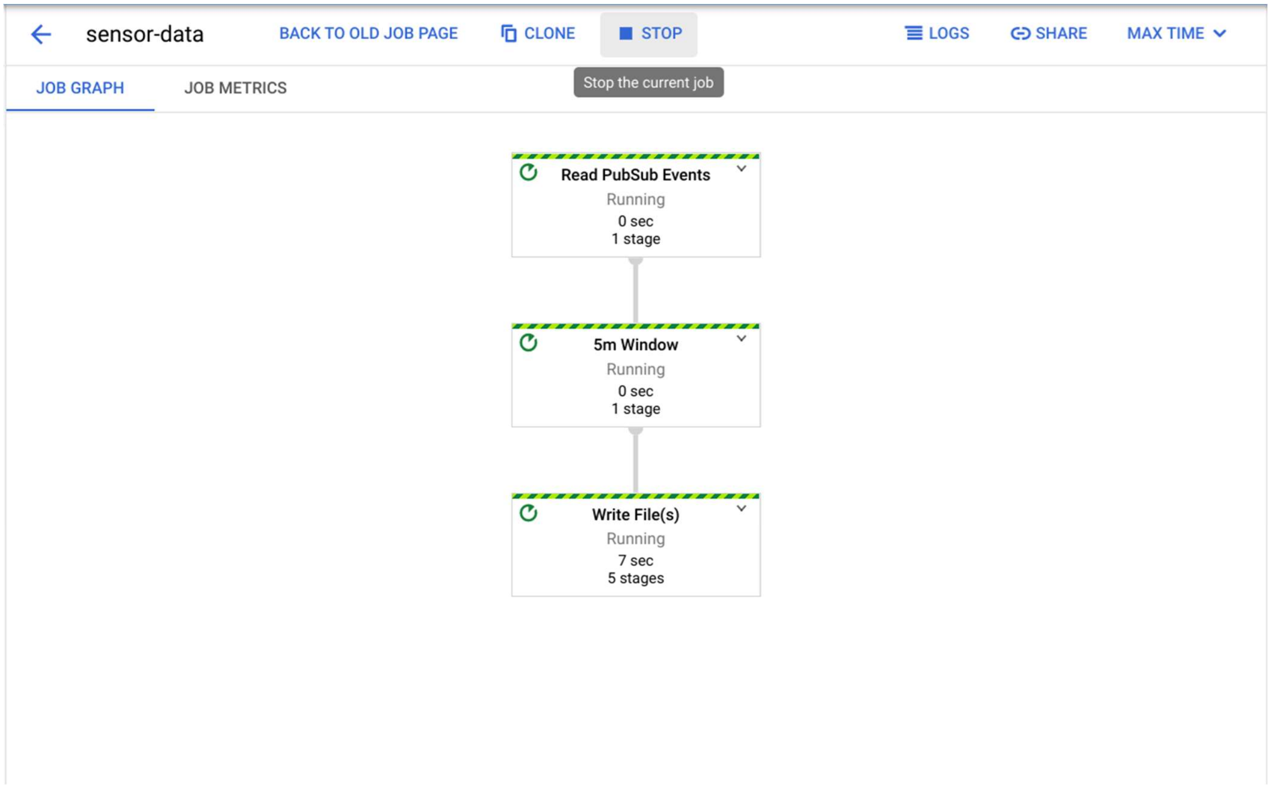


□ Name	Size	Type	Storage class	Last modified
□ output-2018-09-26T00:15:00.000Z-2018-09-26T00:20:00.000...	52.75 KB	text/plain	Multi-Regional	9/25/18, 5:20 PM
□ output-2018-09-26T00:20:00.000Z-2018-09-26T00:25:00.000...	65.83 KB	text/plain	Multi-Regional	9/25/18, 5:25 PM
□ output-2018-09-26T00:25:00.000Z-2018-09-26T00:30:00.000...	65.79 KB	text/plain	Multi-Regional	9/25/18, 5:30 PM
□ output-2018-09-26T00:30:00.000Z-2018-09-26T00:35:00.000...	65.66 KB	text/plain	Multi-Regional	9/25/18, 5:35 PM
□ output-2018-09-26T00:35:00.000Z-2018-09-26T00:40:00.000...	66.06 KB	text/plain	Multi-Regional	9/25/18, 5:40 PM
□ output-2018-09-26T00:40:00.000Z-2018-09-26T00:45:00.000...	65.67 KB	text/plain	Multi-Regional	9/25/18, 5:45 PM
□ output-2018-09-26T00:45:00.000Z-2018-09-26T00:50:00.000...	57.74 KB	text/plain	Multi-Regional	9/25/18, 5:50 PM

Cleanup

Because you're in a Qwiklab, when you're lab completes all of your resources will be deleted. However, it's good to know how how to clean up a project! The following steps specify an orderly shutdown of a Dataflow job:

1. Click **Navigation menu** > **Dataflow**.
2. Click on the **sensor-data** job.
3. Then Click **STOP**.



4. When the page pops up, select **Drain** > **STOP JOB**.

Test your knowledge

Test your knowledge about Google cloud Platform by taking our quiz.

Cloud Dataprep is used to explore and transform raw data from disparate and/or large datasets into clean and structured data for further analysis and processing.
True

Integrations with datastores other than BigQuery, Cloud Storage, and the local filesystem are not supported in Cloud Dataprep.
True

Congratulations!

In this lab you sent streaming data from Cloud Storage to Dataprep.

Finish Your Quest

This self-paced lab is part of the Qwiklabs **IoT in the Google Cloud** 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 (or badges) 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

Be sure to check out the following labs for more practice with IoT, Firebase, and Cloud Functions:

- A Tour of Cloud IoT Core
- Building an IoT Analytics Pipeline on Google Cloud
- Firebase SDK for Cloud Functions

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 April 28, 2021
Lab Last Tested April 28, 2021