

Streaming IoT Data to Cloud Storage

GSP278



Google Cloud Self-Paced Labs

Overview

In this lab you will learn how to configure Cloud IoT Core and Cloud Pub/Sub to create a Pub/Sub topic and registry on Google Cloud. You use this topic to ingest data streaming from a simulated device.

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

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:

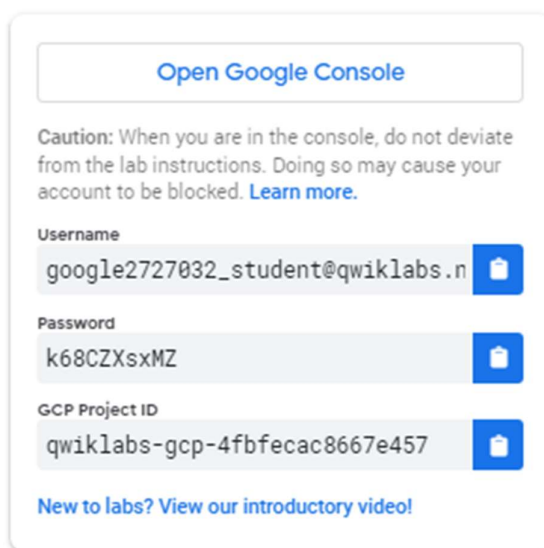
- Access to a standard internet browser (Chrome browser recommended).
- Time to complete the lab.

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.

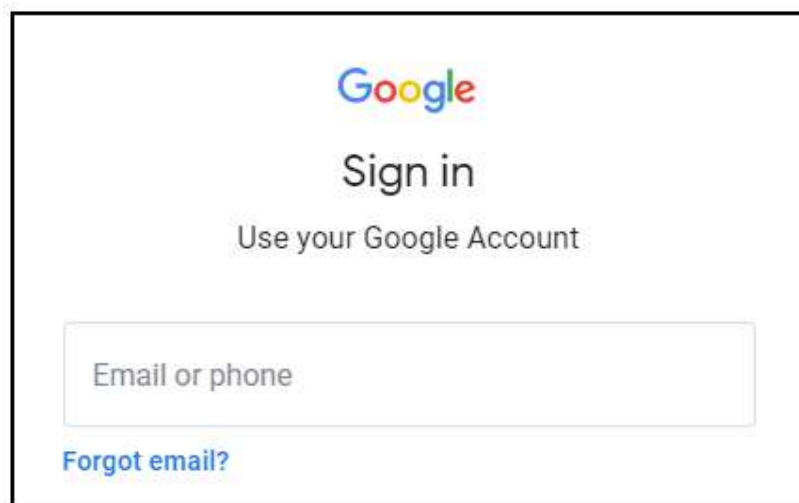
How to start your lab and sign in to the Google Cloud Console

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.



A screenshot of a web panel titled "Open Google Console". It contains a caution message: "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.](#)". Below this are three input fields, each with a copy icon to its right: "Username" with the value "google2727032_student@qwiklabs.n", "Password" with the value "k68CZXsxMZ", and "GCP Project ID" with the value "qwiklabs-gcp-4fbfecac8667e457". At the bottom is a link: "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.



A screenshot of the Google Sign in page. It features the Google logo at the top, followed by the text "Sign in" and "Use your Google Account". Below this is a large input field labeled "Email or phone". At the bottom left is a link: "Forgot email?"

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.

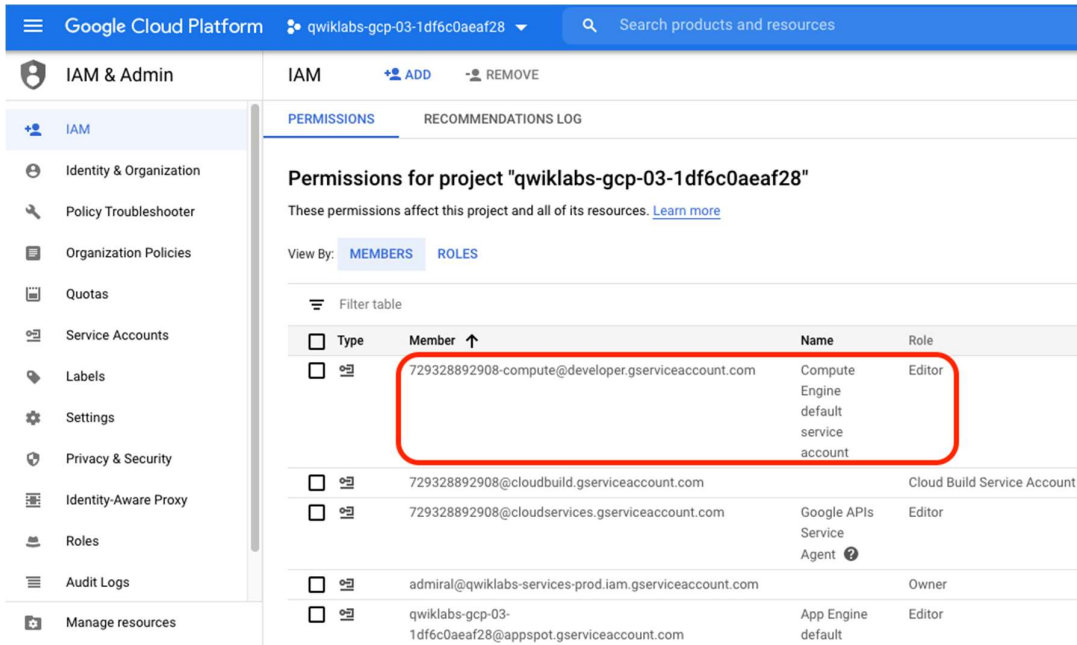
Note: You can view the menu with a list of Google Cloud Products and Services by clicking the **Navigation menu** at the top-left.



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**.



Google Cloud Platform • qwklabs-gcp-03-1df6c0aeaf28 Search products and resources

IAM & Admin

IAM + ADD - REMOVE






PERMISSIONS RECOMMENDATIONS LOG

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

These permissions affect this project and all of its resources. [Learn more](#)

View By: MEMBERS ROLES

Filter table

Type	Member ↑	Name	Role
<input type="checkbox"/>		729328892908-compute@developer.gserviceaccount.com	Compute Engine default service account
<input type="checkbox"/>		729328892908@cloudbuild.gserviceaccount.com	Cloud Build Service Account
<input type="checkbox"/>		729328892908@cloudservices.gserviceaccount.com	Google APIs Service Agent ?
<input type="checkbox"/>		admiral@qwklabs-services-prod.iam.gserviceaccount.com	Owner
<input type="checkbox"/>		qwklabs-gcp-03-1df6c0aeaf28@appspot.gserviceaccount.com	App Engine default

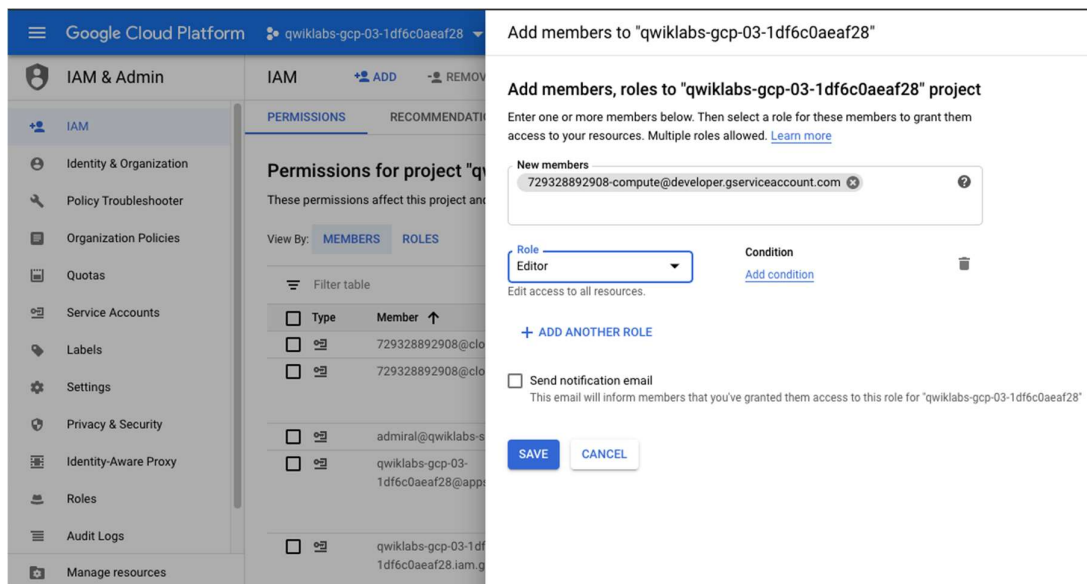
If the account is not present in IAM or does not have the `editor` role, follow the steps below to assign the required role.

- 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**.



Cloud Pub/Sub setup and topics

First you need to 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** section.
4. Click **CREATE TOPIC**.

Test Completed Task

Click **Check my progress** to verify your performed task. If you have completed the task successfully you will be granted with an assessment score.

Setting topic permissions

You now have a Pub/Sub topic. To allow the project to publish this topic, you need to add the project as a member/publisher.

To add members, click **Add Member**.

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

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

Create a location for data storage

You need to create a storage bucket and folder to store the data streaming from the IoT device.

1. From the **Navigation menu** click **Storage > Browser**.
2. Click **Create bucket**.
3. Bucket names must be [unique](#). Enter a unique bucket name, and then click **Create**.

Test Completed Task

Click **Check my progress** to verify your performed task. If you have completed the task successfully you will be granted with an assessment score.

4. In the bucket you just created, click **Create folder**.
5. For **Name**, type **Sensor-Data**.
6. Click **Create**.

Test Completed Task

Click **Check my progress** to verify your performed task. If you have completed the task successfully you will be granted with an assessment score.

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.

1. On the **Navigation menu**, in the Big Data section, click **Dataflow**.
2. Click **Create job from template**.
3. Enter the following values in the template.

Property	Value(type or select)
Job name	sensor-data
Regional endpoint	us-central1
Cloud Dataflow template	PubSub 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.

Property	Value(type or select)
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

Your template should resemble the following:

Job name *

sensor-data

Must be unique among running jobs

Regional endpoint *

us-central1

Choose a Dataflow regional endpoint to deploy worker instances and store job metadata. You can optionally deploy worker instances to any available Google Cloud region or zone by using the worker region or worker zone parameters. Job metadata is always stored in the Dataflow regional endpoint. [Learn more](#)

Dataflow template *

Pub/Sub to Text Files on Cloud Storage

Streaming pipeline. Reads records from Pub/Sub and writes them to Cloud Storage, creating a text file for each five minute window. Note that this pipeline assumes no newlines in the body of the Pub/Sub message and thus each message becomes a single line in the output file.

Required parameters

Input Pub/Sub topic *

projects/qwiklabs-gcp-00-c625a70c76ac/topics/iotlab

The Pub/Sub topic to read the input from. Ex: projects/your-project-id/topics/your-topic-name

Output file directory in Cloud Storage *

gs://qwiklabs-gcp-00-c625a70c76ac/Sensor-Data/

The path and filename prefix for writing output files. DateTime formatting is used to parse directory path for date & time formatters. Ex: gs://your-bucket/your-path

Output filename prefix *

output-

The prefix to place on each windowed file. Ex: output-

Temporary location *

gs://qwiklabs-gcp-00-c625a70c76ac/tmp

Path and filename prefix for writing temporary files. Ex: gs://your-bucket/temp

Encryption



Google-managed key

No configuration required



Customer-managed key

Manage via Google Cloud Key Management Service

✓ SHOW OPTIONAL PARAMETERS

RUN JOB

5. Once you have verified that all the fields are properly filled out, click **Run job**.

Test Completed Task

Click **Check my progress** to verify your performed task. If you have completed the task successfully you will be granted with an assessment score.

Prepare Your Compute Engine 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.

To connect to the **iot-device-simulator** VM instance:

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

VM instances						
CREATE INSTANCE IMPORT VM REFRESH START STOP SHOW INFO PANEL						
Filter VM instances						
Columns						
<input type="checkbox"/> Name	Zone	Recommendation	In use by	Internal IP	External IP	Connect
<input type="checkbox"/> iot-device-simulator	us-central1-a			10.128.0.2 (nic0)	35.224.136.88	SSH
<input type="checkbox"/> sensor-data-03261253-b1ke-harness-9z8j	us-central1-a		dataflow-sensor-data-03261253-b1ke-harness	10.128.0.3 (nic0)	35.225.178.187	SSH

2. 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 by running:

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

Enter through all the prompts that follow.

5. End your SSH session on the **iot-device-simulator** VM instance:

```
exit
```

6. Start another SSH session on the **iot-device-simulator** VM instance.
7. Initialize 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.

8. When you are asked whether to authenticate with an `@developer.gserviceaccount.com` account or to log in with a new account, choose to log in with a new account.

9. When you are asked "Are you sure you want to authenticate with your personal account? Do you want to continue (Y/n)?" enter **Y**.
10. Click on the URL shown to open a new browser window that displays a verification code. If prompted, sign in using your qwiklabs generated student account and click **Allow**.
11. Copy the verification code and paste it in response to the "Enter verification code:" prompt and press **Enter**.
12. In response to "Pick cloud project to use", pick the Google Cloud project that Qwiklabs created for you.
13. Enter this command to make sure that the components of the SDK are up to date:

```
gcloud components update
```

14. Enter the following command to install the beta components. Enter **Y** when prompted to continue:

```
gcloud components install beta
```

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

```
sudo apt-get update
```

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

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

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

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

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

```
git clone http://github.com/GoogleCloudPlatform/training-data-analyst
```

19. In your SSH session on the **iot-device-simulator** VM instance, run the following, adding your project ID as the value for **PROJECT_ID**:

```
export PROJECT_ID=
```

Your completed command will look like this: `export PROJECT_ID=qwiklabs-gcp-d2e509fed105b3ed`

19. You must choose a region for your IoT registry. Set an environment variable containing the `us-central1` region with the following command:

```
export MY_REGION=us-central1
```

Leave this SSH window open. You will come back to it after creating a device registry in the following step.

Open IoT Core

1. Return to the Cloud Console. From the Navigation menu, 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:

Property	Value(type or select)
Registry ID	iotlab-registry
Region	us-central1 (or select a region that is closest to you)
Select a Cloud Pub/Sub topic	projects/<project-id>/topics/iotlab (Choose from dropdown)

4. Click **Create**.

Test Completed Task

Click **Check my progress** to verify your performed task. If you have completed the task successfully you will be granted with an assessment score.

Create a Cryptographic Keypair

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

In your SSH session on the **iot-device-simulator** VM instance, enter these commands to create the keypair 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 on the `iot-device-simulator`, type

```
cat rsa_cert.pem
```

2. The output will be your certificate. Select and copy the entire certificate (including the `begin/end certificate`).

Example only (do not copy)

```
-----BEGIN CERTIFICATE-----
MIIC+DCCAeCgAwIBAgIJA0JikTScq9oPMA0GCSqGSIb3DQEBCwUAMBExDzANBgNV
BAMMBnVudXNlZDAeFw0xODA4MTMxNjQ2MTNaFw0xODA5MTIxNjQ2MTNaMBExD
zANBgNVBAMMBnVudXNlZDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggE
BAL+wLyITE5Tj1H50I63ew3HdvoGty2aOpP04nMyOYZoooAw5o2rj5mkNb/hbkoMTkzo
6/5Jo0zgDYPVRpz2nGAhTfeQzPuvOfPZe7KPPzXyvmSN3pYT9kkiVo9pXwynG7q8kW
72Q9f0pffXS/VElPrC63Y9kcAg0yveZVX61qSokz4DVIj0Z6+1b1utxe2TnxR1q3Hce289
1re6qnxYp6YuW0gVYtn8HdgEKKMqeSozqJP7dq8EvNkwY8BAUFU2NmuvwK2Z6hB1E
u0DImyhtKRxZ4pUbWuefC+P6GU2fB3rp4pR9Lc7xd5BuWXHgR6f0lV57e1L9f1Q/iXippP
8RjhMCAwEAAaNTMFewHQYDVR00BBYEFF7808W+vP7vbgg6cS5Fky9xCstNMB8GA
1UdIwQYMBaAFF7808W+vP7vbgg6cS5Fky9xCstNMA8GA1UdEwEB/wQFMAMBAf8w
DQYJKoZIhvcNAQELBQADggEBAD9mSbWQRz8QHI947gGSMrsA+a04dgWIujkypFw/p
7gSefleCCwGV4Wpfq6zoIjru9bnciWRLHZMKVbhptBDseyBnoPXxnJMgVYBAVzRRMhT
qPeo146Pv99dn3c310M2tkpQeQzP/wE9XFVqEud2sZCKXgXtydIsyTEX3wmG9s9m7f
6TJDknvJltOj1R7m+xO6GHPEbK29x/r+LzPuYjIDYoG+mxLQUltDOM3v8QwZ4bneo+HI
BZX6FOBRb+x/fEE3EANCY3J5sKwRCxxXJ6l/Mts7aLUE6MrT8BM0n1fxnY7BX+6dvsJ
H/OeONG2tk3Y0ci/ly245NQyurqa3x35Ws=
-----END CERTIFICATE-----
```

3. Your IoT Core Console page should resemble the following:

Registry details
EDIT REGISTRY
DELETE REGISTRY

Registry ID: **iotlab-registry**

Region	us-central1
Protocol	MQTT
	HTTP
Cloud Logging	Disabled View logs

Cloud Pub/Sub topics

A registry can have 1 or more topics for publishing device telemetry and state events.

[Add or edit topics](#)

Pub/Sub topics

Topic name	Topic type ?	Subfolder
projects/qwiklabs-gcp-00-c625a70c76ac/topics/iotlab	Default telemetry	—
—	Device state	—

[CA CERTIFICATES](#)

From the left-hand menu, select **Devices** and ensure that the Registry ID is set to **iotlab-registry**.

- Click **+ CREATE A DEVICE** from the top menu and enter in the following:
Click on the **COMMUNICATION, CLOUD LOGGING, AUTHENTICATION** option to see advance fields.

Device ID	temp-sensor-buenos-aires
Authentication	Enter manually
Public key format	RS256_X509
Public key value	Paste the certificate that you copied

- Click **Create**.

- You will now add a second device. Hit the back button to return to the device menu and once again click **+ CREATE A DEVICE**. Then enter the following settings:

Click on the **COMMUNICATION, CLOUD LOGGING, AUTHENTICATION** option to see advance fields.

Device ID	temp-sensor-istanbul
Authentication	Enter manually
Public key format	RS256_X509
Public key value	Paste the certificate that you copied

- Click **Create**.

8. Hit the back button to return to the device overview. Your page should resemble the following:

Registry ID: iotlab-registry

us-central1

Devices are things that connect to the internet directly or through a gateway. [Learn more](#)

☰ Enter exact device ID

<input type="checkbox"/>	Device ID	Communication	Last seen	Stackdriver Logging
<input type="checkbox"/>	temp-sensor-buenos-aires	✓ Allowed	—	Registry default
<input type="checkbox"/>	temp-sensor-istanbul	✓ Allowed	—	Registry default

[Cloud IoT Core documentation](#)

Test Completed Task

Click **Check my progress** to verify your performed task. If you have completed the task successfully you will be granted with an assessment score.

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 clouddiot 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 \  
  --message_payload=event
```



```
--algorithm=RS256 --num messages=200 > 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=200
```

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 Cloud Storage bucket.

NOTE: Wait for few seconds for the logs to get generated in the file.

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. Return to the Cloud Console. Open the Navigation menu and select **Storage**.
2. Select the bucket you created for this project.
3. Select the folder **Sensor-Data**. Dataflow is writing the data from the device to this folder.

Files are written every five minutes. If the folder is empty, wait about 5-10 minutes and periodically click the Refresh Bucket button.

4. After refreshing, you will soon see an output file in your directory:

Upload files

Upload folder

Create folder

Manage holds

Delete

🔍

Filter by prefix...

Buckets

/

treysan

/

Sensor-Data

<input type="checkbox"/>	Name	Size	Type	Storage class	Last modified
<input type="checkbox"/>	<div><div><div></div></div>output-2019-03-26T20:10:00.000Z-2019-03-26T20:...</div>	42.18 KB	text/plain	Multi-Regional	3/26/19, 1:15:06 PM UTC-7

Open the file by clicking on its name. Your file contents should be similar to what is shown below.

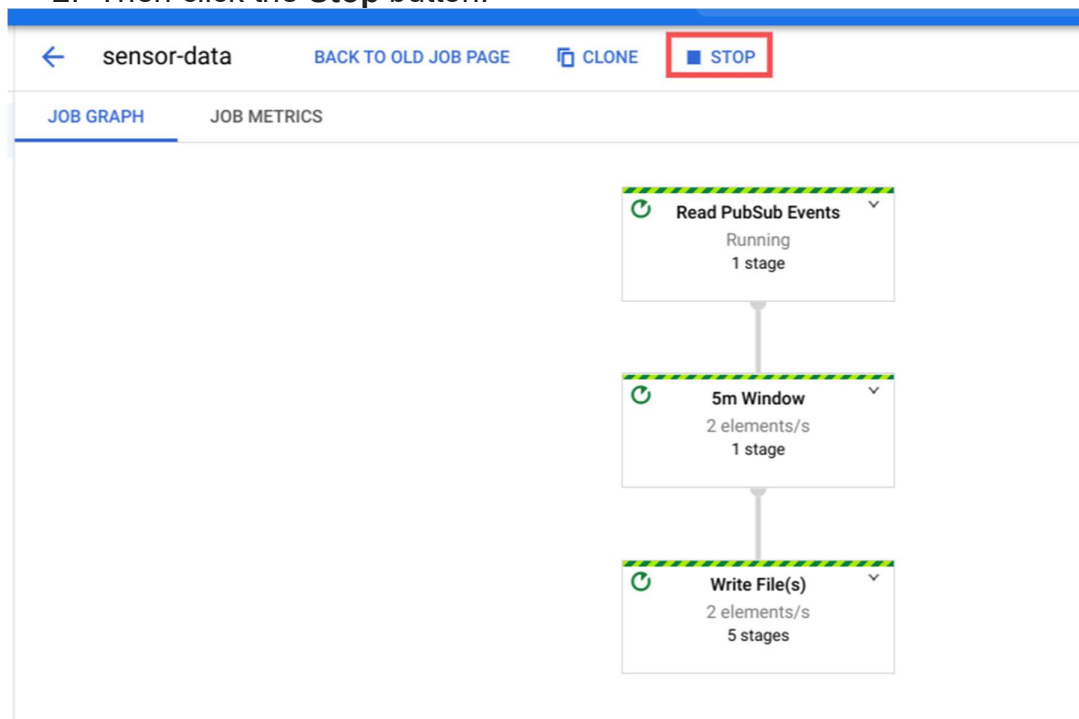
Output (do not copy)


```
{
  "device": "temp-sensor-buenos-aires",
  "timestamp": 1553631107,
  "temperature": 15.799777248019161
}
{
  "device": "temp-sensor-buenos-aires",
  "timestamp": 1553631108,
  "temperature": 15.787863403464552
}
{
  "device": "temp-sensor-buenos-aires",
  "timestamp": 1553631106,
  "temperature": 15.80888889949552
}
}
```

Stop the Dataflow job


In this section you will learn how to stop collecting data through Dataflow by stopping the provisioned job.

1. From the Navigation menu, click **Dataflow** and select the dataflow you created earlier.
2. Then click the **Stop** button:



3. In the dialog box, select **Drain**, then **Stop Job**. It will take a few minutes for the job to stop.


4. Using the back arrow, return to the Dataflow page. Make sure the job status of the sensor-data job is *Drained*:

 Dataflow

Jobs

[+ CREATE JOB FROM TEMPLATE](#)

Filter jobs

Name	Type	End time	Elapsed time	Start time	Status	SDK version	ID	Region
 sensor-data	Streaming	Mar 26, 2019, 1:25:39 PM	32 min 11 sec	Mar 26, 2019, 12:53:28 PM	Drained	2.10.0	2019-03-26_12_53_26-8003433111629408348	us-central1

Test your Understanding

Below are multiple-choice questions to reinforce your understanding of this lab's concepts. Answer them to the best of your abilities.

Cloud IoT Core supports two protocols for device connection and communication:

HTTP
MQTT

A Cloud Storage bucket name can only be modified after creation.

False

Congratulations!

In this lab you received hands-on practice with the Cloud Storage, Cloud Pub/Sub, Compute Engine, and IoT Core services. After creating a Pub/Sub topic and a Cloud Storage bucket, you created a Dataflow job from the Cloud PubSub to Text Files on Cloud Storage template. From there, you configured a virtual machine to stream telemetry data to two sample devices and a text file hosted in a Cloud Storage bucket.

Finish Your Quest



This self-paced lab is part of the Qwiklabs [IoT in the Google Cloud](#). 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 October 13, 2020

Lab Last Tested October 13, 2020

Copyright 2021 Google LLC All rights reserved. Google and the Google logo are trademarks of Google LLC. All other company and product names may be trademarks of the respective companies with which they are associated.