Building an IoT Analytics Pipeline on Google Cloud

GSP088

Google Cloud Self-Paced Labs

Overview

The term **Internet of Things (IoT)** refers to the interconnection of physical devices with the global Internet. These devices are equipped with sensors and networking hardware, and each is globally identifiable. Taken together, these capabilities afford rich data about items in the physical world.

Cloud IoT Core is a fully managed service that allows you to easily and securely connect, manage, and ingest data from millions of globally dispersed devices. The service connects IoT devices that use the standard `Message Queue Telemetry Transport (MQTT)` protocol to other Google Cloud data services.
Cloud IoT Core has two main components:

- A **device manager** for registering devices with the service, so you can then monitor and configure them.
- A **protocol bridge** that supports MQTT, which devices can use to connect to Google Cloud.

What You'll Learn

In this lab, you will dive into Google Cloud's IoT services and complete the following tasks:

- Connect and manage MQTT-based devices using Cloud IoT Core (using simulated devices)
- Ingest a stream of information from Cloud IoT Core using Cloud Pub/Sub.
- Process the IoT data using Cloud Dataflow.
- Analyze the IoT data using BigQuery.

Prerequisites

This is an **advanced level** lab and having a baseline level of knowledge on the following services will help you better understand the steps and commands that you'll be writing:

- **BigQuery**
- **Cloud Pub/Sub**
- **Dataflow**
- **IoT**

If you want to get up to speed in any of the above services, check out the following Qwiklabs:

- BigQuery: Qwik Start - Console
- Weather Data in BigQuery
- Dataflow: Qwik Start - Templates
- Internet of Things: Qwik Start

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

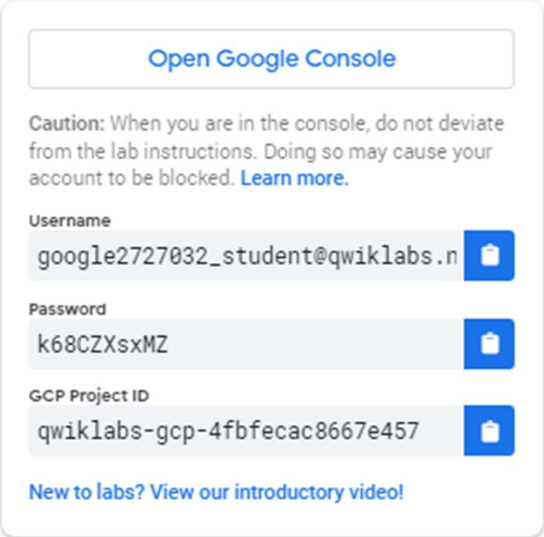
To complete this lab, you need:

- 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.



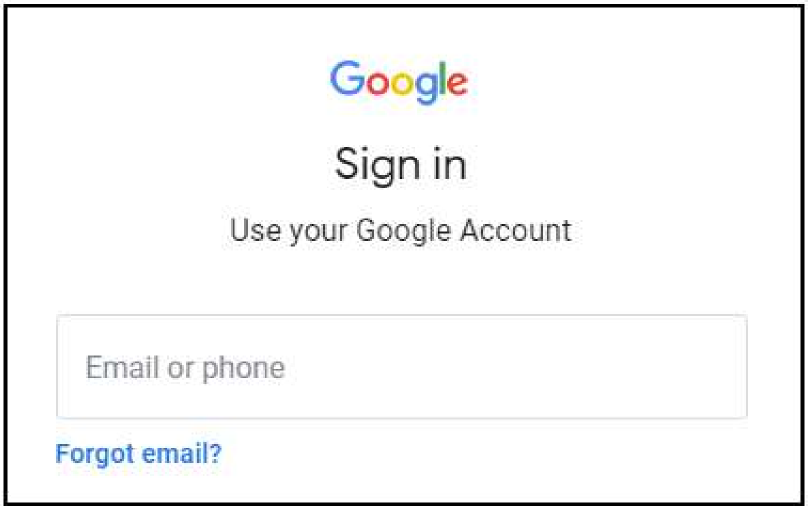
Caution: When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked. Learn more.

Username
google2727032_student@qwiklabs.n

Password
k68CZXsxMZ

GCP Project ID
qwiklabs-gcp-4fbfecac8667e457

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.



Google

Sign in

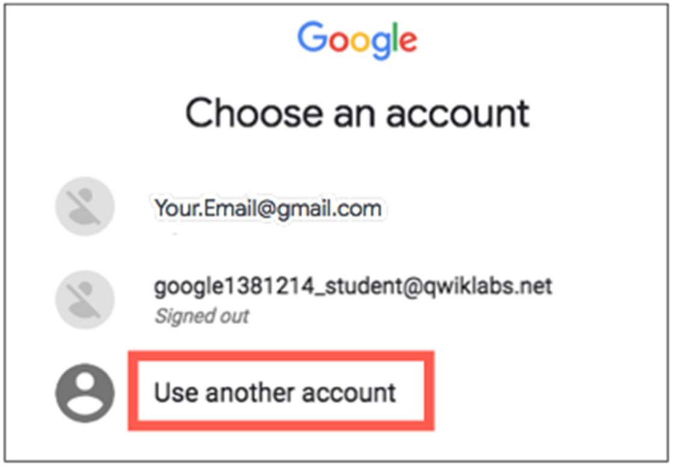
Use your Google Account

Email or phone

Forgot email?

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

If you see the **Choose an account** page, click **Use Another Account**.



Google

Choose an account

Your.Email@gmail.com

google1381214_student@qwiklabs.net
Signed out

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.

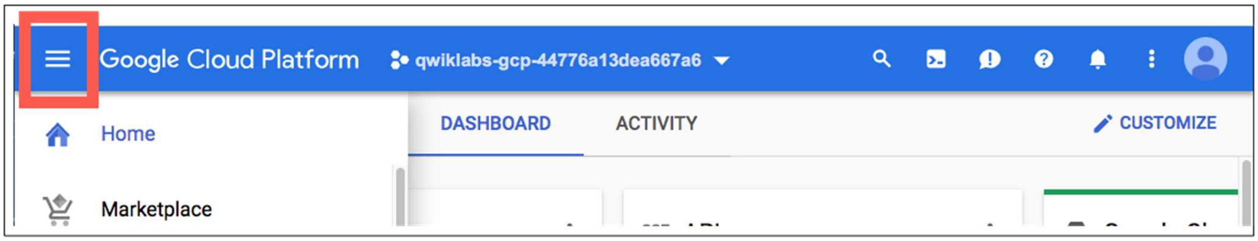
   Important: You must use the credentials from the Connection Details panel. Do not use your Qwiklabs credentials. If you have your own Google Cloud account, do not use it for this lab (avoids incurring charges).

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.

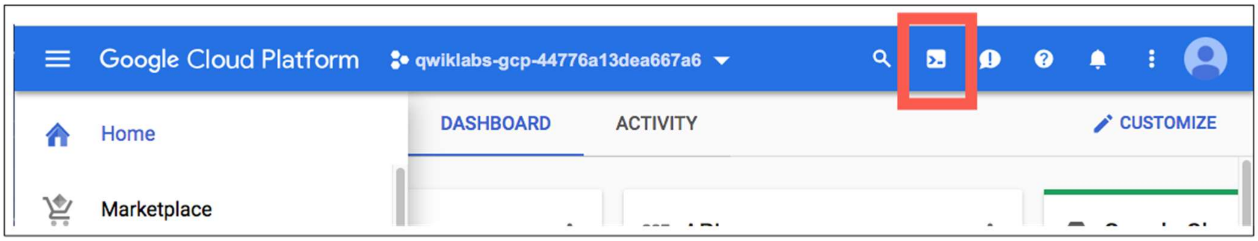
Note: You can view the menu with a list of Google Cloud Products and Services by clicking the **Navigation menu** at the top-left.



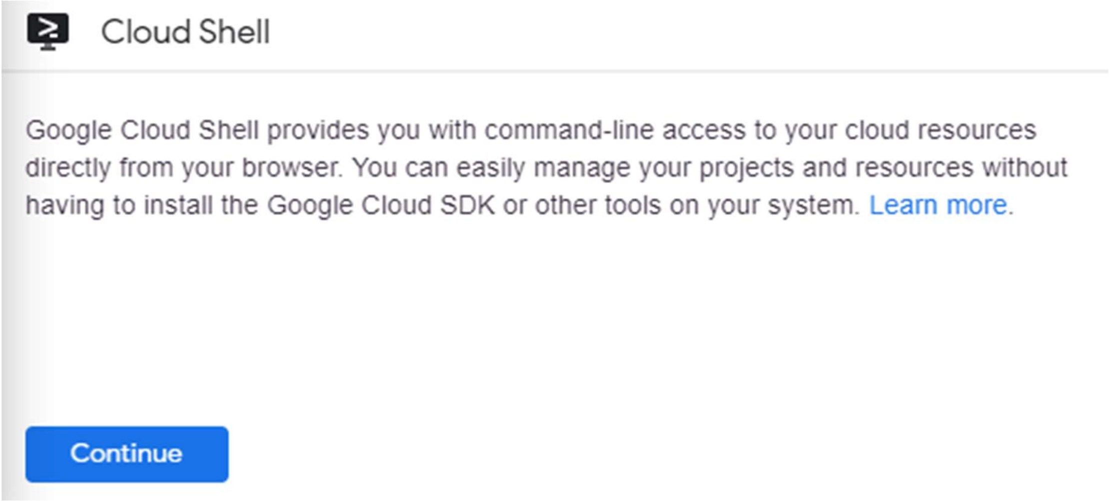
Activate Cloud Shell

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.



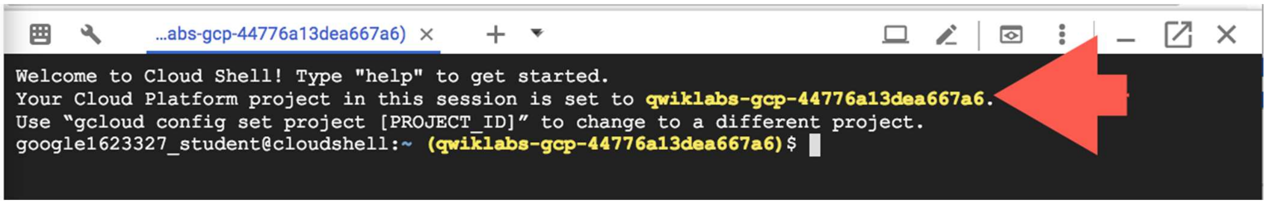
Click **Continue**.



Cloud Shell

Google Cloud Shell provides you with command-line access to your cloud resources directly from your browser. You can easily manage your projects and resources without having to install the Google Cloud SDK or other tools on your system. Learn more.

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:



```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to qwiklabs-gcp-44776a13dea667a6.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
google1623327_student@cloudshell:~ (qwiklabs-gcp-44776a13dea667a6)$
```

`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
```

(Output)

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

(Example output)

```
Credentialed accounts:
 - google1623327_student@qwiklabs.net
```

You can list the project ID with this command:

```
gcloud config list project
```

(Output)

```
[core]
project = <project_ID>
```

(Example output)

```
[core]
project = qwiklabs-gcp-44776a13dea667a6
```

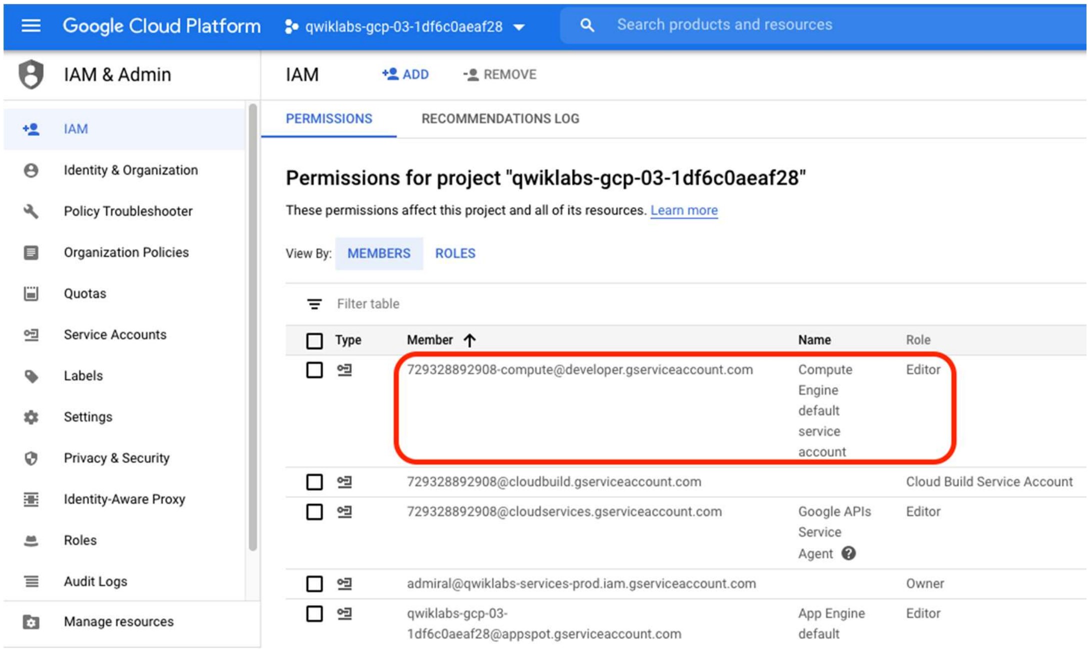
For full documentation of `gcloud` see the gcloud command-line tool overview.

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

View By: MEMBERS ROLES

Filter table

□ 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

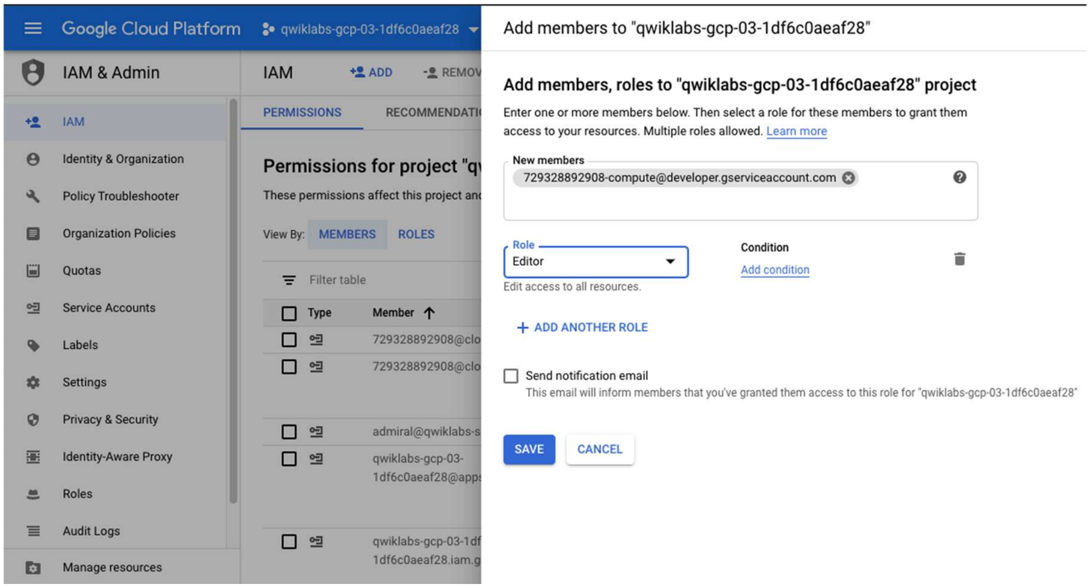
If the account is not present in IAM or does not have the `editor` role, follow the steps below to assign the required role.

- 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.

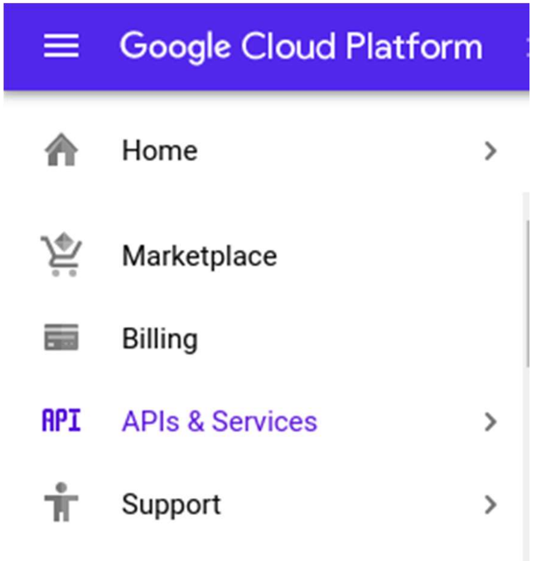
Condition: Add condition

□ Send notification email

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

Throughout this lab you'll be using your Project ID. You may want to copy and save it now for use later.

1. In the Cloud Console, click **Navigation menu** > **APIs & Services**.



2. Scroll down in the list of enabled APIs, and confirm that these APIs are enabled:

    - **Cloud IoT API**
    - **Cloud Pub/Sub API**
    - **Dataflow API**

3. If one or more API is not enabled, click the **ENABLE APIS AND SERVICES** button at the top. Search for the APIs by name and enable each API for your current project.

Create a Cloud Pub/Sub topic

Cloud Pub/Sub is an asynchronous global messaging service. By decoupling senders and receivers, it allows for secure and highly available communication between independently written applications. Cloud Pub/Sub delivers low-latency, durable messaging.
In Cloud Pub/Sub, publisher applications and subscriber applications connect with one another through the use of a shared string called a **topic**. A publisher application creates and sends messages to a topic. Subscriber applications create a subscription to a topic to receive messages from it.

In an IoT solution built with Cloud IoT Core, device telemetry data is forwarded to a Cloud Pub/Sub topic.

To define a new Cloud Pub/Sub topic:

1. In the Cloud Console, go to **Navigation menu** > **Pub/Sub** > **Topics**.
2. Click **+ CREATE TOPIC**. The **Create a topic** dialog shows you a partial URL path.

Note: If you see `qwiklabs-resources` as your project name, cancel the dialog and return to the Cloud Console. Use the menu to the right of the Google Cloud logo to select the correct project. Then return to this step.

3. Add this string as your topic name:

```
iotlab
```

4. Click **CREATE TOPIC**.

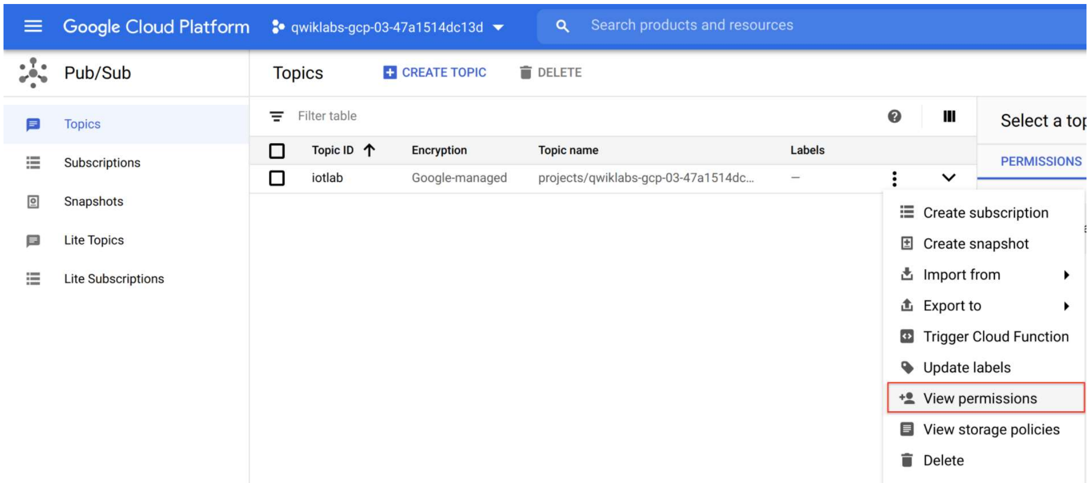
Test Completed Task

Click **Check my progress** to verify your performed task. If you have successfully created a Cloud Pub/Sub topic, you will see an assessment score.

Create a Cloud Pub/Sub Topic

Check my progress

5. In the list of topics, you will see a new topic whose partial URL ends in `iotlab`. Click the three-dot icon at the right edge of its row to open the context menu. Choose **View permissions**.



Topic ID	Encryption	Topic name	Labels
iotlab	Google-managed	projects/qwiklabs-gcp-03-47a1514dc...	–

6. In the **Permissions** dialogue, click **ADD MEMBER** and copy the below member as **New members**:

```
cloud-iot@system.gserviceaccount.com
```

7. From the `Select a role` menu, give the new member the **Pub/Sub** > **Pub/Sub Publisher** role.

8. Click **Save**.

Test Completed Task

Click **Check my progress** to verify your performed task. If you have successfully added IAM binding policy to Pub/Sub topic, you will see an assessment score.

Add IAM binding policy to Pub/Sub topic

Check my progress

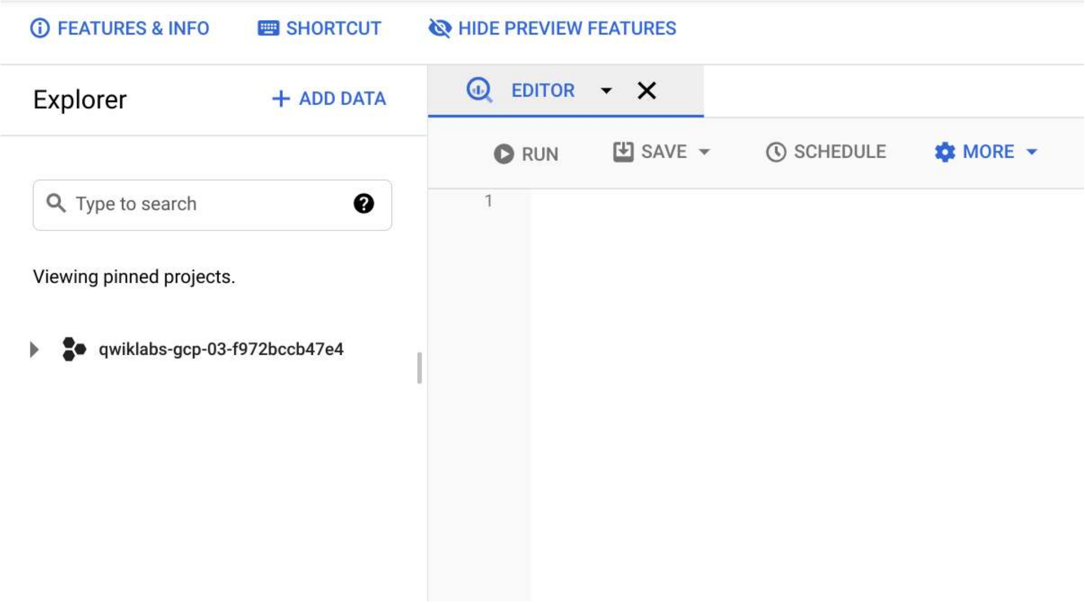
Create a BigQuery dataset

BigQuery is a serverless data warehouse. Tables in BigQuery are organized into datasets. In this lab, messages published into Pub/Sub will be aggregated and stored in BigQuery. To create a new BigQuery dataset:

1. In the Cloud Console, go to **Navigation menu** > **BigQuery**.

2. Click **Done**.

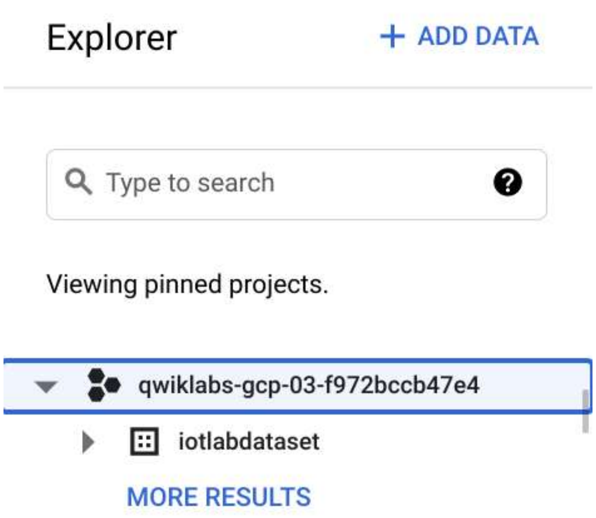
3. Click on your Project ID from the left-hand menu:



4. On the right-hand side of the console click **CREATE DATASET**.

5. Name the dataset **iotlabdataset**, leave all the other fields the way they are, and click **Create dataset**.

6. You should see your newly created dataset under your project:



Test Completed Task

Click **Check my progress** to verify your performed task. If you have successfully created a BigQuery dataset, you will see an assessment score.

Create a BigQuery dataset

Check my progress

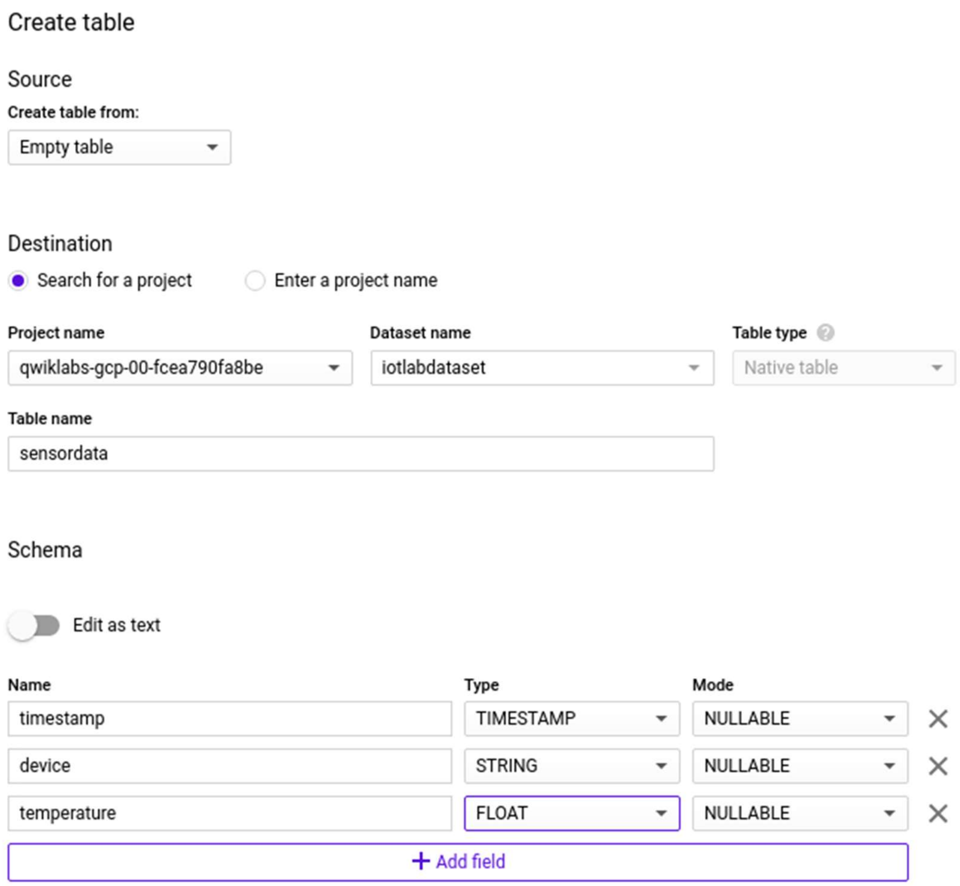
7. Click on your dataset, then on the right-hand side of the console click **+ CREATE TABLE**.

8. Ensure that the source field is set to **Empty table**.

9. In the **Destination** section's **Table name** field, enter `sensordata`.

10. In the **Schema** section, click the **+ Add field** button and add the following fields:

    - **timestamp**, set the field's **Type** to **TIMESTAMP**.
    - **device**, set the field's **Type** to **STRING**.
    - **temperature**, set the field's **Type** to **FLOAT**.
11. Leave the other defaults unmodified. Click **Create table**.



Create table

Source

Create table from: Empty table

Destination

● Search for a project ○ Enter a project name

Project name: qwiklabs-gcp-00-fcea790fa8be

Dataset name: iotlabdataset

Table type: Native table

Table name: sensordata

Schema

Edit as text

Name	Type	Mode
timestamp	TIMESTAMP	NULLABLE
device	STRING	NULLABLE
temperature	FLOAT	NULLABLE

Test Completed Task

Click **Check my progress** to verify your performed task. If you have successfully created an empty table in BigQuery Dataset, you will see an assessment score.

Create an empty table in BigQuery Dataset

Check my progress

Create a cloud storage bucket

Cloud Storage allows world-wide storage and retrieval of any amount of data at any time. You can use Cloud Storage for a range of scenarios including serving website content, storing data for archival and disaster recovery, or distributing large data objects to users via direct download.
For this lab Cloud Storage will provide working space for your Cloud Dataflow pipeline.

1. In the Cloud Console, go to **Navigation menu** > **Cloud Storage**.
2. Click **CREATE BUCKET**.
3. For **Name**, use your Project ID then add `-bucket`.
4. For **Location type**, click **Multi-region** if it is not already selected.
5. For **Location**, choose the selection closest to you.
6. Click **CREATE**.

Test Completed Task

Click **Check my progress** to verify your performed task. If you have successfully created a Cloud Storage bucket, you will see an assessment score.

Create a Cloud Storage Bucket

Check my progress

Set up a Cloud Dataflow Pipeline

Cloud Dataflow is a serverless way to carry out data analysis. In this lab, you will set up a streaming data pipeline to read sensor data from Pub/Sub, compute the maximum temperature within a time window, and write this out to BigQuery.

1. In the Cloud Console, go to **Navigation menu** > **Dataflow**.

2. In the top menu bar, click **+ CREATE JOB FROM TEMPLATE**.

3. In the job-creation dialog, for **Job name**, enter `iotlabflow`.

4. For **Regional Endpoint**, choose the region as **us-central1**.

5. For **Dataflow template**, choose **Pub/Sub Topic to BigQuery**. When you choose this template, the form updates to review new fields below.

6. For **Input Pub/Sub topic**, enter `projects/` followed by your Project ID then add `/topics/iotlab`. The resulting string will look like this: `projects/qwiklabs-gcp-d2e509fed105b3ed/topics/iotlab`

7. The **BigQuery output table** takes the form of Project ID:dataset.table (`:iotlabdataset.sensordata`). The resulting string will look like this: `qwiklabs-gcp-d2e509fed105b3ed:iotlabdataset.sensordata`

8. For **Temporary location**, enter `gs://` followed by your Cloud Storage bucket name (should be your Project ID if you followed the instructions) then `/tmp/`. The resulting string will look like this: `gs://qwiklabs-gcp-d2e509fed105b3ed-bucket/tmp/`

9. Click **SHOW OPTIONAL PARAMETERS**.

10. For Max workers, enter **2**.

11. For Machine type, enter **n1-standard-1**.

12. Click **RUN JOB**.

A new streaming job is started. You can now see a visual representation of the data pipeline.

Test Completed Task

Click **Check my progress** to verify your performed task. If you have successfully set up a Cloud Dataflow Pipeline, you will see 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** > **Compute Engine** > **VM Instances**. You'll see your VM instance listed as `iot-device-simulator`.

2. Click the **SSH** drop-down arrow and select **Open in browser window**.

3. In your SSH session, enter following commands to create a virtual environment.

```
sudo pip install virtualenv
virtualenv -p python3 venv
source venv/bin/activate
```

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

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

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

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

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

```
exit
```

7. Start another SSH session on the `iot-device-simulator` VM instance and execute the following command to activate the virtual environment.

```
source venv/bin/activate
```

8. 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.

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

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

9. Click on the URL shown to open a new browser window that displays a verification code.

10. Copy the verification code and paste it in response to the "Enter verification code:" prompt, then press **Enter**.

11. In response to "Pick cloud project to use", pick your lab project.

12. Enter this command to make sure that the components of the SDK are up to date:

```
gcloud components update
```

13. Enter this command to install the beta components:

```
gcloud components install beta
```

If you are asked "Do you want to continue (Y/n)?" enter "Y".

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

```
sudo apt-get update
```

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

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

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

```
pip install pyjwt paho-mqtt cryptography
```

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

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

Create a registry for IoT devices

To register devices, you must create a registry for the devices. The registry is a point of control for devices.

To create the registry:

1. 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`

2. You must choose a region for your IoT registry. At this time, these regions are supported:

- `us-central1`
- `europe-west1`
- `asia-east1`

Choose the region that is closest to you. To set an environment variable containing your preferred region, enter this command followed by the region name:

```
export MY_REGION=
```

Your completed command will look like this: `export MY_REGION=us-central1`.

3. Enter this command to create the device registry:

```
gcloud beta iot registries create iotlab-registry \
   --project=$PROJECT_ID \
   --region=$MY_REGION \
   --event-notification-config=topic=projects/$PROJECT_ID/topics/iotlab
```

Test Completed Task

Click **Check my progress** to verify your performed task. If you have successfully created a Registry for IoT Devices, you will see an assessment score.

Create a Registry for IoT Devices

Check my progress

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`.

Add simulated devices to the registry

For a device to be able to connect to Cloud IoT Core, it must first be added to the registry.

1. In your SSH session on the **iot-device-simulator** VM instance, enter this command to create a device called `temp-sensor-buenos-aires`:

```
gcloud beta iot devices create temp-sensor-buenos-aires \
  --project=$PROJECT_ID \
  --region=$MY_REGION \
  --registry=iotlab-registry \
  --public-key path=rsa_cert.pem,type=rs256
```

2. Enter this command to create a device called `temp-sensor-istanbul`:

```
gcloud beta iot devices create temp-sensor-istanbul \
  --project=$PROJECT_ID \
  --region=$MY_REGION \
  --registry=iotlab-registry \
  --public-key path=rsa_cert.pem,type=rs256
```

Test Completed Task

Click **Check my progress** to verify your performed task. If you have successfully added Simulated Devices to the Registry, you will see 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:

```
python 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 > buenos-aires-log.txt 2>&1 &
```

It will continue to run in the background.

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

```
python 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
```

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.

Analyze the Sensor Data Using BigQuery

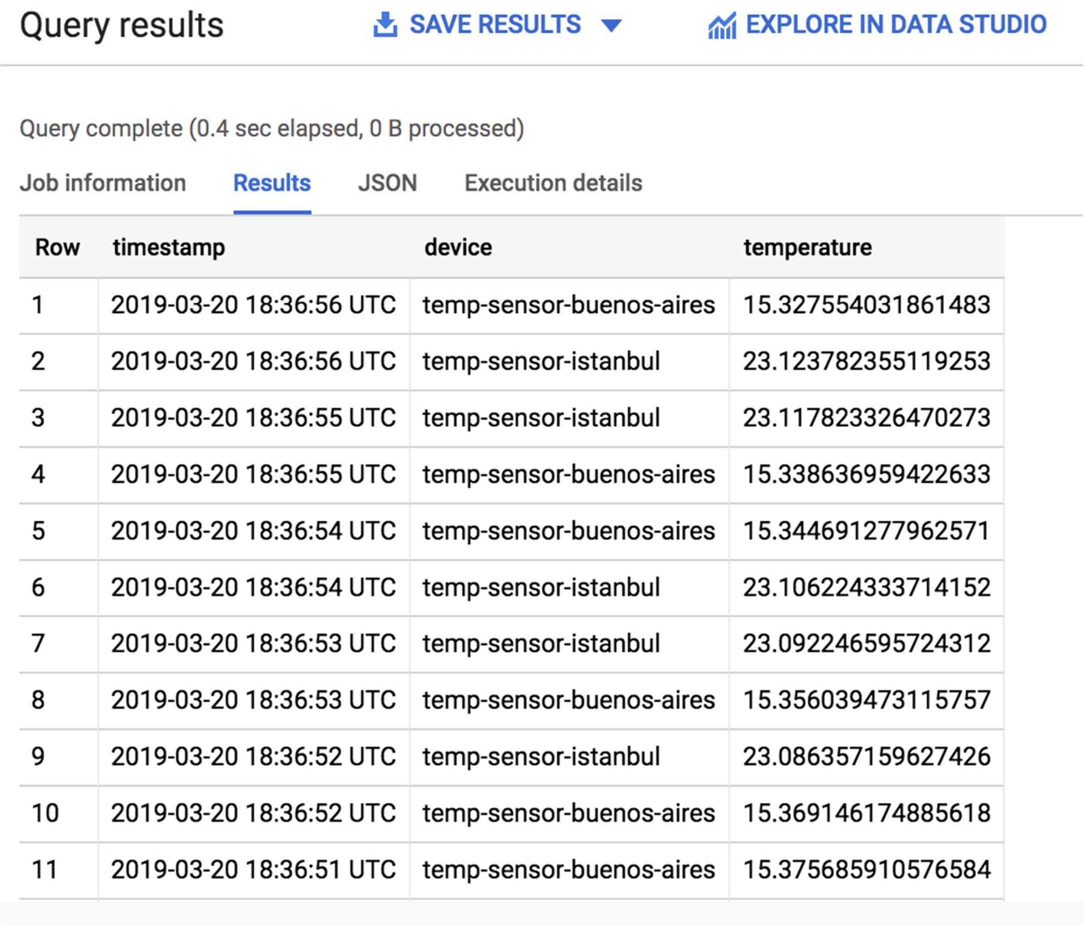
To analyze the data as it is streaming:

1. In the Cloud Console, open the **Navigation menu** and select **BigQuery**.

2. Enter the following query in the Query editor and click **RUN**:

```
SELECT timestamp, device, temperature from iotlabdataset.sensordata
ORDER BY timestamp DESC
LIMIT 100
```

3. You should receive a similar output:



Query complete (0.4 sec elapsed, 0 B processed)

Row	timestamp	device	temperature
1	2019-03-20 18:36:56 UTC	temp-sensor-buenos-aires	15.327554031861483
2	2019-03-20 18:36:56 UTC	temp-sensor-istanbul	23.123782355119253
3	2019-03-20 18:36:55 UTC	temp-sensor-istanbul	23.117823326470273
4	2019-03-20 18:36:55 UTC	temp-sensor-buenos-aires	15.338636959422633
5	2019-03-20 18:36:54 UTC	temp-sensor-buenos-aires	15.344691277962571
6	2019-03-20 18:36:54 UTC	temp-sensor-istanbul	23.106224333714152
7	2019-03-20 18:36:53 UTC	temp-sensor-istanbul	23.092246595724312
8	2019-03-20 18:36:53 UTC	temp-sensor-buenos-aires	15.356039473115757
9	2019-03-20 18:36:52 UTC	temp-sensor-istanbul	23.086357159627426
10	2019-03-20 18:36:52 UTC	temp-sensor-buenos-aires	15.369146174885618
11	2019-03-20 18:36:51 UTC	temp-sensor-buenos-aires	15.375685910576584

4. Browse the Results. What is the temperature trend at each of the locations?

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.

□ d IoT Core supports two protocols for device connection and communication:

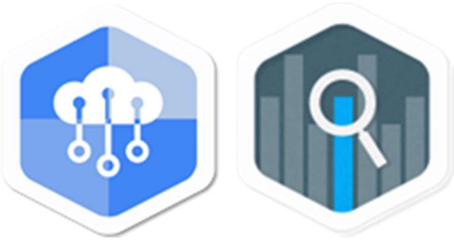
□ 'P

MQTT

When you create a device within a registry, you define the device as a Cloud IoT Core resource.
True

Congratulations!

In this lab, you used Cloud IoT Core and other services on Google Cloud to collect, process, and analyze IoT data.



Finish your Quest

This self-paced lab is part of the Qwiklabs IoT in the Google Cloud and Data Engineering Quests. A Quest is a series of related labs that form a learning path. Completing a Quest earns you a badge 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 a Quest and get immediate completion credit if you've taken this lab. See other available Qwiklabs Quests.

Take Your Next Lab

Continue your Quest with ETL Processing on Google Cloud Using Dataflow and BigQuery, or check out these suggestions:

- Predict Visitor Purchases with a Classification Model in BQML
- Predict Housing Prices with Tensorflow and AI Platform

Next steps

- Read the Getting Started guide for Cloud IoT Core.
- For more information, see Overview of Internet of Things

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 19, 2021

Lab Last Tested April 19, 2021