

# Streaming Analytics into BigQuery: Challenge Lab

## Task 1. Create a Cloud Storage bucket

- Create a Cloud Storage bucket using your Project ID as the bucket name: `qwiklabs-gcp-00-6d813e62ffff`

◆ 1. Go to Cloud Storage

- Open the Google Cloud Console Storage Browser

◆ 2. Click “Create” or “Create Bucket”

◆ 3. Configure Bucket Settings


Setting	Value
<b>Name</b>	<code>qwiklabs-gcp-00-6d813e62ffff</code> 🔗 Must match your project ID exactly
<b>Location type</b>	<code>Multi-region</code> or <code>Region</code> (default is fine)
<b>Location</b>	Choose <code>US</code> unless the lab specifies otherwise
<b>Storage class</b>	<code>Standard</code> (default is fine)
<b>Access control</b>	<code>Uniform</code> (default is fine)

Click “**Create**” once done.

## Task 2. Create a BigQuery dataset and table

1. Create a BigQuery dataset called `BigQuery dataset name` in the region named **US (multi region)**.
2. In the created dataset, create a table called `BigQuery table name` and **add** column `data` with `STRING` type.

### 1. Open BigQuery

- Go to the Google Cloud Console
- In the top left menu () go to **BigQuery** under “Big Data” or search for **BigQuery** in the search bar.

### 2. Create a Dataset

1. In the BigQuery console, click on your project name in the left panel.
2. Click **“Create dataset.”**
3. Fill in the following:
  - **Dataset ID:** `sensors_451`
  - **Data location:** `US` (make sure it is **multi-region: US**)
  - Leave other settings as default.
4. Click **Create dataset.**

### 3. Create a Table in the Dataset

1. In the left panel, click on the dataset `sensors_451`.
2. Click **“Create table.”**
3. For **Create table from**, choose:
  - **Empty table**
4. Under **Table name**, enter: `temperature_669`
5. Under **Schema**, click **+ Add field** and add:
  - **Name:** `data`
  - **Type:** `STRING`
6. Leave all other fields as default.
7. Click **Create table.**

## Task 3. Set up a Pub/Sub topic

1. Create a Pub/Sub topic called `Pub/Sub topic name`.
- Use the default settings, which has enabled the checkbox for **Add a default subscription**.

### ◆ Step-by-Step Instructions:

#### 1. Open Pub/Sub in the Console

- Go to Pub/Sub Topics

#### 2. Click “Create Topic”

*Fill out the form:*

- **Name your topic:**

`Pub/Sub topic name`

*(Exactly as shown, unless the lab tells you otherwise)*

- **Leave the checkbox checked** for:  
✓ *Add a default subscription*
- Leave all other settings as default.

#### 3. Click Create

## Task 4. Run a Dataflow pipeline to stream data from Pub/Sub to BigQuery

1. Create and run a Dataflow job called `Dataflow job name` to stream data from Pub/Sub topic to BigQuery, using the Pub/Sub topic and BigQuery table you created in the previous tasks.

- Use the **Custom Dataflow Template**.
- Use the below Path for the template file stored in Cloud Storage:

```
gs://dataflow-templates-Region/latest/PubSub_to_BigQuery  
Copied!
```

```
content_copy
```

- Use the Pub/Sub topic that you created in a previous task: `Pub/Sub topic name`
- Use the Cloud Storage bucket that you created in a previous task as the temporary location: `<filled in at lab start>`
- Use the BigQuery dataset and table that you created in a previous task as the output table: `BigQuery dataset name.BigQuery table name`
- Use `Region` as the regional endpoint.

### ◆ 1. Open the Dataflow Console

- Go to Dataflow in Google Cloud Console
- Click **“Create job from template”**

### ◆ 2. Fill in Job Configuration

Field	Value
Job name	dfjob-90510
Region	us-central1
Dataflow template	Select <b>“Custom template”</b>

Field	Value
Template path	gs://dataflow-templates-us-central1/latest/PubSub_to_BigQuery

### ◆ 3. Set Template Parameters

Enter the required parameters as follows:

```

✓ inputTopic:
bash
CopyEdit
projects/qwiklabs-gcp-01-a626ae113e2d/topics/sensors-temp-18715
✓ outputTableSpec:
makefile
CopyEdit
qwiklabs-gcp-01-a626ae113e2d:sensors_292.temperature_208
✓ outputDeadletterTable (Optional, but good practice):
makefile
CopyEdit
qwiklabs-gcp-01-a626ae113e2d:sensors_292.deadletter_table

```

You can skip this if the template allows, or create the `deadletter_table` manually beforehand.

### ◆ 4. Set Temporary Location

**Temporary location:**

```

bash
CopyEdit
gs://qwiklabs-gcp-01-a626ae113e2d/temp

```

If the `temp` folder doesn't exist, it will be created automatically.

### ◆ 5. Run the Job

- Click **"Run job"**
- Wait for the pipeline to start (it may take a minute or two).

## Task 5. Publish a test message to the topic and validate data in BigQuery

1. Publish a message to your topic using the following code syntax for **Message**: `{"data": "73.4 F"}`
- Note: `73.4 F` can be replaced with any value.
2. Run a **SELECT** statement in BigQuery to see the test message populated in your table.
- Note:** If you do not see any test messages in your BigQuery table, check that the Dataflow job has a status of *Running*, and then send another test message.

### 🔗 Option 1: Use Google Cloud Console (UI)

1. Go to **Pub/Sub > Topics**
2. Click your topic: `sensors-temp-18715`
3. Click **"Publish message"**
4. In the **Message body**, paste:

```
{"data": "73.4 F"}
```

5. Click **Publish**

### 🔗 Option 2: Use Cloud Shell (CLI)

Open **Cloud Shell** and run:

```
gcloud pubsub topics publish sensors-temp-18715 --message='{"data": "73.4 F"}'
```

### ✓ Step 2: Check Data in BigQuery

1. Go to **BigQuery > Select your project > Dataset: sensors\_292**
2. Click on the table: `temperature_208`
3. Click **"Query" > "In new tab"**

Paste and run the SQL query:

```
SELECT * FROM `qwiklabs-gcp-01-a626ae113e2d.sensors_292.temperature_208`  
ORDER BY _PARTITIONTIME DESC LIMIT 10;
```