Pub/Sub: Qwik Start - Python

experimentLabschedule30 minutesuniversal\_currency\_altNo costshow\_chartIntroductory

infoThis lab may incorporate AI tools to support your learning.

**GSP094**



**Overview**

The Pub/Sub service allows applications to exchange messages reliably, quickly, and asynchronously. To accomplish this, a data producer publishes messages to a Cloud Pub/Sub topic. A subscriber client then creates a subscription to that topic and consumes messages from the subscription. Cloud Pub/Sub persists messages that could not be delivered reliably for up to seven days.

In this lab, you will learn how to get started publishing messages with Pub/Sub using the Python client library.

What you'll do

In this lab, you'll do the following:

* Learn the basics of Pub/Sub
* Create, delete, and list Pub/Sub topics and subscriptions
* Publish messages to a topic.
* Use a pull subscriber to output individual topic messages.

**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 are made available to you.

This hands-on lab lets you do the lab activities in a real cloud environment, not in a simulation or demo environment. It does so by giving you new, temporary credentials you use to sign in and access Google Cloud for the duration of the lab.

To complete this lab, you need:

* Access to a standard internet browser (Chrome browser recommended).

**Note:** Use an Incognito (recommended) or private browser window to run this lab. This prevents conflicts between your personal account and the student account, which may cause extra charges incurred to your personal account.

* Time to complete the lab—remember, once you start, you cannot pause a lab.

**Note:** Use only the student account for this lab. If you use a different Google Cloud account, you may incur charges to that account.

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 dialog opens for you to select your payment method. On the left is the Lab Details pane with the following:
   * The Open Google Cloud console button
   * Time remaining
   * The temporary credentials that you must use for this lab
   * Other information, if needed, to step through this lab
2. Click **Open Google Cloud console** (or right-click and select **Open Link in Incognito Window** if you are running the Chrome browser).

The lab spins up resources, and then opens another tab that shows the Sign in page.

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

**Note:**If you see the **Choose an account** dialog, click **Use Another Account**.

1. If necessary, copy the **Username** below and paste it into the **Sign in** dialog.

student-02-3702f9847c2e@qwiklabs.net

Copied!

You can also find the Username in the Lab Details pane.

1. Click **Next**.
2. Copy the **Password** below and paste it into the **Welcome** dialog.

ls7Jh2daTJ2L

Copied!

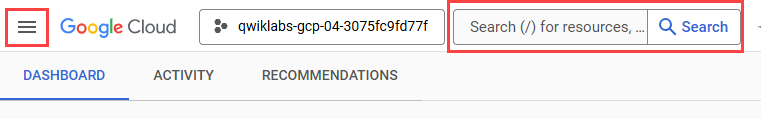
You can also find the Password in the Lab Details pane.

1. Click **Next**.

**Important:**You must use the credentials the lab provides you. Do not use your Google Cloud account credentials.**Note:**Using your own Google Cloud account for this lab may incur extra charges.

1. 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 Google Cloud console opens in this tab.

**Note:** To access Google Cloud products and services, click the **Navigation menu** or type the service or product name in the **Search** field. 

Activate Cloud Shell

Cloud Shell is a virtual machine that is loaded with development tools. It offers a persistent 5GB home directory and runs on the Google Cloud. Cloud Shell provides command-line access to your Google Cloud resources.

1. Click **Activate Cloud Shell** Activate Cloud Shell icon at the top of the Google Cloud console.
2. Click through the following windows:
   * Continue through the Cloud Shell information window.
   * Authorize Cloud Shell to use your credentials to make Google Cloud API calls.

When you are connected, you are already authenticated, and the project is set to your **Project\_ID**, qwiklabs-gcp-00-cac1c74eac15. The output contains a line that declares the **Project\_ID** for this session:

Your Cloud Platform project in this session is set to qwiklabs-gcp-00-cac1c74eac15

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

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

gcloud auth list

Copied!

1. Click **Authorize**.

**Output:**

ACTIVE: \*

ACCOUNT: student-02-3702f9847c2e@qwiklabs.net

To set the active account, run:

$ gcloud config set account `ACCOUNT`

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

gcloud config list project

Copied!

**Output:**

[core]

project = qwiklabs-gcp-00-cac1c74eac15

**Note:**For full documentation of gcloud, in Google Cloud, refer to [the gcloud CLI overview guide](https://cloud.google.com/sdk/gcloud).

**Task 1. Create a virtual environment**

Python virtual environments are used to isolate package installation from the system.

1. Install the virtualenv environment:

sudo apt-get install -y virtualenv

Copied!

1. Build the virtual environment:

python3 -m venv venv

Copied!

1. Activate the virtual environment:

source venv/bin/activate

Copied!

**Task 2. Install the client library**

1. Run the following to install the client library:

pip install --upgrade google-cloud-pubsub

Copied!

1. Get the sample code by cloning a GitHub repository:

git clone https://github.com/googleapis/python-pubsub.git

Copied!

1. Navigate to the directory:

cd python-pubsub/samples/snippets

Copied!

**Task 3. Pub/Sub - the Basics**

Pub/Sub is an asynchronous global messaging service. There are three terms in Pub/Sub that appear often: *topics*, *publishing*, and *subscribing*.

A topic is a shared string that allows applications to connect with one another through a common thread.

Publishers push (or publish) a message to a Pub/Sub topic. Subscribers will then make a *subscription* to that thread, where they will either pull messages from the topic or configure webhooks for push subscriptions. Every subscriber must acknowledge each message within a configurable window of time.

In sum, a publisher creates and sends messages to a topic and a subscriber creates a subscription to a topic to receive messages from it.

Pub/Sub in Google CLoud

Pub/Sub comes preinstalled in Cloud Shell, so there are no installations or configurations required to get started with this service. In this lab you use Python to create the topic, subscriber, and then view the message. You use a gcloud command to publish the message to the topic.

**Task 4. Create a topic**

To publish data to Pub/Sub you create a topic and then configure a publisher to the topic.

1. In Cloud Shell, your Project ID should automatically be stored in the environment variable GOOGLE\_CLOUD\_PROJECT:

echo $GOOGLE\_CLOUD\_PROJECT

Copied!

1. Ensure the output is the same as the Project ID in your CONNECTION DETAILS.

publisher.py is a script that demonstrates how to perform basic operations on topics with the Cloud Pub/Sub API. View the content of publisher script:

cat publisher.py

Copied!

**Note:** Alternatively, you can use the shell editors that are installed on Cloud Shell, such as nano or vim or use the Cloud Shell code editor to view python-pubsub/samples/snippets/publisher.py.

1. For information about the publisher script:

python publisher.py -h

Copied!

*Example output:*

usage: publisher.py [-h]

project

{list,create,delete,publish,publish-with-custom-attributes,publish-with-futures,publish-with-error-handler,publish-with-batch-settings}

...

This application demonstrates how to perform basic operations on topics

with the Cloud Pub/Sub API.

For more information, see the README.md under /pubsub and the documentation

at https://cloud.google.com/pubsub/docs.

positional arguments:

project Your Google Cloud project ID

{list,create,delete,publish,publish-with-custom-attributes,publish-with-futures,publish-with-error-handler,publish-with-batch-settings}

list Lists all Pub/Sub topics in the given project.

create Create a new Pub/Sub topic.

delete Deletes an existing Pub/Sub topic.

publish Publishes multiple messages to a Pub/Sub topic.

publish-with-custom-attributes

Publishes multiple messages with custom attributes to

a Pub/Sub topic.

publish-with-futures

Publishes multiple messages to a Pub/Sub topic and

prints their message IDs.

publish-with-error-handler

Publishes multiple messages to a Pub/Sub topic with an

error handler.

publish-with-batch-settings

Publishes multiple messages to a Pub/Sub topic with

batch settings.

optional arguments:

-h, --help show this help message and exit

1. Run the publisher script to create Pub/Sub Topic:

python publisher.py $GOOGLE\_CLOUD\_PROJECT create MyTopic

Copied!

*Example output:*

Topic created: name: "projects/qwiklabs-gcp-fe27729bc161fb22/topics/MyTopic"

**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.

Assessment Completed! Topic count: 1 Topic name(s): ["MyTopic"]

Create a topic.

*Assessment Completed! Topic count: 1 Topic name(s): ["MyTopic"]*

1. This command returns a list of all Pub/Sub topics in a given project:

python publisher.py $GOOGLE\_CLOUD\_PROJECT list

Copied!

*Example output:*

name: "projects/qwiklabs-gcp-fe27729bc161fb22/topics/MyTopic"

You can also view the topic you just made in the Cloud Console.

1. Navigate to **Navigation menu** > **Pub/Sub** > **Topics**.

You should see MyTopic.

**Task 5. Create a subscription**

1. Create a Pub/Sub subscription for topic with subscriber.py script:

python subscriber.py $GOOGLE\_CLOUD\_PROJECT create MyTopic MySub

Copied!

**Test Completed Task**

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

Assessment Completed! Subscription present in the topic.

Create a subscription.

*Assessment Completed! Subscription present in the topic.*

1. This command returns a list of subscribers in given project:

python subscriber.py $GOOGLE\_CLOUD\_PROJECT list-in-project

Copied!

You'll see only one subscription because you've made only one subscription.

*Example output:*

projects/qwiklabs-gcp-7877af129f04d8b3/subscriptions/MySub

1. Check out the subscription you just made in the console. In the left pane, click **Subscriptions**. You should see the subscription name and other details.
2. For information about the subscriber script:

python subscriber.py -h

Copied!

*Output:*

usage: subscriber.py [-h]

project

{list\_in\_topic,list\_in\_project,create,create-push,delete,update,receive,receive-custom-attributes,receive-flow-control,receive-synchronously,listen\_for\_errors}

...

This application demonstrates how to perform basic operations on

subscriptions with the Cloud Pub/Sub API.

For more information, see the README.md under /pubsub and the documentation

at https://cloud.google.com/pubsub/docs.

positional arguments:

project Your Google Cloud project ID

{list\_in\_topic,list\_in\_project,create,create-push,delete,update,receive,receive-custom-attributes,receive-flow-control,receive-synchronously,listen\_for\_errors}

list\_in\_topic Lists all subscriptions for a given topic.

list\_in\_project Lists all subscriptions in the current project.

create Create a new pull subscription on the given topic.

create-push Create a new push subscription on the given topic.

delete Deletes an existing Pub/Sub topic.

update Updates an existing Pub/Sub subscription's push

endpoint URL. Note that certain properties of a

subscription, such as its topic, are not modifiable.

receive Receives messages from a pull subscription.

receive-custom-attributes

Receives messages from a pull subscription.

receive-flow-control

Receives messages from a pull subscription with flow

control.

receive-synchronously

Pulling messages synchronously.

listen\_for\_errors Receives messages and catches errors from a pull

subscription.

optional arguments:

-h, --help show this help message and exit

**Task 6. Publish messages**

Now that you've set up MyTopic (the topic) and a subscription to MyTopic (MySub), use gcloud commands to publish a message to MyTopic.

1. Publish the message "Hello" to MyTopic:

gcloud pubsub topics publish MyTopic --message "Hello"

Copied!

1. Publish a few more messages to MyTopic—run the following commands (replacing <YOUR NAME> with your name and <FOOD> with a food you like to eat):

gcloud pubsub topics publish MyTopic --message "Publisher's name is <YOUR NAME>"

Copied!

gcloud pubsub topics publish MyTopic --message "Publisher likes to eat <FOOD>"

Copied!

gcloud pubsub topics publish MyTopic --message "Publisher thinks Pub/Sub is awesome"

Copied!

**Task 7. View messages**

Now that you've published messages to MyTopic, pull and view the messages using MySub.

1. Use MySub to pull the message from MyTopic:

python subscriber.py $GOOGLE\_CLOUD\_PROJECT receive MySub

Copied!

*Example output:*

Listening for messages on projects/qwiklabs-gcp-7877af129f04d8b3/subscriptions/MySub

Received message: Message {

data: 'Publisher thinks Pub/Sub is awesome'

attributes: {}

}

Received message: Message {

data: 'Hello'

attributes: {}

}

Received message: Message {

data: "Publisher's name is Harry"

attributes: {}

}

Received message: Message {

data: 'Publisher likes to eat cheese'

attributes: {}

}

1. Click **Ctrl**+**c** to stop listening.

**Task 8. 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.

Google Cloud Pub/Sub service allows applications to exchange messages reliably, quickly, and asynchronously.

checkTrue

False

A \_\_\_\_\_ is a shared string that allows applications to connect with one another.

message

subscription

checktopic

Lab2

**Task 1. Pub/Sub topics**

Pub/Sub comes preinstalled in Cloud Shell, so there are no installations or configurations required to get started with this service.

1. Run the following command to create a topic called myTopic:

gcloud pubsub topics create myTopic

Copied!

**Test completed task**

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

Create a Pub/Sub topic.

1. For good measure, create two more topics; one called Test1 and the other called Test2:

gcloud pubsub topics create Test1

Copied!

gcloud pubsub topics create Test2

Copied!

1. To see the three topics you just created, run the following command:

gcloud pubsub topics list

Copied!

Your output should resemble the following:

---

messageStoragePolicy:

allowedPersistenceRegions:

- us-central1

name: projects/qwiklabs-gcp-01-af5b4aaa2d32/topics/myTopic

---

messageStoragePolicy:

allowedPersistenceRegions:

- us-central1

name: projects/qwiklabs-gcp-01-af5b4aaa2d32/topics/Test1

---

messageStoragePolicy:

allowedPersistenceRegions:

- us-central1

name: projects/qwiklabs-gcp-01-af5b4aaa2d32/topics/Test2

1. Time to clean up. Delete Test1 and Test2 by running the following commands:

gcloud pubsub topics delete Test1

Copied!

gcloud pubsub topics delete Test2

Copied!

1. Run the gcloud pubsub topics list command one more time to verify the topics were deleted:

gcloud pubsub topics list

Copied!

You should get the following output:

---

name: projects/qwiklabs-gcp-3450558d2b043890/topics/myTopic

**Task 2. Pub/Sub subscriptions**

Now that you're comfortable creating, viewing, and deleting topics, time to work with subscriptions.

1. Run the following command to create a subscription called mySubscription to topic myTopic:

gcloud pubsub subscriptions create --topic myTopic mySubscription

Copied!

**Test completed task**

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

Create Pub/Sub Subscription.

1. Add another two subscriptions to myTopic. Run the following commands to make Test1 and Test2 subscriptions:

gcloud pubsub subscriptions create --topic myTopic Test1

Copied!

gcloud pubsub subscriptions create --topic myTopic Test2

Copied!

1. Run the following command to list the subscriptions to myTopic:

gcloud pubsub topics list-subscriptions myTopic

Copied!

Your output should resemble the following:

---

projects/qwiklabs-gcp-3450558d2b043890/subscriptions/Test2

---

projects/qwiklabs-gcp-3450558d2b043890/subscriptions/Test1

---

projects/qwiklabs-gcp-3450558d2b043890/subscriptions/mySubscription

**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.

To receive messages published to a topic, you must create a subscription to that topic.

checkTrue

False

1. Now delete the Test1 and Test2 subscriptions. Run the following commands:

gcloud pubsub subscriptions delete Test1

Copied!

gcloud pubsub subscriptions delete Test2

Copied!

1. See if the Test1 and Test2 subscriptions were deleted. Run the list-subscriptions command one more time:

gcloud pubsub topics list-subscriptions myTopic

Copied!

You should get the following output:

---

projects/qwiklabs-gcp-3450558d2b043890/subscriptions/mySubscription

**Task 3. Pub/Sub publishing and pulling a single message**

Next you'll learn how to publish a message to a Pub/Sub topic.

1. Run the following command to publish the message "hello" to the topic you created previously (myTopic):

gcloud pubsub topics publish myTopic --message "Hello"

Copied!

1. Publish a few more messages to myTopic. Run the following commands (replacing <YOUR NAME> with your name and <FOOD> with a food you like to eat):

gcloud pubsub topics publish myTopic --message "Publisher's name is <YOUR NAME>"

Copied!

gcloud pubsub topics publish myTopic --message "Publisher likes to eat <FOOD>"

Copied!

gcloud pubsub topics publish myTopic --message "Publisher thinks Pub/Sub is awesome"

Copied!

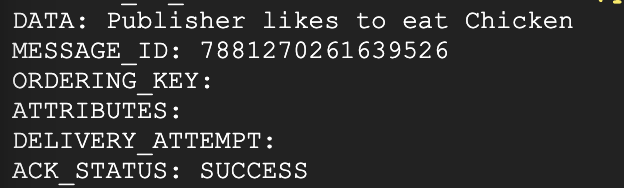
Next, use the pull command to get the messages from your topic. The pull command is subscription based, meaning it should work because earlier you set up the subscription mySubscription to the topic myTopic.

1. Use the following command to pull the messages you just published from the Pub/Sub topic:

gcloud pubsub subscriptions pull mySubscription --auto-ack

Copied!

Your output should resemble the following:



What's going on here? You published 4 messages to your topic, but only 1 was outputted.

Now is an important time to mention a couple features of the pull command that often trip developers up:

* **Using the pull command without any flags will output only one message, even if you are subscribed to a topic that has more held in it.**
* **Once an individual message has been outputted from a particular subscription-based pull command, you cannot access that message again with the pull command.**

1. To see what the second bullet is talking about, run the last command three more times. You will see that it will output the other messages you published before.
2. Now, run the command a 4th time. You'll get the following output (since there were none left to return):

gcpstaging20394\_student@cloudshell:~ (qwiklabs-gcp-3450558d2b043890)$ gcloud pubsub subscriptions pull mySubscription --auto-ack

Listed 0 items.

In the last section, you will learn how to pull multiple messages from a topic with a flag.

**Task 4. Pub/Sub pulling all messages from subscriptions**

Since you pulled all of the messages from your topic in the last example, populate myTopic with a few more messages.

1. Run the following commands:

gcloud pubsub topics publish myTopic --message "Publisher is starting to get the hang of Pub/Sub"

Copied!

gcloud pubsub topics publish myTopic --message "Publisher wonders if all messages will be pulled"

Copied!

gcloud pubsub topics publish myTopic --message "Publisher will have to test to find out"

Copied!

1. Add a flag to your command so you can output all three messages in one request.

You may have not noticed, but you have actually been using a flag this entire time: the --auto-ack part of the pull command is a flag that has been formatting your messages into the neat boxes that you see your pulled messages in.

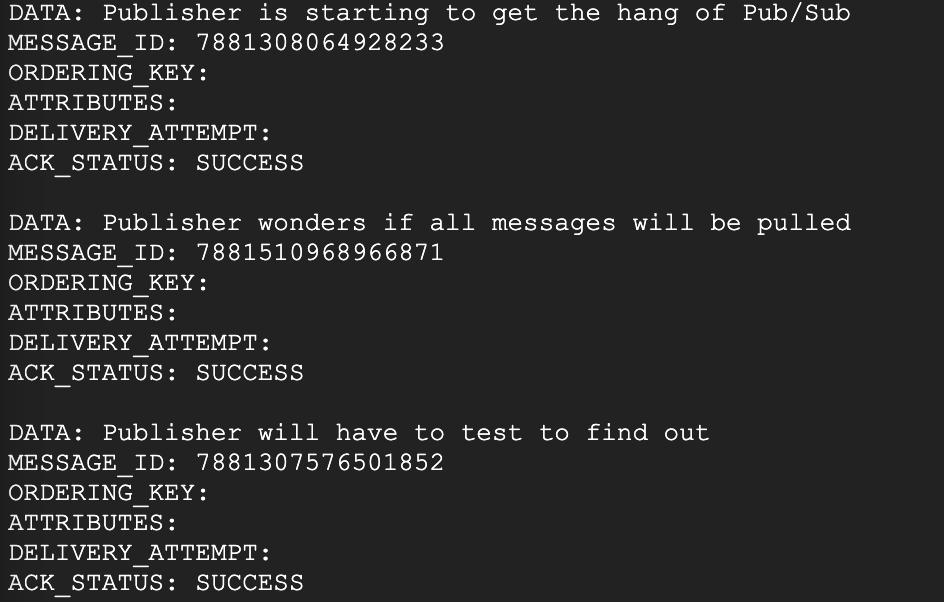
limit is another flag that sets an upper limit on the number of messages to pull.

1. Wait a minute to let the topics get created. Run the pull command with the limit flag:

gcloud pubsub subscriptions pull mySubscription --limit=3

Copied!

Your output should match the following:



Now you know how to add flags to a Pub/Sub command to output a larger pool of messages. You are well on your way to becoming a Pub/Sub master.