

Google Cloud Pub/Sub

Le service Google Cloud Pub/Sub permet aux applications d'échanger des messages de manière fiable, rapide et asynchrone. Pour ce faire, un producteur de données publie des messages dans un sujet (Topic) Cloud Pub/Sub. Un client abonné crée ensuite un abonnement à cette rubrique et consomme les messages de l'abonnement. Cloud Pub/Sub conserve les messages qui n'ont pas pu être envoyés de manière fiable pendant sept jours maximum.

1- Utilisez Python:

Exécutez la commande suivante pour télécharger et mettre à jour la liste des packages.

```
sudo apt-get update
```

Les environnements virtuels Python sont utilisés pour isoler l'installation des packages du système.

```
sudo apt-get install virtualenv
```

```
virtualenv -p python3 venv
```

```
You are running apt-get inside of Cloud Shell. Note that your Cloud Shell
machine is ephemeral and no system-wide change will persist beyond session end.

To suppress this warning, create an empty ~/.cloudshell/no-apt-get-warning file.
The command will automatically proceed in 5 seconds or on any key.

Visit https://cloud.google.com/shell/help for more information.
*****
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  python3-virtualenv
The following NEW packages will be installed:
  python3-virtualenv virtualenv
0 upgraded, 2 newly installed, 0 to remove and 22 not upgraded.
Need to get 78.2 kB of archives.
After this operation, 171 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://deb.debian.org/debian buster/main amd64 python3-virtualenv all 15.1.0+ds-2 [58.1 kB]
Get:2 http://deb.debian.org/debian buster/main amd64 virtualenv all 15.1.0+ds-2 [20.1 kB]
Fetched 78.2 kB in 0s (1,748 kB/s)
debconf: delaying package configuration, since apt-utils is not installed
Selecting previously unselected package python3-virtualenv.
(Reading database ... 127761 files and directories currently installed.)
Preparing to unpack .../python3-virtualenv_15.1.0+ds-2_all.deb ...
Unpacking python3-virtualenv (15.1.0+ds-2) ...
Selecting previously unselected package virtualenv.
Preparing to unpack .../virtualenv_15.1.0+ds-2_all.deb ...
Unpacking virtualenv (15.1.0+ds-2) ...
Setting up python3-virtualenv (15.1.0+ds-2) ...
Setting up virtualenv (15.1.0+ds-2) ...
Processing triggers for man-db (2.8.5-2) ...
ahmedhosni_contact@cloudshell:~$
```

```
ahmedhosni_contact@cloudshell:~$ virtualenv -p python3 venv
created virtual environment CPython3.7.3.final.0-64 in 1124ms
creator CPython3Posix(dest=/home/ahmedhosni_contact/venv, clear=False, global=False)
seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=/home/ahmedhosni_contact/.local/share/virtualenv)
added seed packages: pip==20.2.1, setuptools==49.2.1, wheel==0.34.2
activators BashActivator,CShellActivator,PowerShellActivator,PythonActivator,XonshActivator
ahmedhosni_contact@cloudshell:~$
```

Activez l'environnement virtuel.

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

```
ahmedhosni_contact@cloudshell:~$ source venv/bin/activate
(venv) ahmedhosni_contact@cloudshell:~$
```

2- Installez la bibliothèque cliente

Exécutez ce qui suit pour installer la bibliothèque cliente:

```
pip install --upgrade google-cloud-pubsub
```

```
Collecting chardet<4,>=3.0.2
  Downloading chardet-3.0.4-py2.py3-none-any.whl (133 kB)
Collecting idna<3,>=2.5
  Downloading idna-2.10-py3-none-any.whl (58 kB)
Collecting certifi<2017.4.17
  Downloading certifi-2020.6.20-py2.py3-none-any.whl (156 kB)
Collecting rsa<5,>=3.1.4; python_version >= "3.5"
  Downloading rsa-4.6-py3-none-any.whl (47 kB)
Collecting cachetools<5.0,>=2.0.0
  Downloading cachetools-4.1.1-py3-none-any.whl (10 kB)
Collecting pyasn1-modules<0.2.1
  Downloading pyasn1-modules-0.2.8-py2.py3-none-any.whl (155 kB)
Collecting pyasn1<0.1.3
  Downloading pyasn1-0.4.8-py2.py3-none-any.whl (77 kB)
Building wheels for collected packages: grpc-google-iam-v1
  Building wheel for grpc-google-iam-v1 (setup.py) ... done
  Created wheel for grpc-google-iam-v1: filename=grpc_google_iam_v1-0.12.3-py3-none-any.whl size=18499 sha256=5cf5b1df278baaf38bd49e756fdd56ef9951e5e43620c64a7950f65e77fe879
  Stored in directory: /home/ahmedhosni_contact/.cache/pip/wheels/b9/ee/67/2e444183030cb8d31ce8b34cee34a7afdbd3ba5959ea846380
Successfully built grpc-google-iam-v1
Installing collected packages: six, grpcio, protobuf, googleapis-common-protos, grpc-google-iam-v1, urllib3, chardet, idna, certifi, requests, pytz, pyasn1, rsa, cachetools,
pyasn1-modules, google-auth, google-api-core, google-cloud-pubsub
Successfully installed cachetools-4.1.1 certifi-2020.6.20 chardet-3.0.4 google-api-core-1.22.1 google-auth-1.20.1 google-cloud-pubsub-1.7.0 googleapis-common-protos-1.52.0 g
rpc-google-iam-v1-0.12.3 grpcio-1.31.0 idna-2.10 protobuf-3.15.0 pyasn1-0.4.8 pyasn1-modules-0.2.8 pytz-2020.1 requests-2.24.0 rsa-4.6 six-1.15.0 urllib3-1.25.10
WARNING: You are using pip version 20.2.1; however, version 20.2.2 is available.
You should consider upgrading via the '/home/ahmedhosni_contact/venv/bin/python -m pip install --upgrade pip' command.
(venv) ahmedhosni_contact@cloudshell:~$
```

Obtenez l'exemple de code en clonant un repository GitHub:

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

```
(venv) ahmedhosni_contact@cloudshell:~$ git clone https://github.com/googleapis/python-pubsub.git
Cloning into 'python-pubsub'...
remote: Enumerating objects: 5326, done.
remote: Total 5326 (delta 0), reused 0 (delta 0), pack-reused 5326
Receiving objects: 100% (5326/5326), 1.81 MiB | 3.42 MiB/s, done.
Resolving deltas: 100% (3142/3142), done.
(venv) ahmedhosni_contact@cloudshell:~$ cd python-pubsub/samples/snippets
(venv) ahmedhosni_contact@cloudshell:~/python-pubsub/samples/snippets$
```

3- Pub / Sub - les bases

Google Cloud Pub/Sub est un service de messagerie global asynchrone. Il existe trois termes dans Pub/Sub qui apparaissent souvent: *sujets* , *publication* et *abonnement*.

Un sujet (Topic) est une chaîne partagée qui permet aux applications de se connecter entre elles via un thread commun.

Les éditeurs envoient (ou publient) un message dans un sujet Cloud Pub/Sub. Les abonnés feront ensuite un *abonnement* à ce thread, où ils auront soit à tirer des messages des webhooks de sujet ou bien configurer les paramètres pour les abonnements push. Chaque abonné doit accuser réception de chaque message dans une fenêtre de temps configurable.

En résumé, un éditeur crée et envoie des messages à un sujet et un abonné crée un abonnement à un sujet pour en recevoir des messages.

4- Pub / Sub dans Google CCloud

Pub/Sub est préinstallé dans Cloud Shell. Aucune installation ou configuration n'est donc requise pour démarrer avec ce service. Dans cet lab, vous utilisez Python pour créer la topic, l'abonné, puis afficher le message. Vous utilisez une commande gcloud pour publier le message dans la rubrique.

5- Créer un topic

Pour publier des données dans Cloud Pub / Sub, créez un sujet (topic), puis configurez un éditeur sur le sujet.

Définissez la variable d'environnement GOOGLE_CLOUD_PROJECT. Vous pouvez trouver votre ID de projet dans les DÉTAILS DE CONNEXION:

```
export GLOBAL_CLOUD_PROJECT=GCP Project ID
```

`Publisher.py` est un script qui montre comment effectuer des opérations de base sur des sujets avec l'API Cloud Pub/Sub. Afficher le contenu du script de l'éditeur:
`cat publisher.py`

Pour plus d'informations sur le script de l'éditeur:

`python publisher.py -h`

```
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_id            Your Google Cloud project ID
  {list,create,delete,publish,publish-with-custom-attributes,publish-with-error-handler,publish-with-batch-settings,publish-with-retry-settings,publish-with-ordering-keys,
resume-publish-with-ordering-keys}

  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-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.
  publish-with-retry-settings
                        Publishes messages with custom retry settings.
  publish-with-ordering-keys
                        Publishes messages with ordering keys.
  resume-publish-with-ordering-keys
                        Resume publishing messages with ordering keys when
                        unrecoverable errors occur.

optional arguments:
  -h, --help            show this help message and exit
(venv) ahmedhosni_contact@cloudshell:~/python-pubsub/samples/snippets$
```

Exécutez le script de l'éditeur pour créer un sujet Pub/Sub:

`python publisher.py $GLOBAL_CLOUD_PROJECT create MyTopic`

```
(venv) ahmedhosni_contact@cloudshell:~/python-pubsub/samples/snippets$ python publisher.py $GLOBAL_CLOUD_PROJECT create MyTopic
Topic created: name: "projects/widigital-ci/topics/MyTopic"
```

Cette commande renvoie une liste de tous les sujets Pub/Sub dans un projet donné:

`python publisher.py $GLOBAL_CLOUD_PROJECT list`

```
(venv) ahmedhosni_contact@cloudshell:~/python-pubsub/samples/snippets$ python publisher.py $GLOBAL_CLOUD_PROJECT list
name: "projects/widigital-ci/topics/container-analysis-occurrences-v1beta1"

name: "projects/widigital-ci/topics/container-analysis-occurrences-v1"
name: "projects/widigital-ci/topics/container-analysis-notes-v1beta1"
name: "projects/widigital-ci/topics/container-analysis-notes-v1"
name: "projects/widigital-ci/topics/MyTopic"
```

Vous pouvez également afficher le sujet que vous venez de créer dans Cloud Console. Accédez au **menu de navigation > Pub / Sub > Topics** .

Google Cloud Platform widigital-ci Search products and resources

Home > Topics + CREATE TOPIC DELETE

Filter table

Topic ID ↑	Encryption	Topic name		
container-analysis-notes-v1	Google-managed	projects/widigital-...	⋮	▼
container-analysis-notes-v1beta1	Google-managed	projects/widigital-...	⋮	▼
container-analysis-occurrences-v1	Google-managed	projects/widigital-...	⋮	▼
container-analysis-occurrences-v1beta1	Google-managed	projects/widigital-...	⋮	▼
MyTopic	Google-managed	projects/widigital-...	⋮	▼

Open

```
~/python-pubsub/samples/snippets$ python publisher.py $GLOBAL_CLOUD_PROJECT list
container-analysis-occurrences-v1beta1"
container-analysis-occurrences-v1"
container-analysis-notes-v1beta1"
```

6- Créer un 'subscription'

Créez un 'subscription' Pub/Sub pour le 'topic' avec le script `subscriber.py`:

```
python subscriber.py $GLOBAL_CLOUD_PROJECT create MyTopic MySub
```

```
(venv) ahmedhosni contact@cloudshell:~/python-pubsub/samples/snippets$ python subscriber.py $GLOBAL_CLOUD_PROJECT create MyTopic MySub
Subscription created: name: "projects/widigital-ci/subscriptions/MySub"
topic: "projects/widigital-ci/topics/MyTopic"
push_config {
}
ack_deadline_seconds: 10
message_retention_duration {
  seconds: 604800
}
expiration_policy {
  ttl {
    seconds: 2678400
  }
}
```

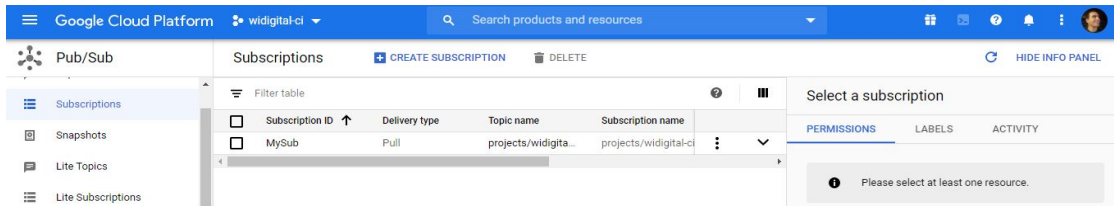
Cette commande renvoie une liste d'abonnés (subscriptions) dans un projet donné.

Vous ne verrez qu'un seul abonnement, car vous n'avez fait qu'un seul abonnement.

```
python subscriber.py $GLOBAL_CLOUD_PROJECT list-in-project
```

```
(venv) ahmedhosni contact@cloudshell:~/python-pubsub/samples/snippets$ python subscriber.py $GLOBAL_CLOUD_PROJECT list-in-project
projects/widigital-ci/subscriptions/MySub
```

Découvrez l'abonnement que vous venez de faire dans la console. Dans le volet gauche, cliquez sur **Subscriptions**. Vous devriez voir le nom du Subscription et d'autres détails.



Pour plus d'informations sur le script subscriber:

```
python subscriber.py -h
```

7- Publier des messages

Maintenant que vous avez configuré le topic MyTopic, une subscription à MyTopic (MySub), voyez si vous pouvez utiliser les commandes gcloud pour publier un message MyTopic. Publiez le message "Bonjour" sur MyTopic:

```
gcloud pubsub topics publish MyTopic --message "Hello"
```

```
(venv) ahmedhosni_contact@cloudshell:~/python-pubsub/samples/snippets$ gcloud pubsub topics publish MyTopic --message "Hello" --project widigital-ci
messageIds:
- '1443180030403079'
```

Publiez quelques messages supplémentaires sur MyTopic, exécutez les commandes suivantes (en remplaçant <VOTRE NOM> par votre nom et <FOOD> par un aliment que vous aimez manger):

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

```
(venv) ahmedhosni_contact@cloudshell:~/python-pubsub/samples/snippets$ gcloud pubsub topics publish MyTopic --message "Publisher's name is Ahmed" --project widigital-ci
messageIds:
- '1443192552831192'
```

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

```
(venv) ahmedhosni_contact@cloudshell:~/python-pubsub/samples/snippets$ gcloud pubsub topics publish MyTopic --message "Publisher likes to eat Pasta" --project widigital-ci
messageIds:
- '1443206933492417'
```

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

```
(venv) ahmedhosni_contact@cloudshell:~/python-pubsub/samples/snippets$ gcloud pubsub topics publish MyTopic --message "Publisher thinks Pub/Sub is awesome" --project widigital-ci
messageIds:
- '1443207368543844'
```


6- Afficher les messages

Maintenant que vous avez publié des messages sur MyTopic, extrayez et affichez les messages à l'aide de MySub. Utilisez MySub pour extraire le message de MyTopic:

```
python subscriber.py $GLOBAL_CLOUD_PROJECT receive MySub
```

```
(venv) ahmedhosni_contact@cloudshell:~/python-pubsub/samples/snippets$ python subscriber.py $GLOBAL_CLOUD_PROJECT receive MySub
Listening for messages on projects/widigital-ci/subscriptions/MySub..

Received message: Message {
  data: b'Hello'
  ordering_key: ''
  attributes: {}
}
Received message: Message {
  data: b'Publisher's name is Ahmed'
  ordering_key: ''
  attributes: {}
}
Received message: Message {
  data: b'Publisher likes to eat Pasta'
  ordering_key: ''
  attributes: {}
}
Received message: Message {
  data: b'Publisher thinks Pub/Sub is awesome'
  ordering_key: ''
  attributes: {}
}
}
```

Cliquez sur **Ctrl + c** pour arrêter l'écoute.

8- Supprimer les ressources créées:

