Overview

Service accounts are a special type of Google account that grant permissions to virtual machines instead of end users. Service accounts are primarily used to ensure safe, managed connections to APIs and Google Cloud services. Granting access to trusted connections and rejecting malicious ones is a must-have security feature for any Google Cloud project. In this lab, you will get hands-on practice with the ins and outs of service accounts.

What are Service Accounts?

A service account is a special Google account that belongs to your application or a virtual machine (VM) instead of an individual end user. Your application uses the service account to call the Google API of a service, so that the users aren't directly involved.

For example, a Compute Engine VM may run as a service account, and that account can be given permissions to access the resources it needs. This way the service account is the identity of the service, and the service account's permissions control which resources the service can access.

A service account is identified by its email address, which is unique to the account.

Types of Service Accounts

User-managed service accounts

When you create a new Cloud project using Cloud Console and if Compute Engine API is enabled for your project, a Compute Engine Service account is created for you by default. It is identifiable using the email:

```
PROJECT_NUMBER-compute@developer.gserviceaccount.com
```
content_copy

If your project contains an App Engine application, the default App Engine service account is created in your project by default. It is identifiable using the email:

```
PROJECT_ID@appspot.gserviceaccount.com
```
content_copy

Google-managed service accounts

In addition to the user-managed service accounts, you might see some additional service accounts in your project’s IAM policy or in the Cloud Console. These service accounts are created and owned by Google. These accounts represent different Google services and each account is automatically granted IAM roles to access your Google Cloud project.

Google APIs service account

An example of a Google-managed service account is a Google API service account identifiable using the email:

```
PROJECT_NUMBER@cloudservices.gserviceaccount.com
```
content_copy

This service account is designed specifically to run internal Google processes on your behalf and is not listed in the **Service Accounts** section of Cloud Console. By default, the account is automatically granted the project editor role on the project and is listed in the **IAM** section of Cloud Console. This service account is deleted only when the project is deleted. Google services rely on the account having access to your project, so you should not remove or change the service account’s role on your project.

Creating and Managing Service Accounts

When you create a new Cloud project, Google Cloud automatically creates one Compute Engine service account and one App Engine service account under that project. You can create up to 98 additional service accounts to your project to control access to your resources.

Creating a service account

Creating a service account is similar to adding a member to your project, but the service account belongs to your applications rather than an individual end user.

To create a service account, run the following command in Cloud Shell:

```
gcloud iam service-accounts create my-sa-123 --display-name "my service account"
```

```
content_copy
```

The output of this command is the service account, which will look similar to the following:

```
Created service account [my-sa-123]
content_copy
```

Granting Roles to Service Accounts

When granting IAM roles, you can treat a service account either as a resource or as an identity.

Your application uses a service account as an identity to authenticate to Google Cloud services. For example, if you have a Compute Engine Virtual Machine (VM) running as a service account, you can grant the editor role to the service account (the identity) for a project (the resource).

At the same time, you might also want to control who can start the VM. You can do this by granting a user (the identity) the serviceAccountUser role for the service account (the resource).

Granting roles to a service account for specific resources
You grant roles to a service account so that the service account has permission to complete specific actions on the resources in your Cloud Platform project. For example, you might grant the storage.admin role to a service account so that it has control over objects and buckets in Cloud Storage.

Run the following in Cloud Shell to grant roles to the service account you just made:

```
gcloud projects add-iam-policy-binding $DEVSHELL_PROJECT_ID \
    --member serviceAccount:my-sa-123@$DEVSHELL_PROJECT_ID.iam.gserviceaccount.com --
role roles/editor
content_copy
```

The output will display a list of roles the service account now has:

```
bindings:
```

```
- members:
    - user:email1@gmail.com
        role: roles/owner
- members:
    - serviceAccount:our-project-123@appspot.gserviceaccount.com
    - serviceAccount:123456789012-compute@developer.gserviceaccount.com
    - serviceAccount:my-sa-123@my-project-123.iam.gserviceaccount.com
    - user:email3@gmail.com
        role: roles/editor
- members:
    - user:email2@gmail.com
role: roles/viewer
etag: BwUm38GGAQk=
version: 1
content_copy
```

Click Check my progress to verify the objective.

Understanding Roles

When an identity calls a Google Cloud API, Google Cloud Identity and Access Management requires that the identity has the appropriate permissions to use the resource. You can grant permissions by granting roles to a user, a group, or a service account.

Types of Roles

There are three types of roles in Cloud IAM:

•**Primitive roles**, which include the Owner, Editor, and Viewer roles that existed prior to the introduction of Cloud IAM.
•**Predefined roles**, which provide granular access for a specific service and are managed by Google Cloud.
•**Custom roles**, which provide granular access according to a user-specified list of permissions.
For more details, please visit IAM Roles.

Use the Client Libraries to Access BigQuery from a Service Account

In this section, you will query the BigQuery public datasets from an instance with the help of a service account which has the necessary roles.

Create a service account

You will first create a new service account from the Cloud Console.

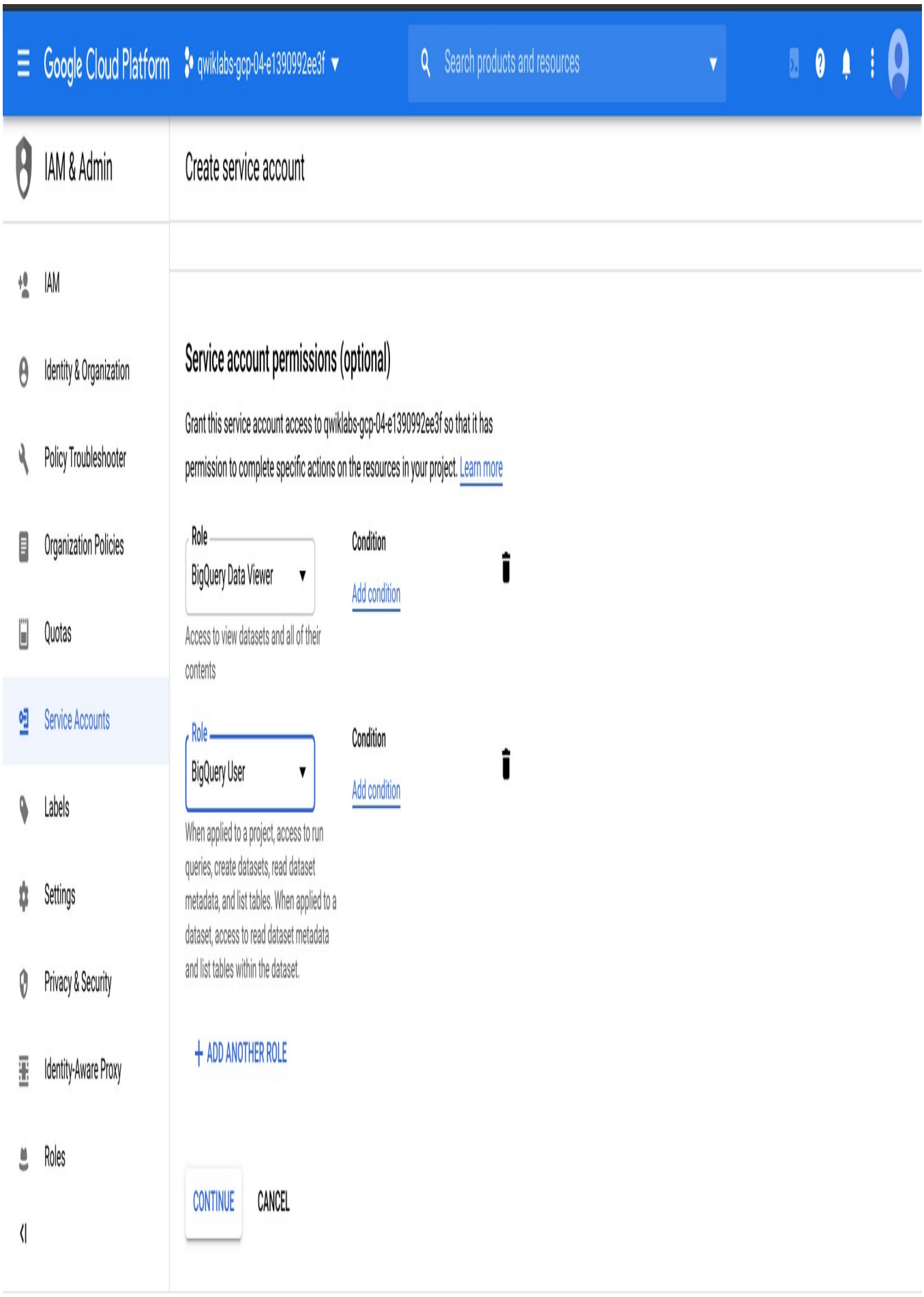
Go to **Navigation menu** > **IAM & Admin**, select **Service accounts** and click on **+ Create Service Account**.

Fill necessary details with:

•**Service account name:** bigquery-qwiklab
Now click **Create and Continue** and then add the following roles.

•**Role:** BigQuery Data Viewer and BigQuery User

Your console should resemble the following:


Google Cloud Platform
qwiklabs-gcp-04-e1390992ee3f
Search products and resources
IAM & Admin
Create service account
IAM
Identity & Organization
Policy Troubleshooter
Organization Policies
Quotas
Service Accounts
Labels
Settings
Privacy & Security
Identity-Aware Proxy
Roles
Service account permissions (optional)
Grant this service account access to qwiklabs-gcp-04-e1390992ee3f so that it has permission to complete specific actions on the resources in your project. Learn more
Role
BigQuery Data Viewer
Access to view datasets and all of their contents
Condition
Add condition
Role
BigQuery User
When applied to a project, access to run queries, create datasets, read dataset metadata, and list tables. When applied to a dataset, access to read dataset metadata and list tables within the dataset.
Condition
Add condition
+ ADD ANOTHER ROLE
CONTINUE
CANCEL

Click **Continue** and then click **Done**.

Create a VM instance

In the Cloud Console, go to **Compute Engine** > **VM Instances**, and click **Create Instance**.

Create your VM with the following information:

Configuration	Value
Name	bigquery-instance
Region	us-central1(Iowa)
Zone	us-central1-a
Series	N1
Machine Type	1 vCPU (n1-standard-1)
Boot Disk	Debian GNU/Linux 9 (stretch)
Service account	bigquery-qwiklab

Note: If the bigquery-qwiklab service account doesn't appear in the drop-down list, try typing the name in to the "Filter" section.

Click **Create**.

Put the example code on a Compute Engine instance

In the Cloud Console, go to **Compute Engine** > **VM Instances**. SSH into bigquery-instance by clicking on the **SSH** button.

Connect without Identity-Aware Proxy.

Execute the following command to download and update the packages list.

```
sudo apt-get update
```
content_copy

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

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

Y and then Enter.

```
virtualenv -p python3 venv
```
content_copy

Activate the virtual environment.

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

In the SSH window, install the necessary dependencies by running the following commands:

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

```
pip install google-cloud-bigquery
```
content_copy

```
pip install pandas
```
content_copy

Now create the example Python file:

```
echo "
from google.auth import compute_engine
from google.cloud import bigquery
credentials = compute_engine.Credentials(
       service_account_email='YOUR_SERVICE_ACCOUNT')
query = '''
SELECT
   year,
   COUNT(1) as num_babies
FROM
   publicdata.samples.natality
WHERE
   year > 2000
GROUP BY
   year
'''
```

```
client = bigquery.Client(
        project='YOUR_PROJECT_ID',
        credentials=credentials)
print(client.query(query).to_dataframe())
" > query.py
content_copy
```

Add the Project ID to query.py with:

```
sed -i -e "s/YOUR_PROJECT_ID/$(gcloud config get-value project)/g" query.py
content_copy
```

Run the following to make sure that the sed command has successfully changed the Project ID in the file:

```
cat query.py
content_copy
```

Example Output (yours will differ):

```
from google.auth import compute_engine
from google.cloud import bigquery
credentials = compute_engine.Credentials(
        service_account_email='YOUR_SERVICE_ACCOUNT')
query = '''
SELECT
   year,
   COUNT(1) as num_babies
FROM
   publicdata.samples.natality
WHERE
   year > 2000
GROUP BY
   year
'''
client = bigquery.Client(
        project='qwiklabs-gcp-186de687ef87f911',
        credentials=credentials)
print(client.query(query).to_dataframe())
content_copy
```

Add the service account email to query.py with:

```
sed -i -e "s/YOUR_SERVICE_ACCOUNT/bigquery-qwiklab@$(gcloud config get-value
project).iam.gserviceaccount.com/g" query.py
content_copy
```

Run the following to make sure that the sed command has successfully changed the service account email in the file:

```
cat query.py
content_copy
```

Example Output (yours will differ):

```
from google.auth import compute_engine
from google.cloud import bigquery
credentials = compute_engine.Credentials(
       service_account_email='bigquery-qwiklab@qwiklabs-gcp-
186de687ef87f911.iam.gserviceaccount.com')
query = '''
SELECT
   year,
   COUNT(1) as num_babies
FROM
   publicdata.samples.natality
WHERE
   year > 2000
GROUP BY
   year
'''
client = bigquery.Client(
       project='qwiklabs-gcp-186de687ef87f911',
       credentials=credentials)
print(client.query(query).to_dataframe())
content_copy
```

The application will now use the permissions that are associated with this service account. Now run the query with the following Python command:

```
python query.py
content_copy
```

You should be returned with a similar output:

```
Row year  num_babies
0  2008  4255156
1  2006  4273225
2  2003  4096092
```

```
3  2004 4118907
4  2002 4027376
5  2005 4145619
6  2001 4031531
7  2007 4324008
content_copy
```

Note: Your row values might not map to the years in the above output. However, make sure that the babies per year are the same.

Awesome work! You made a request to a BigQuery public dataset with a bigquery-qwiklab service account.

Click Check my progress to verify the objective.

Congratulations!

Finish Your Quest

This self-paced lab is part of the Qwiklabs Security & Identity Fundamentals Quest. 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.

Take Your Next Lab

Continue your Quest with Securing Google Cloud with CFT Scorecard, or check out these suggestions:

•Cloud Security Scanner: Qwik Start

•Data Loss Prevention: Qwik Start - Command Line

Next Steps / Learn More

Read more about creating and managing service accounts here.

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 September 03, 2021

Lab Last Tested September 03, 2021