Configuring, Using, and Auditing VM Service Accounts and Scopes

1 hour 30 minutes   No cost

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.

Objectives

In this lab, you will learn how to:

- Create and manage service accounts.
- Create a virtual machine and associate it with a service account.
- Use client libraries to access BigQuery from a service account.
- Run a query on a BigQuery public dataset from a Compute Engine instance.

Setup and requirements

For each lab, you get a new Google Cloud project and set of resources for a fixed time at no cost.

1. Sign in to Qwiklabs using an **incognito window**.

2. Note the lab's access time (for example, `1:15:00`), and make sure you can finish within that time.
   There is no pause feature. You can restart if needed, but you have to start at the beginning.

3. When ready, click **Start lab**.

4. Note your lab credentials (**Username** and **Password**). You will use them to sign in to the Google Cloud Console.

5. Click **Open Google Console**.

6. Click **Use another account** and copy/paste credentials for **this** lab into the prompts.
   If you use other credentials, you'll receive errors or **incur charges**.

7. Accept the terms and skip the recovery resource page.

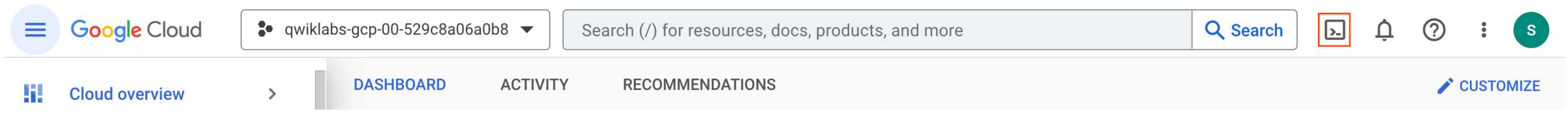
Note: Do not click **End Lab** unless you have finished the lab or want to restart it. This clears your work and removes the project.

Activate Google Cloud Shell

Google 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.

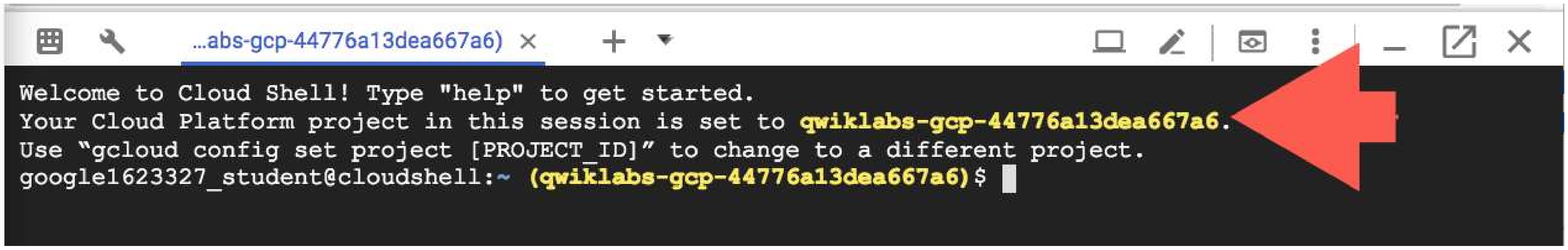
Google Cloud Shell provides command-line access to your Google Cloud resources.

1. In Cloud console, on the top right toolbar, click the Open Cloud Shell button.


Google Cloud
qwiklabs-gcp-00-529c8a06a0b8
Search (/) for resources, docs, products, and more
Search
S
Cloud overview
DASHBOARD
ACTIVITY
RECOMMENDATIONS
CUSTOMIZE

2. Click **Continue**.

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
```

content_c

Output:

```
Credentialed accounts:
 - @.com (active)
```

Example output:

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

- You can list the project ID with this command:

content_c

```
gcloud config list project
```

Output:

```
[core]
project =
```

Example output:

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

Note: Full documentation of **gcloud** is available in the gcloud CLI overview guide .

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 the Cloud Console, if the 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`.

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`.

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 API service accounts

An example of a Google-managed service account is a Google API service account identifiable using the email: `PROJECT_NUMBER@cloudservices.gserviceaccount.com`.

This service account is designed specifically to run internal Google processes on your behalf and is not listed in the **Service Accounts** section of the Cloud Console. By default, the account is automatically granted the project editor role on the project and is listed in the **IAM** section of the 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.

Task 1. Create and manage service accounts

In this task, you create and manage service accounts.

By default, you can create up to 100 user-managed service accounts in a project. If this quota does not meet your needs, you can use the Cloud console to request a quota increase.

Note: The default service accounts described on this page do not count toward this quota.

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_c

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

Output:

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

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 Google Cloud 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_c

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

Output:

```
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
```

Click *Check my progress* to verify the objective.

Creating and Managing Service Account

Check my progress

Understanding roles

When an identity calls a Google Cloud API, Cloud Identity and Access Management (Cloud IAM) 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:

- **Basic 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 refer to the IAM Roles reference documentation.

Task 2. Use the client libraries to access BigQuery from a service account

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

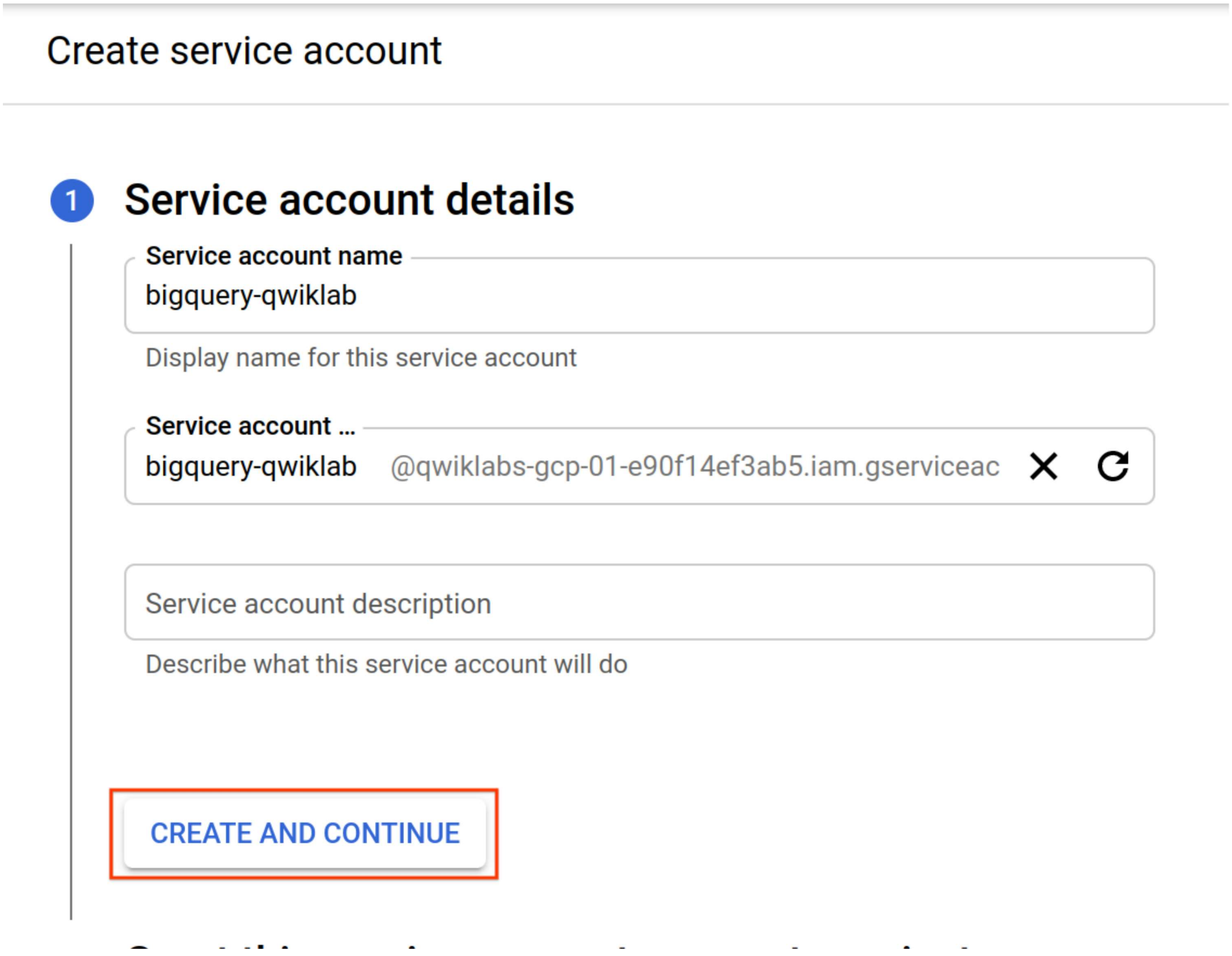
Create a service account

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

1. In the Cloud Console, from the navigation menu, click **IAM & Admin > Service accounts**.

2. Click **Create service account**.

3. Specify the Service account name as `bigquery-qwiklab`.



Create service account

Service account details

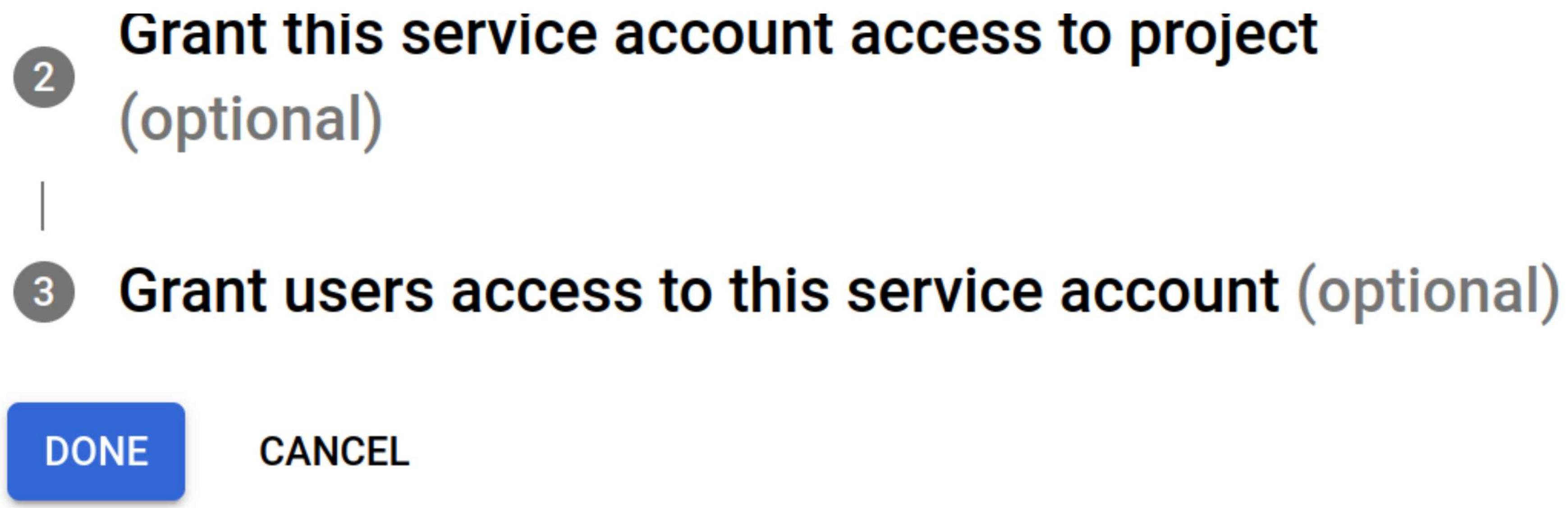
Service account name: bigquery-qwiklab

Display name for this service account

Service account ...: bigquery-qwiklab @qwiklabs-gcp-01-e90f14ef3ab5.iam.gserviceac

Service account description

Describe what this service account will do

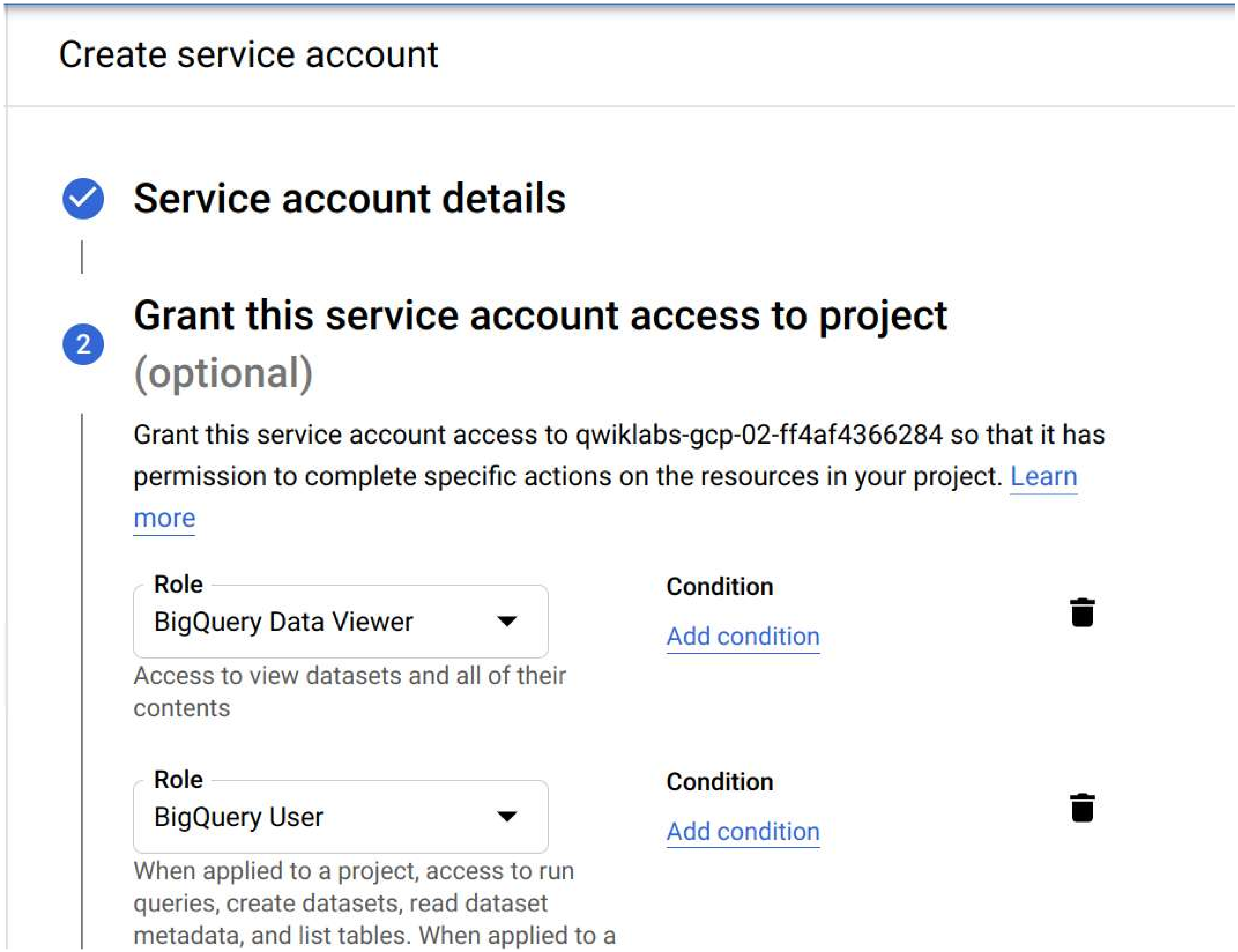

2
Grant this service account access to project (optional)
3
Grant users access to this service account (optional)
DONE
CANCEL

4. Click **Create and Continue**.

5. Specify the Role as **BigQuery > BigQuery Data Viewer**.

6. Click **Add Another Role**.

7. Specify the other Role as **BigQuery > BigQuery User**.



Create service account

Service account details

Grant this service account access to project (optional)

Grant this service account access to qwiklabs-gcp-02-ff4af4366284 so that it has permission to complete specific actions on the resources in your project. Learn more

Role: BigQuery Data Viewer

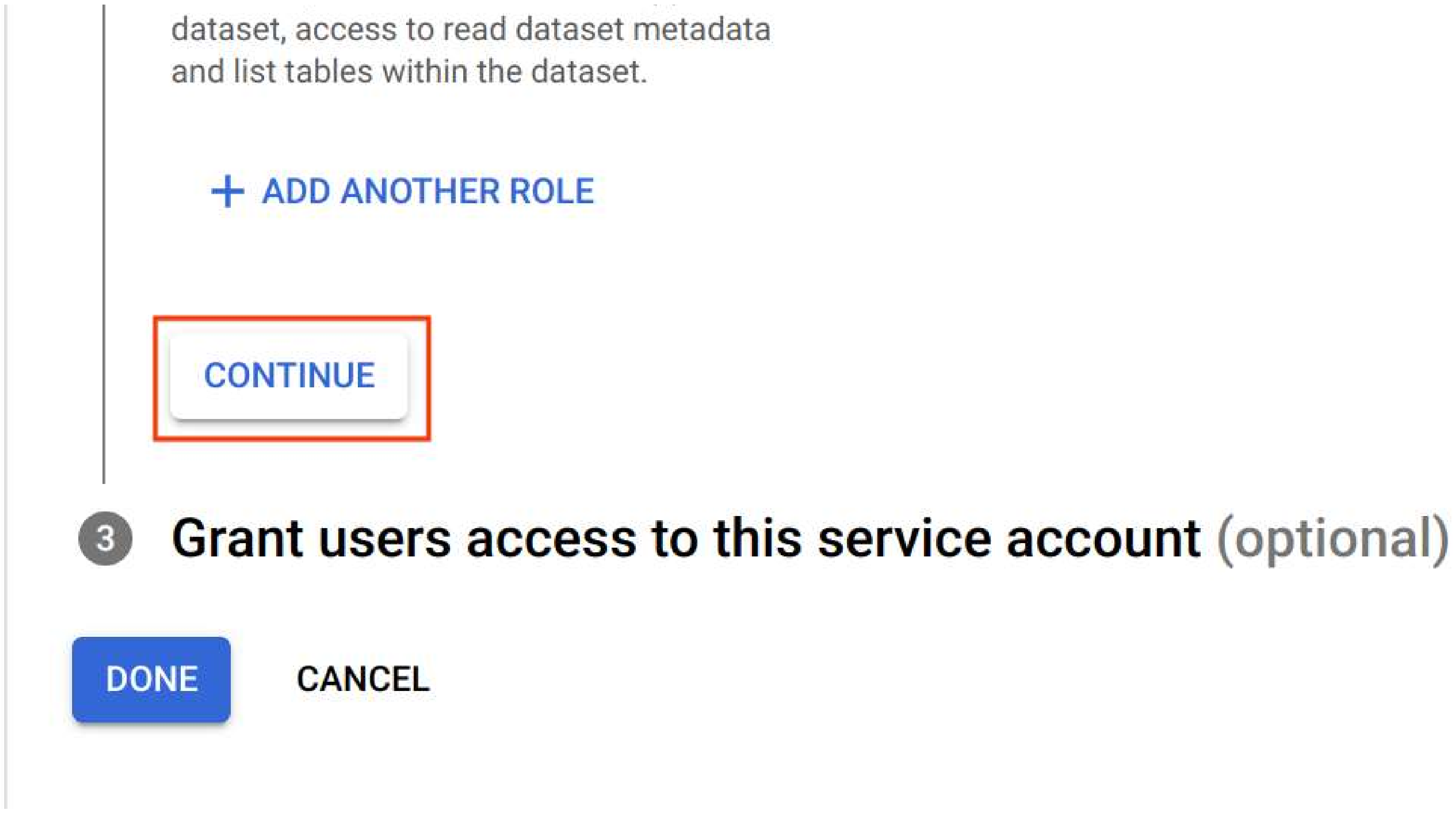
Access to view datasets and all of their contents

Condition: Add condition

Role: BigQuery User

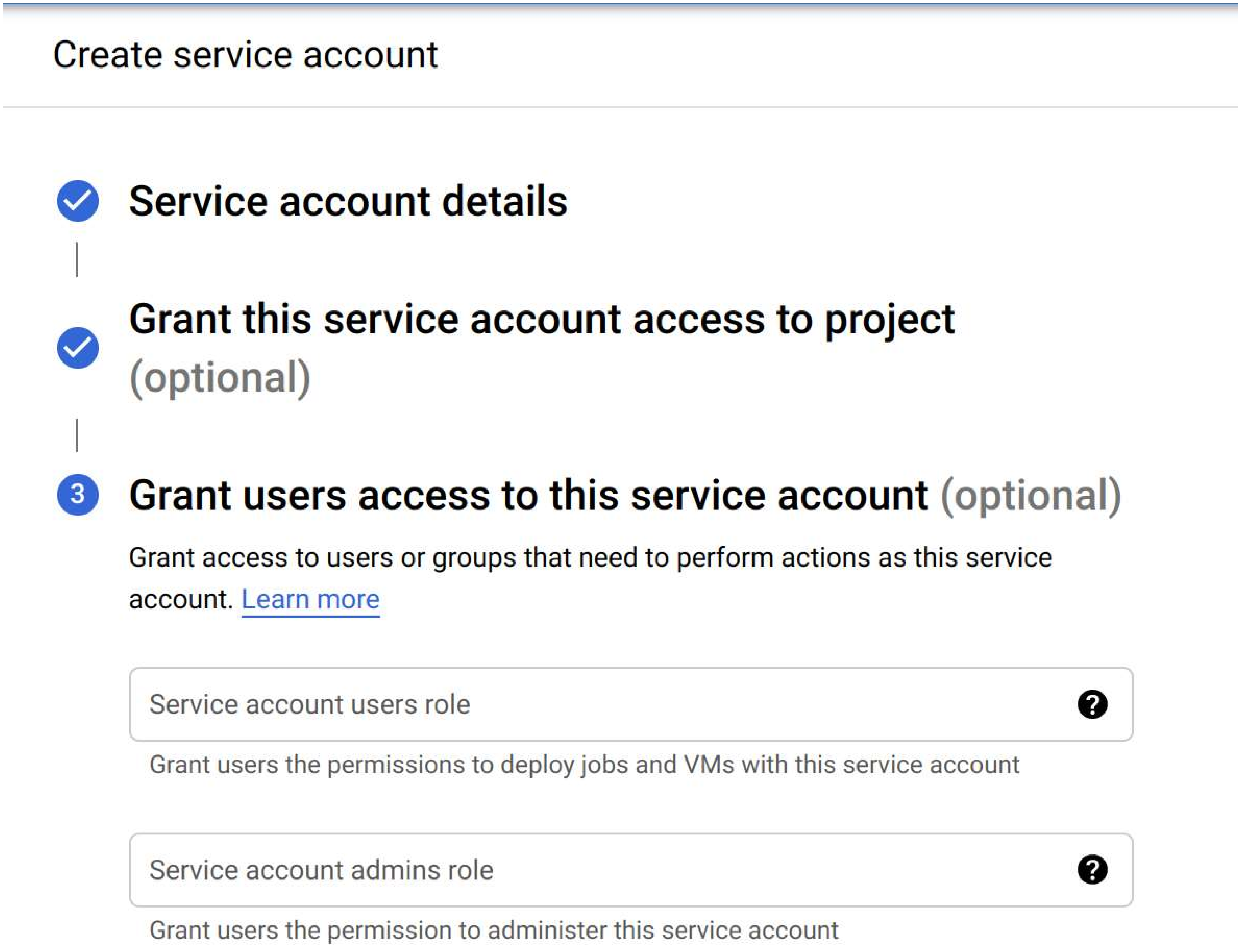
When applied to a project, access to run queries, create datasets, read dataset metadata, and list tables. When applied to a

Condition: Add condition


dataset, access to read dataset metadata and list tables within the dataset.

8. Click **Continue**.

Your Cloud Console should resemble the following.



Create service account

✓ Service account details

✓ Grant this service account access to project (optional)

3 Grant users access to this service account (optional)

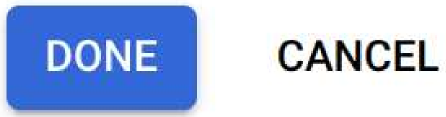
Grant access to users or groups that need to perform actions as this service account. Learn more

Service account users role

Grant users the permissions to deploy jobs and VMs with this service account

Service account admins role

Grant users the permission to administer this service account



9. Click **Done**.

Create a VM instance

1. In the Cloud Console, from the navigation menu, click **Compute Engine > VM Instances**, and click **Create instance**.

2. Create your VM with the following information:

Configuration	**Value**
Name	`bigquery-instance`
Region	`us-central1(Iowa)`
Zone	`us-central1-a`
Series	`E2`
Machine Type	`e2-standard-2 (2 vCPU)`
Boot Disk	`Debian GNU/Linux 10 (buster)`

Service Account	`bigquery-qwiklab`

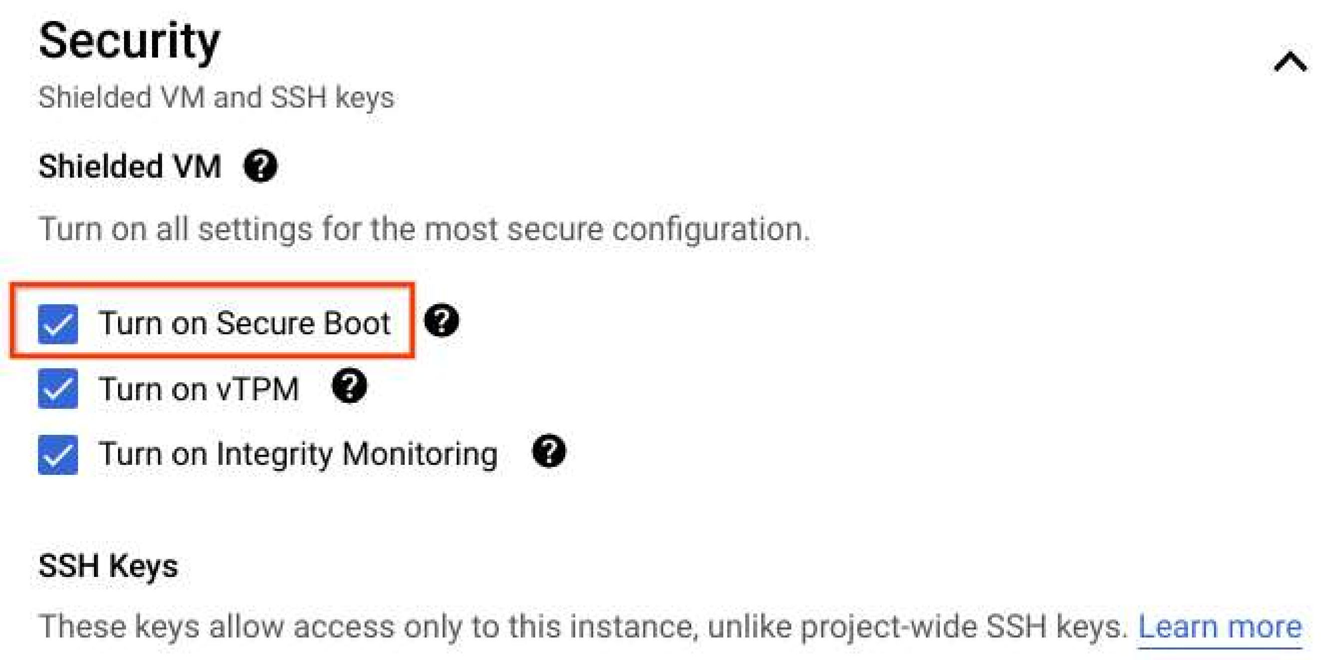
Note: If the `bigquery-qwiklab` service account doesn't appear in the drop-down list, try typing the name in to the *Filter* section.

Next, you will look at the options that were added when you selected a boot disk that supports shielded VMs.

3. Toward the bottom of the page, click the **Networking, disks, security, management, sole tenancy** link.

4. Click the **Security** tab. At the top of the tab, the Shielded VM options appear. If you had not selected a boot disk with shielded VM features, these options would not have appeared. Some of these options are selected by default.

5. Click the **Turn on Secure Boot** checkbox.



Security

Shielded VM and SSH keys

Shielded VM

Turn on all settings for the most secure configuration.

✓ Turn on Secure Boot

✓ Turn on vTPM

✓ Turn on Integrity Monitoring

SSH Keys

These keys allow access only to this instance, unlike project-wide SSH keys. Learn more

Compute Engine does not enable this option by default, because unsigned drivers and other low-level software might not be compatible. Even so, if possible, Google recommends enabling Secure Boot.

6. Click **Create**.

Put the example code on a Compute Engine instance

1. In the Cloud Console, go to **Compute Engine > VM Instances**.
2. SSH into `bigquery-instance` by clicking on the **SSH** button. Click **Connect**.
3. In the SSH window, install the necessary dependencies by running the following commands:

```
sudo apt-get update -y
```

content_c

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

content_c

```
sudo pip3 install six==1.13.0
```

content_c

```
sudo pip3 install --upgrade pip
```

content_c

content_c

```
sudo pip3 install --upgrade google-cloud-bigquery
```

```
sudo pip3 install pandas
```

content_c

4. Using the `echo` command, create `query.py`. You will use this file, written in Python, to run the query.

```
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_c

5. Add the Project ID to `query.py` with:

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

content_c

```
query.py
```

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

```
cat query.py
```

content_c

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())
```

7. Add the service account email to `query.py` with:

```
sed -i -e "s/YOUR_SERVICE_ACCOUNT/bigquery-qwiklab@$(gcloud config
```

content_c

```
get-value project).iam.gserviceaccount.com/g" query.py
```

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

```
cat query.py
```

content_c

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())
```

The application will now use the permissions that are associated with this service account.

9. Install the `pyarrow` library using this command:

```
sudo pip3 install pyarrow==5.0.0
```

content_c

10. Install the `db-dtypes` library using this command:

```
sudo pip3 install db-dtypes
```

content_c

11. Run the `query.py` using this command:

```
python3 query.py
```

content_c

You should be returned with a similar output as the following.

Output:

```
   year  num_babies
0  2008     4255156
1  2002     4027376
2  2005     4145619
3  2006     4273225
4  2001     4031531
5  2007     4324008
6  2003     4096092
7  2004     4118907
```

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.

You made a request to a BigQuery public dataset with a `bigquery-qwiklab` service account.

Click *Check my progress* to verify the objective.

Use the Client Libraries to access BigQuery from a service account

Check my progress

Congratulations!

In this lab, you have learned how to do the following:

- Create and manage service accounts.
- Create a virtual machine and associate it with a service account.
- Use client libraries to access BigQuery from a service account.

- Run a query on a BigQuery public dataset from a Compute Engine instance.

End your lab