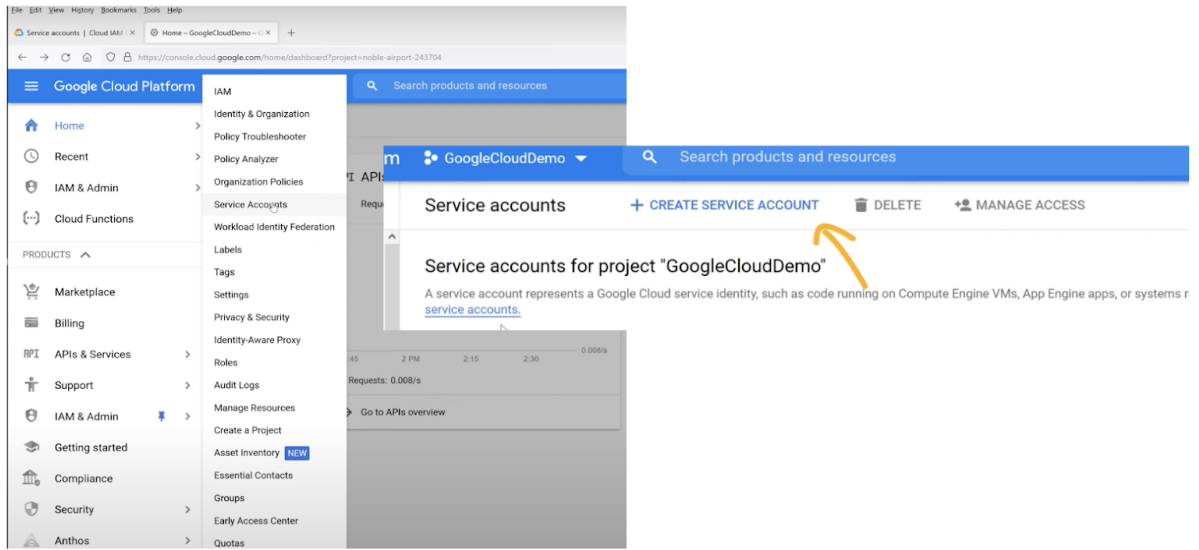


BigQuery Setup Instructions

1. Make sure you have a Gmail account set up .
2. Go to console.cloud.google.com. Click on the navigation (hamburger icon on upper left), then click on IAM & Admin. Under IAM & Admin click on services accounts and create a service account:




3. Next add name and description. Click create and continue:

1 Service account details

Service account name

Display name for this service account

Service account ID * ✕ ↺

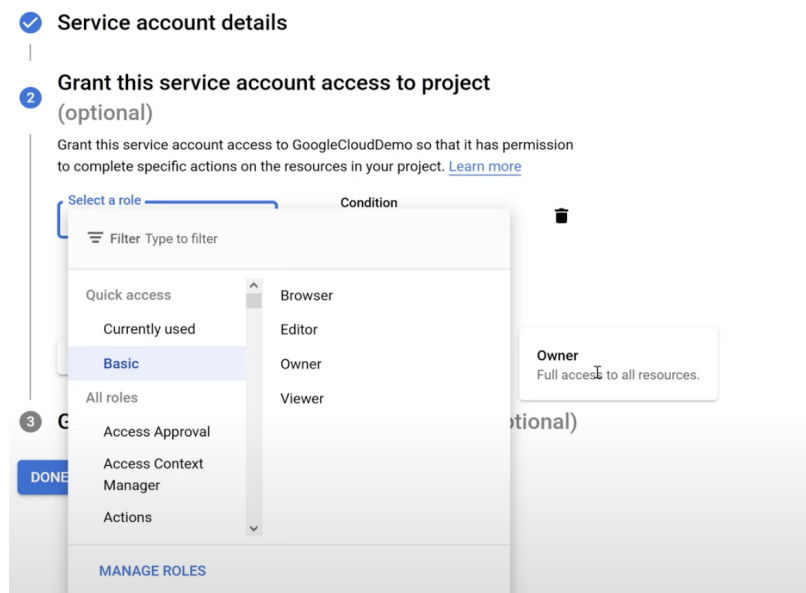
Email address: bigquery-tutorial-sp-admin@boreal-freedom-265422.iam.gserviceaccount.com 

Service account description

Describe what this service account will do

[CREATE AND CONTINUE](#)

1. Next click on basic, then owner, then continue and then done:



Next click on manage keys for your newly created service account:

Service accounts for project "My First Project"

A service account represents a Google Cloud service identity, such as code running on Compute Engine VMs, App Engine apps, or systems running outside Google. [Learn more about service accounts.](#)

Organization policies can be used to secure service accounts and block risky service account features, such as automatic IAM Grants, key creation/upload, or the creation of service accounts entirely. [Learn more about service account organization policies.](#)

Filter

Enter property name or value

<input type="checkbox"/>	Email	Status	Name ↑	Description
<input type="checkbox"/>	bigquery-tutorial@boreal-freedom-265422.iam.gserviceaccount.com	✓	bigquery_tutorial	tutorial ore
<input type="checkbox"/>	bigquery-tutorial-sp-admin@boreal-freedom-265422.iam.gserviceaccount.com	✓	bigquery_tutorial_sp_admin	big tut

- Manage details
- Manage permissions
- Manage keys
- View metrics
- View logs
- Disable
- Delete

Click on add and create keys:

← bigquery_tutorial_sp_admin

DETAILS PERMISSIONS KEYS METRICS LOGS

Keys



Service account keys could pose a security risk if compromised. We recommend you avoid downloading [Identity Federation](#). You can learn more about the best way to authenticate service accounts on Google

Add a new key pair or upload a public key certificate from an existing key pair.

Block service account key creation using [organization policies](#).
[Learn more about setting organization policies for service accounts](#)

ADD KEY ▾

Create new key

Upload existing key

Key creation date

Key expiration date

Select JSON and click create:

Create private key for "bigquery_tutorial_sp_admin"

Downloads a file that contains the private key. Store the file securely because this key can't be recovered if lost.

Key type

☒ JSON

Recommended

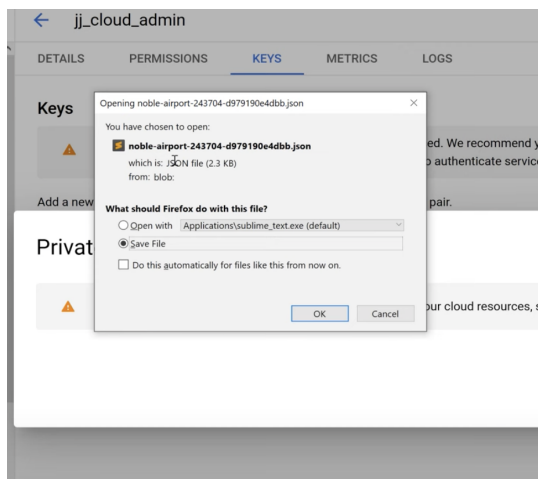
☐ P12

For backward compatibility with code using the P12 format

CANCEL

CREATE

Create JSON and save in project folder. You can now save the JSON to your computer. I recommend renaming the file to something easy to understand:



If you lose your key you can easily create a new one:

Keys



Service account keys could pose a security risk if compromised. We recommend you avoid downloading service account keys and instead use the [Workload Identity Federation](#). You can learn more about the best way to authenticate service accounts on Google Cloud [here](#).

Add a new key pair or upload a public key certificate from an existing key pair.

Block service account key creation using [organization policies](#).

[Learn more about setting organization policies for service accounts](#)

ADD KEY ▾

Create new key

Upload existing key

Key	Key creation date	Key expiration date	
d979190e4dbb3165e1f8f22e03e29eda5944cfa1	Jul 25, 2021	Jan 1, 10000	

Adding BigQuery Job User Role

To make queries we need to add a principal and grant access to the BigQuery Job User Role. Open your JSON file and copy the email in the field `client_email`:

```
{
  "type": "service_account",
  "project_id": "your-project-id",
  "private_key_id": "your-private-key-id",
  "private_key": "your-private-key",
  "client_email": "your-client-email",
  "client_id": "your-client-id",
  "auth_uri": "https://accounts.google.com/o/oauth2/auth",
  "token_uri": "https://oauth2.googleapis.com/token",
  "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
  "client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata/x509/"
}
```

Next click on IAM and then click on grant access:

Google Cloud My First Project

IAM & Admin

IAM

GRANT ACCESS

PERMISSIONS RECOMMENDATIONS HISTORY

Permissions for project "My First Project"

These permissions affect this project and all of its resources. [Learn more](#)

☐ Include Google-provided role grants

VIEW BY PRINCIPALS VIEW BY ROLES

Copy and paste your email into the principal field:

Grant principals access to this resource and add roles to specify what actions the principals can take. Optionally, add conditions to grant access to principals only when a specific criteria is met. [Learn more about IAM conditions](#)

Resource

My First Project

Add principals

Principals are users, groups, domains, or service accounts. [Learn more about principals in IAM](#)

New principals * ?

Assign roles

Roles are composed of sets of permissions and determine what the principal can do with this resource. [Learn more](#)

Select a role *

IAM condition (optional) ?
[+ ADD IAM CONDITION](#)



[+ ADD ANOTHER ROLE](#)

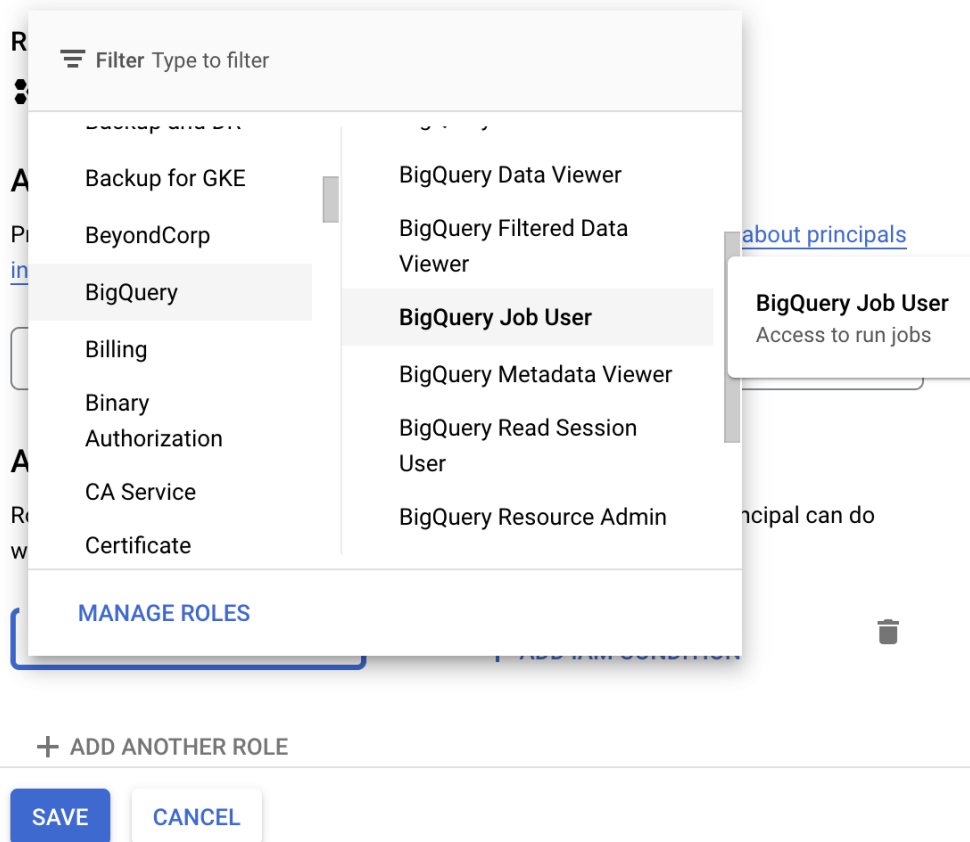
SAVE

CANCEL

Next click on select a role and add the BigQuery Job User role and save:

Grant access to "My First Project"

Grant principals access to this resource and add roles to specify what actions the principals can take. Optionally, add conditions to grant access to principals only when a specific criteria is met. [Learn more about IAM conditions](#)



You should now be able to use your key to run job queries.

To create a BigQuery client, create a new folder on your desktop called 'bigquery_tutorial'. Next, open a Jupyter notebook and save it in the 'bigquery_tutorial' folder. Also drag and drag the JSON private key file into the 'bigquery_tutorial' folder.

You can use the following code to create a client. In the first cell of your Jupyter notebook copy, paste and execute this command:

```
%pip install google-cloud-bigquery
```

In the subsequent cell you can paste the following to define your bigquery client:

```
import os
from google.cloud import bigquery
```

```
os.environ['GOOGLE_APPLICATION_CREDENTIALS'] = 'oreilly_demo_key.json'  
client = bigquery.Client()
```

During the course you will learn how to access tables using this client!