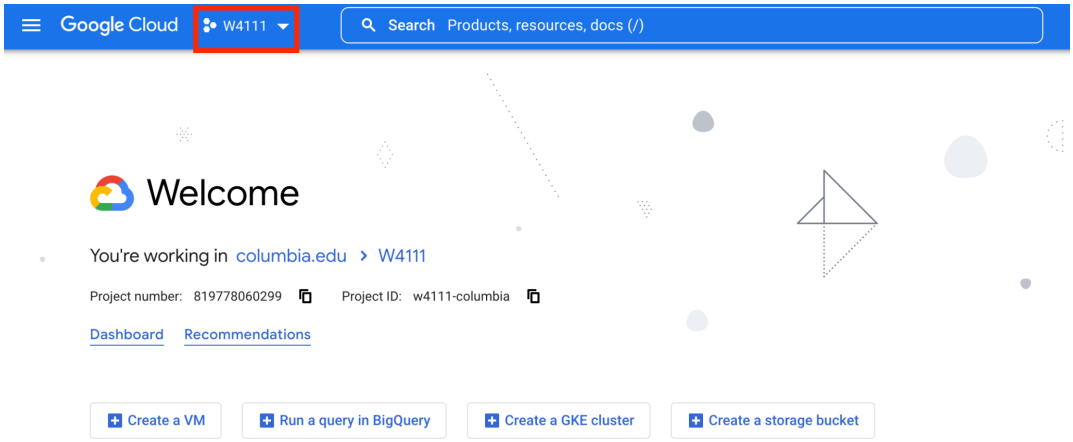
Project 2

Step 1 - Create your own GCP Project

To do this, open https://cloud.google.com/ and click on the console. On the top part of the dashboard, click on the project drop down. Select “New Project” and name it as “<UNI>-4111-PROJECT2”. Finalize by clicking on the create button.

Step 2 - Copy the dataset from 4111 GCP Project to personal project

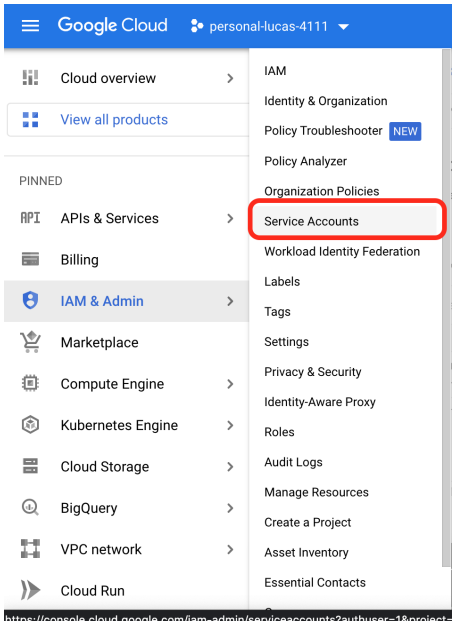
- Switch to the W4111 project (on the same tab you clicked to open the project drop down).



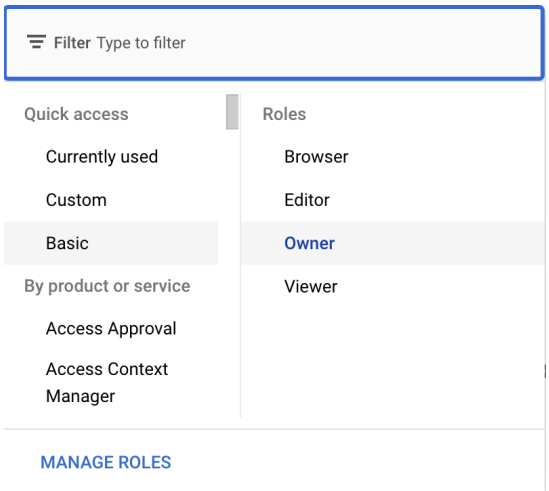
- Go to the BigQuery Console
- Click on the “graph” dataset
- Click on “Copy”
- On the opened tab, there is a “Destination” Field. Click on “create new dataset”
- On the Project ID, select the project you previously **created**.
- Give it an ID: graph
- Select data location as US
- After selecting Create, you’ll be prompted to enable data transfer API (you can just click the button to do so).
- Click copy
    - You might need to have billing enabled for this project. You can do so by going on billing console on gcp.
    - This will schedule a transfer. You can go to the console and click on “data transfer” to force the transfer to happen as soon as its available

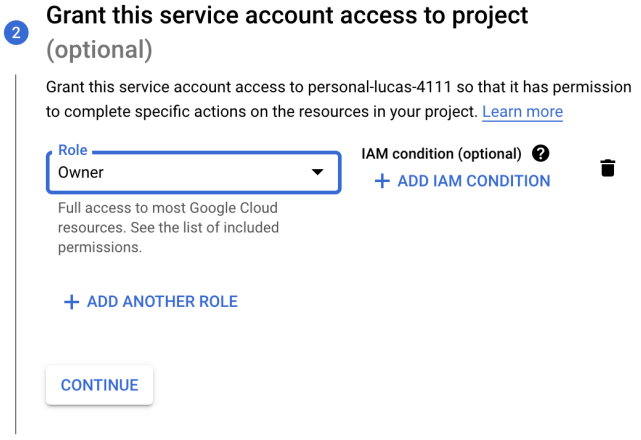
Step 3 - Generate service key

- Select your **own** project, and on the navigation dashboard, select:
    - IAM & ADMIN -> Service Accounts



- Select “+create service account”
    - Give it a name
    - **Add the following role for the account:** ~~BigQuery data owner~~ **owner** role for this account





    - Select Ok
- On this newly created service account, click on the three dots and select “manage keys”
- On this new tab, select “Add keys”, “Create new keys”
- Select type JSON, and a json file should be downloaded.
- Save this file for later.

Step 4 - Create DBT account

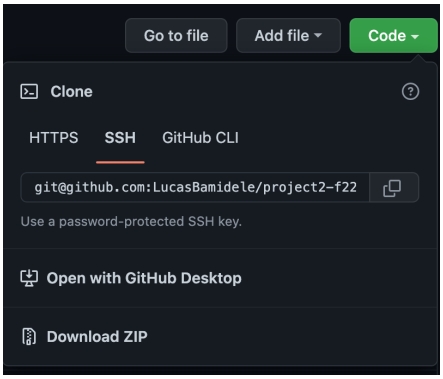
- Go to https://cloud.getdbt.com/
- Create a new account (preferably use your columbia mail. Fill the “Name of the Company” as Columbia University

Step 4.5 - Setting up a git Repository

- This step will allow you to maintain your project more easily and download the files to your machine.
- Go to https://github.com/w4111/project2-f22-template
- Click on **Use this template**
- **IMPORTANT:** When prompted to create a new repository, make sure you set the repository to **private**
    - **DO NOT CREATE A PUBLIC REPOSITORY. This will be treated as academic dishonesty.**
- Create the repository.

Step 5 - Create/setting up DBT Project

- Select Create new account on the top right of the page.
- This will open up a page for you to set up your project. Name your project
- On the Select Warehouse “BigQuery”
- On the “configure environment" phase, click on the “Upload a Service Account JSON file” and upload the file you downloaded in step 3.
    - You can click on “Test connection” to make sure you’ll are able to connect to your BigQuery environment. If something fails, check if you missed any of the previous steps or create a post on ED.
- On the setting up a repository part, you can select git clone.
    - Get the link for cloning the repository on the github website



    - Copy the link in the text bar
    - The project will generate a deploy key. Copy the deploy key

    - Open up your github repository
    - Go to settings -> deploy keys -> add deploy keys
    - Give it a name and paste the deploy key generated from dbt.
    - **Check Allow write access**
    - Save changes.
    - You'll now have a templated directory to work with.
- You should be able to open up the DBT IDE

Step 7 - Run sample query

- In the project 2 folder, create a new file called example.sql
- Write the following in the sheet:

```
{{ config(materialized='table') }}
Select * from `graph.tweets` limit 10
```

    - Preview the query with cmd + enter (or ctrl + enter)
    - You should be able to see 10 entries from the tweets table
- Now save the file (ctrl + S or cmd + s)
- Click on commit (and write any message)
- Now, on the console (on the bottom), run:
- `dbt run --select models/project2/example.sql`
- Go to your BigQuery Console
    - You should be able to see a new dataset with a new table.
- Delete the example.sql file (on the DBT project)
    - Commit and push this change
- Delete the table on your bigquery console.

Step 8 - Important notes

- You'll need to create a new branch every time you want to execute changes to this project!
- The default for your project will be read-only. Everything can be done on the IDE itself The development flow is:
    - Create a new branch
    - Work on this development branch
    - Once you are happy with your changes, save all your changes and click "commit"
    - You can test your changes (run dbt run or other commands). If everything is in order, merge it into the main branch
- You should read the **dbt and BigQuery** docs and have them as a reference. You can leverage it to make your work easier. Part of the goal of this assignment is for you to familiarize with tools that are actually used in industry.

- You should submit only .sql files. **Don't make any changes on the configurations of your project as we need to be able to reproduce it.** You can use the {{config}} command in your own file if you need to.
    - The tables that you generate should be used to make sure your code runs and have predictable results
- Hint: You can run:

  ```
  dbt run
  dbt test
  ```
- To run simple tests and make some simple sanity checks (running these tests DOES NOT mean that you got the correct answer - but passing them means you're probably on the right track).
- Don't hesitate to ask questions on **Ed**. Unforeseen things might happen.

Step 9 - Submission

- Make sure all of your tables work by running **dbt run**.
- Download your dbt project from Github as a zip file.
    - Go to your github repository link
    - Click the "code" button
        - Select Download as ZIP
- Unzip the project file
- Now zip **only** the **models folder**, the one that contains all of your .sql model files.
    - If you're in a mac, you can right click the **models** folder on finder and click compress
    - If you're in a windows OS, you can right click the folder, select **Send to** and then **Compressed (Zipped) Folder**
    - If on Linux (ubuntu), you can run on your terminal (on the project folder):

      ```
      zip -r <filename>.zip models
      ```
- Name the .zip files as {UNI1}-{UNI2}.zip and submit it to **Gradescope.**
- **Your Gradescope submission should only contain your zipped models folder.**