**Overview**

[Dataplex Universal Catalog](https://cloud.google.com/dataplex) Enables centralized discover, management, monitoring, and governing data and AI artifacts across your data platform, providing access to trusted data and powering analytics and AI at scale.

Dataplex Universal Catalog (previously named Dataplex, Dataplex Catalog, and BigQuery Universal Catalog) is a fully managed, scalable metadata management service that you can use to tag data assets using aspects and search for assets to which you have access. Aspects are a template to allow you to attach metadata fields to specific data assets for easy identification and retrieval (such as marking certain assets as containing protected or sensitive data); in addition to predefined aspects, you can also create custom aspects to assign to different data assets.

In this lab, you learn how to use Dataplex Universal Catalog to attach aspects to a data asset and then search for assets.

What you'll do

* Enable the Dataplex and BigQuery APIs
* Create a lake, zone, and asset in Dataplex Universal Catalog
* Create a custom aspect type
* Apply aspects based on an aspect type to assets
* Search for assets using aspects

**Setup and requirements**

Before you click the Start Lab button

Read these instructions. Labs are timed and you cannot pause them. The timer, which starts when you click **Start Lab**, shows how long Google Cloud resources will be made available to you.

This hands-on lab lets you do the lab activities yourself in a real cloud environment, not in a simulation or demo environment. It does so by giving you new, temporary credentials that you use to sign in and access Google Cloud for the duration of the lab.

What you need

To complete this lab, you need:

* Access to a standard internet browser (Chrome browser recommended).
* Time to complete the lab.

**Note:** If you have a personal Google Cloud account or project, do not use it for this lab.**Note:** If you are using a Pixelbook, open an Incognito window to run this lab.

Log in to Google Cloud Console

1. Using the browser tab or window you are using for this lab session, copy the **Username** from the **Connection Details** panel and click the **Open Google Console** button.

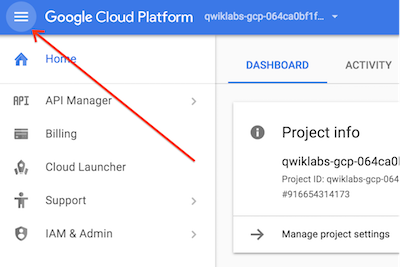
**Note:**If you are asked to choose an account, click **Use another account**.

1. Paste in the **Username**, and then the **Password** as prompted.
2. Click **Next**.
3. Accept the terms and conditions.

Since this is a temporary account, which will last only as long as this lab:

* Do not add recovery options
* Do not sign up for free trials

1. Once the console opens, view the list of services by clicking the **Navigation menu** (Navigation menu icon) at the top-left.



Verify or enable required APIs

1. In the Google Cloud Console, enter **Cloud Dataplex API** in the top search bar.
2. Click on the result for **Cloud Dataplex API** under Marketplace.
3. If the API is not already enabled, click **Enable** to enable the API.
4. Repeat steps 1-3 for **BigQuery API**.

Check IAM permissions

1. From the navigation menu select **IAM & Admin** and from the flyout submenu select **IAM**.
2. Find your entry which will look like **student-xx-xxxxxxxxxxxx@qwiklabs.net**
3. Verify you have the role **Dataplex Administrator** and **Dataplex Catalog Admin**

**Task 1. Create a lake, zone, and asset**

In this task, you create a new Dataplex Universal Catalog lake to store customer order information, add a curated zone to the lake, and then attach a pre-created BigQuery dataset as a new asset in the zone.

Create a lake

1. In the Google Cloud Console, in the **Navigation menu** (Navigation menu), navigate to **View All Products** > **Analytics** > **Dataplex Universal Catalog**.

If prompted Welcome to the new Dataplex experience, click **Close**.

1. Under **Manage lakes**, click **Manage**.
2. Click **Create**.
3. Enter the required information to create a new lake:

|  |  |
| --- | --- |
| **Property** | **Value** |
| **Display Name** | Orders Lake |
| **ID** | Leave the default value. |
| **Region** | europe-west4 |

Leave the other default values including **Metastore service** as None.

1. Click **Create**.

It can take up to 3 minutes for the lake to be created.

Add a zone to the lake

1. On the **Manage** tab, click on the name of your lake.
2. Click **Add zone**.
3. Enter the required information to create a new zone:

|  |  |
| --- | --- |
| **Property** | **Value** |
| **Display Name** | Customer Curated Zone |
| **ID** | Leave the default value. |
| **Type** | **Curated zone** |
| **Data locations** | **Regional** |

Leave the other default values.

For example, the option for **Enable metadata discovery** under **Discovery settings** is enabled by default and allows authorized users to discover the data in the zone.

1. Click **Create**.

It can take up to 2 minutes for the zone to be created.

You can perform the next task once the status of the zone is **Active**.

Attach an asset to a zone

1. On the **Zones** tab, click on the name of your zone.
2. On the **Assets** tab, click **Add assets**.
3. Click **Add an asset**.
4. Enter the required information to attach a new asset:

|  |  |
| --- | --- |
| **Property** | **Value** |
| **Type** | **BigQuery dataset** |
| **Display Name** | Customer Details Dataset |
| **ID** | Leave the default value. |
| **Dataset** | qwiklabs-gcp-01-b5155ae2af7e.customers |

Leave the other default values.

1. Click **Done**.
2. Click **Continue**.
3. For **Discovery settings**, select **Inherit** to inherit the Discovery settings from the zone level, and then click **Continue**.
4. Click **Submit**.

Click **Check my progress** to verify your performed task.

Assessment Completed!

Create a lake, zone, and asset in Dataplex.

*Assessment Completed!*

**Task 2. Dataplex Universal Catalog setup**

We will now work with Dataplex Universal Catalog and enrich the BigQuery asset you just configured.

1. On the left menu click **Search**.
2. The search platform will automatically be **Dataplex Universal Catalog**. Ignore any messages about the prior product **Data Catalog**.
3. In the **Filters** section go to the **Systems** section and select **BigQuery**.
4. Once the BigQuery assets load on the right, search for **Customers**, then **star** it and click **Customers** which has the **Type alias** of **DATASET**. Changing **Sort by** to **Last modified (Recent first)** will make it easier to find.
5. Click the **LIST** link in header.
6. Click **customer\_details**.
7. In the **Tags & aspects** section under **Required aspects** click the down facing **show more** arrow on the right side of the **Storage** entry. Notice the metadata under **Resource Name** showing the complete path to the BigQuery table.
8. In the header click on **SCHEMA** and notice the field names and other metadata for the fields.
9. In the header click on **Data Profile**, next click on **QUICK DATA PROFILE**. Click **CONFIRM** in the message box.
10. In the header click on **Data Quality**, next click on **CREATE DATA QUALITY SCAN**. Enter the required information:

|  |  |
| --- | --- |
| **Property** | **Value** |
| **Display Name** | Customer Detail |
| **Sampling size \*** | 20% |

1. Verify **Publish results to BigQuery and Dataplex Catalog UI** is checked. Leave the rest of the fields as their defaults. Click **CONTINUE**.
2. Click **ADD RULES**.
3. Select **Profile based recommendationions** in the drop down menu.
4. In the field **Choose columns \*** click **BROWSE**. In the pop-up check the box to the left of **Name** to select all the fields then click the **SELECT** button at the bottom. If no data fields appear you will need to wait a couple of minutes for the data profile scan you just created to run then start again at step 12
5. Check the box to the left of **Column name** to select all rows.
6. Scroll to the bottom and click **SELECT**.
7. In the next screen scroll to the bottom and click **CONTINUE**.
8. In the box **Select BigQuery dataset** click **BROWSE**.
9. Select the radio button for **Customers** and click **SELECT**.
10. In the box **BigQuery table** enter **customer\_details\_quality\_scan**.
11. Click **RUN SCAN**.
12. We will now add a new aspect. Under the **Manage Metadata** menu on the left Click **Catalog**.
13. Click **CREATE ASPECT TYPE**.
14. Click the box labeled **Data Sensitivity**.
15. Click **USE EXAMPLE**.
16. Set the **Location** field **europe-west4**.
17. Click **SAVE**.
18. We will now add the new Aspect and connect it to fields in the table. Click **Search** in the left menu bar. Choose search platform by clicking **Dataplex Universal Catalog** on the right of the header bar.
19. Select the box for **BigQuery** in the **Systems** section under **Filters**.
20. Select **customer\_details**.
21. While you can connect an aspect at the table level on the **DETAILS** tab by using **+ ADD** next to **Optional tags & aspects** at the bottom of the page, we are going to connect it to specific fields instead.
22. To add or edit aspects to individual columns of our table. Click the **Schema** tab.
23. Select the **zip** field.
24. Click **+ ADD TAG OR ASPECT** and select **Data\_Sensitivity** on the pop-up menu.
25. On the fly-out set **Is Encrypted** False **Has PII** True. **PII Type** Other then click **SAVE**.
26. Repeat step 33-34 for each of the following fields using the same value for **Is Encrypted** and **Has PII** as above:

|  |  |
| --- | --- |
| **Name** | **PII Type** |
| **age** | Other |
| **email** | EMAIL |
| **latitude** | Other |
| **city** | Address |
| **longitude** | Other |

Click **Check my progress** to verify your performed task.

Please add or edit aspects for specific columns in the table as instructed.

Dataplex Universal Catalog setup.

*Please add or edit aspects for specific columns in the table as instructed.*

**Task 3. Searching Dataplex Universal Catalog**

1. On the left menu click **Search**.
2. In the **Filters** section scroll down to **Aspects** and select **Data Sensitivity**.
3. Once the BigQuery assets load on the right, click **customer\_details** which has the **Type alias** of **TABLE**.
4. Click on the **SCHEMA** tab. Notice the fields marked **Data Sensitivity**.
5. Click on one of the **Data Sensitivity** tags and notice the setting in the pop-out. Click **CLOSE** on the pop-out.
6. You now know how to locate data by the **aspect** associated to it.

Redshift to BigQuery Migration using Data Transfer Service

experimentLabschedule1 hour 30 minutesuniversal\_currency\_altNo costshow\_chartIntermediate

infoThis lab may incorporate AI tools to support your learning.

**Introduction**

In this lab, you will look at the process and execution of end-to-end data migration from Redshift to BigQuery using provided sample data. To demonstrate this process, the exercise uses the BigQuery Data Transfer Service for both data and schema migration from Redshift to BigQuery.

To learn more about this process read the following documentation on [Redshift to BigQuery Migration Overview](https://cloud.google.com/bigquery/docs/migration/redshift-overview).

What you'll learn

* Create BigQuery transfers using the BigQuery Data Transfer Service
* End-to-end data migration from Redshift to BigQuery

**Setup**

**Before you click the Start Lab button**

Read these instructions. Labs are timed and you cannot pause them. The timer, which starts when you click **Start Lab**, shows how long Google Cloud resources will be made available to you.

This Qwiklabs hands-on lab lets you do the lab activities yourself in a real cloud environment, not in a simulation or demo environment. It does so by giving you new, temporary credentials that you use to sign in and access Google Cloud for the duration of the lab.

**What you need**

To complete this lab, you need:

* Access to a standard internet browser (Chrome browser recommended).
* Time to complete the lab.

**Note:** If you already have your own personal Google Cloud account or project, do not use it for this lab.

**Note:** If you are using a Pixelbook, open an Incognito window to run this lab.

**How to start your lab and sign in to the Google Cloud Console**

1. Click the **Start Lab** button. If you need to pay for the lab, a pop-up opens for you to select your payment method. On the left is a panel populated with the temporary credentials that you must use for this lab.



2. Copy the username, and then click **Open Google Console**. The lab spins up resources, and then opens another tab that shows the **Sign in** page.



***Tip:*** Open the tabs in separate windows, side-by-side.

 If you see the **Choose an account** page, click **Use Another Account**.



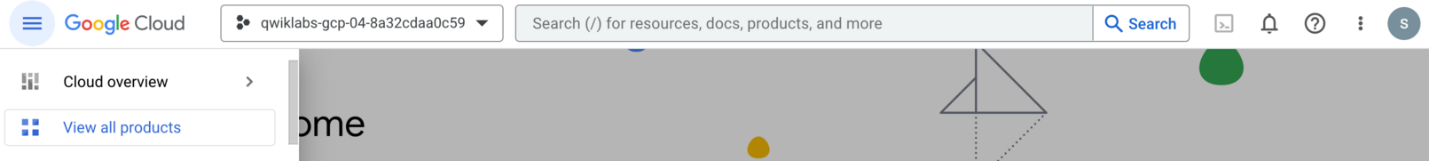
3. In the **Sign in** page, paste the username that you copied from the Connection Details panel. Then copy and paste the password.

***Important:*** You must use the credentials from the Connection Details panel. Do not use your Qwiklabs credentials. If you have your own Google Cloud account, do not use it for this lab (avoids incurring charges).

4. Click through the subsequent pages:

* Accept the terms and conditions.
* Do not add recovery options or two-factor authentication (because this is a temporary account).
* Do not sign up for free trials.

After a few moments, the Cloud Console opens in this tab.

**Note:** You can view the menu with a list of Google Cloud Products and Services by clicking the **Navigation menu** at the top-left. 

Activate Cloud Shell

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

In the Cloud Console, in the top right toolbar, click the **Activate Cloud Shell** button.



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:



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

Copied!

(Output)

Credentialed accounts:

- <myaccount>@<mydomain>.com (active)

(Example output)

Credentialed accounts:

- google1623327\_student@qwiklabs.net

You can list the project ID with this command:

gcloud config list project

Copied!

(Output)

[core]

project = <project\_ID>

(Example output)

[core]

project = qwiklabs-gcp-44776a13dea667a6

For full documentation of gcloud see the [gcloud command-line tool overview](https://cloud.google.com/sdk/gcloud).

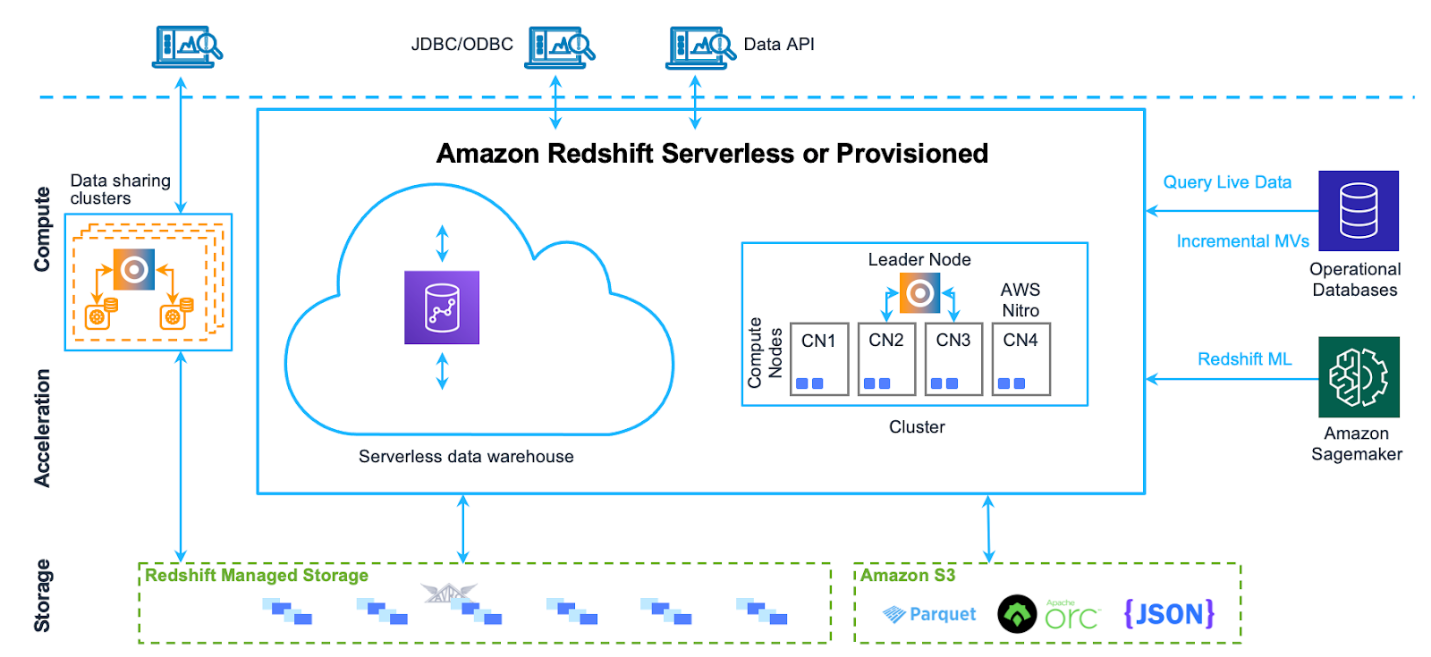
**Reference Architecture**

BigQuery and Amazon Redshift are both based on a massively parallel processing (MPP) architecture. Queries are distributed across multiple servers to accelerate their execution. As for system architecture, Amazon Redshift and BigQuery primarily differ in how data is stored and how queries are executed.

In BigQuery, the underlying hardware and configurations are abstracted away so that storage and compute allow your data warehouse to grow without any intervention.

Redshift and BigQuery are both analytic data warehouses but have some key architectural differences.

In Amazon Redshift, CPU, memory, and disk storage are tied together through [compute nodes](https://docs.aws.amazon.com/redshift/latest/dg/c_high_level_system_architecture.html), as illustrated in [this diagram from the Amazon Redshift documentation](https://docs.aws.amazon.com/redshift/latest/dg/c_high_level_system_architecture.html).



Cluster performance and storage capacity are determined by the type and the quantity of compute nodes, both of which must be configured. To change compute or storage, you need to resize your cluster through a process ([over a couple of hours and possibly up to two days or longer](https://docs.aws.amazon.com/redshift/latest/mgmt/managing-cluster-operations.html)).

This process creates a brand new cluster and copies the data over. Amazon Redshift also offers RA3 nodes with managed storage that help separate compute and storage. The largest node in the RA3 category caps at 64 TB of managed storage for each node.

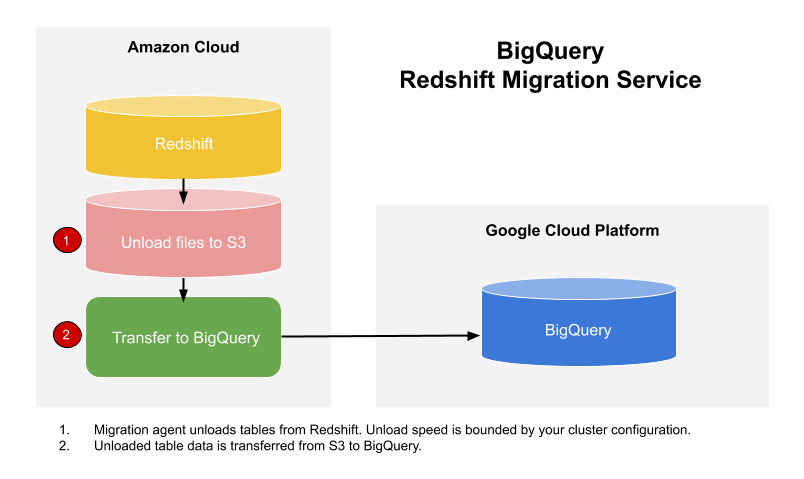
From the start, BigQuery does not tie together compute, memory, and storage but instead treats each separately. BigQuery compute is defined by [slots](https://cloud.google.com/bigquery/docs/slots), a unit of computational capacity required to execute queries. Google manages the entire infrastructure that a slot encapsulates, eliminating all but the task of choosing the proper slot quantity for your BigQuery workloads. BigQuery memory is provided by a [remote distributed service](https://cloud.google.com/blog/products/gcp/in-memory-query-execution-in-google-bigquery), connected to compute slots by Google's petabit network, all managed by Google.

BigQuery and Amazon Redshift both use columnar storage, but BigQuery uses [variations and advancements](https://cloud.google.com/blog/products/gcp/inside-capacitor-bigquerys-next-generation-columnar-storage-format) on columnar storage. While columns are being encoded, various statistics about the data are persisted and later are used during query execution to compile optimal plans and to choose the most efficient runtime algorithm.

BigQuery stores your data in [Google's distributed file system](https://cloud.google.com/static/files/storage_architecture_and_challenges.pdf), where it's automatically compressed, encrypted, replicated, and distributed. This is all accomplished without affecting the computing power available for your queries. Separating storage from compute lets you scale up to dozens of petabytes in storage seamlessly, without requiring additional expensive compute resources. A number of other [benefits of separating compute and storage](https://cloud.google.com/blog/products/gcp/separation-of-compute-and-state-in-google-bigquery-and-cloud-dataflow-and-why-it-matters) exist as well.

This lab describes the process of migrating data from Amazon Redshift to BigQuery using the BigQuery Data Transfer Service. The service engages migration agents on Google Kubernetes Engine (GKE) and triggers an unload operation from Amazon Redshift to a staging area in an Amazon S3 bucket. Then the BigQuery Data Transfer Service transfers your data from the Amazon S3 bucket to BigQuery.

This diagram shows the overall flow of data between an Amazon Redshift data warehouse and BigQuery during a migration.

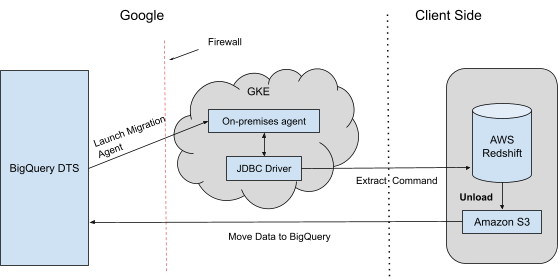


**Migration Prerequisites**

The BigQuery Data Transfer Service spins up agents on GKE which connect to Redshift via a JDBC URL. The Data Transfer Service also takes care of schema conversions for moving data from Redshift to BigQuery.

The migration process has two steps:

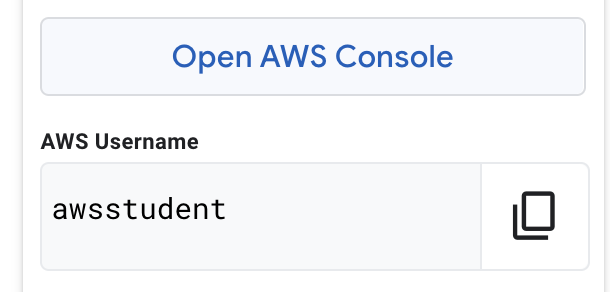
* **UNLOAD from Redshift to S3**: The GKE agent initiates an UNLOAD operation from Redshift to a staging bucket in S3. The agent extracts Redshift data as a compressed file, which helps customers minimize the egress costs.
* **Transfer Unloaded Data from S3 to BigQuery**: The unloaded data is then copied from the S3 staging bucket to BigQuery dataset.



**Task 1: Generating Sample Data in Redshift for Migration**

The following CSV will be used as the data sample redshift\_sample\_data.csv.

1. Login to the **AWS Console** using the **AWS Username** awsstudent and the **AWS Password** provided. Use the **Open AWS Console** button provided in the lab details pane to open the AWS console login page in a new incognito browser tab.



1. In the top search bar enter S3 and select the service when it appears.
2. Click on **Create bucket** to create an S3 bucket.

Enter the following details for the bucket:

|  |  |
| --- | --- |
| **Name** | **Value** |
| Bucket name | qwiklabs-gcp-04-6b8ab7514f17 |
| AWS Region | us-east-1 |
| Block all public access | (Uncheck) |
| I acknowledge that the current settings might result in this bucket and the objects within becoming public | Check |

Then scroll to the bottom of the page and select **Create bucket**.

1. Once the S3 bucket is created upload the file redshift\_sample\_data.csv to the bucket. You can download the file to your local machine using the link provided below:

https://storage.googleapis.com/partner-usecase-bucket/ucase047/redshift\_sample\_data.csv

Copied!

Click **Upload** to upload the file to S3 once downloaded.

1. On the **Upload** page of S3, select **Add files**, select redshift\_sample\_data.csv from your local machine then click **Upload** at the bottom of the page.

When the upload completes click **Close**.

Click **Check my progress** to verify the objective.

Assessment completed!

Create s3 storage bucket and upload a file on it

*Assessment completed!*

1. To ensure that the BigQuery Data Transfer Service can access the bucket click the **Permissions** tab to edit the bucket's permissions.
2. In the **Bucket Policy** section, click **Edit**.
3. Copy and paste in the below JSON string to the bucket policy.

{

"Version": "2008-10-17",

"Statement": [

{

"Sid": "AllowPublicRead",

"Effect": "Allow",

"Principal": {

"AWS": "\*"

},

"Action": "s3:\*",

"Resource": "arn:aws:s3:::qwiklabs-gcp-04-6b8ab7514f17/\*"

}

]

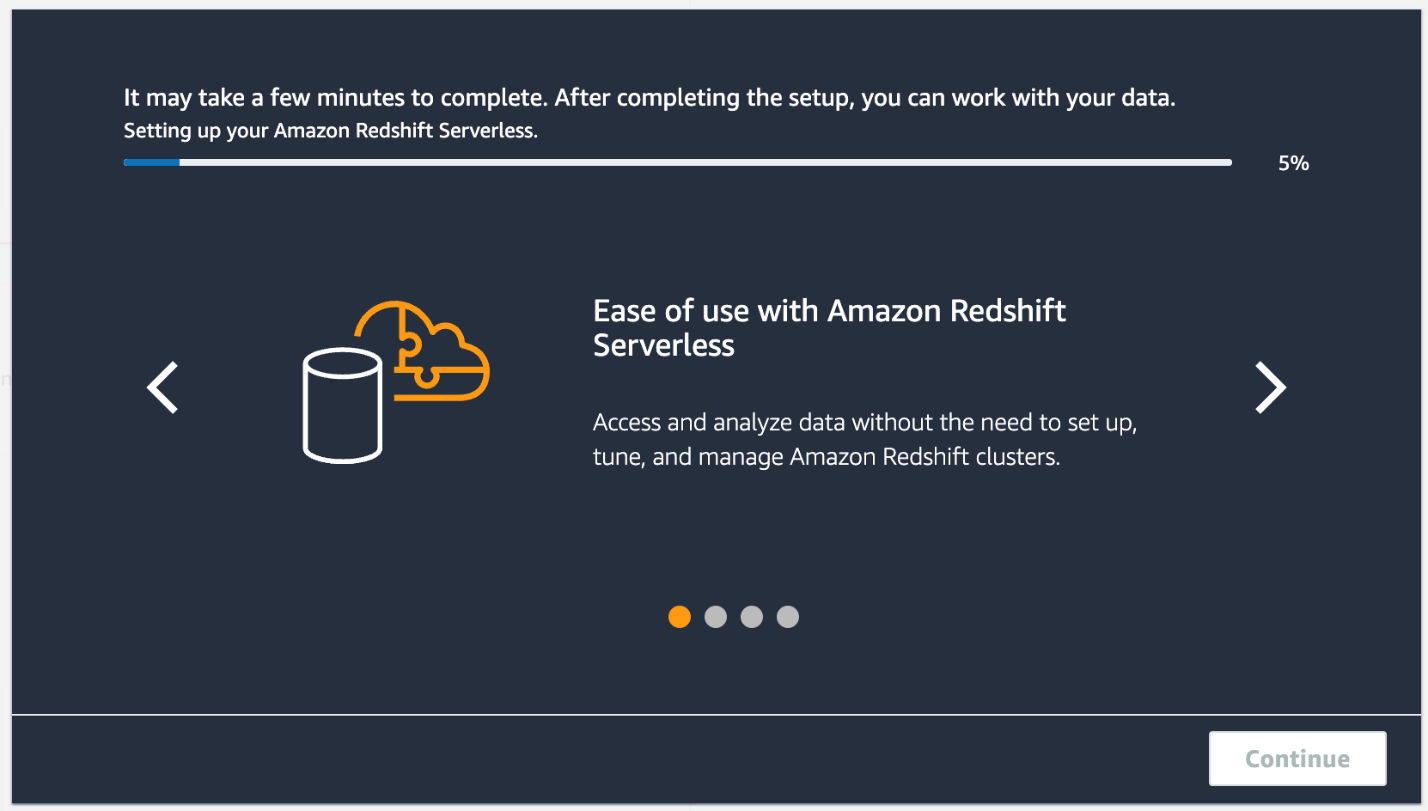
}

Copied!

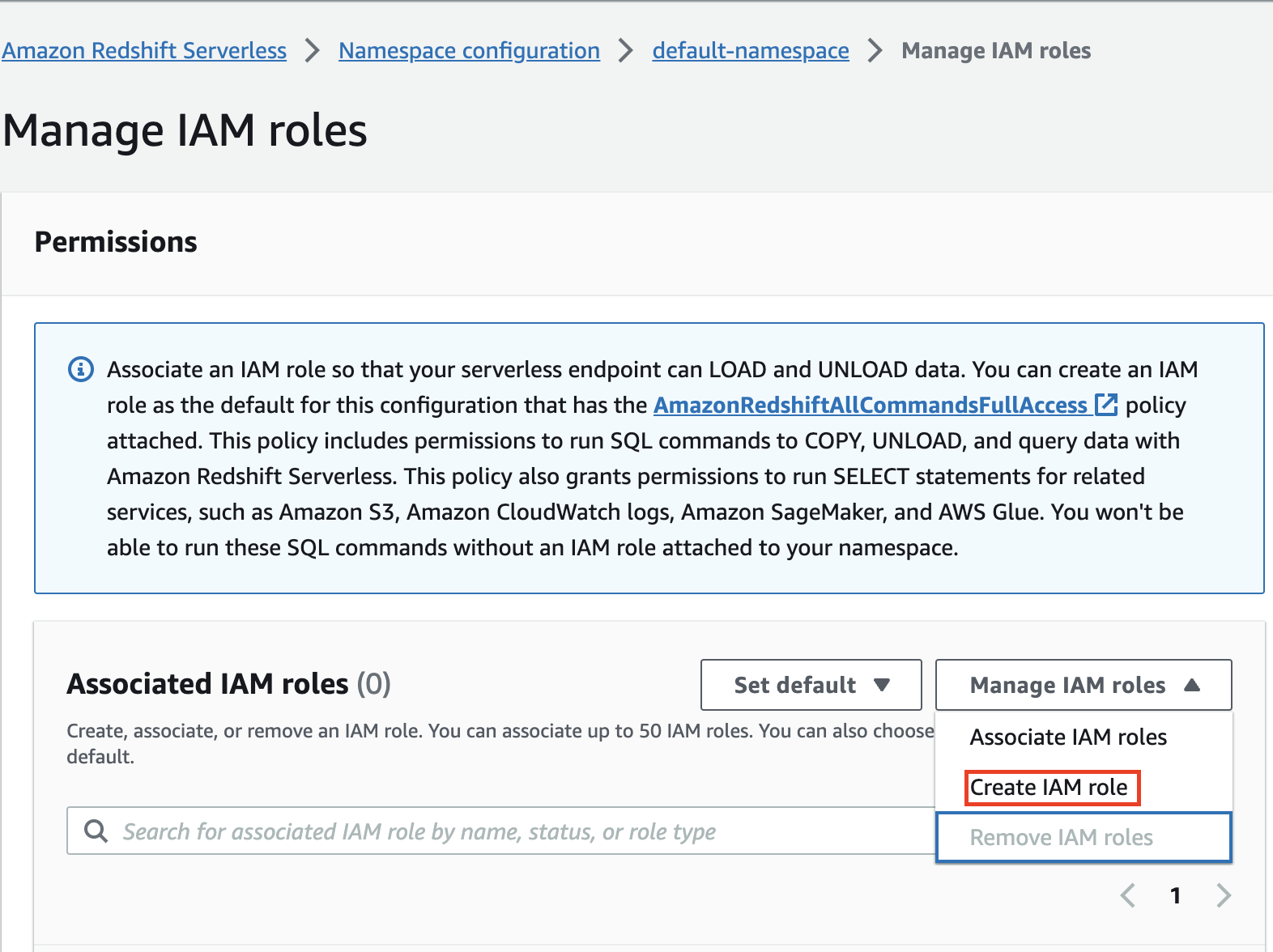
Click **Save changes** when finished.

Load the sample data

1. Navigate to the Redshift service in the AWS Console. In the top search, type Redshift and select the Redshift service when it appears.
2. Click on **Try Redshift Serverless** then on the next page leave "Use the default settings" selected and click **Save Configuration** at the bottom of the page. You will be presented with a dialog similar to the image below. Wait until the service completes provisioning before proceeding.

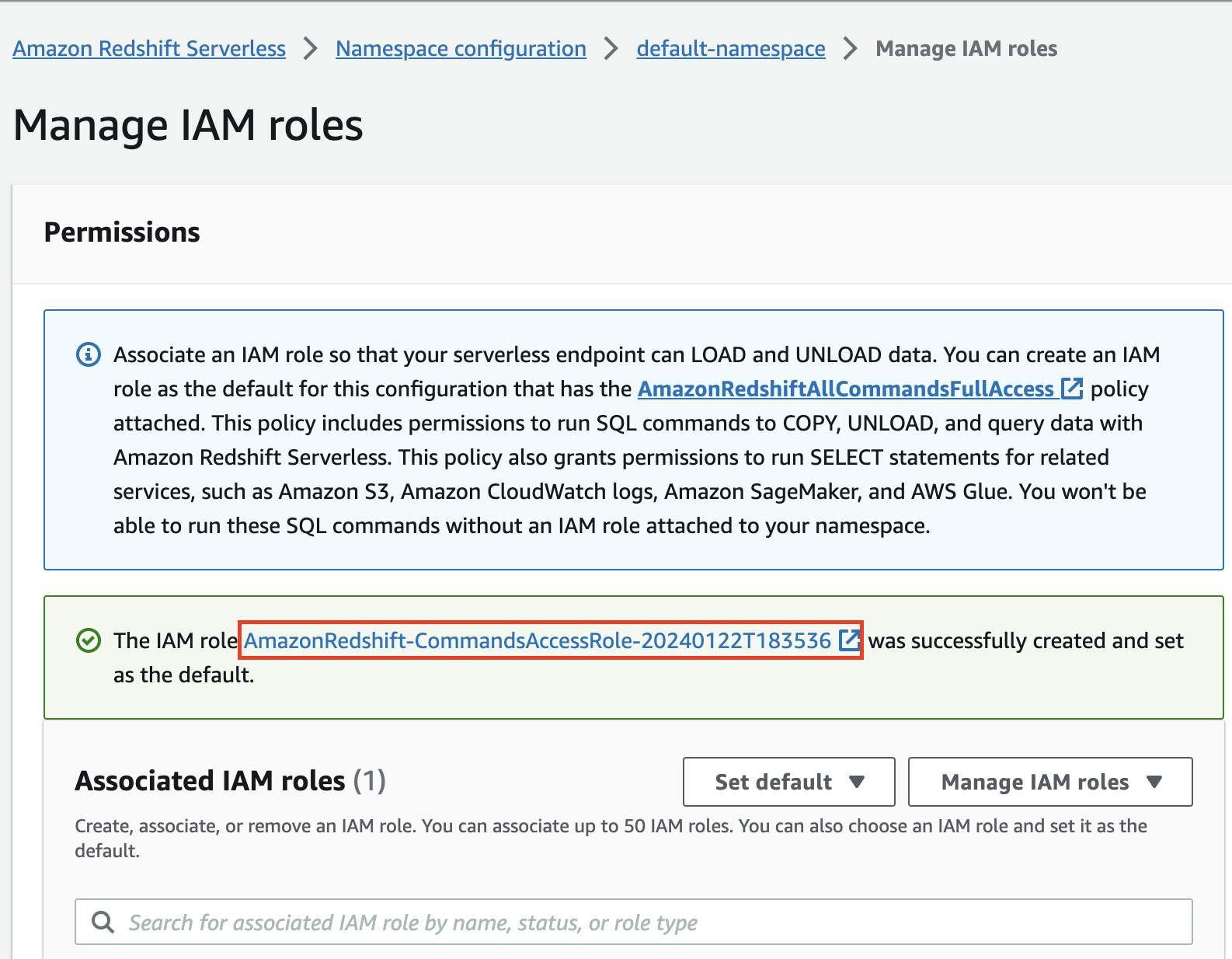


1. Click **Continue** when the provisioning dialog has finished.
2. In the left navigation click on **Namespace configuration**. On the resulting page click on the default-namespace hyperlink to see its details.
3. On the default-namespace namespace configuration, select the **Security and encryption** tab, then click on the **Manage IAM roles** button.
4. On the **Manage IAM Roles** page, click on the **Manage IAM Roles** dropdown and select **Create IAM role**.

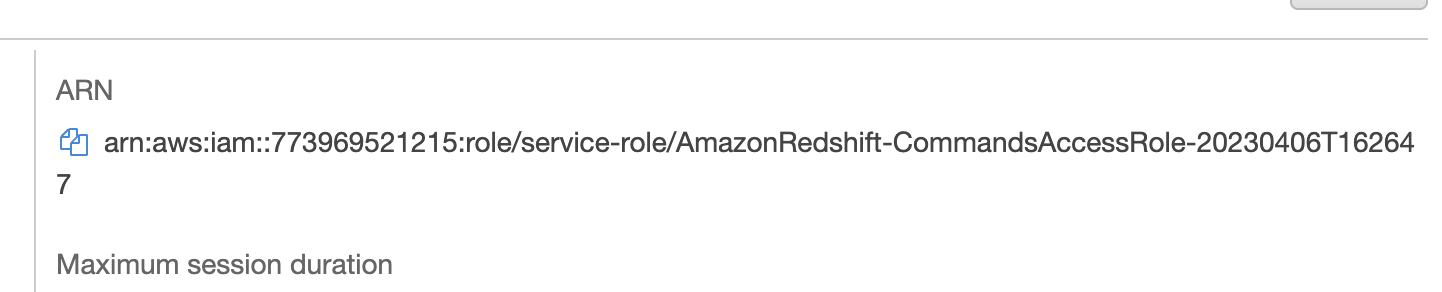


1. Select the radio option **Any S3 bucket** then select **Create IAM role as default**. Once the Role creates you will be re-directed to the **Manage IAM roles** page.

Click on the link for the newly created role. It will look similar to the following AmazonRedshift-CommandsAccessRole-20230406T162647.

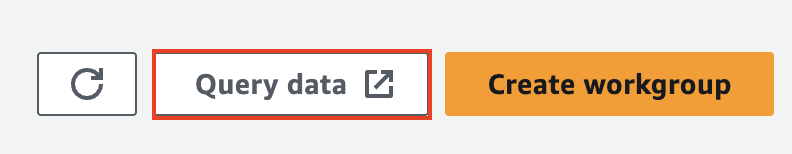


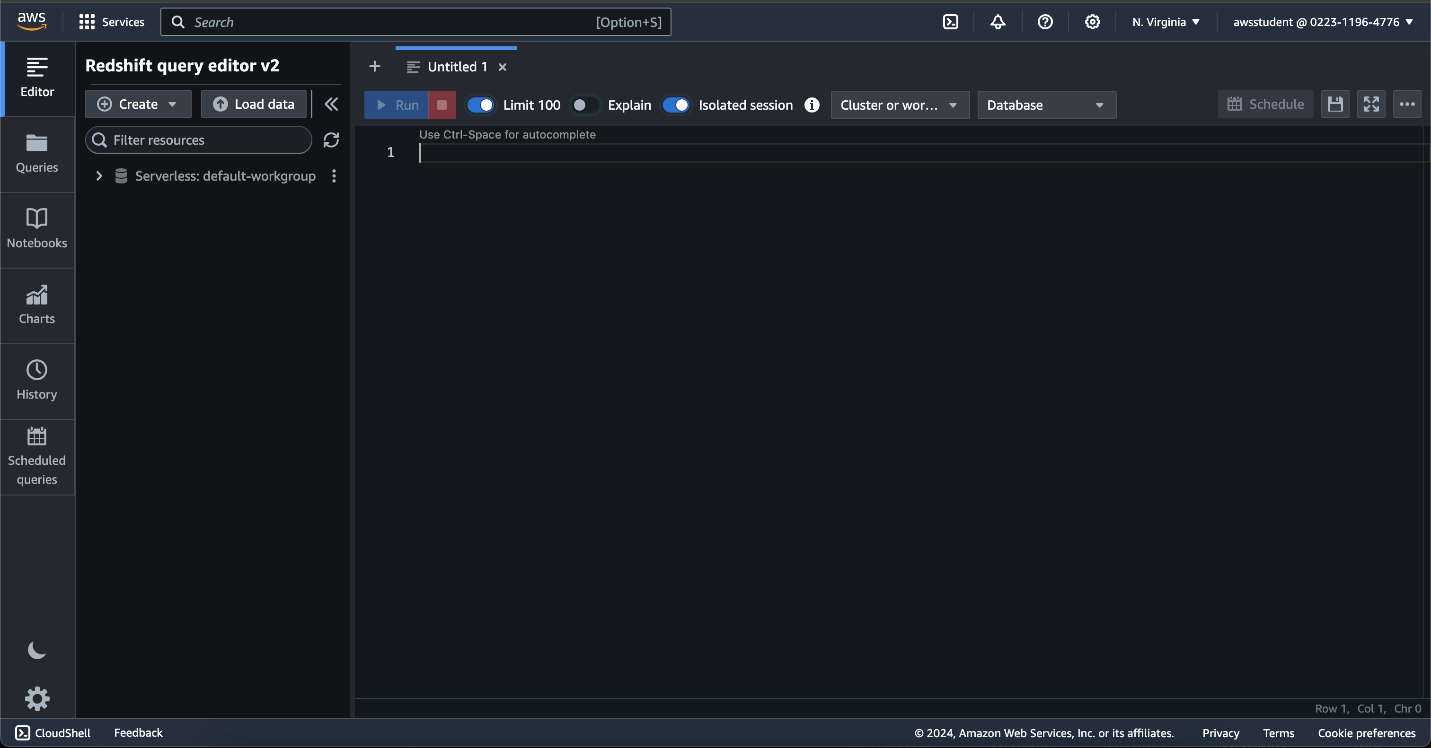
On the IAM Role page, copy the Role ARN and note it down, you will use this in the steps below.



Navigate to the prevision IAM role page and click on **Save changes**.

1. Navigate back to the Redshift landing page by searching for Redshift in the top search. Click **Query data** near the top right corner of the Redshift landing page.



1. Now you can see Redshift query editior page as follow: 

**Note: If you are not able to see the above Redshift query editor UI then please use the normal window tab instead of incognito and login to the AWS account using the login credentials provided in the connection details panel on the left-hand side.**

1. From left navigation, click on Serverless: default-workgroup and then select Federated user and click on **Create connection**.
2. Run the following query to create the table used to store the sample data.

CREATE TABLE "public"."category\_sample"(

"catid" SMALLINT NULL,

"catgroup" VARCHAR NULL,

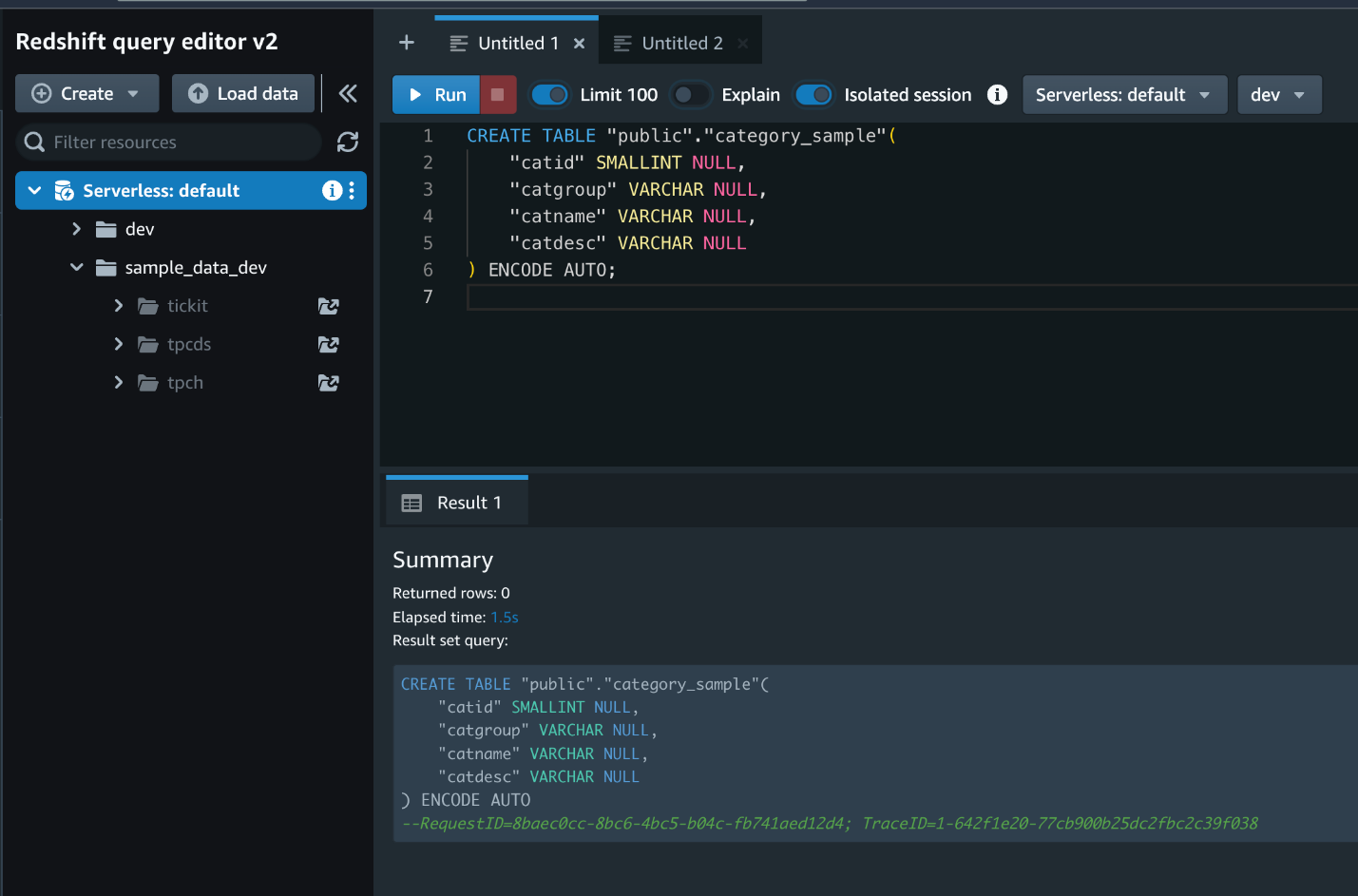
"catname" VARCHAR NULL,

"catdesc" VARCHAR NULL

) ENCODE AUTO;

Copied!

Press **Run** to execute the query. The result should look similar to the below figure.

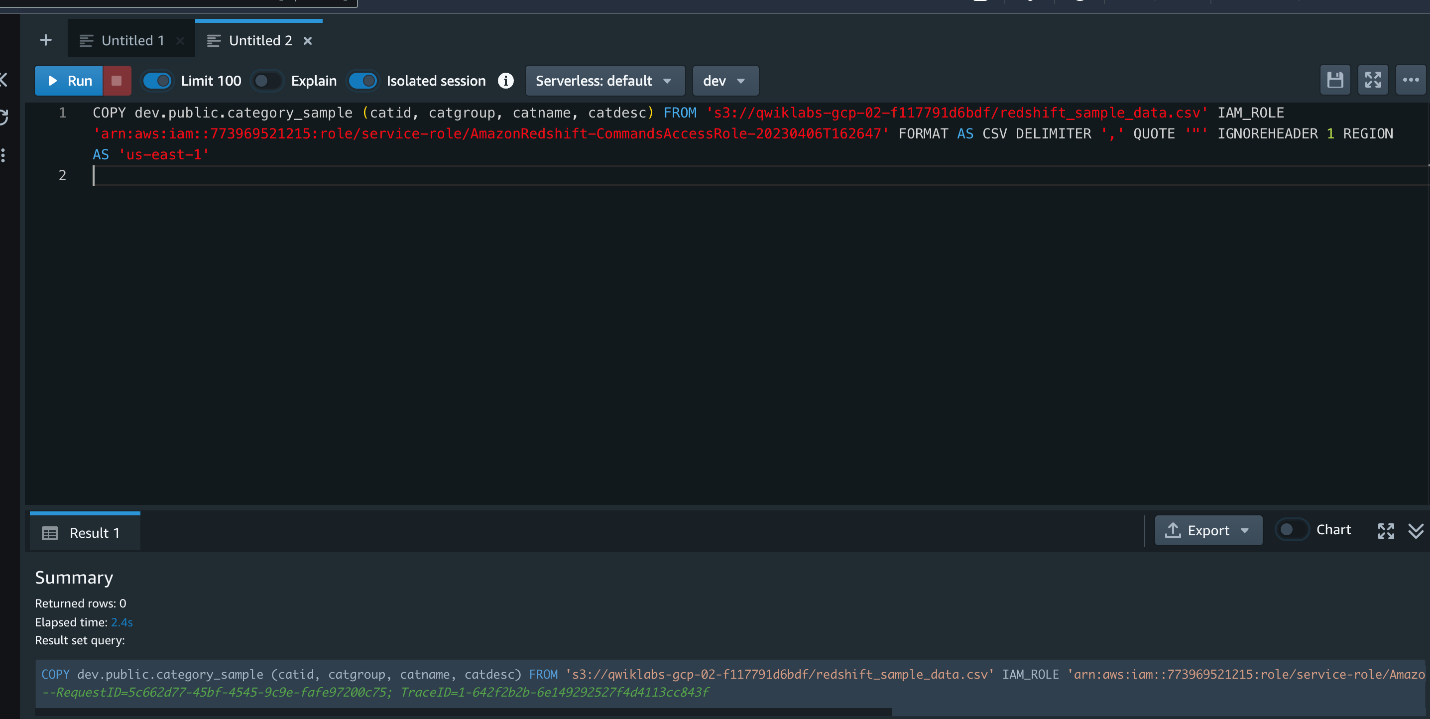


1. Now that the table has been created, you can copy data from the S3 bucket to Redshift using the IAM Role created in earlier steps.

To do so, run the query below in the Redshift query editor in a new query tab. Replace the S3 bucket with qwiklabs-gcp-04-6b8ab7514f17 and enter the **Role ARN** in the query where denoted (copied from the IAM Roles page earlier).

COPY dev.public.category\_sample (catid, catgroup, catname, catdesc) FROM 's3://qwiklabs-gcp-04-6b8ab7514f17/redshift\_sample\_data.csv' IAM\_ROLE 'ENTER Redshift IAM ROLE ARN' FORMAT AS CSV DELIMITER ',' QUOTE '"' IGNOREHEADER 1 REGION AS 'us-east-1'

Copied!

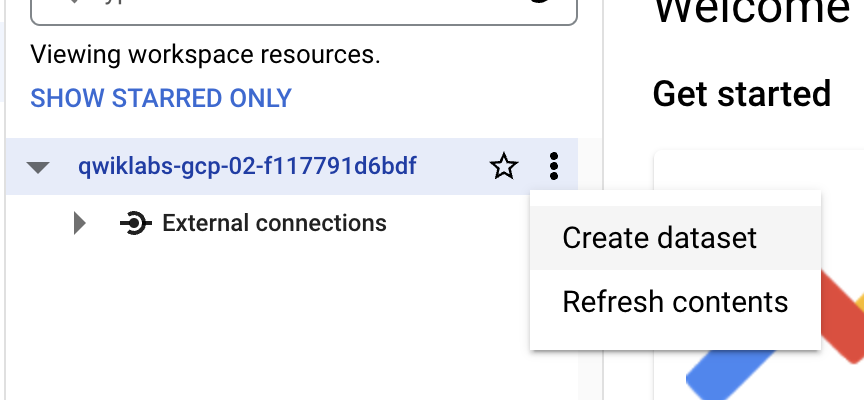


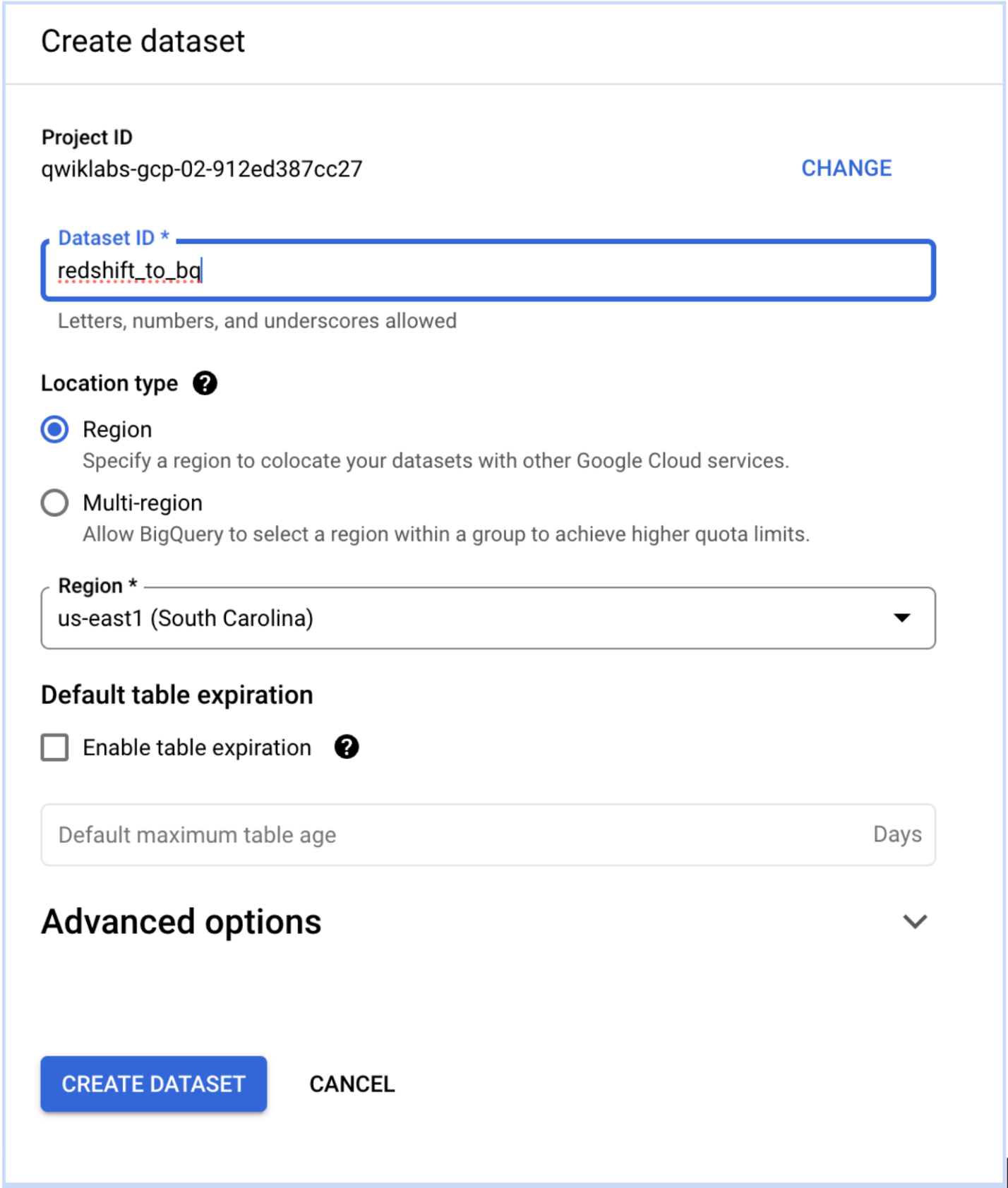
**Task 2: Create the BigQuery Dataset**

You are now ready to create a BigQuery dataset to store the migration data and create empty tables corresponding to each Redshift table to be migrated.

Log in to the Google Cloud Console using an incognito browser tab and the student credentials provided with this lab.

1. In Cloud Console, navigate to **BigQuery** > **select your GCP project**(qwiklabs-gcp-04-6b8ab7514f17).
2. Click **Create Dataset** then enter redshift\_to\_bq in the us-east1 region.





Click **Check my progress** to verify the objective.

Create the BigQuery Dataset

1. Navigate to **IAM & Admin** > **Roles**. Click **+ Create Role** at the top of the landing page.
2. Enter the following details for the Role:

|  |  |
| --- | --- |
| **Name** | **Value** |
| Title | BigQuery Transfer Role |
| ID | BQ\_DTS |

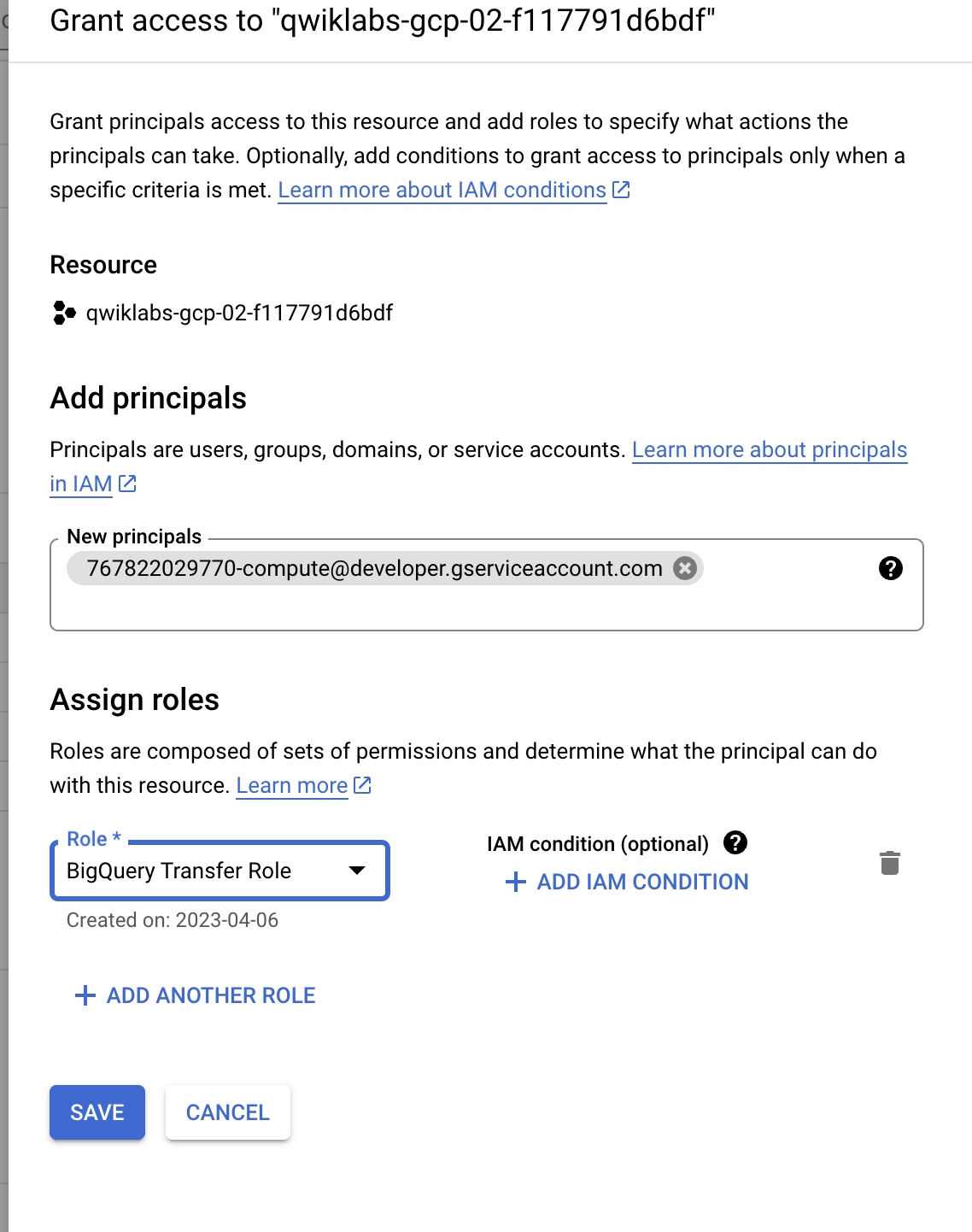
1. Click on the **+ ADD PERMISSIONS** button and add the following three permissions to the role. Use the search filter to do so.

* bigquery.datasets.get
* bigquery.datasets.update
* bigquery.transfers.update

Scroll down to the bottom of the page and click **CREATE**.

1. Navigate to **IAM & Admin** > **IAM**.
2. Click **+ GRANT ACCESS**. Enter Compute Engine default and select the principal that populates in the **New principals** field. Select BigQuery Transfer Role for the **Role** dropdown (the role will display in the **Custom** section).

Click **Save**.

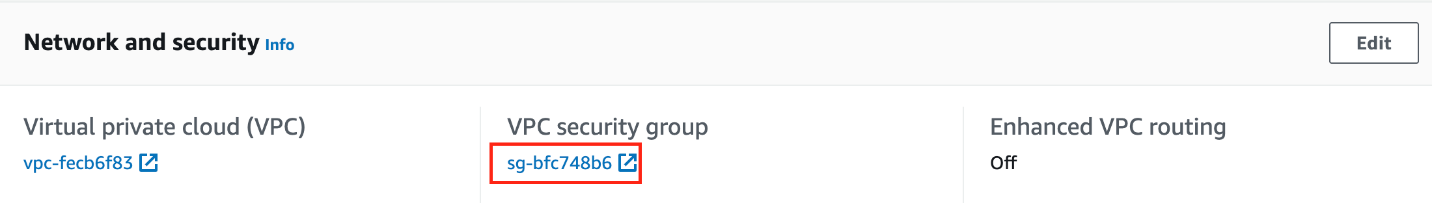


**Task 3: AWS Configuration**

Configure inbound rules for SQL clients

In order to allow the Data Transfer Service to access the Redshift cluster you must configure an allowlist for the IP addresses [reserved for Redshift migration to BigQuery](https://cloud.google.com/bigquery/docs/migration/redshift#regional-locations).

1. Navigate to **Redshift** > **Workgroup configuration** > **default-workgroup**.
2. On the resulting page, click on the **VPC security group** link found in the **Network and security** settings section.



1. On **Security Groups** page, click on **Security group ID**, then click on **Edit inbound rules**.
2. Click **Add rule** and set the following properties.

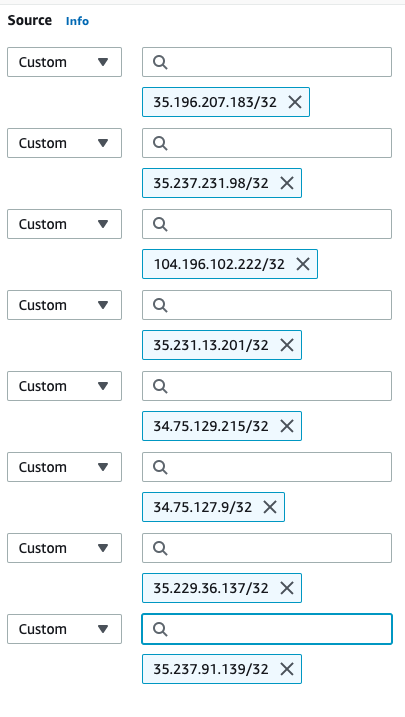
|  |  |
| --- | --- |
| **Name** | **Value** |
| Type | All Traffic |
| Source | Custom |
| Search Icon Column | 35.196.207.183/32 |

Repeat the same process for the following IP addresses.

* 35.237.231.98/32
* 104.196.102.222/32
* 35.231.13.201/32
* 34.75.129.215/32
* 34.75.127.9/32
* 35.229.36.137/32
* 35.237.91.139/32

**Note:**The IP addresses above correspond to the region us-east1 selected for the purposes of this lab. Google Cloud Platform reserves IP addresses for migration by region so the values will change if using another region.

The result of adding the inbound rules to the security group will looks similar to the following image.



Click **Save rules**.

**Important:**You must enable public access for the Redshift cluster in the context of this lab for the transfer to work properly. Ensure the below step is completed before proceeding!

1. Search for **Redshift** in the top search bar and navigate to the service.
2. Click on **Workgroup configuration** > **default-workgroup** > Under **Network and security** click on **Edit**.
3. Under Publicly accessible select the checkbox to enable public access to the instance.
4. Click **Save changes**.

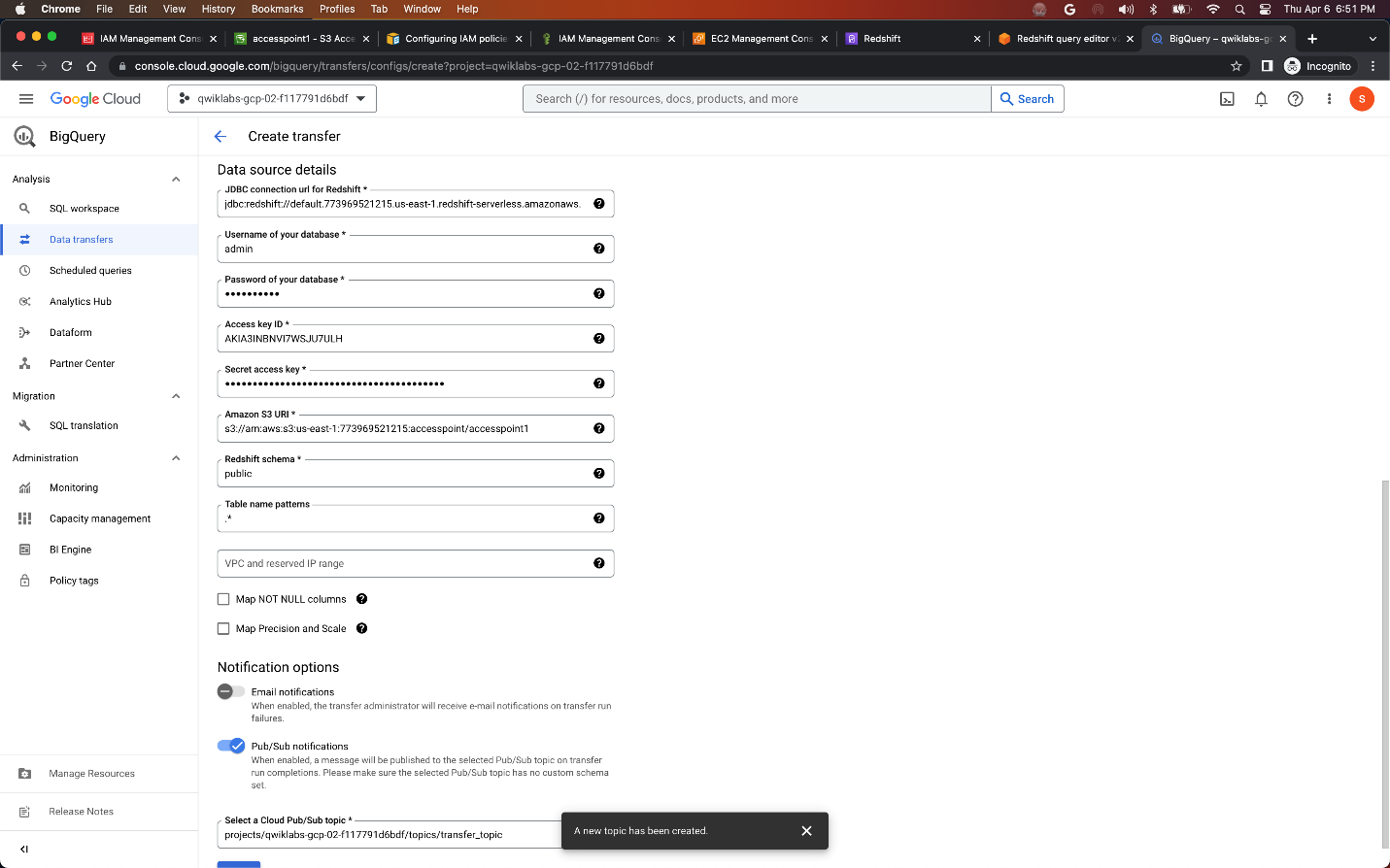
**Task 4: Migrate data using Data Transfer Service**

In this task, you will configure the Data Transfer Service and migrate data from the AWS Redshift instance.

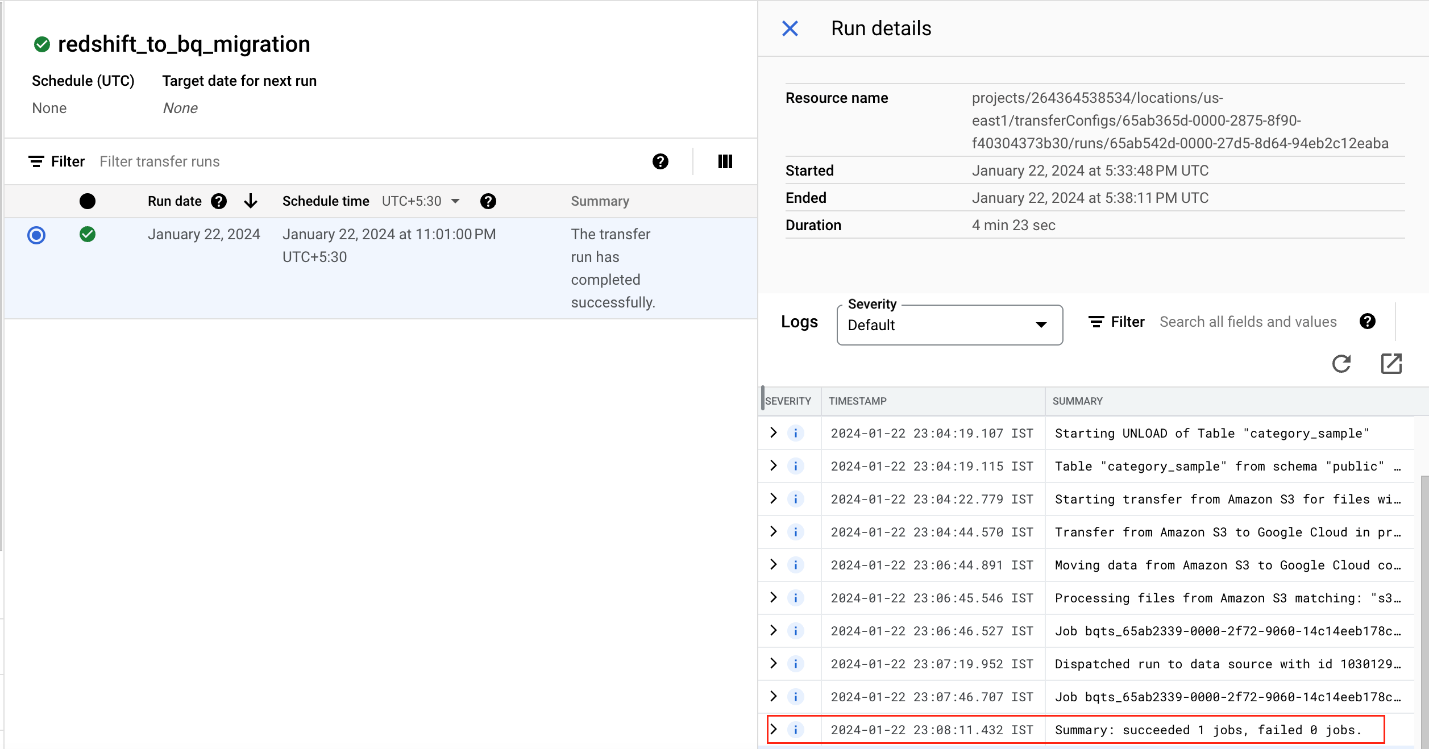
1. In the AWS Console, navigate to **Redshift** > **Workgroup configuration** > **default-workgroup**.
2. Copy the **JDBC URL** provided on the page. You will need to note this value down to refer to later.
3. Navigate to **Namespace configuration** > **default-namespace**.
4. Click on **Action** and select **Edit admin credentials** from dropdown.
5. Enter admin for **Admin user name**. For **Admin password** field, select **Manually add the admin password** and enter the **Admin user password** as Admin1234! and then click on **Save Changes**.
6. In the GCP Console, navigate to **BigQuery** > **Data transfers**.
7. Click **Create a transfer**.
8. Specify the following configuration for the data transfer.

|  |  |
| --- | --- |
| **Name** | **Value** |
| Source | Migration: Redshift |
| Display Name | redshift\_to\_bq\_migration |
| Dataset | redshift\_to\_bq |
| JDBC connection for Redshift URL | (Enter the JDBC URL copied from the Redshift page) |
| Username of your database | admin |
| Password of your database | Admin1234! |
| Access key ID | (AWS Access Key provided in the lab details pane) |
| Secret access key | (AWS Secret Key provided in the lab details pane) |
| Amazon S3 URI | s3://qwiklabs-gcp-04-6b8ab7514f17 |
| Redshift schema | public |
| Table name patterns | .\* |
| Service account | Leave the service account field unselected |
| VP and reserved IP range | (Leave blank) |
| Pub/Sub notifications | Toggle to **On**. Click **CREATE A TOPIC** button. Provide a Topic ID of choice then click **CREATE**. |

Click **SAVE**. When the **Choose an account** dialog appears select the student student-02-86f2f99c22b0@qwiklabs.net then click the **Allow** button on the next screen.



1. The Data Transfer Job will now execute. View the status of the transfer by clicking on the display name redshift\_to\_bq\_migration in the **Data transfers** section of **BigQuery**.
2. On the redshift\_to\_bq\_migration to status page, select the radio option next to the transfer job name to view the logs of the transfer. After a few minutes you should see the transfer completes successfully.



Click **Check my progress** to verify the objective.

Assessment Completed!

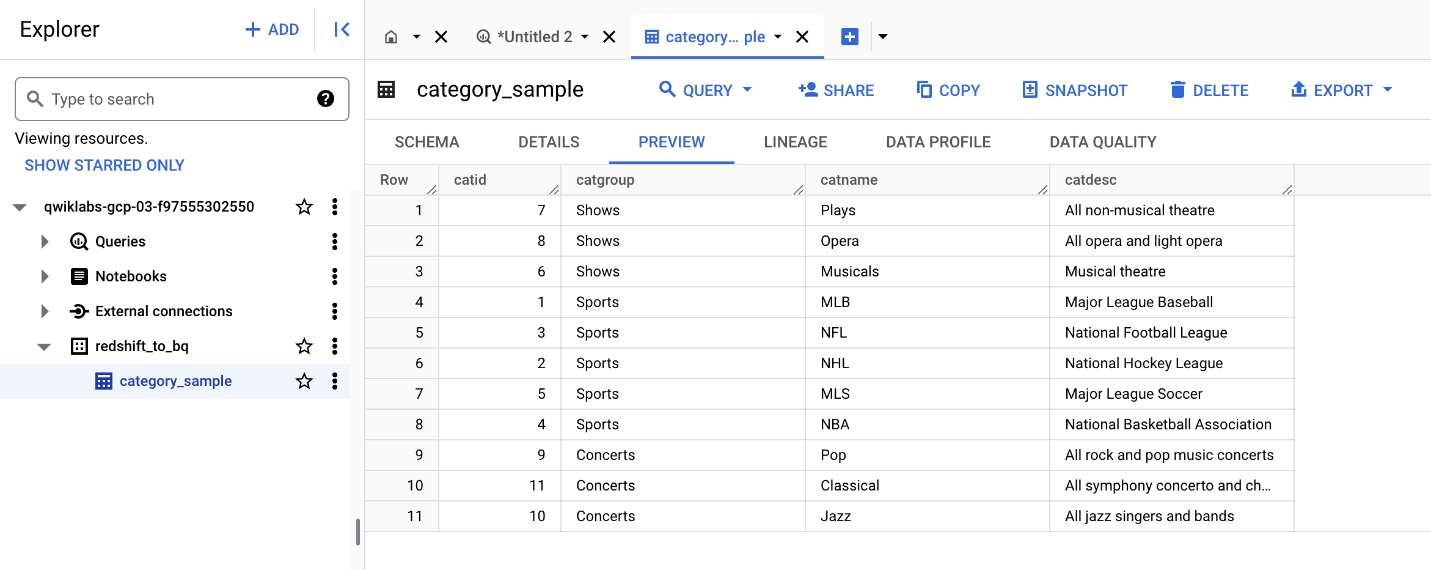
Migrate data using data transfer service

*Assessment Completed!*

**Task 5: Validate Migration**

Now that the Data Transfer Service migration has completed you can confirm that the data is present in the BigQuery Dataset created earlier.

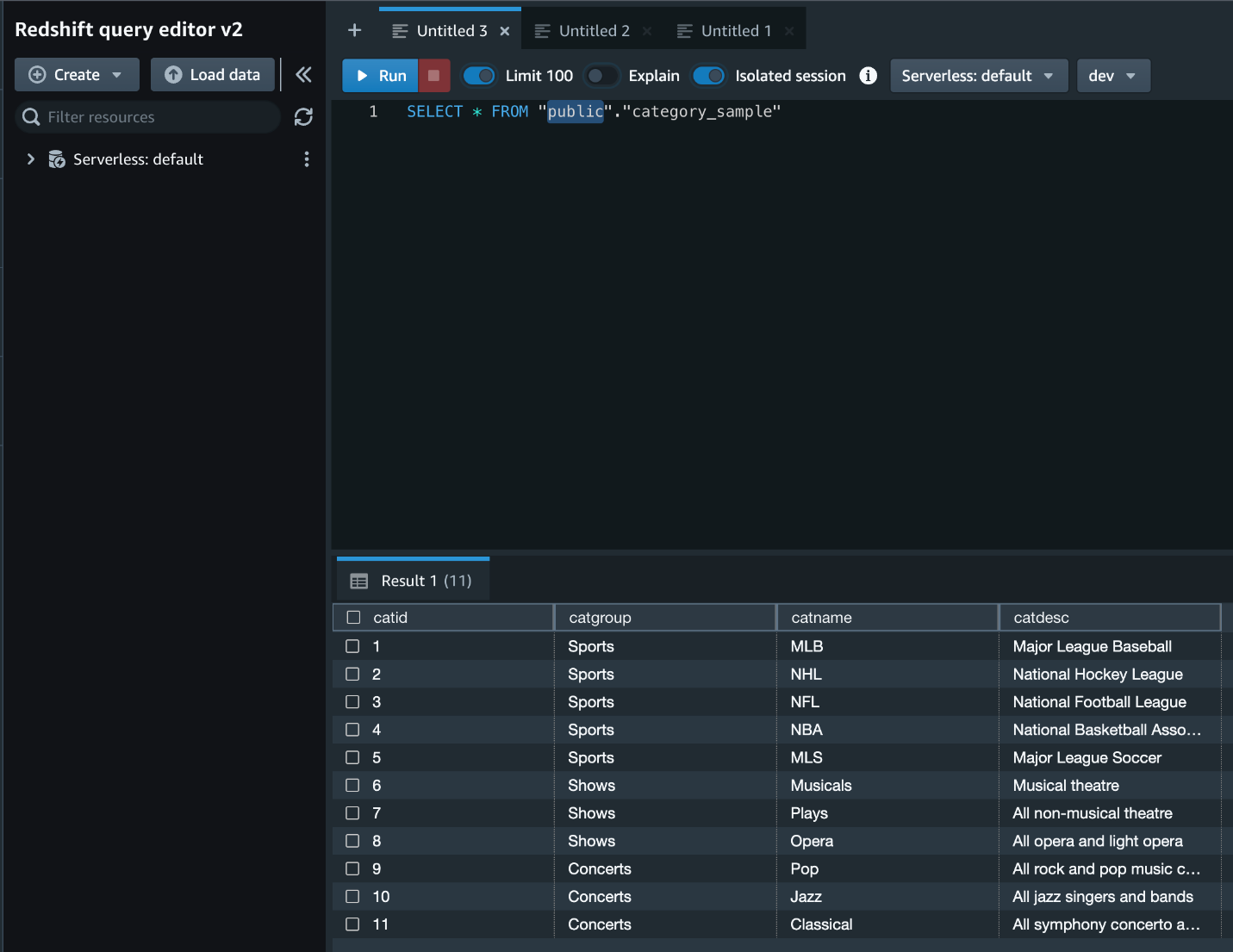
1. Navigate to **BigQuery** > **SQL BigQuery Studio**.
2. Expand the project qwiklabs-gcp-04-6b8ab7514f17 listed in the workspace as well as the Dataset redshift\_to\_bq then select the table category\_sample.
3. Click on the **PREVIEW** tab of the table to display the data it contains.



1. To confirm the data matches that of the AWS Redshift instance, in the AWS Console navigate to **Redshift**.
2. Click the **Query data** button on the top navigation.
3. In a new query tab, run the following SQL statement.

SELECT \* FROM "public"."category\_sample"

Copied!



The data returned from the Redshift query should match the BigQuery result.