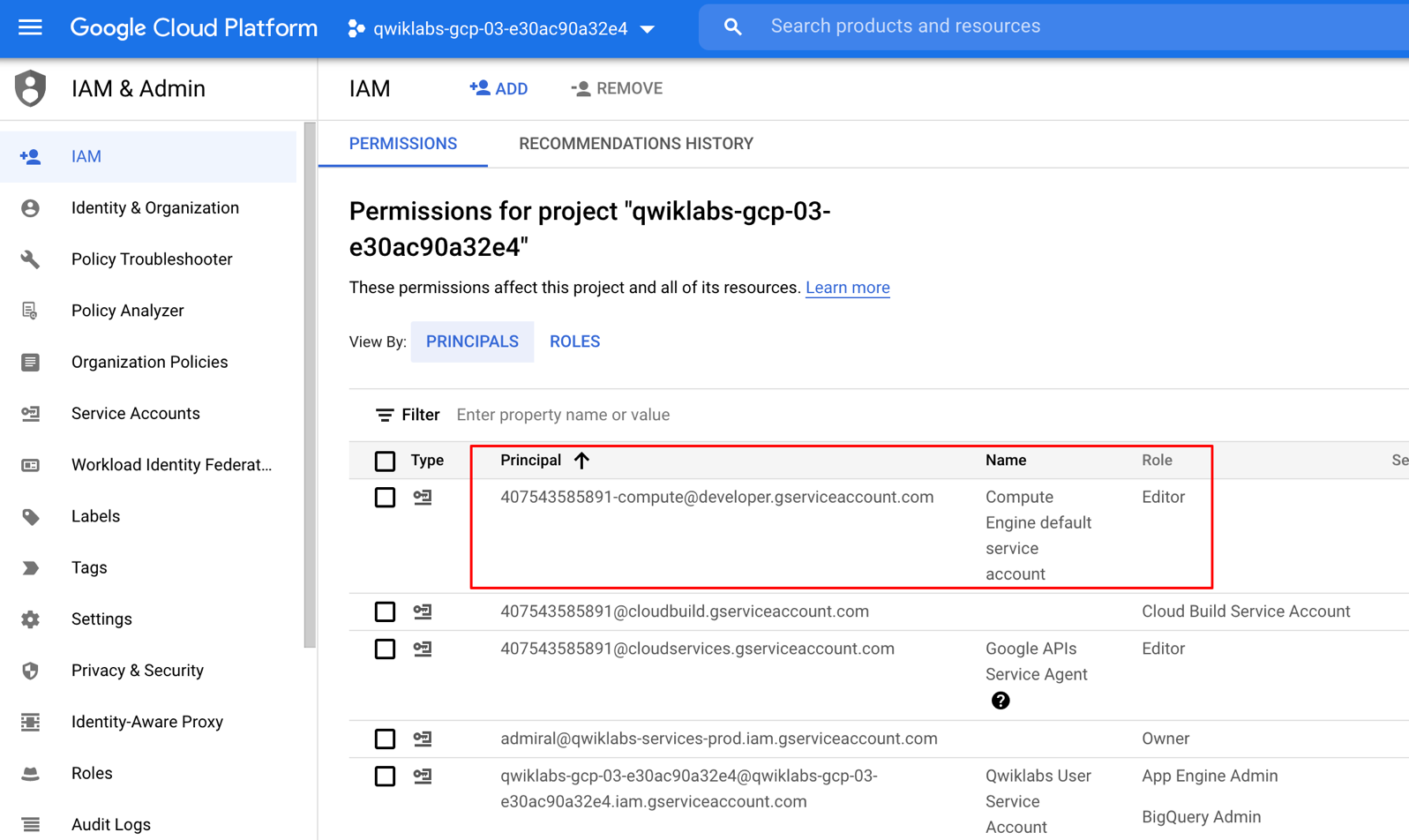
# **Building Batch Pipelines in Cloud Data Fusion**

# Check project permissions

Before you begin your work on Google Cloud, you need to ensure that your project has the correct permissions within Identity and Access Management (IAM).

1. In the Google Cloud console, on the **Navigation menu** (Navigation menu icon), click **IAM & Admin** > **IAM**.
2. Confirm that the default compute Service Account {project-number}-compute@developer.gserviceaccount.com is present and has the editor role assigned. The account prefix is the project number, which you can find on **Navigation menu** > **Home**.



If the account is not present in IAM or does not have the editor role, follow the steps below to assign the required role.

* In the Google Cloud console, on the **Navigation menu**, click **Home**.
* Copy the project number (e.g. 729328892908).
* On the **Navigation menu**, click **IAM & Admin** > **IAM**.
* At the top of the **IAM** page, click **Add**.
* For **New principals**, type:

{project-number}-compute@developer.gserviceaccount.com

Replace {project-number} with your project number.

* For **Role**, select **Project** (or Basic) > **Editor**. Click **Save**.

**Load the data**

Next, you will create a GCS bucket in your project and stage the CSF file. Cloud Data Fusion will later read data out of this storage bucket.

1. In Cloud Shell, execute the following commands to create a new bucket: and copy the relevant data into it:

export BUCKET=$GOOGLE\_CLOUD\_PROJECT

gsutil mb gs://$BUCKET

The created bucket name is your Project ID.

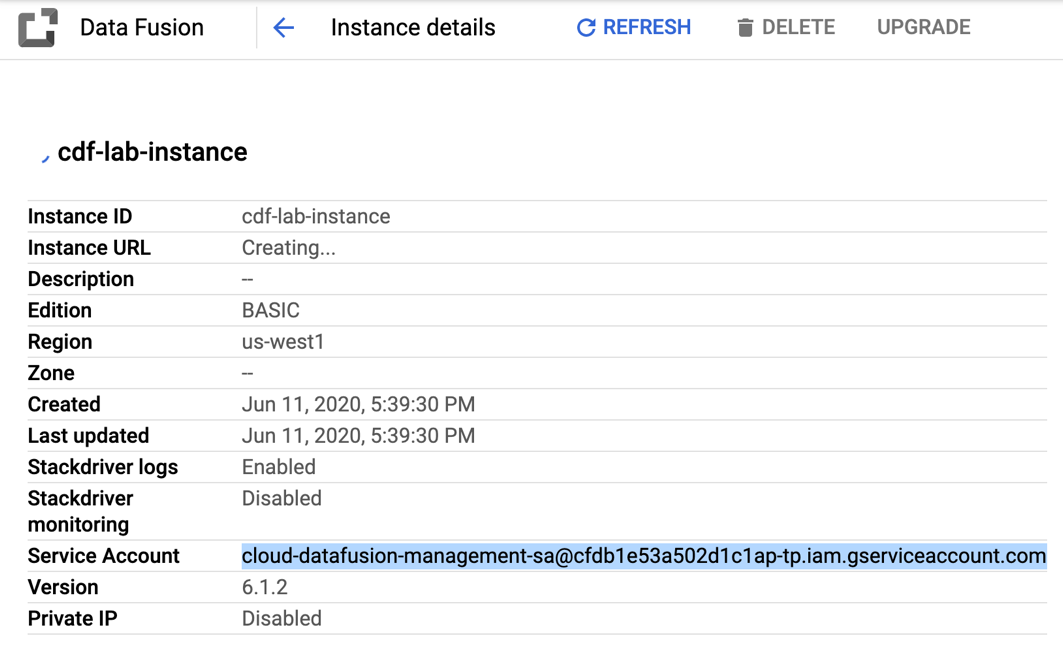
1. Run the command below to copy the data files (a CSV and an XML file) into your bucket:

gsutil cp gs://cloud-training/OCBL163/titanic.csv gs://$BUCKET

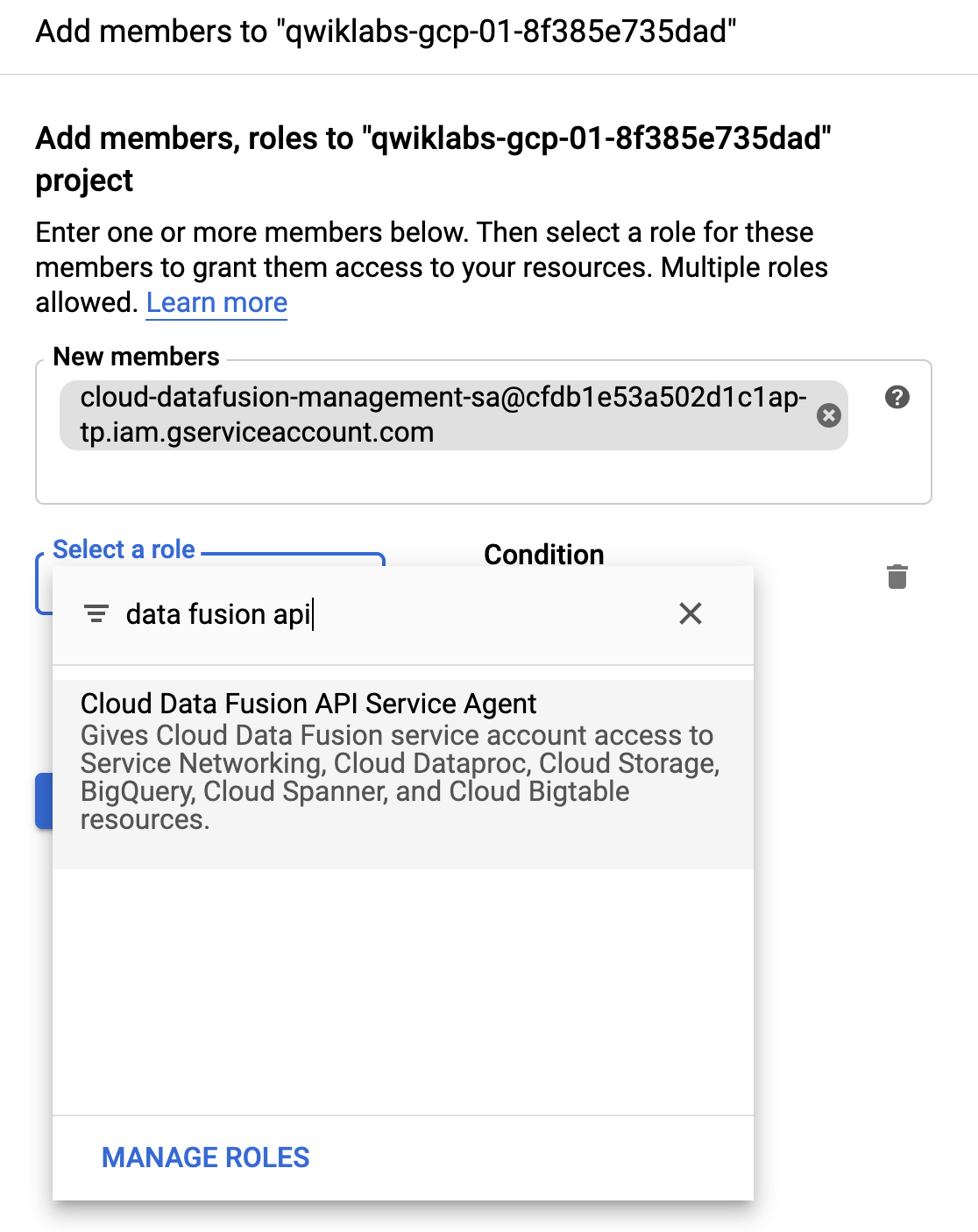
**Add necessary permissions for your Cloud Data Fusion instance**

Next, you will grant permissions to the service account associated with the instance, using the following steps.

1. In the Cloud Console, from the **Navigation menu** select **Data Fusion** > **Instances**. You should see a Cloud Data Fusion instance already setup and ready for use.
2. Click on the instance name. On the Instance details page copy the **Service Account** to your clipboard.



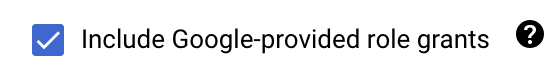
1. In the Console navigate to the **IAM & admin > IAM**.
2. On the IAM Permissions page, click **Add**.
3. In the **New Principals** field paste the service account.
4. Click into the Select a role filed and start typing **Cloud Data Fusion API Service Agent**, then select it.



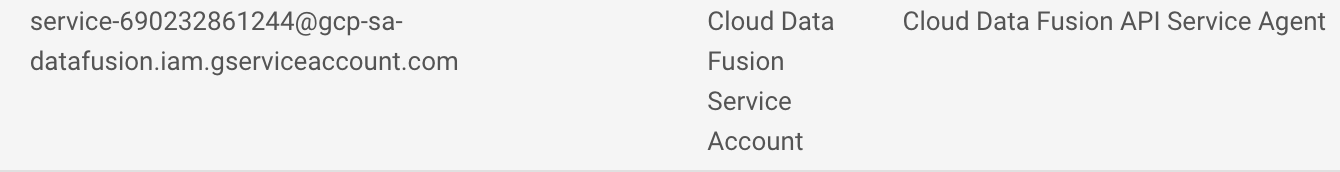
1. Click **Save**.

Grant service account user permission

1. In the Console navigate to the **IAM & admin > IAM**.
2. Select the checkbox next to **Include Google-provided role grants**



1. Scroll down the list to find the Google-managed Cloud Data Fusion service account that looks like service-{project-number}@gcp-sa-datafusion.iam.gserviceaccount.com and then copy it to your clipboard.

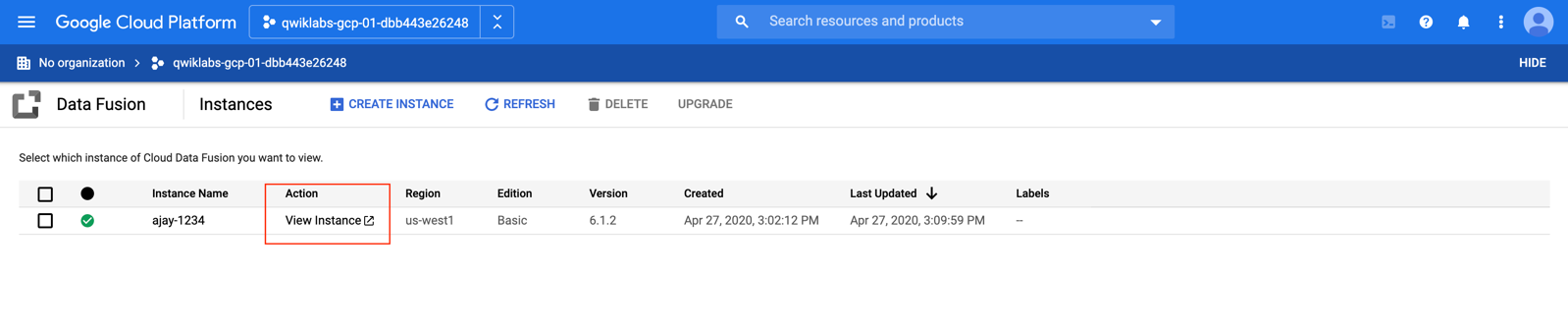


1. Next, navigate to the **IAM & admin > Service Accounts**.
2. Click on the default compute engine account that looks like {project-number}-compute@developer.gserviceaccount.com, and select the **PERMISSIONS** tab on the top navigation
3. Click on the **GRANT ACCESS** button
4. In the **New Principals** field paste the service account you copied earlier.
5. In the **Role** dropdown menu, select **Service Account User**
6. Click **Save**.

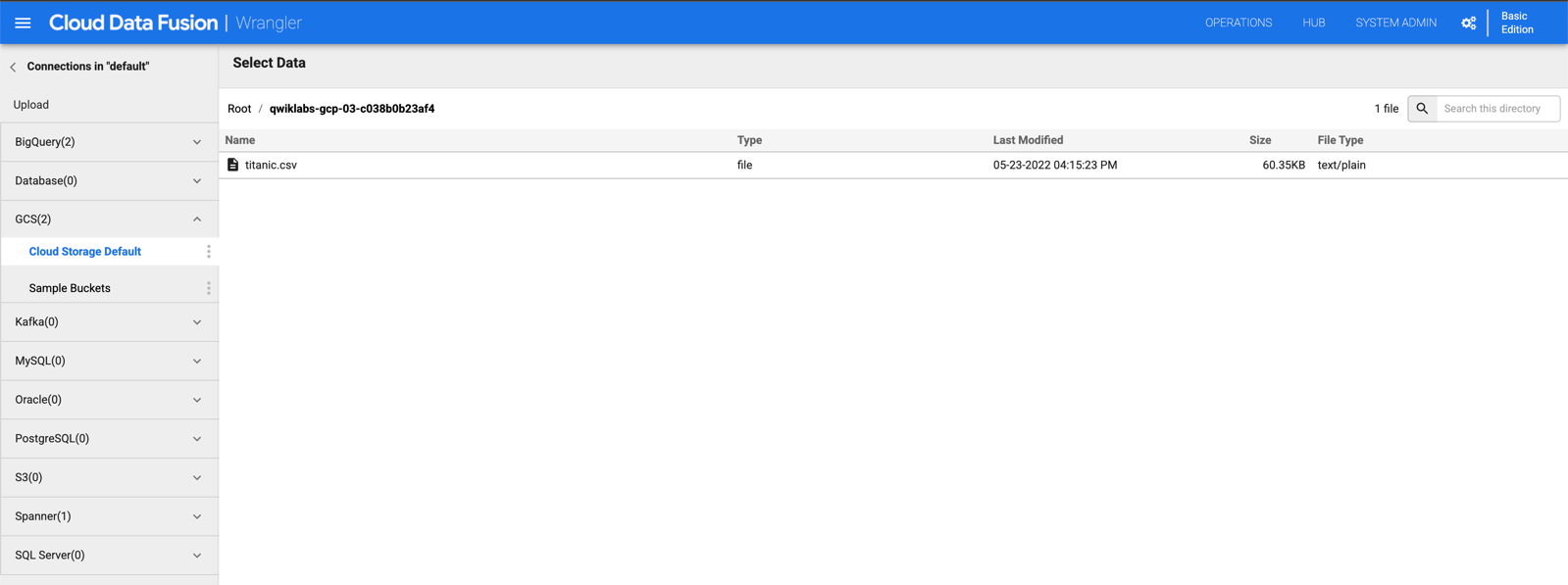
**Build a Batch Pipeline**

When working with data it’s always handy to be able to see what the raw data looks like so that we can use it as a starting point for our transformation. For this purpose you’ll be using Data Fusion’s Wrangler component for preparing and cleaning data. This data-first approach will allow us to quickly visualize our transformations and the real-time feedback ensures us that you’re on the right track.

1. In the Console return to **Navigation menu** > **Data Fusion > Instances**, then click the **View Instance** link next to your Data Fusion instance. Select your lab credentials to sign in. If prompted to take a tour of the service click on **No, Thanks**. You should now be in the Cloud Data Fusion UI.

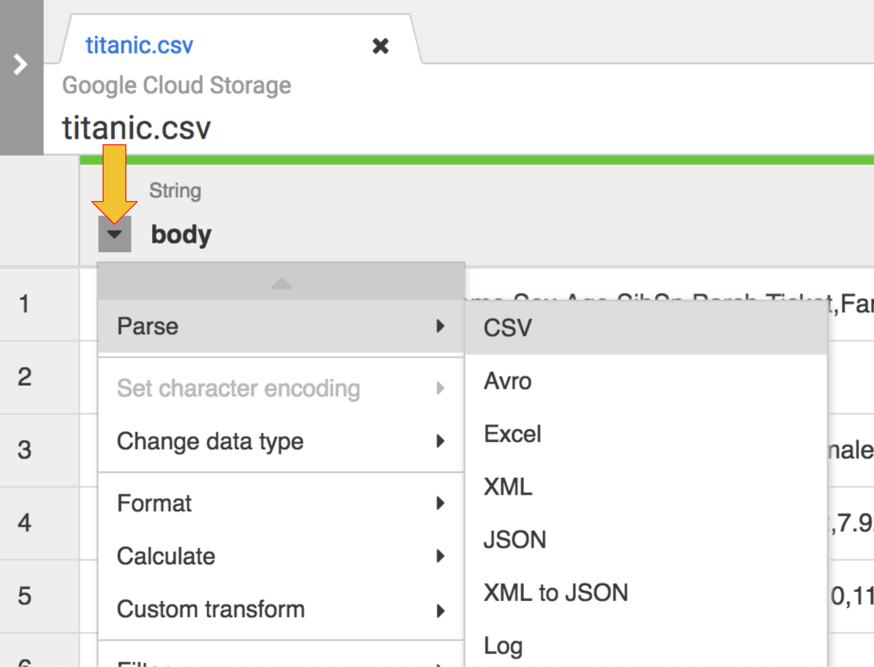


1. The Cloud Data Fusion web UI comes with its own navigation panel (on the left) to navigate to the page you need. In the Cloud Data UI, click on the **Navigation menu** on the top left to expose the navigation panel, then choose **Wrangler**.
2. On the left side is a panel with the pre-configured connections to your data, including the Cloud Storage connection.
3. Under **(GCS) Google Cloud Storage**, select **Cloud Storage Default**.
4. Click on the bucket corresponding to your project name.
5. Click on **titanic.csv**. The data is loaded into the Wrangler screen in row/column form.

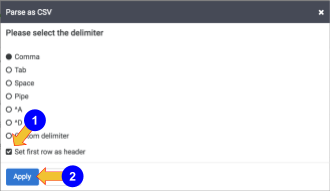


Once the file has been loaded into Wrangler you can start applying the data transformations iteratively.

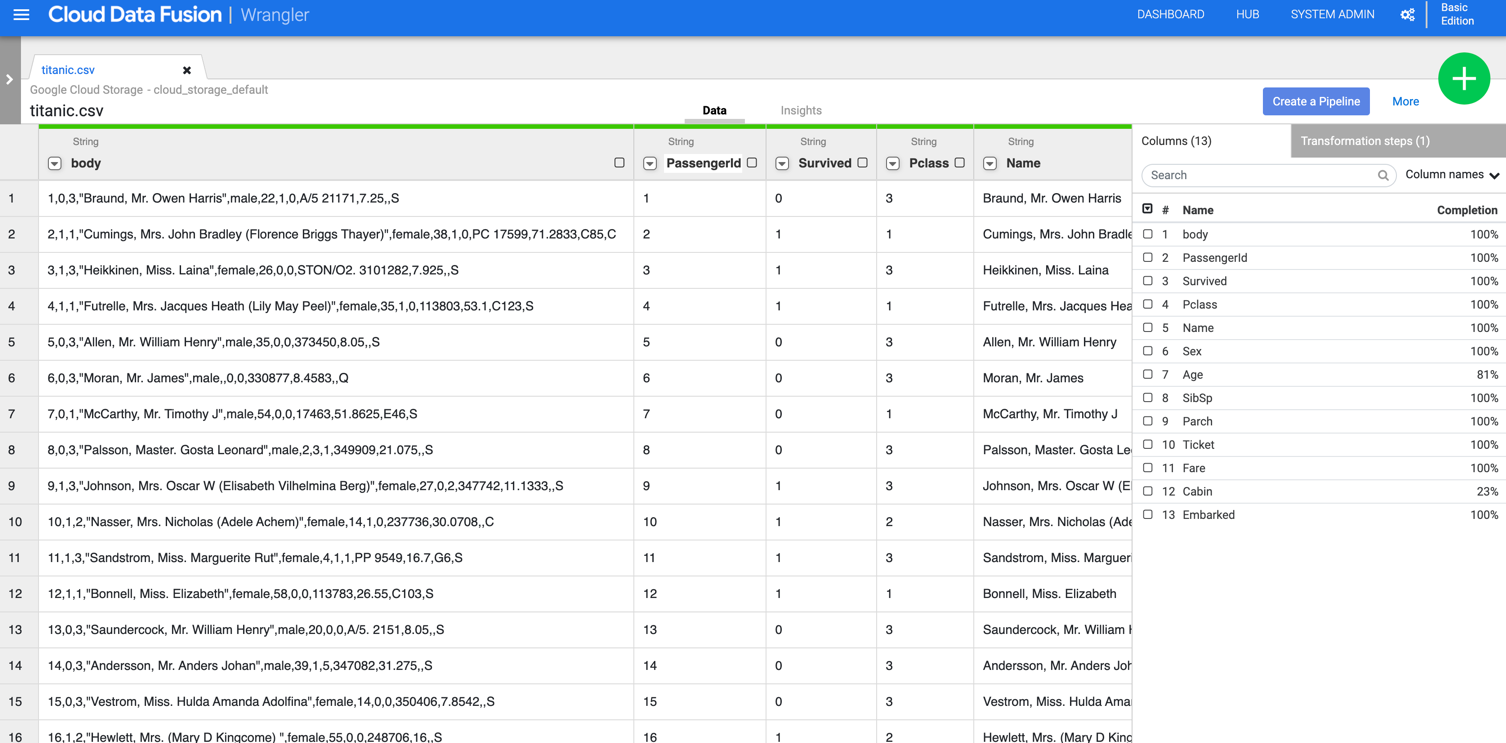
1. The first operation is to parse the raw CSV data into a tabular representation that is split into rows and columns. To do this, select the dropdown icon from the first column heading (body), and select the **Parse** menu item, and **CSV** from the submenu.



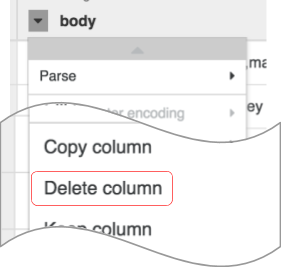
1. In the raw data you can see that the first row consists of column headings, so you’ll select the option to **Set first row as header** in the dialog box for **Parse as CSV** that is presented to us.



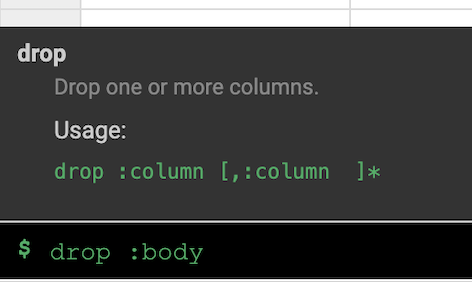
1. At this stage the raw data is parsed and you can see the columns generated by this operation (columns to the right of the *body* column). On the far right, you see the list of all column names.



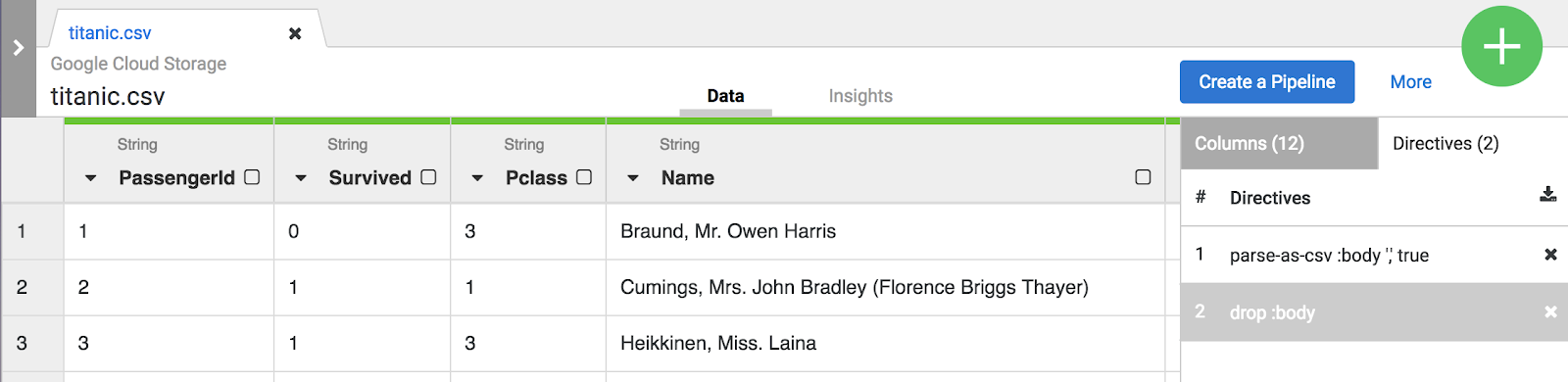
1. Now it’s time for some cleanup. You no longer need the **body** column that represents the raw CSV data, so remove it using the menu option: select the dropdown icon from the first column heading **body**, and select the **Delete column** menu item.



Note: To apply transformations, you can also use the command line interface (CLI). The CLI is the black bar at the bottom of the screen (with the green **$** prompt). As you start typing commands the autofill feature kicks in and presents you with a matching option. For example, to drop the body column, you could have alternatively used the directive: **drop :body**



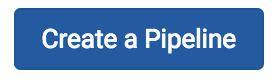
1. Click on the **Transformation steps** tab on the far right of your Wrangler UI. You will see the two transformations you have applied so far.



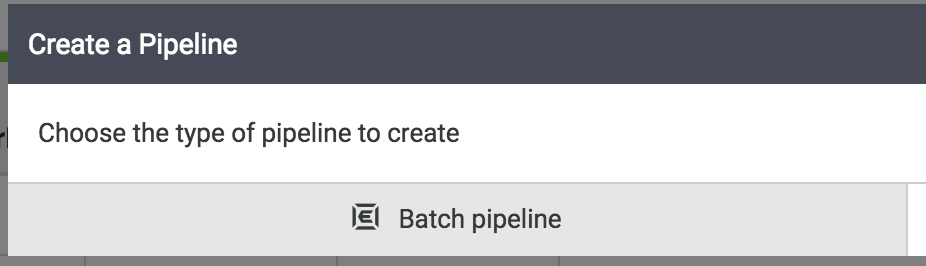
Both the menu selections and the CLI create directives that are visible on the **Transformation steps** tab on the right of the screen. Directives are individual transformations that are collectively referred to as a recipe.

For the purpose of this lab, the two transformations, or recipe, are sufficient to create the ETL pipeline. The next step is to bring this recipe into a pipeline building step where the recipe represents the T in ETL.

1. Click on the **Create a Pipeline** button to jump into the next section to create a pipeline, where you’ll see how the ETL pipeline comes together.



When presented with the next dialog select **Batch pipeline** to continue.



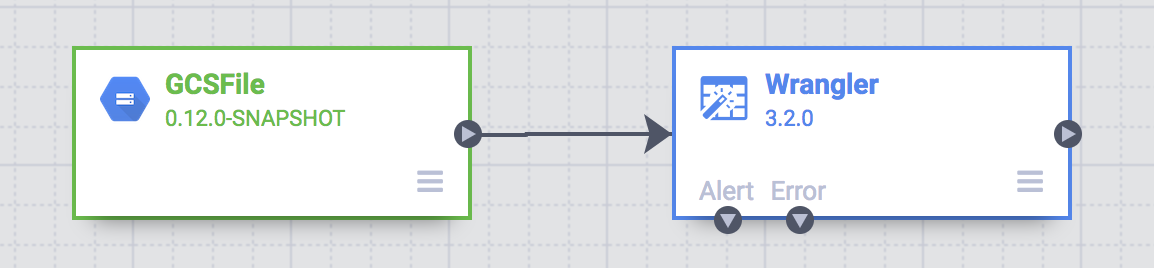
A batch pipeline can be run interactively or scheduled to run as frequently as every 5 minutes or as little as once a year.

**Pipeline Studio**

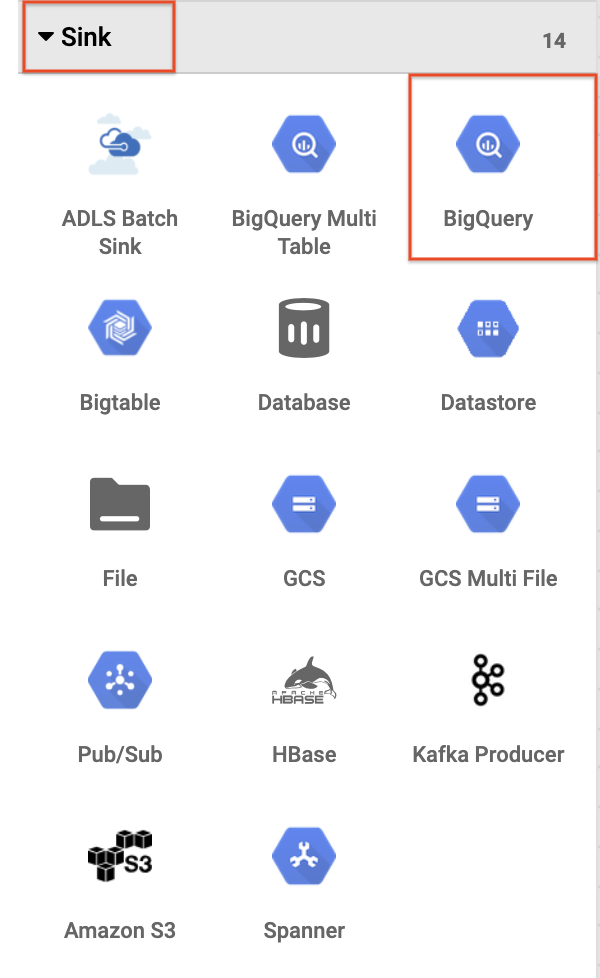
The rest of the pipeline building tasks will take in the pipeline studio, the UI that lets you compose data pipelines visually. You should now see the primary building blocks of your ETL pipeline in the studio. At this point you will see 2 nodes in your pipeline: the **GCS File** plugin that will read the CSV file from Google Cloud Storage and the **Wrangler** plugin that contains the recipe with the transformations.

A node in a pipeline is an object that is connected in a sequence to produce a Directed Acyclic Graph. E.g. Source, Sink, Transform, Action, etc.

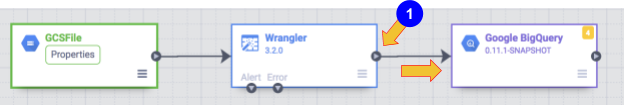
These two plugins (nodes) represent the E and the T in your ETL pipeline. To complete this pipeline add the BigQuery sink, the L portion of our ETL.



1. To add the BigQuery sink to the pipeline navigate to the **Sink** section on the left panel and click on the **BigQuery** icon to place it on the canvas.



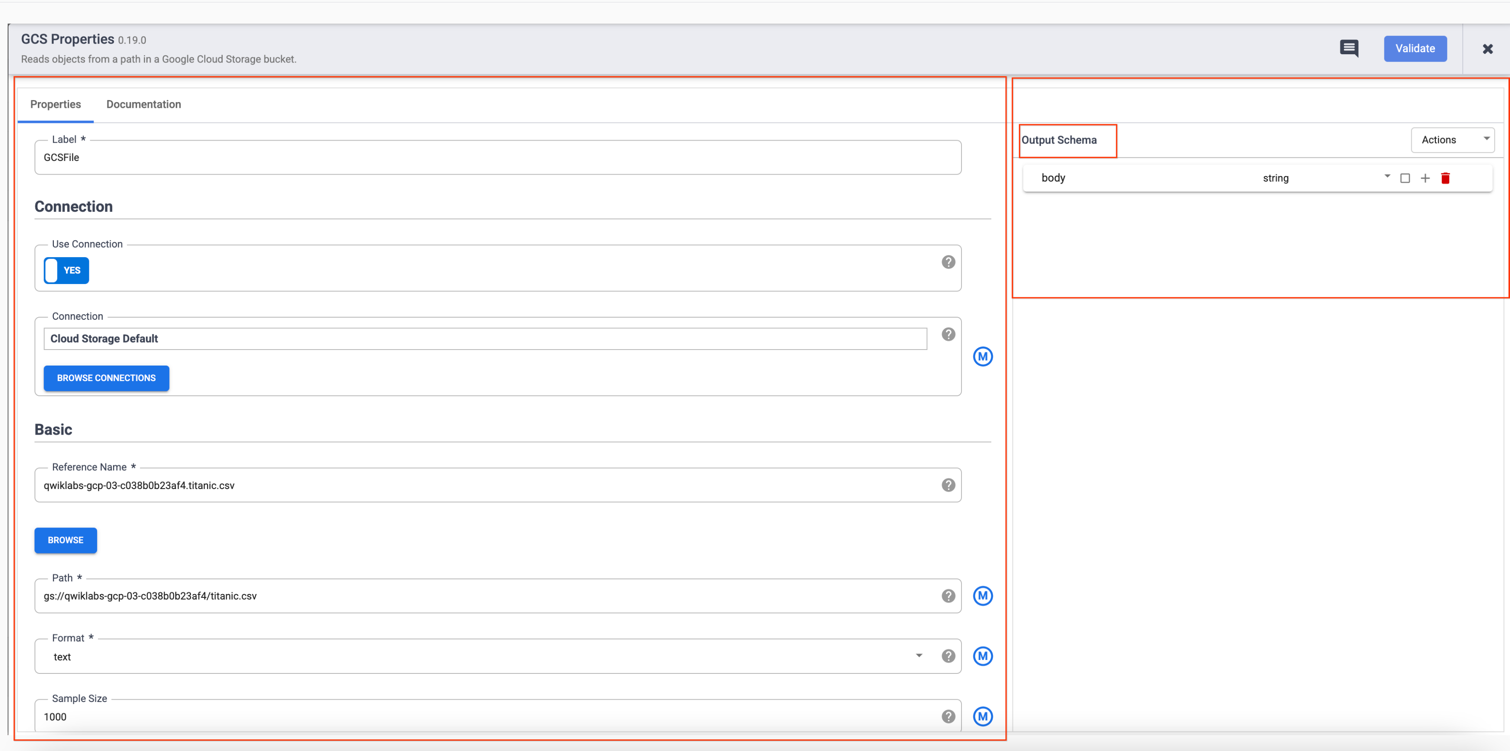
1. Once the BigQuery sink has been placed on the canvas, connect the Wrangler node with the BigQuery node. Do this by dragging the arrow from the Wrangler node to connect to the BigQuery node as illustrated below. All that’s left to do now is to specify some configuration options so that you can write the data to the dataset you want.



**Configure Pipeline**

Now it’s time to configure the pipeline. You can do this by opening each of the nodes' properties to verify its settings and/or make any additional changes.

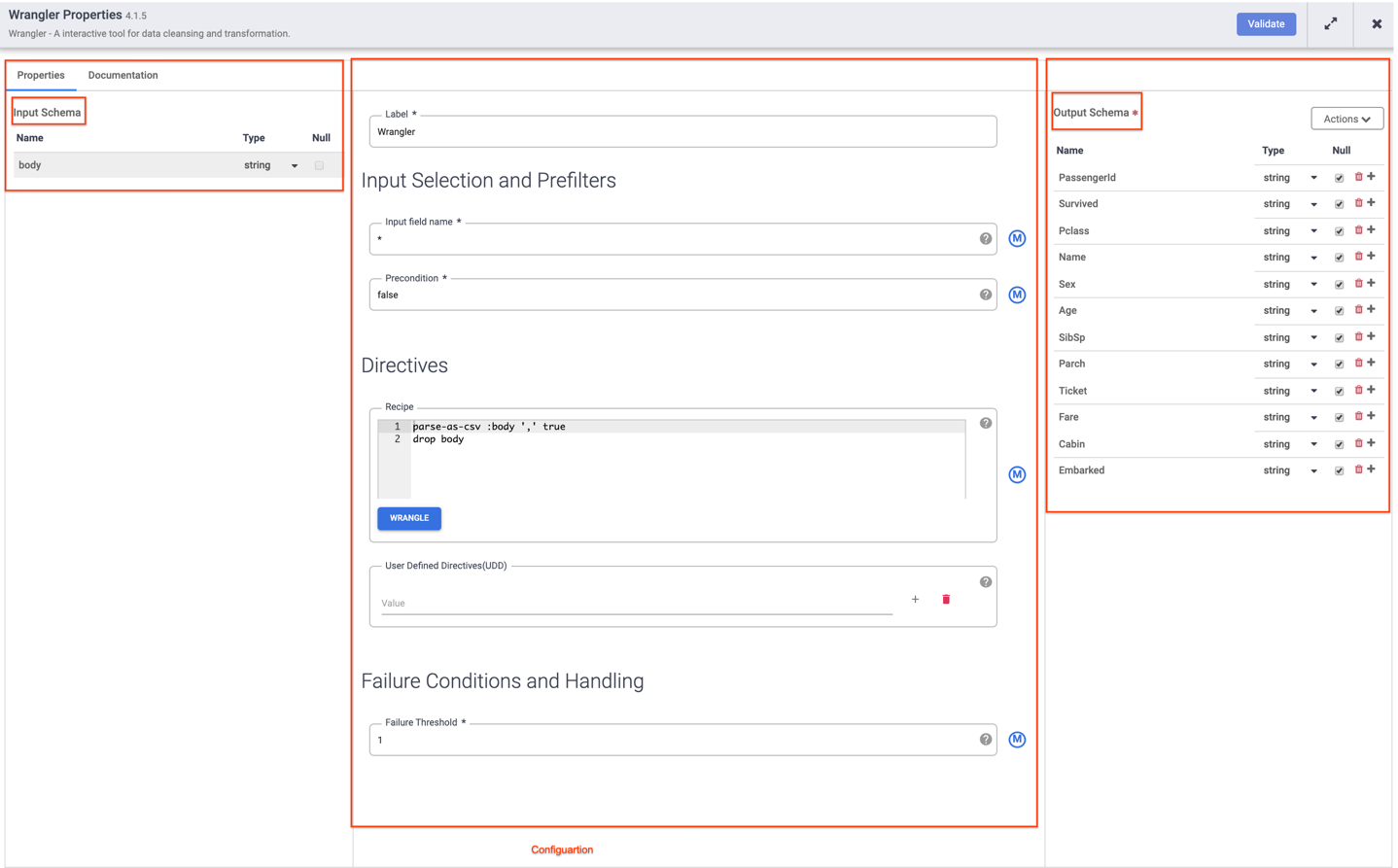
1. Hover your mouse over the **GCS** node and a **Properties** button will be displayed. Click on this button to open up the configuration settings.



Each plugin has a few mandatory fields that must be present and they are marked with a ( \* ). Depending on the plugin you are using you may see an **Input** **Schema**, on the left, a **Configuration** section in the middle, and an **Output** **Schema** on the right. You will notice that Sink plugins don’t have an output schema and Source plugins don’t have an input schema. Both Sink and Source plugins will have a mandatory **Reference Name** field as well to identify the data source/sink for lineage.

Each plugin will have a **Label** field. This is the label of the node you see on the canvas where your pipeline is displayed.

1. Click on **X** on the top right of the Properties box to close it.
2. Next, hover over your **Wrangler** node and click on **Properties**.



Plugins like Wrangler that contain an input schema. These are the fields that are being passed in to the plugin to be processed. Once they are processed by the plugin outgoing data may be sent out in the output schema to the next node in the pipeline, or in the case of a sink written to a dataset.

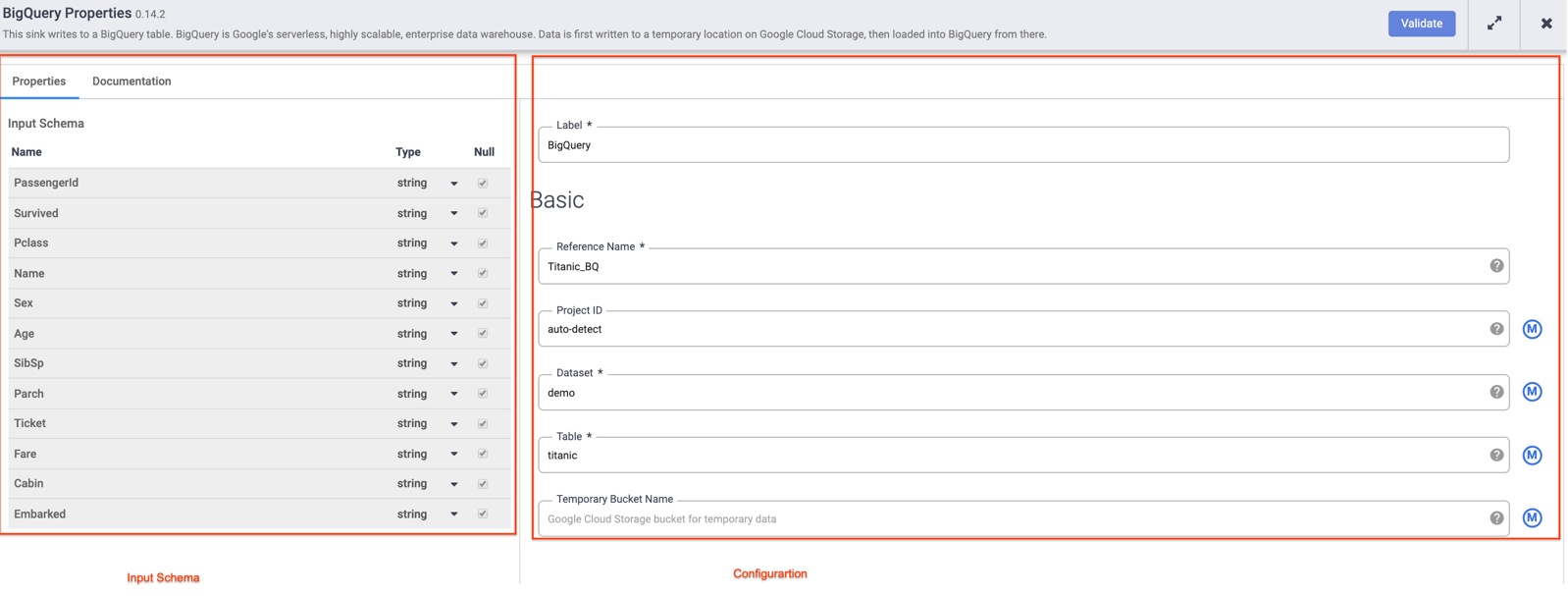
1. Click on **X** on the top right of the Properties box to close it.
2. Hover your mouse over your BigQuery node, click on **Properties** and enter the following configuration settings:

a. For **Reference Name**, enter Titanic\_BQ

b. For **Dataset**, enter demo

c. For **Table**, enter titanic

d. Click on **X** on the top right of the Properties box to close it.



**Test Pipeline**

All that’s left to do now is to test your pipeline to see that it works as expected, but before you do that make sure to name and save your draft so you won’t lose any of your work.

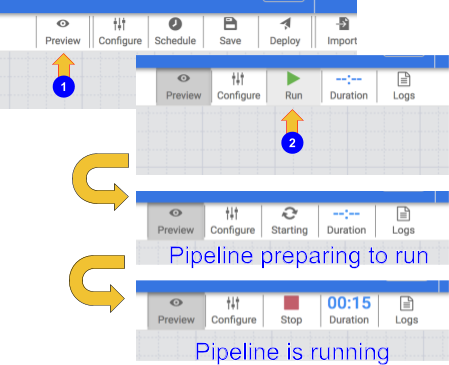
1. Now click **Save** from the upper right corner menu. You will be prompted to give **Name** and add a **description** to the pipeline.

a. Enter ETL-batch-pipeline as the name of your **pipeline**.

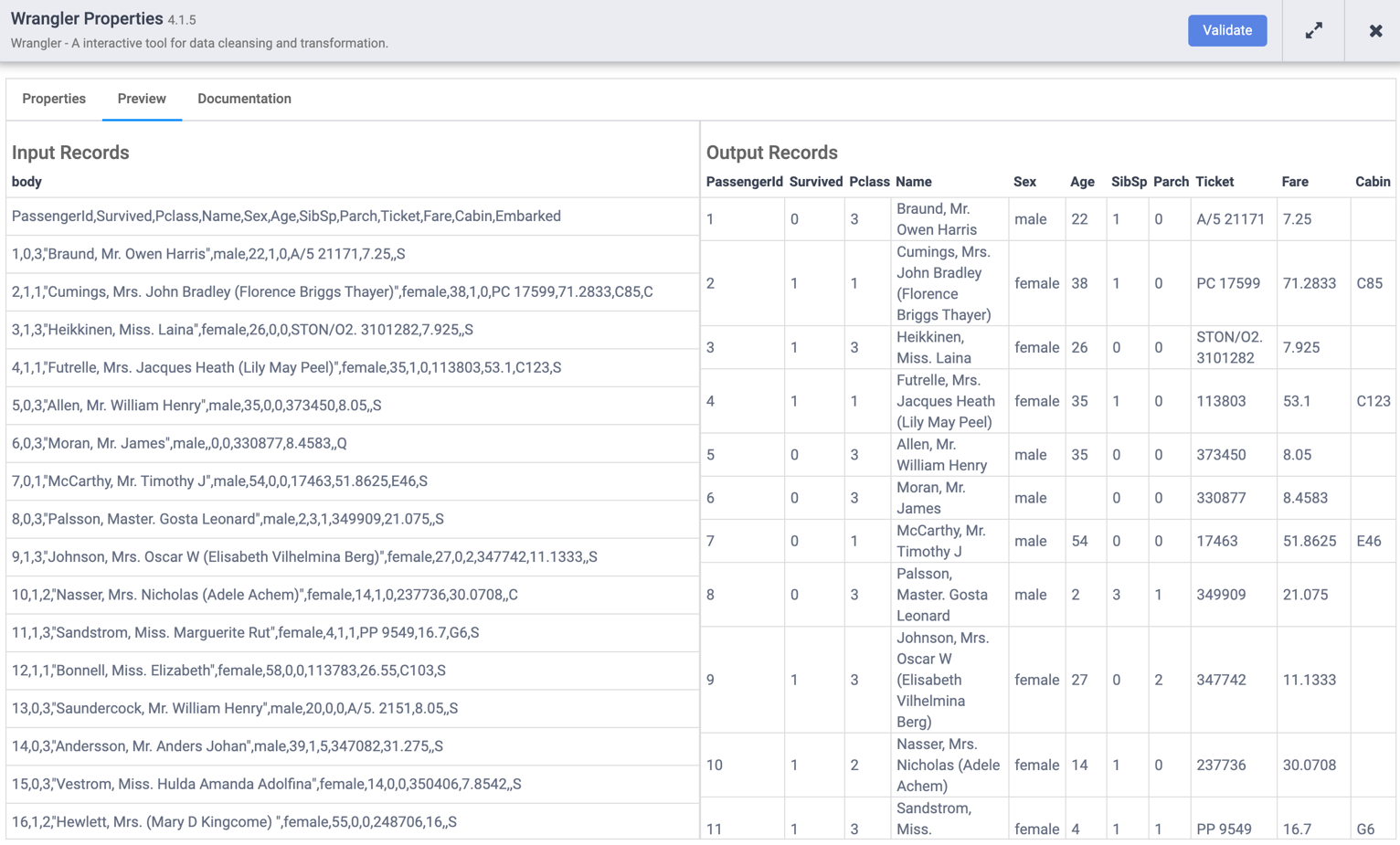
b. Enter ETL pipeline to parse CSV, transform and write output to BigQuery for the **description**.

c. Click on **Save**.

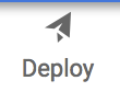
1. To test your pipeline click on the **Preview** icon. The button bar will now show a run icon that you can click to run the pipeline in preview mode.
2. Click on the **Run** icon. While the pipeline is running in preview mode no data is actually written to the BigQuery table, but you will be able to confirm that data is being read properly and that it will be written as expected once the pipeline is deployed. The preview button is a toggle so be sure to click it again to get out of preview mode when done.



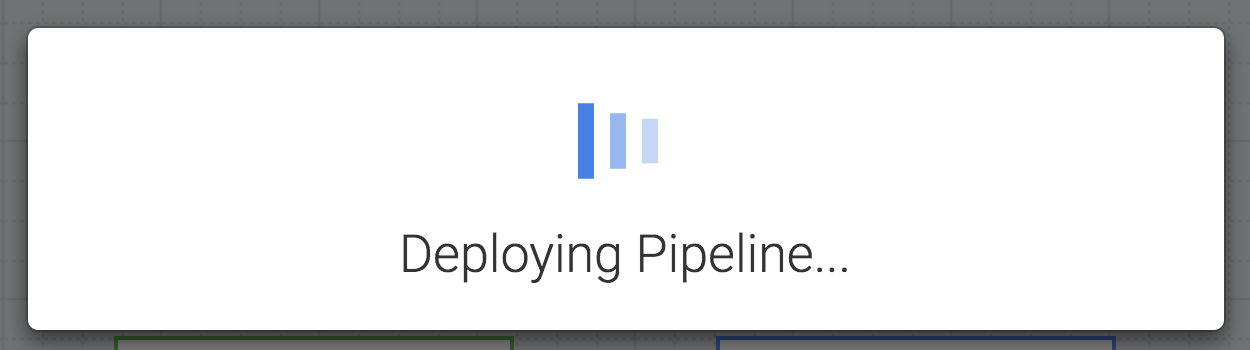
1. When your pipeline completes running, click on the **Wrangler** node. If all went well you should see the raw data that came in from the input, the node to the left, and the parsed records that will be emitted as output, to the node on the right.



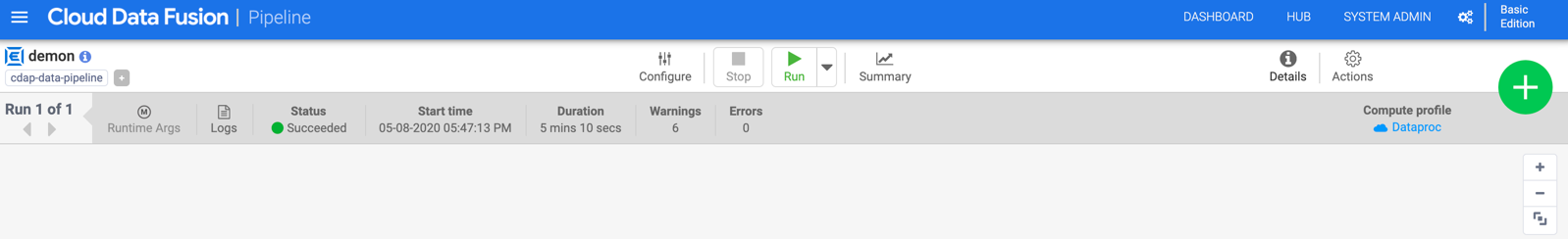
Each node that operates on data should show you similar output. This is a good way to prove your work to make sure you are on the right track before deploying your pipeline. If you encounter any errors you can easily fix it while in draft mode.

1. Click on the **Preview** icon again, this time to toggle out of Preview mode.
2. If everything looks good so far, you can proceed to deploy the pipeline. Click on the **Deploy** icon on the top right  to deploy the pipeline.

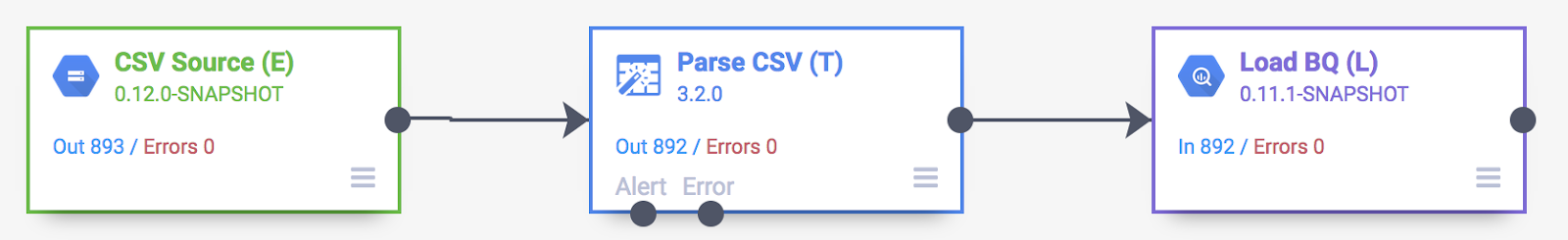
You will see a confirmation dialog that your pipeline is being deployed:



1. Once your pipeline has successfully deployed, you’re now ready to run your ETL pipeline and load some data into BigQuery.
2. Click on the **Run** icon to execute the ETL job.
3. When done you should see pipeline status changes to **Succeeded** indicating that the pipeline ran successfully.



1. As data is processed by the pipeline you will see metrics being emitted by each node in the pipeline indicating how many records have been processed. In the parse operation it displays 892 records whereas in the source there were 893 records, so what happened? The parse operation took the first row and consumed it to set the column headings, so the remaining 892 records are what was left to process.



**View the results**

The pipeline writes output into a BigQuery table. You can verify that using the following steps.

1. In a new tab, [open the BigQuery UI in the Cloud Console](https://console.cloud.google.com/bigquery) or right-click on the Console tab and select **Duplicate**, then use the **Navigation menu** to select **BigQuery**. When prompted click **Done**.
2. In the left pane, in the **Explorer** section, click your Project ID (it will start with qwiklabs).
3. Under the **demo** dataset in your project, click the **titanic** table and click **Compose New Query**, then run a simple query, such as:

SELECT \* FROM `demo.titanic` LIMIT 10

