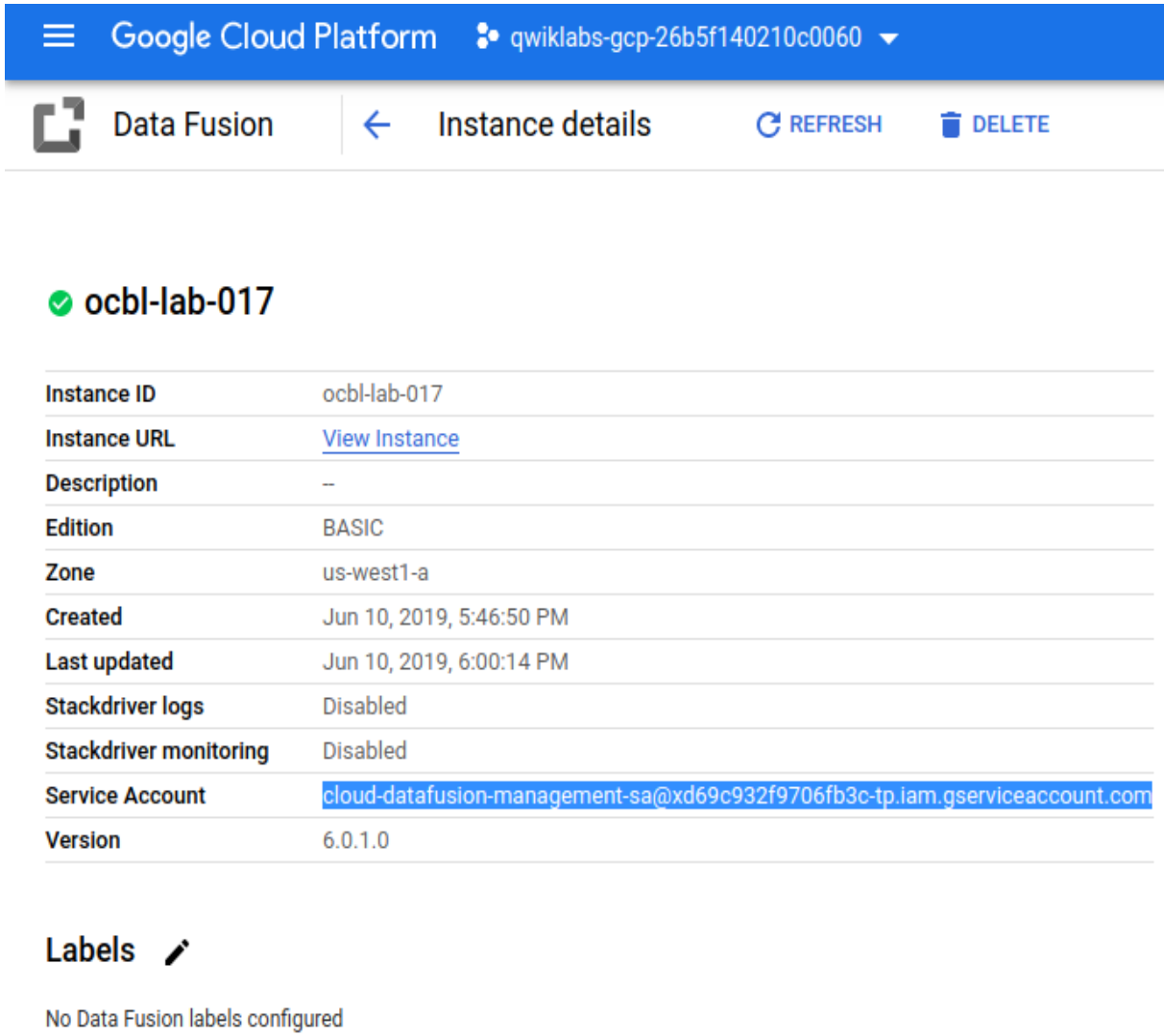
Lab: Building and executing a pipeline graph in Cloud Data Fusion

Configuration(create data fusion instance)

- ✓ **Activate Google Cloud Shell**
  gcloud auth list
  gcloud config list project
  # disable datafusion
  gcloud services disable datafusion.googleapis.com
- ✓ **Check project permissions**
- ✓ **Creating a Cloud Data Fusion instance**
  Enable Cloud Data Fusion API
- ✓ **Data Fusion > Create an Instance**.
  name: what you want
  Edition type : Basic
  Authorization : Grant Permission

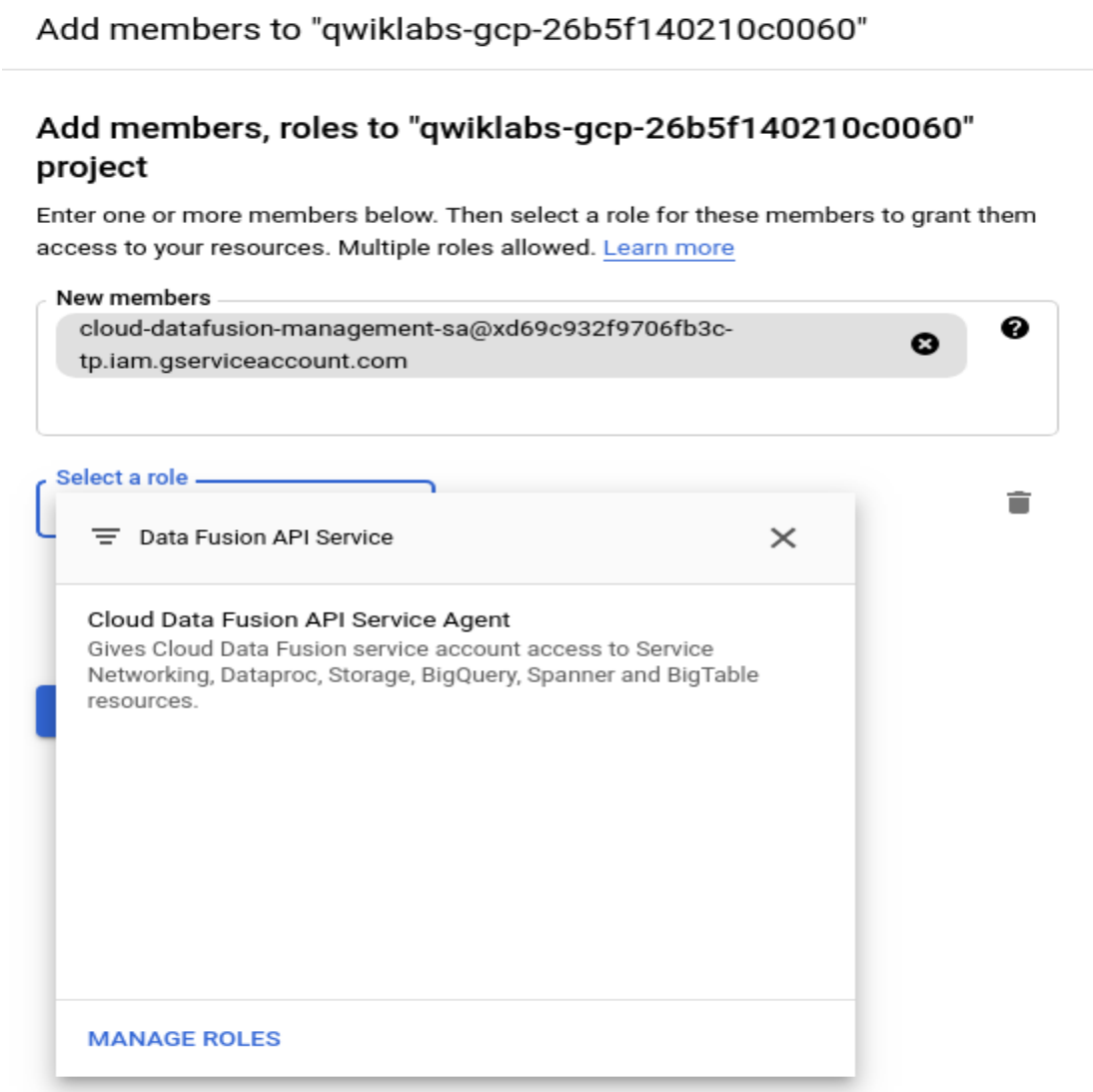
It need 15 minutes to complete

- ✓ Copy the service account to your clipboard



Configuration(add role)

- ✓ **IAM & Admin > IAM**
- ✓ On the IAM Permissions page, add the service account you copied earlier as a new member and grant the **Cloud Data Fusion API Service Agent** role, by clicking the **Add** button.

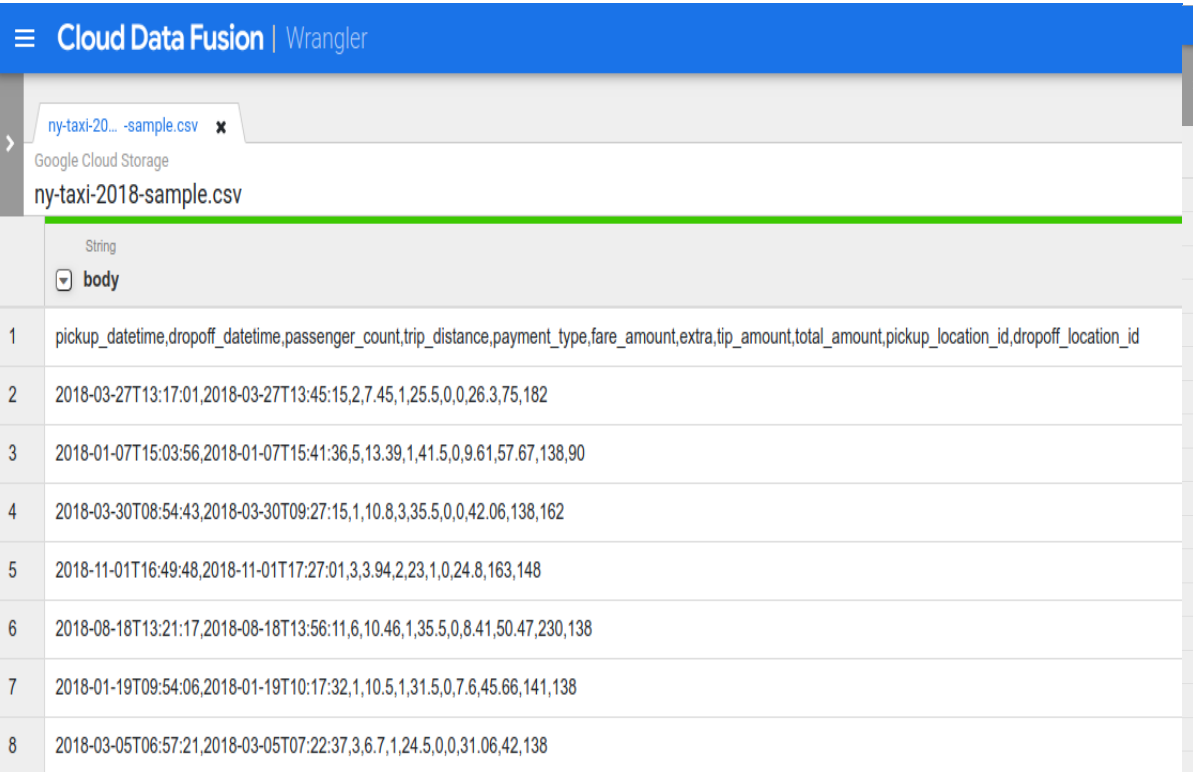


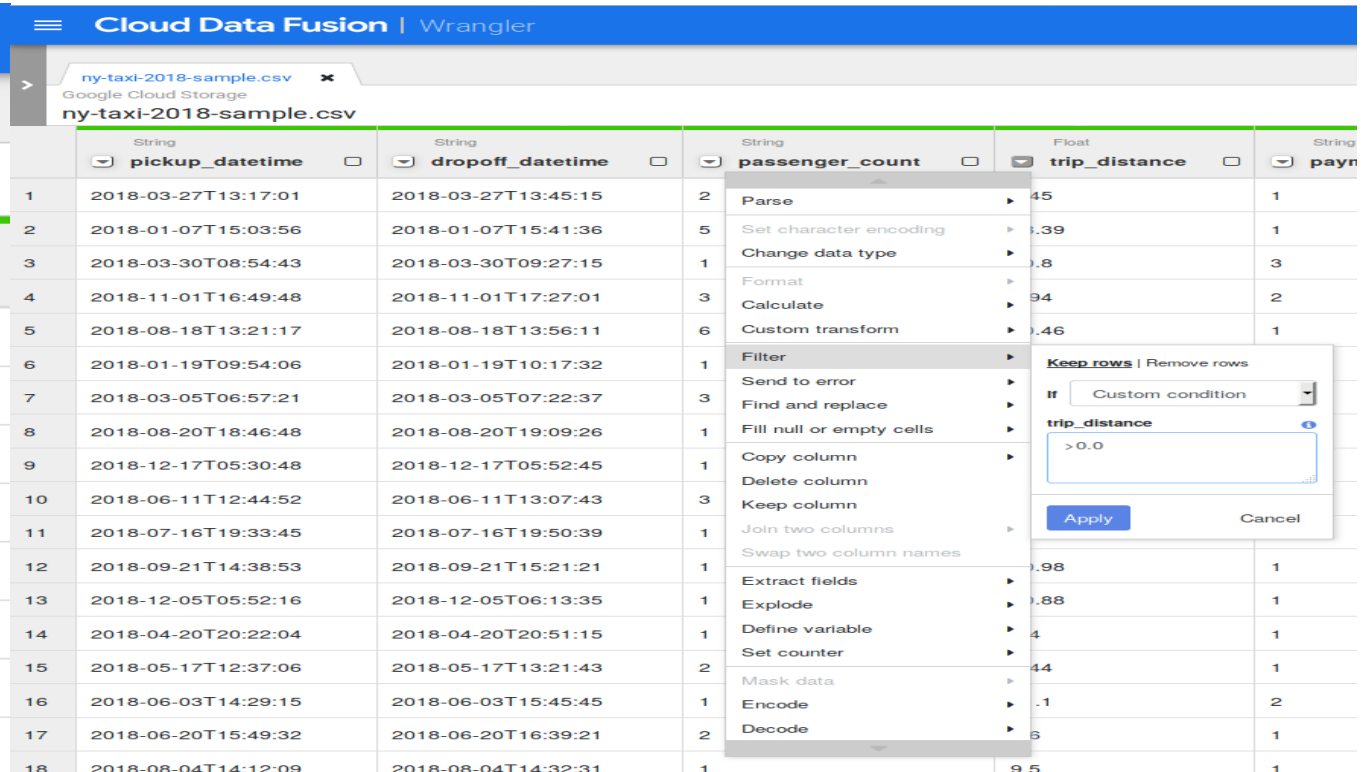
Preparing(Loading the data, wrangler (clean data))

- ✓ **export BUCKET=$GOOGLE_CLOUD_PROJECT**
  gsutil mb gs://$BUCKET
  gsutil cp gs://cloud-training/OCBL017/ny-taxi-2018-sample.csv gs://$BUCKET
- ✓ #create a bucket for temporary storage items that Cloud data Fusion will create.
  gsutil mb gs://$BUCKET-temp
- ✓ **View Instance** link on the Cloud Data Fusion instances page
- ✓ **Wrangler** is an interactive, visual tool that lets you see the effects of transformations on a small subset of your data before dispatching large, parallel-processing jobs on the entire dataset. On the Cloud Data Fusion UI, choose **Wrangler**. On the left side, there is a panel with the pre-configured connections to your data, including the Cloud Storage connection.
- ✓ Under **Google Cloud Storage**, select **Cloud Storage Default**.
- ✓ Click on the bucket corresponding to your project name.
- ✓ Select **ny-taxi-2018-sample.csv**. The data is loaded into the Wrangler screen in row/column form.

wrangler (clean data)

- ✓ **Parse > CSV**, select **Set first row as header** and then click **Apply**
- ✓ **Delete column** body
- ✓ column types have been loaded in as String
- ✓ **Change data type** trip_distance , total_amount column,**Float**.
- ✓ trip_distance column and select **Filter**. Click if **Custom condition** and input >0.0



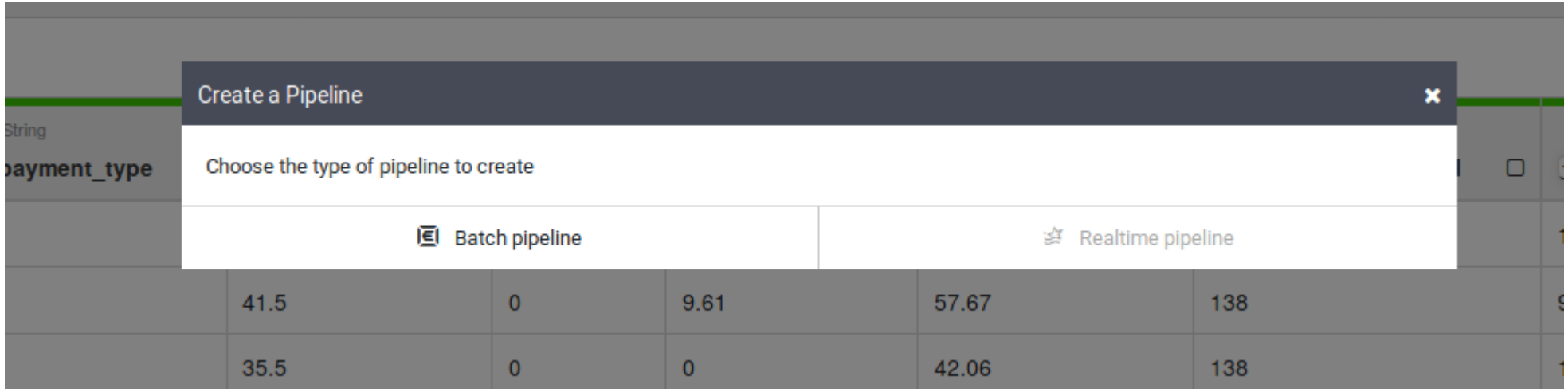


Creating pipeline

Cloud Data Fusion translates your visually built pipeline into an Apache Spark or MapReduce program that executes transformations on an ephemeral Cloud Dataproc cluster in parallel. This enables you to easily execute complex transformations over vast quantities of data in a scalable, reliable manner, without having to wrestle with infrastructure and technology.

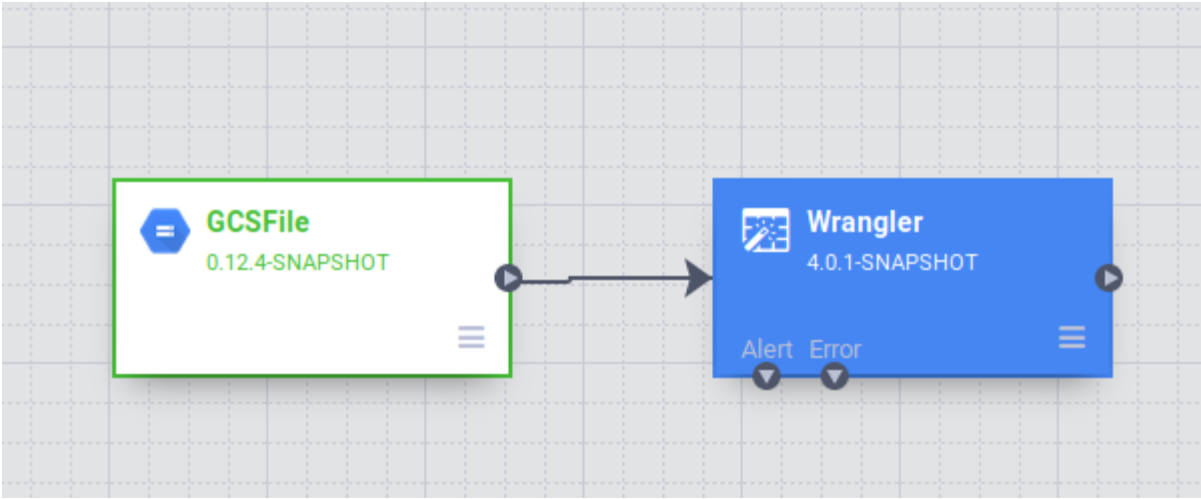
1.On the upper-right side of the Google Cloud Fusion UI, click **Create a Pipeline**.

2.In the dialog that appears, select **Batch pipeline**.



3.In the Data Pipelines UI, you will see a GCSFile source node connected to a Wrangler node. The Wrangler node contains all the transformations you applied in the Wrangler view captured as directive grammar. Hover over the Wrangler node and select **Properties**.

Creating pipeline



4.At this stage, you can apply more transformations by clicking the **Wrangle** button. Delete the extra column by pressing the red trashcan icon beside its name. To close the Wrangler tool click the **X** button in the top right corner.

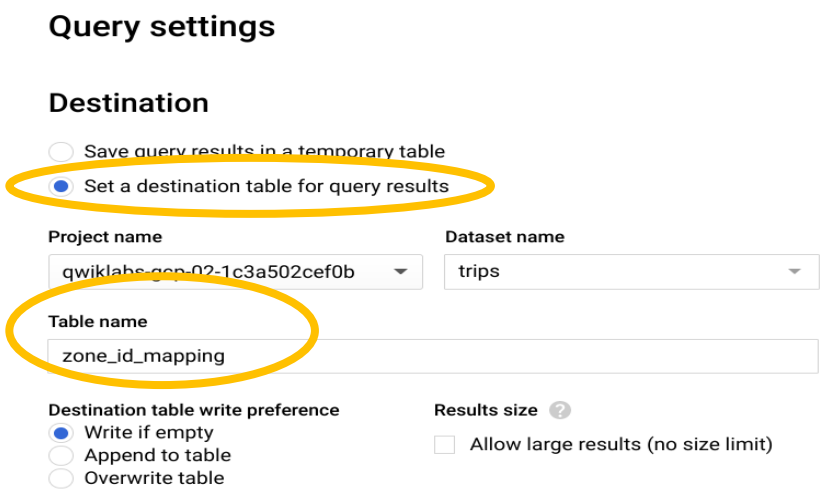
Adding a data source

- ✓ **Create Dataset** trips
- ✓ in bigquery editor **More > Query Settings** > **Set a destination table for query results**. Also, under **Table name** input zone_id_mapping.

- ✓ Run query

```
SELECT
  zone_id,
  zone_name,
  borough
FROM
  `bigquery-public-data.new_york_taxi_trips.taxi_zone_geom`
```

- ✓ Cloud Data Fusion > **Source > bigquery (add it) and click on properties (see next slide to change setting)**



Job information | Results | JSON | Execution details

Row	zone_id	zone_name	borough
1	1	Newark Airport	EWR
2	31	Bronx Park	Bronx
3	81	Eastchester	Bronx
4	254	Williamsbridge/Olinville	Bronx
5	250	Westchester Village/Unionport	Bronx
6	69	East Concourse/Concourse Village	Bronx
7	174	Norwood	Bronx
8	58	Country Club	Bronx
9	147	Longwood	Bronx

Properties Documentation

Label *
BigQuery

Label is bigquery

Basic

Reference Name *
zone_mapping

Reference : zone_mapping

BROWSE

Project ID
auto-detect

Dataset Project ID
Project the dataset belongs to, if different from the Project ID.

Dataset : trips

Dataset *
trips

Table *
zone_id_mapping

Table : zone_id_mapping

GET SCHEMA

Partition Start Date
Partition start date in format yyyy-MM-dd

Partition End Date
Partition end date in format yyyy-MM-dd

Filter

Temporary Bucket Name
qwiklabs-gcp-00-a1ffccd2637e-temp

Temporary Bucket Name name of your project followed by "-temp"

Credentials

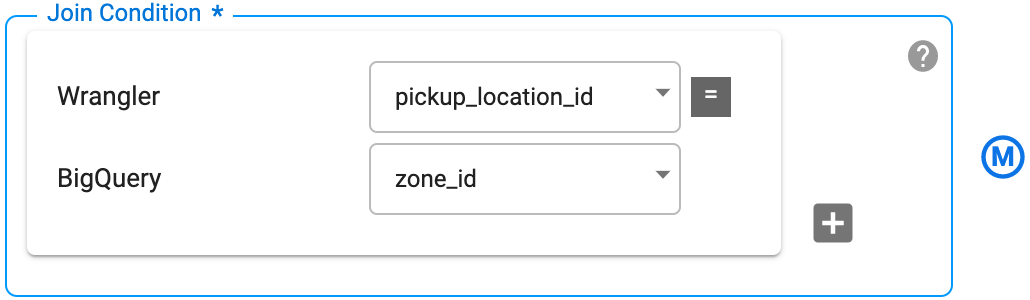
Service Account File Path
auto-detect

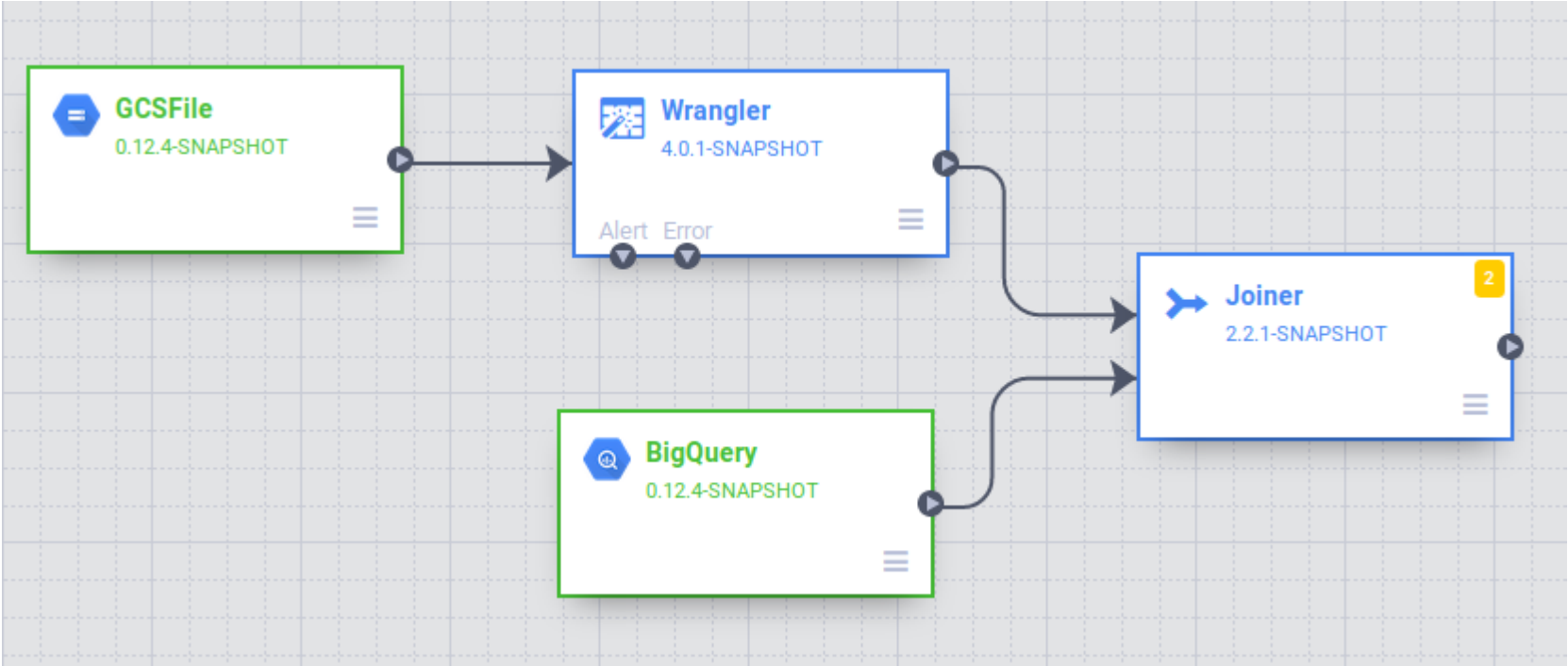
✓ To populate the schema of this table from BigQuery, click **Get Schema**. The fields will appear on the right side of the wizard.

✓ To close the BigQuery Properties window click the **X** button in the top right corner.

Joining two sources(add ,properties)

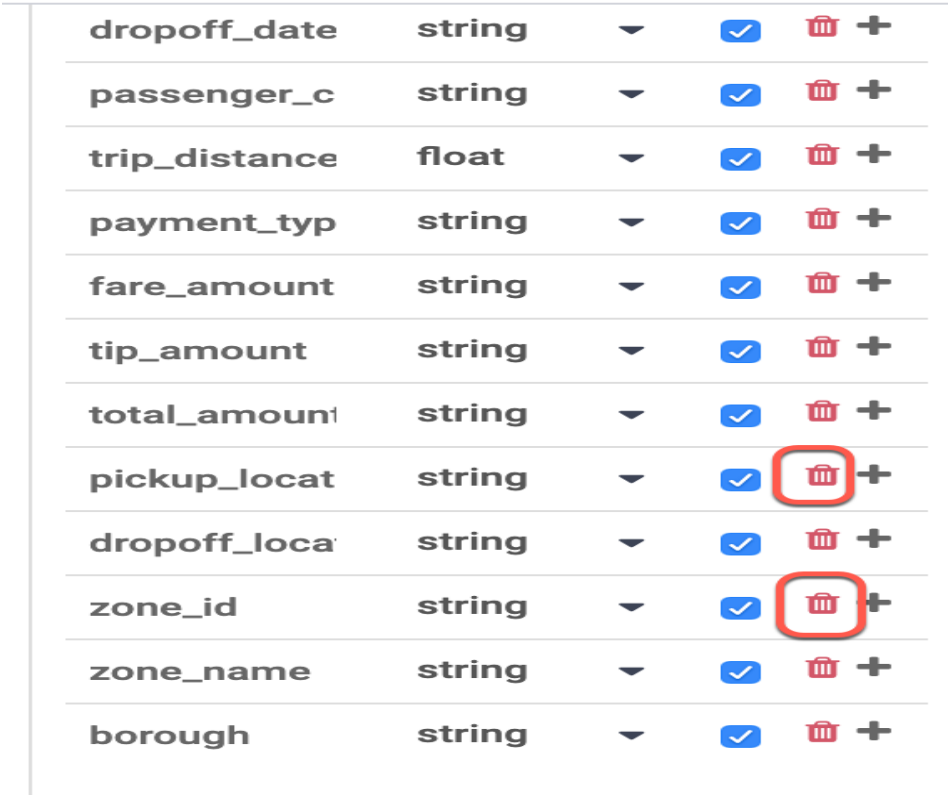
- ✓ **Analytics** section > **Joiner**
- ✓ Drag a connection arrows as shown in graph
- ✓ **Properties** of **Joiner**
  - * **Join Type :Inner**
  - * **fill join condition**





Joining two sources(properties : remove unnecessary columns)

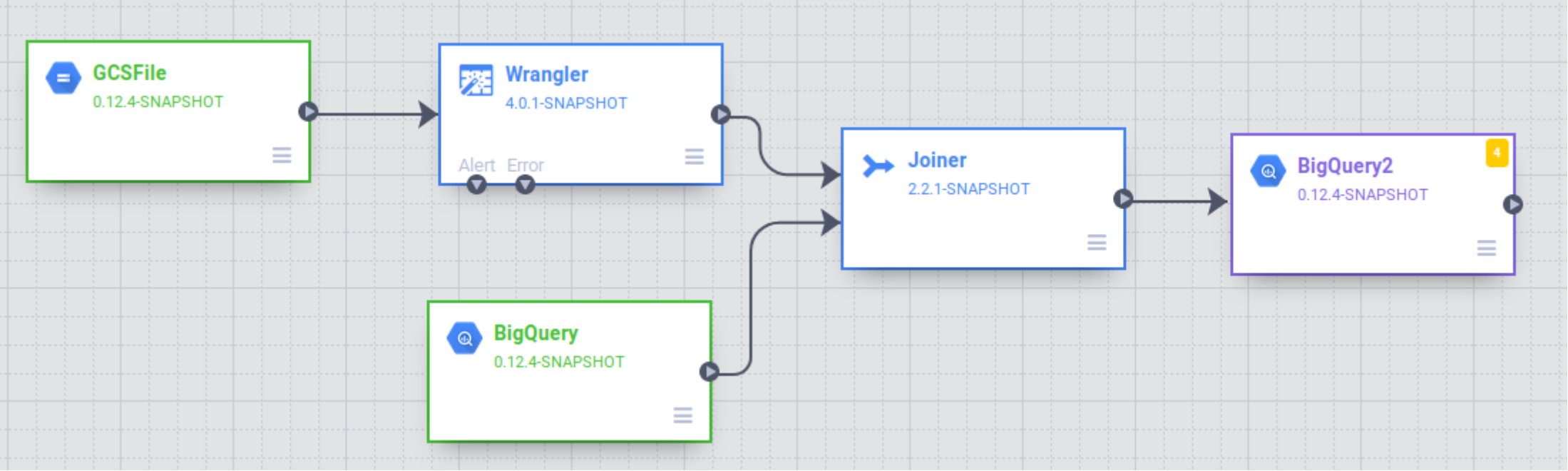
•To generate the schema of the resultant join, click **Get Schema**.
•In the **Output Schema** table on the right, **remove** the zone_id and pickup_location_id fields by hitting the red garbage can icon.
Close the window by clicking the **X** button in the top right corner.



dropoff_date	string	
passenger_c	string	
trip_distance	float	
payment_typ	string	
fare_amount	string	
tip_amount	string	
total_amount	string	
pickup_locat	string	
dropoff_loca	string	
zone_id	string	
zone_name	string	
borough	string	

Storing the output to BigQuery

- ✓ **Sink** section >> **BigQuery**.
- ✓ Drag a connection arrow
- ✓ Click on properties (see next slide)
- ✓ Close the window



BigQuery Properties 0.14.2

This sink writes to a BigQuery table. BigQuery is Google's serverless, highly scalable, enterprise data warehouse. Data is first written to a temp...

Validate

Properties | Documentation

dropoff_datetime	string	✓
passenger_count	string	✓
trip_distance	float	✓
payment_type	string	✓
fare_amount	string	✓
tip_amount	string	✓
total_amount	string	✓
dropoff_location_id	string	✓
zone_name	string	✓
borough	string	✓

Basic

Reference Name *
bq_insert

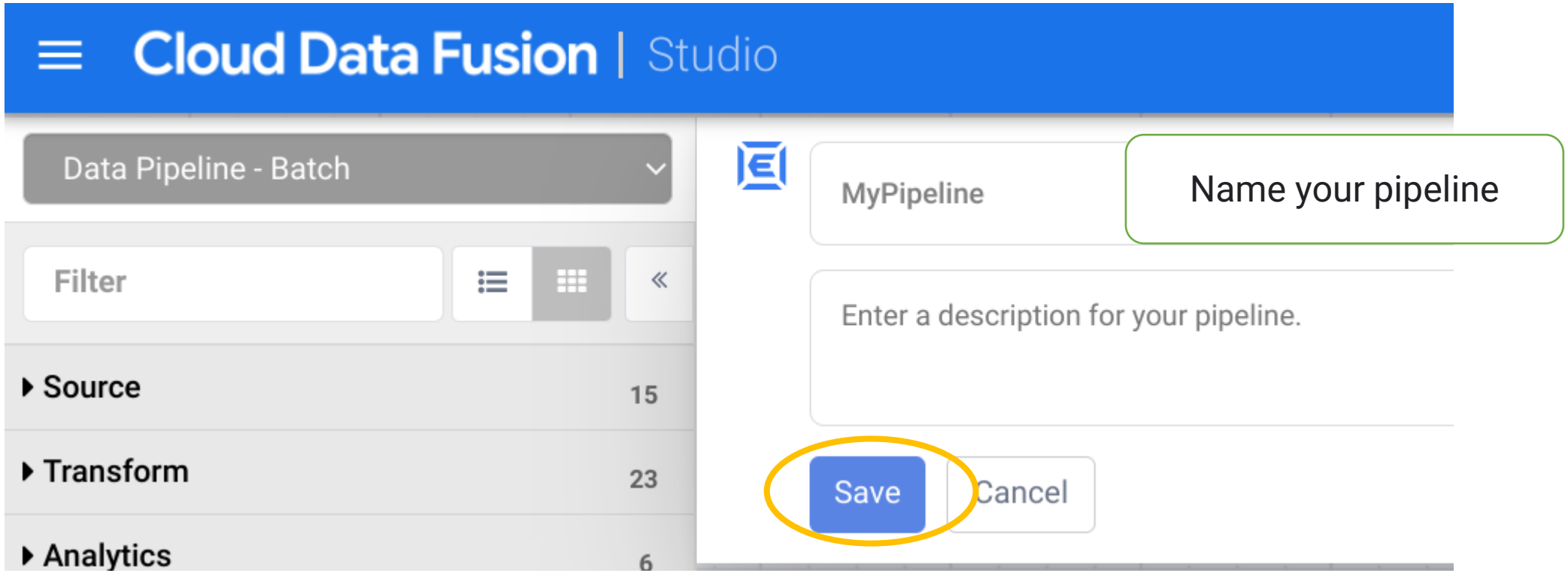
Project ID
auto-detect

Dataset *
trips

Table *
trips_pickup_name

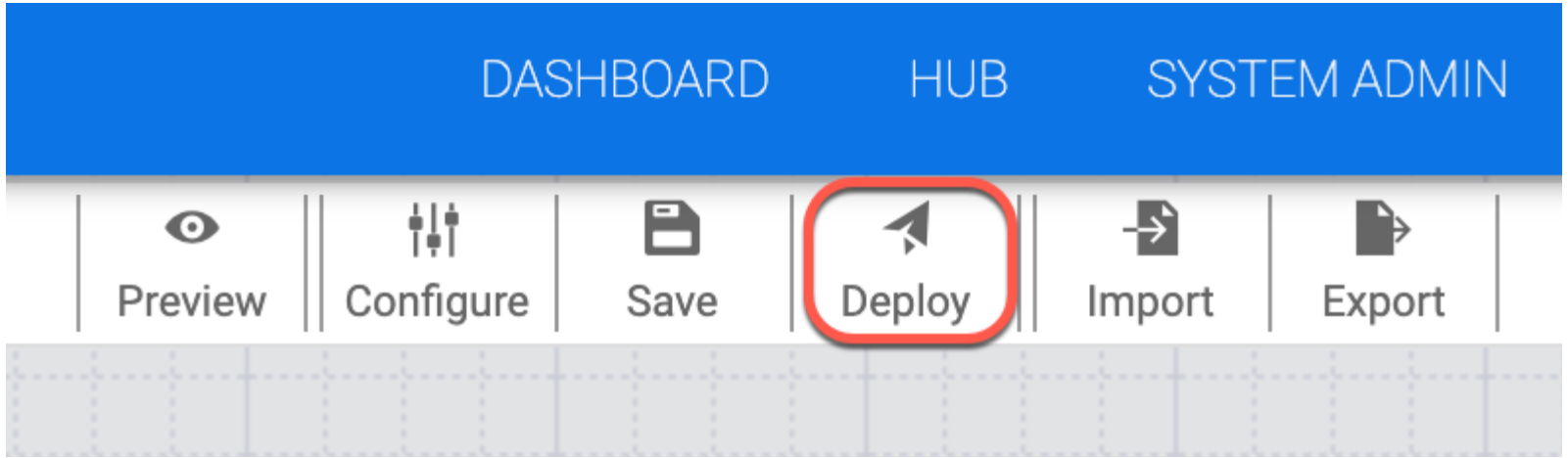
Temporary Bucket Name
qwiklabs-gcp-03-b70d69904768-temp

Deploying and running the pipeline

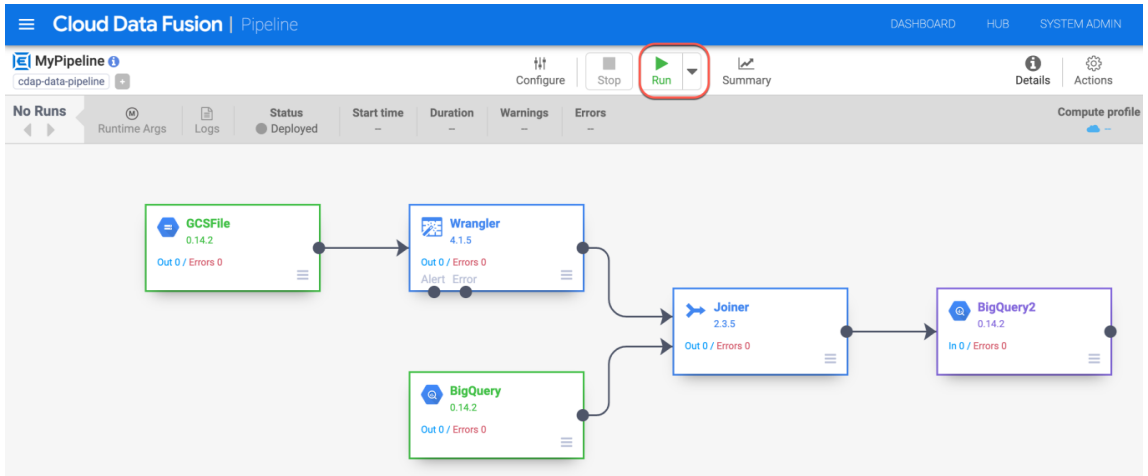


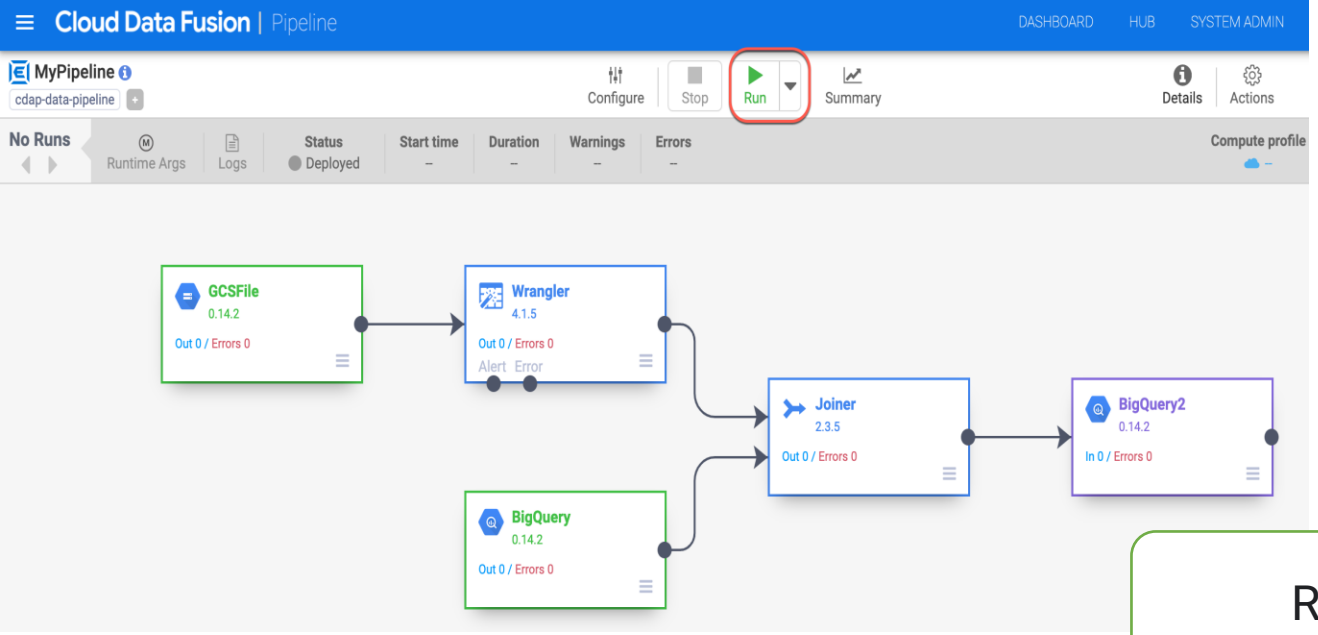
Name your pipeline

2.Now you will deploy the pipeline. In the upper-right corner of the page, click **Deploy**.

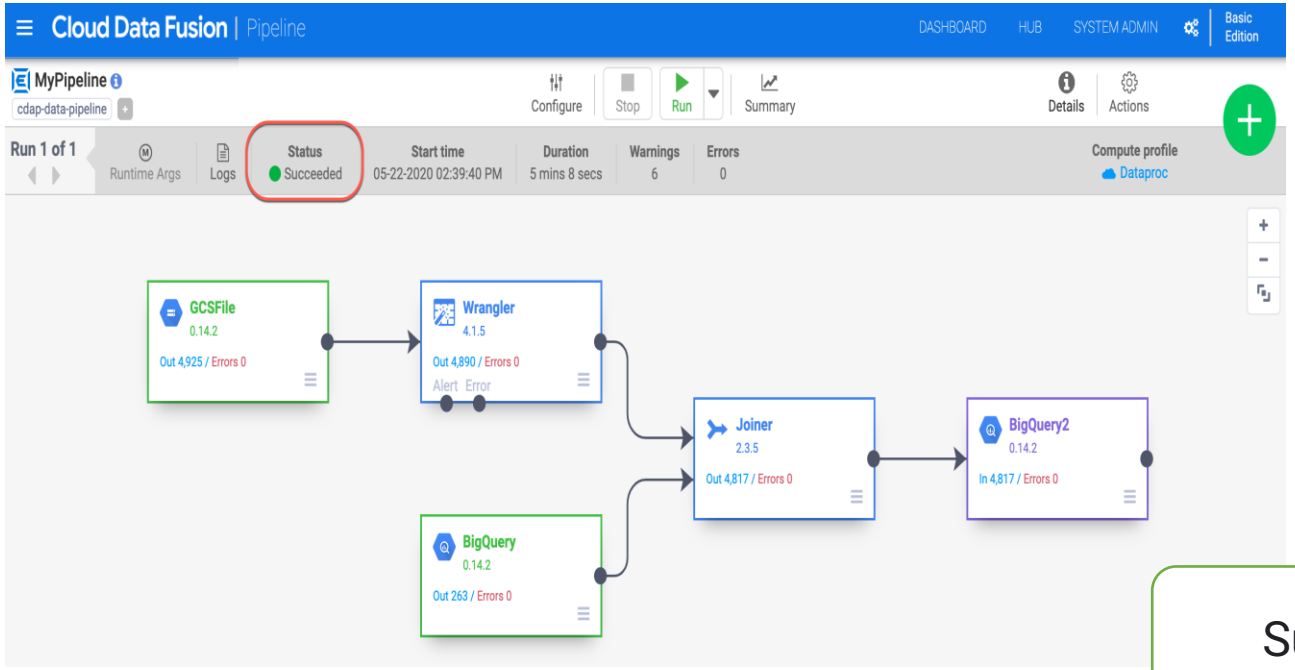


3.On the next screen click **Run** to start processing data.





Run pipeline



Success process

Viewing result in bigquery

```
SELECT * FROM `trips.trips_pickup_name
```