Cloudwick

aws

Runbook for Amorphic Training

Day 2

LAB Session 1

Datasets:

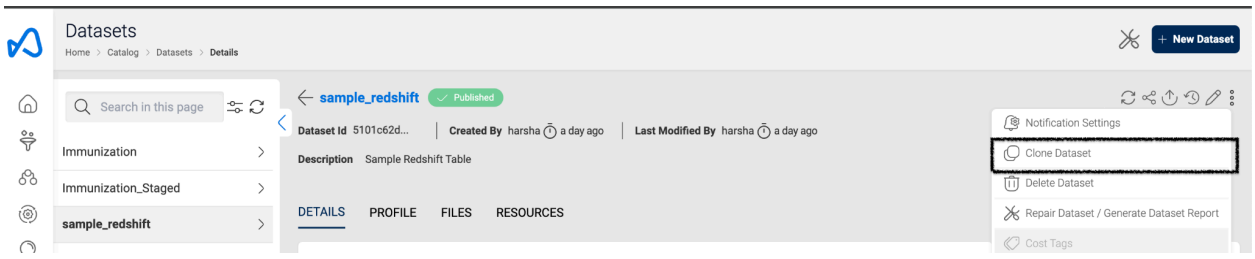
The easiest way to create a Dataset is by cloning.
You can clone a Dataset in Amorphic. The Clone Dataset page auto-populates with the metadata from the existing dataset, so you only need to change the Dataset Name. You can edit any field before clicking Register, and a new dataset will appear in the Datasets page.

Below are some different types of Datasets that are already created for you to view and play with.

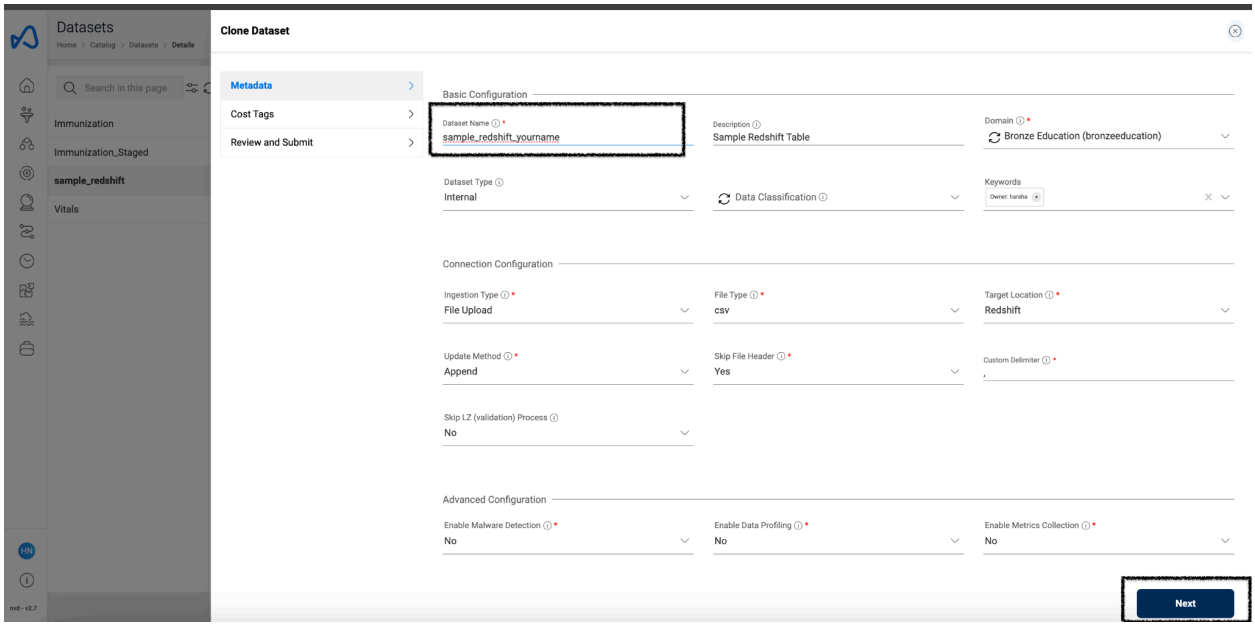
1. sample_redshift
2. sample_athena
3. sample_lakeformation
4. sample_s3

Cloning a Dataset.

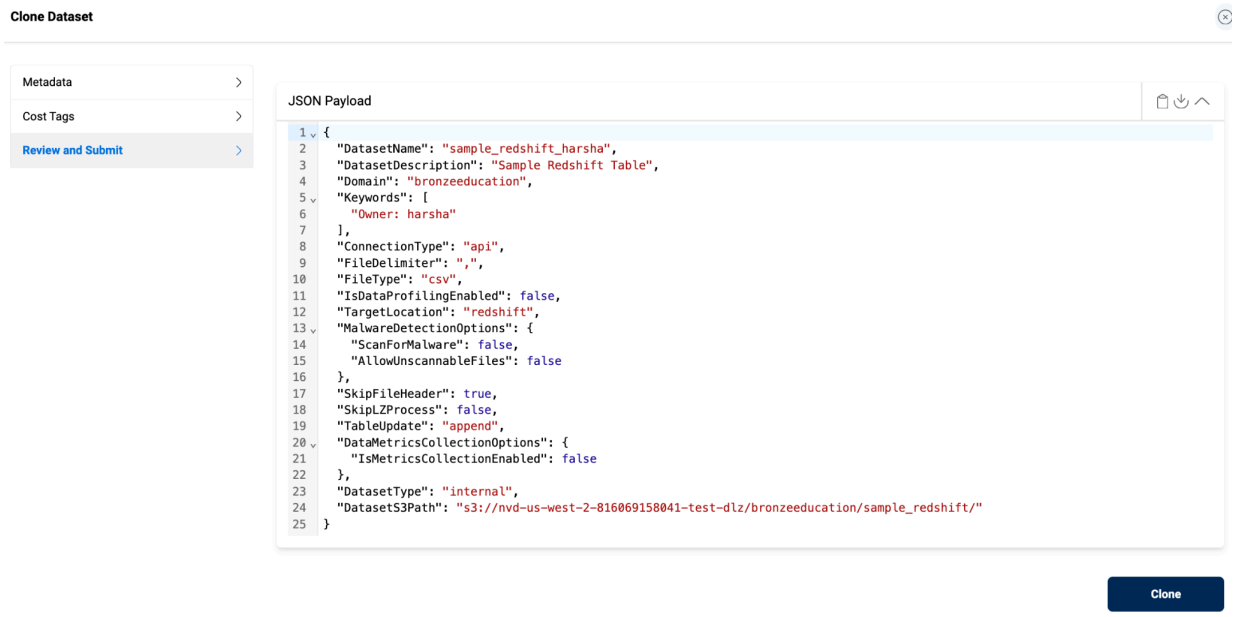
1. Choose a Dataset you want to clone.
2. Click on the three dots on the right top corner of Dataset Page



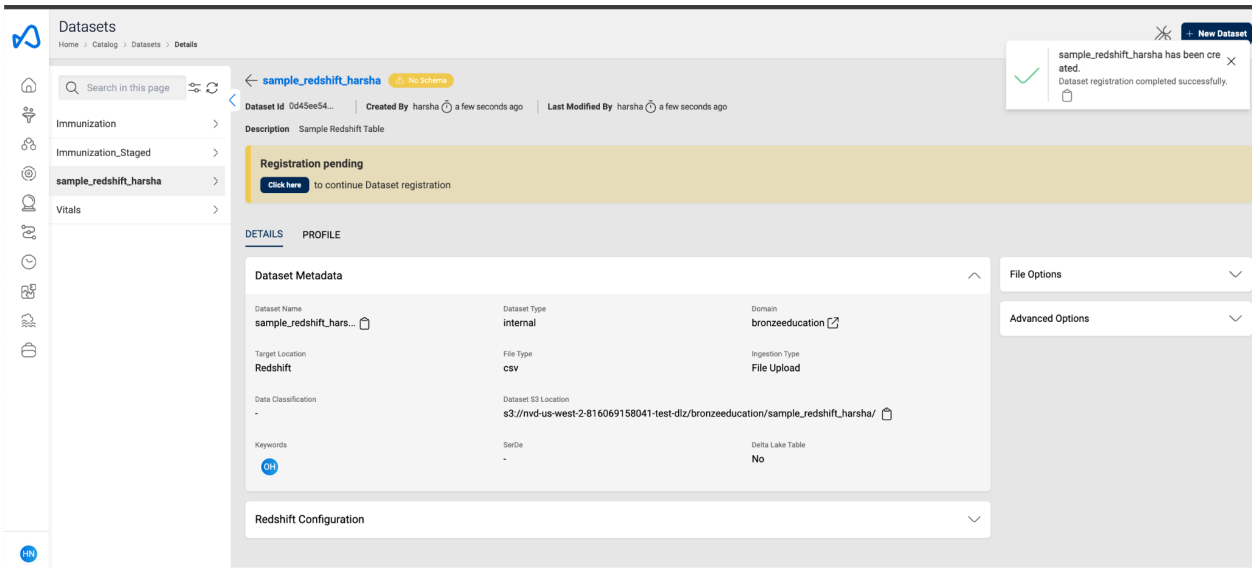
3. Change the properties you are interested in.
   For this demo lets change the name of a Dataset. Since a Datasets Name needs to be unique for a given Domain, change the Dataset Name to `sample_redshift_<yourname>`. Ex : *sample_redshift_harsha*



4. Click on Next. Skip the Cost tags and Press on Continue.
5. Review the changes you made and hit on Clone.

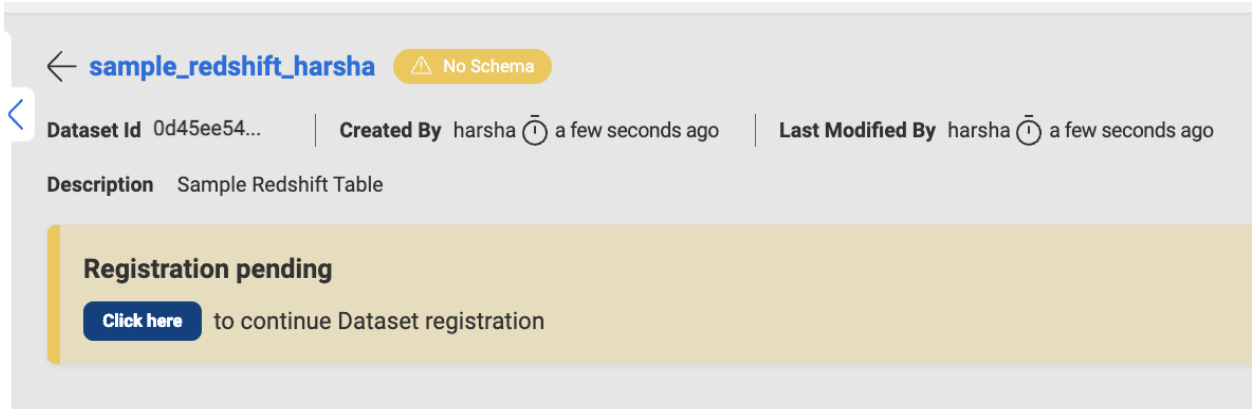


6. If Successful you would be taken to the Dataset page as below.



Dataset Registration:

1. Continue Registering the previous Dataset you created with a valid schema.



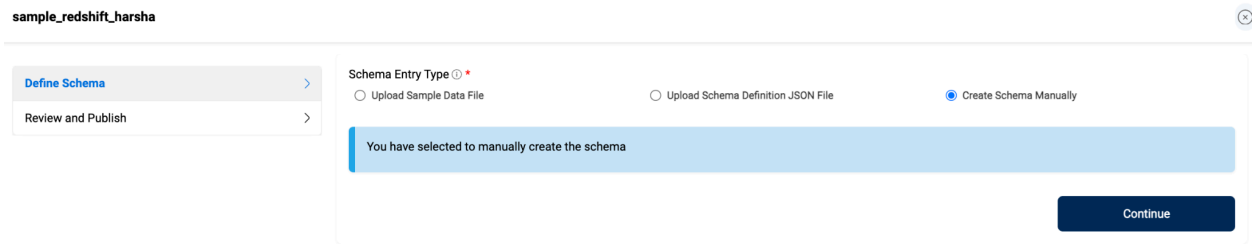
2. You are given different options to register a schema.

   A. upload a sample data file

   B. upload a JSON file containing the schema definition

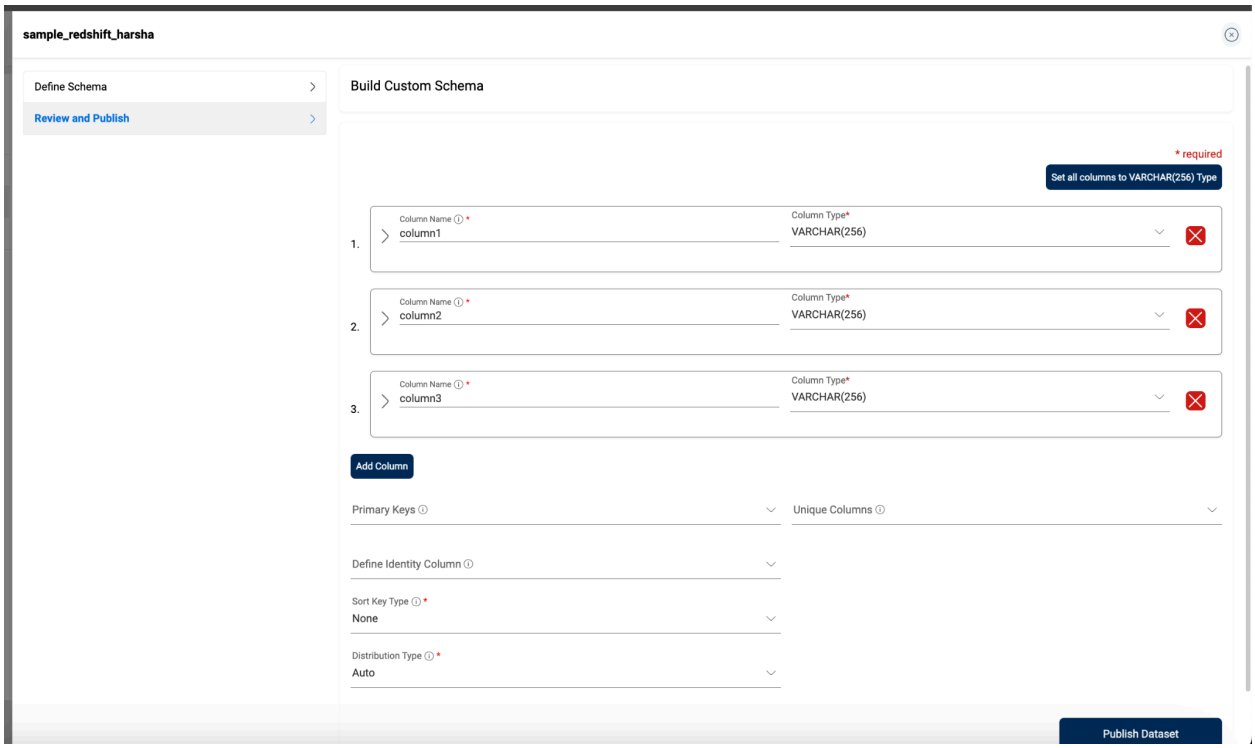
   C. manually enter the schema fields.

We will choose option C for simplicity. Click on Continue.

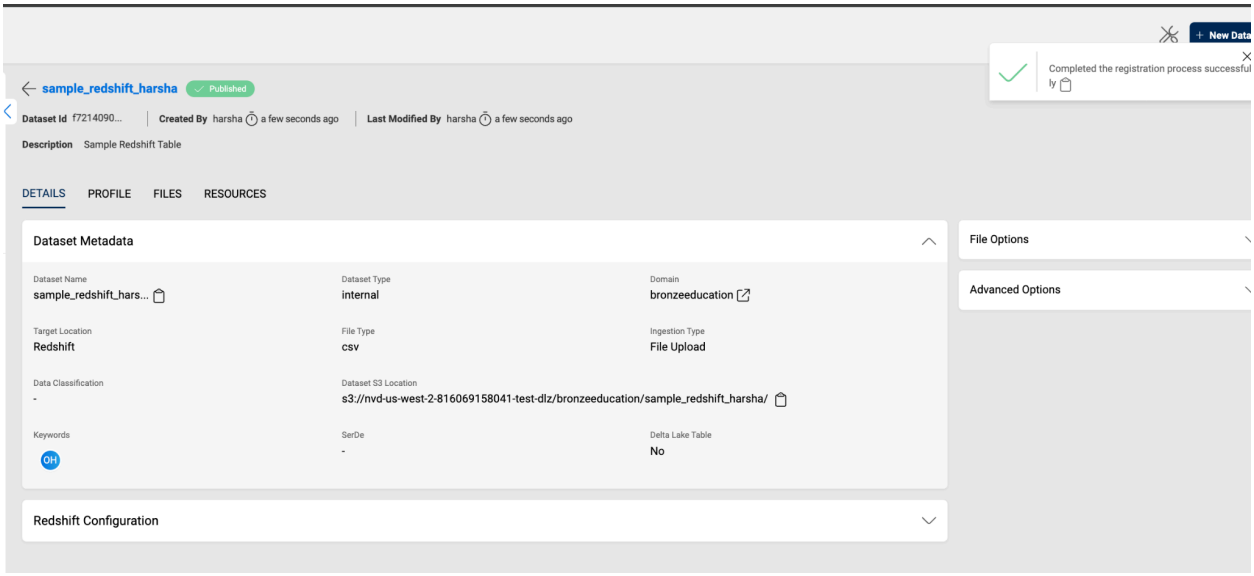


3. Click on Add Columns to fill column details.
   Select appropriate DataType from the drop down. Set Sort Key Type to None. Leave everything else to Default and click on Publish Dataset.

   Note: This page could slightly vary depending on the type of Dataset. Below screenshot is for the redshift Dataset.



You will be redirected to Dataset Page with Notification banner on the side with Success message.



Congrats! Now you have learnt to clone the Dataset from the existing one.

LAB Session 2:

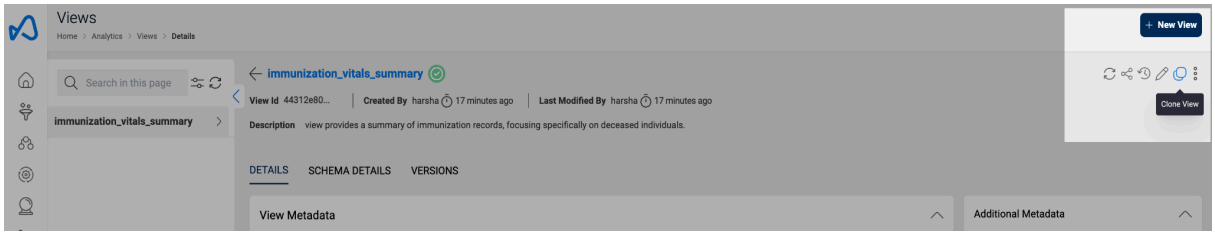
Views in Amorphic:

The easiest way to create a view is by cloning.

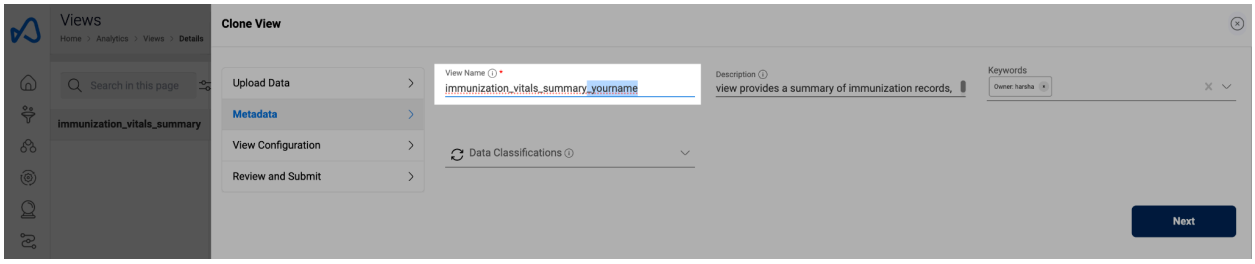
You can clone a view in Amorphic. The Clone View page auto-populates with the metadata from the existing view, so you only need to change the view name and update the view name in the view query.

Cloning a View.

1. Choose a View you want to clone.
2. In the top right corner, you can see the clone option.

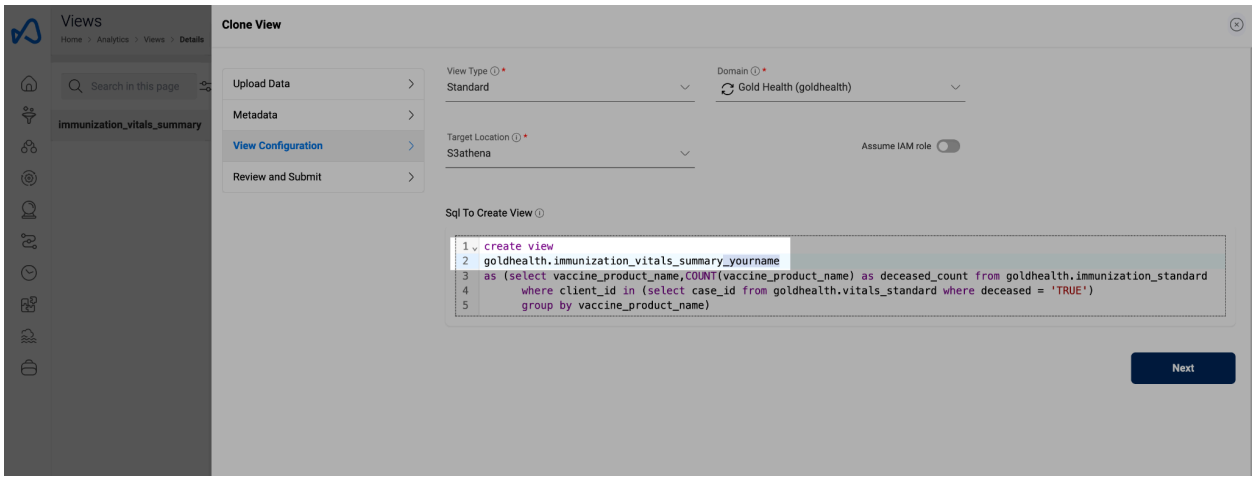


3. For this demo lets change the name of a View. Since a View Name needs to be unique for a given Domain, change the View Name to 'immunization_vitals_summary_<yourname>'.
   Ex : immunization_vitals_summary_harsha

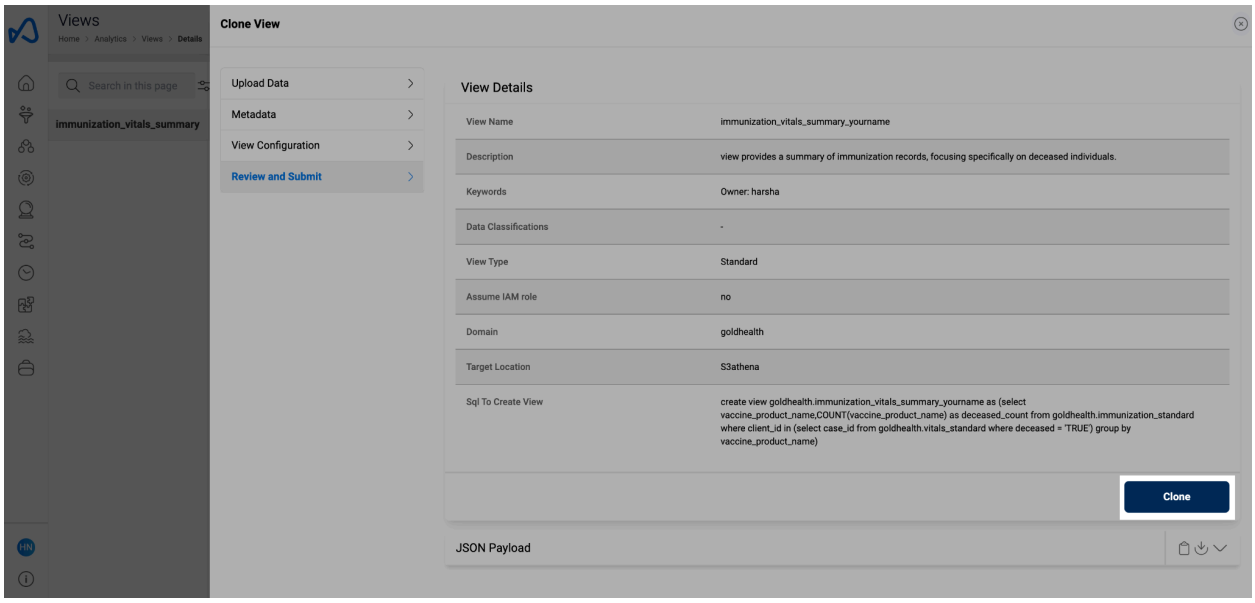


4. Click on Next.

5. Change the view name in the query, then click Next.



6. Review the changes you made and hit on Clone.

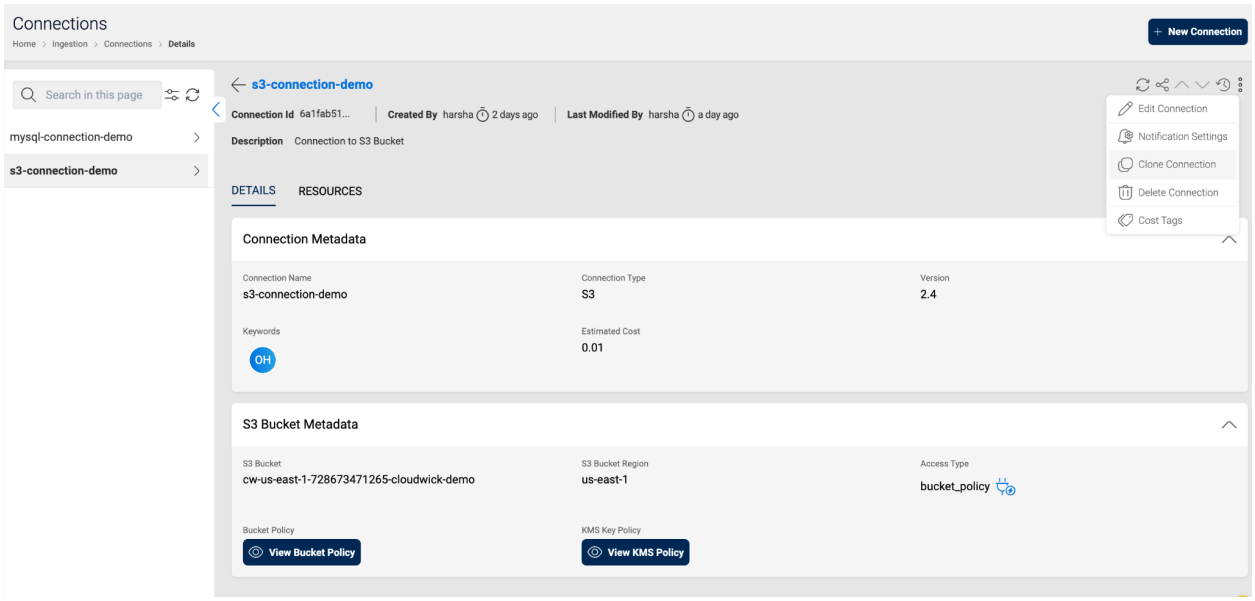


Congrats! Now you have learnt to clone the View from the existing one.

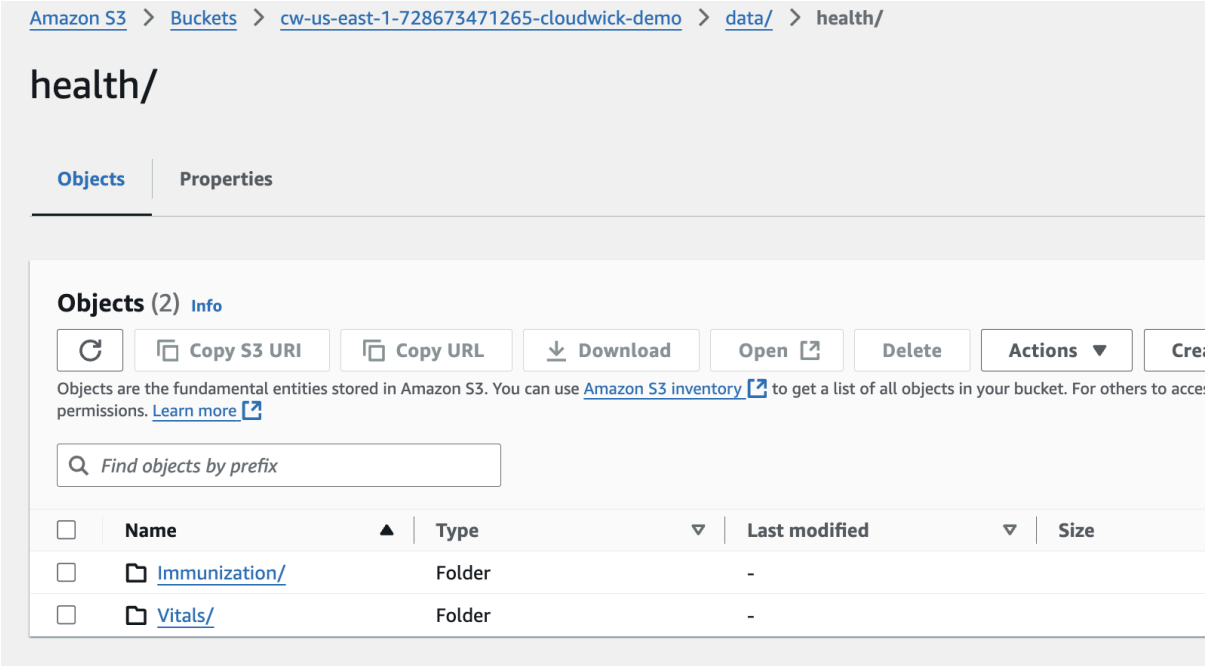
Connections:

Using a S3 Connection to Ingest data

1. A S3 connection **s3-connection-demo** is already created and shared for you. We will use this connection to ingest data from S3 bucket.

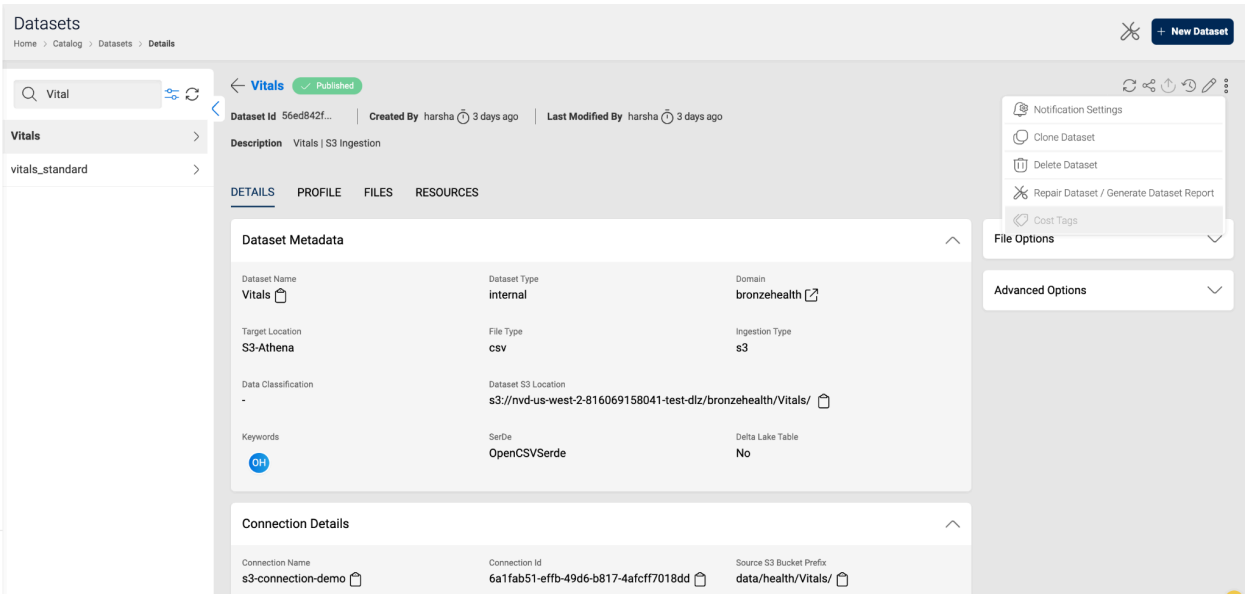


Below picture shows the files present in S3. We will ingest data from the folder *"data/health/Vitals/"* in this demo.

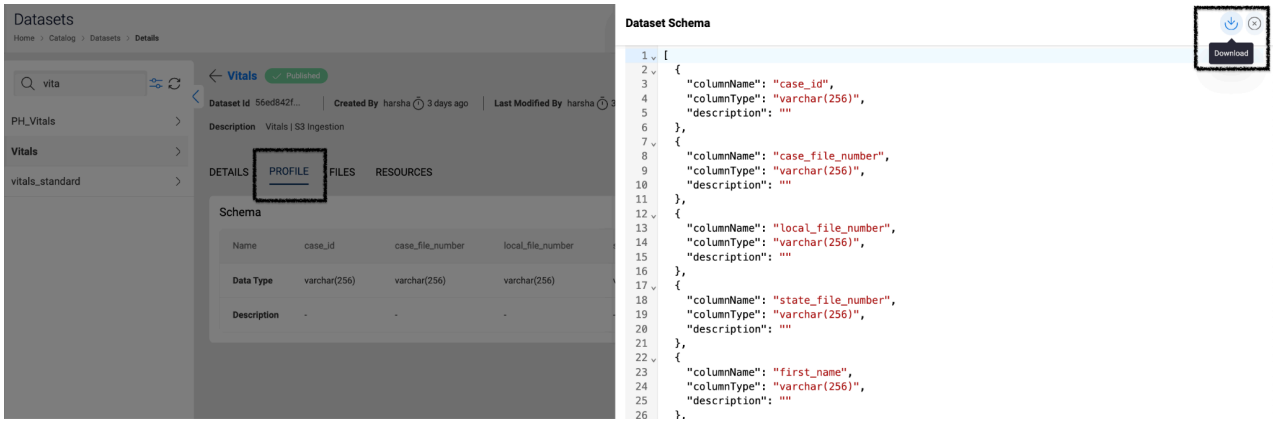


2. Head to the Dataset Page and search for Vitals. If you don't see it on your page. Increase the results per page displayed in the settings beside the reload button. If you still do not see a request for access.

3. Click on Vitals Dataset and clone the Dataset by clicking on the three dots on the right top corner of Dataset Page

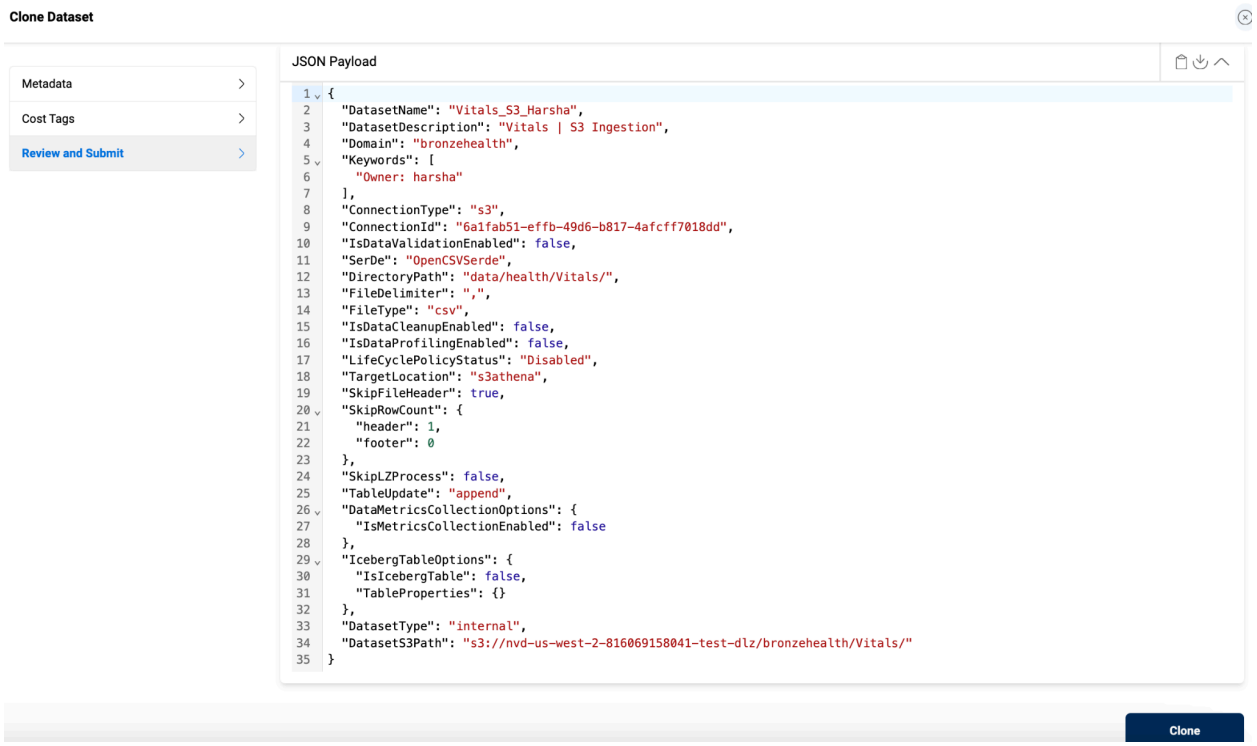


4. Download the schema from Profiles tab. A json file will be downloaded which will contain the schema of the Dataset. (which you will use to register Dataset)

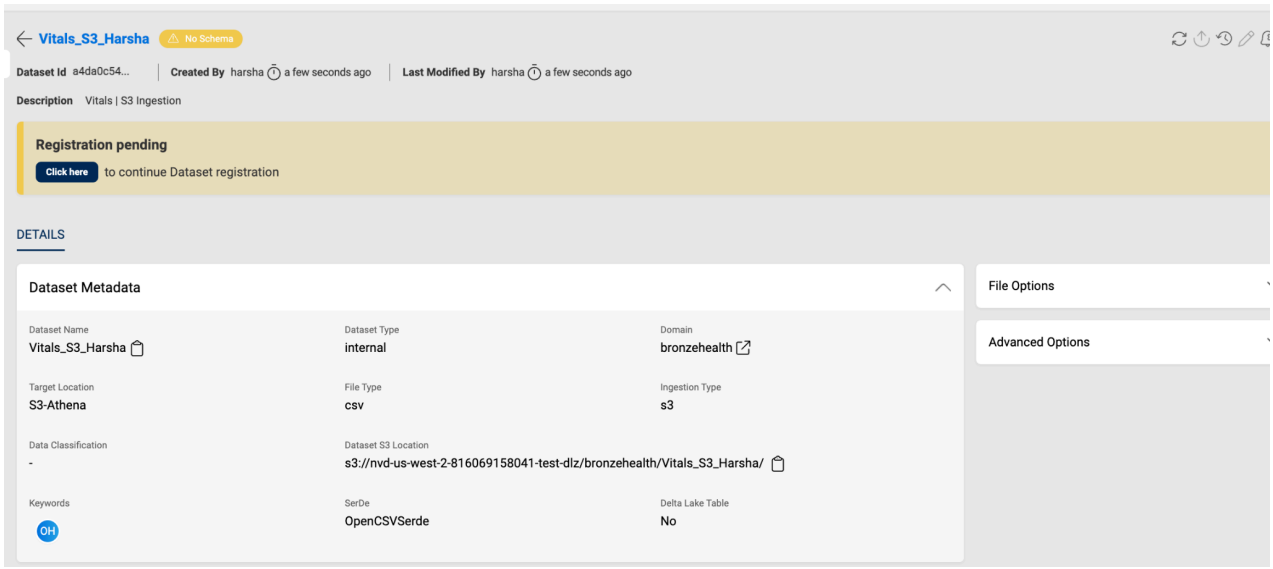


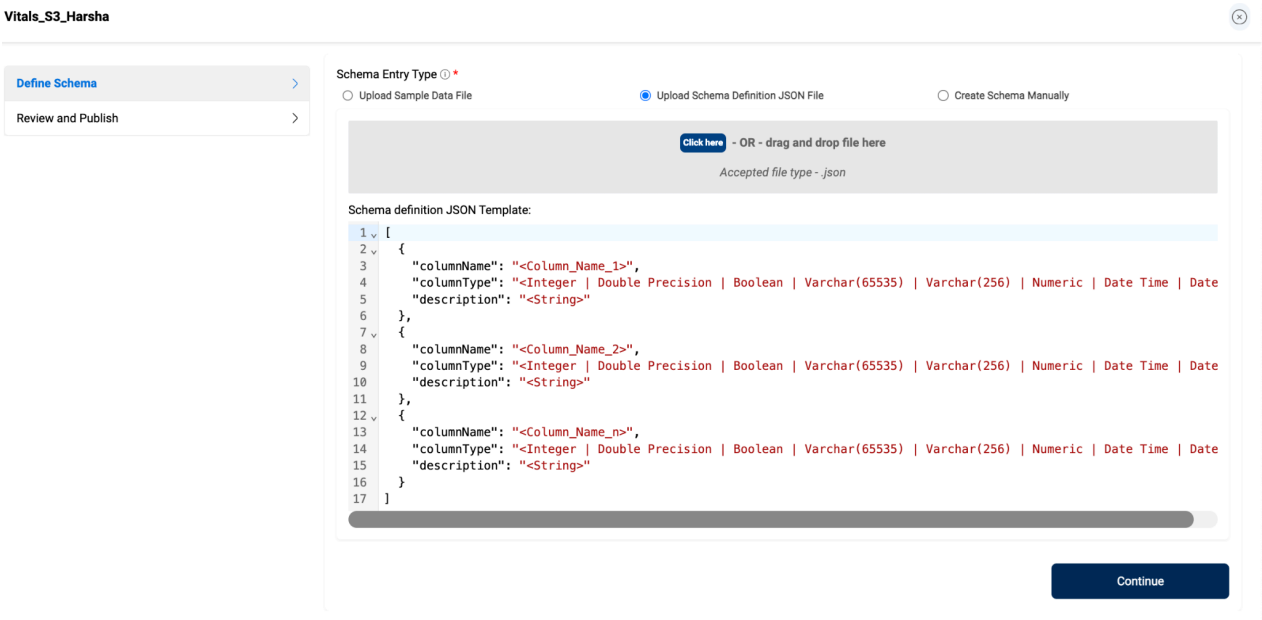
5. Take a look at all the properties. Especially the Connection and Directory Path.
   The connection **s3-connection-demo** has already been created. Directory path is the path where the data resides in source S3. In this case we are ingesting from *data/health/Vitals/* to the Dataset we are creating.

6. Change the name of Dataset to *Vitals_S3_<yourusername> Ex: Vitals_S3_Harsha* and click on Next. Skip the cost tags and Clone the Dataset.

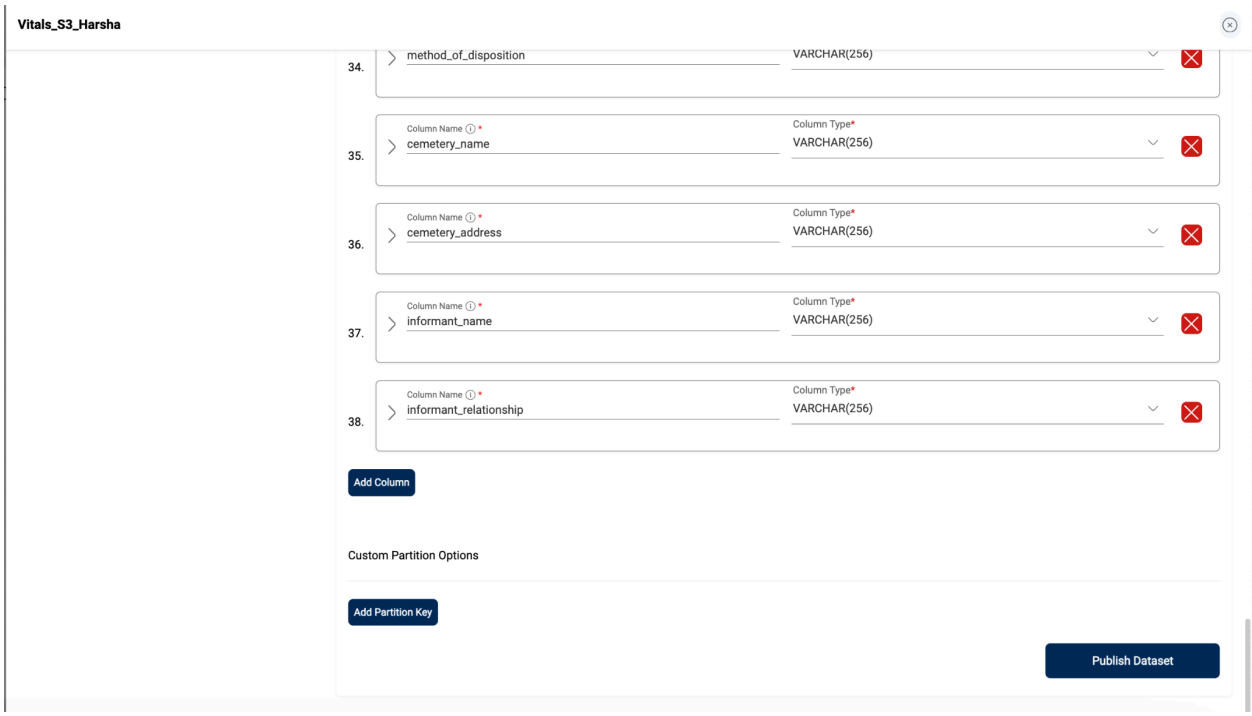


7. Continue to register the Dataset using the schema you downloaded from step 4. Upload the schema and wait for the extraction to complete.




Vitals_S3_Harsha
Define Schema
Review and Publish
Schema Entry Type *
Upload Sample Data File
Upload Schema Definition JSON File
Create Schema Manually
Click here - OR - drag and drop file here
Accepted file type - .json
Schema definition JSON Template:
[
{
"columnName": "<Column_Name_1>",
"columnType": "<Integer | Double Precision | Boolean | Varchar(65535) | Varchar(256) | Numeric | Date Time | Date
"description": "<String>"
},
{
"columnName": "<Column_Name_2>",
"columnType": "<Integer | Double Precision | Boolean | Varchar(65535) | Varchar(256) | Numeric | Date Time | Date
"description": "<String>"
},
{
"columnName": "<Column_Name_n>",
"columnType": "<Integer | Double Precision | Boolean | Varchar(65535) | Varchar(256) | Numeric | Date Time | Date
"description": "<String>"
}
]
Continue

8. Once the Extraction is complete, Scroll to the bottom and publish the Dataset.

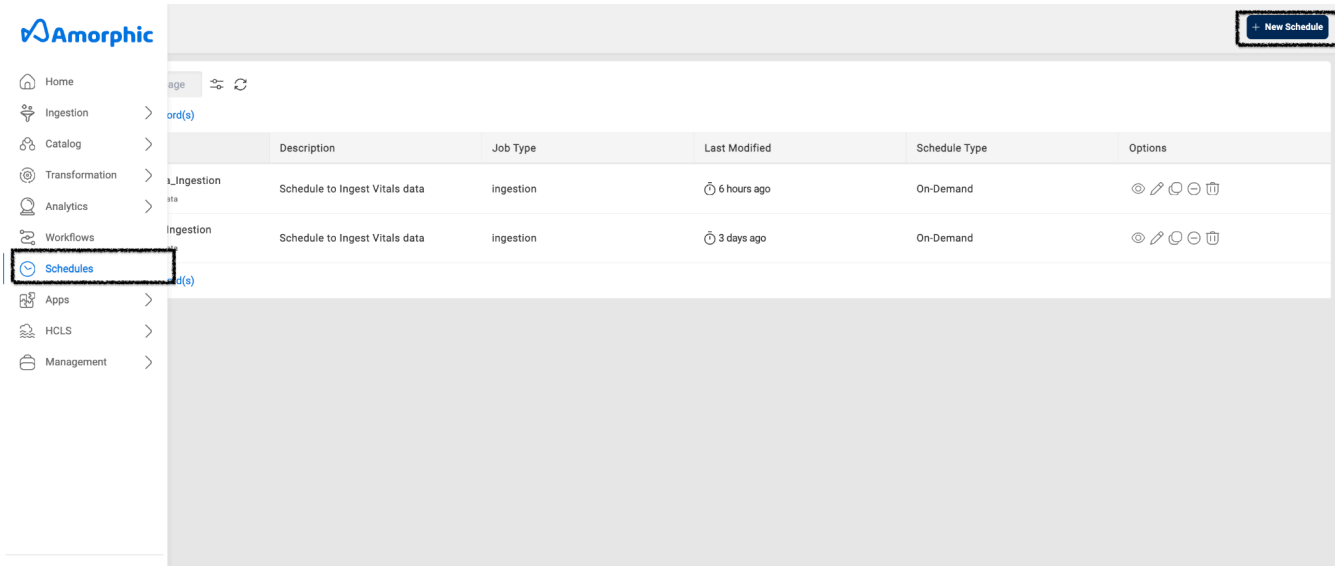


You have now successfully created a Dataset!

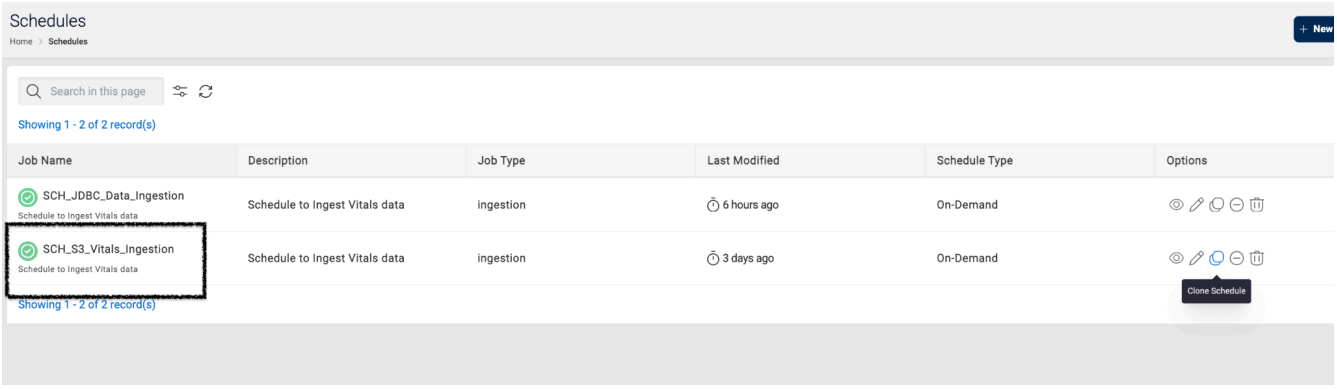
Scheduling Ingestion:

We can schedule the ingestion from S3 either on-demand or time based. Lets create a Schedule.

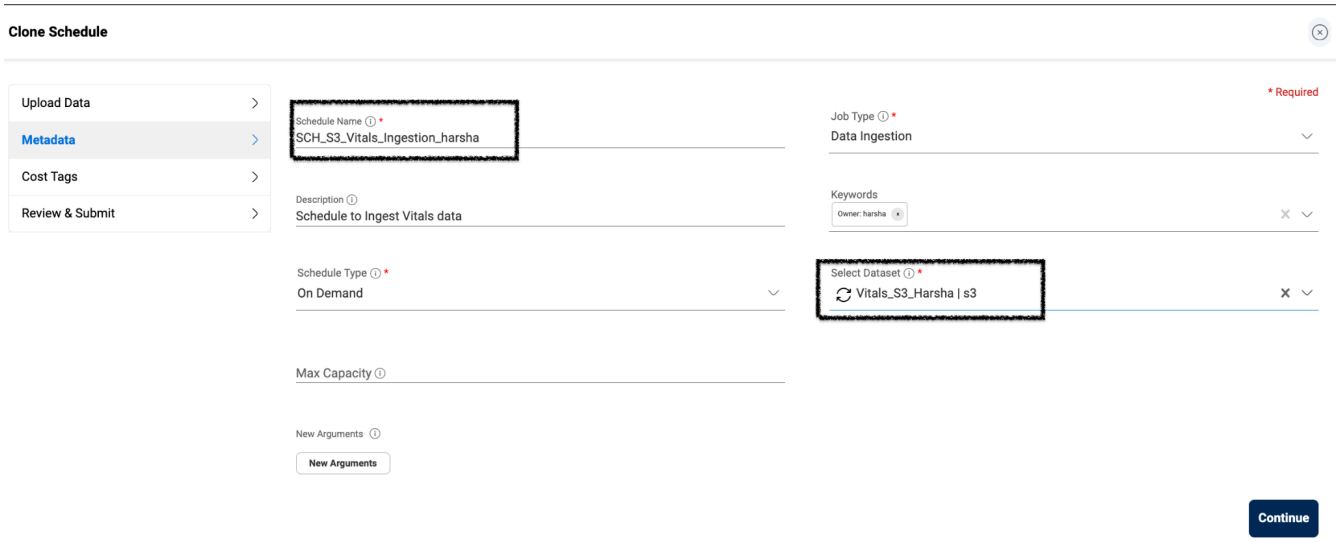
1. Navigate to schedules from the Home Page.



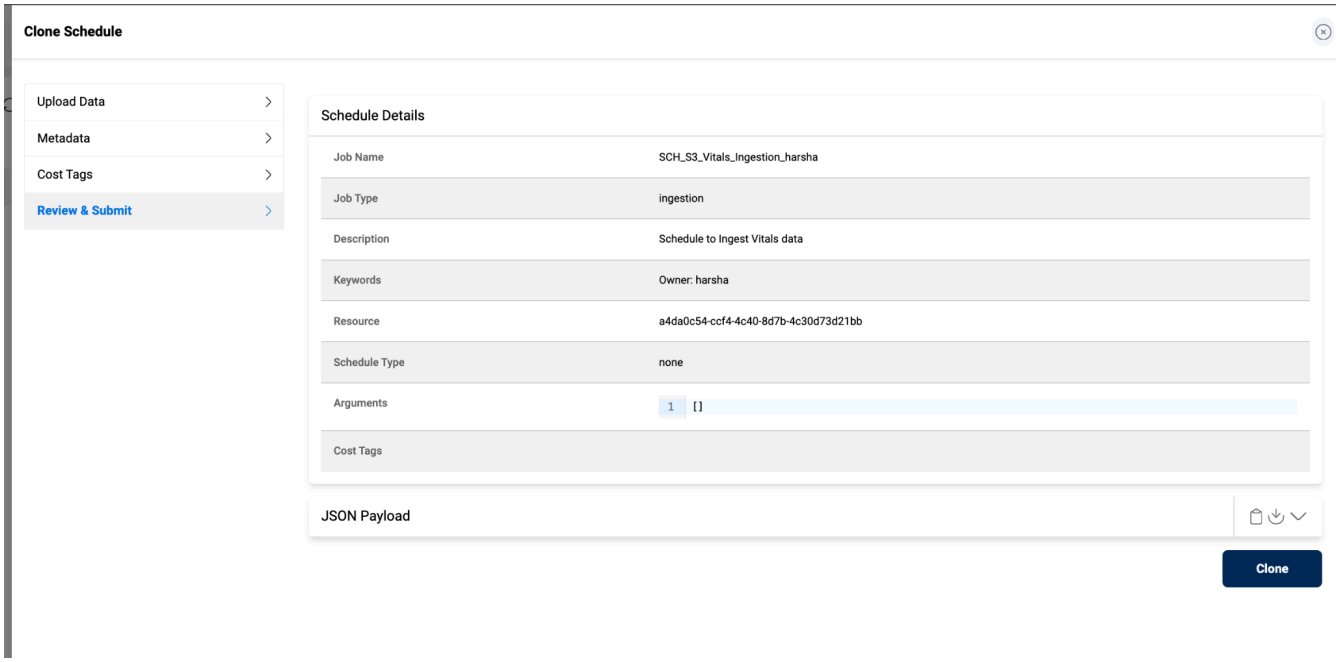
2. Clone the Schedule SCH_S3_Vitals_Ingestion and rename the schedule to SCH_S3_Vitals_Ingestion_<yourname> Ex: SCH_S3_Vitals_Ingestion_harsha



3. Update the Dataset to the one you created earlier and click on continue.

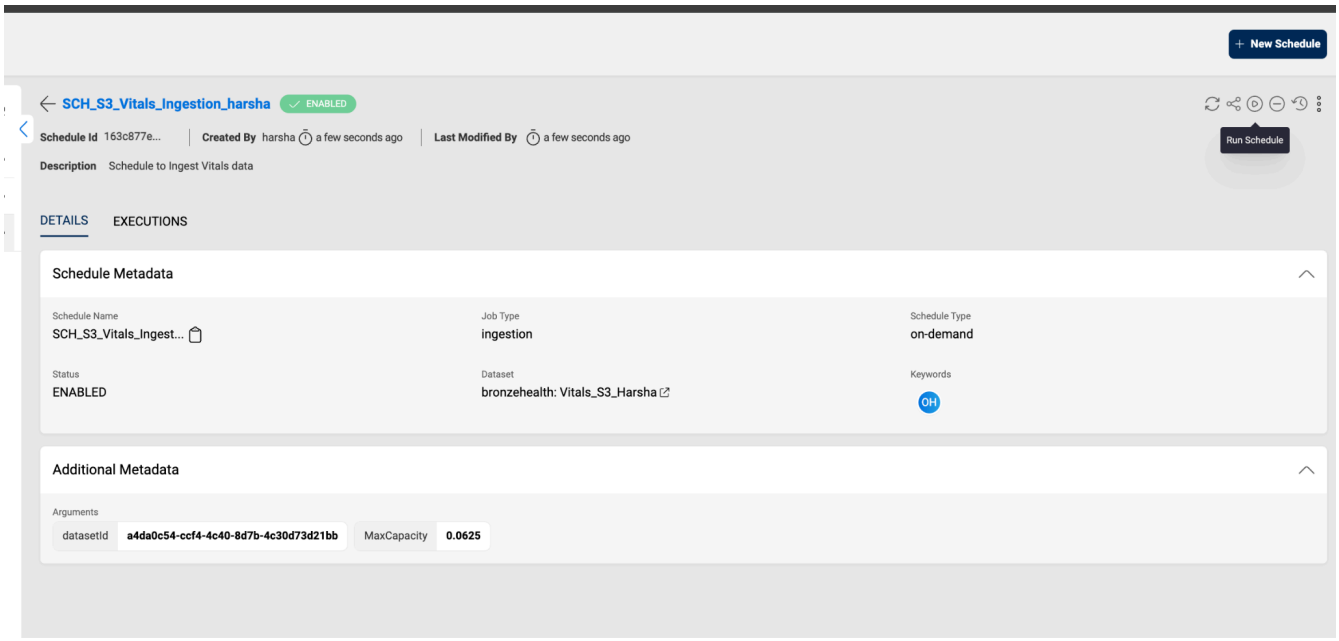


4. Skip Cost tags > Review > Clone

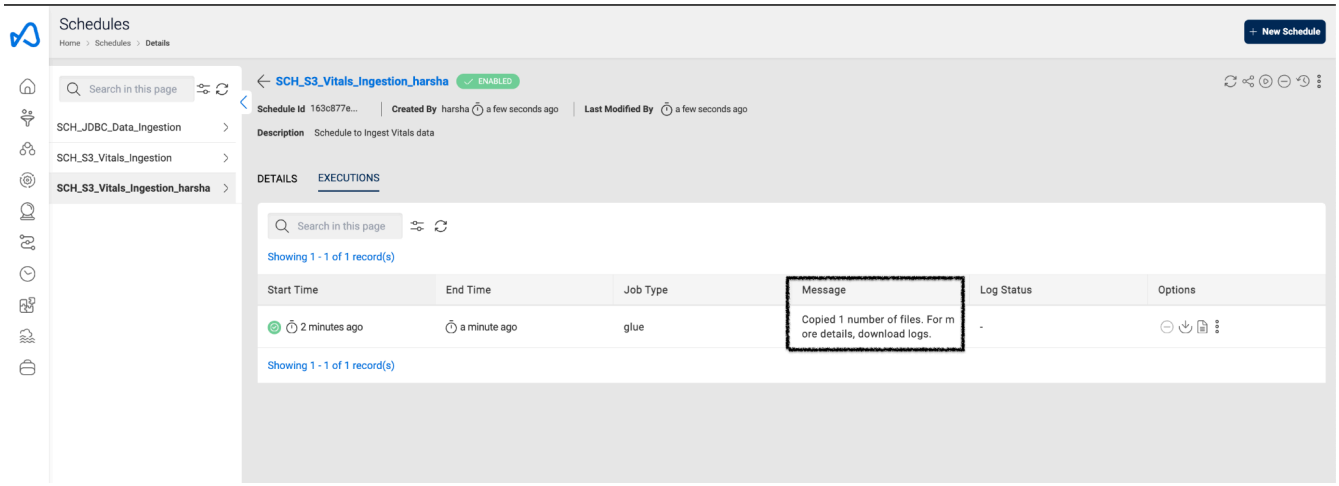


The schedule creation is now complete.

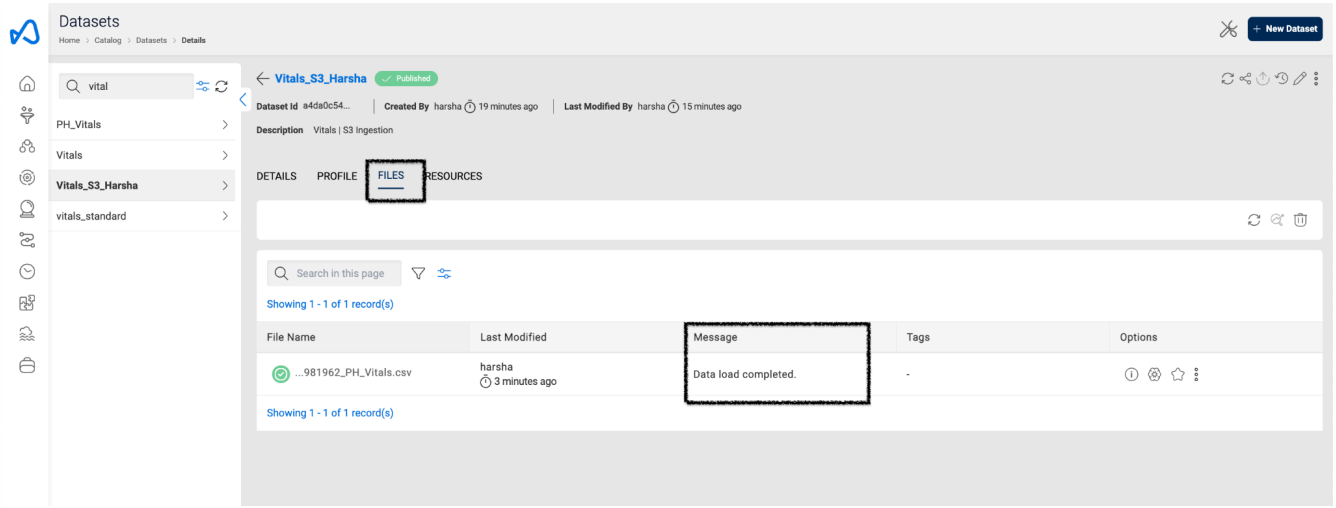
5. Let's run the schedule. To trigger the S3 Ingestion.



6. Once the execution is complete, your data would have been loaded to the Dataset you created.

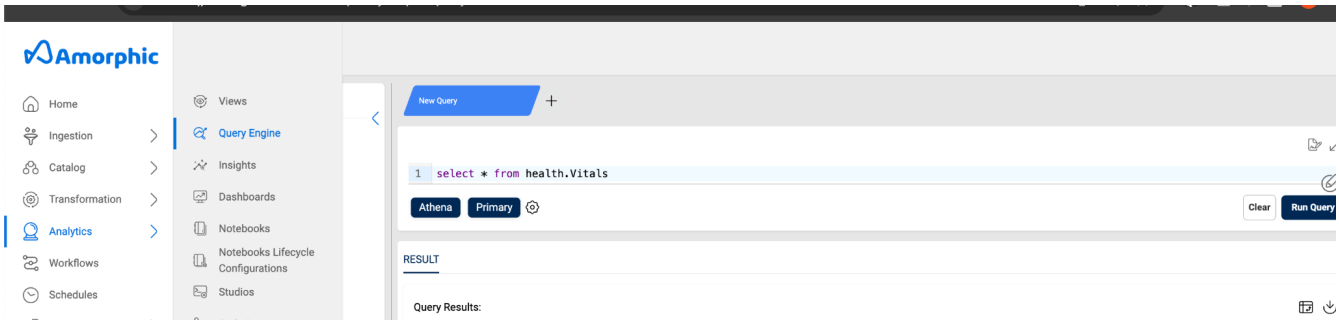


7. You can go back to the Dataset you created and confirm that file has been loaded.

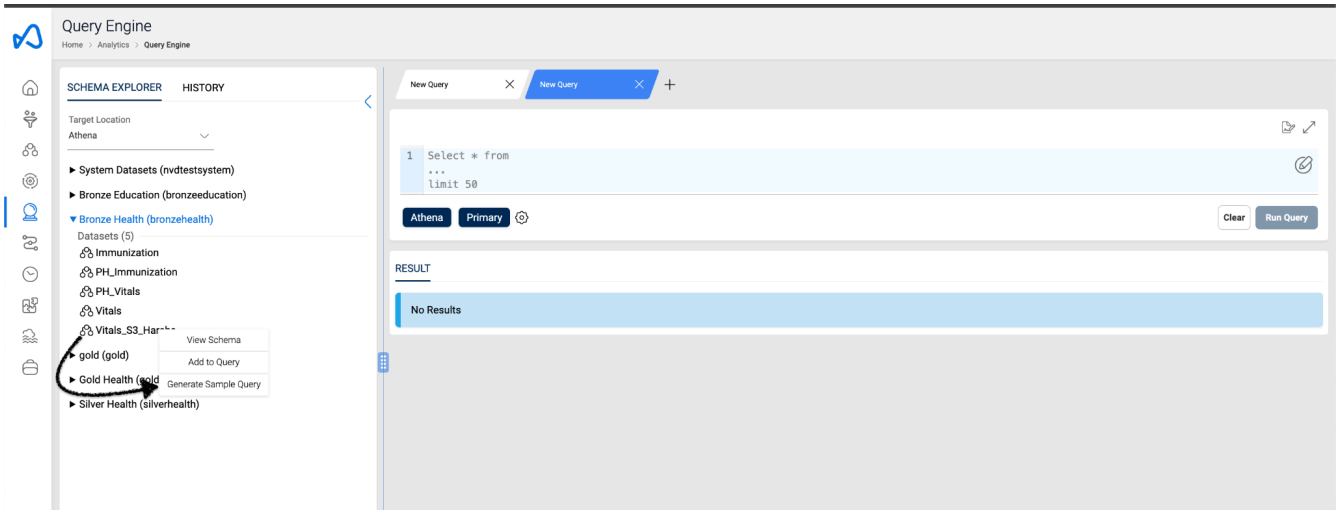


Querying the Ingested S3 Data:

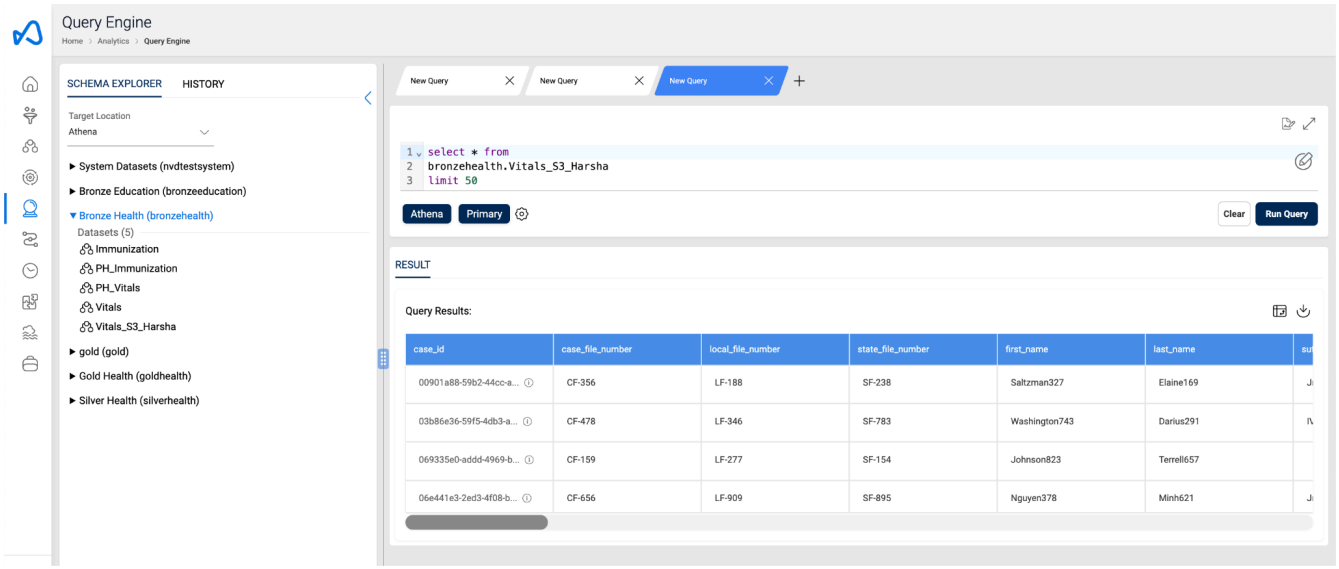
1. Lets query the data we just ingested. Navigate to Analytics > Query Engine and click on it to open.



2. From the Schema Explorer on the left right click on the Dataset you ingested and Click on “Generate Sample Query”.



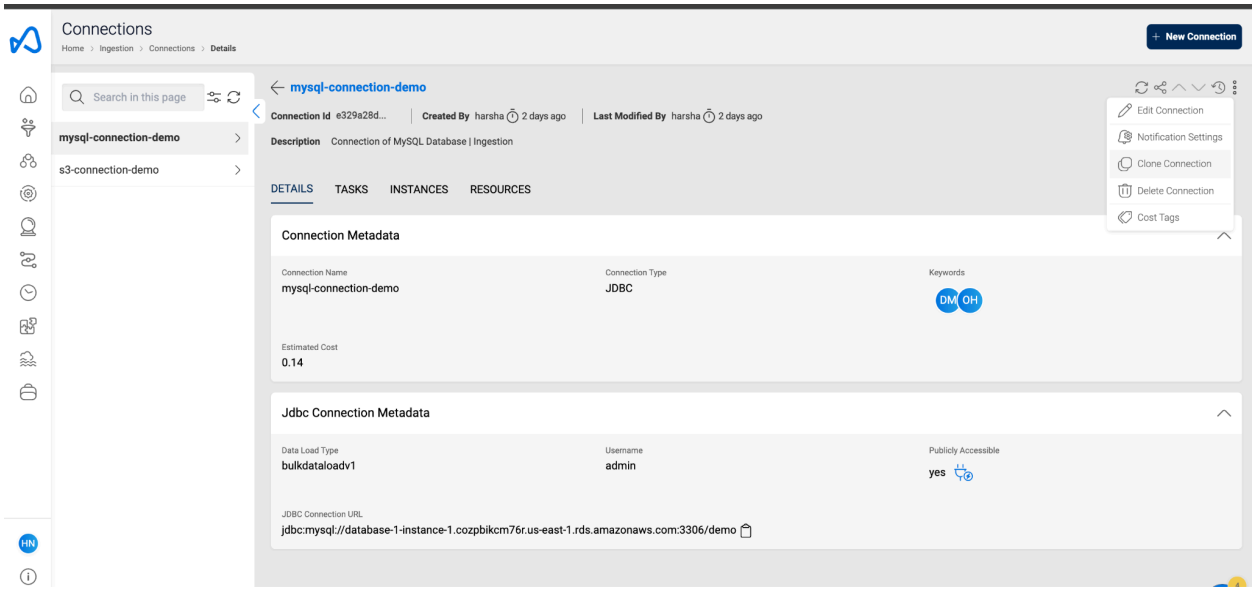
3. The query will be autopopulated to you. Run the query and wait for it to complete. You will have the results displayed.



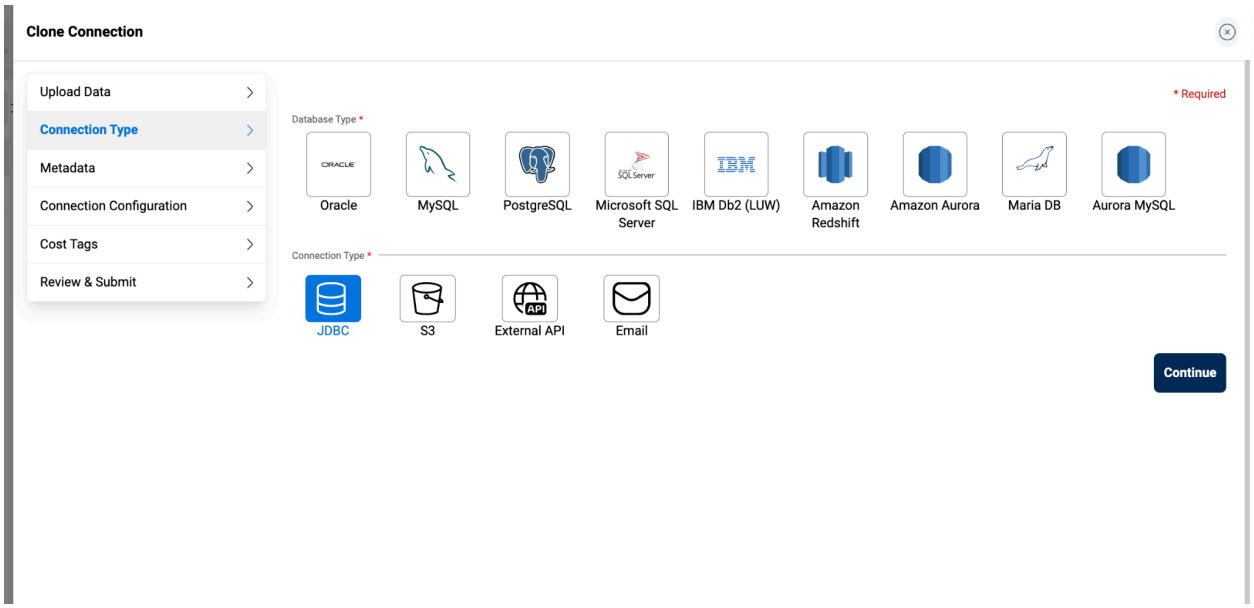
Congrats! You have learnt to Ingest the S3 Data and Query it on Amorphic!

Creating a JDBC Connection

1. Clone the JDBC Connection: mysql-connection-demo by clicking on 3 dots and click on Clone Connection.

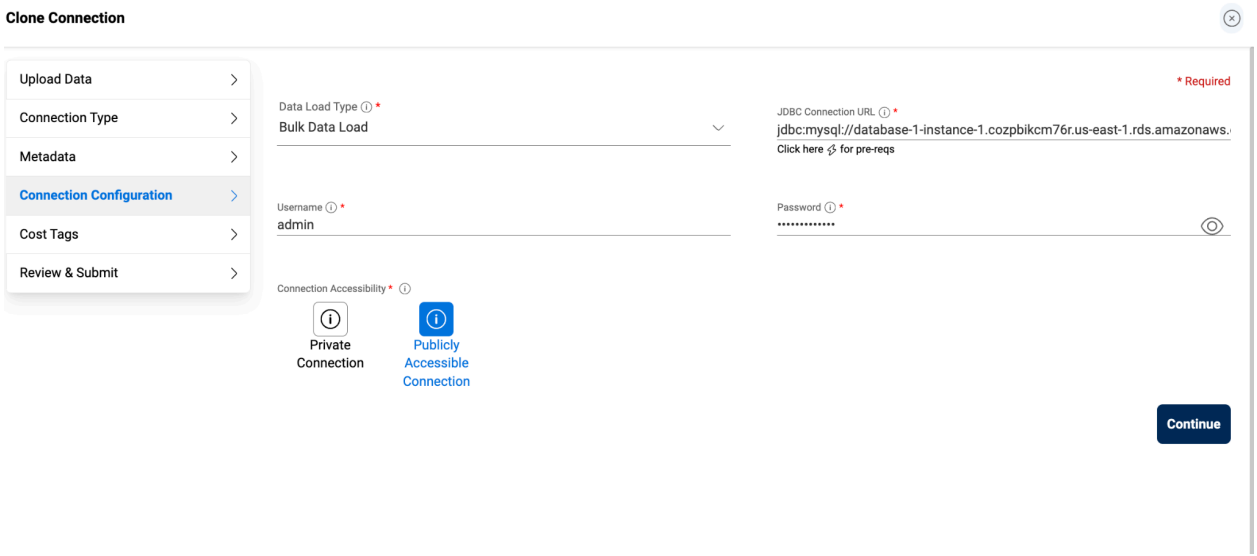


2. Leave the ConnectionType as JDBC and click on Continue.

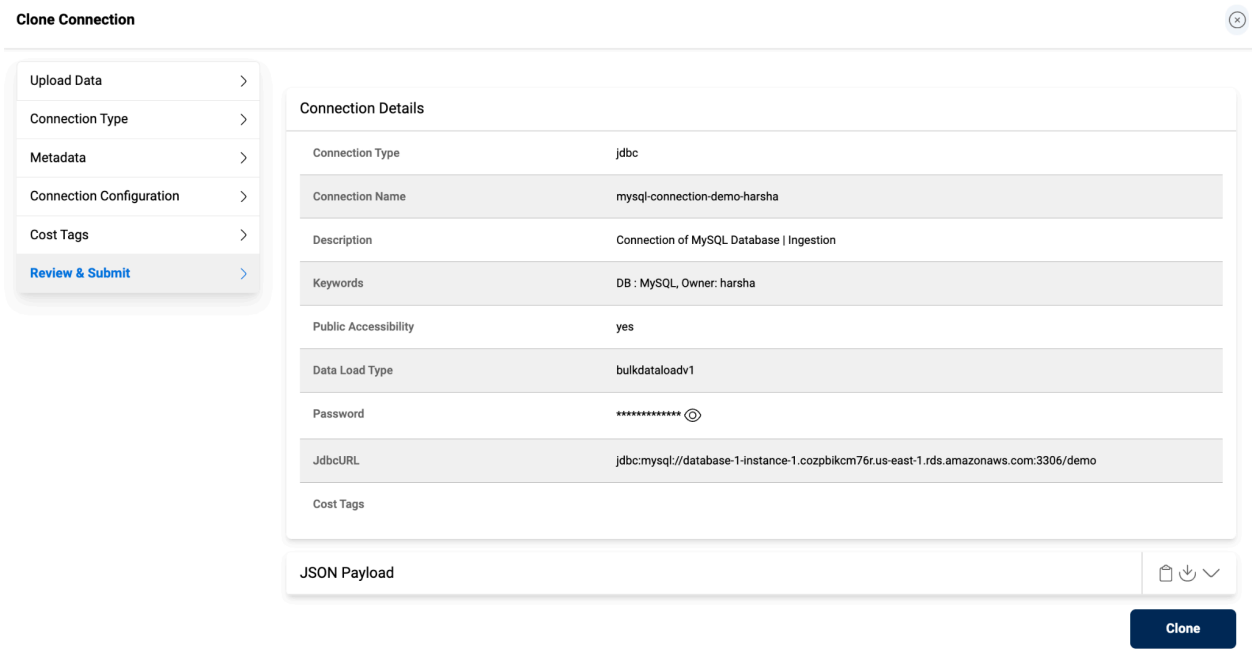


3. Change the name of connection from mysql-connection-demo to mysql-connection-demo-<yourusername> Ex: *mysql-connection-demo-harsha*

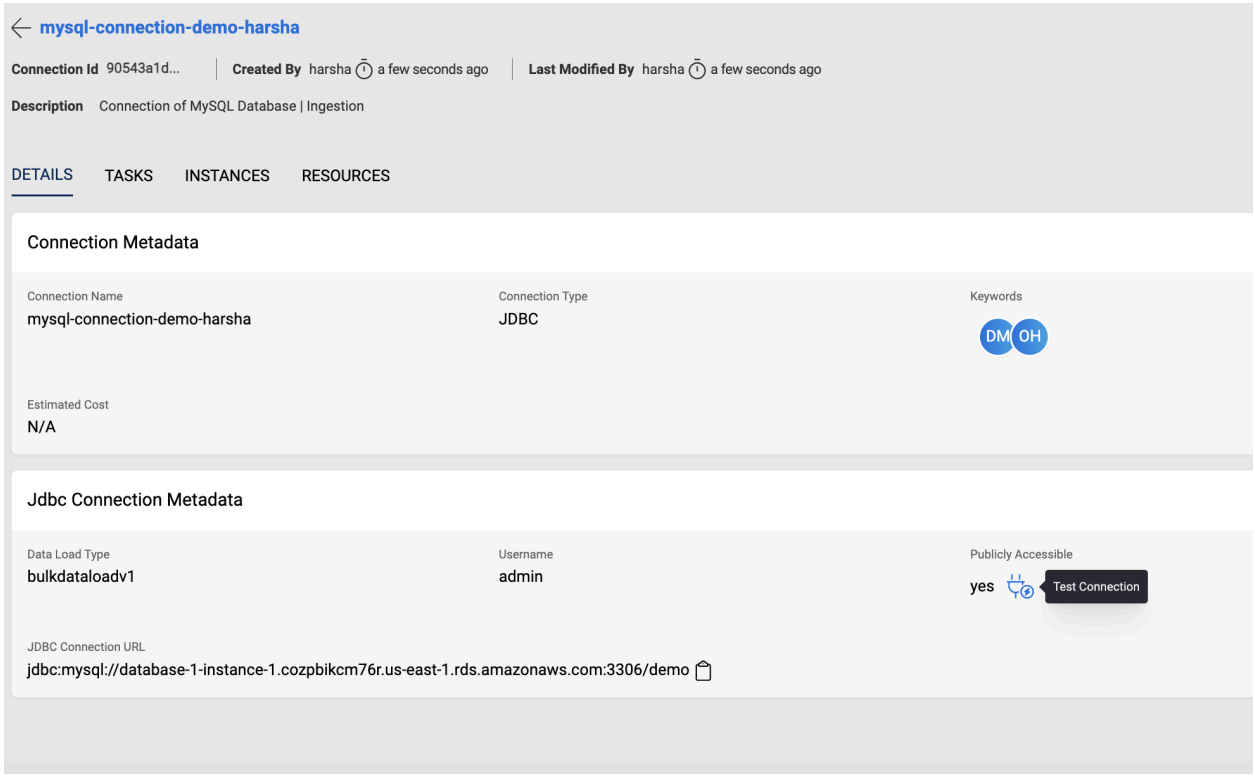
Fill the password: Cloudwick#123 and click on Continue.



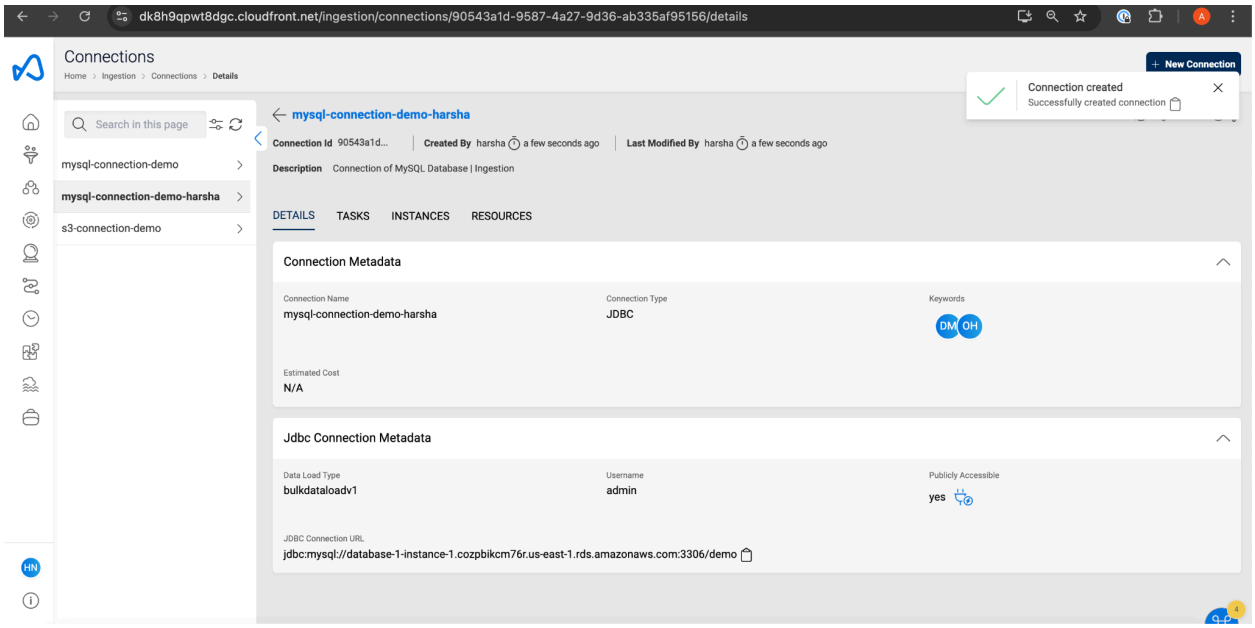
4. Skip Cost Tags, Click on Review and Proceed to Next page and click on Clone.



5. You can test the connection to see its working as below by clicking on test connection



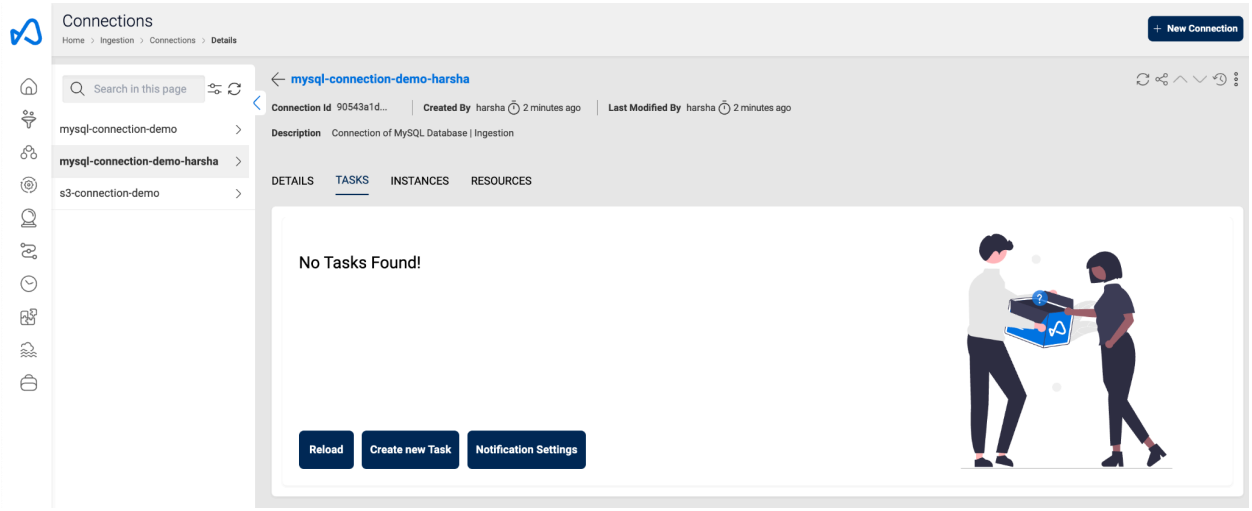
6. You have successfully created a mysql connection.



Tasks:

Tasks automate the data ingestion process in Amorphic.

1. To create a task navigate to Tasks tab and click on Create Task



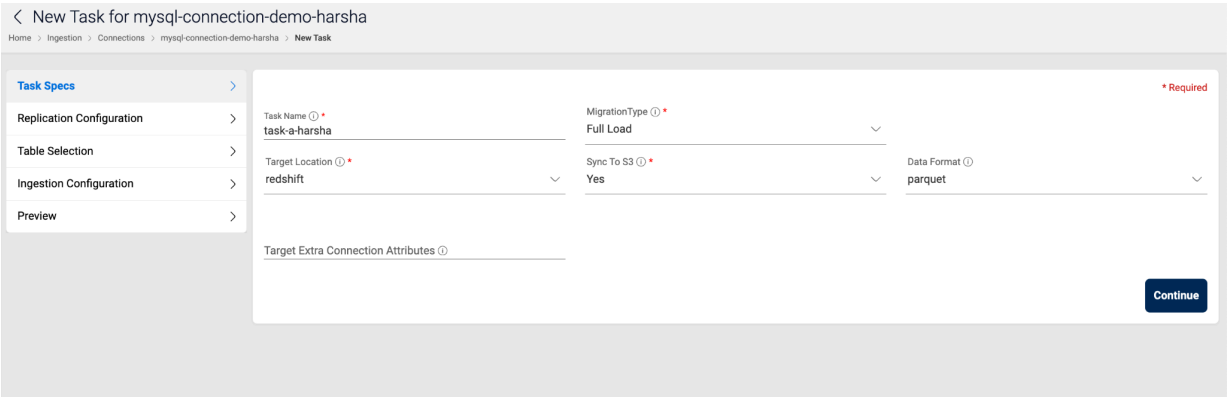
2. Fill the Details for Task Name, Migration Type, Target Location as below and click on continue.

   A. Task Name: task-a-<yourusername>

   B. Migration Type: Full Load

   C. Target Location: s3athena

   D. Data Format: parquet

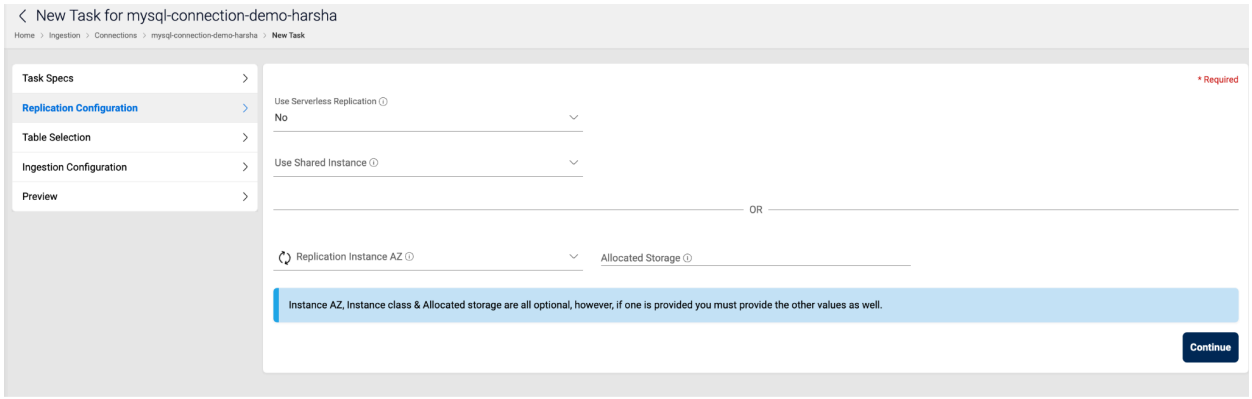


3. Proceed to Replication Configuration.

   For this demo we will create an on demand instance instead of a shared instance. Hence Choose No for Shared Instance and Serverless Instance.

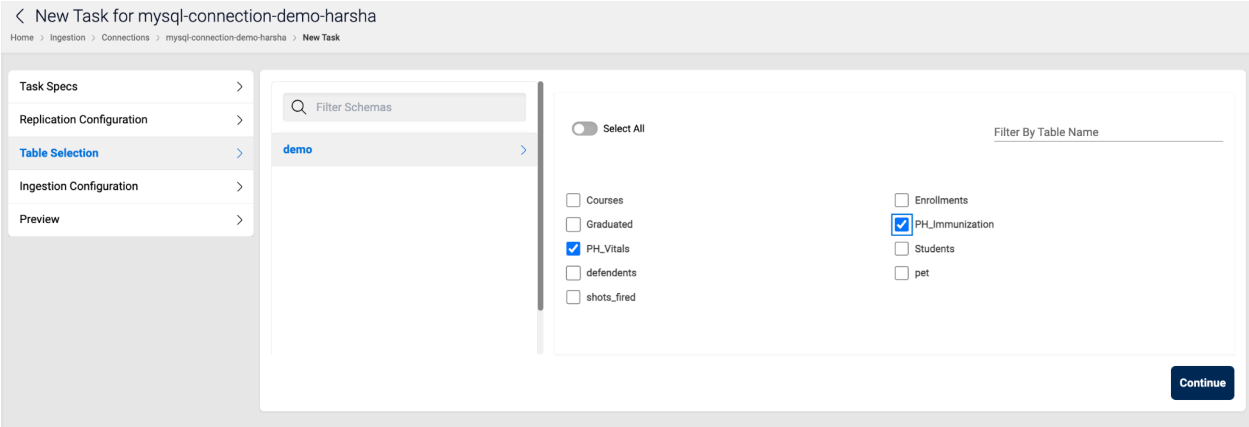
   The right size and storage will be automatically configured based on the table size.

   Hence you can skip filling the details for replication instances.



4. Choose the table you want to Ingest and click on continue.

   Let's ingest one table for the purpose of this demo. Choose the tables PH_Immunization and PH_Vitals and Click on Continue.

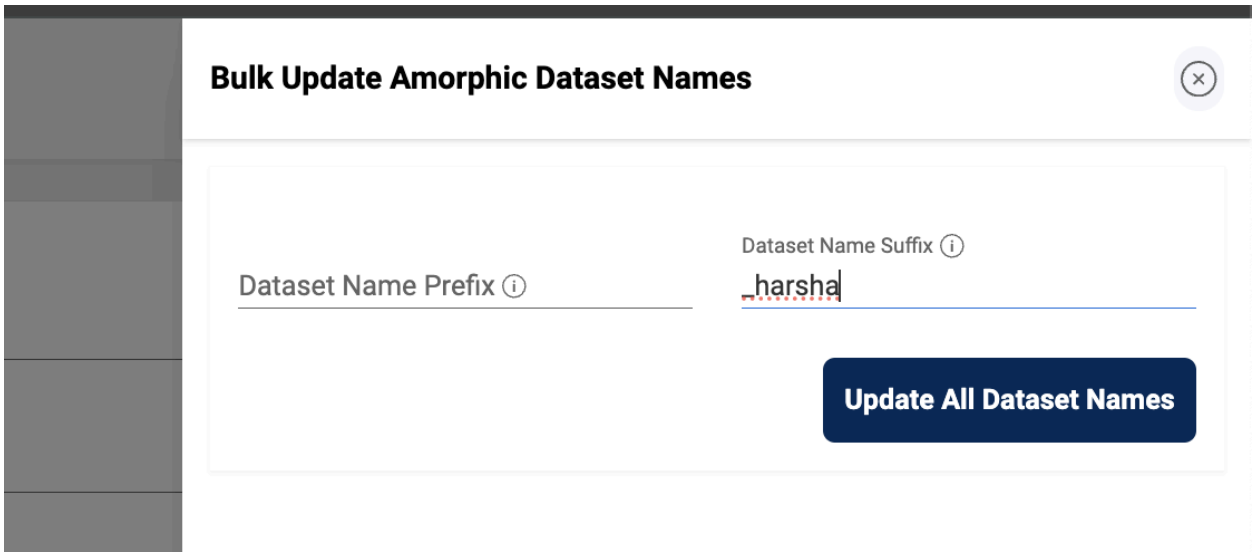


5. Some info is already prefilled for you in the next page.

   Let's put the data into the correct Domain and Change the Dataset Name using Bulk Edit.

   Click on Bulk Edit > Dataset Name > Update the Prefix and Suffix as below.

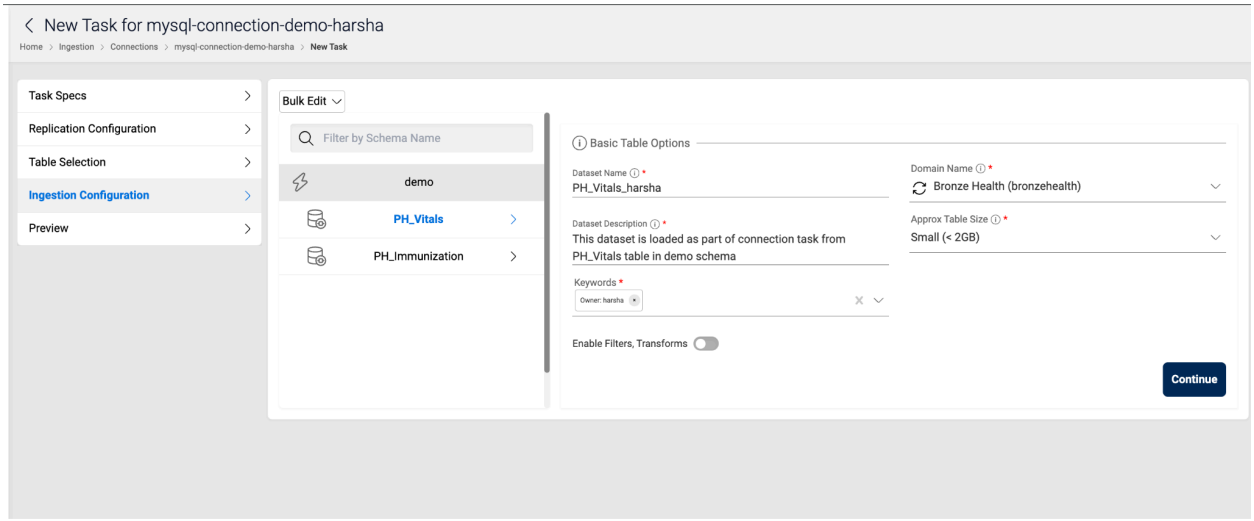
Remove the Dataset Prefix and add _<yourname> to Dataset Suffix and Update All Dataset Names.



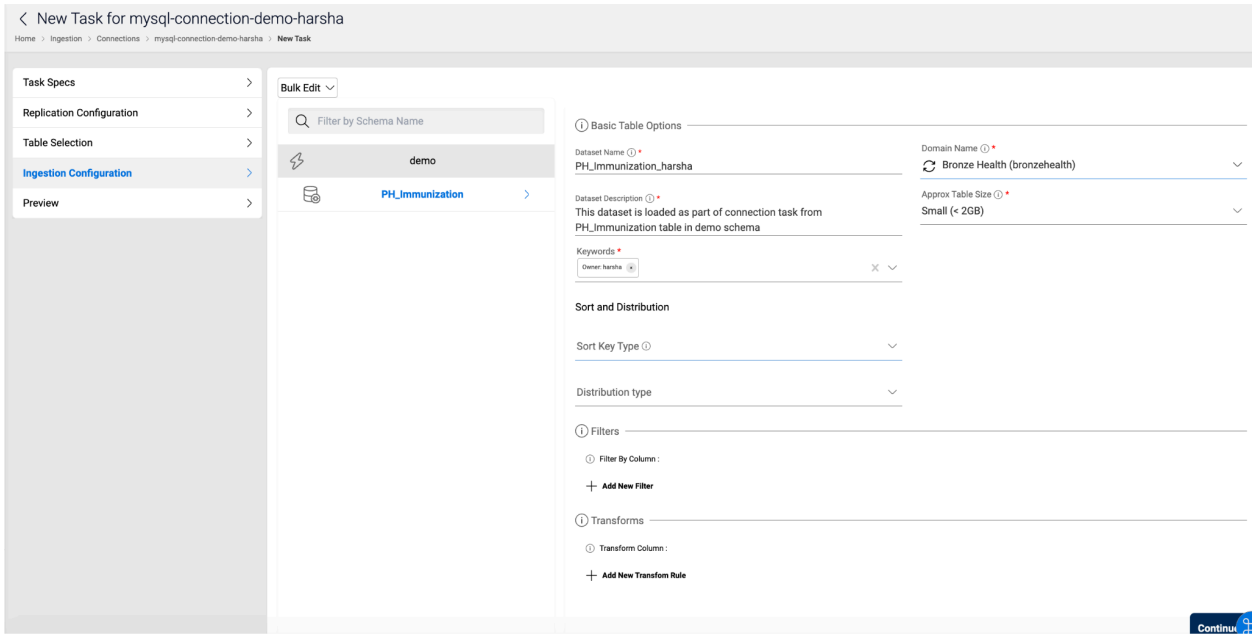
Similarly lets update Domain Name:

Bulk Edit > Domains > Choose the appropriate Domain > Update All Domains.

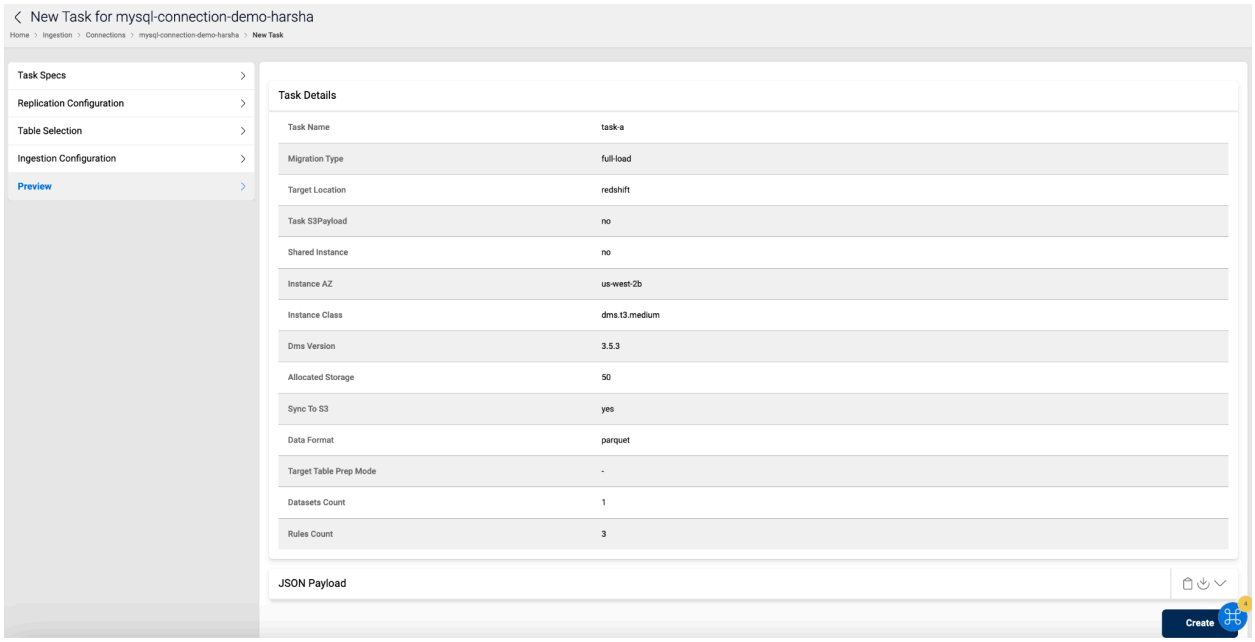
6. Once done, review the changes on the page to see if the updates are reflected and Click on Continue.



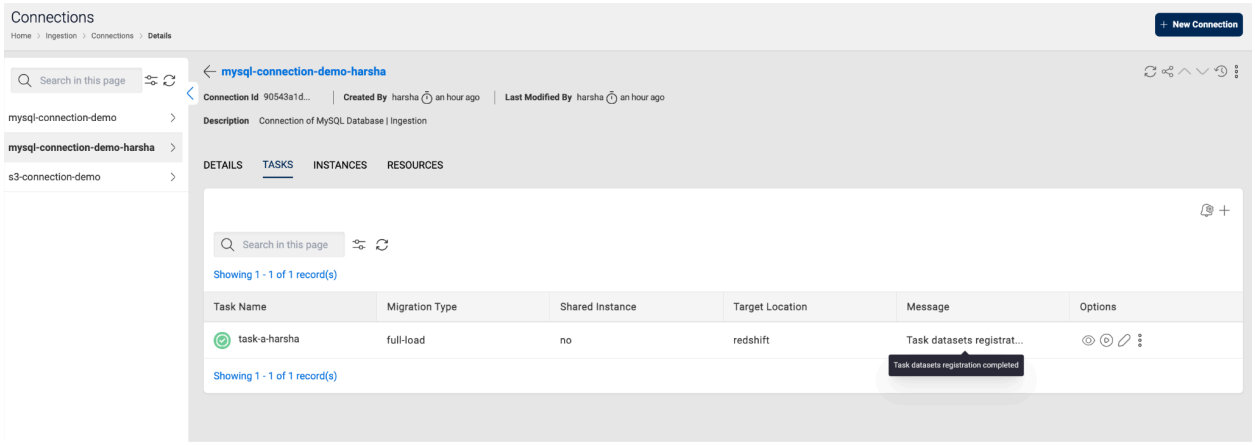
Note: You can also Toggle on Enable Filters, Transforms to filter your data that you are ingesting and Choose any transforms from a list of available filter and transformation rules. (We are skipping this for this demo)



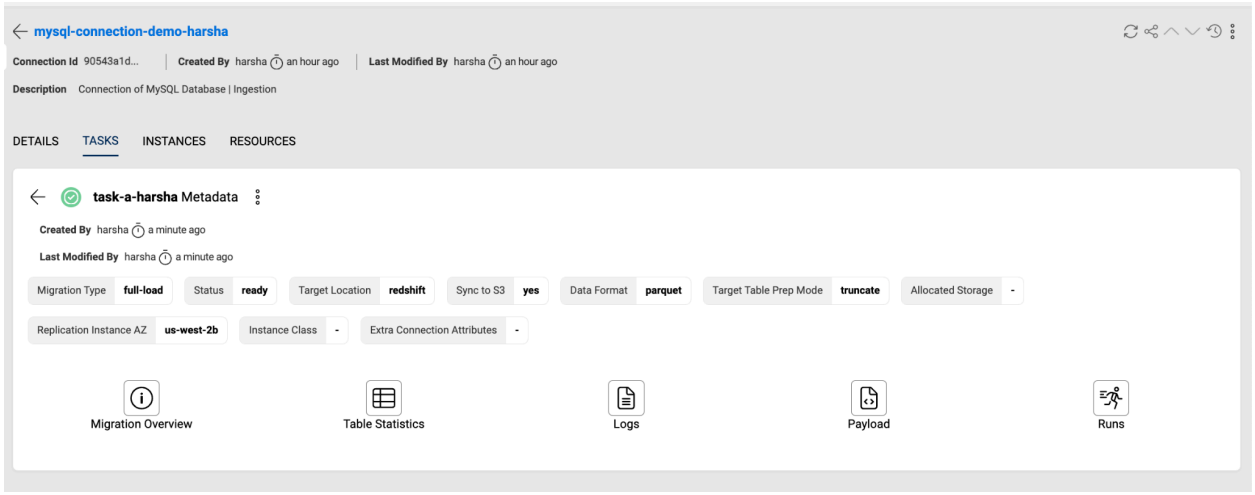
7. Preview the changes and click on Create.



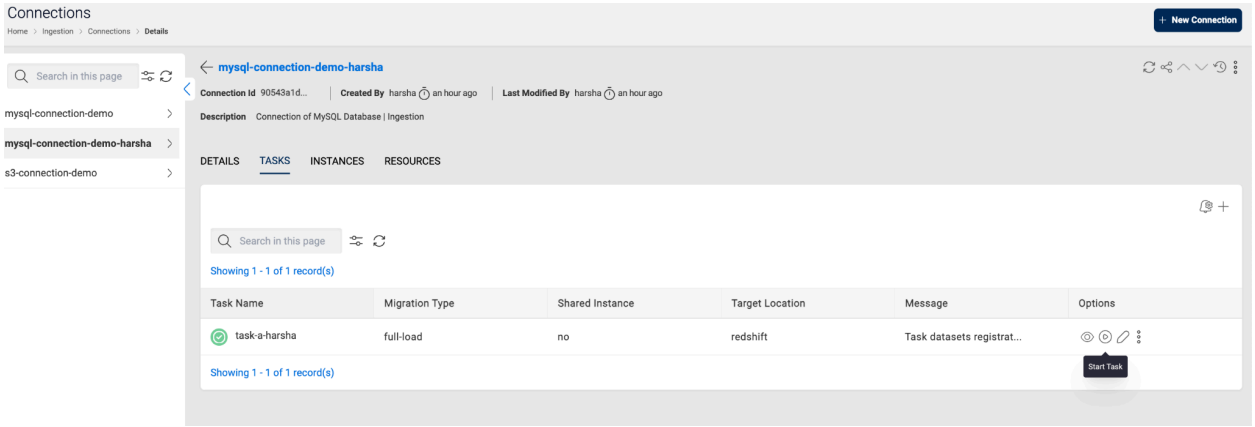
8. Back to the main page you will see the Task Creation has been successfully completed after a min.



9. You can click on it to expand and see the properties you configured.



10. Go back to the Tasks and start the task to ingest data.



This process would take some time depending on the size of the tables.

11. You have now learnt to create JDBC connections and Bulk Load Tables using DMS Task on Amorphic. Cheers!

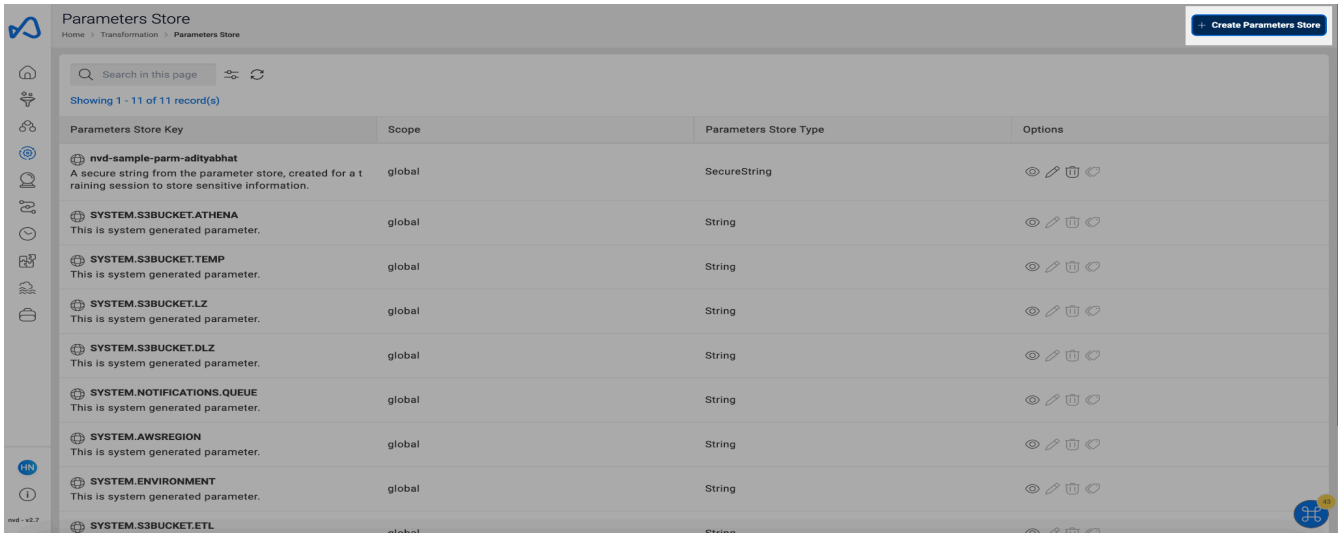
LAB Session 3:

This is focused for Data Engineers

Paramstore:

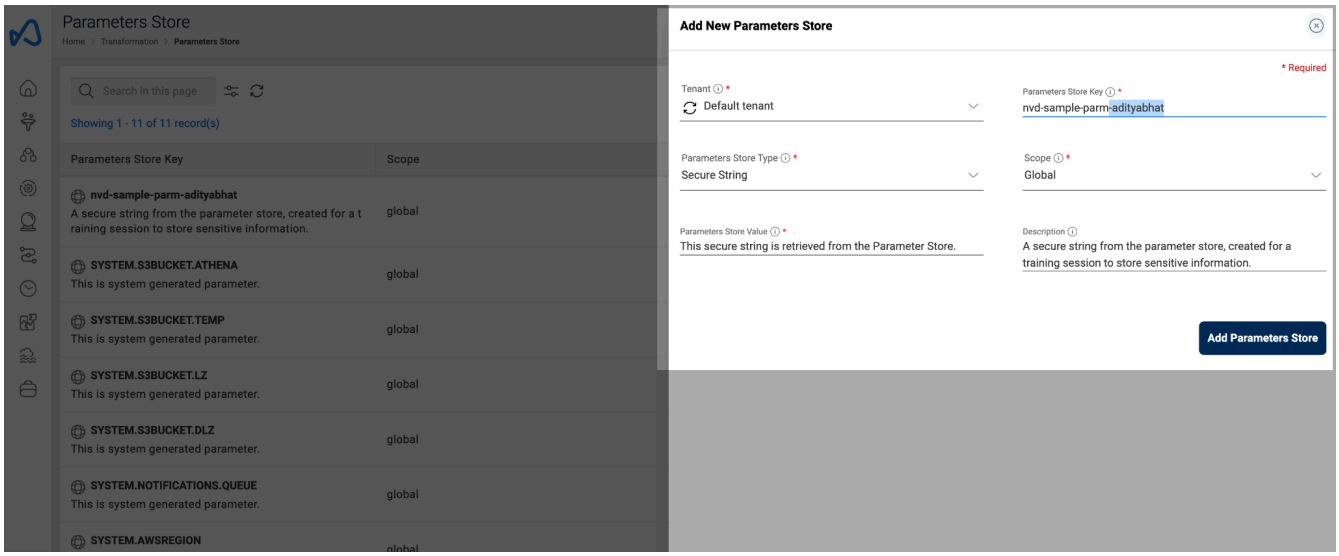
In today's session let's start by creating a Parameters Store key.

1. Create a new parameter in the Parameter Store by navigating to the Parameter Store section under Transformations.(Here, you'll also find some default system-generated parameters, such as bucket names, AWS region, etc.)



2. Name the parameter **sample-param-<yourusername>**. Ex: **nvd-sample-param-adityabhat**. Choose Parameters Store Type as Secure String, Scope as Global, Fill in Parameter Store Value.

3. Once you've filled in the details, click on **Add Parameter Store**.



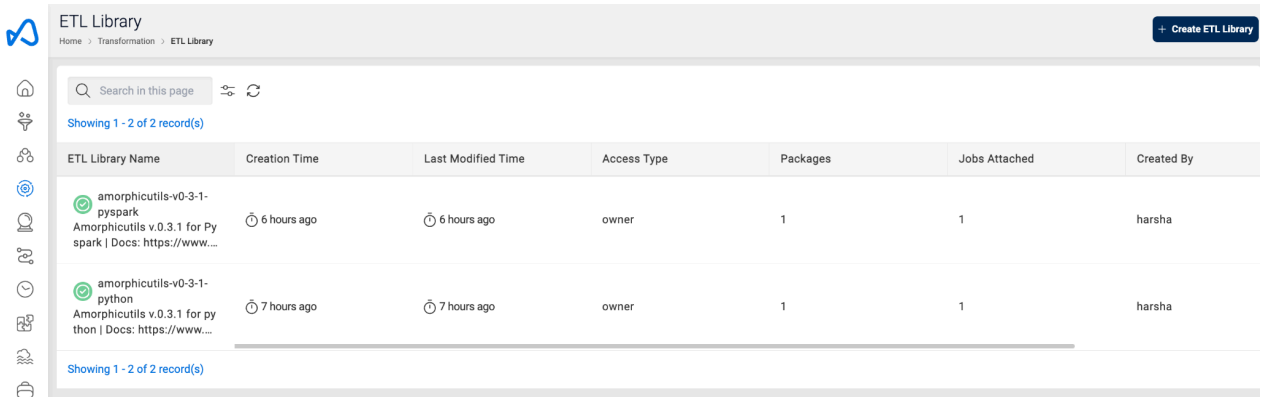
Congratulations! You have now learned how to create a parameter in the Parameter Store.

ETL Library:

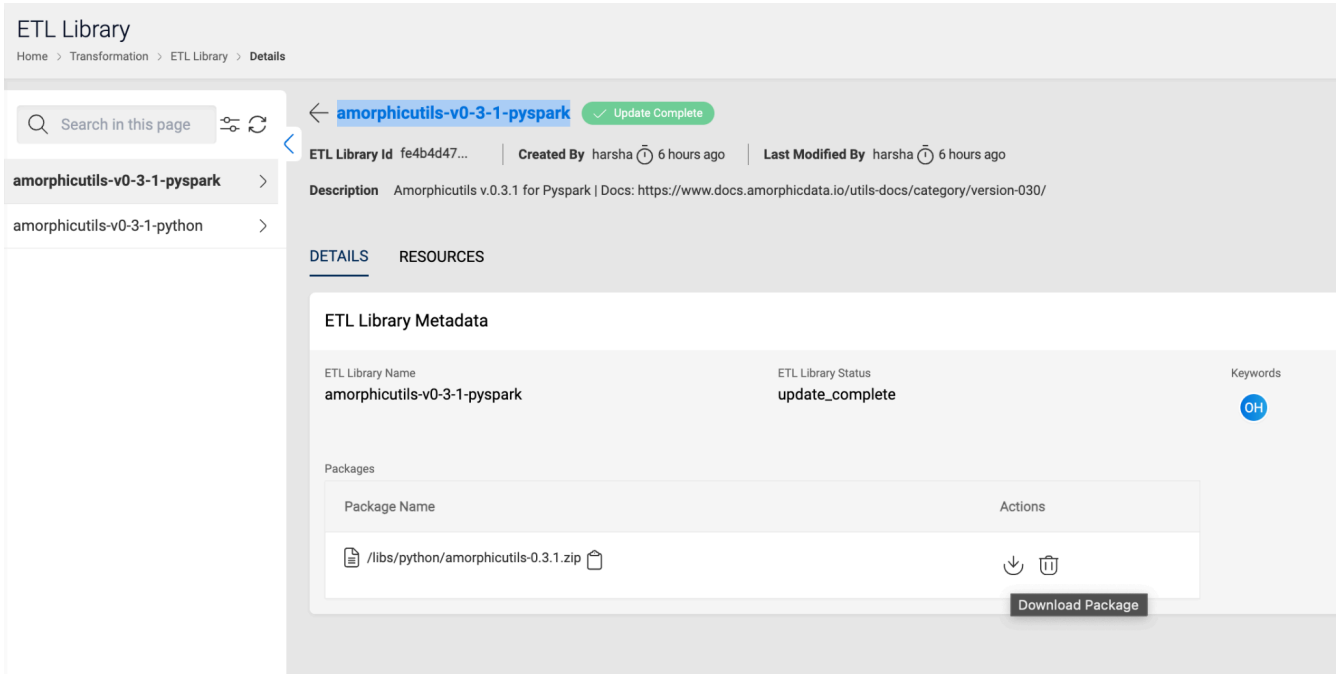
1. Navigate to the ETL Library page. Transformations > ETL Library
2. Let's take a look at ETL Library as a reference.

A. amorphicutils-v0-3-1-pyspark

B. amorphicutils-v0-3-1-python

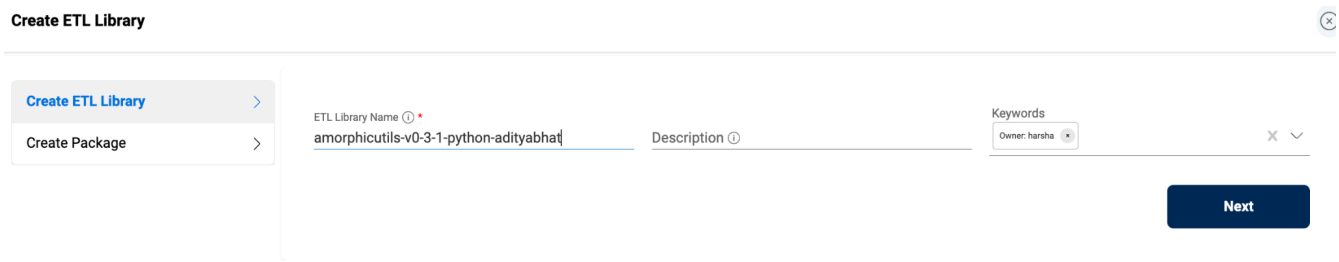


3. Let's download the packages for both the libraries (pyspark and python).

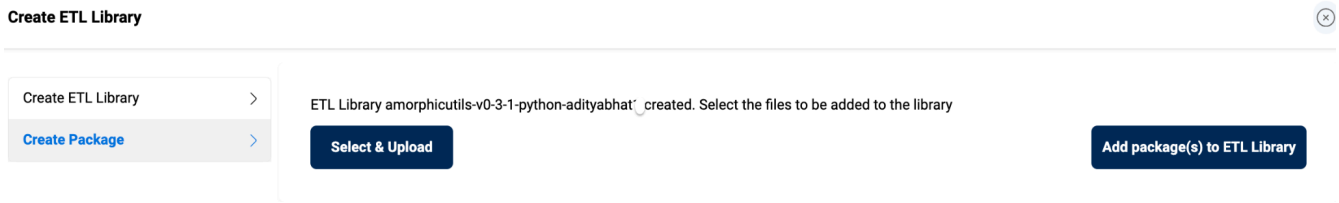


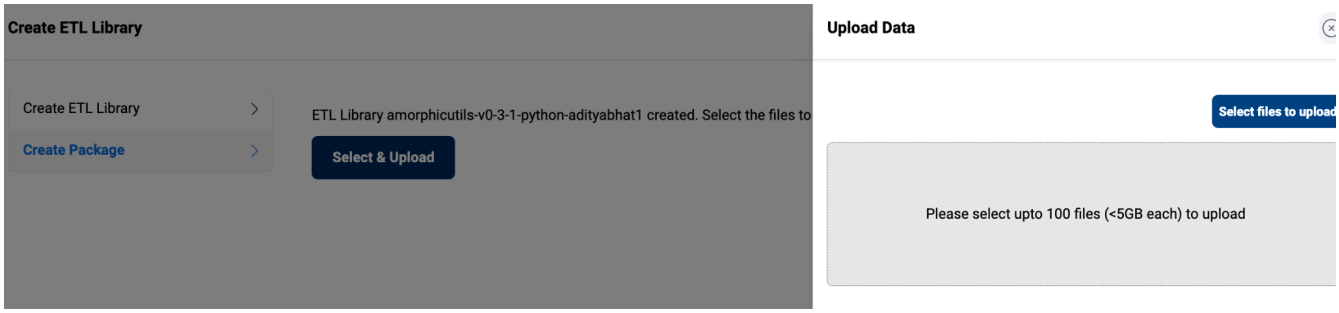
Once downloaded we can use them to create our own Shared ETL library.

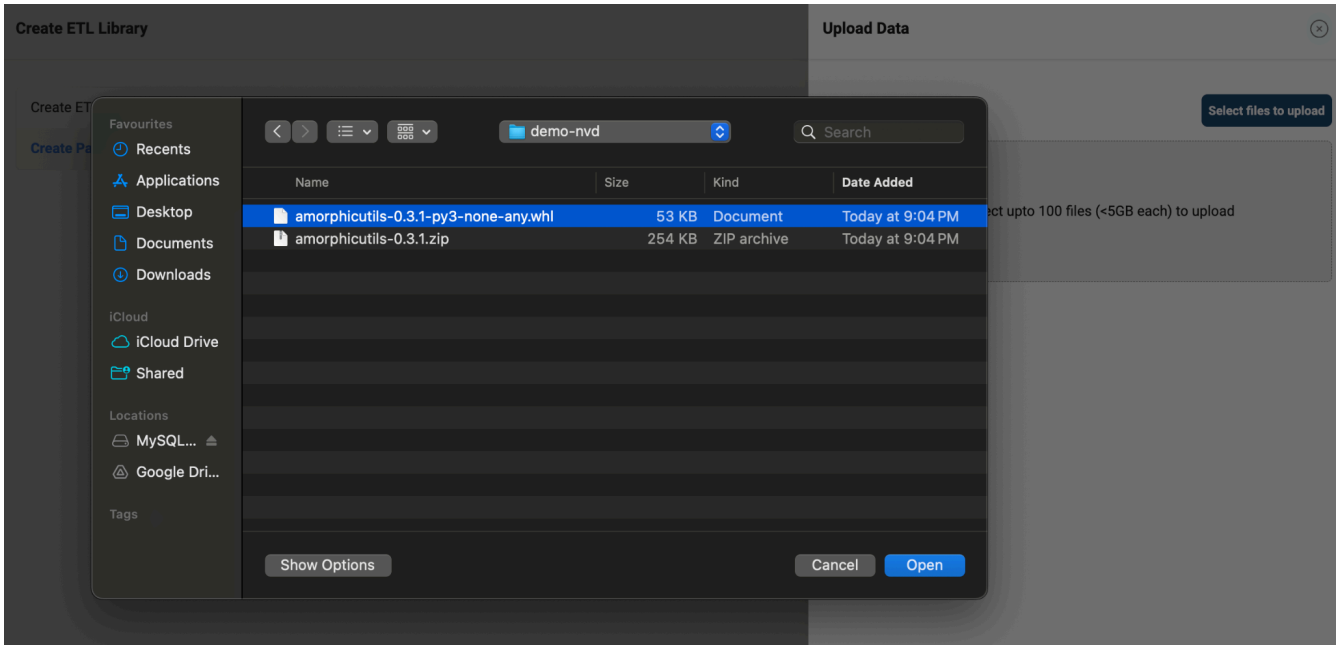
4. Click on create ETL Library. Name it **`as amorphicutils-v0-3-1-python`**. Example: **`amorphicutils-v0-3-1-python-adityabhat`**.



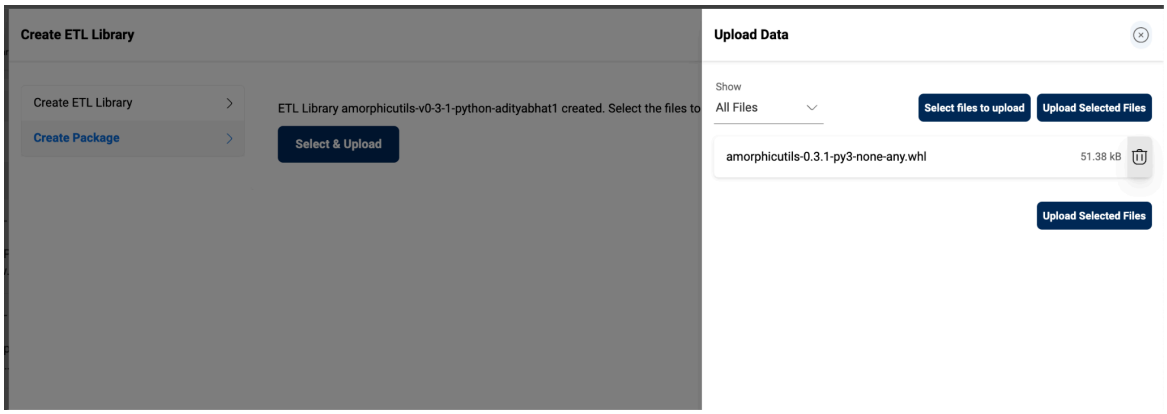
5. Click on next and then select the package from local to add the package you downloaded.
   Note that in this example we are creating a python library hence uploading the .**whl** file (to be used in pythonshell job) Follow below screenshots.



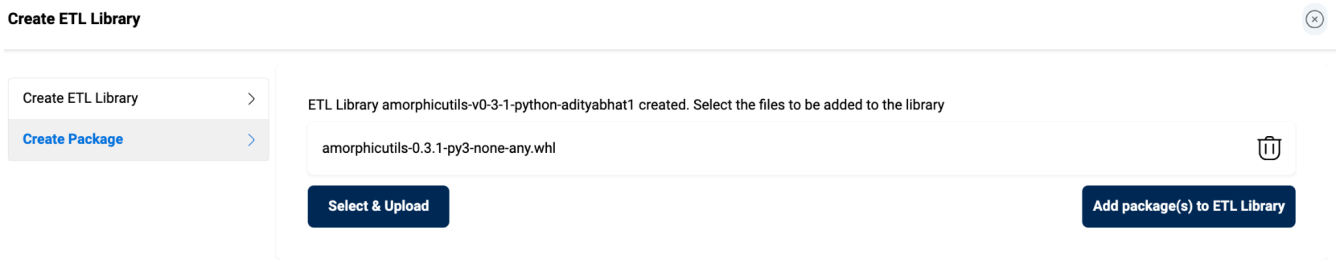




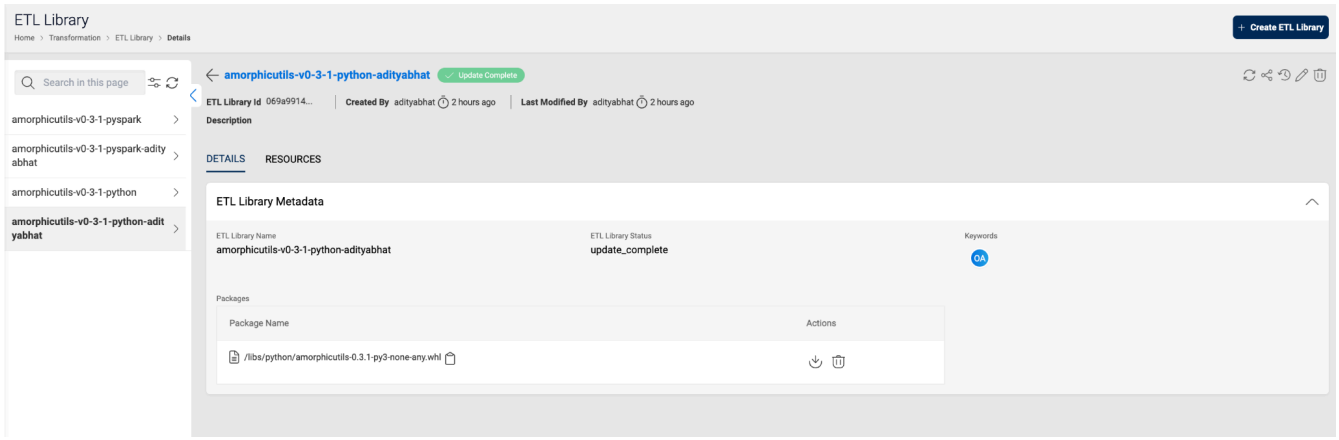
6. Click on Upload Selected Files.



7. Proceed further by clicking on Add packages(s) to ETL Library.



8. Once successful you are taken back to the main page and you can see the status of the library as Update Complete. (If you don't see it reload the page)



You have now successfully created the ETL Library.

Repeat the steps from 4 to 8 for creating the pyspark library. Use **`amorphicutils-0.3.1.zip`** file for upload.

ETL Job:

Let's create an ETL job to Read from Dataset, Apply simple Transformations and Write to Dataset.

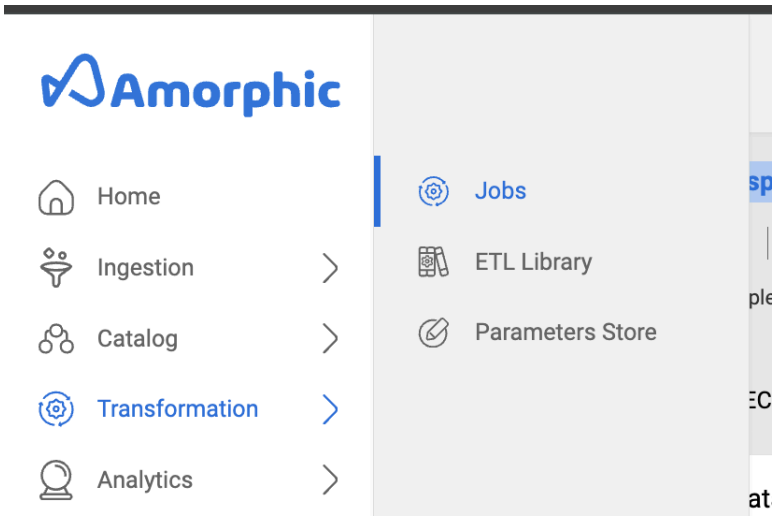
Pre- Requisites:

Make sure you have access to the below Datasets.

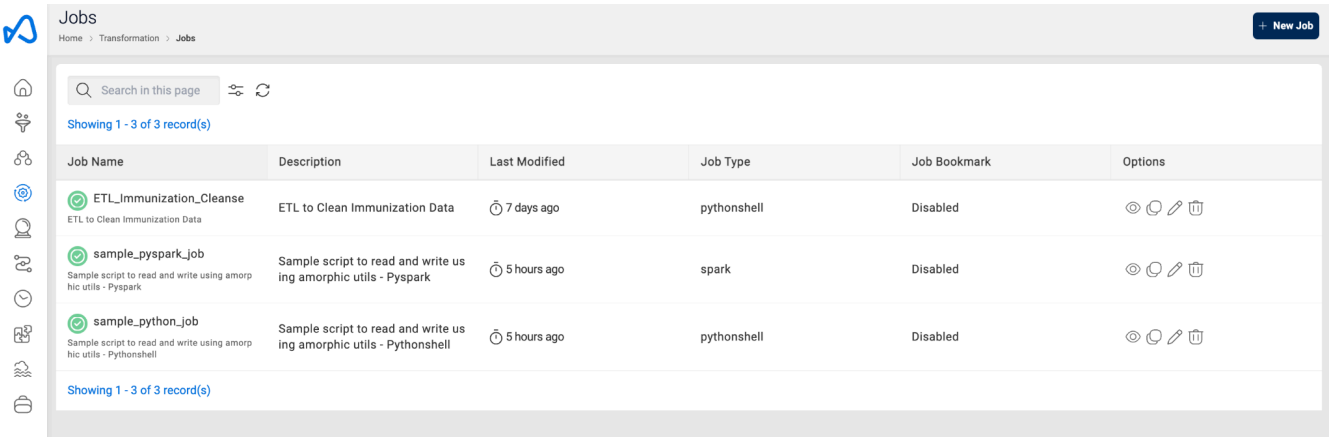
a. ***Immunization*** (Domain: bronzehealth, Read Access)
b. ***Immunization_Staged*** (Domain: silverhealth , Write Access)

Procedure:

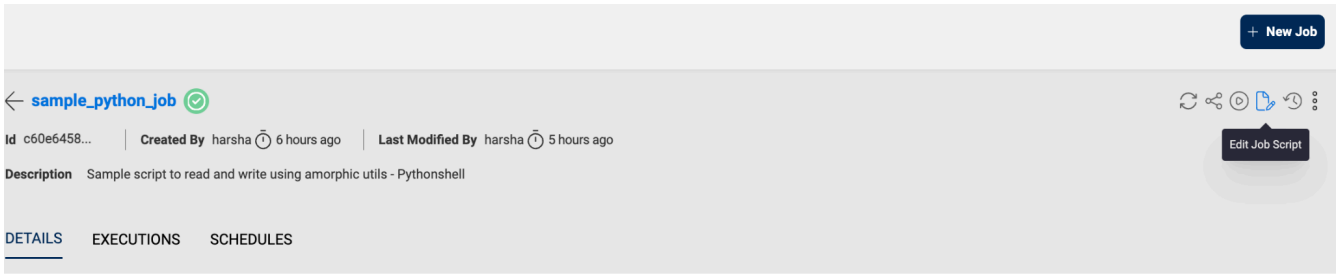
1. Navigate to ETL Jobs from the left pane. Transformation > Jobs



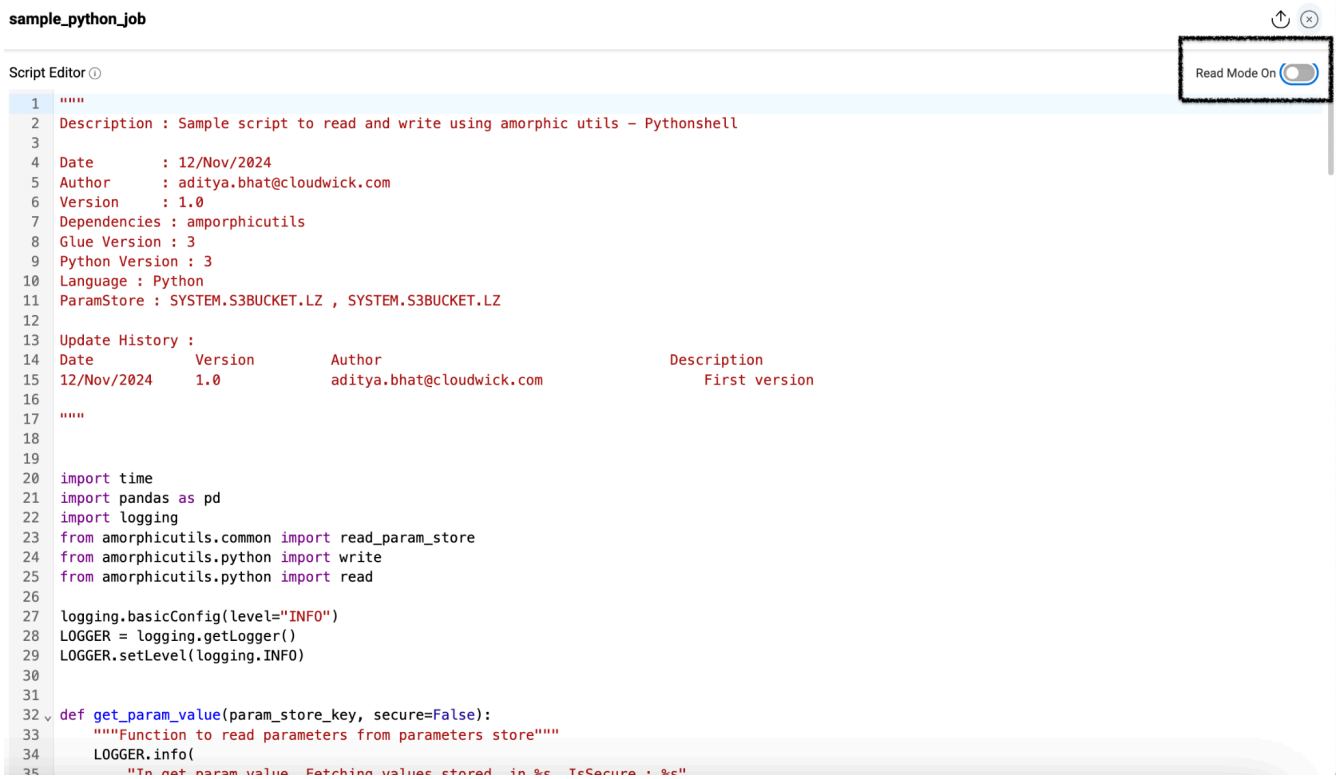
2. You should already see jobs ***sample_pyspark_job*** and ***sample_python_job*** shared with you.



3. Let's copy the script for our reference which we can use later in our lab.
   Select the job ***sample_python_job*** and click on Edit Script.



4. Deselect the toggle for Read Mode On, Select all and copy it into your editor.

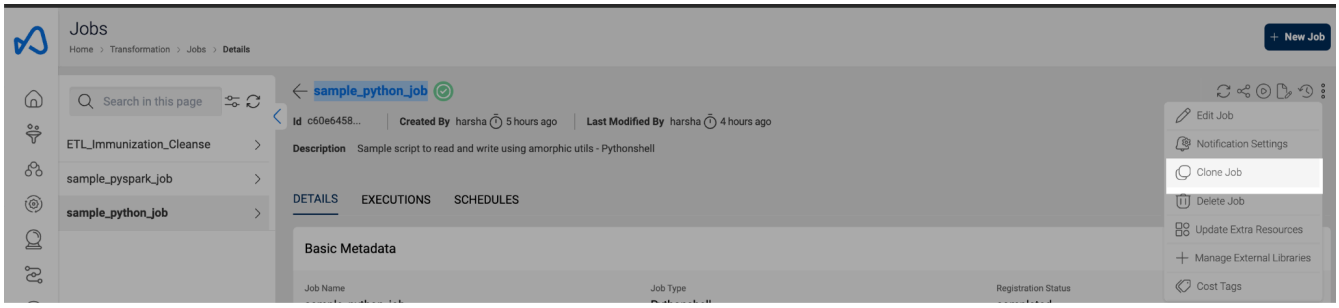


5. You need to make a few updates to the script.
   A. Replace the user_id with your login user id. (Line no 62 from below SS)
   B. Replace the param_store_key with the one you created. (Line no 81 from below SS).
   C. You can replace your read/write domain and read/write dataset if you wish. Let's skip this for this demo.

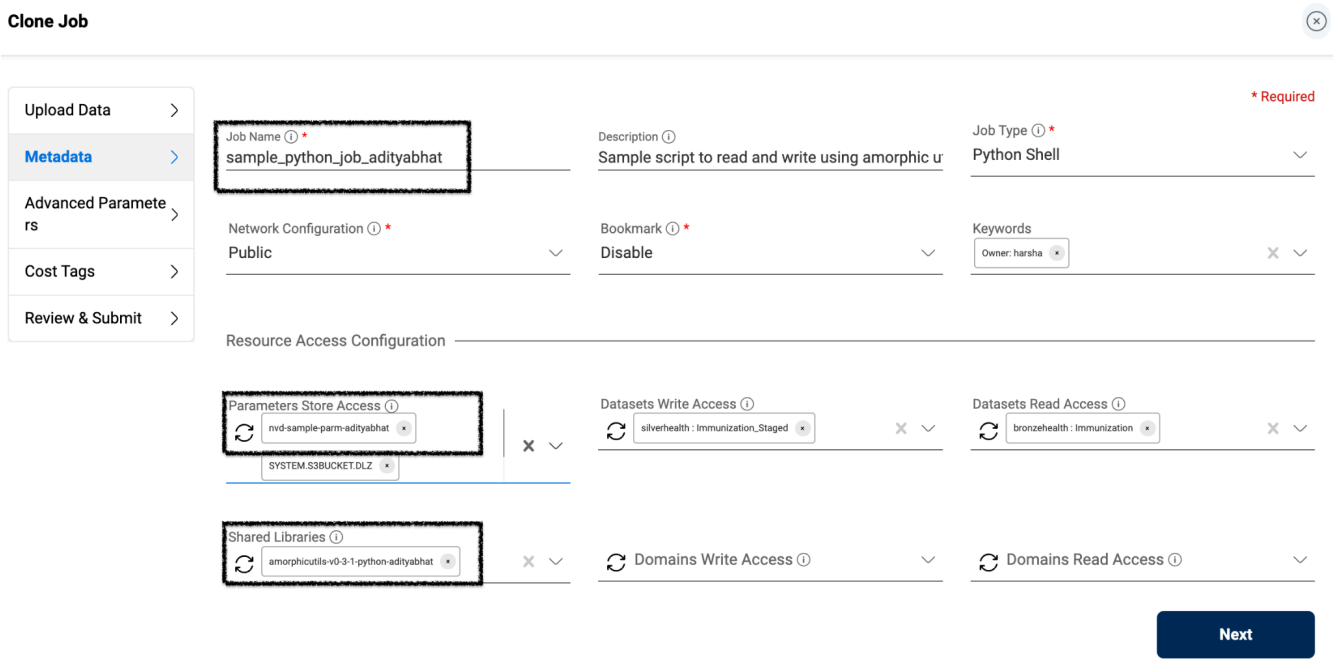
```python
def main():
    """
    Main
    """

    LOGGER.info("In Main,")

    user_id = "adityabhat"  # Your Login UserId with access to write

    # Replace below if you have different Domain to to Read from
    r_domain = "bronzehealth"
    # Replace this if you have different Domain to write into.
    w_domain = "silverhealth"
    # Replace this if you created new Dataset to Read from.
    r_dataset = "Immunization"
    # Replace this if you created new Dataset to write into.
    w_dataset = "Immunization_Staged"

    # LZ Bucket is Output Bucket
    # lz_bucket = "nvd-us-west-2-816069158041-test-lz"
    # DLZ Bucket is Input Bucket
    # dlz_bucket = "nvd-us-west-2-816069158041-test-dlz"

    lz_bucket = get_param_value(param_store_key="SYSTEM.S3BUCKET.LZ")
    dlz_bucket = get_param_value(param_store_key="SYSTEM.S3BUCKET.DLZ")
    my_param = get_param_value(
        param_store_key="nvd-sample-parm-adityabhat", secure=True
    )  # Replace with your paramstore name
    print(my_param)

```

Fig: Sample Snippet of the ETL Code.

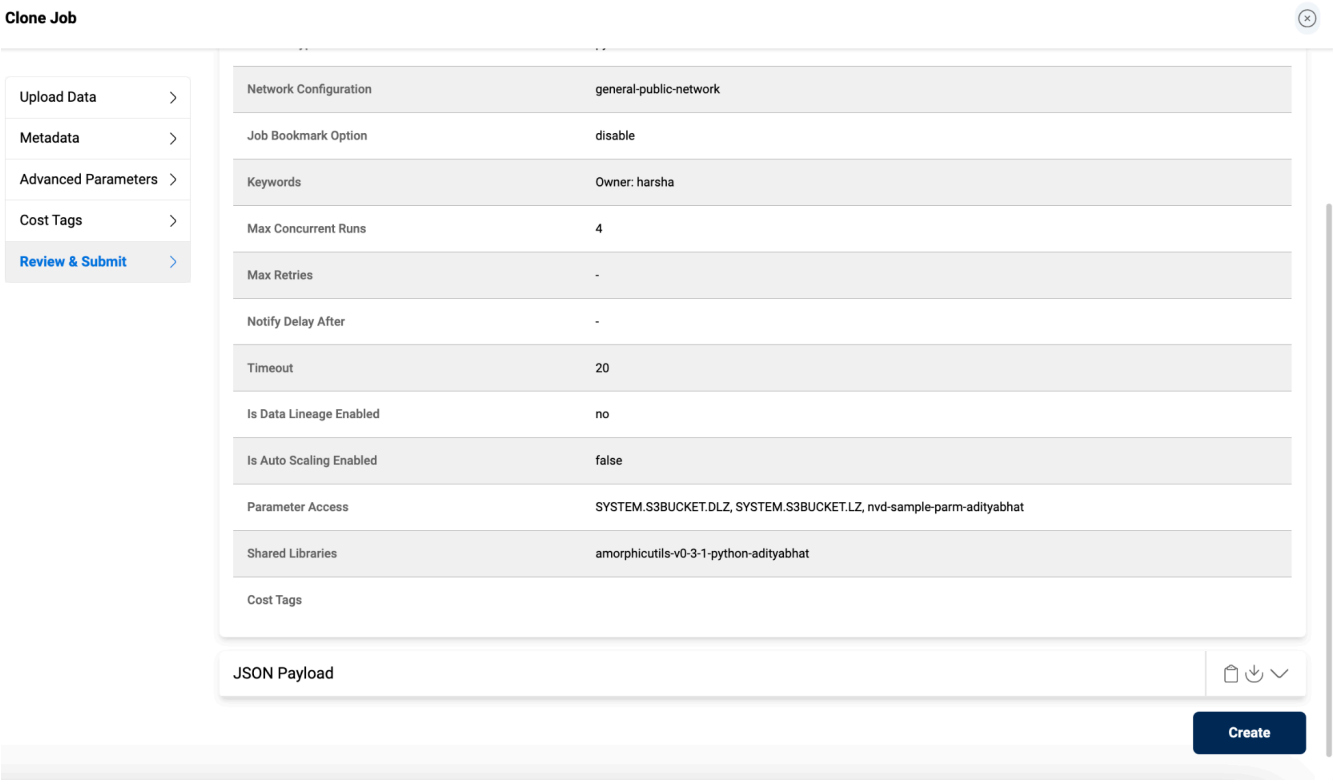
6. Back to Amorphic UI. Clone the ***sample_python_job*** by clicking on three dots on the right corner of the page and clone.



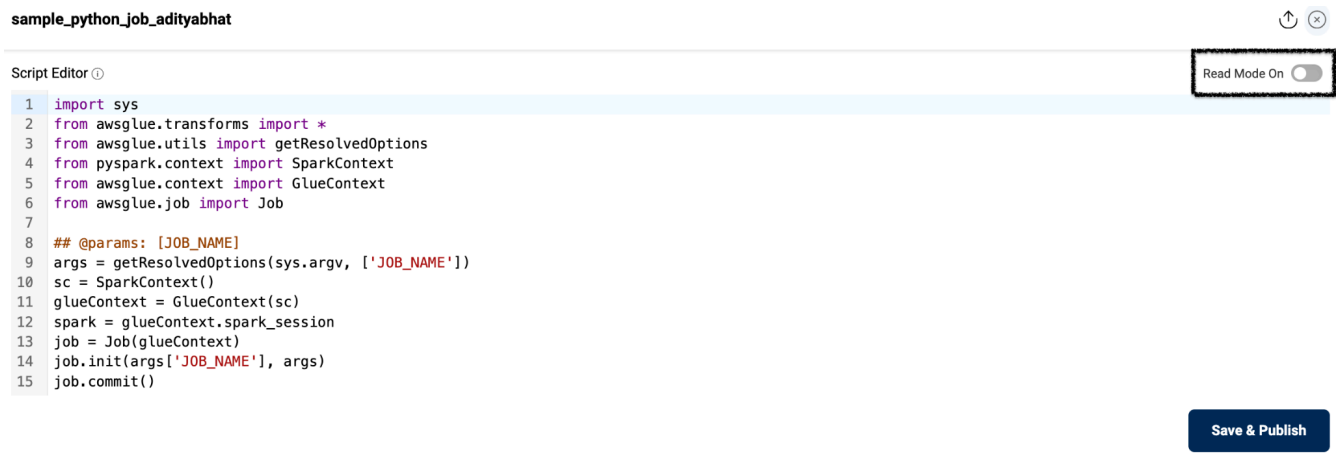
7. Update the **Job Name** with *sample_python_job_<yourusername>*. Example *sample_python_job_adityabhat*
   Remove the existing shared library and update it with the one you created for python.
   Add the new paramstore key you created to the Job.
   Refer below screenshot for more info.



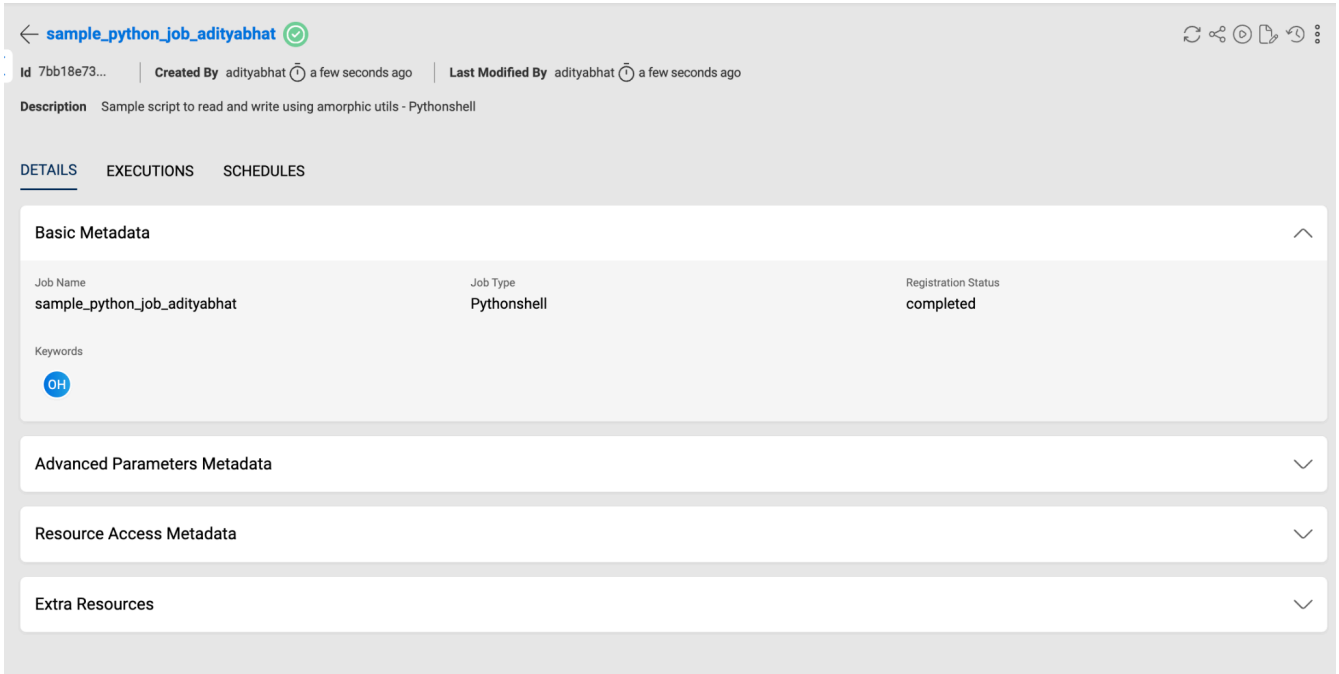
8. Skip the Advanced Configuration and Cost Tags. Review the Job and click on create.



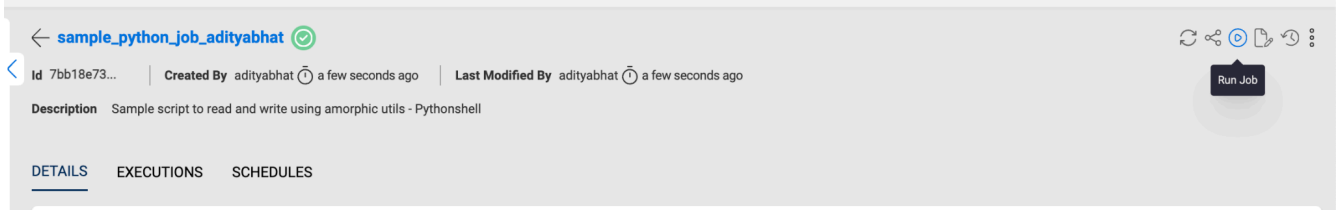
9. You will be taken to the script editor. Deselect the Read Mode On toggle and update your script with the one you made changes previously. Once done click on **Save & Exit**



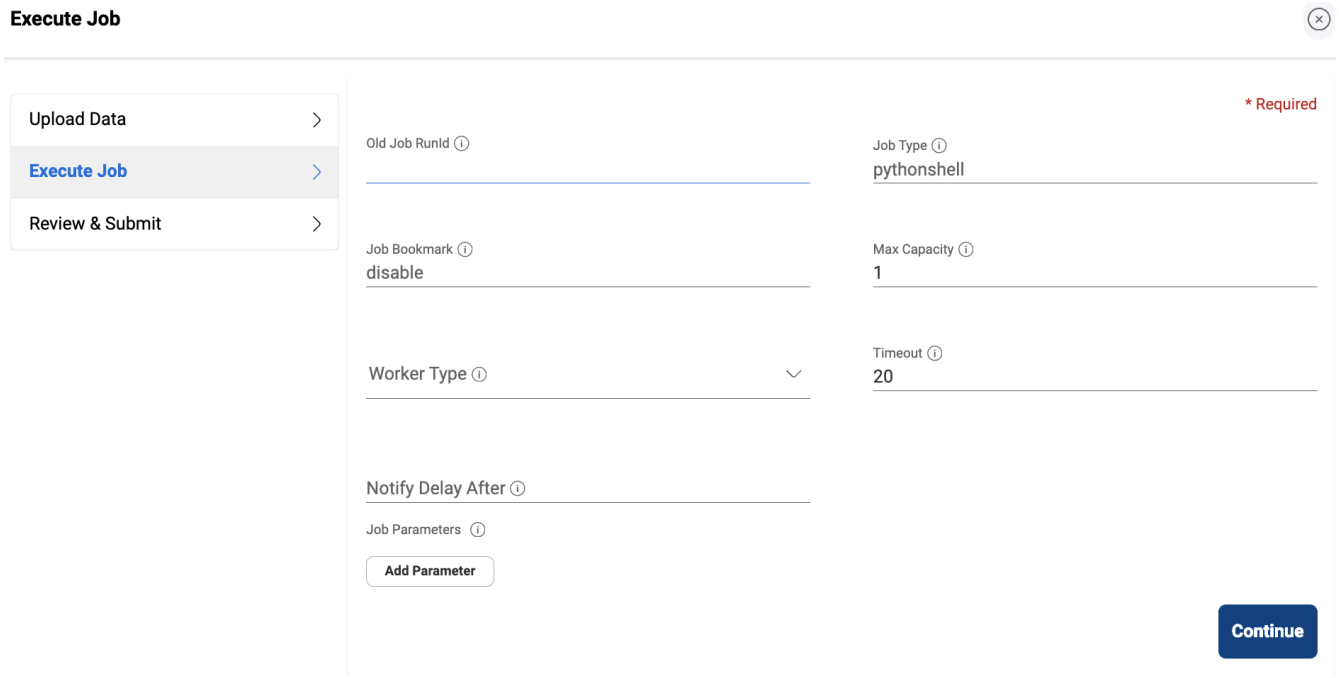
10. You have now successfully created the pythonshell Job.

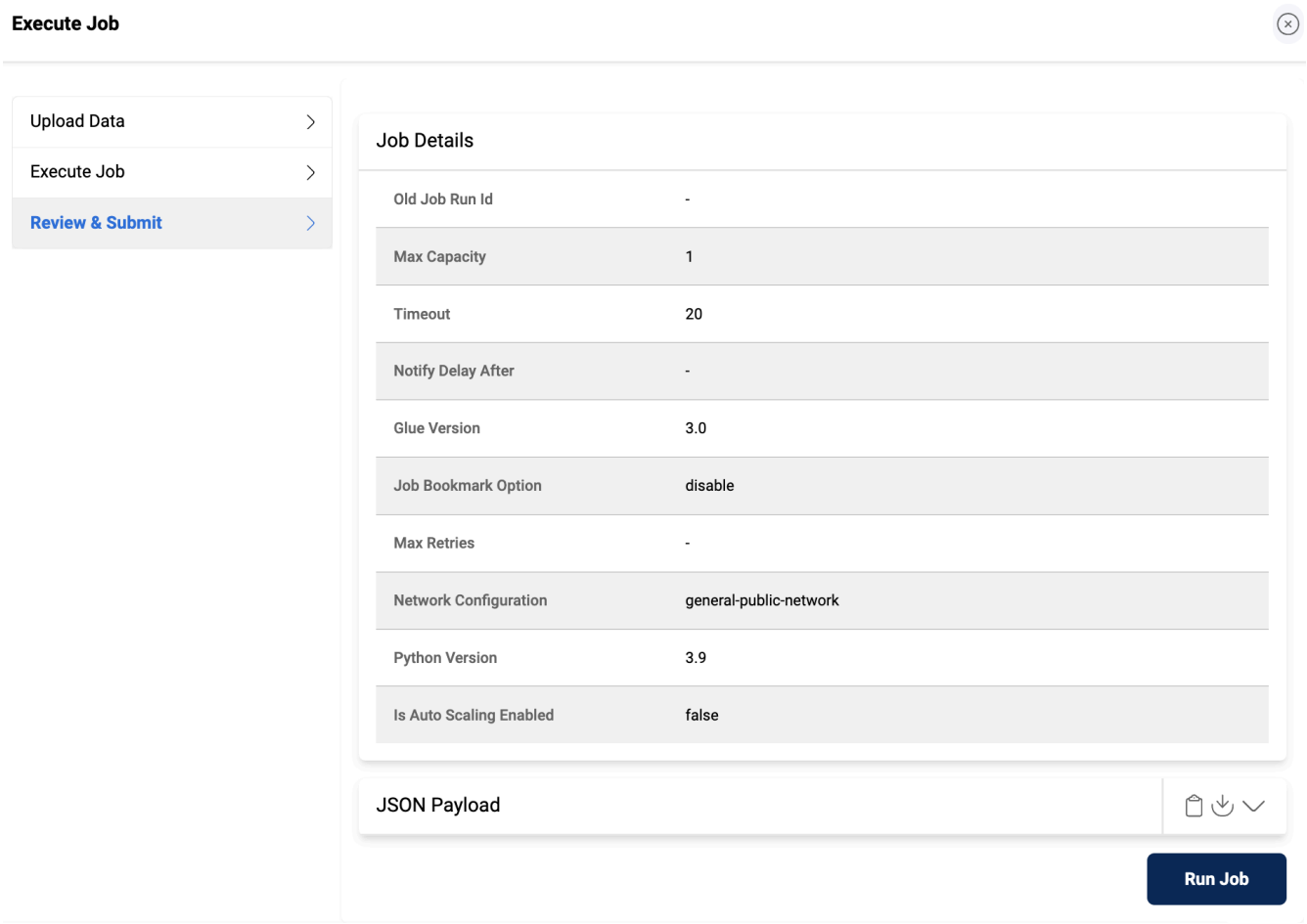


11. Let's run the ETL. Click on Run button on the right corner of ETL page for your selected Job.

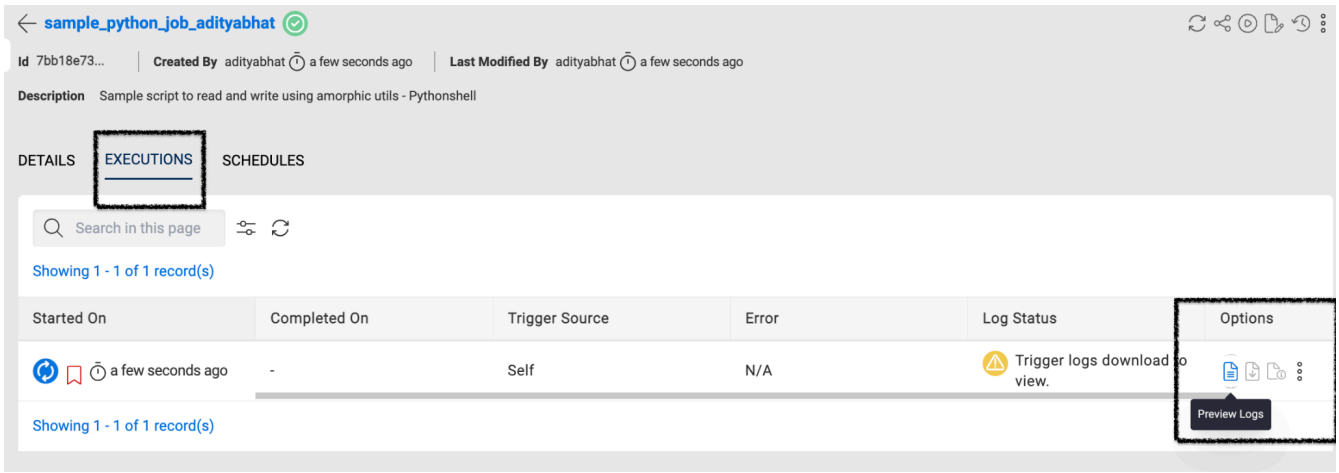


You will be taken to the **Execute Job** page. Click on Continue and then Click Run Job.

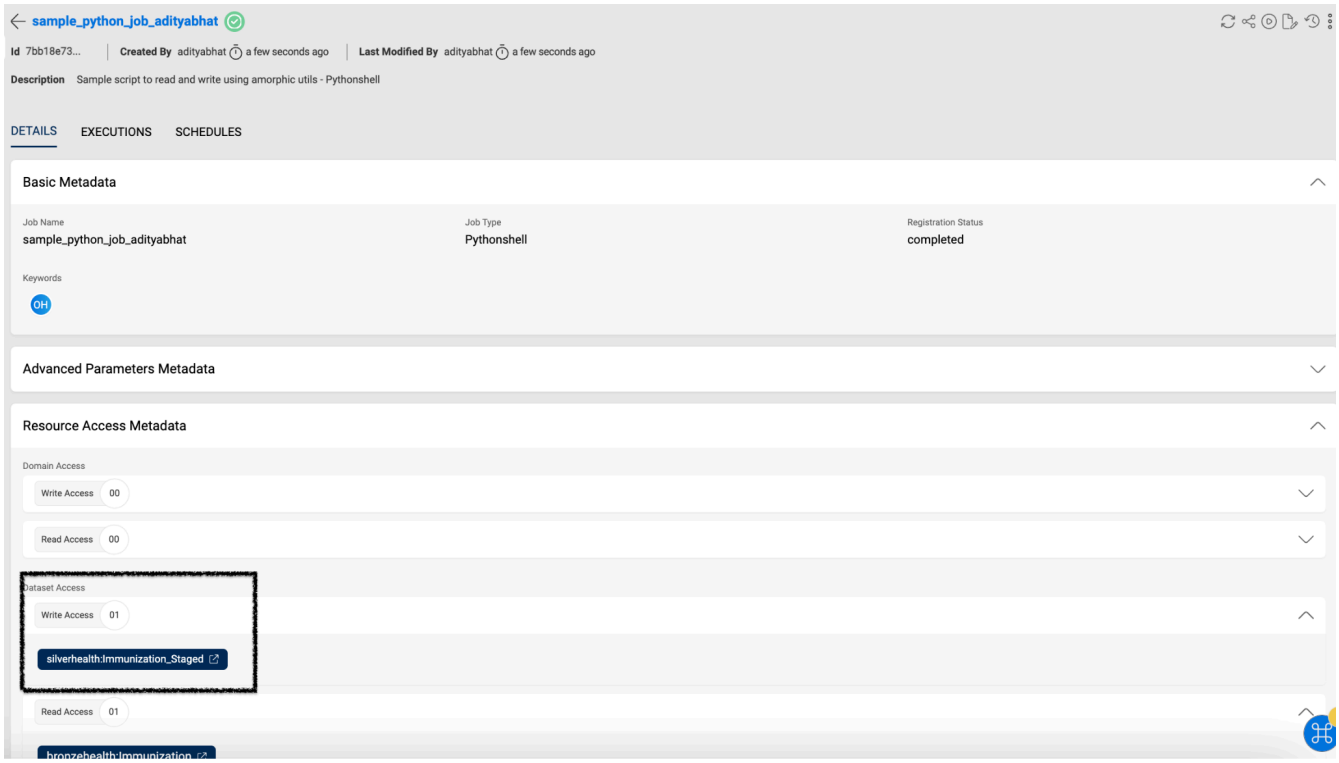




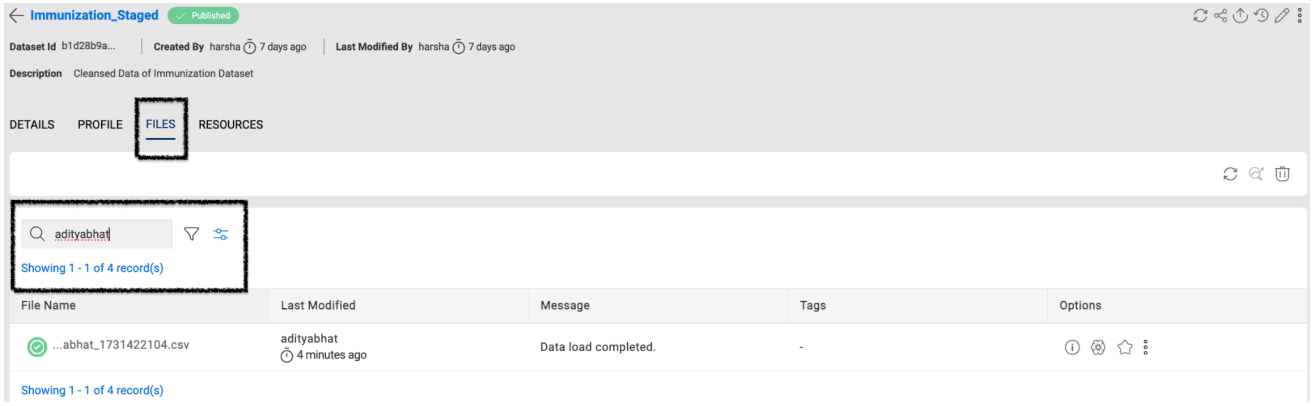
12. You can track your execution and preview logs.



13. Once the Job is successfully complete you can navigate to the Dataset you are writing to from the Details page by expanding Resource Access Metadata and clicking on the Dataset which will open in a new tab.



14. Now navigate to the Files tab of the Dataset and search for your login user id to see if the file was successfully loaded.



Congrats! You were successfully able Read and Write Data using ETL Job.
Repeat the same steps as above to create and execute PySpark Job by using ***sample_pyspark_job*** as a reference.

LAB Session 4:

Workflows:

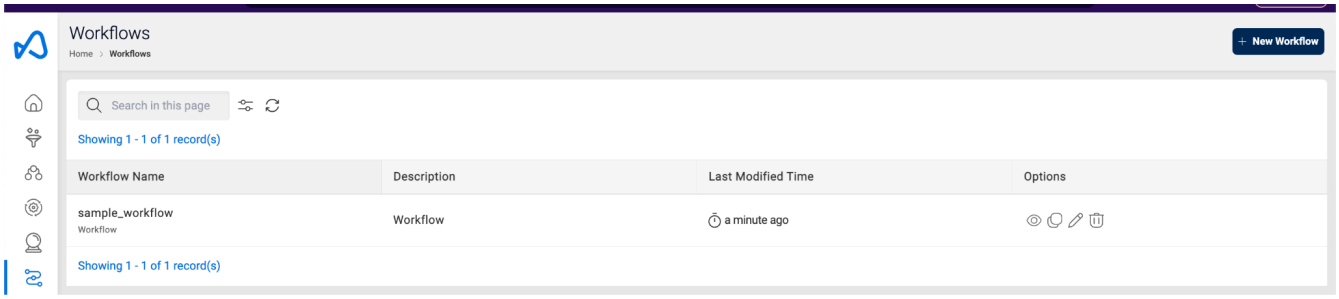
Lets create a workflow by cloning the existing one.

Purpose:

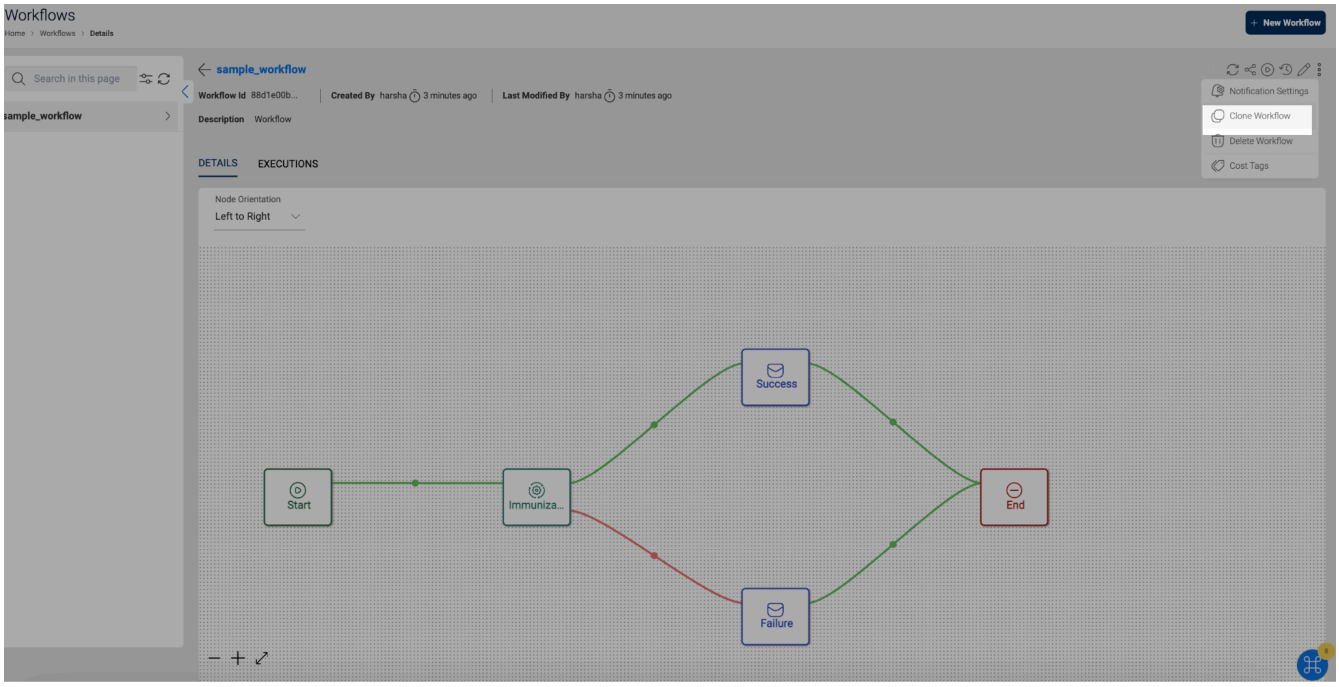
Setup a workflow to trigger an ETL, configure success and failure nodes to receive email communication

Procedure:

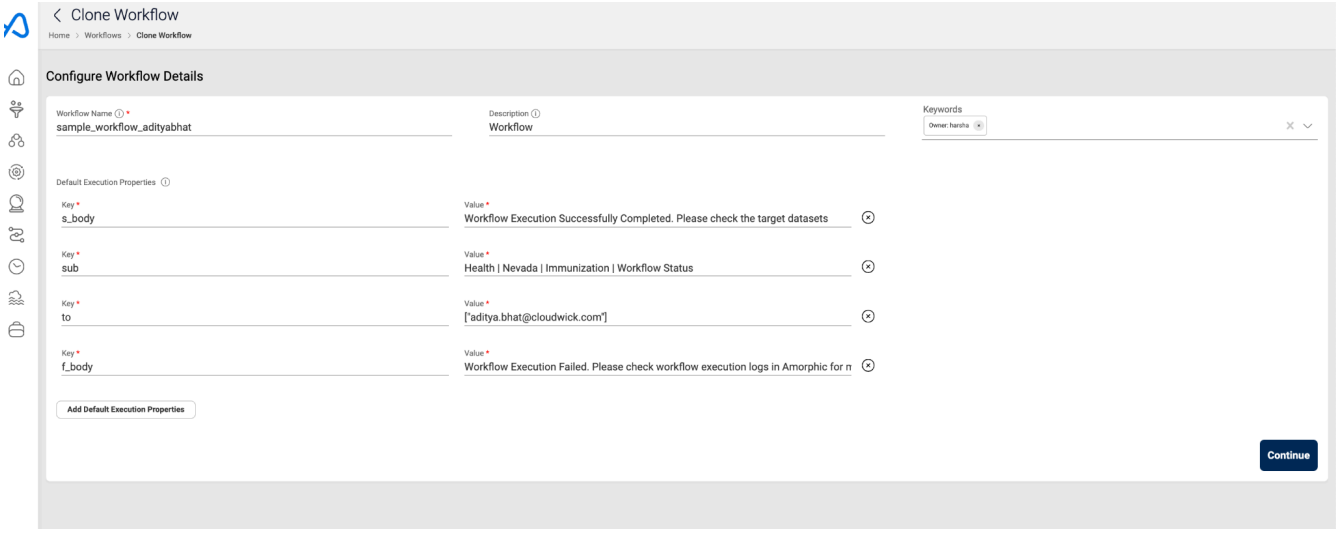
1. Navigate to Workflows from the side pane. You should already see one workflow that has been shared to you.



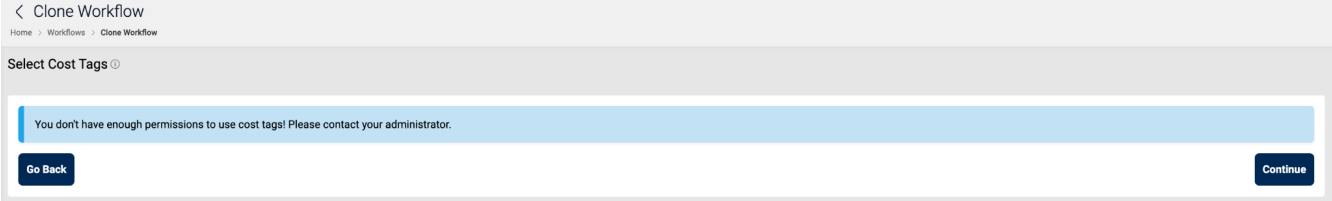
2. Click on it to see the details and click on three dots in the right corner to see the clone option and hit clone.



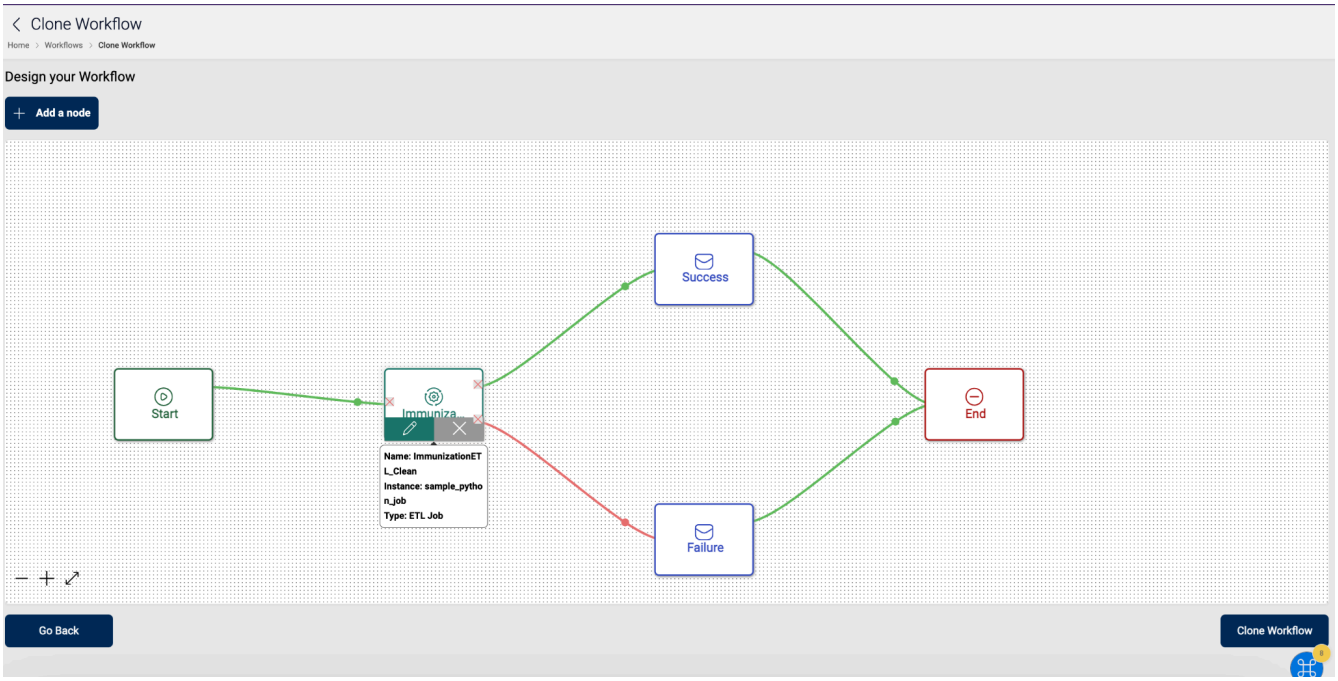
3. Rename the workflow as *sample_workflow_<yourname>* Ex: *sample_workflow_adityabhat*
   Under Default Execution Properties you will see a few key value pairs, these can be used in your workflow nodes.
   In our workflow we will be using it in a Email Node
   Update the ***'to'*** key with the list of **values** of the email address of people who you want to send email to. For this demo use your own mail_id that you used to register on the platform.
   Click on **Next** to continue.



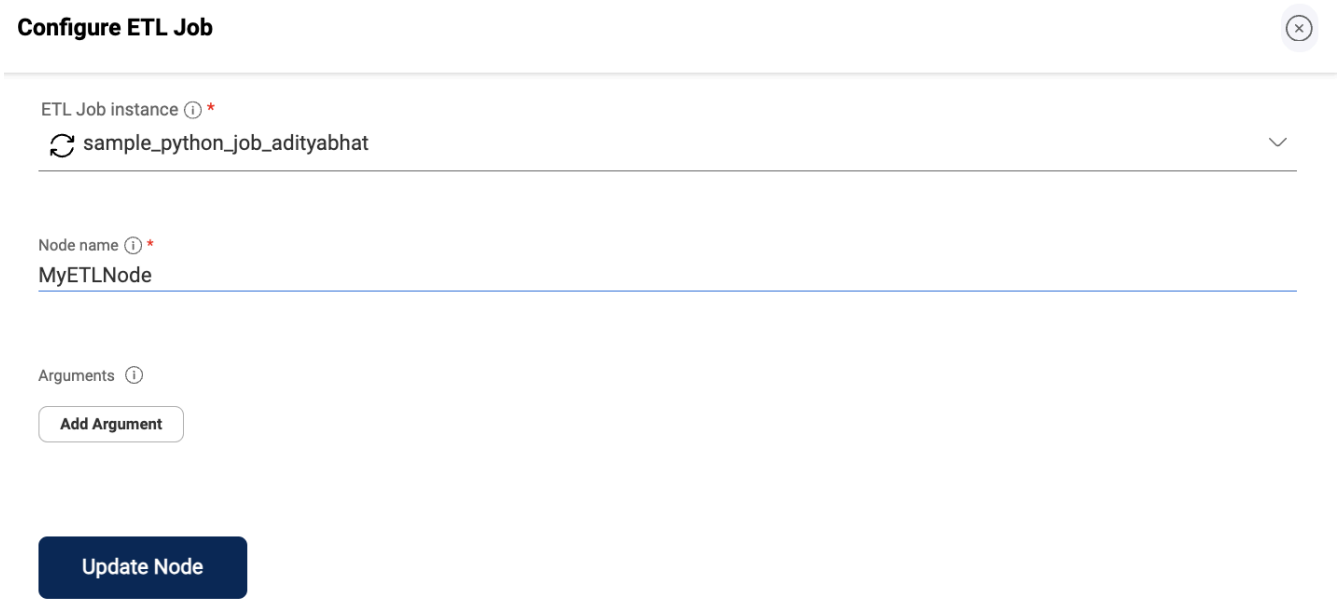
4. Skip Cost Tags and Continue.



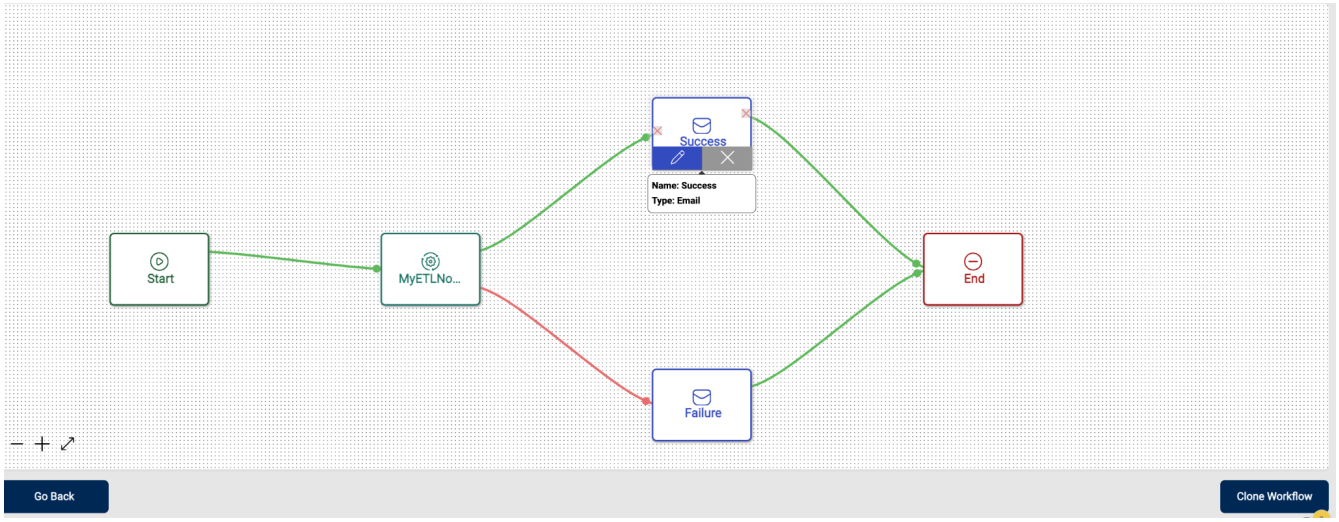
5. You can now design your workflow. Hover on any node that you want to edit and click on the pencil icon to edit/view the details of the node.



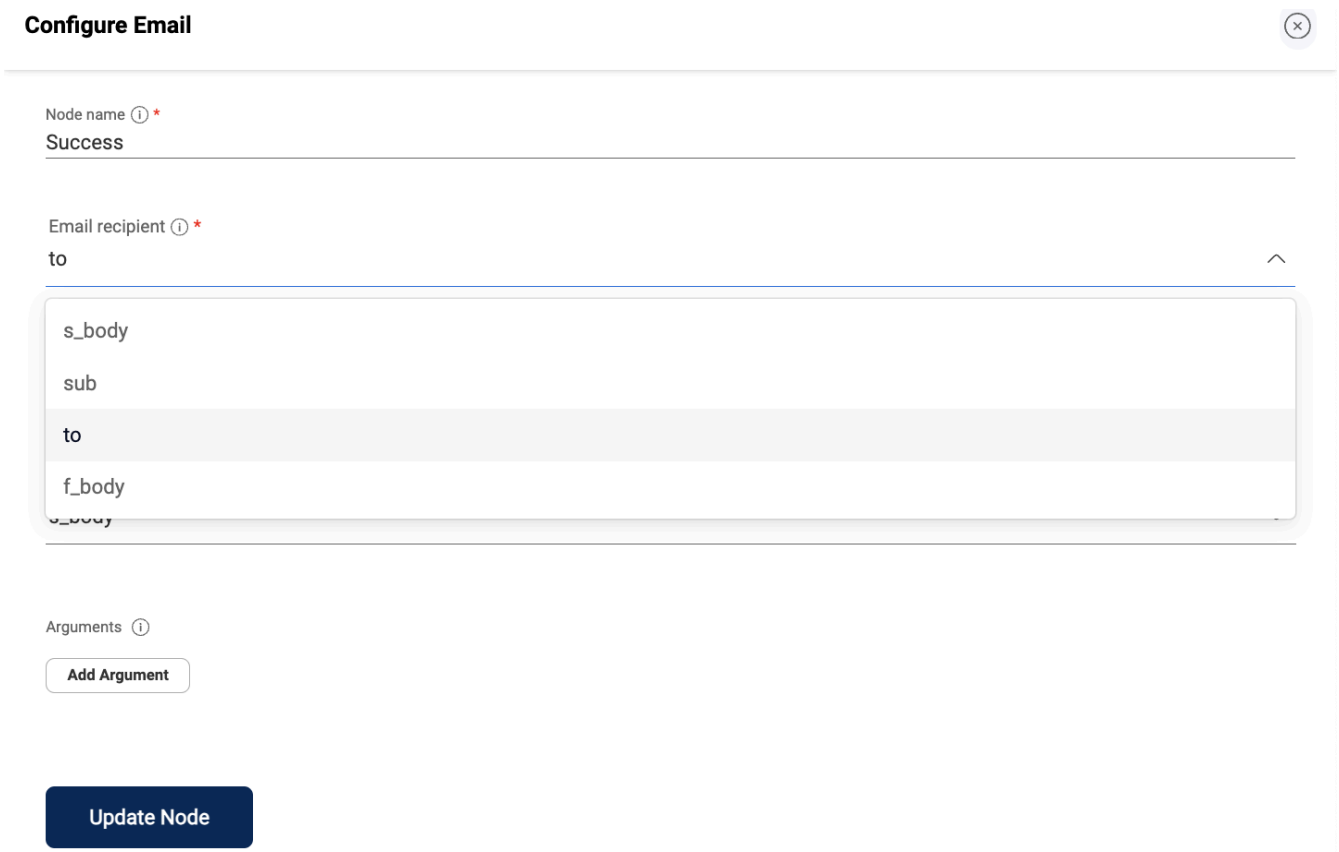
6. Let's first edit the ETL node. Change the ETL Job to the one you created in session 3. Update the node name if required and click on **Update Node**



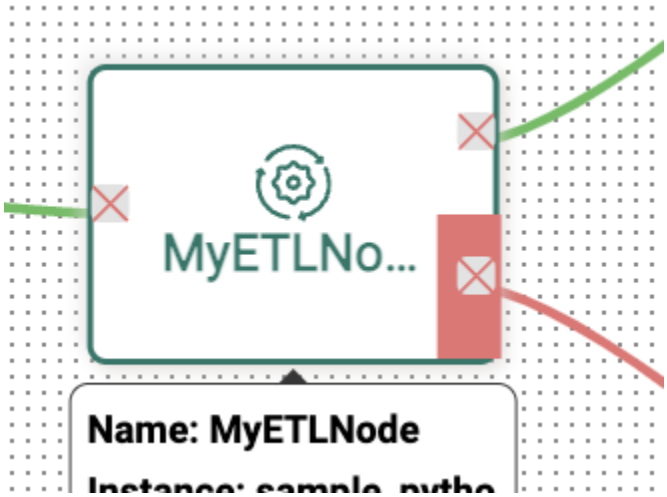
7. Lets inspect what's inside an Email Node. Click on Edit on any one of the email node (Success/Failure)



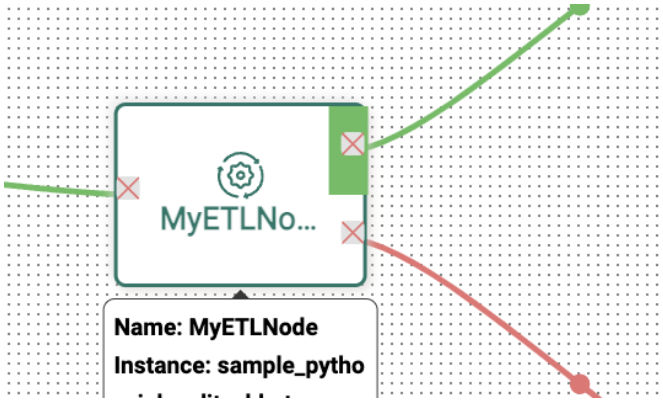
Notice that the arguments you passed earlier are now visible in the drop down. Leave the Configure Email page by clicking anywhere outside.



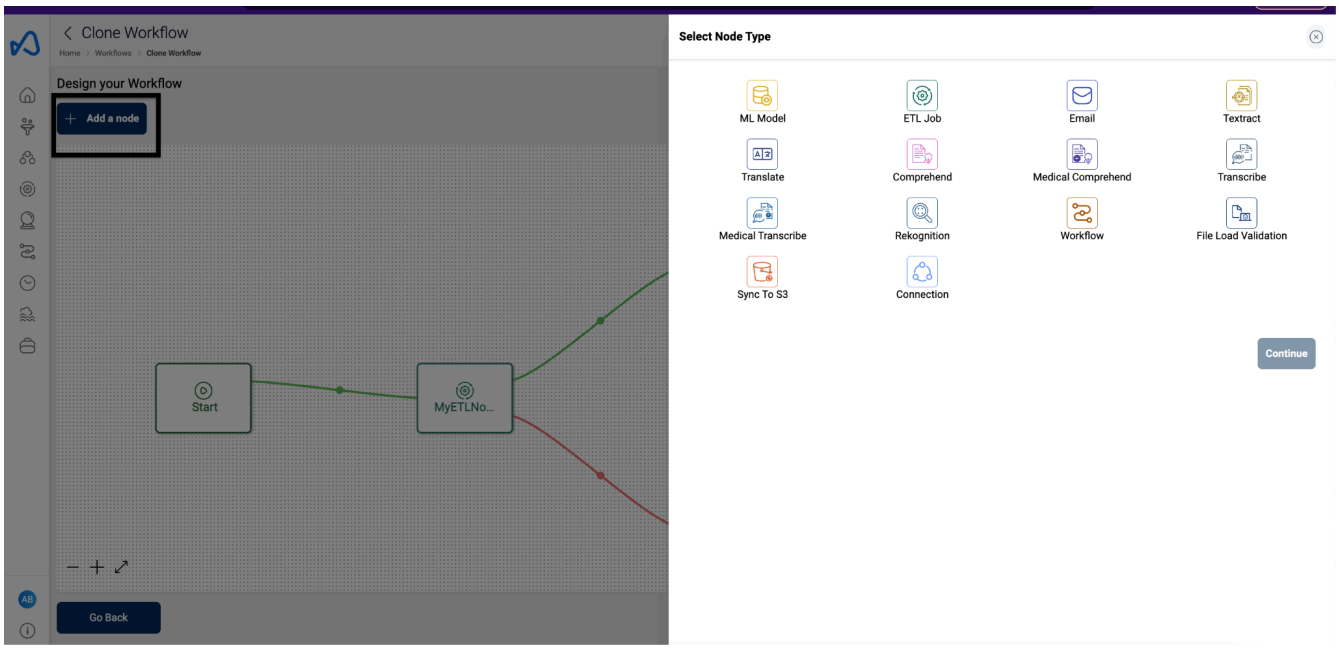
8. Notice the red bar in the node, you can connect it to any other node which would then execute on failure of the previous node. (which in this case is a failure node)



9. Notice the green bar on the node, you can connect it to any other node which would then execute on the success of the previous node. (which in this case is a success node)



10. You can also add any of the available nodes by clicking on Add Node. You can get to know more about each node in the documentation

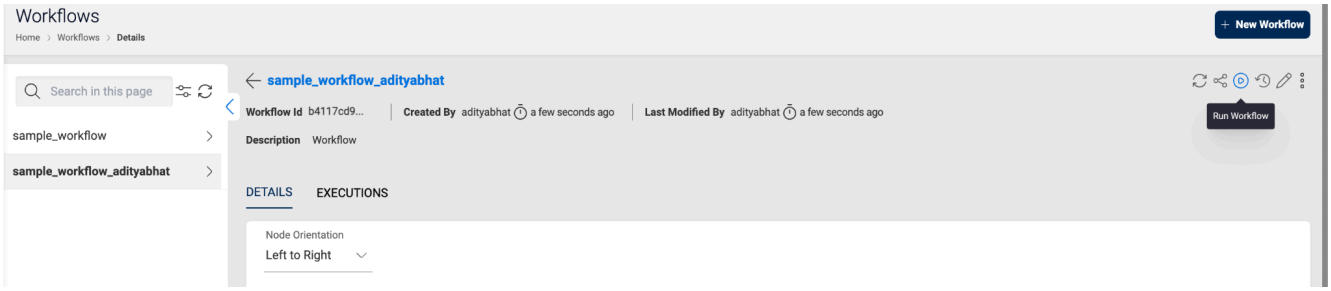


11. Let's now simply clone the workflow.

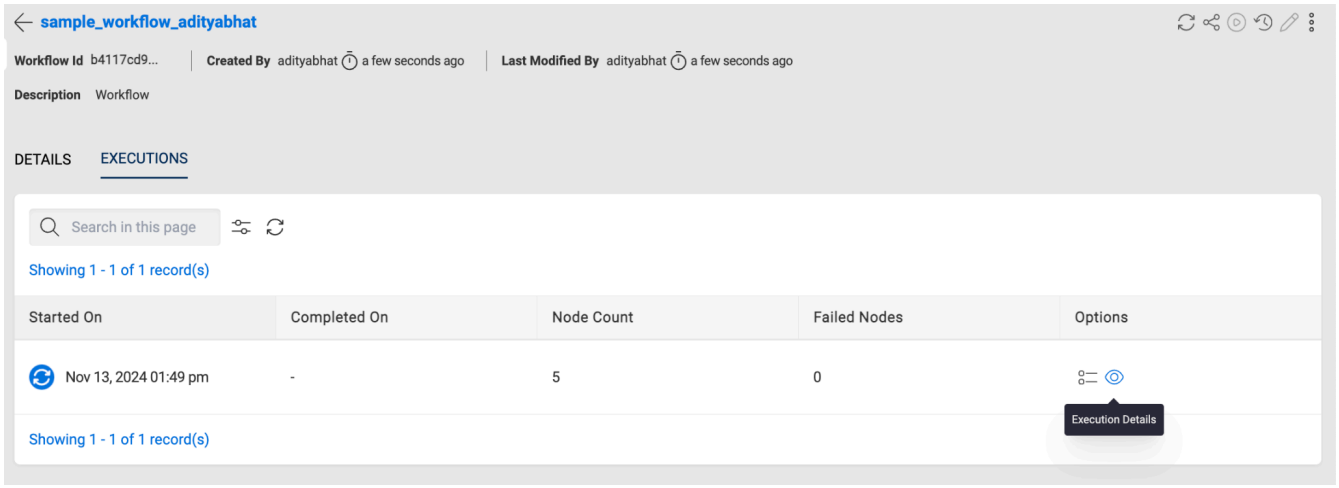
    You have successfully created a workflow.

Executing a workflow:

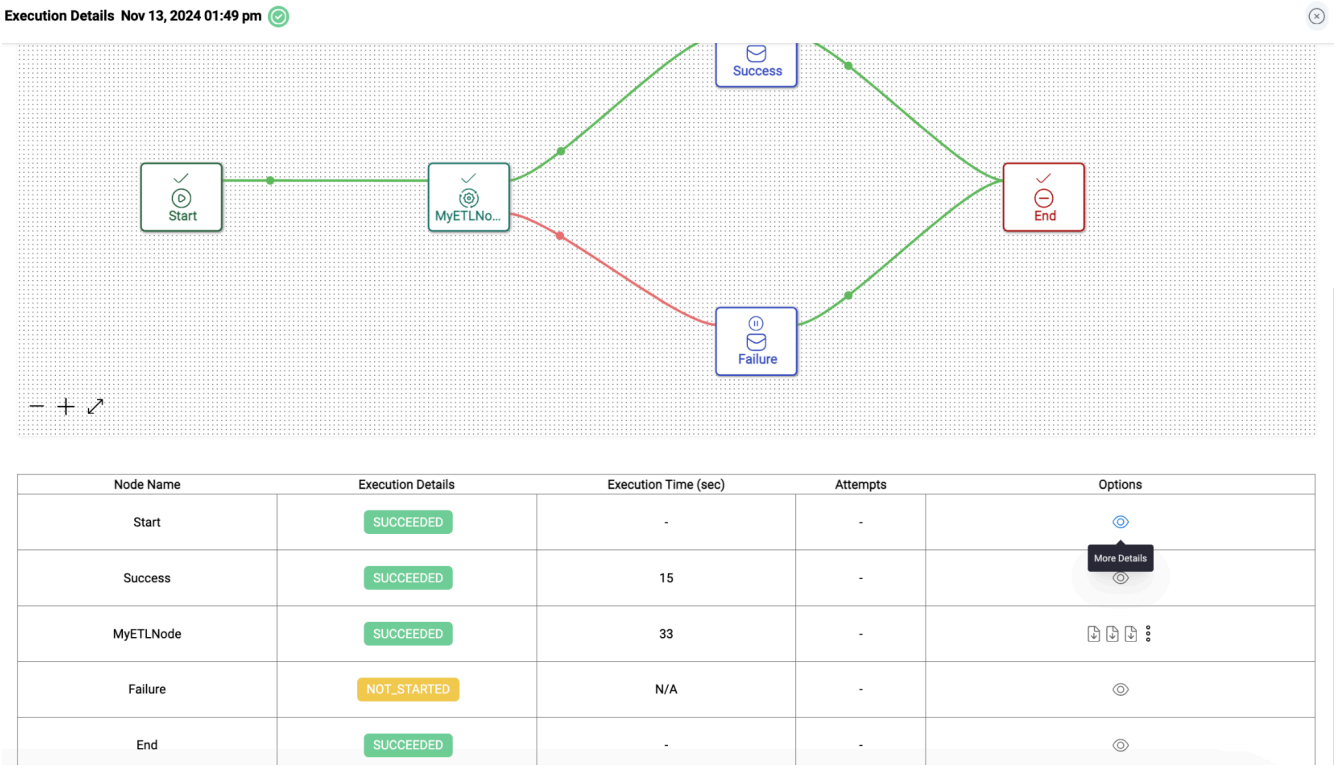
1. Select the workflow you want to run and click on the run workflow button on the left. Confirm if you see a popup.



2. Go to the Execution tab to see the progress.



3. You can click on the Options to see more info of progress of each node.



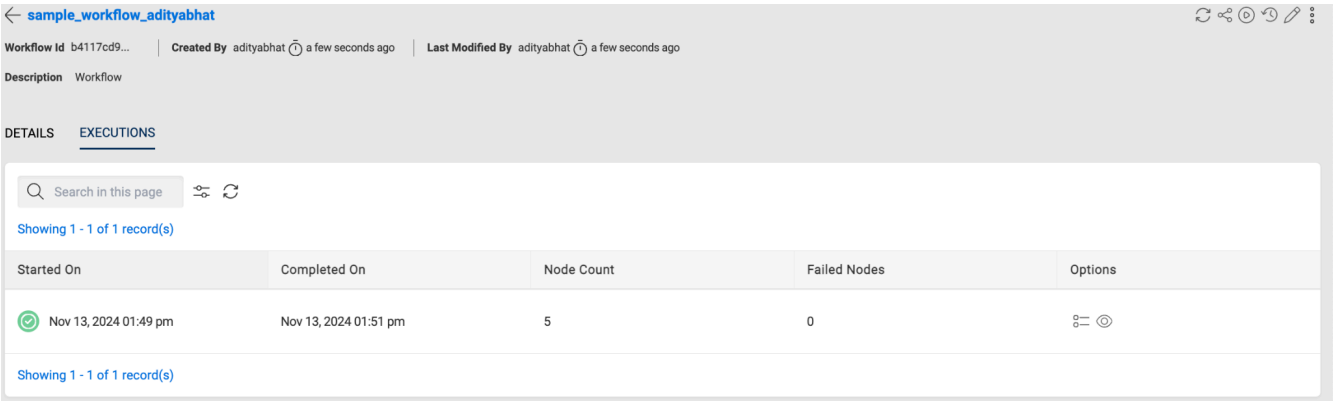
Node Name	Execution Details	Execution Time (sec)	Attempts	Options
Start	SUCCEEDED	-	-	
Success	SUCCEEDED	15	-	
MyETLNode	SUCCEEDED	33	-	
Failure	NOT_STARTED	N/A	-	
End	SUCCEEDED	-	-	

4. You can also see the logs of your ETL run straight from this Execution Details page.

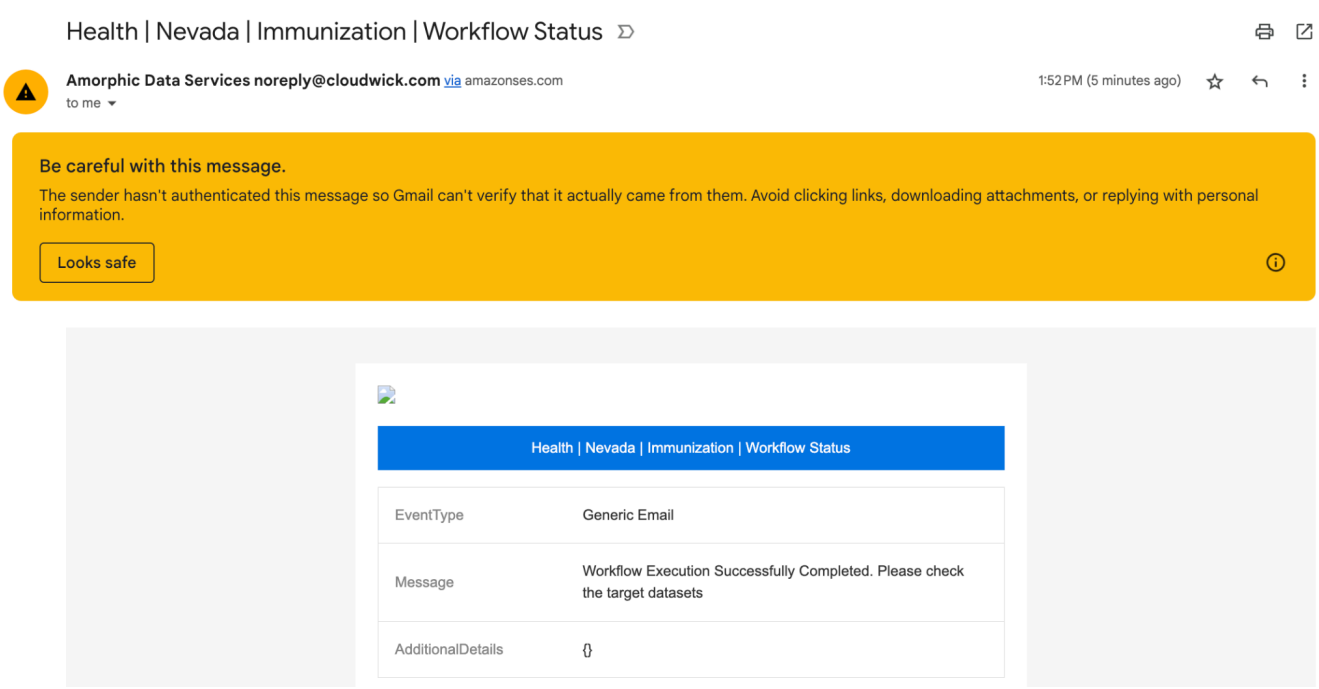
Node Name	Execution Details	Execution Time (sec)	Attempts	Options
Start	SUCCEEDED	-	-	
Success	SUCCEEDED	15	-	
MyETLNode	SUCCEEDED	33	-	
Failure	NOT_STARTED	N/A	-	
End	SUCCEEDED	-	-	

Error Logs

5. Back at the Execution tab, you can see the workflow has now completed successfully. The summary of success and failed nodes is also displayed. In this case all the five are completed with success status.



6. Let's check if we received mail on the success of this workflow.



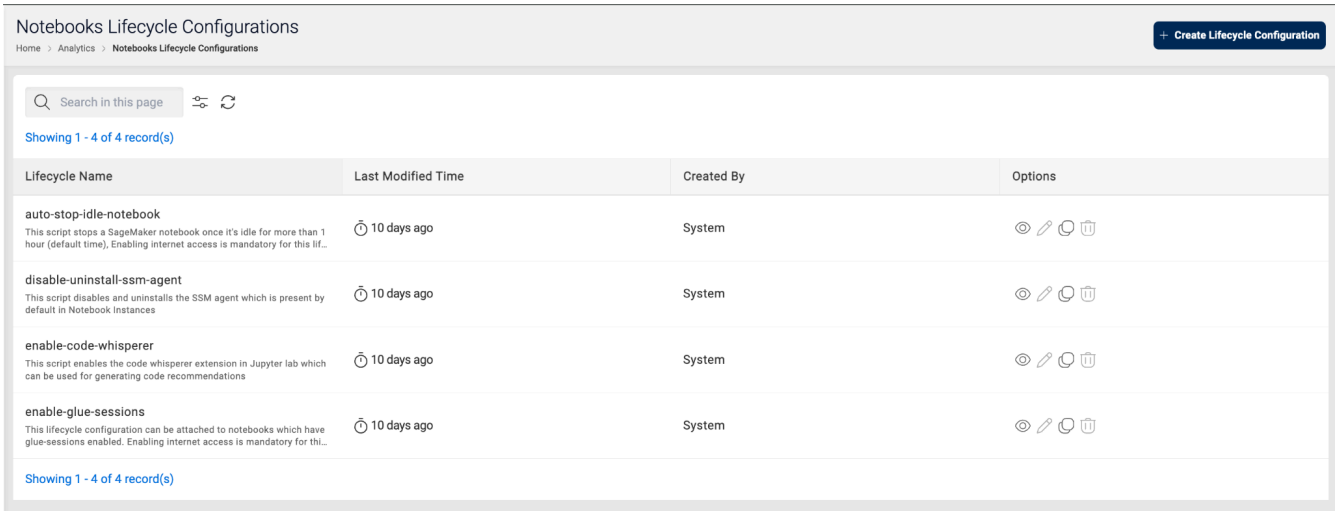
Congrats you have now successfully created and executed a workflow.

Notebook Lifecycle Configurations:

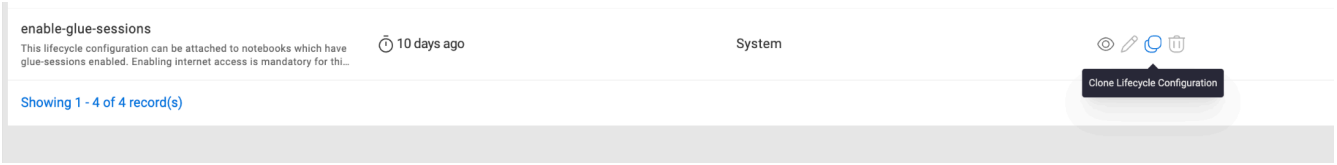
A lifecycle configuration provides shell scripts that run only when you create the notebook instance or whenever you start one

Procedure:

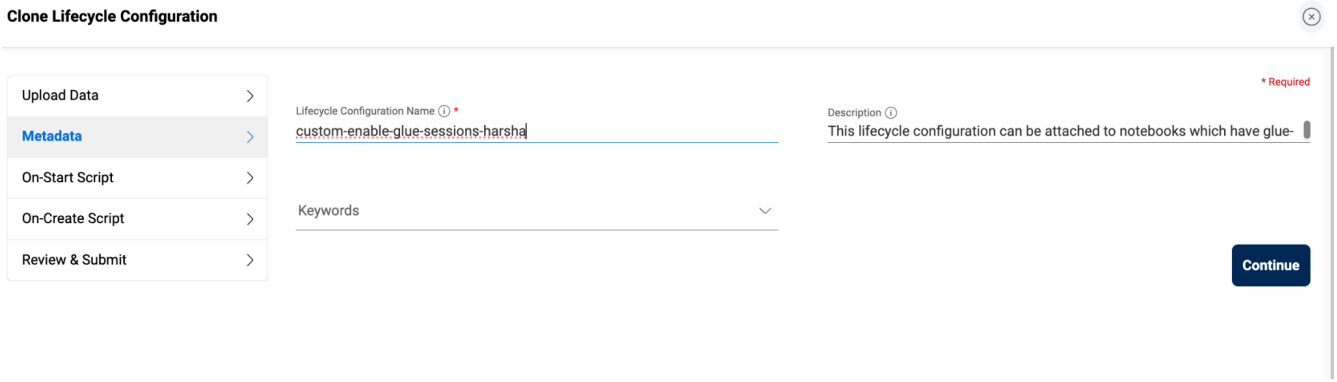
1. Navigate to Analytics > Lifecycle Configurations.
2. You will already see list of system created Lifecyle Configs.



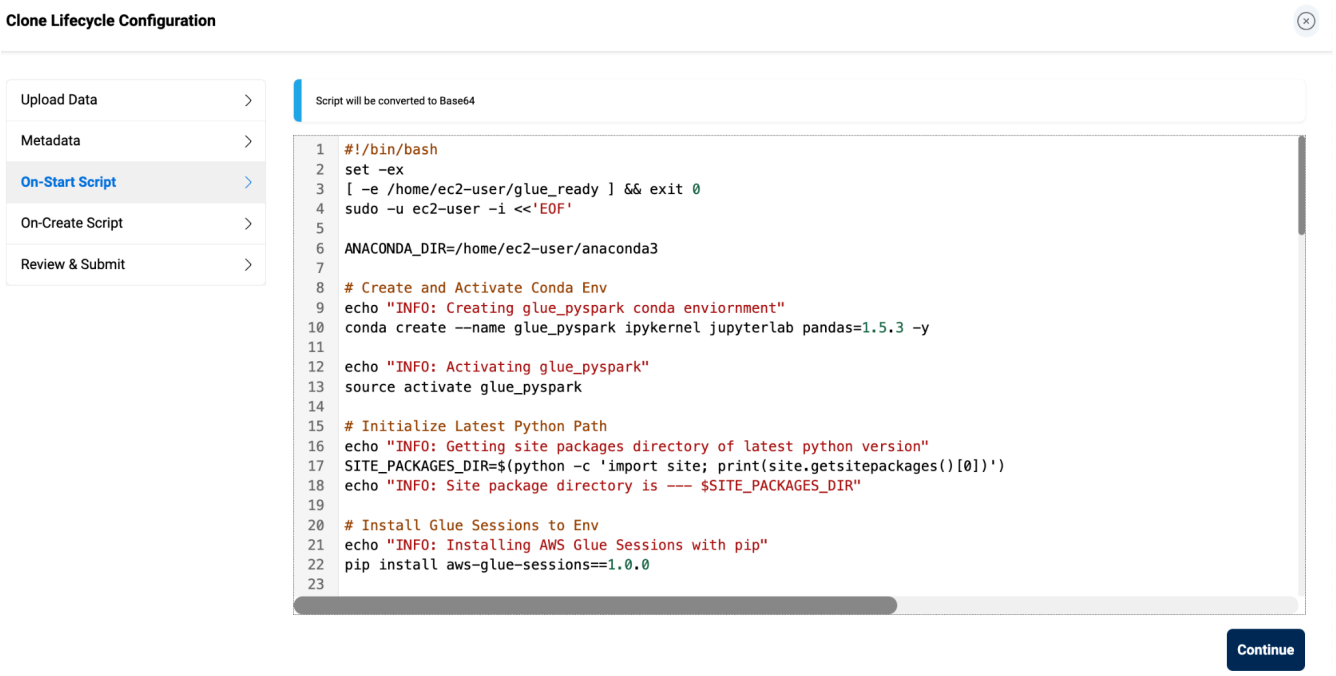
3. Lets create a create custom LC that combines enabling glue session and auto stopping the idle instance. Clone the [enable-glue-sessions]



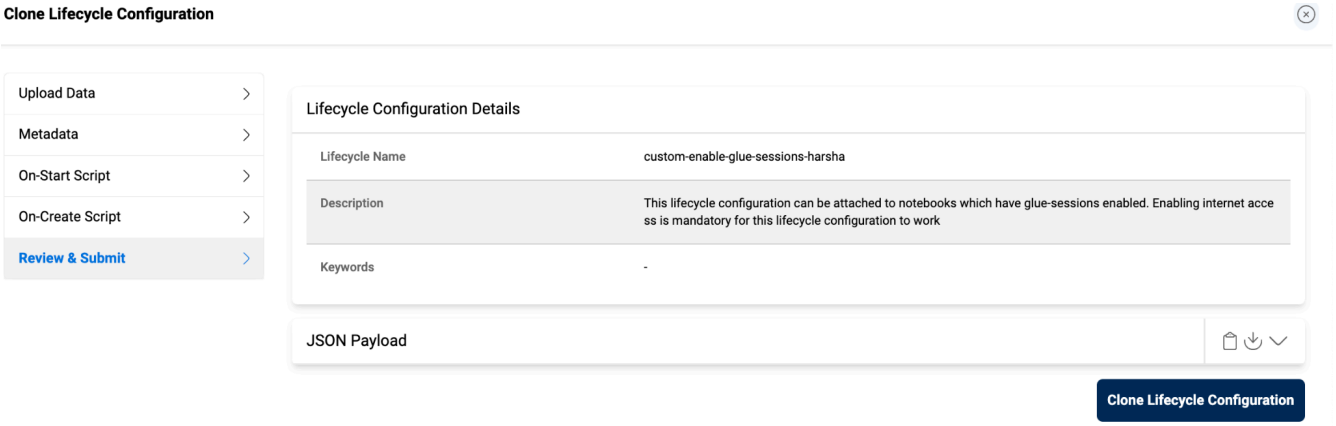
4. Rename it to custom-enable-glue-sessions-<yourusername> Ex: custom-enable-glue-sessions-harsha and click continue.



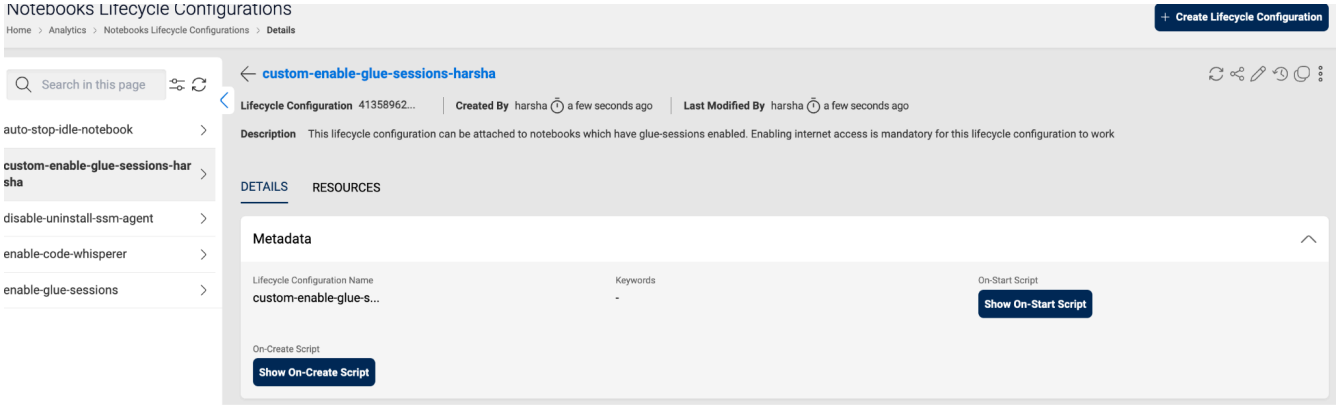
5. Replace the On-Start Script from the script given below. https://github.com/cwkadityabhat/nevada-public-datalake/blob/main/Lab4/custom-enable-glue-sessions-lc.sh



6. Leave the On-Create Script unchanged and Proceed to Review. Clone the LifeCycle Configuration



7. You will be redirected to the main page upon successful creation.



You have successfully created a custom life cycle configuration. We will use this to attach it to our Notebook in next part.

Code Repsitory:

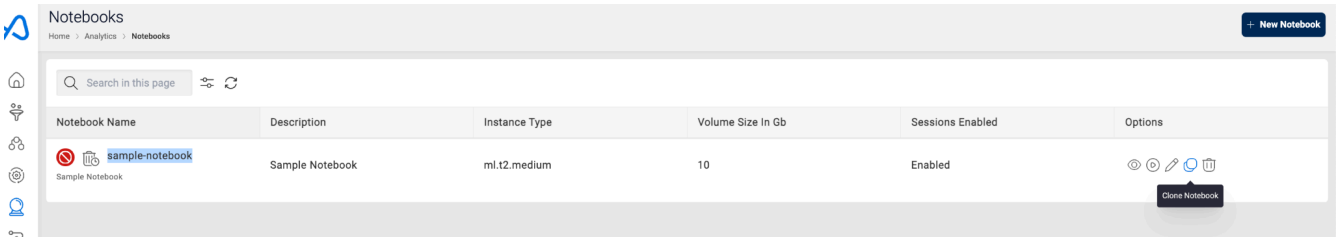
The code repository has already been created for you **sample-public-repo**. You will be using this to attach it to a notebook.
Only a unique code repository can exist in amorphic with a particular repository url. Hence you do not need to create any repo for this demo

Notebooks:

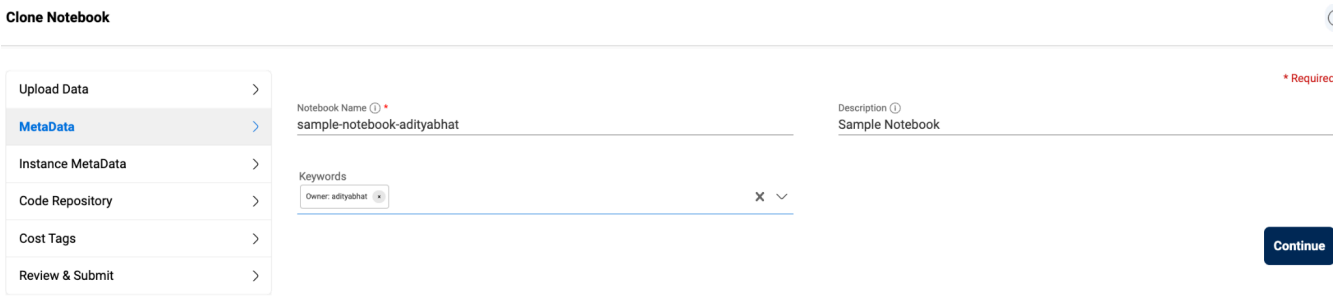
Procedure:

1. Navigate to Analytics > Notebooks from the left pane. A sample notebook is shared with you [sample-notebook]

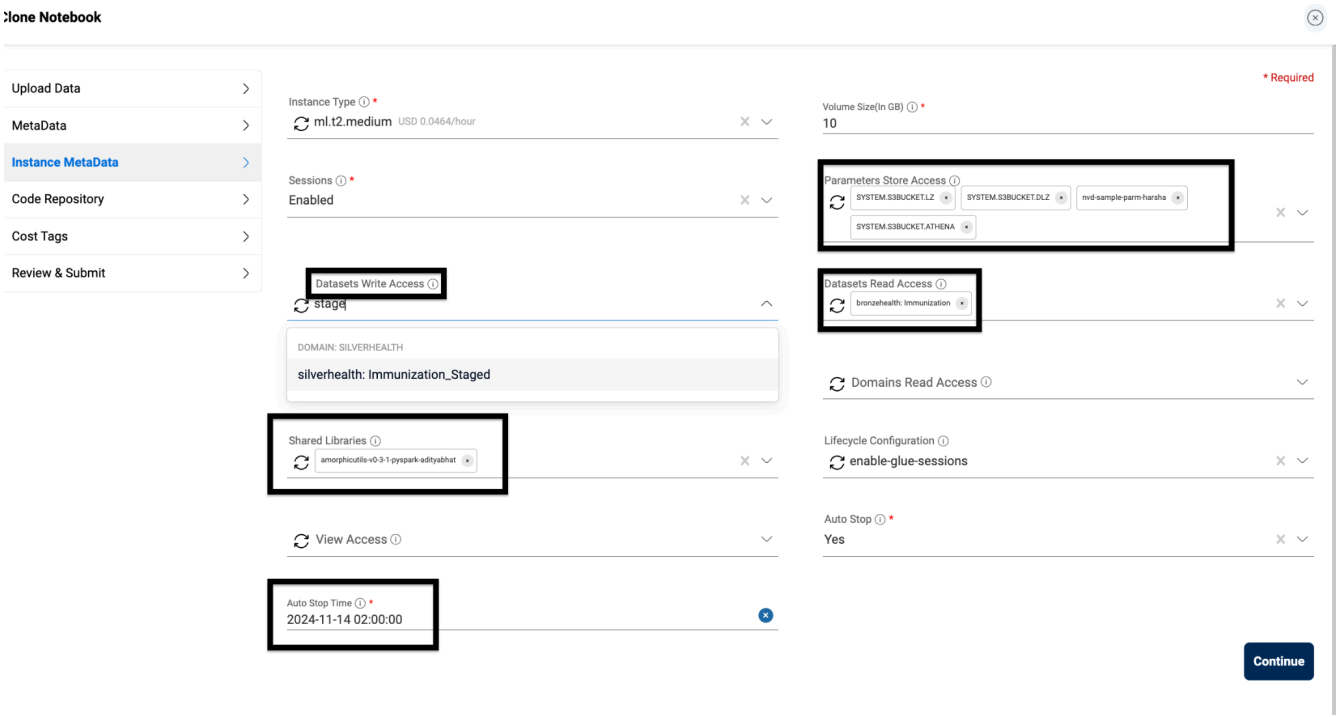
2. Clone the notebook from notebooks page under options of the notebook that you want to clone.



3. Rename the notebook from sample-notebook to sample-notebook-<username>. Ex: sample-notebook-adityabhat. Optionally you can update the keywords. Click on **continue**.



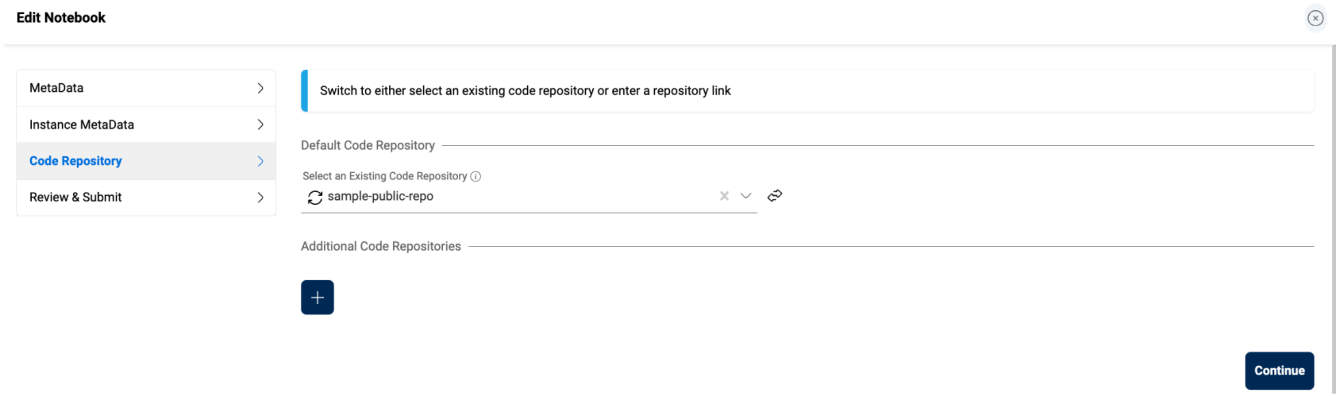
4. You might have to update below fields.
   Dataset Read Access: ***Immunization*** (Domain: bronzehealth)
   Dataset Write Access: ***Immunization_Staged*** (Domain: silverhealth)
   Parameter Store: The paramstore value that you created in session 3. (*nvd-sample-parm-adityabhat*)
   Shared Libraries: The pyspark library that you created in session 3. Use amorphicutils-v0-3-1-pyspark if you havent.
   Lifecycle Configuration: enable-glue-sessions
   Auto Stop Time: Select this time that you want instance to be stopped to stop incurring additional cost.



Click on continue after updating all of the above mentioned fields.

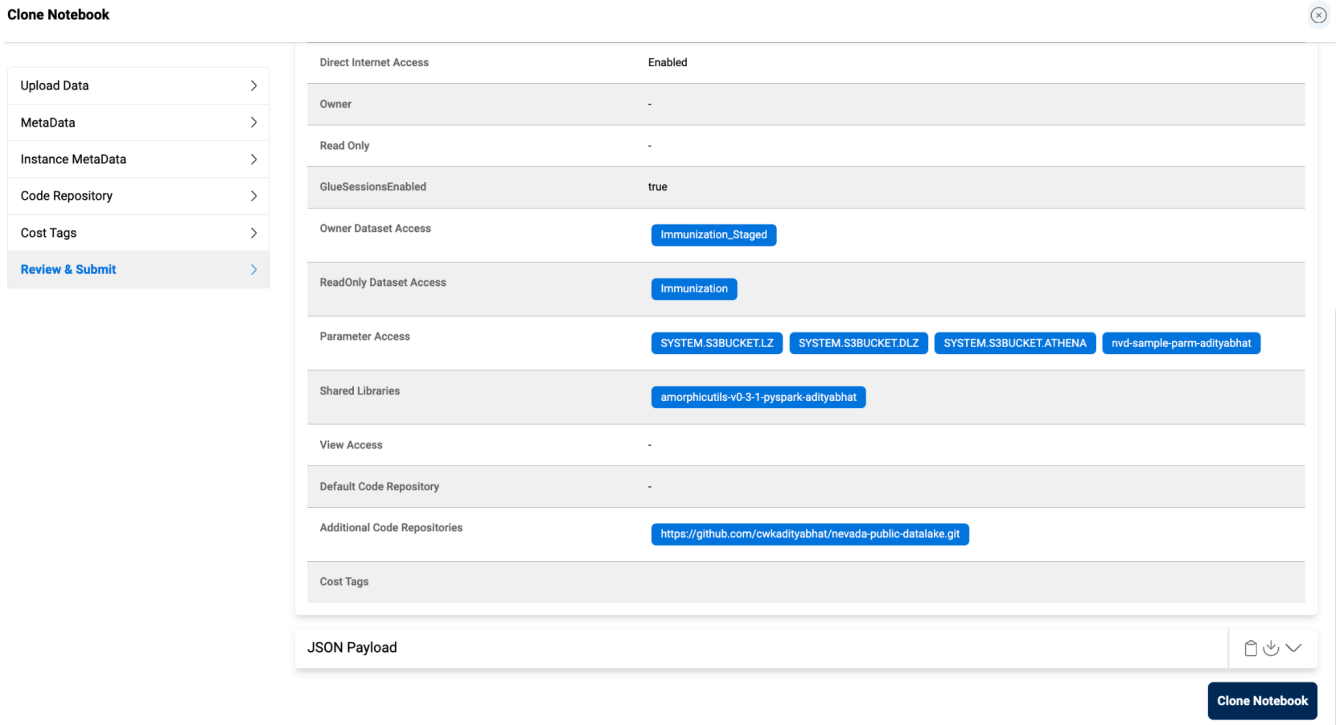
5. You can setup code-repository to work/ test the code.
   Under Default Code Repository choose from drop down sample-public-repo

   Repor Used: https://github.com/cwkadityabhat/nevada-public-datalake.git
   You can click on the above link to check the public repo hosted in GitHub.

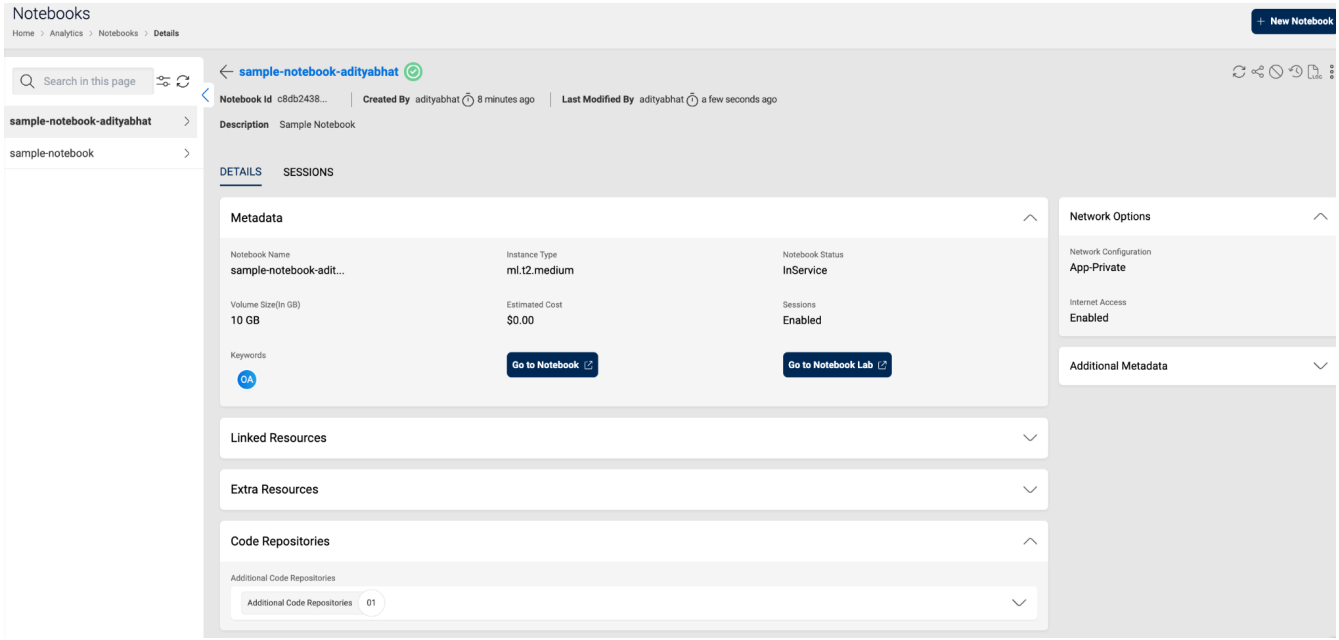


   Click on **Continue**.

6. Skip Cost tags. Review the configurations and hit on Clone Notebook at the bottom



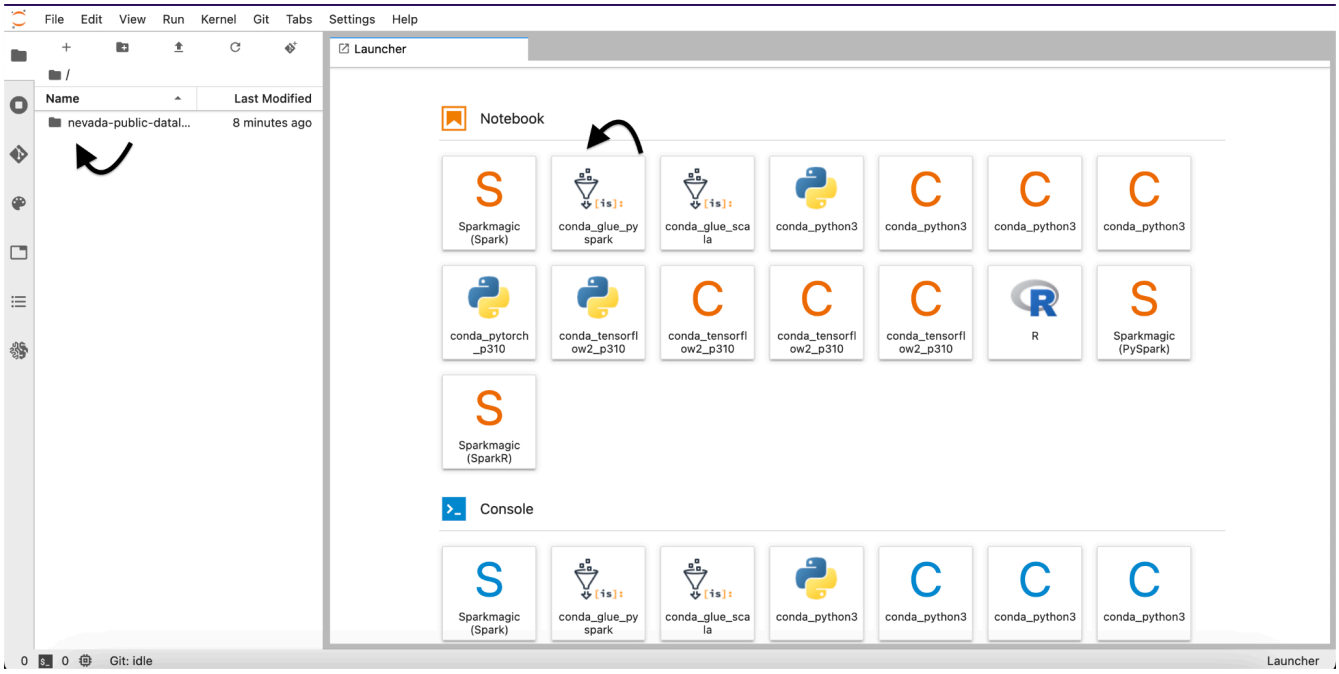
7. A notebook is created for you. The status of the notebook will be in **pending** Status for approx 10-15 mins. Once the notebook instance is up you will see the Status **In-Service**



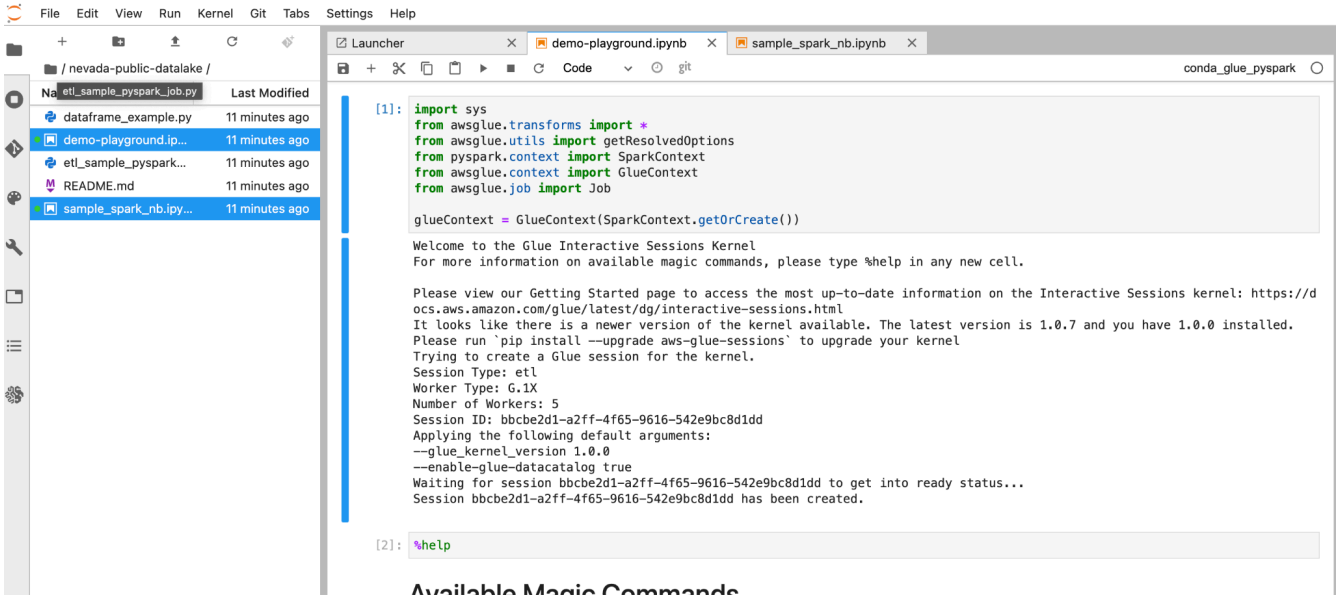
You have successfully created a Notebook.

Using a Notebook:

1. From the notebook you have created click on Go to Notebook Lab. This will take you to the lab where you can create a notebook, use a variety of consoles also notice that the git repository has been cloned for you.

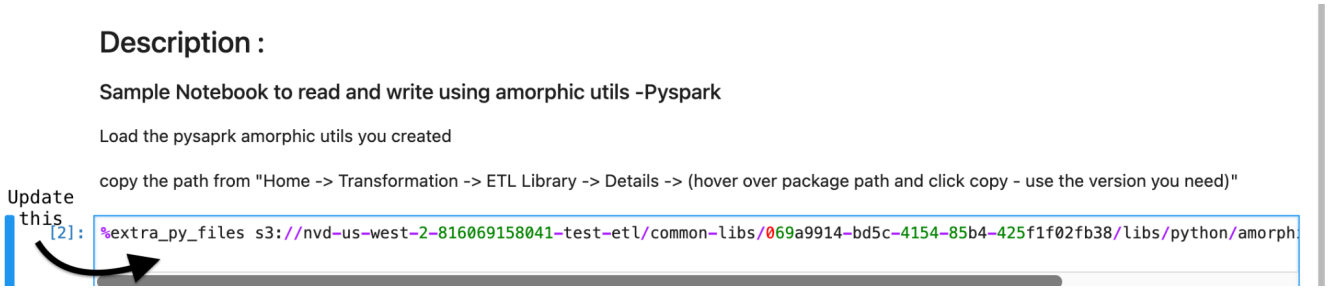


2. Navigate inside the repo and you can make use of the notebook thats already been created for you.



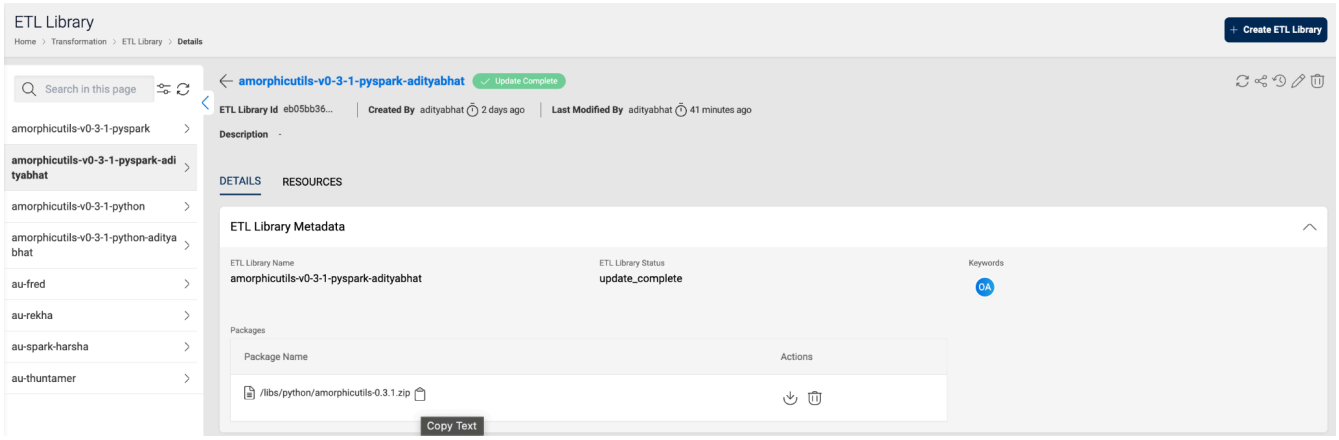
demo-playground.ipynb: Use this to play around
sample_spark_nb.ipynb: This is the same code you ran on ETL yesterday using amorphicutils.

3. Add the path for pyspark library to be used while using ***sample_spark_nb.ipynb***



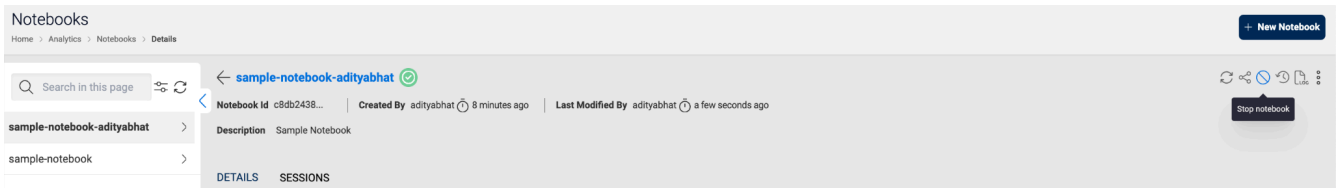
You can navigate to Home -> Transformation -> ETL Library -> Details -> (hover over package path and click copy - use the version you need)

Note: Your session will fail if you dont copy the correct path/ have missed attaching your lib to instance.



4. Use Shift + Enter to run each cell.

5. Stop the notebook after your use.



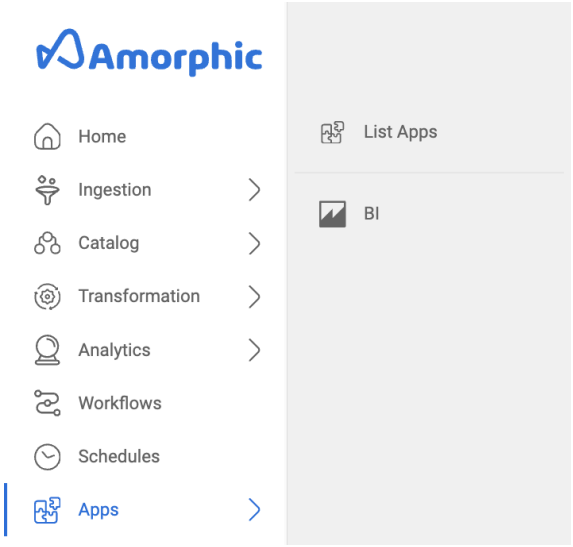
Congrats you have successfully learnt to Create, Use and Stop the Notebook.

LAB Session 5:

Amorphic BI:

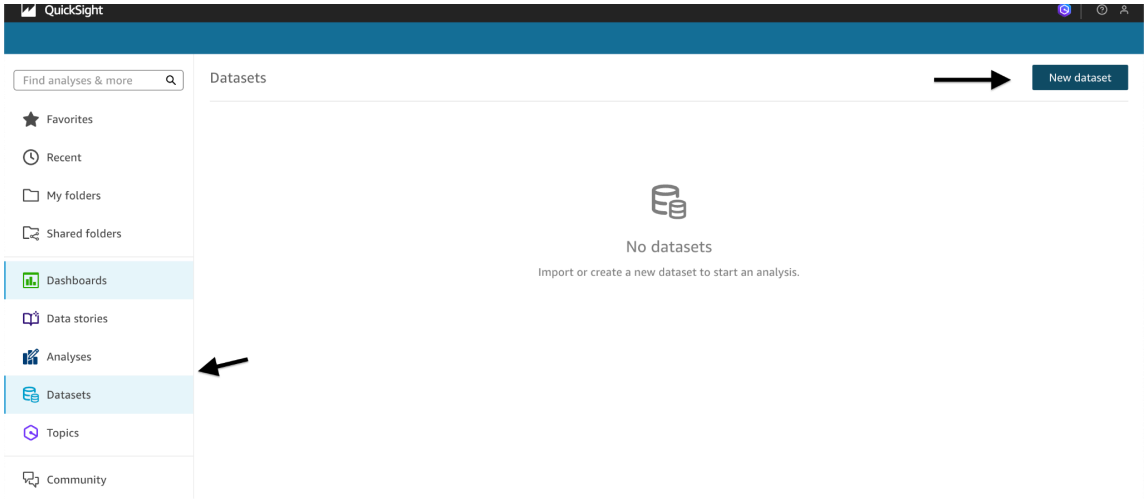
Creating a Quicksight Dashboard:

1. Navigate to Apps > Click on BI

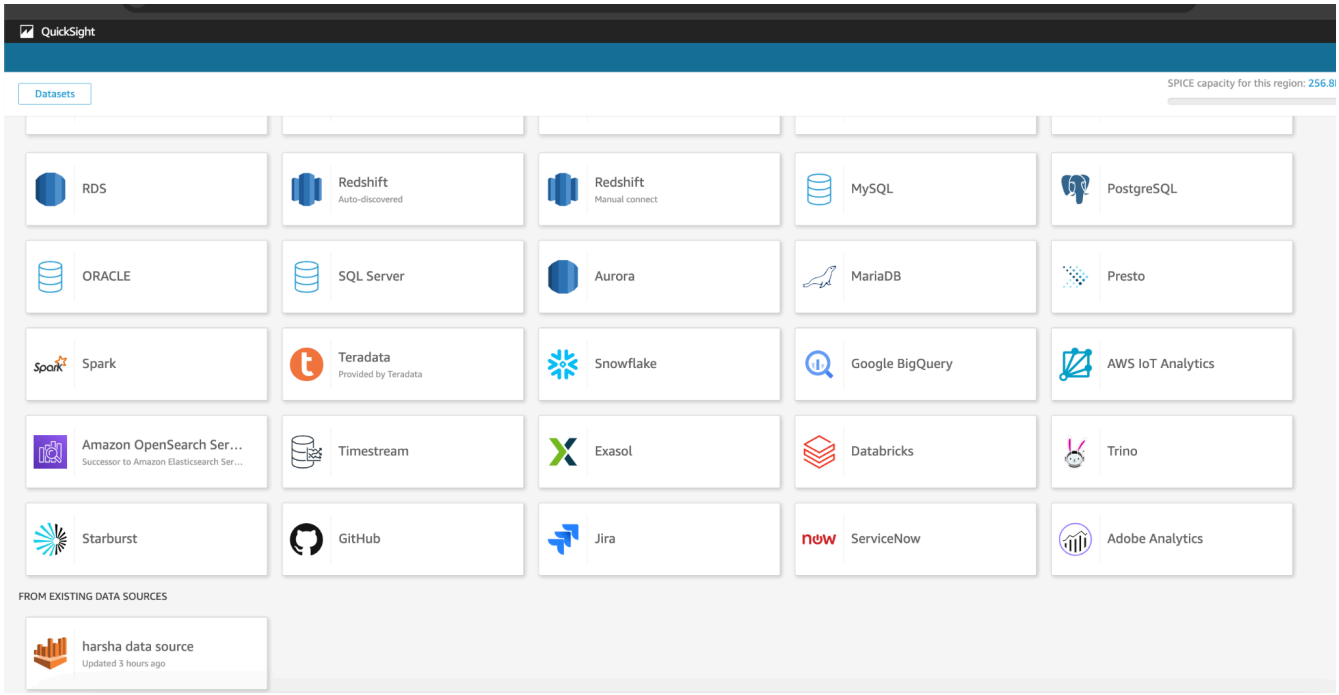


You will be redirected to Quicksight Dashboard

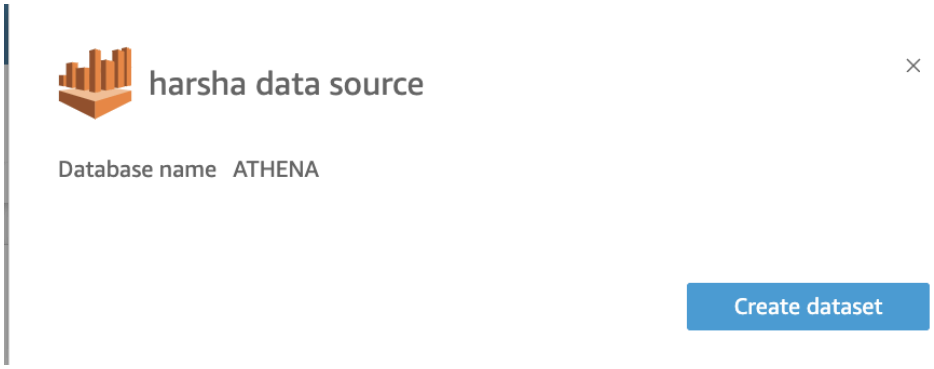
2. Click on Dataset on the left pane and then click on ‘New Dataset’ to add new Dataset from source.



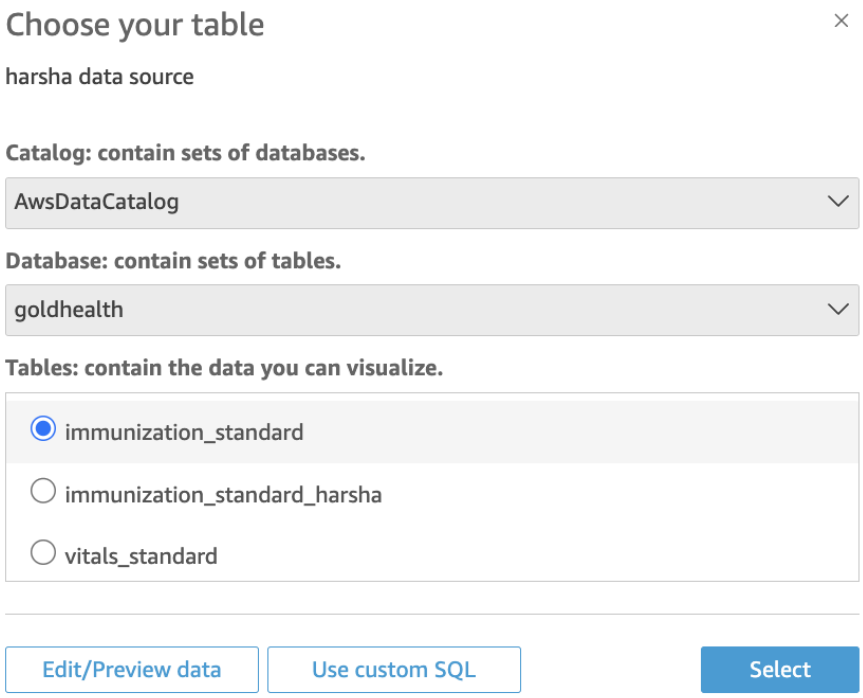
3. Scroll to the bottom and use the Existing Data Source created for your user_id



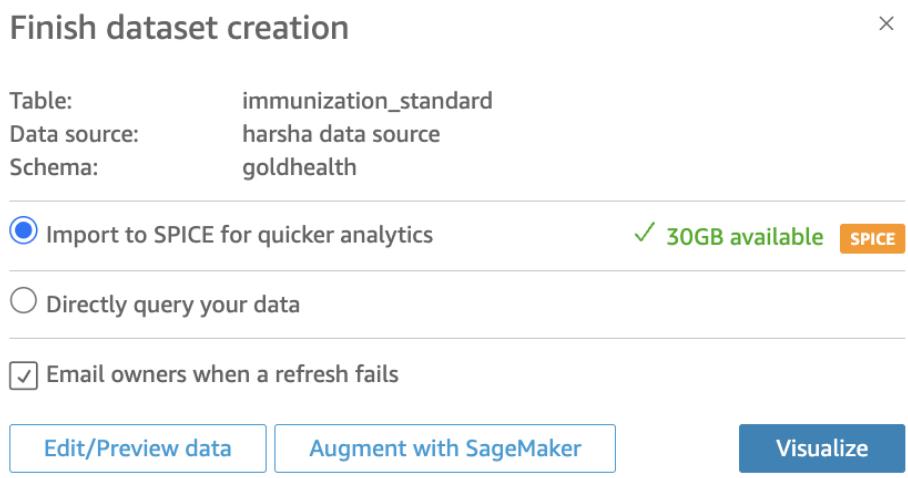
4. Click on Create Dataset



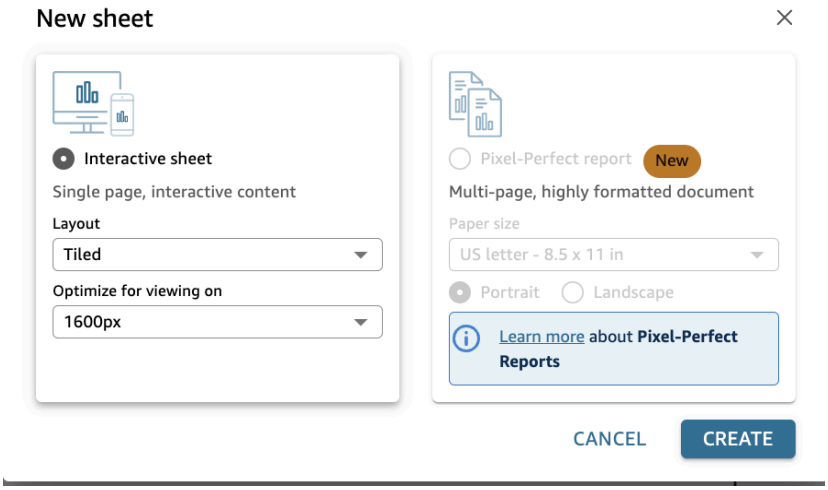
5. Choose the table to use for analysis. Lets use goldhealth.immunization_standard for this demo.



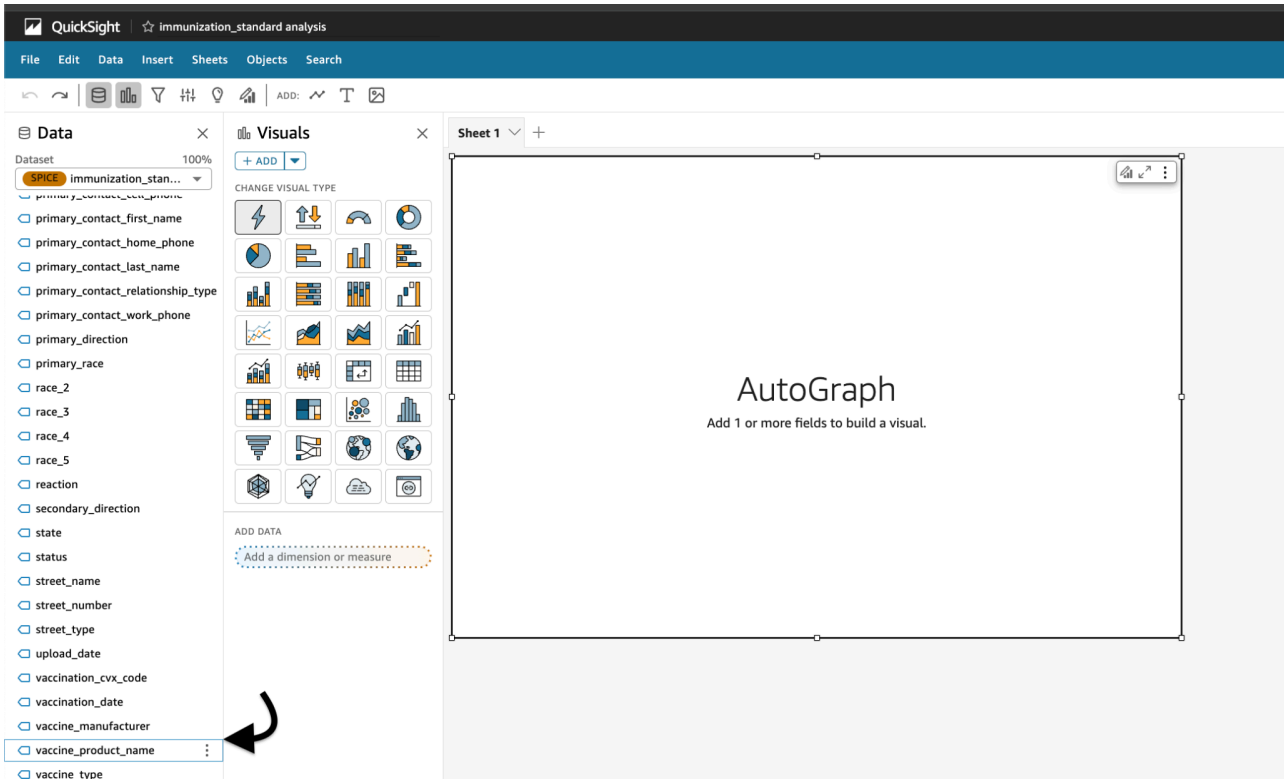
6. Click on Visualise



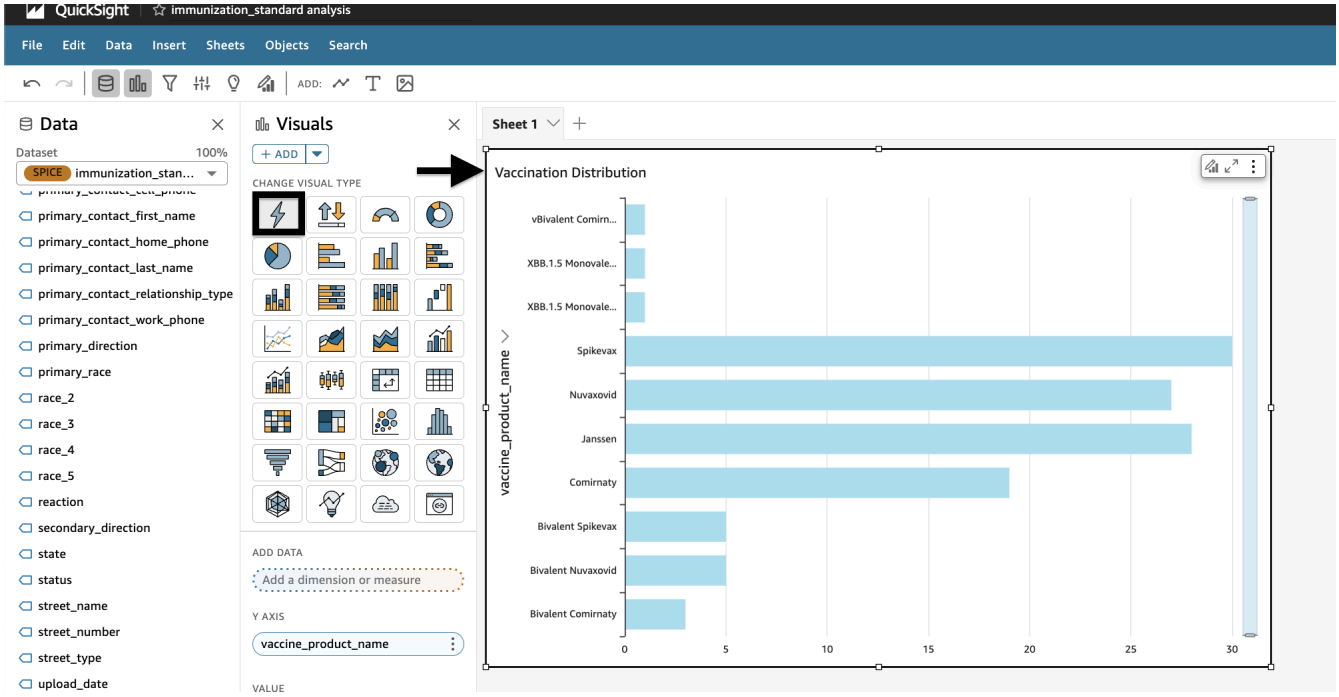
7. Click on Create to create a new sheet for analysis.



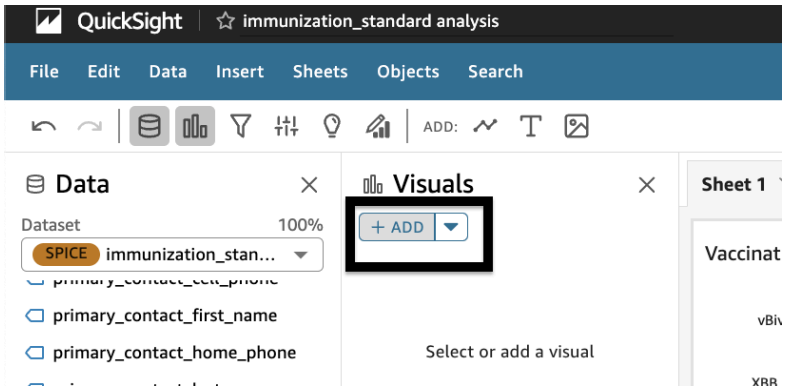
8. Let's create the first visualization.
   A. **Vaccination Distribution:** To analyze the number of different vaccinations administered to citizens.
   Drag the vaccine_product_name to build the visual graph on the sheet.



The AutoGraph will auto select the type of graph for you based on the dimension.
Update the title by double clicking on the title: Give it an appropriate name ‘Vaccination Distribution’ and Save.

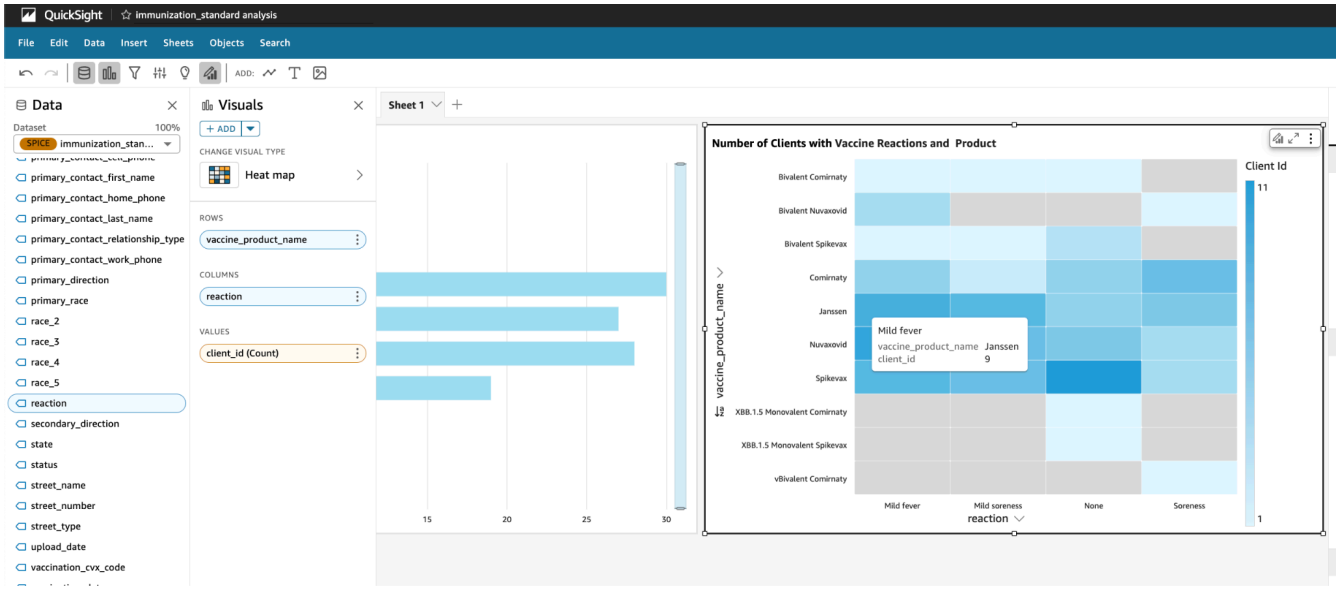


Click on Add Visuals to add another graph

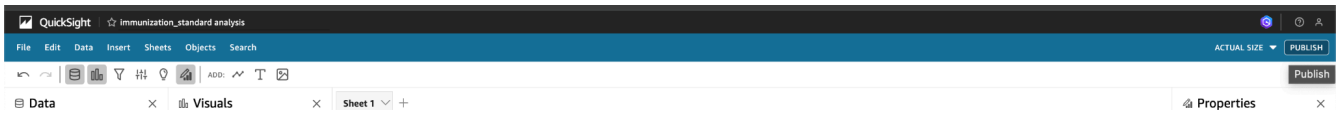


B. **Number of Clients with Vaccine Reactions and Product:** Count of clients who had a reaction categorized for each product.

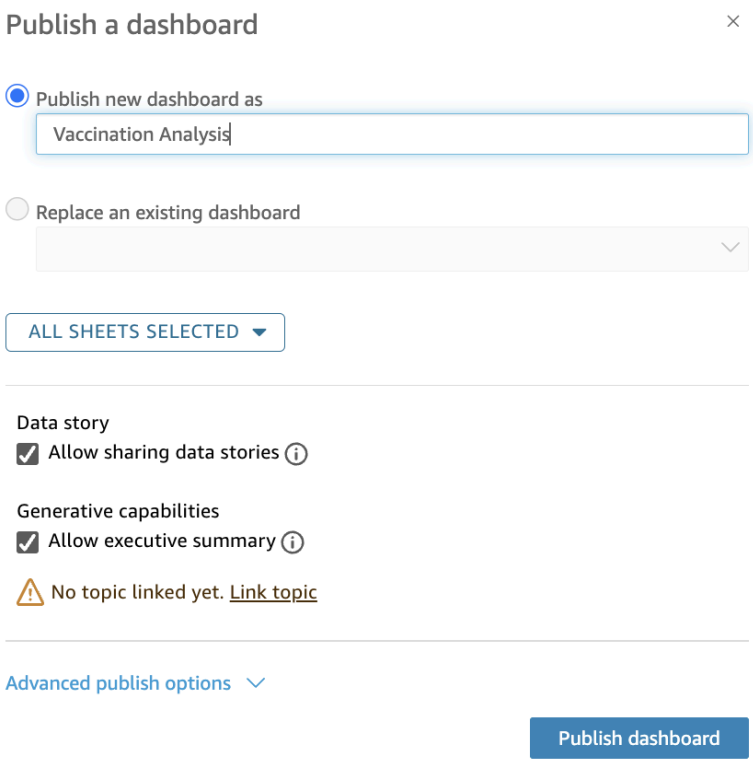
Let's create a heat map. Fill the row, columns and values as vaccine_product_name,reaction and client_id respectively.(See below SS for more info.) Update the title.



Click on Publish Dashboard to create a dashboard with the visuals you created.

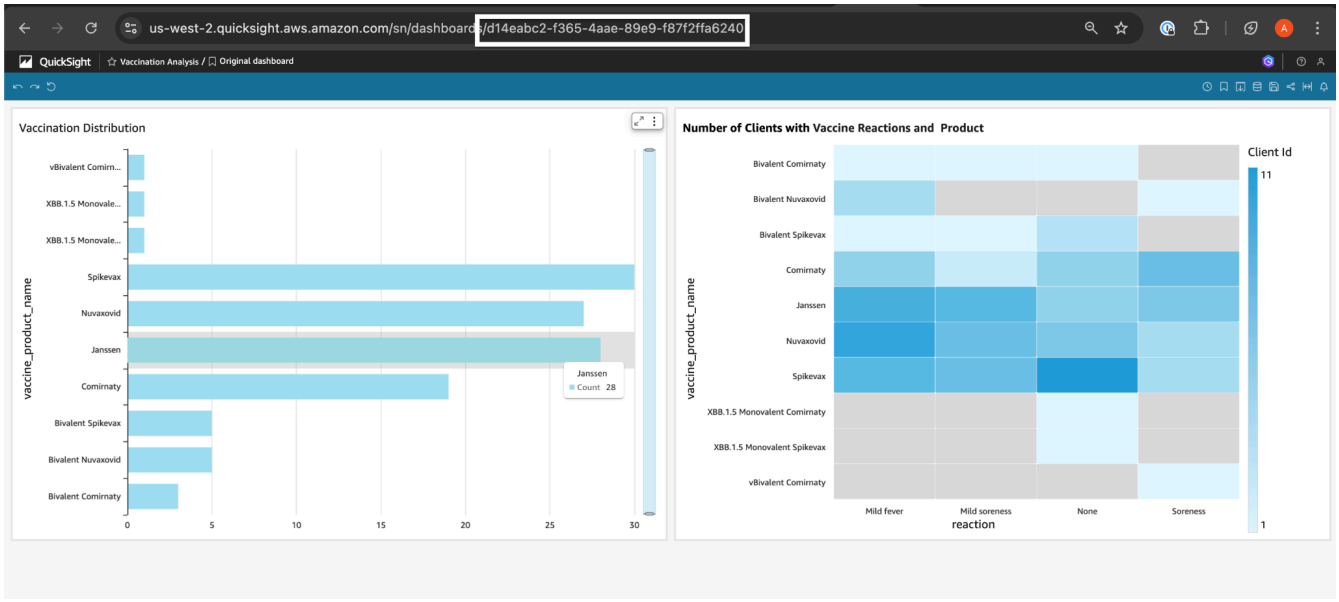


Give an appropriate name and click on the publish dashboard.



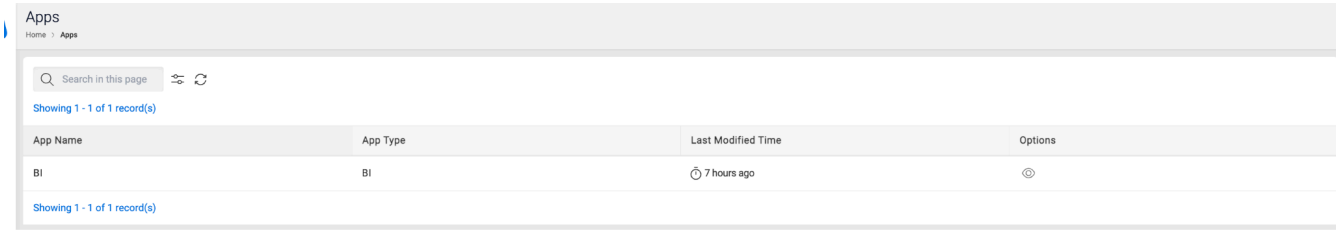
Your dashboard is now created.

9. Copy the Quicksight Dashboard ID and keep it handy.

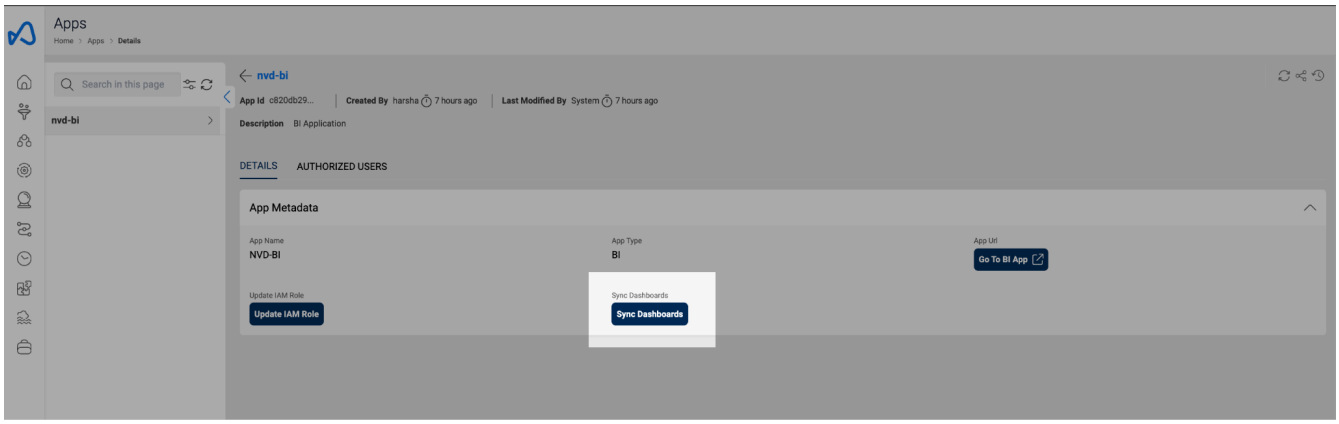


Sync Dashboard to Amorphic.

1. Back at the Amorphic UI. Navigate to Apps > List Apps. Click on the BI application from the list.

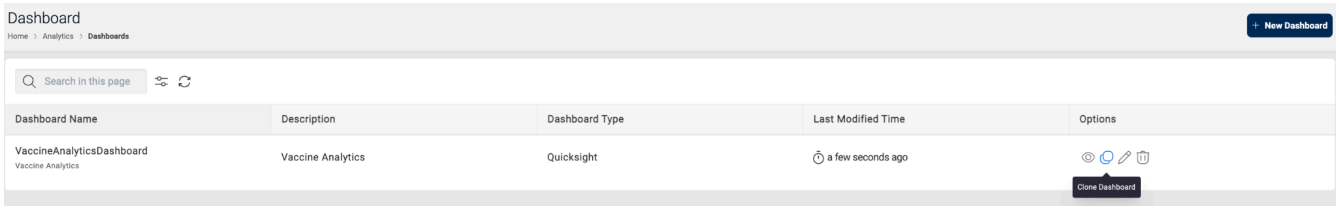


2. Click on Sync Dashboards to sync the dashboard to Amorphic.
You will see a notification that says “Successfully triggered dashboard sync process”

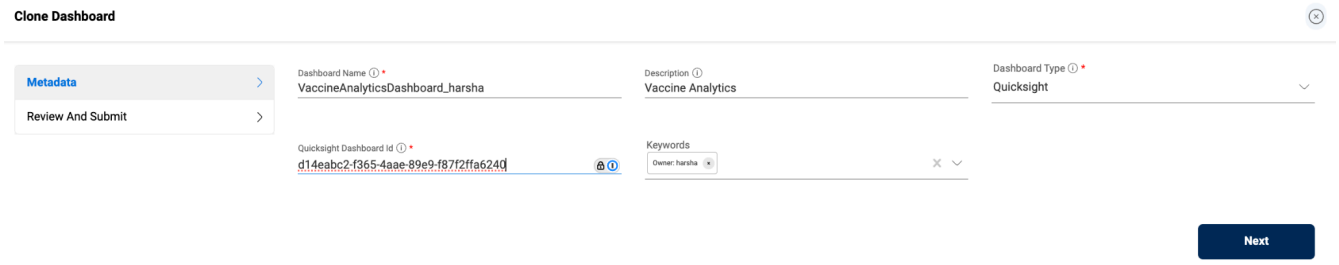


Amorphic Dashboard:

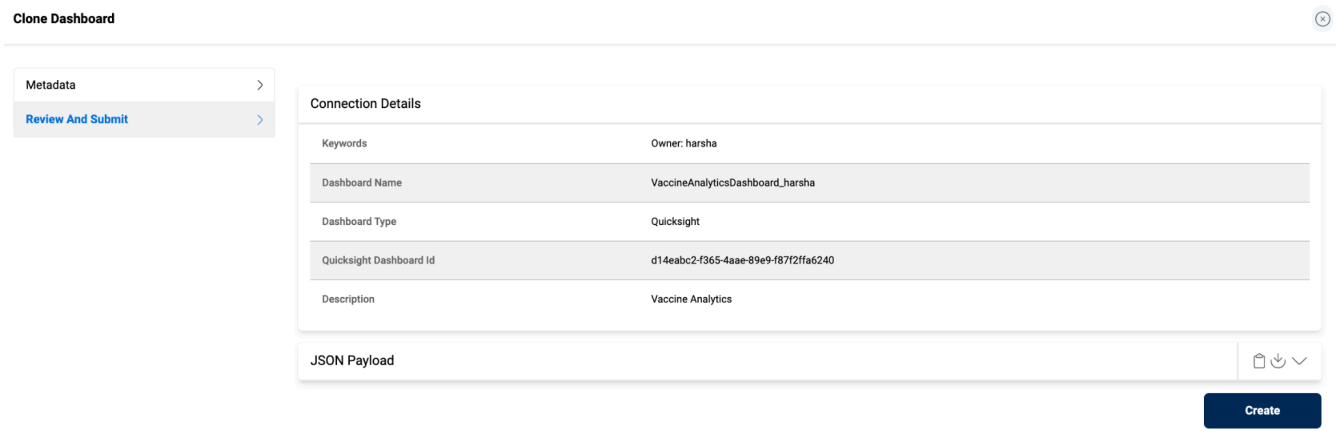
1. Navigate to Analytics > Dashboard. A sample dashboard has been added. Clone it.



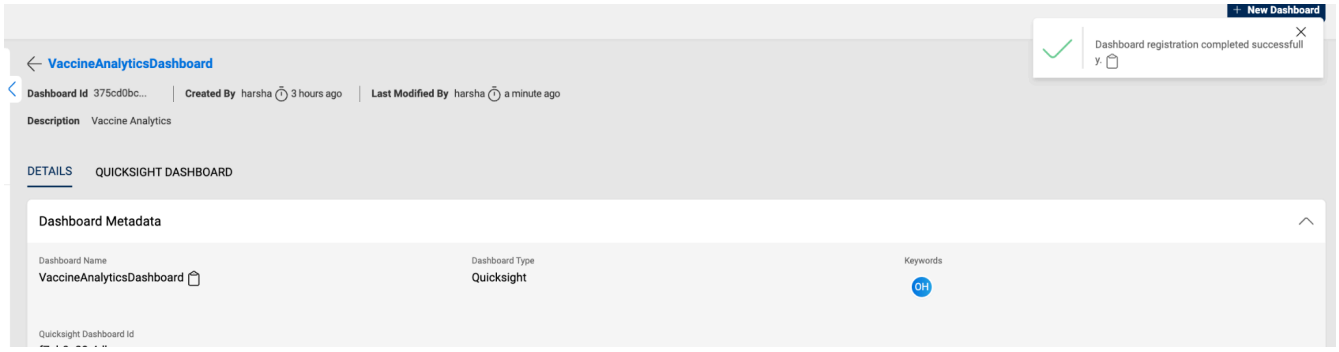
2. Update the Dashboard name (prefix it with your name) and QuickSight Dashboard Id that you copied in the previous step and the click on Next



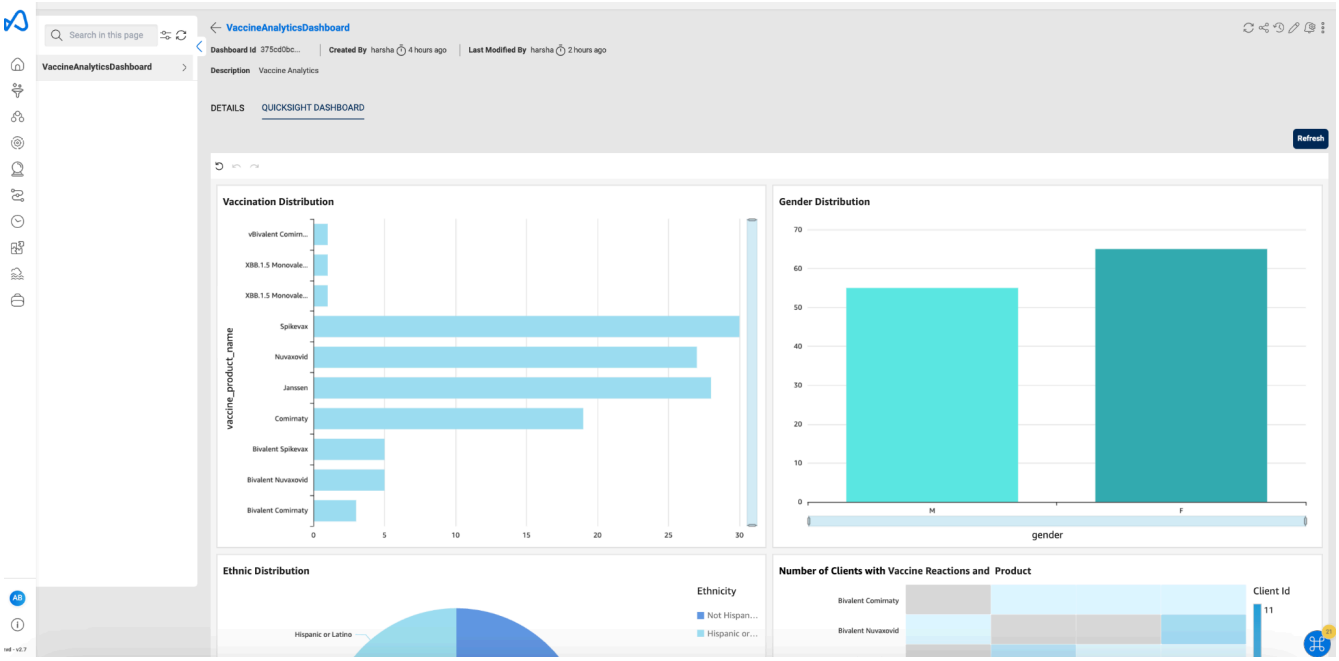
3. Finally click on Create to create the dashboard.



4. Dashboard will be created

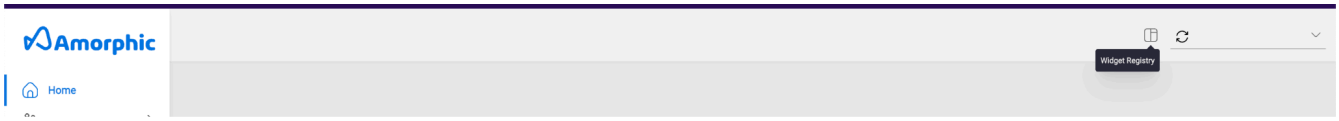


5. Click on Quicksight Dashboard to view the Dashboard.

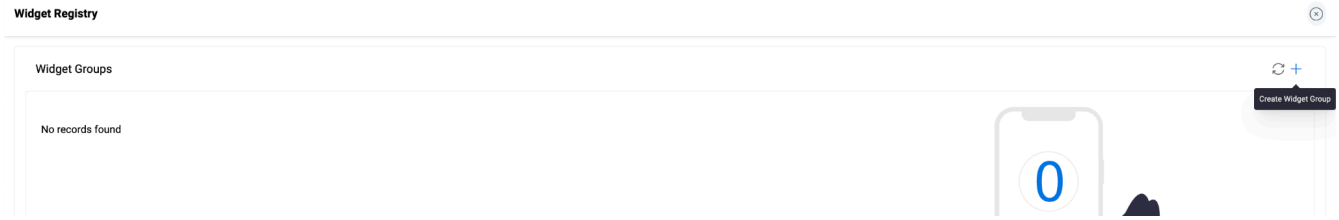


Adding Dashboard as widget:

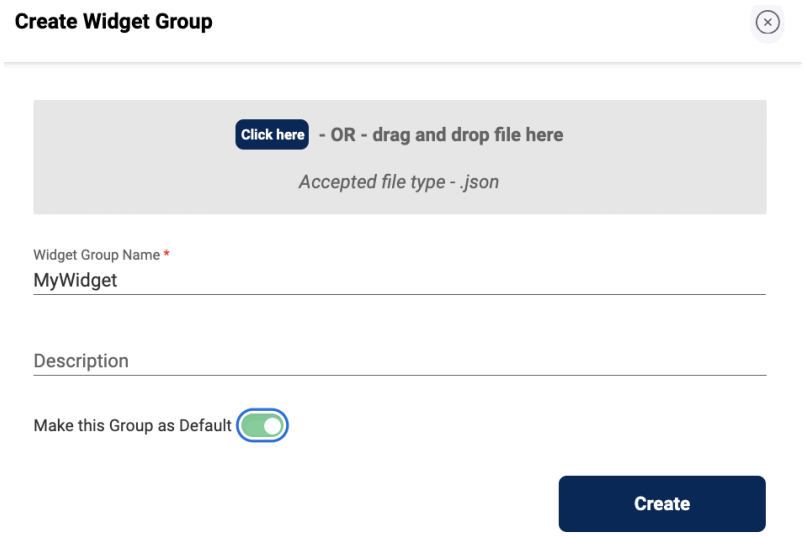
1. Navigate to home and click on widget registry to open it



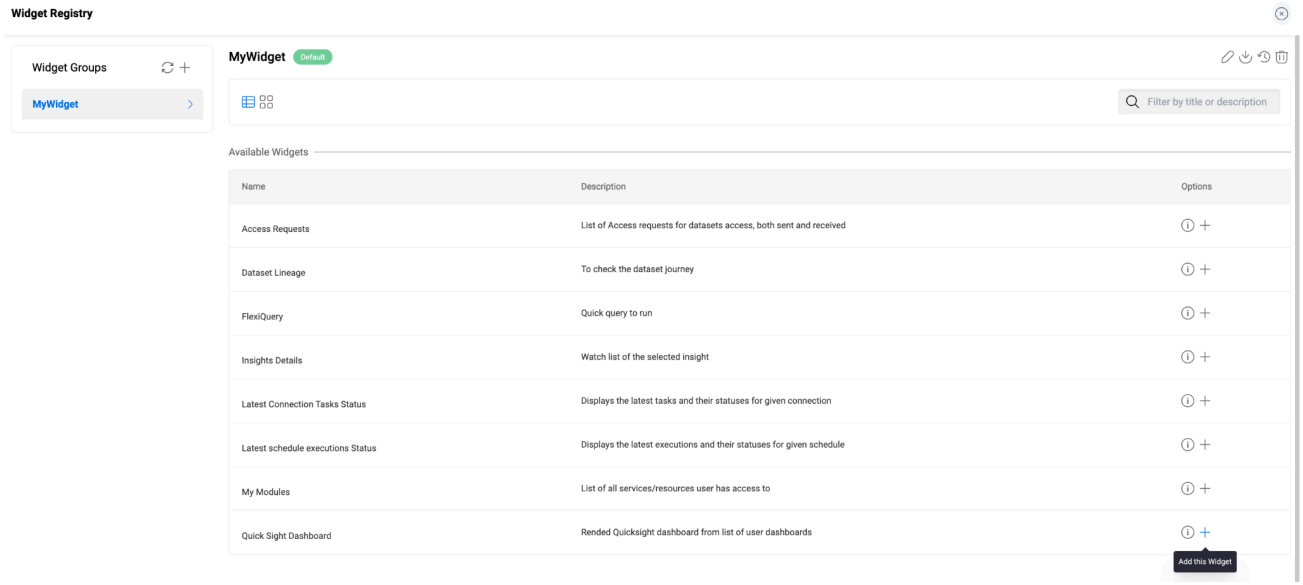
Click on create widget group.



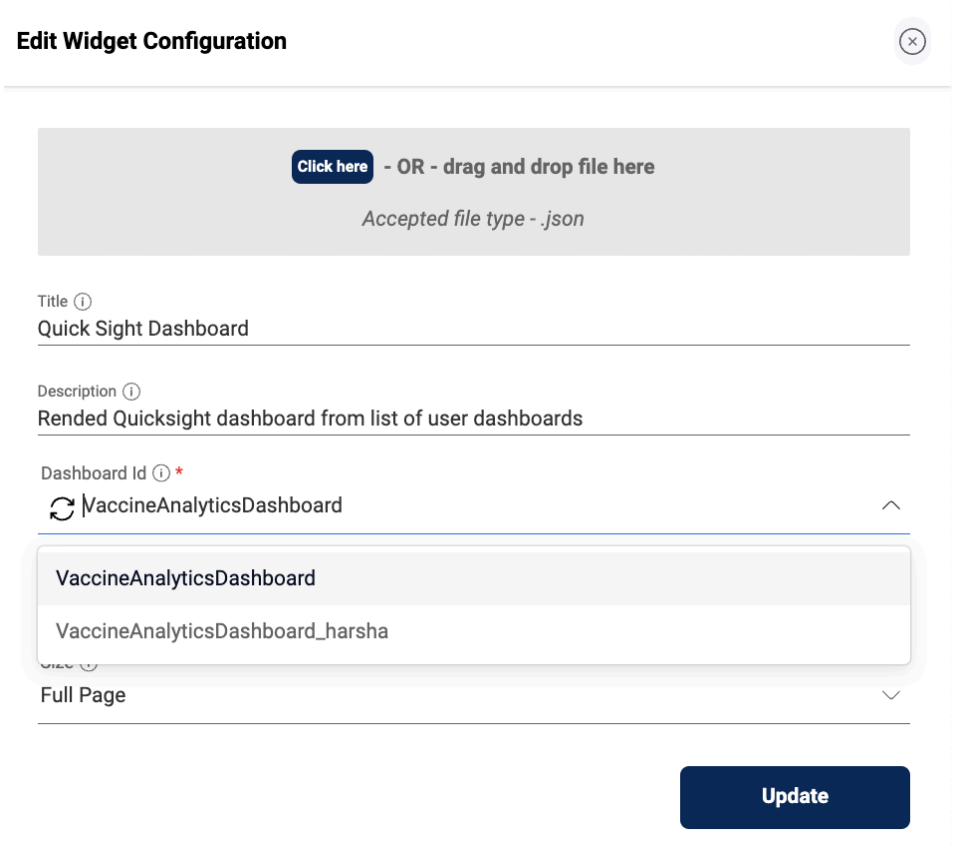
2. Give the widget a name and toggle the button to make this default and click on Create.



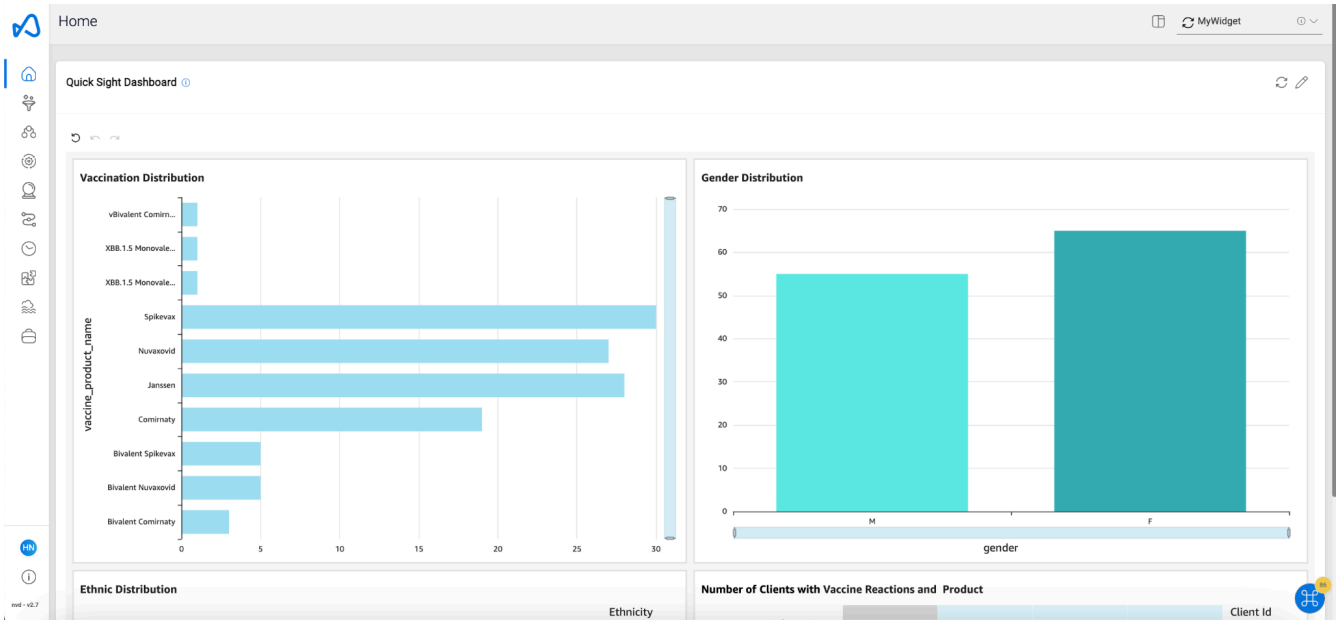
3. Choose a widget you want to add. In this case a quicksight Dashboard. Click on the corresponding + icon.



4. Choose from the available dashboard and click on update.



5. Go back to the Home Page and you should now see a QuickSight Dashboard you created as a widget.



LAB Session 6:

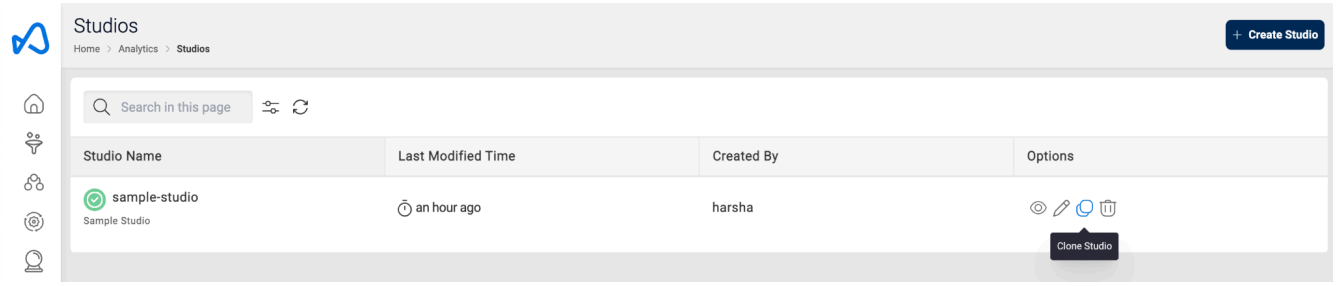
Studios:

Amazon SageMaker Studio is an integrated development environment(IDE) that provides a single web-based visual interface where you can access purpose-built tools to perform all machine learning (ML) development steps, from preparing data to building, training, and deploying your ML models.

Note: For the purpose of this demo we will only see the capability of the studio and will not be creating any models as it would take some time to train and procure the data. You can choose to experiment.

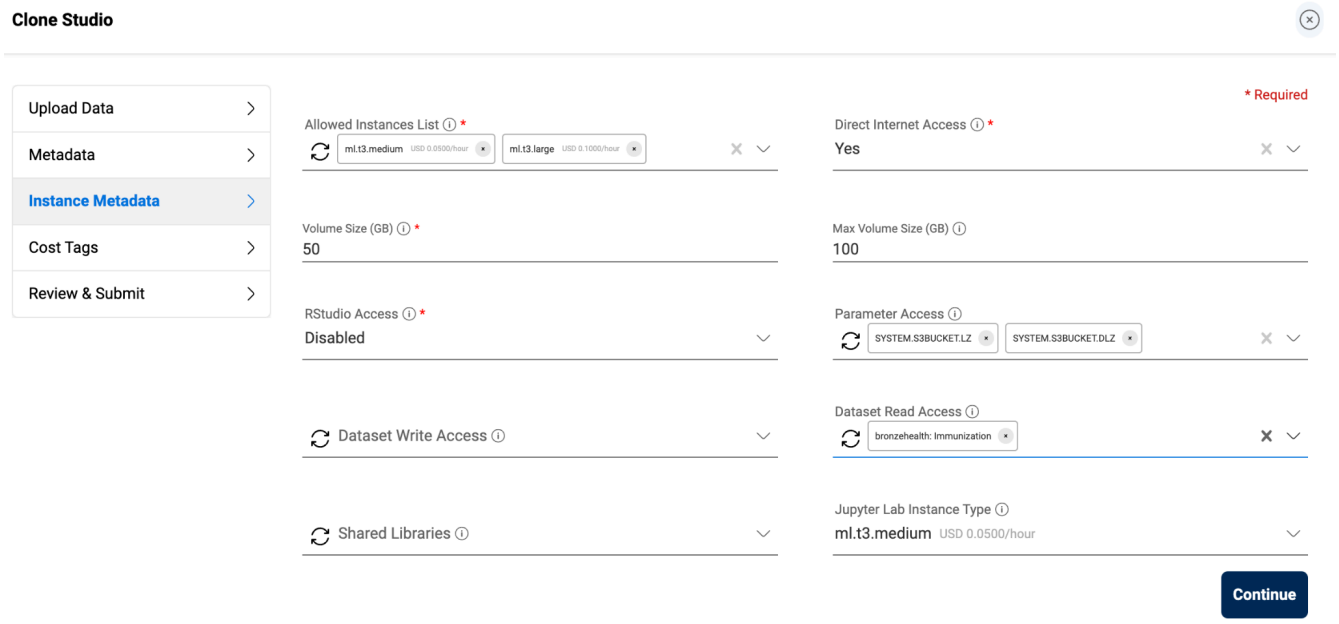
Procedure:

1. Navigate to Analytics > Studio
2. A studio has been shared with you. Let's create a new studio by cloning the existing ones. Click on the **Clone Studio** button to begin

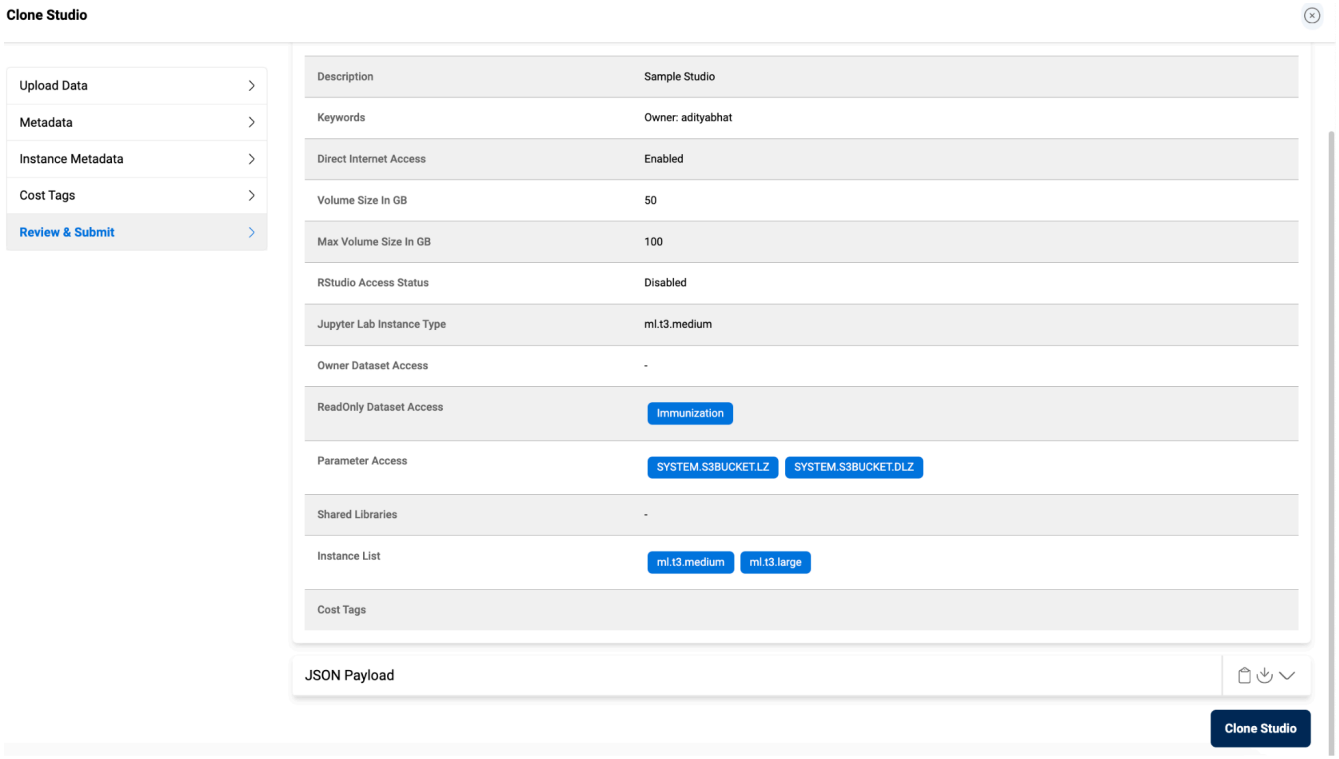


3. Give a studio a name. Rename ***sample-studio*** to **sample-studio-<yourname>** Ex: ***sample-studio-adityabhat***

4. Take a look at the properties in Instance Metadata. Give access to the Dataset you want to read and Parameter Access that needs to be accessed in the notebook. Click on **Continue.**

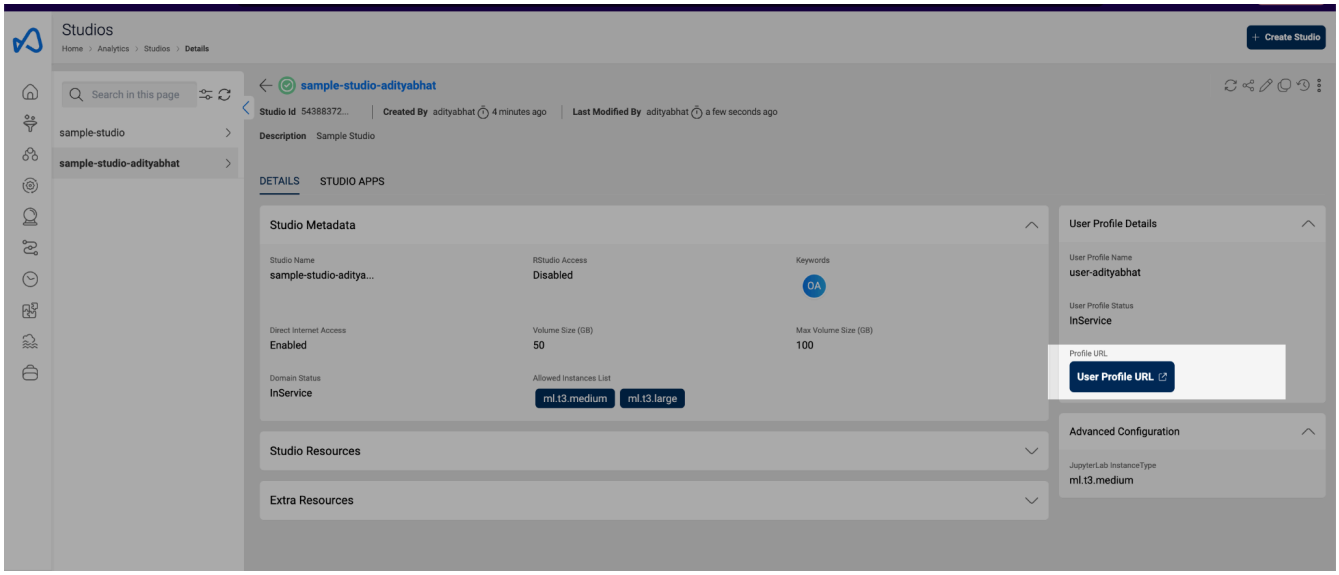


5. Skip Cost Tags and head to *Review and Submit*. Click on Clone Studio.



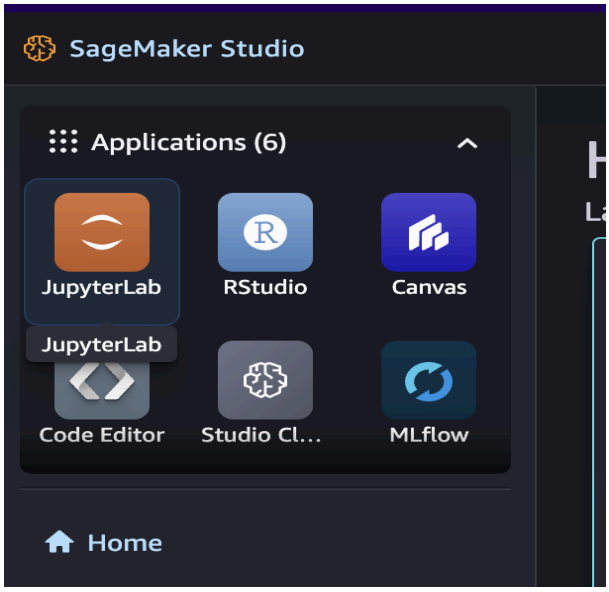
6. Wait for a few minutes for the studio instance to spin up. Your studio will be ready.

7. Once the instance is up click on the User Profile URL to jump directly to the studio.

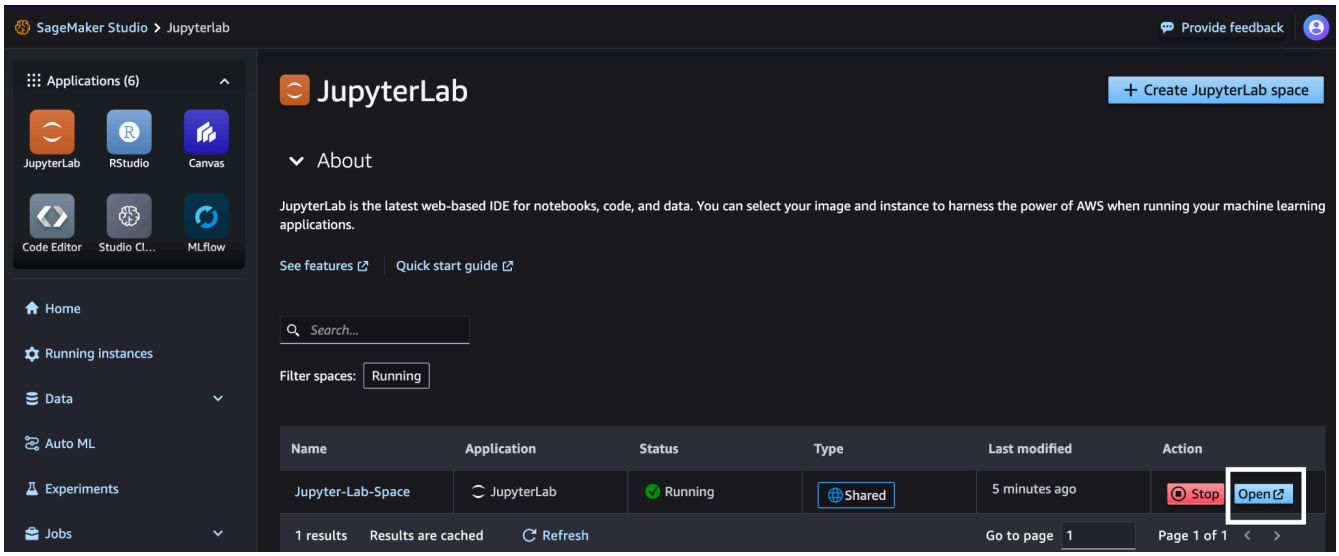


8. Take a tour of the studio if you would like to else skip tour

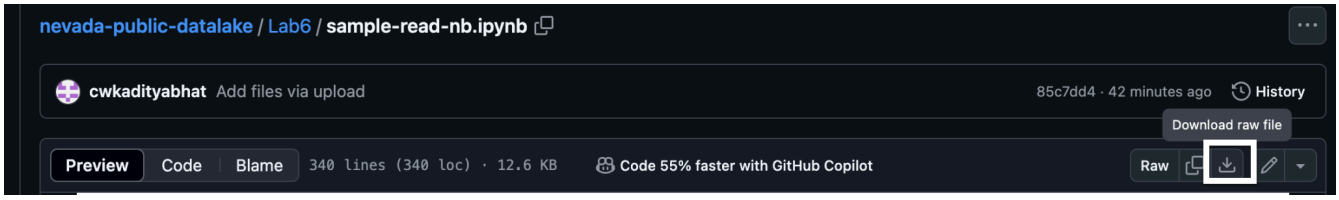
9. Choose JupyterLab under the Applications from the left pane.



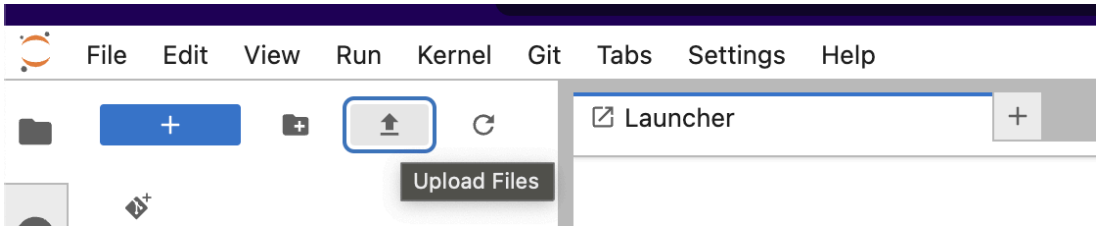
10. Open the JupyterLab.



11. For this lab we will only be reading from the Dataset. Download the script from the URL below. https://github.com/cwkadityabhat/nevada-public-datalake/blob/main/Lab6/sample-read-nb.ipynb

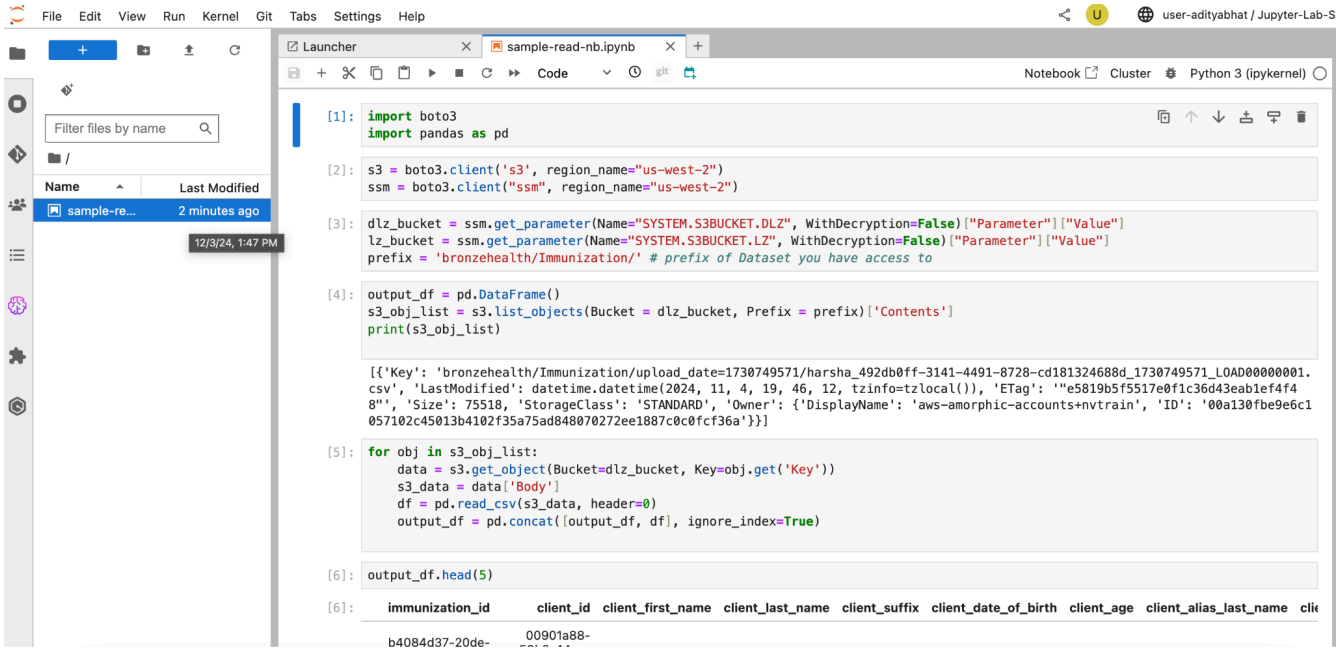


12. Lets upload the notebook downloaded to our JupyterLab.Back in the studio on the left hand pane on the top hand side you will see an upload button.

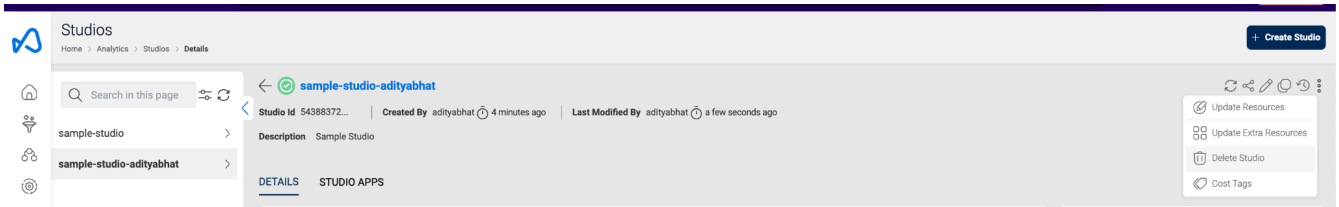


Click on it to choose the file you downloaded (sample-read-nb.ipynb).

13. You can now see the notebook uploaded. Double click on the notebook to open in the new tab.



14. Run each cell by clicking Shift + Enter or Run Icon on the top.
15. Once done to stop incurring any cost, Delete the studio.



Congrats!! You have successfully created the Studio ,Run a notebook and Delete a Studio.

Feel free to explore other great functionality of SageMaker studio
https://aws.amazon.com/sagemaker-ai/studio/
https://docs.aws.amazon.com/sagemaker/latest/dg/studio-updated.html