

Cloudwick



Runbook for Amorphic Training



Table of Contents

<u>Datasets:</u>	3
<u>Cloning a Dataset</u>	3
<u>Dataset Registration</u>	5
<u>Views in Amorphic:</u>	8
<u>Cloning a View</u>	8
<u>Connections:</u>	10
<u>Using a S3 Connection to Ingest data.</u>	10
<u>Scheduling Ingestion:</u>	15
<u>Creating a JDBC Connection</u>	21
<u>Tasks:</u>	24
<u>Paramstore:</u>	31
<u>ETL Library:</u>	32
<u>ETL Job:</u>	36
<u>Pre- Requisites:</u>	36
<u>Procedure:</u>	36
<u>Workflows:</u>	47
<u>Purpose:</u>	47
<u>Procedure:</u>	47
<u>Executing a workflow:</u>	55
<u>Notebook Lifecycle Configurations:</u>	58
<u>Procedure:</u>	58
<u>Code Repository:</u>	60
<u>Notebooks:</u>	61
<u>Procedure:</u>	61
<u>Using a Notebook:</u>	64
<u>Amorphic BI:</u>	68
<u>Creating a Quicksight Dashboard:</u>	68
<u>Sync Dashboard to Amorphic</u>	76
<u>Amorphic Dashboard:</u>	77
<u>Adding Dashboard as widget:</u>	79
<u>Studios:</u>	82
<u>Procedure:</u>	82
<u>References:</u>	87



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

The screenshot shows the Amorphic Dataset Details page for a dataset named 'sample_redshift'. The page includes a sidebar with navigation links like Home, Catalog, and Datasets. The main content area displays the dataset's ID, creation details, and a description. A context menu is open on the right side, with the 'Clone Dataset' option highlighted. Other options visible in the menu include 'Delete Dataset', 'Repair Dataset / Generate Dataset Report', and 'Cost Tags'.

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*



Screenshot of the Amorphic Datasets interface showing the 'Clone Dataset' process. The 'Metadata' section is selected, displaying fields for Dataset Name ('sample_redshift_yourname'), Description ('Sample Redshift Table'), Domain ('Bronze Education (bronzeeducation)'), and Keywords ('Owner: harsha'). The 'Basic Configuration' section includes Ingestion Type ('File Upload'), File Type ('csv'), Target Location ('Redshift'), Update Method ('Append'), Skip File Header ('Yes'), and Custom Delimiter (','). The 'Advanced Configuration' section includes Enable Malware Detection ('No'), Enable Data Profiling ('No'), and Enable Metrics Collection ('No'). A 'Next' button is visible at the bottom right.

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

Screenshot of the Amorphic Datasets interface showing the 'Clone Dataset' process. The 'Review and Submit' section is selected, displaying a JSON Payload window. The payload content is as follows:

```
1 v {
2   "DatasetName": "sample_redshift_harsha",
3   "DatasetDescription": "Sample Redshift Table",
4   "Domain": "bronzeeducation",
5   "Keywords": [
6     "Owner: harsha"
7   ],
8   "ConnectionType": "api",
9   "FileDelimiter": ",",
10  "FileType": "csv",
11  "IsDataProfilingEnabled": false,
12  "TargetLocation": "redshift",
13  "MalwareDetectionOptions": {
14    "ScanForMalware": false,
15    "AllowUnscannablefiles": false
16  },
17  "SkipFileHeader": true,
18  "SkipLZProcess": false,
19  "TableUpdate": "append",
20  "DataMetricsCollectionOptions": {
21    "IsMetricsCollectionEnabled": false
22  },
23  "DatasetType": "internal",
24  "DatasetS3Path": "s3://nvd-us-west-2-816069158041-test-dlz/bronzeeducation/sample_redshift/"
25 }
```

A 'Clone' button is visible at the bottom right.



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.

- upload a sample data file
- upload a JSON file containing the schema definition
- manually enter the schema fields.



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

The screenshot shows the 'Create Schema Manually' step in the dataset creation wizard. It has a sidebar with 'Define Schema' and 'Review and Publish' options. The main area shows 'Schema Entry Type' with 'Create Schema Manually' selected. A message says 'You have selected to manually create the schema'. A 'Continue' button is at the bottom right.

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.

The screenshot shows the 'Build Custom Schema' step for a Redshift dataset. It has a sidebar with 'Define Schema' and 'Review and Publish' options. The main area shows a table for adding columns:

Column Name	Column Type
column1	VARCHAR(256)
column2	VARCHAR(256)
column3	VARCHAR(256)

Below the table are dropdowns for Primary Keys, Unique Columns, Define Identity Column, Sort Key Type (set to 'None'), and Distribution Type (set to 'Auto'). A 'Set all columns to VARCHAR(256) Type' button is visible. A 'Publish Dataset' button is at the bottom right.

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



Completed the registration process successfully

sample_redshift_harsha

Published

Dataset Id f7214090... | Created By harsha a few seconds ago | Last Modified By harsha a few seconds ago

Description Sample Redshift Table

DETAILS PROFILE FILES RESOURCES

Dataset Metadata

Dataset Name sample_redshift_hars...	Dataset Type internal	Domain bronzeeducation
Target Location Redshift	File Type csv	Ingestion Type File Upload
Data Classification	Dataset S3 Location s3://nvd-us-west-2-816069158041-test-diz/bronzeeducation/sample_redshift_harsha/	
Keywords OH	SerDe	Delta Lake Table No

Redshift Configuration

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.

The screenshot shows the Amorphic interface for viewing a specific view. The title bar says 'Views' and the path is 'Home > Analytics > Views > Details'. The main content area displays the view 'immunization_vitals_summary' with the following details:

- View Id: 44312e80...
- Created By: harsha (17 minutes ago)
- Last Modified By: harsha (17 minutes ago)
- Description: view provides a summary of immunization records, focusing specifically on deceased individuals.

Below this, there are tabs for 'DETAILS', 'SCHEMA DETAILS', and 'VERSIONS'. At the bottom, there are buttons for 'View Metadata' and 'Additional Metadata'. In the top right corner of the main content area, there is a blue button labeled 'Clone View' with a white icon.

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



Clone View

View Name: immunization_vitals_summary_yourname

Description: view provides a summary of immunization records,

Keywords: Owner: harsha

Next

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

Clone View

View Type: Standard

Domain: Gold Health (goldhealth)

Target Location: S3athena

Sql To Create View:

```
1. create view
2. goldhealth.immunization_vitals_summary_yourname
3. as (select vaccine_product_name,COUNT(vaccine_product_name) as deceased_count from goldhealth.immunization_standard
4. where client_id in (select case_id from goldhealth.vitals_standard where deceased = 'TRUE')
5. group by vaccine_product_name)
```

Next

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

Clone View

View Details

View Name: immunization_vitals_summary_yourname

Description: view provides a summary of immunization records, focusing specifically on deceased individuals.

Keywords: Owner: harsha

Data Classifications: -

View Type: Standard

Assume IAM role: no

Domain: goldhealth

Target Location: S3athena

Sql To Create View:

```
create view goldhealth.immunization_vitals_summary_yourname as (select
vaccine_product_name,COUNT(vaccine_product_name) as deceased_count from goldhealth.immunization_standard
where client_id in (select case_id from goldhealth.vitals_standard where deceased = 'TRUE')
group by vaccine_product_name)
```

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.

The screenshot shows the Amorphic Connections interface. On the left, there's a sidebar with a search bar and navigation links for Home, Ingestion, Connections, and Details. The main area displays a list of connections: mysql-connection-demo and s3-connection-demo. The s3-connection-demo connection is selected and shown in detail. The connection metadata includes:

- Connection Name: s3-connection-demo
- Connection Type: S3
- Version: 2.4
- Keywords: OH
- Estimated Cost: 0.01

Below this, the S3 Bucket Metadata section shows:

- S3 Bucket: cw-us-east-1-728673471265-cloudwick-demo
- S3 Bucket Region: us-east-1
- Access Type: bucket_policy
- Bucket Policy: View Bucket Policy
- KMS Key Policy: View KMS Policy

A context menu on the right side of the connection card offers options like Edit Connection, Notification Settings, Clone Connection, Delete Connection, and Cost Tags.

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



Amazon S3 > Buckets > cw-us-east-1-728673471265-cloudwick-demo > data/ > health/

health/

Objects Properties

Objects (2) Info

Copy S3 URI Copy URL Download Open Delete Actions Create

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you must grant them permissions. [Learn more](#)

Find objects by prefix

<input type="checkbox"/>	Name	Type	Last modified	Size
<input type="checkbox"/>	<input type="checkbox"/> Immunization/	Folder	-	
<input type="checkbox"/>	<input type="checkbox"/> Vitals/	Folder	-	

1. 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.

Datasets

Home > Catalog > Datasets

+ New Dataset

🔍 Vital Increase the Results Per Page

Showing 1 - 2 of 46 record(s)

Dataset Name	Domain	File Type	Target Location	Options
Vitals	bronzehealth	csv	s3athena	<input type="button"/> <input type="button"/> <input type="button"/> <input type="button"/>
vitals_standard	goldhealth	csv	s3athena	<input type="button"/> <input type="button"/> <input type="button"/> <input type="button"/>

Showing 1 - 2 of 46 record(s)

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



The screenshot shows the Amorphic Dataset Details page for a dataset named 'Vitals'. The top navigation bar includes 'Datasets', 'Home > Catalog > Datasets > Details', and a 'New Dataset' button. The main content area has tabs for 'DETAILS', 'PROFILE', 'FILES', and 'RESOURCES'. The 'DETAILS' tab is selected, showing 'Dataset Metadata' and 'Connection Details'. In the 'Dataset Metadata' section, fields include 'Dataset Name' (Vitals), 'Dataset Type' (internal), 'Domain' (bronzehealth), 'Target Location' (S3-Athena), 'File Type' (csv), 'Ingestion Type' (s3), 'Data Classification' (None), 'Dataset S3 Location' (s3://nvd-us-west-2-816069158041-test-diz/bronzehealth/Vitals/), 'Keywords' (OpenCSVSerde), and 'SerDe' (OpenCSVSerde). In the 'Connection Details' section, fields include 'Connection Name' (s3-connection-demo), 'Connection Id' (6a1fab51-effb-49d6-b817-4acf7018dd), and 'Source S3 Bucket Prefix' (data/health/Vitals/). A context menu on the right side offers options like 'Notification Settings', 'Clone Dataset', 'Delete Dataset', 'Repair Dataset / Generate Dataset Report', and 'Cost Tags'. A 'File Options' section contains 'Advanced Options'.

3. 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)

The screenshot shows the Amorphic Dataset Details page for a dataset named 'vita'. The top navigation bar includes 'Datasets', 'Home > Catalog > Datasets > Details', and a 'Download' button. The main content area has tabs for 'DETAILS', 'PROFILE', 'FILES', and 'RESOURCES'. The 'PROFILE' tab is selected, showing 'Dataset Schema'. The schema is displayed as a JSON object with 26 numbered lines. The schema defines columns for 'case_id', 'case_file_number', and 'local_file_number' with their respective data types (varchar(256)) and descriptions. The 'FILES' tab is also visible, showing a table with columns 'Name', 'case_id', 'case_file_number', and 'local_file_number', with data types 'varchar(256)', 'varchar(256)', and 'varchar(256)' respectively.

```
1. {  
2.   "columnName": "case_id",  
3.   "columnType": "varchar(256)",  
4.   "description": ""  
5. },  
6. {  
7.   "columnName": "case_file_number",  
8.   "columnType": "varchar(256)",  
9.   "description": ""  
10. },  
11. {  
12.   "columnName": "local_file_number",  
13.   "columnType": "varchar(256)",  
14.   "description": ""  
15. },  
16. {  
17.   "columnName": "state_file_number",  
18.   "columnType": "varchar(256)",  
19.   "description": ""  
20. },  
21. {  
22.   "columnName": "first_name",  
23.   "columnType": "varchar(256)",  
24.   "description": ""  
25. },  
26. }.
```

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.



4. 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.

Clone Dataset (x)

Metadata >
Cost Tags >
Review and Submit >

JSON Payload

```
1 v {  
2   "DatasetName": "Vitals_S3_Harsha",  
3   "DatasetDescription": "Vitals | S3 Ingestion",  
4   "Domain": "bronzehealth",  
5 v   "Keywords": [  
6     "Owner: harsha"  
7   ],  
8   "ConnectionType": "s3",  
9   "ConnectionId": "6a1fab51-ffb-49d6-b817-4acf7018dd",  
10  "IsDataValidationEnabled": false,  
11  "SerDe": "openCSVSerde",  
12  "DirectoryPath": "data/health/Vitals/",  
13  "FileDelimiter": ",",  
14  "FileType": "csv",  
15  "IsDataCleanupEnabled": false,  
16  "IsDataProfilingEnabled": false,  
17  "LifeCyclePolicyStatus": "Disabled",  
18  "TargetLocation": "s3athena",  
19  "SkipFileHeader": true,  
20 v  "SkipRowCount": {  
21    "header": 1,  
22    "footer": 0  
23  },  
24  "SkipLZProcess": false,  
25  "TableUpdate": "append",  
26 v  "DataMetricsCollectionOptions": {  
27    "IsMetricsCollectionEnabled": false  
28  },  
29 v  "IcebergTableOptions": {  
30    "IsIcebergTable": false,  
31    "TableProperties": {}  
32  },  
33  "DatasetType": "internal",  
34  "DatasetS3Path": "s3://nvd-us-west-2-816069158041-test-dlz/bronzehealth/Vitals/"  
35 }
```

Clone

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 No Schema

Dataset Id a4da0c54... | Created By harsha a few seconds ago | Last Modified By harsha a few seconds ago

Description Vitals | S3 Ingestion

Registration pending

Click here to continue Dataset registration

DETAILS

Dataset Metadata

Dataset Name Vitals_S3_Harsha	Dataset Type internal	Domain bronzehealth
Target Location S3-Athena	File Type CSV	Ingestion Type s3
Data Classification -	Dataset S3 Location s3://nvd-us-west-2-816069158041-test-dlz/bronzehealth/Vitals_S3_Harsha/	
Keywords OH	SerDe OpenCSVSerde	Delta Lake Table No

File Options

Advanced Options

Vitals_S3_Harsha

Define Schema >

Schema Entry Type *

Upload Sample Data File Upload Schema Definition JSON File Create Schema Manually

Review and Publish >

Click here - OR - drag and drop file here
Accepted file type - json

Schema definition JSON Template:

```
1 v [
2 v   [
3 v     {
4 v       "columnName": "<Column_Name_1>",
5 v       "columnType": "<Integer | Double Precision | Boolean | Varchar(65535) | Varchar(256) | Numeric | Date Time | Date",
6 v       "description": "<String>"
7 v     },
8 v     [
9 v       {
10 v         "columnName": "<Column_Name_2>",
11 v         "columnType": "<Integer | Double Precision | Boolean | Varchar(65535) | Varchar(256) | Numeric | Date Time | Date",
12 v         "description": "<String>"
13 v       },
14 v       {
15 v         "columnName": "<Column_Name_n>",
16 v         "columnType": "<Integer | Double Precision | Boolean | Varchar(65535) | Varchar(256) | Numeric | Date Time | Date",
17 v         "description": "<String>"
18 v     ]
19 v ]
```

Continue

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



Vitals_S3_Harsha

34.	> method_of_disposition	VARCHAR(256)	X
35.	> cemetery_name	Column Type* VARCHAR(256)	X
36.	> cemetery_address	Column Type* VARCHAR(256)	X
37.	> informant_name	Column Type* VARCHAR(256)	X
38.	> informant_relationship	Column Type* VARCHAR(256)	X

Add Column

Custom Partition Options

Add Partition Key

Publish 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.



Amorphic

+ New Schedule

Home Ingestion Catalog Transformation Analytics Workflows Schedules Apps HCLS Management

Job Name	Description	Job Type	Last Modified	Schedule Type	Options
SCH_JDBC_Data_Ingestion	Schedule to Ingest Vitals data	ingestion	6 hours ago	On-Demand	
SCH_S3_Vitals_Ingestion	Schedule to Ingest Vitals data	ingestion	3 days ago	On-Demand	

2. Clone the Schedule [SCH_S3_Vitals_Ingestion](#) and rename the schedule to SCH_S3_Vitals_Ingestion_<yourname> Ex: SCH_S3_Vitals_Ingestion_harsha

Schedules

Home > Schedules

+ New

Search in this page

Showing 1 - 2 of 2 record(s)

Job Name	Description	Job Type	Last Modified	Schedule Type	Options
SCH_JDBC_Data_Ingestion	Schedule to Ingest Vitals data	ingestion	6 hours ago	On-Demand	
SCH_S3_Vitals_Ingestion	Schedule to Ingest Vitals data	ingestion	3 days ago	On-Demand	

Showing 1 - 2 of 2 record(s)

Clone Schedule



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

Clone Schedule

Upload Data >

Metadata > Schedule Name ⓘ *
SCH_S3_Vitals_Ingestion_harsha

Cost Tags >

Review & Submit >

Job Type ⓘ * Data Ingestion

Description ⓘ Schedule to Ingest Vitals data

Keywords Owner: harsha

Schedule Type ⓘ * On Demand

Select Dataset ⓘ * C Vitals_S3_Harsha | s3

Max Capacity ⓘ

New Arguments ⓘ New Arguments

Continue

4. Skip Cost tags > Review > Clone

Clone Schedule

Upload Data >

Metadata >

Cost Tags >

Review & Submit >

Schedule Details

Job Name	SCH_S3_Vitals_Ingestion_harsha
Job Type	ingestion
Description	Schedule to Ingest Vitals data
Keywords	Owner: harsha
Resource	a4da0c54-ccf4-4c40-8d7b-4c30d73d21bb
Schedule Type	none
Arguments	1 []
Cost Tags	

JSON Payload

Clone

The schedule creation is now complete.



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

The screenshot shows the Amorphic interface for managing schedules. The top navigation bar includes a search bar, a user icon, and a 'New Schedule' button. The main content area displays a schedule named 'SCH_S3_Vitals_Ingestion_harsha' which is currently 'ENABLED'. The schedule details include:

- Schedule Name: SCH_S3_Vitals_Ingestion_harsha
- Status: ENABLED
- Job Type: ingestion
- Schedule Type: on-demand
- Dataset: bronzehealth:Vitals_S3_Harsha
- Keywords: OH
- Arguments: datasetId: a4da0c54-ccf4-4c40-8d7b-4c30d73d21bb, MaxCapacity: 0.0625

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

The screenshot shows the Amorphic interface displaying the execution history for the 'SCH_S3_Vitals_Ingestion_harsha' schedule. The history table shows one record:

Start Time	End Time	Job Type	Message	Log Status	Options
2 minutes ago	a minute ago	glue	Copied 1 number of files. For more details, download logs.	-	



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

The screenshot shows the Amorphic Datasets interface. On the left, there's a sidebar with icons for Home, Ingestion, Catalog, Transformation, Analytics (which is selected), Workflows, and Schedules. The main area shows a list of datasets: PH_Vitals, Vitals, Vitals_S3_Harsha (selected), and vitals_standard. The 'Vitals_S3_Harsha' dataset details are shown: Dataset Id: a4da0c54..., Created By: harsha (19 minutes ago), Last Modified By: harsha (15 minutes ago). Description: Vitals | S3 Ingestion. The 'FILES' tab is selected. It displays a table with one record: File Name: ...981962_PH_Vitals.csv, Last Modified: harsha 3 minutes ago. A message box contains the text: 'Data load completed.'

Querying the Ingested S3 Data:

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

The screenshot shows the Amorphic Query Engine interface. The left sidebar has sections for Home, Ingestion, Catalog, Transformation, Analytics (selected), Workflows, and Schedules. The main area has tabs for Views, Query Engine (selected), Insights, Dashboards, Notebooks, Notebooks Lifecycle Configurations, and Studios. A 'New Query' button is at the top. Below it, a code editor shows the query: '1 select * from health.Vitals'. It is set to run on 'Athena' and 'Primary'. There are 'Clear' and 'Run Query' buttons. The results section is currently empty.

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



The screenshot shows the Amorphic Query Engine interface. On the left, the Schema Explorer pane displays a hierarchical tree of datasets under a target location of 'Athena'. A dataset named 'gold (gold)' is highlighted with a red circle. The main area is the 'New Query' editor, which contains a query window with the following code:

```
1 Select * from
... limit 50
```

Below the code, there are tabs for 'Athena' and 'Primary', and buttons for 'Clear' and 'Run Query'. The 'RESULT' pane below is currently empty, showing 'No Results'.

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

The screenshot shows the Amorphic Query Engine interface after the query has been run. The Schema Explorer pane remains the same. The New Query editor now contains the executed query code:

```
1 select * from
2 bronzehealth.Vitals_S3_Harsha
3 limit 50
```

The 'RESULT' pane displays the query results as a table:

case_id	case_file_number	local_file_number	state_file_number	first_name	last_name	...
00901a88-59b2-4acc-a...	CF-356	LF-188	SF-238	Saltzman327	Elaine169	J...
03b86e36-59f5-4db3-a...	CF-478	LF-346	SF-783	Washington743	Darius291	I...
069335e0-adddd-4969-b...	CF-159	LF-277	SF-154	Johnson823	Terrell657	J...
06e441e3-2ed3-4f08-b...	CF-656	LF-909	SF-895	Nguyen378	Minh621	J...

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



Creating a JDBC Connection

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

The screenshot shows the Amorphic Connections interface. On the left, there's a sidebar with various icons for different services. The main area displays a list of connections: 'mysql-connection-demo' and 's3-connection-demo'. Below this, the 'mysql-connection-demo' connection is selected and shown in detail. The 'DETAILS' tab is active, displaying 'Connection Metadata' and 'Jdbc Connection Metadata'. In the 'Jdbc Connection Metadata' section, the 'JDBC Connection URL' field contains the value 'jdbc:mysql://database-1-instance-1.cozbikcm76r.us-east-1.rds.amazonaws.com:3306/demo'. A context menu is open on the right side of the screen, with 'Clone Connection' being the highlighted option.

- Leave the ConnectionType as JDBC and click on Continue.

The screenshot shows the 'Clone Connection' wizard. The left sidebar lists steps: 'Upload Data', 'Connection Type' (which is currently selected), 'Metadata', 'Connection Configuration', 'Cost Tags', and 'Review & Submit'. The main area shows a 'Database Type' section with icons for Oracle, MySQL, PostgreSQL, Microsoft SQL Server, IBM Db2 (LUW), Amazon Redshift, Amazon Aurora, Maria DB, and Aurora MySQL. Below this, a 'Connection Type' section shows icons for JDBC, S3, External API, and Email, with 'JDBC' being the selected option. At the bottom right is a 'Continue' button.



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.

Clone Connection

Upload Data >

Connection Type >

Metadata >

Connection Configuration >

Cost Tags >

Review & Submit >

Data Load Type ⓘ * Bulk Data Load

JDBC Connection URL ⓘ * jdbc:mysql://database-1-instance-1.cozbikcm76r.us-east-1.rds.amazonaws.com:3306/demo

Click here ⓘ for pre-reqs

Username ⓘ * admin

Password ⓘ * Cloudwick#123

Connection Accessibility ⓘ Private Connection Publicly Accessible Connection

Continue

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

Clone Connection

Upload Data >

Connection Type >

Metadata >

Connection Configuration >

Cost Tags >

Review & Submit >

Connection Details

Connection Type: jdbc

Connection Name: mysql-connection-demo-harsha

Description: Connection of MySQL Database | Ingestion

Keywords: DB : MySQL, Owner: harsha

Public Accessibility: yes

Data Load Type: bulkdataloadv1

Password: ****

JdbcURL: jdbc:mysql://database-1-instance-1.cozbikcm76r.us-east-1.rds.amazonaws.com:3306/demo

Cost Tags

JSON Payload

Clone



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

The screenshot shows the details of a MySQL connection named "mysql-connection-demo-harsha".

Connection Metadata:

- Connection Name: mysql-connection-demo-harsha
- Connection Type: JDBC
- Keywords: DM OH
- Estimated Cost: N/A

Jdbc Connection Metadata:

- Data Load Type: bulkdataloaderv1
- Username: admin
- Publicly Accessible: yes (with a "Test Connection" button)
- JDBC Connection URL: jdbc:mysql://database-1-instance-1.cozbikcm76r.us-east-1.rds.amazonaws.com:3306/demo

6. You have successfully created a mysql connection.



The screenshot shows the Amorphic interface for managing connections. On the left, there's a sidebar with various icons. The main area displays a list of connections: 'mysql-connection-demo', 'mysql-connection-demo-harsha' (which is selected), and 's3-connection-demo'. A success message 'Connection created' is visible at the top right. The 'mysql-connection-demo-harsha' details page is shown, featuring tabs for DETAILS, TASKS, INSTANCES, and RESOURCES. Under DETAILS, it shows 'Connection Name: mysql-connection-demo-harsha', 'Connection Type: JDBC', 'Estimated Cost: N/A', and 'Keywords: DM OH'. Under JDBC Connection Metadata, it shows 'Data Load Type: bulkdataaloay1', 'Username: admin', and 'Publicly Accessible: yes'. The JDBC Connection URL is listed as 'jdbc:mysql://database-1-instance-1.cozbikcm76r.us-east-1.rds.amazonaws.com:3306/demo'.

Tasks:

Tasks automate the data ingestion process in Amorphic.

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

This screenshot shows the same Amorphic interface as above, but the focus is on the 'TASKS' tab of the 'mysql-connection-demo-harsha' connection details page. It displays the message 'No Tasks Found!' and features three buttons: 'Reload', 'Create new Task', and 'Notification Settings'. There's also a small illustration of two people interacting with a computer screen.

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

< New Task for mysql-connection-demo-harsha

Home > Ingestion > Connections > mysql-connection-demo-harsha > New Task

Task Specs	MigrationType (Required)
Replication Configuration	Full Load
Table Selection	Target Location (Required)
Ingestion Configuration	Sync To S3 (Required)
Preview	Data Format

Task Name: task-a-harsha

Target Location: redshift

Sync To S3: Yes

Data Format: parquet

Target Extra Connection Attributes

Continue

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.

< New Task for mysql-connection-demo-harsha

Home > Ingestion > Connections > mysql-connection-demo-harsha > New Task

Task Specs	Use Serverless Replication (Required)
Replication Configuration	No
Table Selection	Use Shared Instance
Ingestion Configuration	OR
Preview	Replication Instance AZ

Allocated Storage

Instance AZ, Instance class & Allocated storage are all optional, however, if one is provided you must provide the other values as well.

Continue



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.

The screenshot shows the Amorphic interface for creating a new task. The left sidebar lists steps: Task Specs, Replication Configuration, Table Selection, Ingestion Configuration, and Preview. The 'Table Selection' step is active, showing the 'demo' schema. A list of tables is displayed with checkboxes: Courses, Graduated, PH_Vitals (checked), defendents, shots_fired, Enrollments, PH_Immunization (checked), Students, and pet. A 'Select All' button is at the top, and a 'Continue' button is at the bottom right.

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.

The screenshot shows the 'Bulk Update Amorphic Dataset Names' dialog. It has two input fields: 'Dataset Name Prefix' containing 'harsha' and 'Dataset Name Suffix' containing '_harsha'. Below the fields is a large blue button labeled '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.

< New Task for mysql-connection-demo-harsha

Home > Ingestion > Connections > mysql-connection-demo-harsha > New Task

Task Specs >

Replication Configuration >

Table Selection >

Ingestion Configuration >

Preview >

Bulk Edit ▾

Filter by Schema Name

demo

PH_Vitals >

PH_Immunization >

Basic Table Options

Dataset Name *
PH_Vitals_harsha

Dataset Description *
This dataset is loaded as part of connection task from PH_Vitals table in demo schema

Keywords *
Owner: harsha

Enable Filters, Transforms

Domain Name *
Bronze Health (bronzehealth)

Approx Table Size *
Small (< 2GB)

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)



< New Task for mysql-connection-demo-harsha

Home > Ingestion > Connections > mysql-connection-demo-harsha > New Task

Task Specs >

Replication Configuration >

Table Selection >

Ingestion Configuration > **PH_Immunication**

Preview >

Bulk Edit <

Filter by Schema Name demo PH_Immunication

Basic Table Options

Dataset Name **PH_Immunication_harsha**

Domain Name **Bronze Health (bronzehealth)**

Dataset Description This dataset is loaded as part of connection task from PH_Immunication table in demo schema

Approx Table Size Small (< 2GB)

Keywords Owner: harsha

Sort and Distribution

Sort Key Type

Distribution type

Filters

Filter By Column

Add New Filter

Transforms

Transform Column

Add New Transform Rule

Continue >

7. Preview the changes and click on Create.

< New Task for mysql-connection-demo-harsha

Home > Ingestion > Connections > mysql-connection-demo-harsha > New Task

Task Specs >

Replication Configuration >

Table Selection >

Ingestion Configuration > **task-a**

Preview >

Task Details

Task Name	task-a
Migration Type	full-load
Target Location	redshift
Task S3Payload	no
Shared Instance	no
Instance AZ	us-west-2b
Instance Class	dms.t3.medium
Dms Version	3.5.3
Allocated Storage	50
Sync To S3	yes
Data Format	parquet
Target Table Prep Mode	-
Datasets Count	1
Rules Count	3

JSON Payload

Create

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



Connections

Home > Ingestion > Connections > Details

+ New Connection

← mysql-connection-demo-harsha

Connection Id 90543a1d... | Created By harsha an hour ago | Last Modified By harsha an hour ago

Description Connection of MySQL Database | Ingestion

DETAILS TASKS INSTANCES RESOURCES

Search in this page

Showing 1 - 1 of 1 record(s)

Task Name	Migration Type	Shared Instance	Target Location	Message	Options
task-a-harsha	full-load	no	redshift	Task datasets registr...	

Showing 1 - 1 of 1 record(s)

Task datasets registration completed

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

← mysql-connection-demo-harsha

Connection Id 90543a1d... | Created By harsha an hour ago | Last Modified By harsha an hour ago

Description Connection of MySQL Database | Ingestion

DETAILS TASKS INSTANCES RESOURCES

← task-a-harsha Metadata :

Created By harsha a minute ago

Last Modified By harsha a minute ago

Migration Type full-load Status ready Target Location redshift Sync to S3 yes Data Format parquet Target Table Prep Mode truncate Allocated Storage -

Replication Instance AZ us-west-2b Instance Class - Extra Connection Attributes -

Migration Overview Table Statistics Logs Payload Runs

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

Connections

Home > Ingestion > Connections > Details

+ New Connection

← mysql-connection-demo-harsha

Connection Id 90543a1d... | Created By harsha an hour ago | Last Modified By harsha an hour ago

Description Connection of MySQL Database | Ingestion

DETAILS TASKS INSTANCES RESOURCES

Search in this page

Showing 1 - 1 of 1 record(s)

Task Name	Migration Type	Shared Instance	Target Location	Message	Options
task-a-harsha	full-load	no	redshift	Task datasets registr...	

Showing 1 - 1 of 1 record(s)

Start Task



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.)

The screenshot shows the Amorphic Parameters Store interface. The top navigation bar includes 'Home', 'Transformation', and 'Parameters Store'. A search bar and a 'Create Parameters Store' button are on the right. The main area displays a table with columns: 'Parameters Store Key', 'Scope', 'Parameters Store Type', and 'Options'. The table lists several parameters, including 'nvd-sample-parm-adityabhat' (SecureString, global), 'SYSTEM.S3BUCKET.ATHENA' (String, global), 'SYSTEM.S3BUCKET.TEMP' (String, global), 'SYSTEM.S3BUCKET.LZ' (String, global), 'SYSTEM.S3BUCKET.DLZ' (String, global), 'SYSTEM.NOTIFICATIONS.QUEUE' (String, global), 'SYSTEM.AWSREGION' (String, global), 'SYSTEM.ENVIRONMENT' (String, global), and 'SYSTEM.S3BUCKET.ETL' (String, global). Each row has an 'Edit' icon in the 'Options' column.

Parameters Store Key	Scope	Parameters Store Type	Options
nvd-sample-parm-adityabhat	global	SecureString	
SYSTEM.S3BUCKET.ATHENA	global	String	
SYSTEM.S3BUCKET.TEMP	global	String	
SYSTEM.S3BUCKET.LZ	global	String	
SYSTEM.S3BUCKET.DLZ	global	String	
SYSTEM.NOTIFICATIONS.QUEUE	global	String	
SYSTEM.AWSREGION	global	String	
SYSTEM.ENVIRONMENT	global	String	
SYSTEM.S3BUCKET.ETL	global	String	

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**.

The screenshot shows the Amorphic Parameters Store interface. On the left, there's a list of existing parameters with their keys, scopes, and descriptions. On the right, a modal window titled 'Add New Parameters Store' is open, prompting for new parameter details like Tenant, Parameters Store Key, Parameters Store Type, Scope, and Description.

Parameters Store Key	Scope
nvd-sample-parm-adityabhat	global
SYSTEM.S3BUCKET.ATHENA	global
SYSTEM.S3BUCKET.TEMP	global
SYSTEM.S3BUCKET.LZ	global
SYSTEM.S3BUCKET.DLZ	global
SYSTEM.NOTIFICATIONS.QUEUE	global
SYSTEM.AWSREGION	global

Add New Parameters Store

Tenant	Parameters Store Key *	Parameters Store Type	Scope
Default tenant	nvd-sample-parm-adityabhat	Secure String	Global

Description: A secure string from the parameter store, created for a training session to store sensitive information.

Add Parameters 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

The screenshot shows the Amorphic ETL Library interface. It displays a list of ETL packages with columns for Name, Creation Time, Last Modified Time, Access Type, Packages, Jobs Attached, and Created By. Two packages are listed: 'amorphicutils-v0-3-1-pyspark' and 'amorphicutils-v0-3-1-python'.

ETL Library Name	Creation Time	Last Modified Time	Access Type	Packages	Jobs Attached	Created By
amorphicutils-v0-3-1-pyspark	6 hours ago	6 hours ago	owner	1	1	harsha
amorphicutils-v0-3-1-python	7 hours ago	7 hours ago	owner	1	1	harsha



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

The screenshot shows the Amorphic ETL Library interface. At the top, there's a search bar and navigation links. Below that, the library details are displayed: ETL Library Id (fe4b4d47...), Created By (harsha), Last Modified By (harsha), and a description mentioning Amorphicutils v.0.3.1 for Pyspark with a link to the documentation. There are two tabs: DETAILS (selected) and RESOURCES. Under DETAILS, there's a section for ETL Library Metadata showing the ETL Library Name (amorphicutils-v0-3-1-pyspark), ETL Library Status (update_complete), and a Keywords field with a blue circular icon containing 'OH'. Under RESOURCES, there's a Packages section with a table. The table has columns for Package Name and Actions. It contains one row with the package name '/libs/python/amorphicutils-0.3.1.zip' and actions for download (down arrow) and edit (pencil). A 'Download Package' button is at the bottom of the table.

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**.

The screenshot shows the 'Create ETL Library' step. On the left, there are two buttons: 'Create ETL Library' (selected) and 'Create Package'. The main form has fields for 'ETL Library Name' (with a note icon and placeholder 'amorphicutils-v0-3-1-python-adityabhat'), 'Description' (with a note icon), and 'Keywords' (with a note icon and a dropdown showing 'Owner: harsha'). On the right, there's a 'Next' button.

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.



Create ETL Library



Create ETL Library >

Create Package >

ETL Library amorphicutils-v0-3-1-python-adityabhat1 created. Select the files to be added to the library

Select & Upload

Add package(s) to ETL Library

Create ETL Library



Create ETL Library >

Create Package >

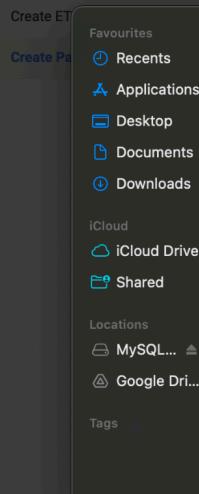
ETL Library amorphicutils-v0-3-1-python-adityabhat1 created. Select the files to be added to the library

Select & Upload

Select files to upload

Please select upto 100 files (<5GB each) to upload

Create ETL Library



Name	Size	Kind	Date Added
amorphicutils-0.3.1-py3-none-any.whl	53 KB	Document	Today at 9:04 PM
amorphicutils-0.3.1.zip	254 KB	ZIP archive	Today at 9:04 PM

Select files to upload



6. Click on Upload Selected Files.

The screenshot shows the 'Upload Data' interface. On the left, there's a sidebar with 'Create ETL Library' and 'Create Package' options. The main area displays a message: 'ETL Library amorphicutils-v0-3-1-python-adityabhat1 created. Select the files to upload'. A file named 'amorphicutils-0.3.1-py3-none-any.whl' is listed with a size of '51.38 kB'. At the bottom right is a large blue button labeled 'Upload Selected Files'.

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

The screenshot shows the 'Create ETL Library' interface. The left sidebar has 'Create ETL Library' and 'Create Package' options. The main area shows a message: 'ETL Library amorphicutils-v0-3-1-python-adityabhat1 created. Select the files to be added to the library'. Below this, the file 'amorphicutils-0.3.1-py3-none-any.whl' is listed. At the bottom right is a blue button labeled 'Add package(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)

The screenshot shows the 'ETL Library' details page. The top navigation bar includes 'Home', 'Transformation', 'ETL Library', and 'Details'. The main content area shows the ETL Library 'amorphicutils-v0-3-1-python-adityabhat' with an 'Update Complete' status. It lists several packages, including 'amorphicutils-0.3.1-pyspark', 'amorphicutils-0.3.1-pyspark-adityabhat', 'amorphicutils-0.3.1-python', and 'amorphicutils-v0-3-1-python-adityabhat'. The 'DETAILS' tab is active, showing 'ETL Library Metadata' with fields like 'ETL Library Name' (amorphicutils-v0-3-1-python-adityabhat), 'ETL Library Status' (update_complete), and 'Keywords' (OA). The 'RESOURCES' tab is also visible.



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

The screenshot shows the Amorphic platform interface. On the left, there is a navigation sidebar with the following items:

- Home
- Ingestion
- Catalog
- Transformation (highlighted in blue)
- Analytics

On the right, under the Transformation category, there is a sub-menu with the following options:

- Jobs (highlighted in blue)
- ETL Library
- Parameters Store



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

The screenshot shows the Amorphic Jobs interface. At the top, there's a search bar and a 'New Job' button. Below it is a table with columns: Job Name, Description, Last Modified, Job Type, Job Bookmark, and Options. The table contains three rows:

Job Name	Description	Last Modified	Job Type	Job Bookmark	Options
ETL_Immunization_Cleanse	ETL to Clean Immunization Data	7 days ago	pythonshell	Disabled	
sample_pyspark_job	Sample script to read and write using amorphic utils - Pyspark	5 hours ago	spark	Disabled	
sample_python_job	Sample script to read and write using amorphic utils - Pythonshell	5 hours ago	pythonshell	Disabled	

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.

The screenshot shows the details page for the sample_python_job. At the top, there's a back arrow and a 'New Job' button. Below that, there are fields for ID, Created By, and Last Modified By, all showing 'harsha' and '5 hours ago'. There's also a 'Description' field containing the script content. On the right side, there's a 'More Options' menu and a prominent 'Edit Job Script' button.

Id c60e6458... | Created By harsha 5 hours ago | Last Modified By harsha 5 hours ago

Description Sample script to read and write using amorphic utils - Pythonshell

DETAILS EXECUTIONS SCHEDULES



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

```
sample_python_job

Script Editor ⓘ

1 """
2 Description : Sample script to read and write using amorphic utils - Pythonshell
3
4 Date      : 12/Nov/2024
5 Author    : aditya.bhat@cloudwick.com
6 Version   : 1.0
7 Dependencies : amorphicutils
8 Glue Version : 3
9 Python Version : 3
10 Language : Python
11 ParamStore : SYSTEM.S3BUCKET.LZ , SYSTEM.S3BUCKET.LZ
12
13 Update History :
14 Date      Version      Author          Description
15 12/Nov/2024  1.0        aditya.bhat@cloudwick.com  First version
16
17 """
18
19
20 import time
21 import pandas as pd
22 import logging
23 from amorphicutils.common import read_param_store
24 from amorphicutils.python import write
25 from amorphicutils.python import read
26
27 logging.basicConfig(level="INFO")
28 LOGGER = logging.getLogger()
29 LOGGER.setLevel(logging.INFO)
30
31
32 def get_param_value(param_store_key, secure=False):
33     """Function to read parameters from parameters store"""
34     LOGGER.info(
35         "To get param value. Fetching values stored in %s. IsSecure : %s"
36     )
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
```

5. You need to make a few updates to the script.

- Replace the user_id with your login user id. (Line no 62 from below SS)
- Replace the param_store_key with the one you created. (Line no 81 from below SS).
- You can replace your read/write domain and read/write dataset if you wish. Let's skip this for this demo.



```
55 def main():
56     """
57     Main
58     """
59
60     LOGGER.info("In Main,")
61
62     user_id = "adityabhat" # Your Login UserId with access to write
63
64     # Replace below if you have different Domain to to Read from
65     r_domain = "bronzehealth"
66     # Replace this if you have different Domain to write into.
67     w_domain = "silverhealth"
68     # Replace this if you created new Dataset to Read from.
69     r_dataset = "Immunization"
70     # Replace this if you created new Dataset to write into.
71     w_dataset = "Immunization_Staged"
72
73     # LZ Bucket is Output Bucket
74     # lz_bucket = "nvd-us-west-2-816069158041-test-lz"
75     # DLZ Bucket is Input Bucket
76     # dlz_bucket = "nvd-us-west-2-816069158041-test-dlz"
77
78     lz_bucket = get_param_value(param_store_key="SYSTEM.S3BUCKET.LZ")
79     dlz_bucket = get_param_value(param_store_key="SYSTEM.S3BUCKET.DLZ")
80     my_param = get_param_value(
81         param_store_key="nvd-sample-param-adityabhat", secure=True
82     ) # Replace with your paramstore name
83     print(my_param)
84
```

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.

The screenshot shows the Amorphic UI interface for managing jobs. On the left, there's a sidebar with icons for Home, Transformation, and Jobs. The main area displays a list of jobs under the 'Jobs' tab. One job, 'sample_python_job', is selected and shown in detail. The 'DETAILS' tab is active, displaying basic metadata such as Job Name, Job Type, and Registration Status. To the right of the job details, a context menu is open, showing options like 'Edit Job', 'Clone Job' (which is highlighted), 'Delete Job', 'Update Extra Resources', 'Manage External Libraries', and 'Cost Tags'. The URL in the browser address bar is https://app.amorphic.com/jobs/ETL_Immunization_Cleanse/sample_pyspark_job/sample_python_job.

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.

Clone Job

Upload Data >

Metadata > * Required

Job Name *: sample_python_job_adityabhat

Description: Sample script to read and write using amorphic u

Job Type *: Python Shell

Advanced Parameters >

Network Configuration *: Public

Bookmark *: Disable

Keywords: Owner: harsha

Cost Tags >

Review & Submit >

Resource Access Configuration

Parameters Store Access *: nvd-sample-parm-adityabhat, SYSTEM.S3BUCKET.DLZ

Datasets Write Access *: silverhealth : Immunization_Staged

Datasets Read Access *: bronzehealth : Immunization

Shared Libraries *: amorphicutils-v0-3-1-python-adityabhat

Domains Write Access

Domains Read Access

Next

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



Clone Job

Upload Data >

Metadata >

Advanced Parameters >

Cost Tags >

Review & Submit >

Network Configuration	general-public-network
Job Bookmark Option	disable
Keywords	Owner: harsha
Max Concurrent Runs	4
Max Retries	-
Notify Delay After	-
Timeout	20
Is Data Lineage Enabled	no
Is Auto Scaling Enabled	false
Parameter Access	SYSTEM.S3BUCKET.DLZ, SYSTEM.S3BUCKET.LZ, nvd-sample-parm-adityabhat
Shared Libraries	amorphicutils-v0-3-1-python-adityabhat
Cost Tags	

JSON Payload

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

sample_python_job_adityabhat

Script Editor ⓘ

```
1 import sys
2 from awsglue.transforms import *
3 from awsglue.utils import getResolvedOptions
4 from pyspark.context import SparkContext
5 from awsglue.context import GlueContext
6 from awsglue.job import Job
7
8 ## @params: [JOB_NAME]
9 args = getResolvedOptions(sys.argv, ['JOB_NAME'])
10 sc = SparkContext()
11 glueContext = GlueContext(sc)
12 spark = glueContext.spark_session
13 job = Job(glueContext)
14 job.init(args['JOB_NAME'], args)
15 job.commit()
```

Read Mode On

Save & Publish



10. You have now successfully created the pythonshell Job.

Basic Metadata

Job Name	Job Type	Registration Status
sample_python_job_adityabhat	Pythonshell	completed

Keywords: OH

Advanced Parameters Metadata

Resource Access Metadata

Extra Resources

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

Run Job

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



Execute Job



Upload Data >
Execute Job >
Review & Submit >

* Required

Old Job Run Id ⓘ

Job Type ⓘ

pythonshell

Job Bookmark ⓘ
disable

Max Capacity ⓘ
1

Worker Type ⓘ

Timeout ⓘ
20

Notify Delay After ⓘ

Job Parameters ⓘ

Add Parameter

Continue

Execute Job



Upload Data >
Execute Job >
Review & Submit >

Job Details

Old Job Run Id -

Max Capacity 1

Timeout 20

Notify Delay After -

Glue Version 3.0

Job Bookmark Option disable

Max Retries -

Network Configuration general-public-network

Python Version 3.9

Is Auto Scaling Enabled false

JSON Payload



Run Job



12. You can track your execution and preview logs.

The screenshot shows the Amorphic job details page for a job named "sample_python_job_adityabhat". The "EXECUTIONS" tab is selected. It displays one execution record:

Started On	Completed On	Trigger Source	Error	Log Status	Options
⌚ a few seconds ago	-	Self	N/A	⚠ Trigger logs download to view.	

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.

The screenshot shows the Amorphic job details page for the same job. The "DETAILS" tab is selected. The "Resource Access Metadata" section is expanded, showing the following configuration:

- Domain Access:**
 - Write Access: 00
 - Read Access: 00
- Dataset Access:**
 - Write Access: 01
 - silverhealth:Immunization_Staged** (highlighted with a black box)
 - Read Access: 01
 - bronzehealth:Immunization** (highlighted with a blue box)



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.

The screenshot shows the Amorphic dataset interface for 'Immunization_Staged'. The 'FILES' tab is selected. A search bar at the top contains the text 'adityabhat'. Below it, a table displays one file entry:

File Name	Last Modified	Message	Tags	Options
...abhat_1731422104.csv	adityabhat 4 minutes ago	Data load completed.	-	Info Tags Star More

Congrats! You were successfully able to 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:

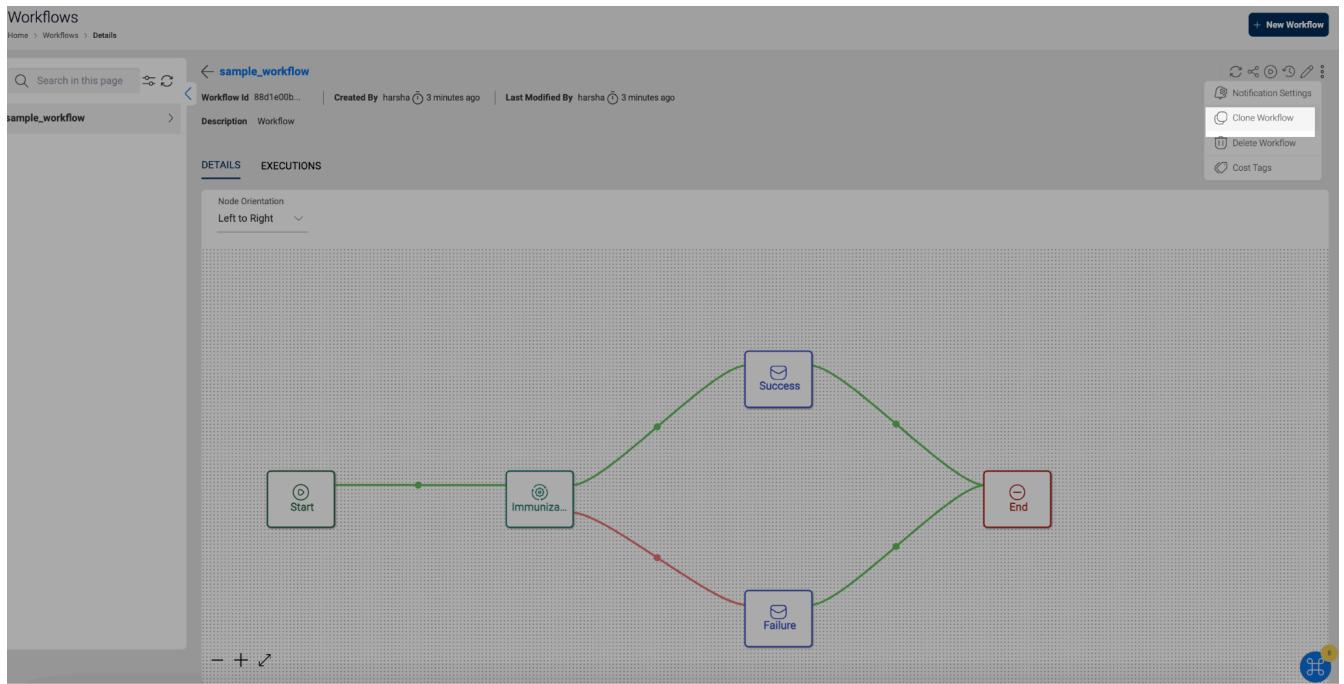
Setup a workflow to trigger an ETL, configure success and failure nodes to receive email communication

Procedure:

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

Workflow Name	Description	Last Modified Time	Options
sample_workflow	Workflow	a minute ago	

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.



< Clone Workflow

Home > Workflows > Clone Workflow

Configure Workflow Details

Workflow Name **sample_workflow_adityabhat**

Description **Workflow**

Keywords **Owner: harsha**

Default Execution Properties

Key *	Value *
s.body	Workflow Execution Successfully Completed. Please check the target datasets
sub	Health Nevada Immunization Workflow Status
to	[aditya.bhat@cloudwick.com]
f.body	Workflow Execution Failed. Please check workflow execution logs in Amorphic for n

Add Default Execution Properties

Continue

4. Skip Cost Tags and Continue.

< Clone Workflow

Home > Workflows > Clone Workflow

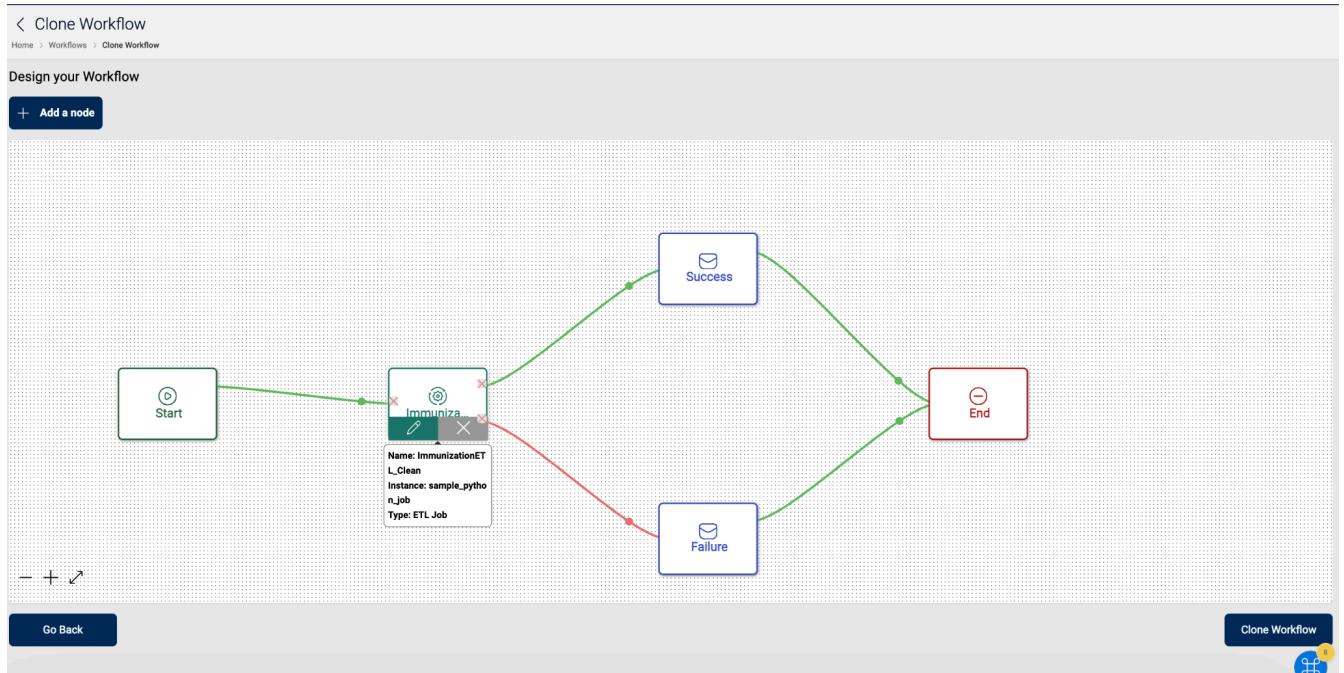
Select Cost Tags

You don't have enough permissions to use cost tags! Please contact your administrator.

Go Back

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.



- 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

Configure ETL Job

ETL Job instance ⓘ *

sample_python_job_adityabhat

Node name ⓘ *

MyETLNode

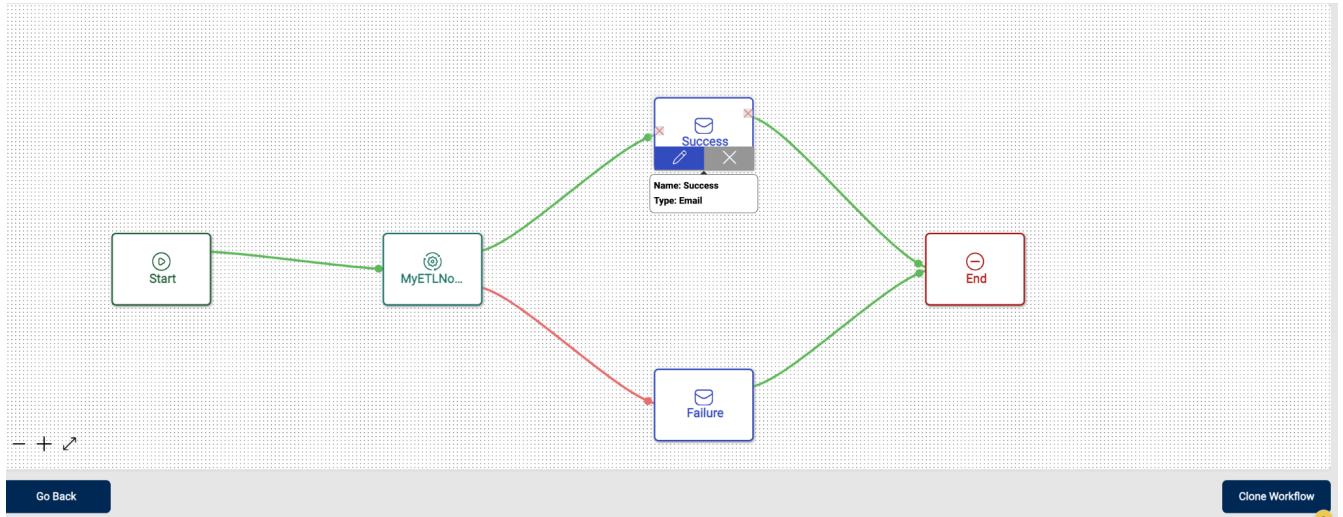
Arguments ⓘ

Add Argument

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.



Configure Email



Node name ⓘ *

Success

Email recipient ⓘ *

to



s_body

sub

to

f_body

.....

Arguments ⓘ

Add Argument

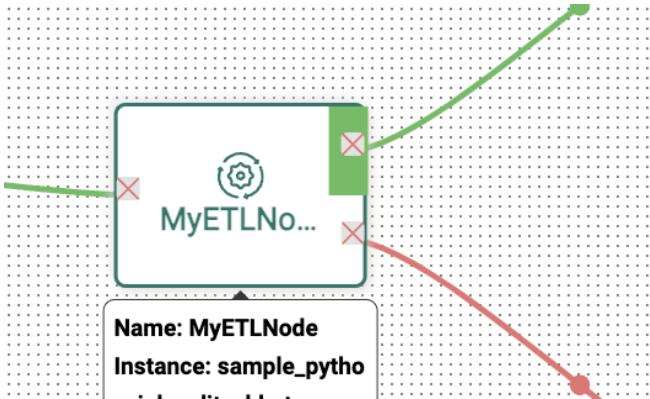
Update Node

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

A screenshot of the Amorphic workflow editor. On the left, there's a dark sidebar with icons for file operations and a "Design your Workflow" section with a "Start" node and a "MyETLNo..." node. A green arrow connects the "Start" node to the "MyETLNo..." node. On the right, a "Select Node Type" dialog is open, displaying a grid of 12 node types with icons and names:

ML Model		ETL Job		Email		Textract	
Translate	Comprehend	Medical Comprehend	Workflow	File Load Validation			
Medical Transcribe	Rekognition	Sync To S3	Connection				

At the bottom right of the dialog is a "Continue" button.

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.

The screenshot shows the 'Workflows' section with the URL 'Home > Workflows > Details'. A search bar at the top left contains 'sample_workflow_adityabhat'. On the right, there's a large 'Run Workflow' button with a blue outline and white text. Below the button are several small circular icons with symbols like a magnifying glass, a gear, and a refresh. The main content area displays the workflow details: 'Workflow Id b4117cd9...', 'Created By adityabhat a few seconds ago', 'Last Modified By adityabhat a few seconds ago', and a 'Description' field containing 'Workflow'. There are two tabs at the bottom: 'DETAILS' (selected) and 'EXECUTIONS'. Under 'DETAILS', the 'Node Orientation' is set to 'Left to Right'. The URL for this page is [/workflows/sample_workflow_adityabhat](#).

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

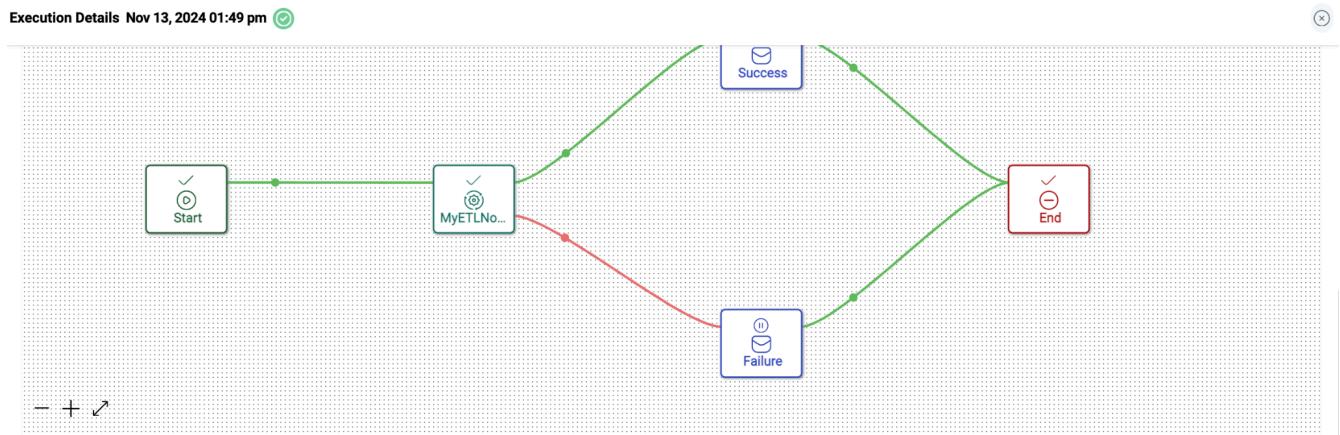
The screenshot shows the 'EXECUTIONS' tab from the previous screen. At the top, it says 'Showing 1 - 1 of 1 record(s)'. Below is a table with the following data:

Started On	Completed On	Node Count	Failed Nodes	Options
Nov 13, 2024 01:49 pm	-	5	0	Execution Details

At the bottom, it says 'Showing 1 - 1 of 1 record(s)' again. The URL for this page is [/workflows/sample_workflow_adityabhat/executions](#).



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	-	-	

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.



← sample_workflow_adityabhat

Workflow Id b4117cd9... | Created By adityabhat (a few seconds ago) | Last Modified By adityabhat (a few seconds ago)

Description Workflow

DETAILS EXECUTIONS

Search in this page

Showing 1 - 1 of 1 record(s)

Started On	Completed On	Node Count	Failed Nodes	Options
Nov 13, 2024 01:49 pm	Nov 13, 2024 01:51 pm	5	0	⋮

Showing 1 - 1 of 1 record(s)

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

Health | Nevada | Immunization | Workflow Status

Amorphic Data Services noreply@cloudwick.com via amazonses.com
to me ▾ 1:52PM (5 minutes ago) ⚡ ⌂ ⋮

Be careful with this message.
The sender hasn't authenticated this message so Gmail can't verify that it actually came from them. Avoid clicking links, downloading attachments, or replying with personal information.

Looks safe ⓘ

Health | Nevada | Immunization | Workflow Status

EventType	Generic Email
Message	Workflow Execution Successfully Completed. Please check the target datasets
AdditionalDetails	{}

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 Lifecycle Configs.

The screenshot shows a table with the following data:

Lifecycle Name	Last Modified Time	Created By	Options
auto-stop-idle-notebook	10 days ago	System	
disable-uninstall-ssm-agent	10 days ago	System	
enable-code-whisperer	10 days ago	System	
enable-glue-sessions	10 days ago	System	

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

The screenshot shows a table with the following data:

enable-glue-sessions	10 days ago	System	
Showing 1 - 4 of 4 record(s)			



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

Clone Lifecycle Configuration

Upload Data >

Metadata > * Required

Lifecycle Configuration Name ⓘ *

custom-enable-glue-sessions-harsha

Description ⓘ

This lifecycle configuration can be attached to notebooks which have glue-

Keywords

Review & Submit >

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>

Clone Lifecycle Configuration

Upload Data >

Metadata >

On-Start Script > * Required

Script will be converted to Base64

```
1 #!/bin/bash
2 set -ex
3 [ -e /home/ec2-user/glue_ready ] && exit 0
4 sudo -u ec2-user -i <<'EOF'
5
6 ANACONDA_DIR=/home/ec2-user/anaconda3
7
8 # Create and Activate Conda Env
9 echo "INFO: Creating glue_pyspark conda enviornment"
10 conda create --name glue_pyspark ipykernel jupyterlab pandas=1.5.3 -
11
12 echo "INFO: Activating glue_pyspark"
13 source activate glue_pyspark
14
15 # Initialize Latest Python Path
16 echo "INFO: Getting site packages directory of latest python version"
17 SITE_PACKAGES_DIR=$(python -c 'import site; print(site.getsitepackages()[0])')
18 echo "INFO: Site package directory is --- $SITE_PACKAGES_DIR"
19
20 # Install Glue Sessions to Env
21 echo "INFO: Installing AWS Glue Sessions with pip"
22 pip install aws-glue-sessions==1.0.0
23
```

On-Create Script >

Review & Submit >

Continue

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.

The screenshot shows the 'Notebooks Lifecycle Configurations' page. A sidebar on the left lists configurations: 'auto-stop-idle-notebook', 'custom-enable-glue-sessions-harsha' (which is selected and highlighted in blue), 'disable-uninstall-ssm-agent', 'enable-code-whisperer', and 'enable-glue-sessions'. The main content area displays the details for 'custom-enable-glue-sessions-harsha'. At the top, there's a search bar, a back arrow, and a 'Create Lifecycle Configuration' button. Below the search bar, the configuration name is 'custom-enable-glue-sessions-harsha', and it was 'Created By' harsha a few seconds ago, last modified by harsha a few seconds ago. A description states: 'This lifecycle configuration can be attached to notebooks which have glue-sessions enabled. Enabling internet access is mandatory for this lifecycle configuration to work.' There are two tabs: 'DETAILS' (which is active) and 'RESOURCES'. Under 'DETAILS', there's a 'Metadata' section with a table showing 'Lifecycle Configuration Name' as 'custom-enable-glue-s...' and 'Keywords' as '-'. There are 'On-Create Script' and 'Show On-Start Script' buttons. On the right side of the main content area, there are several small icons for editing, deleting, and sharing.

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

Code Repository:

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.



The screenshot shows the AWS SageMaker Notebooks interface. At the top, there's a search bar and a 'New Notebook' button. Below that is a table with columns: Notebook Name, Description, Instance Type, Volume Size In Gb, Sessions Enabled, and Options. One row is highlighted, showing 'sample-notebook' as the name, 'Sample Notebook' as the description, 'ml.t2.medium' as the instance type, '10' as the volume size, and 'Enabled' as sessions enabled. The 'Options' column contains icons for edit, delete, and clone, with a 'Clone Notebook' button highlighted.

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

This is a screenshot of the 'Clone Notebook' configuration page. On the left, there's a sidebar with options: Upload Data, MetaData (which is selected), Instance Metadata, Code Repository, Cost Tags, and Review & Submit. The main area has fields for 'Notebook Name' (set to 'sample-notebook-adityabhat'), 'Description' (set to 'Sample Notebook'), and 'Keywords' (set to 'Owner: adityabhat'). A note at the top right says '* Required'. A 'Continue' button is at the bottom right.

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.



Clone Notebook

Instance Type **ml.t2.medium** * * Required
Sessions **Enabled**
Parameters Store Access
SYSTEM.S3BUCKET.LZ SYSTEM.S3BUCKET.DLZ nvd-sample-parm-harsha
SYSTEM.S3BUCKET.ATHENA
Datasets Read Access
bronzehealth: Immunization
Domains Read Access
Lifecycle Configuration
enable-glue-sessions
Auto Stop **Yes**
Auto Stop Time **2024-11-14 02:00:00**

Continue

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

Repo Used: <https://github.com/cwkadityabhat/nevada-public-datalake.git>

You can click on the above link to check the public repo hosted in GitHub.

Switch to either select an existing code repository or enter a repository link

Default Code Repository –
Select an Existing Code Repository **sample-public-repo**

Additional Code Repositories –

Continue

Click on **Continue**.



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

Clone Notebook

Upload Data >	Direct Internet Access	Enabled
MetaData >	Owner	-
Instance MetaData >	Read Only	-
Code Repository >	GlueSessionsEnabled	true
Cost Tags >	Owner Dataset Access	Immunization_Staged
Review & Submit >	ReadOnly Dataset Access	Immunization
	Parameter Access	SYSTEM.S3BUCKET.LZ SYSTEM.S3BUCKET.DLZ SYSTEM.S3BUCKET.ATHENA nvd-sample-parm-adityabhat
	Shared Libraries	amorphicutils-v0-3-1-pyspark-adityabhat
	View Access	-
	Default Code Repository	-
	Additional Code Repositories	https://github.com/cwkadityabhat/nevada-public-datalake.git
	Cost Tags	
JSON Payload		
Clone Notebook		

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**



Notebooks

Home / Analytics / Notebooks / Details

+ New Notebook

sample-notebook-adityabhat

Description Sample Notebook

DETAILS SESSIONS

Metadata

Notebook Name	sample-notebook-adit...	Instance Type	ml.t2.medium	Notebook Status	InService
Volume Size(in GB)	10 GB	Estimated Cost	\$0.00	Sessions	Enabled
Keywords	QA	Go to Notebook		Go to Notebook Lab	

Linked Resources

Extra Resources

Code Repositories

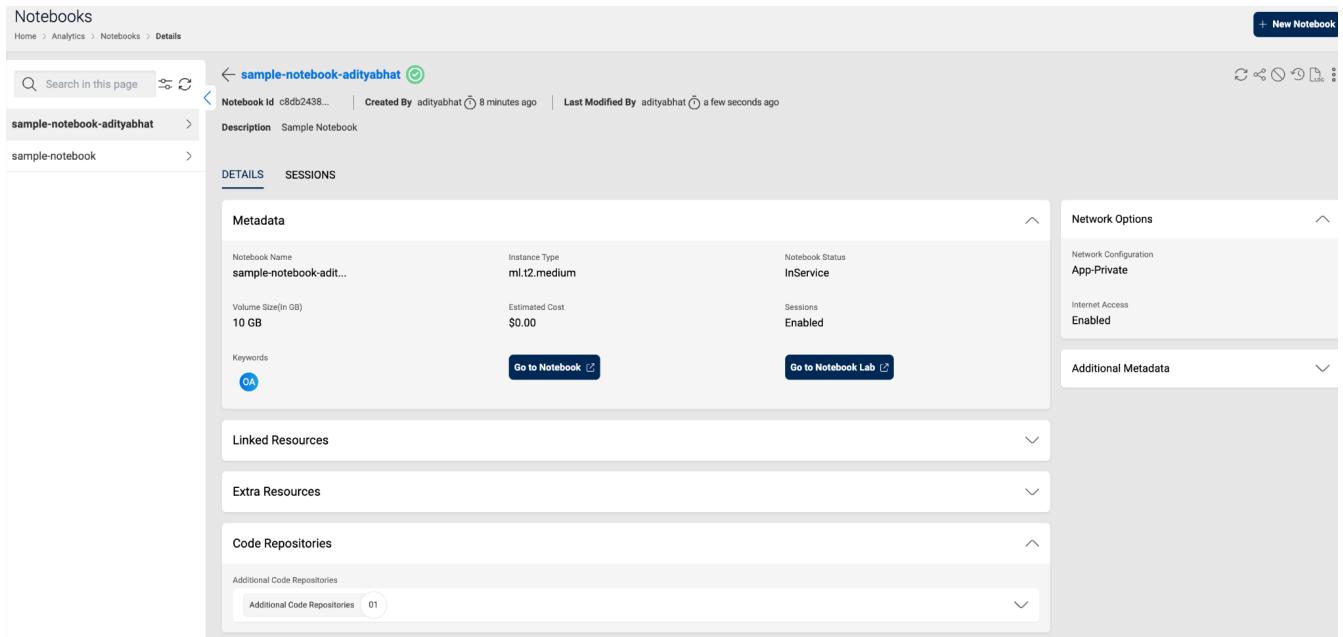
Additional Code Repositories	01
------------------------------	----

Network Options

Network Configuration App-Private

Internet Access Enabled

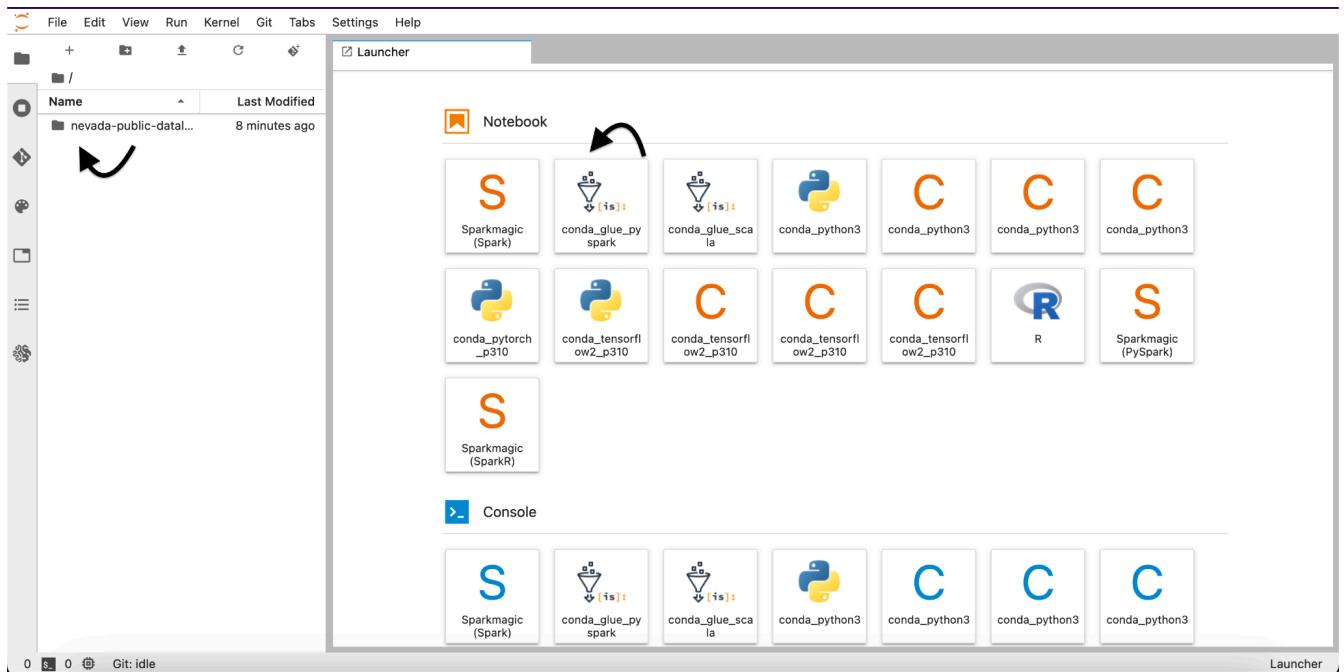
Additional Metadata



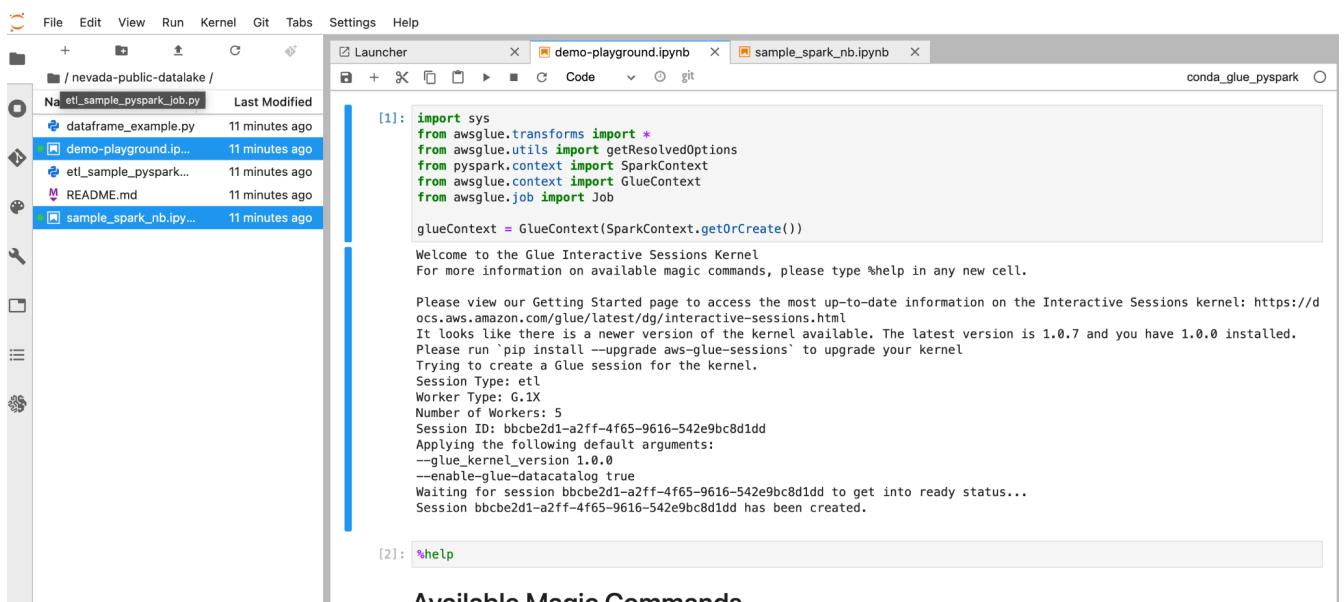
You have successfully created a Notebook.

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 that's 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**

Description :

Sample Notebook to read and write using amorphic utils -Pyspark

Load the pysaprk amorphic utils you created

copy the path from "Home -> Transformation -> ETL Library -> Details -> (hover over package path and click copy - use the version you need)"
Update this [2]: %extra_py_files s3://nvd-us-west-2-816069158041-test-etl/common-libs/069a9914-bd5c-4154-85b4-425f1f02fb38/libs/python/amorphicutils-v0-3-1-pyspark-adityabhat

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.

The screenshot shows the Amorphic ETL Library details page. On the left, there's a sidebar with a search bar and a list of ETL Library names. The main area shows the details for the 'amorphicutils-v0-3-1-pyspark-adityabhat' library. It has an 'Update Complete' status. The 'DETAILS' tab is selected, showing the ETL Library Name, Status, and Keywords (0). Under the 'Packages' section, there's a table with one row: 'Package Name' (amorphicutils-v0-3-1-pyspark-adityabhat) and 'Actions' (Copy Text). A blue box highlights the 'Copy Text' button, and a black arrow points from the text above to this button. The URL in the browser address bar is: https://app.amorphic.com/etl-libraries/eb05bb36-069a-4154-85b4-425f1f02fb38/details

4. Use Shift + Enter to run each cell.



5. Stop the notebook after your use.

Notebooks

Home > Analytics > Notebooks > Details

← sample-notebook-adityabhat ⚡

Notebook Id c8db2438... | Created By adityabhat 8 minutes ago | Last Modified By adityabhat a few seconds ago

Description Sample Notebook

sample-notebook-adityabhat >

sample-notebook >

DETAILS SESSIONS

+ New Notebook

Stop notebook

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

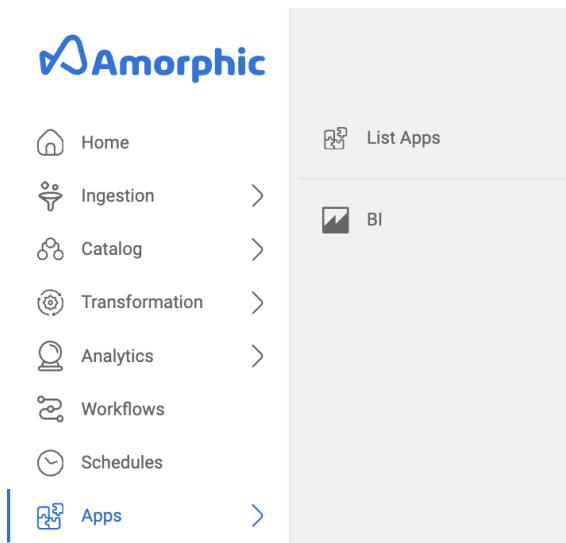


LAB Session 5:

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.

The screenshot shows the QuickSight interface. On the left, there is a sidebar with various options: Favorites, Recent, My folders, Shared folders, Dashboards (selected), Data stories, Analyses, Datasets (selected), Topics, and Community. In the main area, it says 'No datasets' and 'Import or create a new dataset to start an analysis.' There is a prominent blue 'New dataset' button in the top right corner. Two black arrows are overlaid on the image: one pointing to the 'Datasets' option in the sidebar, and another pointing to the 'New dataset' button.

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

The screenshot shows the QuickSight interface with a grid of data source icons. The grid is organized into five columns and six rows. The data sources include: RDS, Redshift (Auto-discovered), Redshift (Manual connect), MySQL, PostgreSQL; ORACLE, SQL Server, Aurora, MariaDB, Presto; Spark, Teradata (Provided by Teradata), Snowflake, Google BigQuery, AWS IoT Analytics; Amazon OpenSearch Serverless (Successor to Amazon Elasticsearch Service...), Timestream, Exasol, Databricks, Trino; Starburst, GitHub, Jira, ServiceNow, Adobe Analytics. Below this grid, there is a section titled 'FROM EXISTING DATA SOURCES' containing a single item: 'harsha data source' with a status of 'Updated 3 hours ago'.

4. Click on Create Dataset



harsha data source

x

Database name ATHENA

Create dataset

5. Choose the table to use for analysis. Lets use [goldhealth.immunization_standard](#) for this demo.

Choose your table

x

harsha data source

Catalog: contain sets of databases.

AwsDataCatalog

▼

Database: contain sets of tables.

goldhealth

▼

Tables: contain the data you can visualize.

immunization_standard

immunization_standard_harsha

vitals_standard

Edit/Preview data

Use custom SQL

Select



6. Click on Visualise

Finish dataset creation X

Table: immunization_standard
Data source: harsha data source
Schema: goldhealth

Import to SPICE for quicker analytics ✓ 30GB available SPICE

Directly query your data

Email owners when a refresh fails

Edit/Preview data Augment with SageMaker Visualize

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

New sheet X

Interactive sheet
Single page, interactive content

Layout Tiled

Optimize for viewing on 1600px

Pixel-Perfect report New
Multi-page, highly formatted document

Paper size US letter - 8.5 x 11 in

Portrait Landscape

[Learn more about Pixel-Perfect Reports](#)

CANCEL CREATE

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.



QuickSight | immunization_standard analysis

File Edit Data Insert Sheets Objects Search

Dataset: SPICE immunization_stan... 100%

Visuals

Sheet 1 +

CHANGE VISUAL TYPE

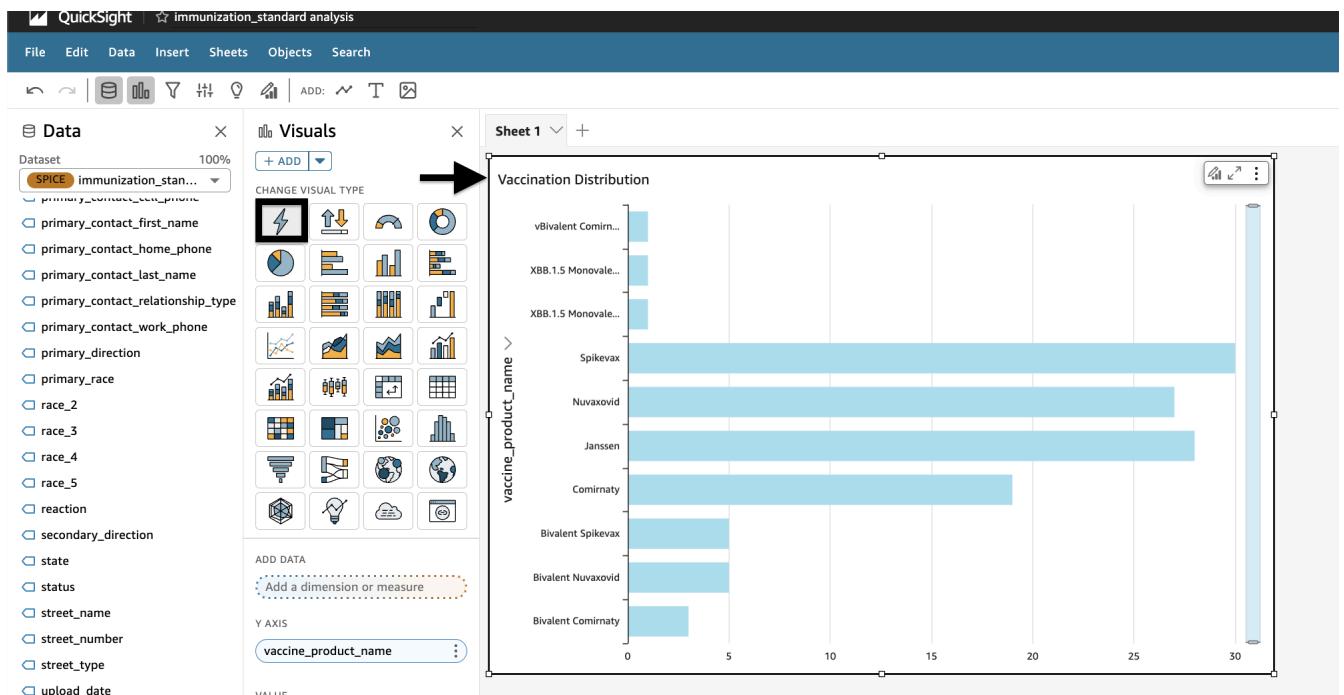
ADD DATA

Add a dimension or measure

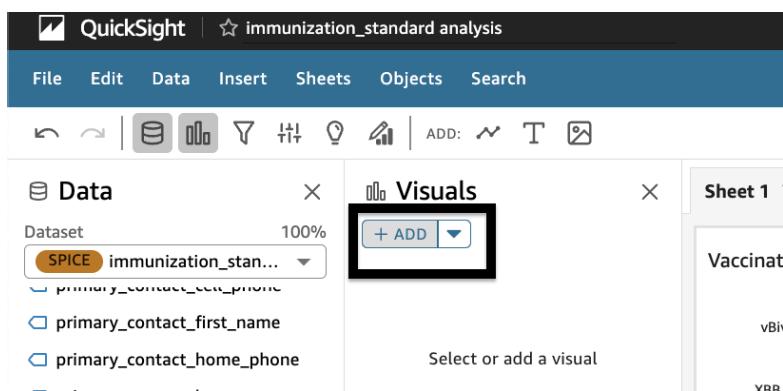
AutoGraph
Add 1 or more fields to build a visual.

The screenshot shows the QuickSight interface with the 'Data' and 'Visuals' tabs selected. The 'Data' tab displays a list of fields from a dataset named 'SPICE immunization_stan...'. The 'Visuals' tab shows a grid of visualization icons. A large text area titled 'AutoGraph' with the sub-instruction 'Add 1 or more fields to build a visual.' is present. A black arrow points from the bottom left towards the 'vaccine_product_name' field in the Data list.

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.

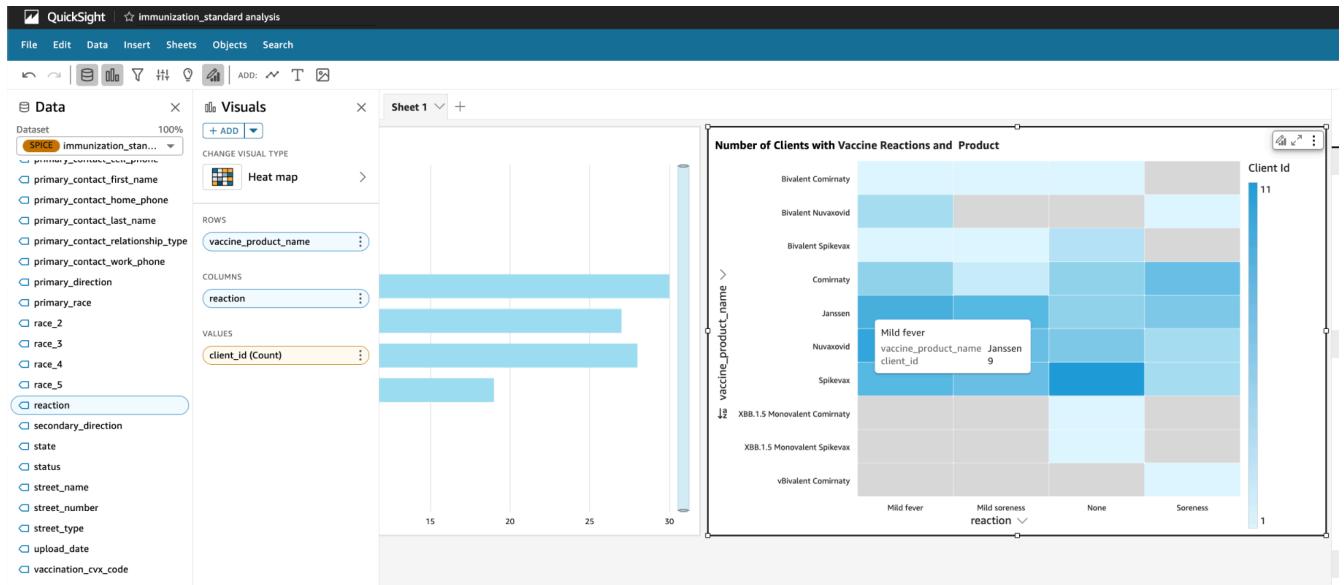


Click on Add Visuals to add another graph



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.



Publish a dashboard

Publish new dashboard as

Vaccination Analysis

Replace an existing dashboard

ALL SHEETS SELECTED ▾

Data story

Allow sharing data stories ⓘ

Generative capabilities

Allow executive summary ⓘ

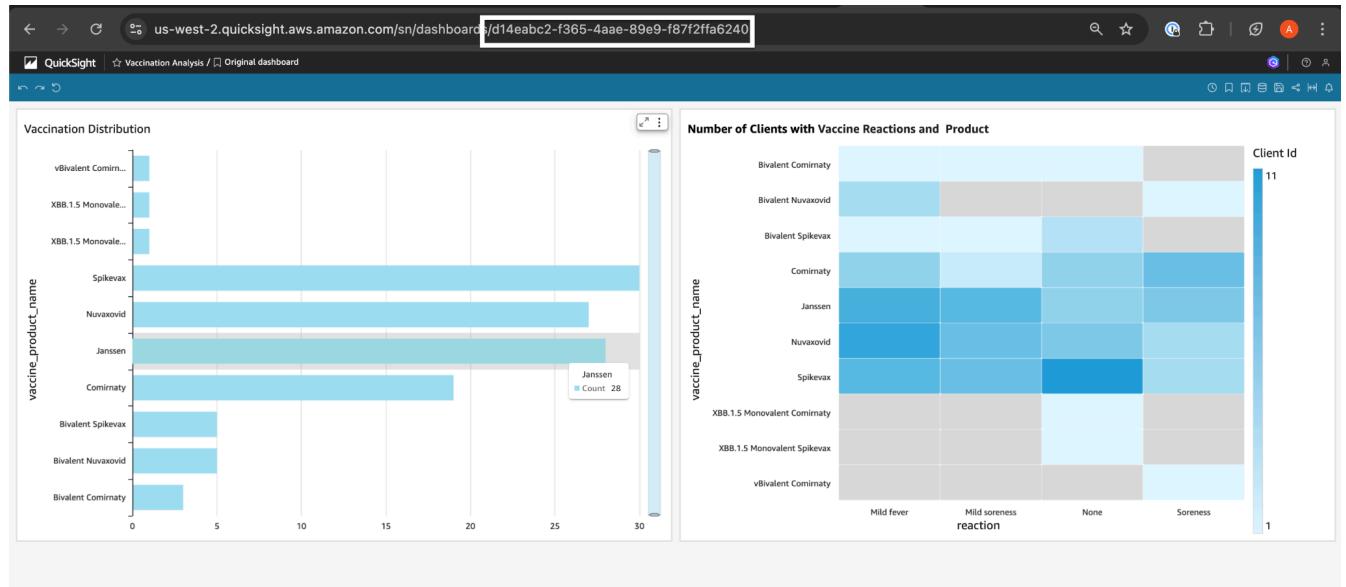
! No topic linked yet. [Link topic](#)

Advanced publish options ▾

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.

Showing 1 - 1 of 1 record(s)

App Name	App Type	Last Modified Time	Options
BI	BI	7 hours ago	(dropdown menu)

Showing 1 - 1 of 1 record(s)

2. Click on Sync Dashboards to sync the dashboard to Amorphic.

You will see a notification that says "Successfully triggered dashboard sync process"

Showing 1 - 1 of 1 record(s)

← nvd-bi

App Id c820db29... | Created By harsha 7 hours ago | Last Modified By System 7 hours ago

Description BI Application

DETAILS AUTHORIZED USERS

App Metadata

App Name NVD-BI App Type BI App Url Go To BI App

Update IAM Role Sync Dashboards



Amorphic Dashboard:

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

The screenshot shows a table with columns: Dashboard Name, Description, Dashboard Type, Last Modified Time, and Options. The first row contains the following data:

Dashboard Name	Description	Dashboard Type	Last Modified Time	Options
VaccineAnalyticsDashboard Vaccine Analytics	Vaccine Analytics	Quicksight	a few seconds ago	

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

The screenshot shows the 'Clone Dashboard' wizard. Step 1: Metadata. The form fields are:

- Dashboard Name: VaccineAnalyticsDashboard_harsha
- Description: Vaccine Analytics
- Dashboard Type: Quicksight
- Quicksight Dashboard Id: d14eabc2-f365-4aae-89e9-f87f2ffa6240
- Keywords: Owner: harsha

At the bottom right is a 'Next' button.

3. Finally click on Create to create the dashboard.

The screenshot shows the 'Clone Dashboard' wizard. Step 2: Connection Details. The form fields are identical to the previous step:

- Dashboard Name: VaccineAnalyticsDashboard_harsha
- Dashboard Type: Quicksight
- Quicksight Dashboard Id: d14eabc2-f365-4aae-89e9-f87f2ffa6240
- Description: Vaccine Analytics

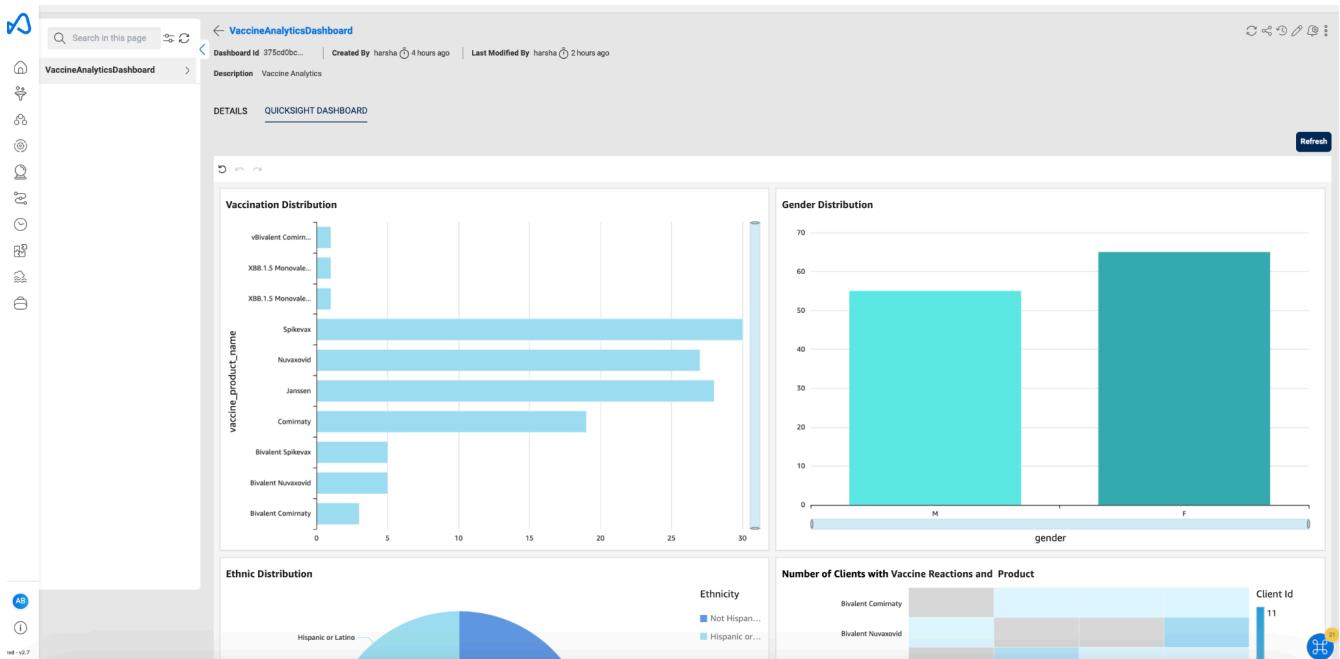
Below the main form is a 'JSON Payload' section with a download icon. At the bottom right is a 'Create' button.



4. Dashboard will be created

The screenshot shows the Amorphic interface for creating a new dashboard. At the top right, there is a green checkmark icon and the message "Dashboard registration completed successful". The dashboard title is "VaccineAnalyticsDashboard". Below the title, it shows the dashboard ID, creation date, and last modified date. The description is "Vaccine Analytics". Under the "DETAILS" tab, the "QUICKSIGHT DASHBOARD" section is selected. It displays "Dashboard Metadata" with fields: Dashboard Name (VaccineAnalyticsDashboard), Dashboard Type (Quicksight), and Keywords (OH). There is also a "QuickSight Dashboard Id" field.

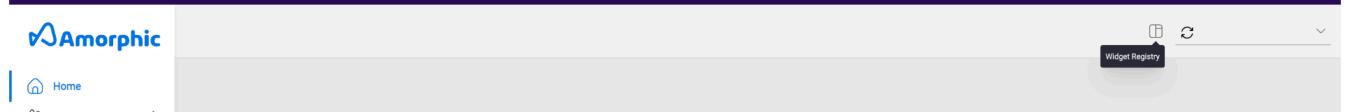
5. Click on Quicksight Dashboard to view the Dashboard.



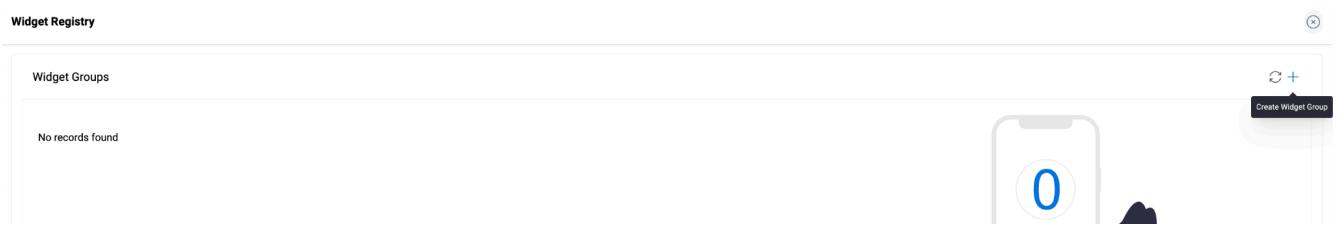


Adding Dashboard as widget:

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.

Create Widget Group (X)

Click here - OR - drag and drop file here
Accepted file type - .json

Widget Group Name *

Description

Make this Group as Default

Create



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

Widget Registry

Widget Groups MyWidget

MyWidget Filter by title or description

Available Widgets

Name	Description	Options
Access Requests	List of Access requests for datasets access, both sent and received	
Dataset Lineage	To check the dataset journey	
FlexiQuery	Quick query to run	
Insights Details	Watch list of the selected insight	
Latest Connection Tasks Status	Displays the latest tasks and their statuses for given connection	
Latest schedule executions Status	Displays the latest executions and their statuses for given schedule	
My Modules	List of all services/resources user has access to	
Quick Sight Dashboard	Rended Quicksight dashboard from list of user dashboards	

Add this Widget

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

Edit Widget Configuration

Click here - OR - drag and drop file here
Accepted file type - .json

Title Quick Sight Dashboard

Description Rended Quicksight dashboard from list of user dashboards

Dashboard Id *
 VaccineAnalyticsDashboard

VaccineAnalyticsDashboard

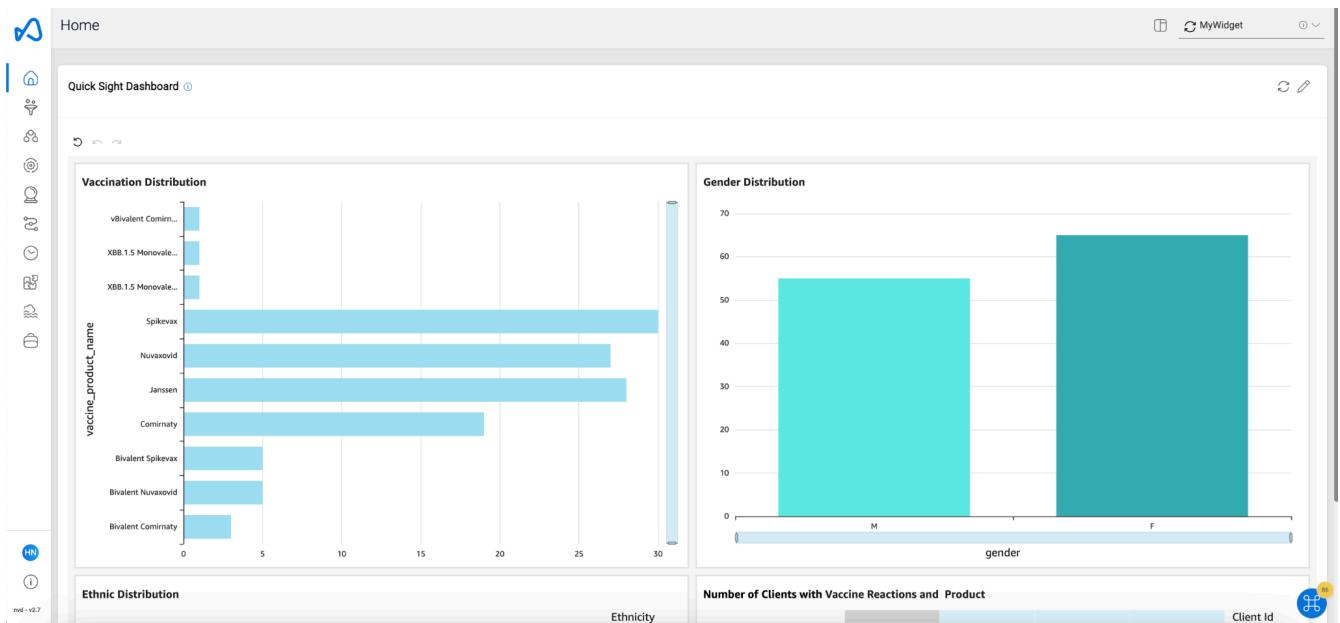
VaccineAnalyticsDashboard_harsha

Full Page

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

The screenshot shows the 'Studios' page in the Amazon SageMaker console. The top navigation bar includes the Amorphic logo, a search bar, and a 'Create Studio' button. The main content area displays a table with a single row of data. The columns are 'Studio Name', 'Last Modified Time', 'Created By', and 'Options'. The 'Studio Name' column contains 'sample-studio' with a green circular icon. The 'Last Modified Time' column shows 'an hour ago'. The 'Created By' column shows 'harsha'. The 'Options' column contains three icons: a copy icon, a delete icon, and a 'Clone Studio' button, which is highlighted with a dark blue arrow pointing to it.

Studio Name	Last Modified Time	Created By	Options
sample-studio Sample Studio	an hour ago	harsha	(highlighted)

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**.

Clone Studio

Upload Data

Metadata

Instance Metadata

Cost Tags

Review & Submit

Allowed Instances List * ml.t3.medium USD 0.0500/hour ml.t3.large USD 0.1000/hour

Direct Internet Access * Yes

Volume Size (GB) * 50

Max Volume Size (GB) 100

RStudio Access * Disabled

Parameter Access SYSTEM.S3BUCKET.LZ SYSTEM.S3BUCKET.DLZ

Dataset Write Access

Shared Libraries

Jupyter Lab Instance Type ml.t3.medium USD 0.0500/hour

Continue

* Required

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

Clone Studio

Upload Data

Metadata

Instance Metadata

Cost Tags

Review & Submit

Description	Sample Studio
Keywords	Owner: adityabhat
Direct Internet Access	Enabled
Volume Size In GB	50
Max Volume Size In GB	100
RStudio Access Status	Disabled
Jupyter Lab Instance Type	ml.t3.medium
Owner Dataset Access	-
ReadOnly Dataset Access	Immunization
Parameter Access	SYSTEM.S3BUCKET.LZ SYSTEM.S3BUCKET.DLZ
Shared Libraries	-
Instance List	ml.t3.medium ml.t3.large
Cost Tags	-

JSON Payload

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.

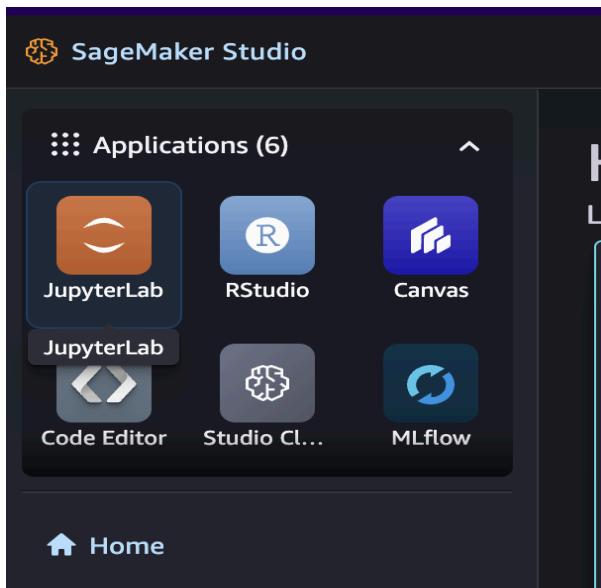
The screenshot shows the AWS SageMaker Studio Details page. On the right side, there is a panel titled "User Profile Details" which contains the "User Profile URL" field, which is highlighted with a red box. The URL value is "user-adityabhat".

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

The screenshot shows the SageMaker Studio welcome tour screen. It features a dark background with a purple brain icon at the top. Below it, the text "Welcome to the new SageMaker Studio" is displayed, followed by "We've built a new experience to empower you and your work." At the bottom, there are two buttons: "Skip Tour for now" and "Take a quick tour" (which has a play icon). A message at the bottom asks if the user is an existing Studio Classic user and provides a link to learn how to migrate.



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



10. Open the JupyterLab.

A screenshot of the JupyterLab instance details page. The top navigation bar shows "SageMaker Studio > Jupyterlab". The main content area has a title "JupyterLab" with a "Create JupyterLab space" button. Below it is an "About" section with a description of JupyterLab as a web-based IDE. There are "See features" and "Quick start guide" links, a search bar, and a filter for "Running" spaces. A table lists the running space: Name (Jupyter-Lab-Space), Application (JupyterLab), Status (Running with green checkmark), Type (Shared), Last modified (5 minutes ago), Action (with "Stop" and "Open" buttons). The "Open" button is highlighted with a blue box. The sidebar on the left includes sections for Home, Running instances, Data, Auto ML, Experiments, and Jobs.

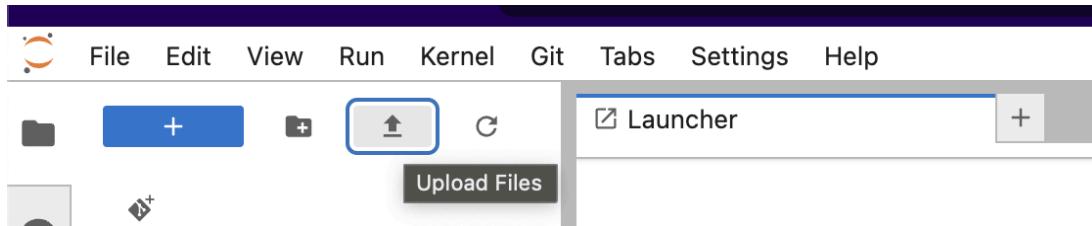
11. For this lab we will only be reading from the Dataset. Download the script from the URL below.

<https://github.com/cwkadityabhat/nevada-public-datalake/blob/main/Lab6/sample-read-nb.ipynb>



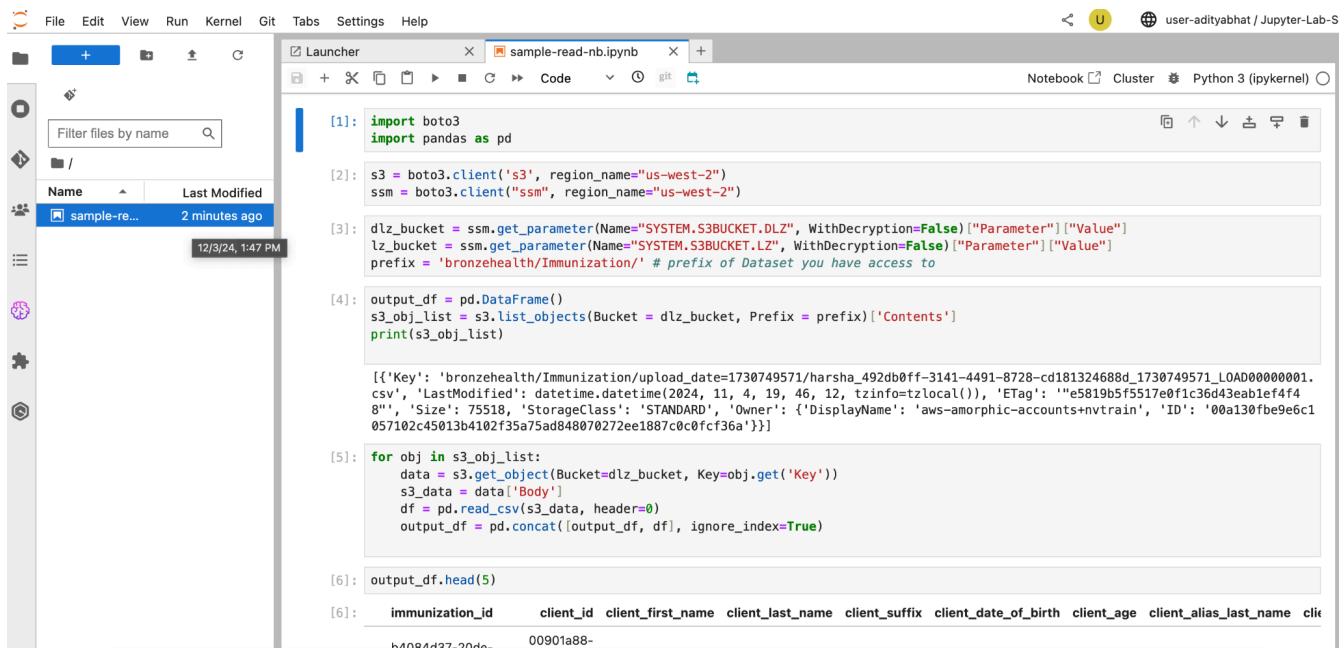
The screenshot shows a GitHub Copilot interface for a notebook titled 'sample-read-nb.ipynb'. At the top, it displays the repository path 'nevada-public-datalake / Lab6 / sample-read-nb.ipynb'. Below that, there's a user profile icon for 'cwkadityabhat' with the text 'Add files via upload'. To the right, it shows a commit hash '85c7dd4 · 42 minutes ago' and a 'History' link. The main area shows the notebook content with tabs for 'Preview', 'Code', and 'Blame'. It indicates '340 lines (340 loc) · 12.6 KB'. A note says 'Code 55% faster with GitHub Copilot'. On the right, there are download and raw file links, along with edit and settings icons.

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.

The screenshot shows the AWS SageMaker Studio interface. On the left, there's a sidebar with icons for Home, Analytics, Studios, and Details. The main area displays a list of studios: 'sample-studio' and 'sample-studio-adityabhat'. The 'sample-studio-adityabhat' item is selected and expanded, showing its details. At the top of this expanded view, it says 'sample-studio-adityabhat'. Below that are fields for 'Studio Id' (54388372...), 'Created By' (adityabhat), 'Last Modified By' (adityabhat), and a 'Description' field containing 'Sample Studio'. At the bottom of this section are two tabs: 'DETAILS' (which is active) and 'STUDIO APPS'. To the right of the main content area, there's a context menu with several options: 'Update Resources', 'Update Extra Resources', 'Delete Studio', and 'Cost Tags'. The 'Delete Studio' option is highlighted with a red box.

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

References:

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>