

# Project Deliverable 2

Team - 4

Abhinav Botla

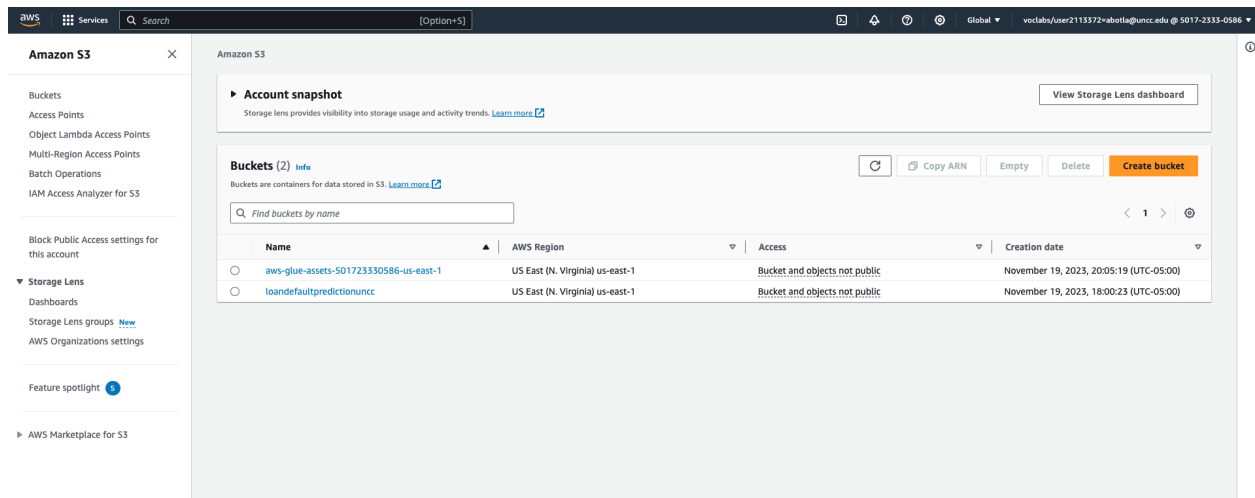
Ajith Gannamaneni

Hrushikesh Dandge

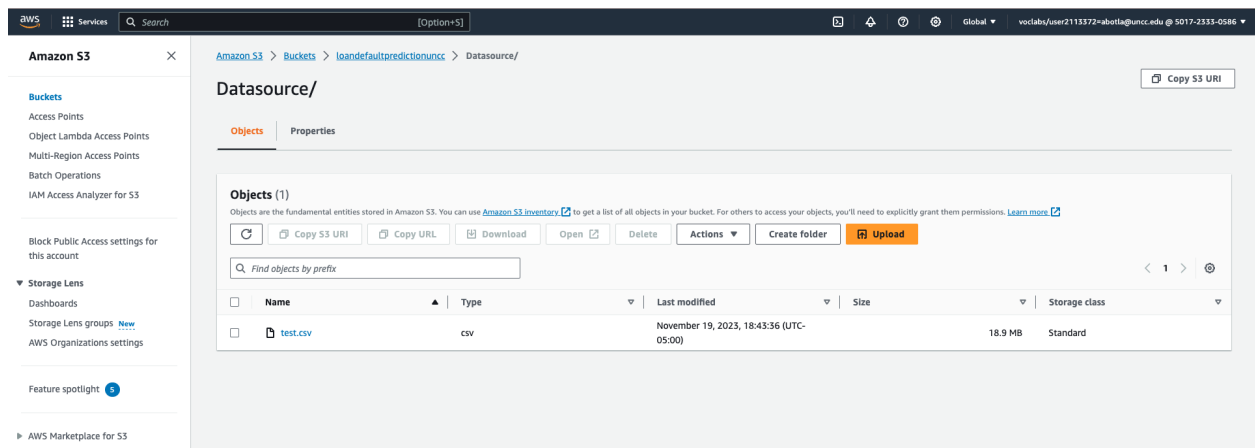
Mitra Buggaveeti

Prateek Chanddra

Created the S3 bucket with name of **loandefaultpredictionuncc** to store the project data in the bucket



Created the Datasource folder in the bucket to store the data



Analysis of data by using the AWS Athena:

Creating table DDL in Athena with predefined attributes already available in the data source.

The screenshot shows the AWS Athena Query Editor interface. On the left, the 'Data' panel is configured with 'Data source' as 'AwsDataCatalog', 'Database' as 'loan\_data', and 'Tables and views' showing 'test'. The main editor area contains a SQL query to create an external table:

```
1 CREATE EXTERNAL TABLE IF NOT EXISTS loan_data.test
2 - (
3   UniqueID INT,
4   disbursed_amount INT,
5   asset_cost INT,
6   ltv INT,
7   branch_id INT,
8   supplier_id INT,
9   manufacturer_id INT,
10  Current_pincode_ID INT,
11  Date_of_Birth DATE,
12  Employment_Type STRING,
13  DisbursalDate DATE,
14  State_ID INT,
15  Employee_code_ID INT,
```

Below the query, there are buttons for 'Run again', 'Explain', 'Cancel', 'Clear', and 'Create'. The 'Query results' tab is active, showing 'Query stats'.

Time based exploratory data analysis:

The screenshot shows the AWS Athena Query Editor interface with a different query. The 'Data' panel on the left is the same. The main editor area contains a SQL query for time-based analysis:

```
1 SELECT
2   EXTRACT(YEAR FROM DisbursalDate) AS year,
3   EXTRACT(MONTH FROM DisbursalDate) AS month,
4   COUNT(*) AS loan_count
5 FROM loan_data.test
6 GROUP BY EXTRACT(YEAR FROM DisbursalDate), EXTRACT(MONTH FROM DisbursalDate)
7 ORDER BY year, month;
```

Below the query, there are buttons for 'Run again', 'Explain', 'Cancel', 'Clear', and 'Create'. The 'Query results' tab is active, showing 'Query stats' and 'Results (27)'. The results are displayed in a table:

#	year	month	loan_count
1	3	11	2453
2	5	11	1288
3	6	11	1033
4	7	11	189
5	8	11	1611

## Analyzing the credit history of the customers:

The screenshot shows the AWS Athena console interface. At the top, a notification banner states: "Athena now supports typeahead code suggestions to speed up SQL query development. Typeahead suggestions are turned on by default. You can change this setting in query editor preferences." Below this, the left sidebar contains the "Data" section with "Data source" set to "AwsDataCatalog" and "Database" set to "loan\_data". The "Tables and views" section shows a table named "test". The main query editor displays the following SQL query:

```
1 SELECT
2   CREDIT_HISTORY_LENGTH,
3   COUNT(*) AS count
4 FROM loan_data.test
5 GROUP BY CREDIT_HISTORY_LENGTH
6 ORDER BY CREDIT_HISTORY_LENGTH;
7
```

Below the query editor, the "Query results" tab is active, showing a "Completed" status. The query statistics are: Time in queue: 117 ms, Run time: 959 ms, Data scanned: 18.95 MB. The results are displayed in a table with 252 rows:

#	CREDIT_HISTORY_LENGTH	count
1	Oyrs 0mon	58051
2	Oyrs 10mon	1163

## Analyzing the count of customers with specific CNS Score:

The screenshot shows the AWS Athena console interface. The left sidebar is the same as in the previous screenshot. The main query editor displays the following SQL query:

```
1 SELECT
2   PERFORM_CNS_SCORE,
3   COUNT(*) AS count
4 FROM loan_data.test
5 GROUP BY PERFORM_CNS_SCORE
6 ORDER BY PERFORM_CNS_SCORE;
7
```

Below the query editor, the "Query results" tab is active, showing a "Completed" status. The query statistics are: Time in queue: 102 ms, Run time: 646 ms, Data scanned: 18.95 MB. The results are displayed in a table with 566 rows:

#	PERFORM_CNS_SCORE	count
1	0	57893
2	14	470

**Athena now supports typeahead code suggestions to speed up SQL query development**  
Typeahead suggestions are turned on by default. You can change this setting in query editor preferences. [Edit preferences](#)

**Data**

Data source: **AwsDataCatalog**

Database: **loan\_data**

Tables and views: [Create](#)

**Tables (1)**

**Views (0)**

**Query 4**

```
1 SELECT
2   AVG(disbursed_amount) AS avg_disbursed_amount,
3   MIN(disbursed_amount) AS min_disbursed_amount,
4   MAX(disbursed_amount) AS max_disbursed_amount
5 FROM loan_data.test;
```

SQL Ln 1, Col 1

[Run again](#) [Explain](#) [Cancel](#) [Clear](#) [Create](#)

[Reuse query results](#)  
up to 60 minutes ago

**Query results** **Query stats**

**Completed** Time in queue: 97 ms Run time: 685 ms Data scanned: 18.95 MB

**Results (1)** [Copy](#) [Download results](#)

Search rows

#	avg_disbursed_amount	min_disbursed_amount	max_disbursed_amount
1	56076.80326891594	11613	940690

## AWS GLUE: CRAWLERS:

**AWS Glue**

Getting started  
ETL Jobs  
Visual ETL  
Notebooks  
Job run monitoring  
Data Catalog tables  
Data connections  
Workflows (orchestration)

**Data Catalog**  
Databases  
Tables  
Stream schema registries  
Schemas  
Connections  
**Crawlers**  
Classifiers  
Catalog settings

**Data integration and ETL**  
ETL Jobs  
Visual ETL  
Notebooks  
Job run monitoring  
Interactive Sessions  
Data classification tools  
Sensitive data detection  
Record Matching  
Triggers  
Workflows (orchestration)  
Blueprints  
Security configurations  
Legacy pages

**AWS Glue > Crawlers**

**Crawlers**  
A crawler connects to a data store, progresses through a prioritized list of classifiers to determine the schema for your data, and then creates metadata tables in your data catalog.

**Crawlers (1)** [info](#) Last updated (UTC) November 20, 2023 at 01:41:21 [Action](#) [Run](#) [Create crawler](#)

View and manage all available crawlers.

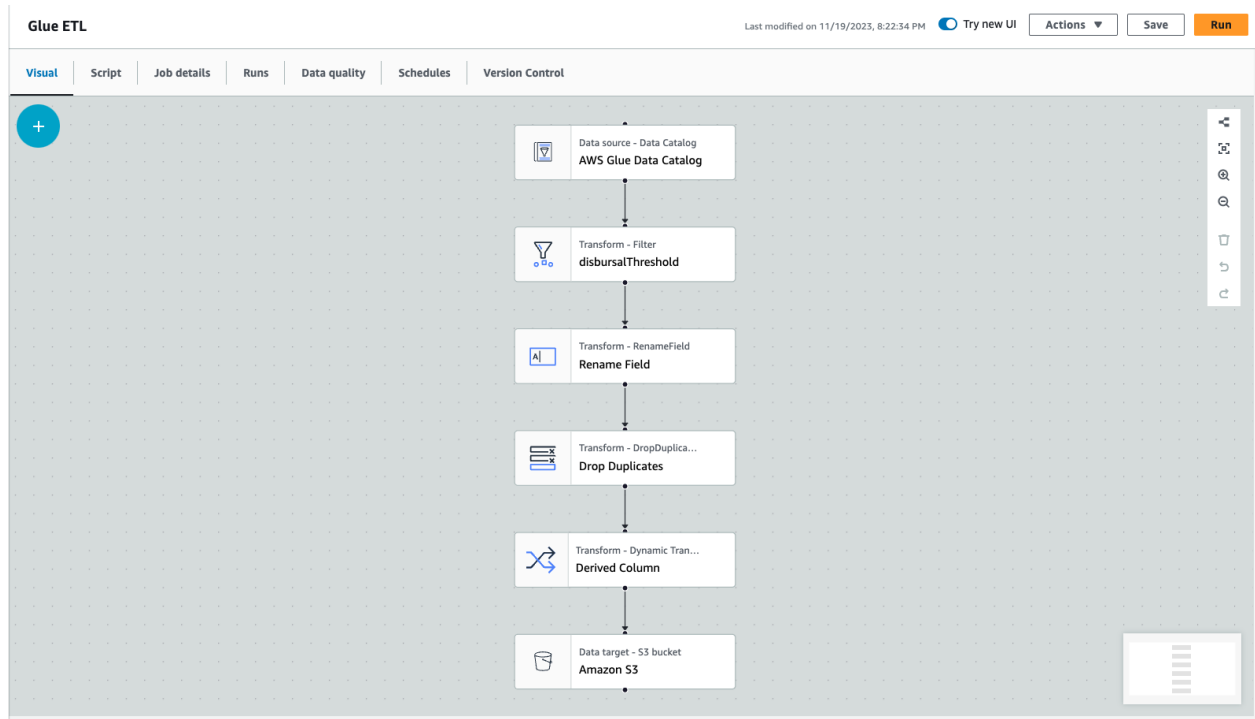
Filter crawlers

<input type="checkbox"/>	Name	State	Schedule	Last run	Last run timestamp	Log	Table changes from last r...
<input type="checkbox"/>	gluecrawler	Ready		Succeeded	November 20, 2023 at 24:...	<a href="#">View log</a>	1 updated

© 2023, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)



## Visualization of the data flow



We doesnt pick the records which have a disbursed amount below 15000.

The diagram shows the same Glue ETL workflow, but with the 'Transform - Filter disbursalThreshold' node highlighted. The 'Transform' panel on the right provides details for this node:

- Name:** disbursalThreshold
- Node parents:** Choose one or more parent node. AWS Glue Data Catalog X Catalog - DataSource
- Filter:** Builds a new output by selecting records from the input data that satisfy a specified predicate function.
  - ☒ **Global AND**: All filter conditions will be applied as a global "AND."
  - ☐ **Global OR**: All filter conditions will be applied as a global "OR."
- Filter condition:** Specify your filter condition by choosing the key, operator, and entering a value.
  - Key:** disbursed\_amount
  - Operation:** <=
  - Value:** 15000
- Buttons:** Add condition, Remove

The 'Data preview' panel at the bottom left shows 'Filter sample dataset' and 'Loading data previews. (0% complete)'.

## Dropping the duplicates from the data

The screenshot shows the Glue ETL console interface. The workflow consists of three nodes: 'Transform - Drop Duplicates', 'Transform - Dynamic Transform - Derived Column', and 'Data target - S3 bucket Amazon S3'. The 'Drop Duplicates' node is selected, and its configuration panel is open on the right. The configuration panel shows the following details:

- Name:** Derived Column
- Node parents:** Choose one or more parent node (Dropdown menu)
- Drop Duplicates:** Drop Duplicates - Transform
- Name of the derived column:** Available Amount
- SQL Expression:** `'pri.sanctioned.amount' - 'pri.disbursed.amount'`

The 'Data preview' section at the bottom shows 'Filter sample dataset' and 'End session' buttons. The status is 'READY'.

Created the new column in the data set with attribute name of Available amount , which gives the information about (sanctioned amount - disbursed amount)

The screenshot shows the Glue ETL console interface. The workflow consists of two nodes: 'Transform - Dynamic Transform - Derived Column' and 'Data target - S3 bucket Amazon S3'. The 'Derived Column' node is selected, and its configuration panel is open on the right. The configuration panel shows the following details:

- Name:** Amazon S3
- Node parents:** Choose one or more parent node (Dropdown menu)
- Derived Column:** DynamicTransform - Transform
- Format:** CSV
- Compression Type:** None
- S3 Target Location:** Choose an S3 location in the format `s3://bucket/prefix/object/` with a trailing slash (/).  
Search: `s3://loandefaultpredictio` (with a close button) | View | Browse S3
- Data Catalog update options:**
  - ☒ Do not update the Data Catalog
  - ☐ Create a table in the Data Catalog and on subsequent runs, update the schema and add new partitions
  - ☐ Create a table in the Data Catalog and on subsequent runs, keep existing schema and add new partitions
- Partition keys - optional:** Add partition keys.  
Add a partition key

A message at the bottom left states: 'Target node not supported. You have selected a data target node which is not supported for data preview. Please select another type of node instead.'

Final destination:

Amazon S3

Buckets

Access Points

Object Lambda Access Points

Multi-Region Access Points

Batch Operations

IAM Access Analyzer for S3

Block Public Access settings for this account

Storage Lens

Dashboards

Storage Lens groups

AWS Organizations settings

Feature spotlight

AWS Marketplace for S3

Amazon S3

Buckets

loandefaultpredictionuncc

Info

Objects

Properties

Permissions

Metrics

Management

Access Points

Objects (38)

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'll need to explicitly grant them permissions. [Learn more](#)

Refresh

Copy S3 URI

Copy URL

Download

Open

Delete

Actions

Create folder

Upload

Find objects by prefix

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	Datasource/	Folder	-	-	-
<input type="checkbox"/>	run-1700443418322-part-r-00000	-	November 19, 2023, 20:24:05 (UTC-05:00)	867.0 B	Standard
<input type="checkbox"/>	run-1700443418322-part-r-00001	-	November 19, 2023, 20:24:10 (UTC-05:00)	700.0 B	Standard
<input type="checkbox"/>	run-1700443418322-part-r-00002	-	November 19, 2023, 20:24:05 (UTC-05:00)	893.0 B	Standard
<input type="checkbox"/>	run-1700443418322-part-r-00003	-	November 19, 2023, 20:24:09 (UTC-05:00)	700.0 B	Standard
<input type="checkbox"/>	run-1700443418322-part-r-00004	-	November 19, 2023, 20:24:09 (UTC-05:00)	700.0 B	Standard
<input type="checkbox"/>	run-1700443418322-part-r-00005	-	November 19, 2023, 20:24:05 (UTC-05:00)	898.0 B	Standard
<input type="checkbox"/>	run-1700443418322-part-r-00006	-	November 19, 2023, 20:24:09 (UTC-05:00)	700.0 B	Standard
<input type="checkbox"/>	run-1700443418322-part-r-00007	-	November 19, 2023, 20:24:09 (UTC-05:00)	700.0 B	Standard
<input type="checkbox"/>	run-1700443418322-part-r-00008	-	November 19, 2023, 20:24:09 (UTC-05:00)	700.0 B	Standard

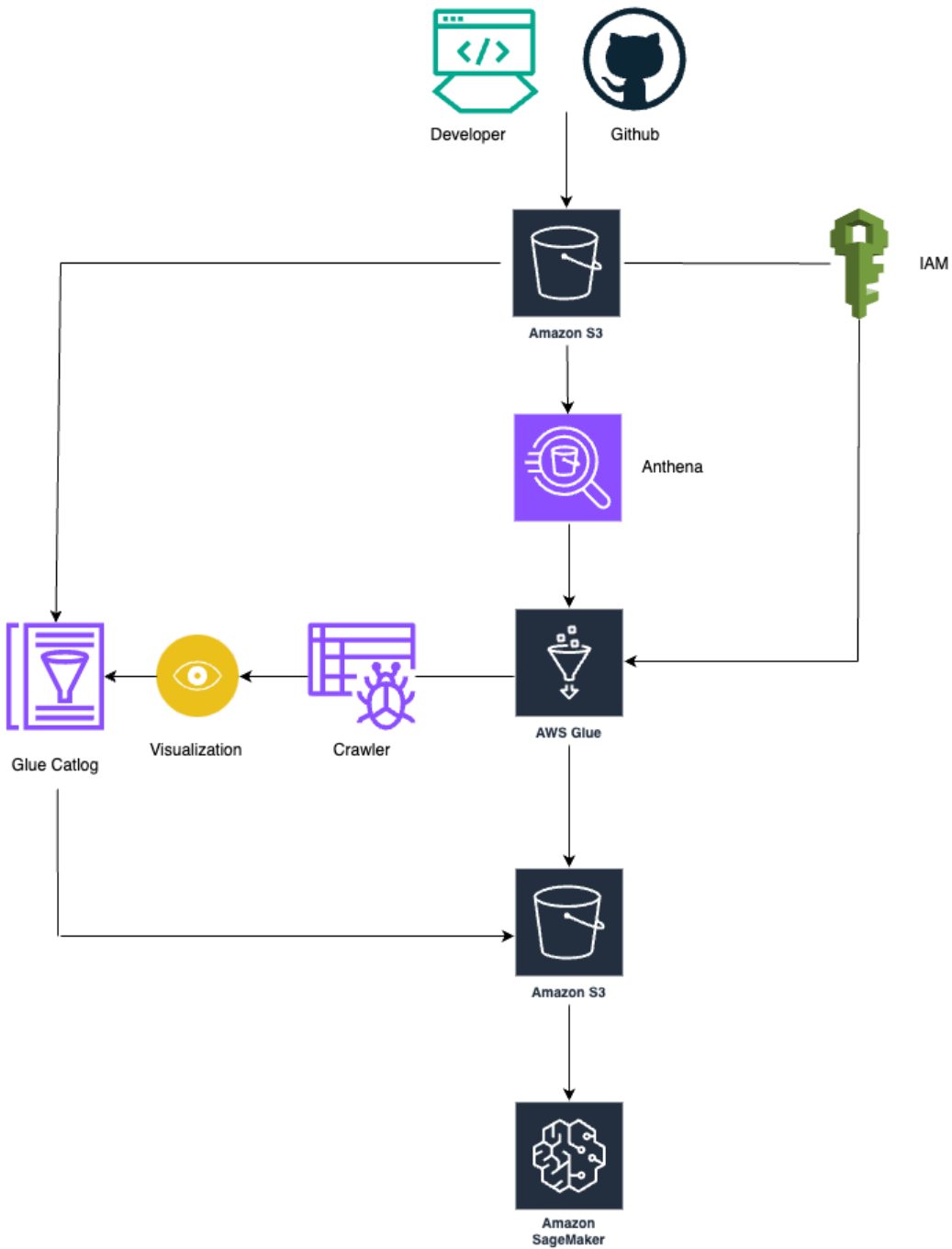
CloudShell

Feedback

© 2023, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preference](#)



## AWS pipeline Chart



Not able to access the Quick Sight.  
Quick sight



Your AWS Account is not signed up for QuickSight. Would you like to sign up now?

**AWS Account**

**501723330586**

[Sign up for QuickSight](#)

To access QuickSight with a different account, [log in](#) again.

