School of Information studies
Syracuse University

IST615 - Cloud Management

Final Project Progress Report

Professor in-charge: Carlos E. Caicedo Bastidas

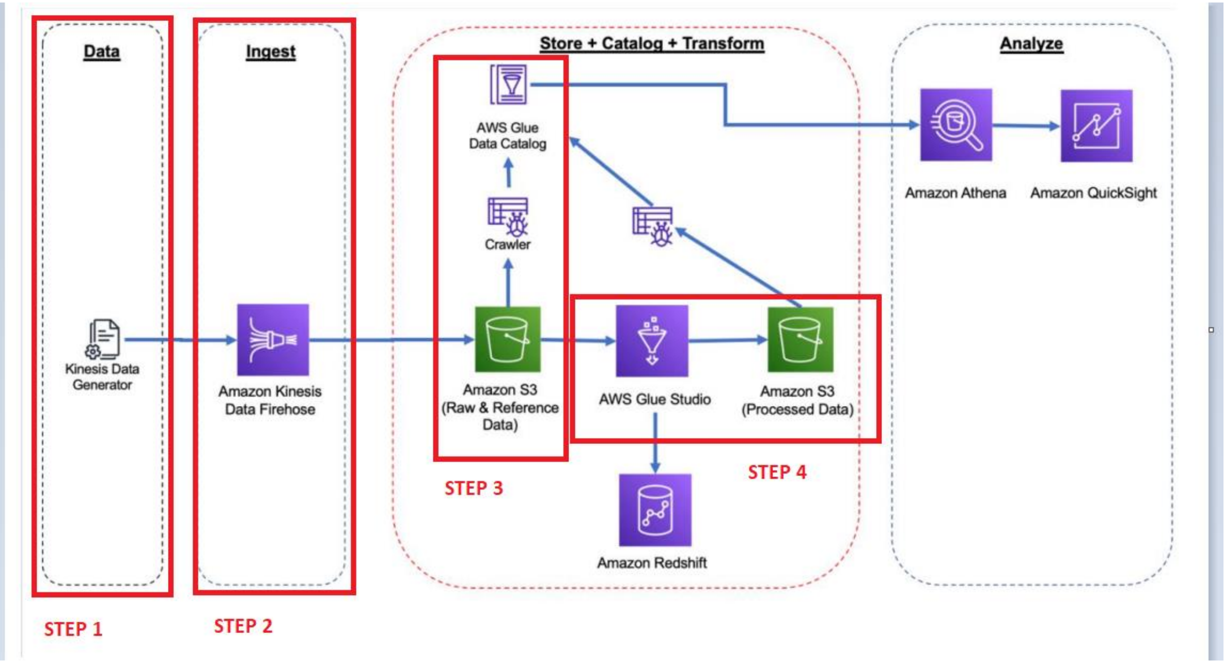
Group 6:

Names:

1) Rahul Jadhav	SUID: 705889151
2) Ramazan Yener	SUID: 4604870161
3) Francia Lizeth Ortiz Leyva	SUID: 2566771622
4) Himamshu Chandrashekara	SUID: 254732602

Due Date – 11/10/2021

Tasks completed so far -



Step 1 -

Generated Data from Kinesis Data generator.

Step 2 –

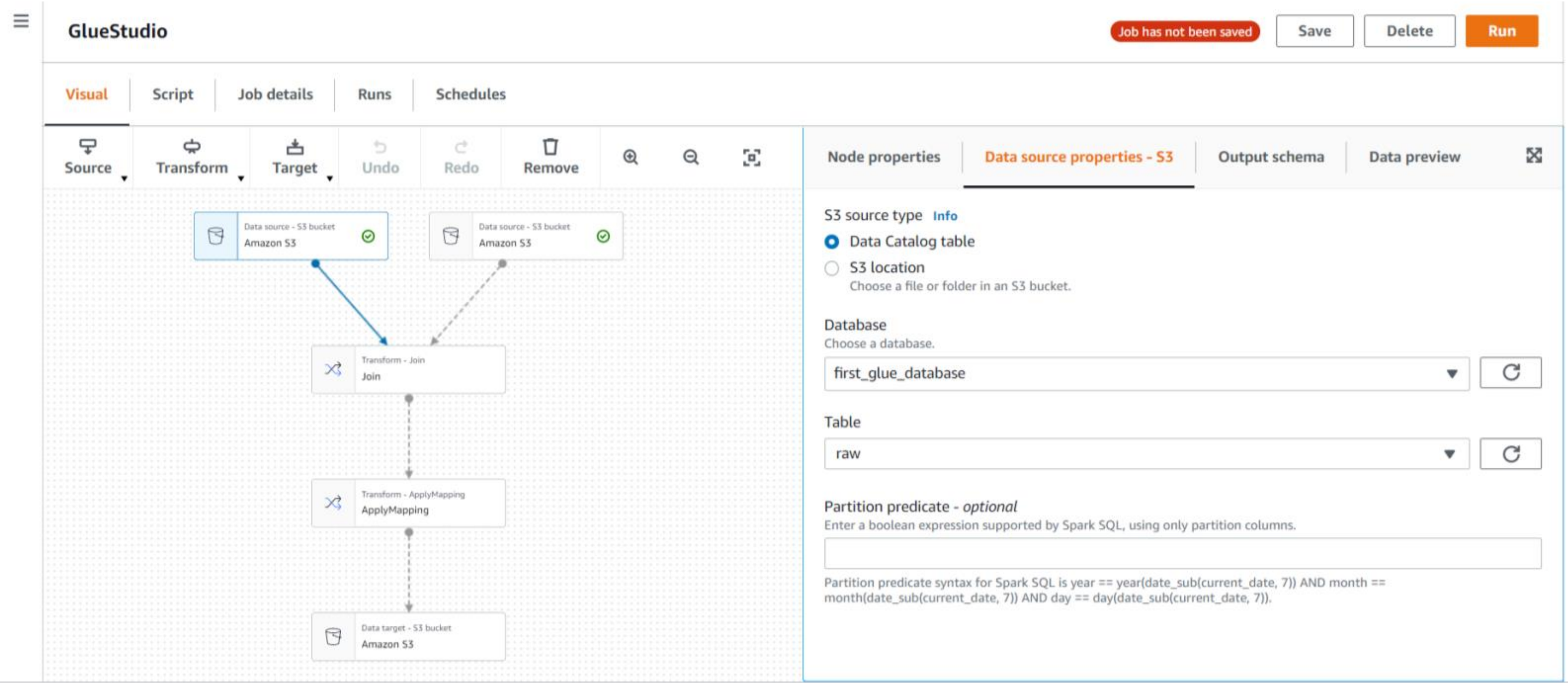
Delivered the data to Amazon S3 bucket with Kinesis Firehose delivery stream. Processed and stored our data into 1st S3 bucket (Raw and Reference data).

Step 3 –

Registered the datasets in the AWS Glue Data Catalog and automated the metadata captured with the help of Glue Crawlers.

Step 4 –

Transformed Data from 1st AWS s3 bucket from table named raw to 2nd AWS S3 bucket (Processed data) from table named reference_data using AWS Glue Studio.



S3 source type Info

● Data Catalog table

○ S3 location
Choose a file or folder in an S3 bucket.

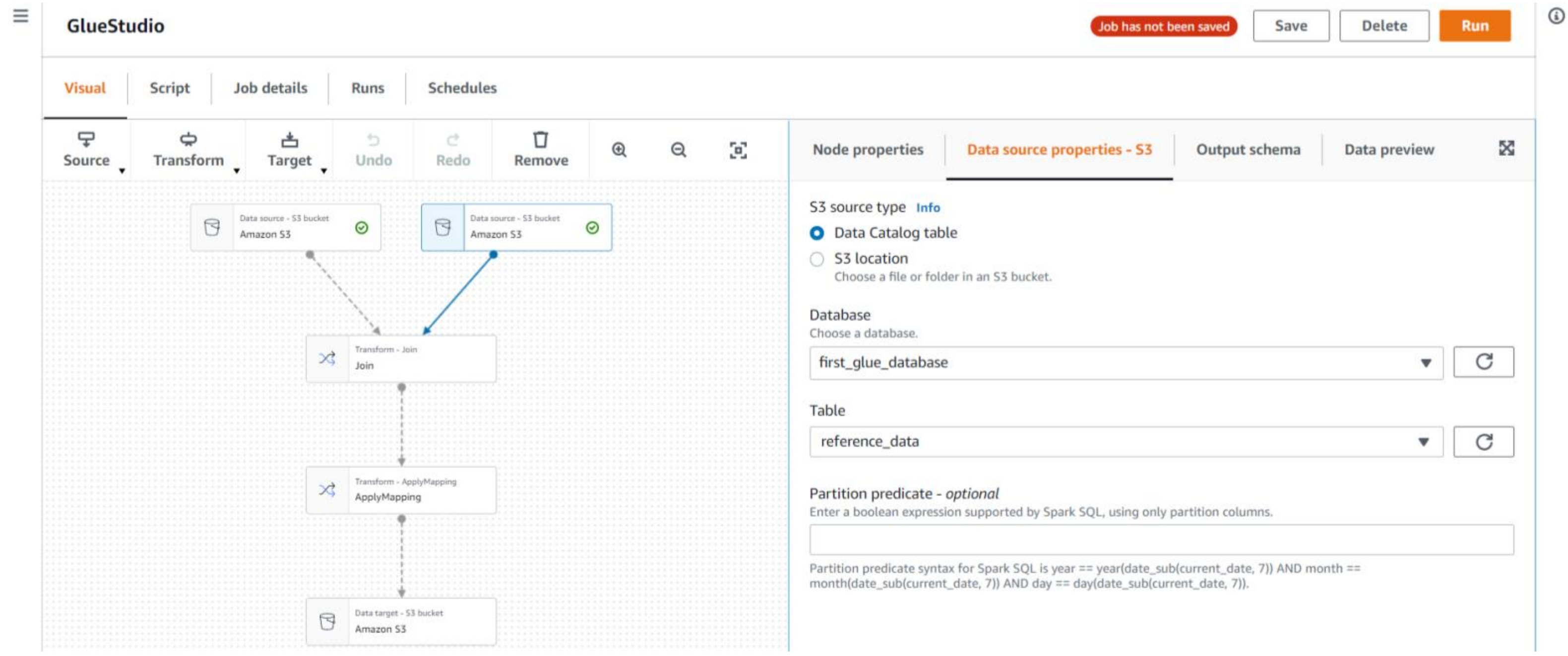
Database
Choose a database.
first_glue_database

Table
raw

Partition predicate - *optional*
Enter a boolean expression supported by Spark SQL, using only partition columns.

Partition predicate syntax for Spark SQL is year == year(date_sub(current_date, 7)) AND month == month(date_sub(current_date, 7)) AND day == day(date_sub(current_date, 7)).

Description: In the above screenshot we selected our first source data from Data Catalog Table i.e. raw Table



S3 source type Info

● Data Catalog table

○ S3 location
Choose a file or folder in an S3 bucket.

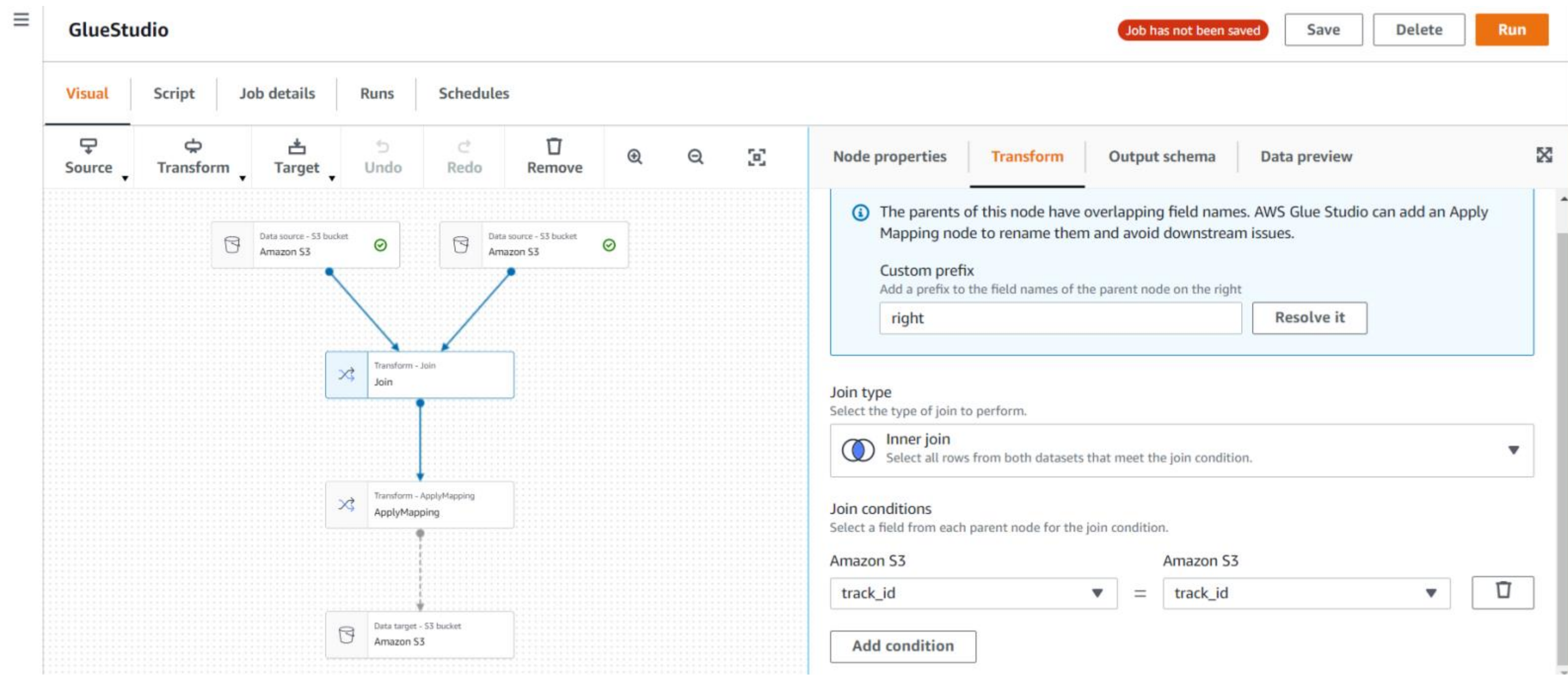
Database
Choose a database.
first_glue_database

Table
reference_data

Partition predicate - *optional*
Enter a boolean expression supported by Spark SQL, using only partition columns.

Partition predicate syntax for Spark SQL is year == year(date_sub(current_date, 7)) AND month == month(date_sub(current_date, 7)) AND day == day(date_sub(current_date, 7)).

Description: In the above screenshot we selected the another source data from Data Catalog Table i.e. reference_data Table.



The parents of this node have overlapping field names. AWS Glue Studio can add an Apply Mapping node to rename them and avoid downstream issues.

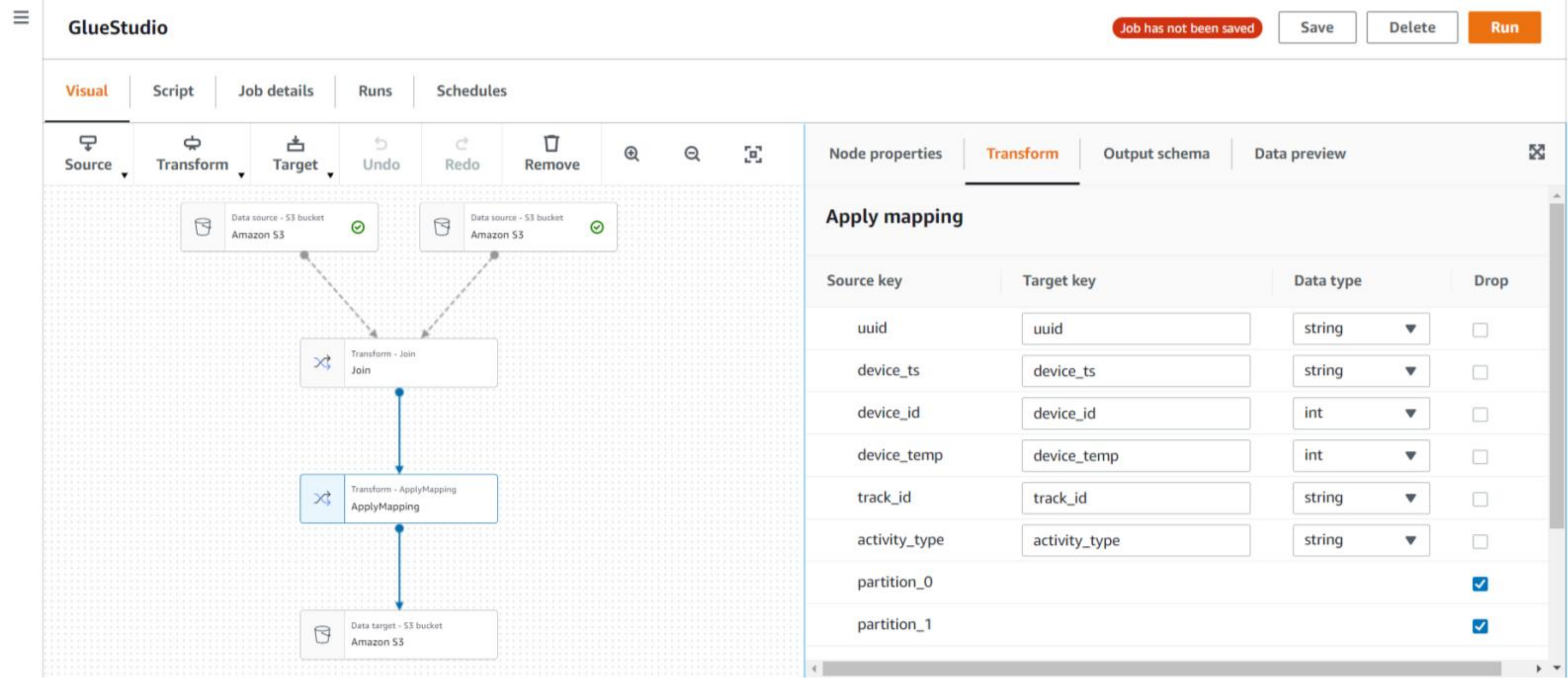
Custom prefix
Add a prefix to the field names of the parent node on the right
right

Join type
Select the type of join to perform.
Inner join
Select all rows from both datasets that meet the join condition.

Join conditions
Select a field from each parent node for the join condition.

Amazon S3		Amazon S3
track_id	=	track_id

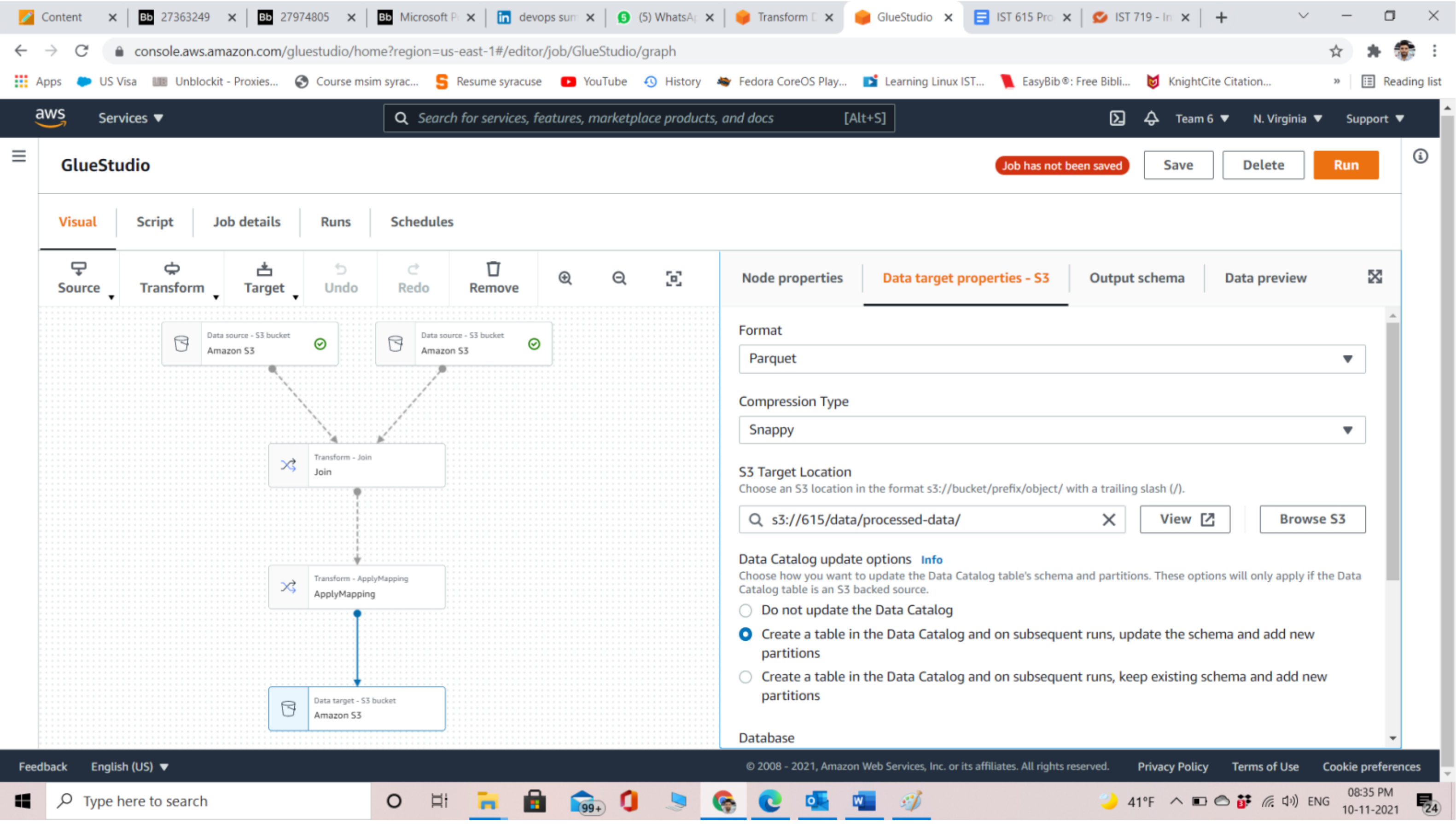
Description: We used inner join transformation to join the parent s3 buckets data(raw & reference_data) using the common field as track_id



Apply mapping

Source key	Target key	Data type	Drop
uuid	uuid	string	□
device_ts	device_ts	string	□
device_id	device_id	int	□
device_temp	device_temp	int	□
track_id	track_id	string	□
activity_type	activity_type	string	□
partition_0			☑
partition_1			☑

Description : In the above screenshot we apply mapping to the new target keys for the pushing the new processed dat into s3 bucket.



Format

Parquet

Compression Type

Snappy

S3 Target Location

Choose an S3 location in the format s3://bucket/prefix/object/ with a trailing slash (/).

s3://615/data/processed-data/

Data Catalog update options Info

Choose how you want to update the Data Catalog table's schema and partitions. These options will only apply if the Data Catalog table is an S3 backed source.

○ 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

Database

Description: In the above screenshot we provided the path for the target s3 bucket.

Issues Encountered

- As of now, we haven’t experienced any obstacles.

Changes to the project/goals

- None

Plan for completion of project week by week plan

Week 11/15 - 11/21

- Pushed this data into an Amazon Redshift Data for easier accessibility and referencing.
- Analyse the data using AWS Athena using SQL queries.

Week 11/29 - 12/05

- Use Amazon Quicksight to build visualizations over the data collected and stored in S3.