Guided Lab: Simplifying ETL Job Creation with AWS Glue Visual ETL

Description

Welcome to the AWS Glue Visual ETL lab! This lab will walk you through creating and saving an ETL (Extract, Transform, Load) job using AWS Glue's visual interface with minimal coding effort. This lab is designed for beginners and experienced professionals to explore the advantages of visual data manipulation over traditional script-based methods and show you how straightforward it is to set up and configure an ETL process visually.

AWS Glue is a fully managed extract, transform, and load (ETL) service that makes it easy for customers to prepare and load their data for analytics. It provides a Python shell script that can be used to perform data transformations, including redacting sensitive data.

The visual ETL (Extract, Transform, Load) in AWS Glue refers to the graphical interface provided by AWS Glue Studio, which allows you to author and manage Glue jobs without writing code directly. AWS Glue Studio offers a drag-and-drop interface where you can define the flow of your data sources, transformations, and targets.

Prerequisites

This lab assumes you have experience creating AWS Glue Jobs using Python Shell and experience creating an Amazon S3 bucket and are familiar with its basic components.

If you find any gaps in your knowledge, consider taking the following labs:

- Creating an Amazon S3 bucket
- Introduction to AWS Glue Using Python Shell

Objectives

In this lab, you will:

- Simulate creating an ETL job using AWS Glue Visual ETL
- Understand the interface and tools provided by Glue Visual ETL.
- Apply redaction, and change schema transformations to a dataset.

Subscribe to access AWS PlayCloud Labs

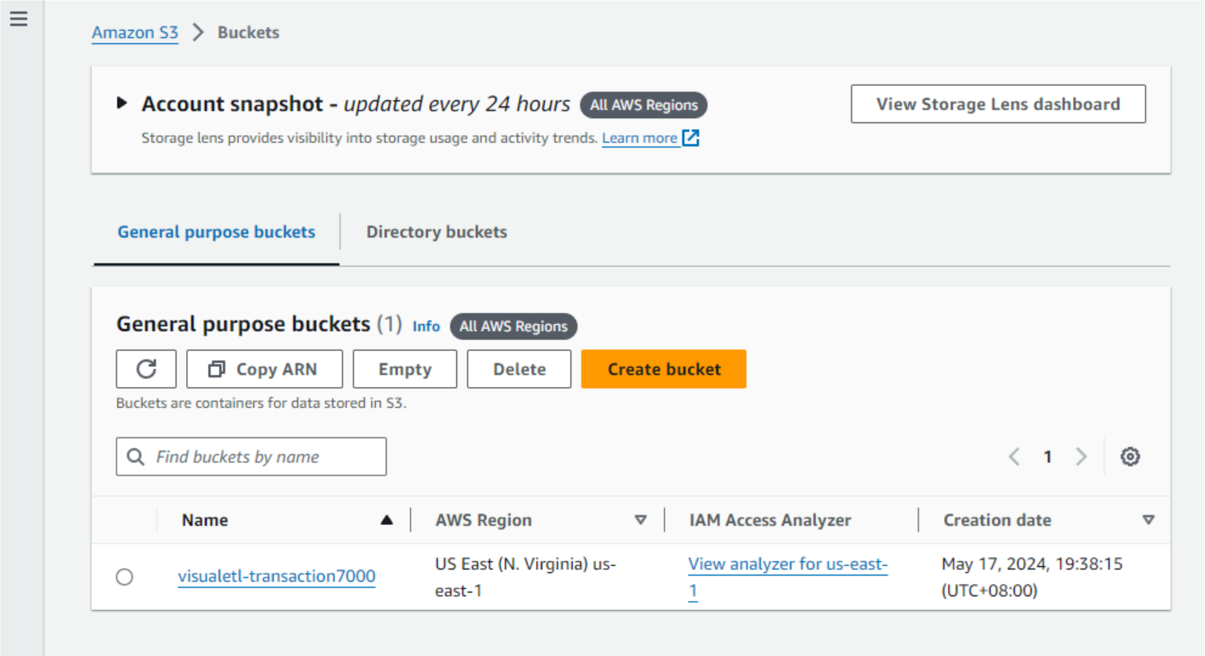
Lab Steps

Prepare your Environment

1. Log into the AWS Management Console.

2. Create an S3 Bucket:

- Go to the S3 service in the AWS Console.
- Click **Create bucket**.
- Give your bucket a unique name.
- Leave the rest of the settings as default and click **Create bucket**



Storage lens provides visibility into storage usage and activity trends. Learn more

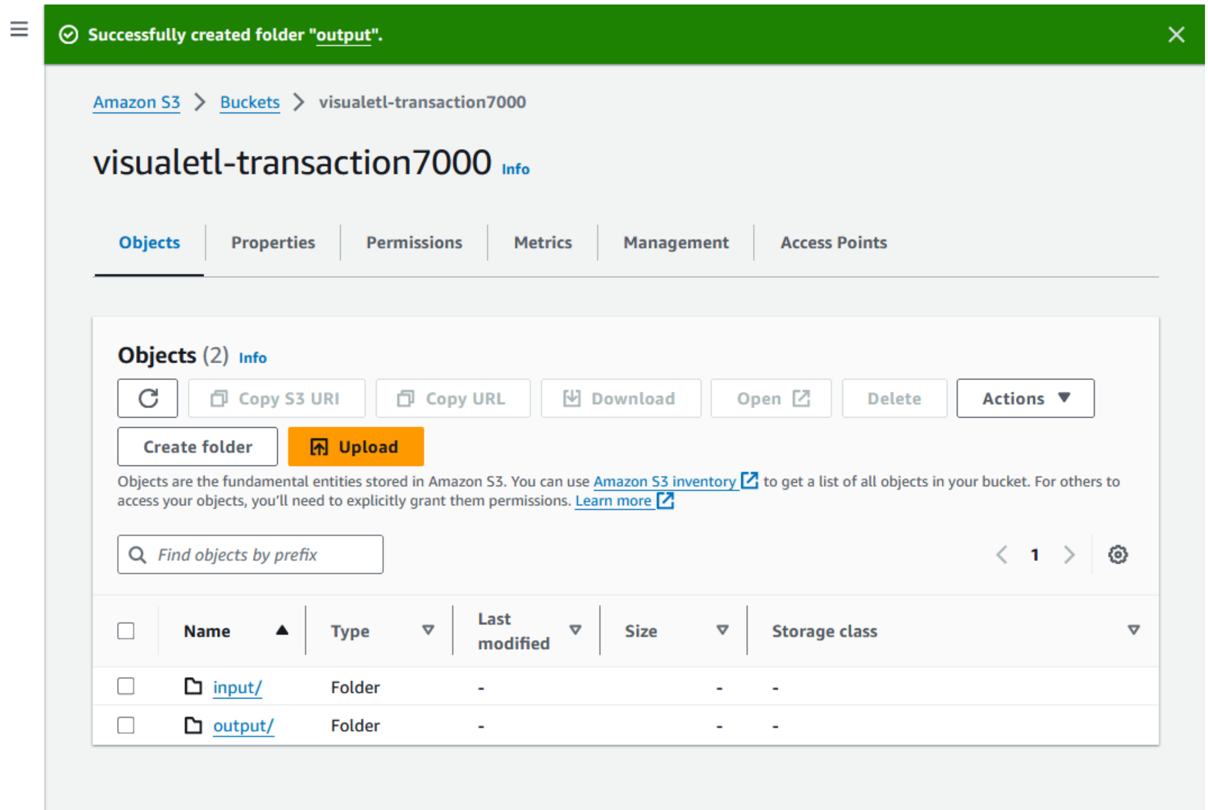
General purpose buckets (1) Info All AWS Regions

Buckets are containers for data stored in S3.

	Name	AWS Region	IAM Access Analyzer	Creation date
○	visualetl-transaction7000	US East (N. Virginia) us-east-1	View analyzer for us-east-1	May 17, 2024, 19:38:15 (UTC+08:00)

3. Create a two folders

- Navigate to your newly created bucket.
- Create two folders
    - **input**
    - **output**



Successfully created folder "output".

visualetl-transaction7000 Info

Objects (2) Info

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

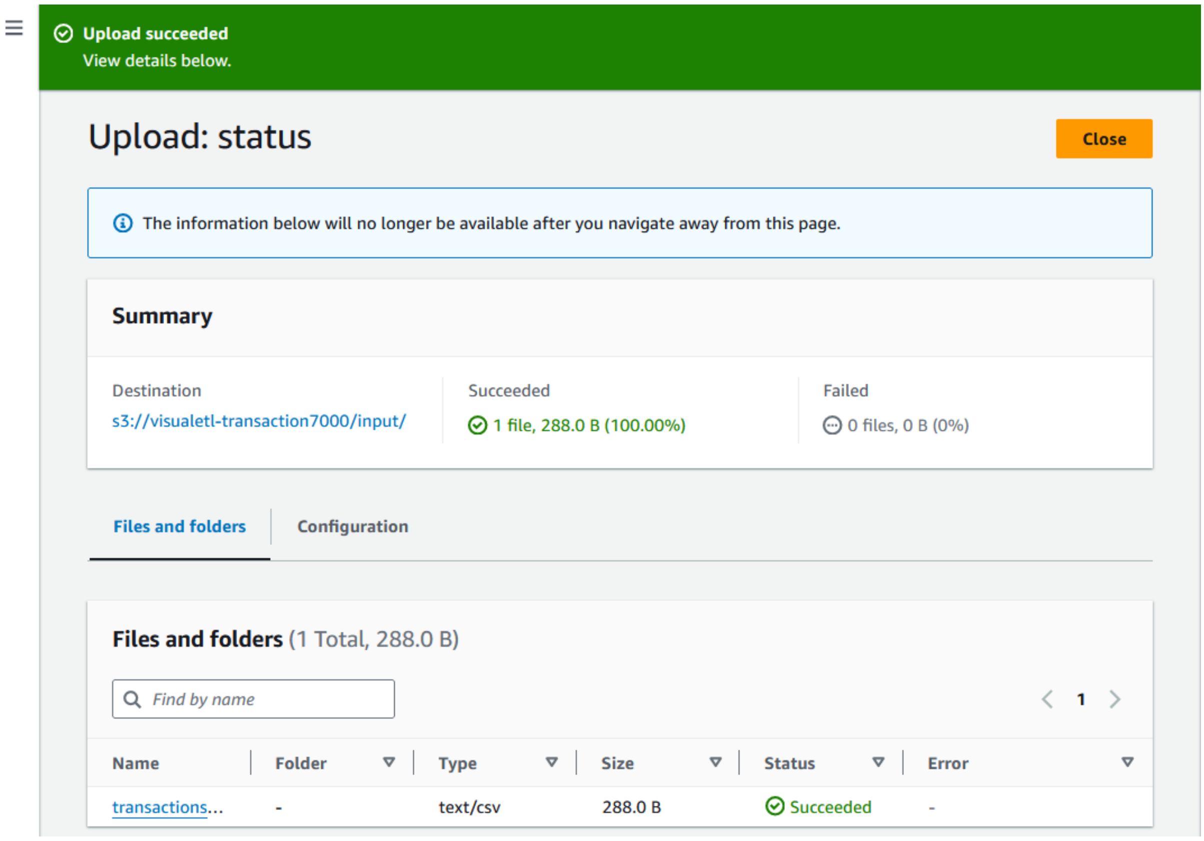
□	Name	Type	Last modified	Size	Storage class
□	input/	Folder	-	-	-
□	output/	Folder	-	-	-

4. Download the CSV Data:

https://media.tutorialsdojo.com/public/transactions.csv

Take a look at the provided CSV file by opening the CSV file using Excel or any other spreadsheet application to visualize the data. When you are done, you can continue uploading the file.

5. Upload the file to the *input/* folder



Upload succeeded
View details below.

Upload: status

The information below will no longer be available after you navigate away from this page.

Summary

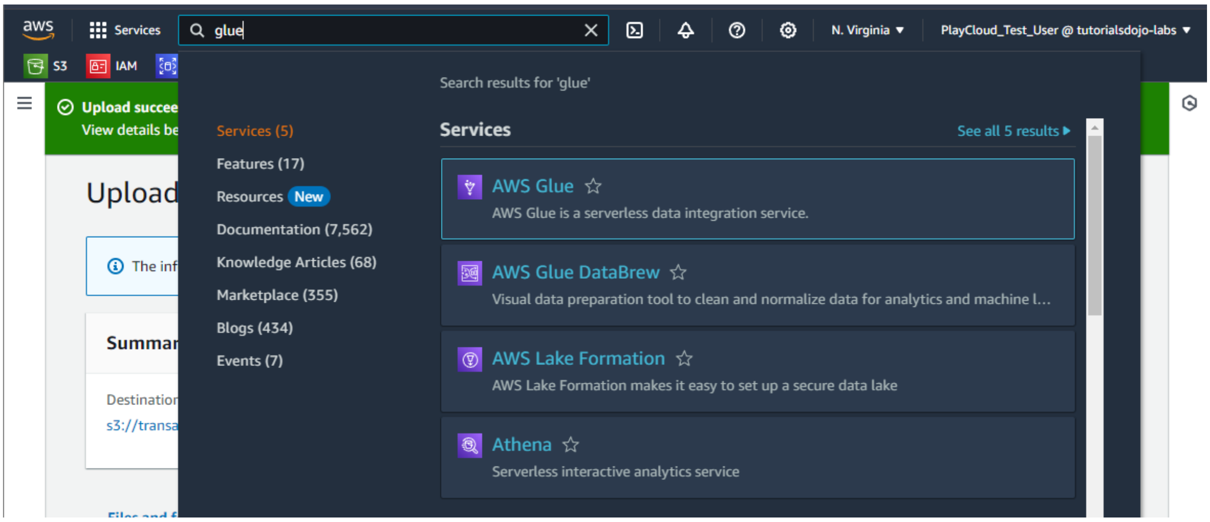
Destination	Succeeded	Failed
s3://visualetl-transaction7000/input/	1 file, 288.0 B (100.00%)	0 files, 0 B (0%)

Files and folders (1 Total, 288.0 B)

Name	Folder	Type	Size	Status	Error
transactions...	-	text/csv	288.0 B	Succeeded	-

Creating your ETL Job using Visual ETL

1. Navigate to the **AWS Glue Console**.



Search results for 'glue'

Services

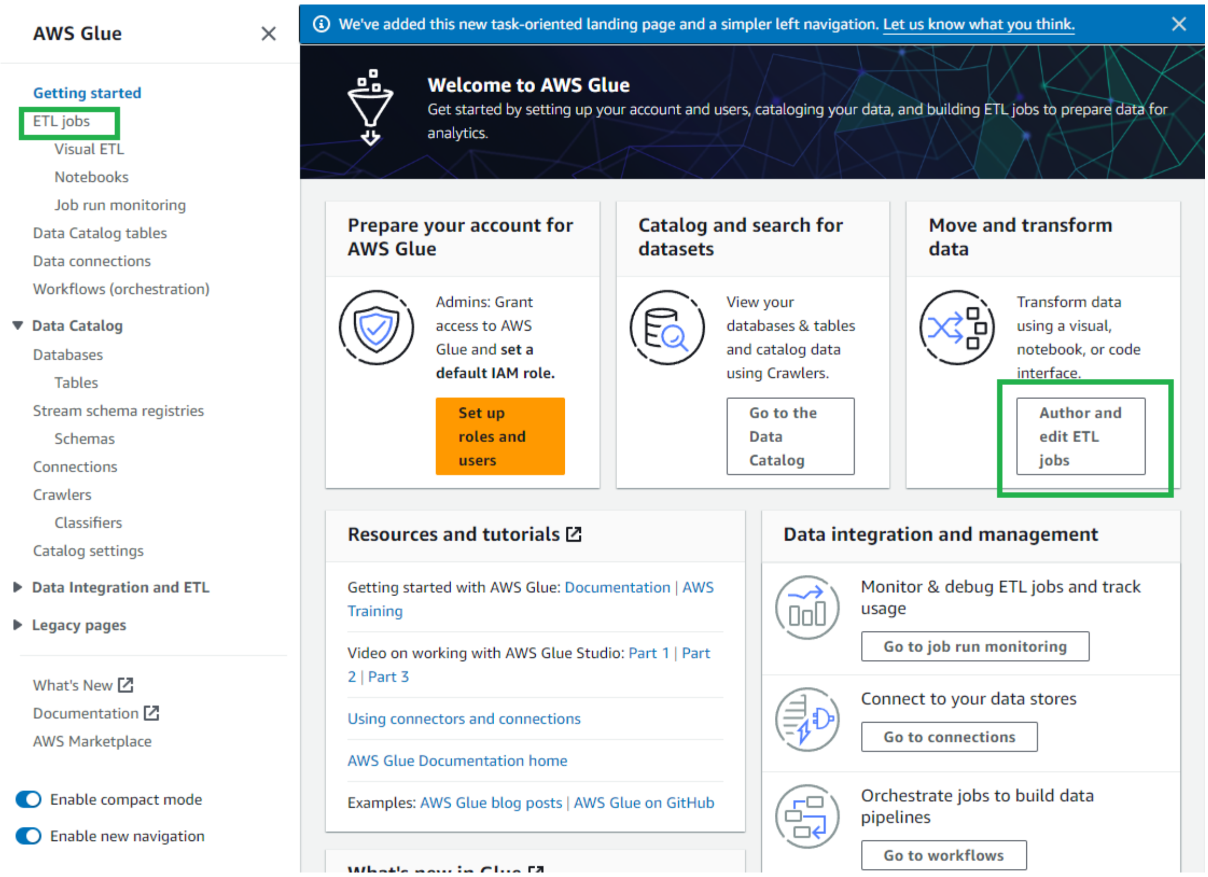
AWS Glue
AWS Glue is a serverless data integration service.

AWS Glue DataBrew
Visual data preparation tool to clean and normalize data for analytics and machine l...

AWS Lake Formation
AWS Lake Formation makes it easy to set up a secure data lake

Athena
Serverless interactive analytics service

2. You can either click on **ETL Jobs** on the left bar or click on the **Author and edit ETL jobs**



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

▶ Legacy pages

What's New

Documentation

AWS Marketplace

Enable compact mode

Enable new navigation

We've added this new task-oriented landing page and a simpler left navigation. Let us know what you think. ✕

Welcome to AWS Glue

Get started by setting up your account and users, cataloging your data, and building ETL jobs to prepare data for analytics.

Prepare your account for AWS Glue

Admins: Grant access to AWS Glue and set a **default IAM role.**

Set up roles and users

Catalog and search for datasets

View your databases & tables and catalog data using Crawlers.

Go to the Data Catalog

Move and transform data

Transform data using a visual, notebook, or code interface.

Author and edit ETL jobs

Resources and tutorials

Getting started with AWS Glue: Documentation | AWS Training

Video on working with AWS Glue Studio: Part 1 | Part 2 | Part 3

Using connectors and connections

AWS Glue Documentation home

Examples: AWS Glue blog posts | AWS Glue on GitHub

Data integration and management

Monitor & debug ETL jobs and track usage

Go to job run monitoring

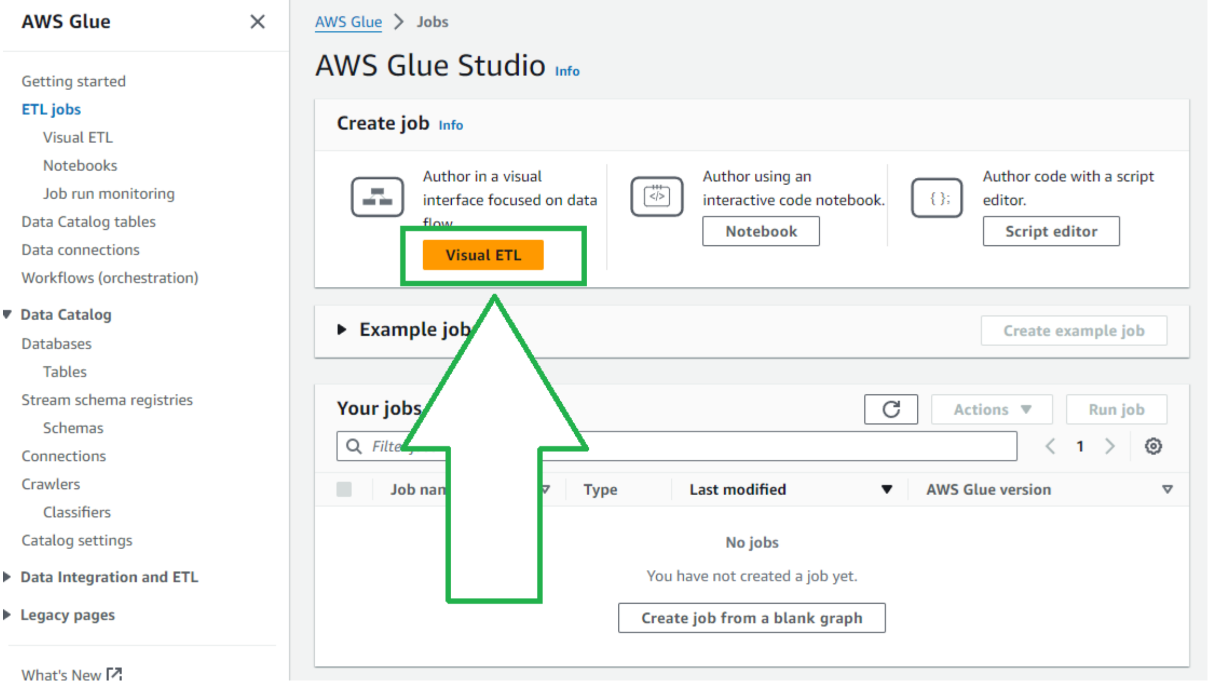
Connect to your data stores

Go to connections

Orchestrate jobs to build data pipelines

Go to workflows

3. Click on **Visual ETL**



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

▶ Legacy pages

What's New

AWS Glue > Jobs

AWS Glue Studio Info

Create job Info

Author in a visual interface focused on data flow.

Visual ETL

Author using an interactive code notebook.

Notebook

Author code with a script editor.

Script editor

▶ **Example job**

Create example job

Your jobs

Actions ▼ Run job

Filte

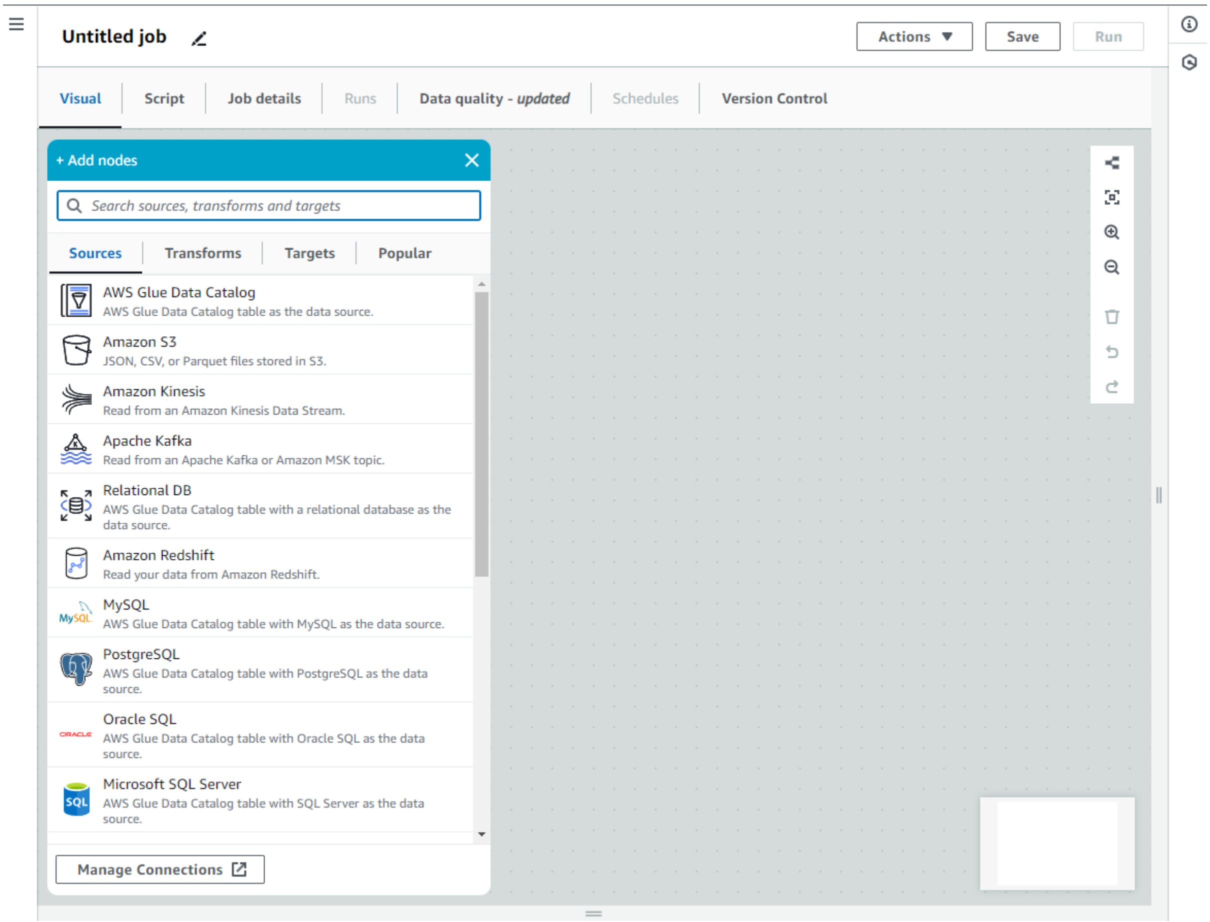
☐	Job nan	Type	Last modified ▼	AWS Glue version ▽

No jobs

You have not created a job yet.

Create job from a blank graph

4. Then, you will be redirected to the Visual ETL interface



Untitled job

AWS Glue Data Catalog
AWS Glue Data Catalog table as the data source.

Amazon S3
JSON, CSV, or Parquet files stored in S3.

Amazon Kinesis
Read from an Amazon Kinesis Data Stream.

Apache Kafka
Read from an Apache Kafka or Amazon MSK topic.

Relational DB
AWS Glue Data Catalog table with a relational database as the data source.

Amazon Redshift
Read your data from Amazon Redshift.

MySQL
AWS Glue Data Catalog table with MySQL as the data source.

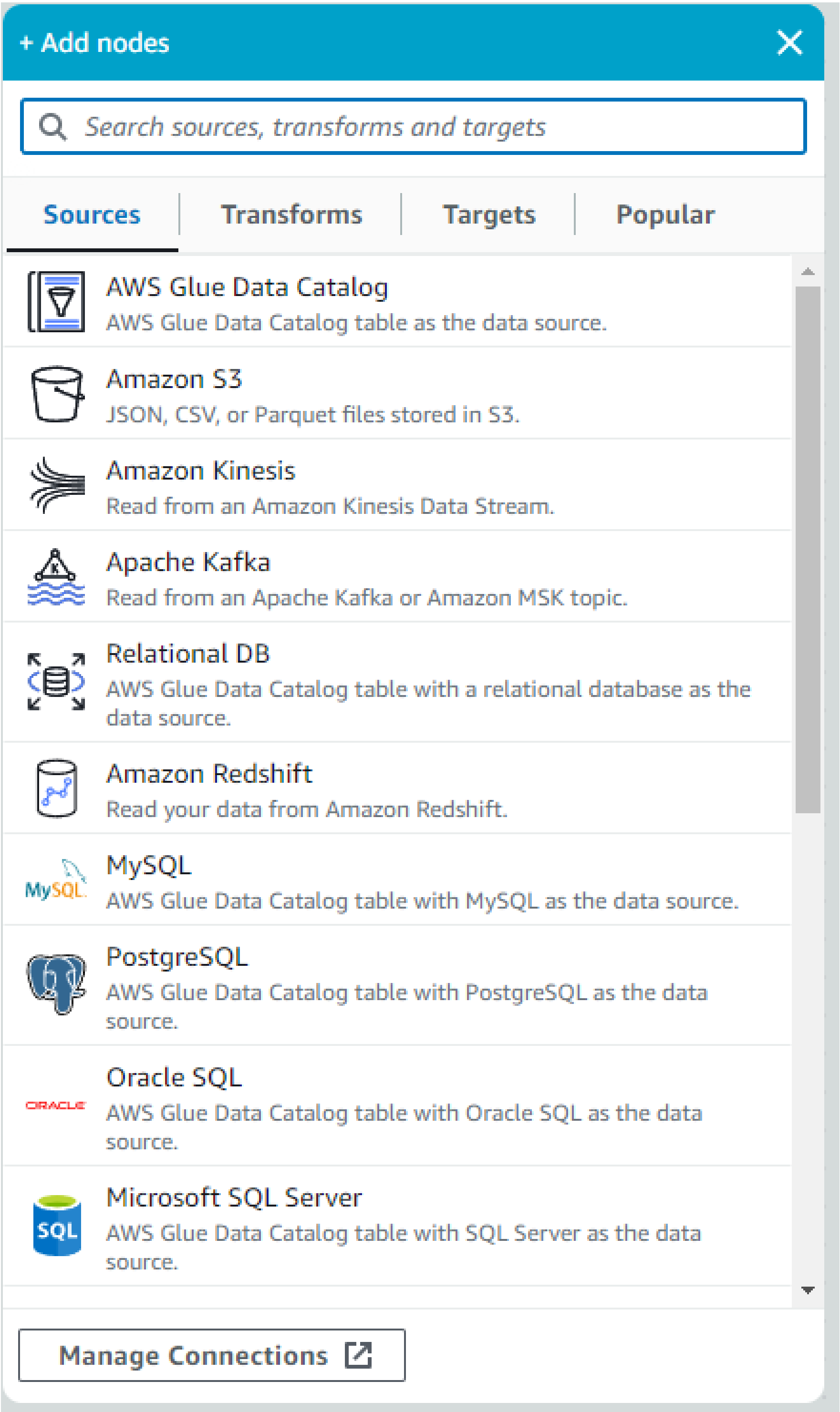
PostgreSQL
AWS Glue Data Catalog table with PostgreSQL as the data source.

Oracle SQL
AWS Glue Data Catalog table with Oracle SQL as the data source.

Microsoft SQL Server
AWS Glue Data Catalog table with SQL Server as the data source.

- If you notice in the **+ Add nodes,** there are tabs inside it:
-

- **Sources:**



AWS Glue Data Catalog
AWS Glue Data Catalog table as the data source.

Amazon S3
JSON, CSV, or Parquet files stored in S3.

Amazon Kinesis
Read from an Amazon Kinesis Data Stream.

Apache Kafka
Read from an Apache Kafka or Amazon MSK topic.

Relational DB
AWS Glue Data Catalog table with a relational database as the data source.

Amazon Redshift
Read your data from Amazon Redshift.

MySQL
AWS Glue Data Catalog table with MySQL as the data source.

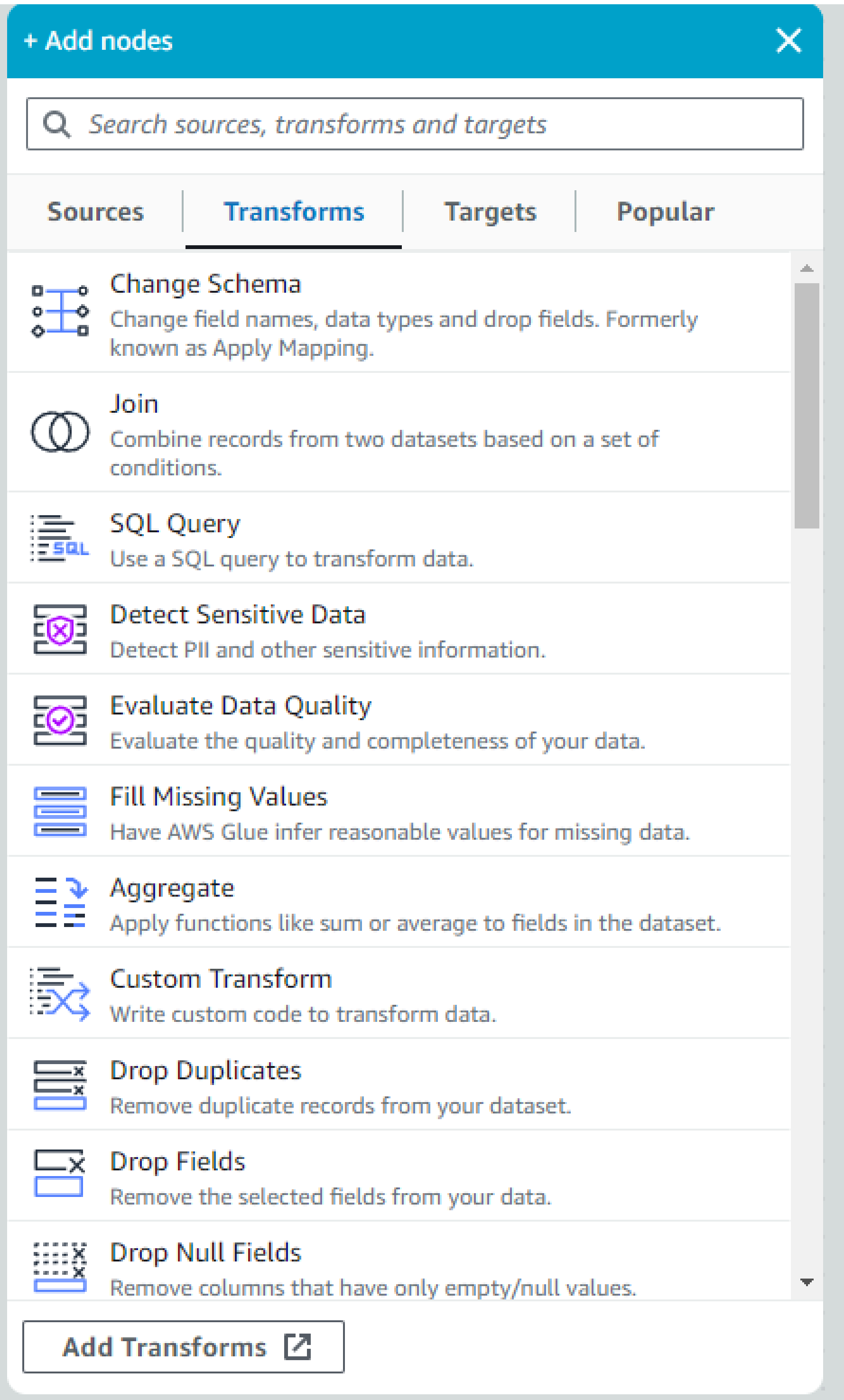
PostgreSQL
AWS Glue Data Catalog table with PostgreSQL as the data source.

Oracle SQL
AWS Glue Data Catalog table with Oracle SQL as the data source.

Microsoft SQL Server
AWS Glue Data Catalog table with SQL Server as the data source.

  - Sources are used to connect to the data sources, such as databases, data lakes, or other data stores. These allow you to read in data using various connectors provided by AWS Glue.

- **Transforms:**



Change Schema
Change field names, data types and drop fields. Formerly known as Apply Mapping.

Join
Combine records from two datasets based on a set of conditions.

SQL Query
Use a SQL query to transform data.

Detect Sensitive Data
Detect PII and other sensitive information.

Evaluate Data Quality
Evaluate the quality and completeness of your data.

Fill Missing Values
Have AWS Glue infer reasonable values for missing data.

Aggregate
Apply functions like sum or average to fields in the dataset.

Custom Transform
Write custom code to transform data.

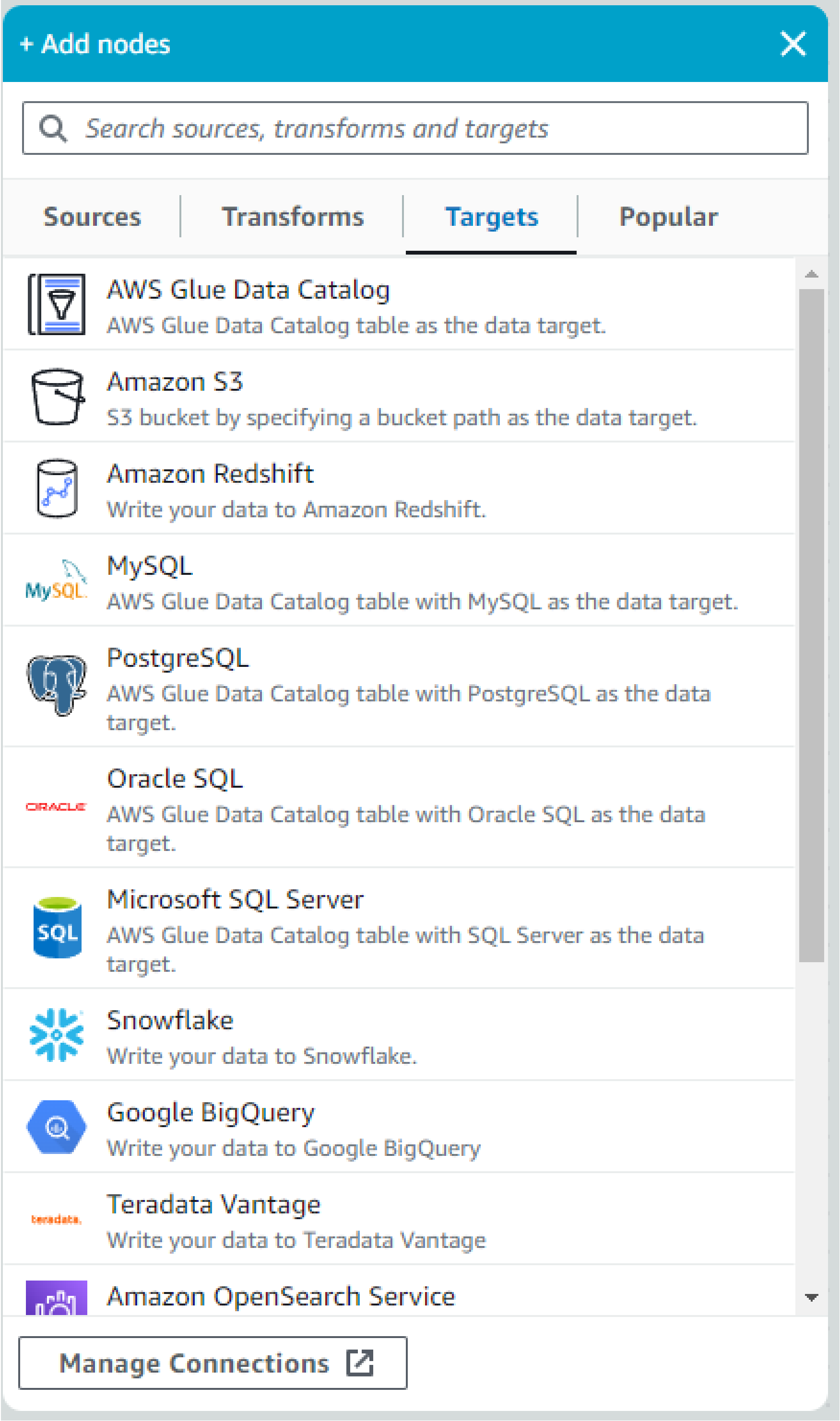
Drop Duplicates
Remove duplicate records from your dataset.

Drop Fields
Remove the selected fields from your data.

Drop Null Fields
Remove columns that have only empty/null values.

  - AWS Glue Studio offers a range of visual transforms, such as Concatenate, Split string, Pivot rows to columns, Lookup, and Derived column, which can be used to build more sophisticated data pipelines without writing code.

- **Targets:**



+ Add nodes

Search sources, transforms and targets

AWS Glue Data Catalog
AWS Glue Data Catalog table as the data target.

Amazon S3
S3 bucket by specifying a bucket path as the data target.

Amazon Redshift
Write your data to Amazon Redshift.

MySQL
AWS Glue Data Catalog table with MySQL as the data target.

PostgreSQL
AWS Glue Data Catalog table with PostgreSQL as the data target.

Oracle SQL
AWS Glue Data Catalog table with Oracle SQL as the data target.

Microsoft SQL Server
AWS Glue Data Catalog table with SQL Server as the data target.

Snowflake
Write your data to Snowflake.

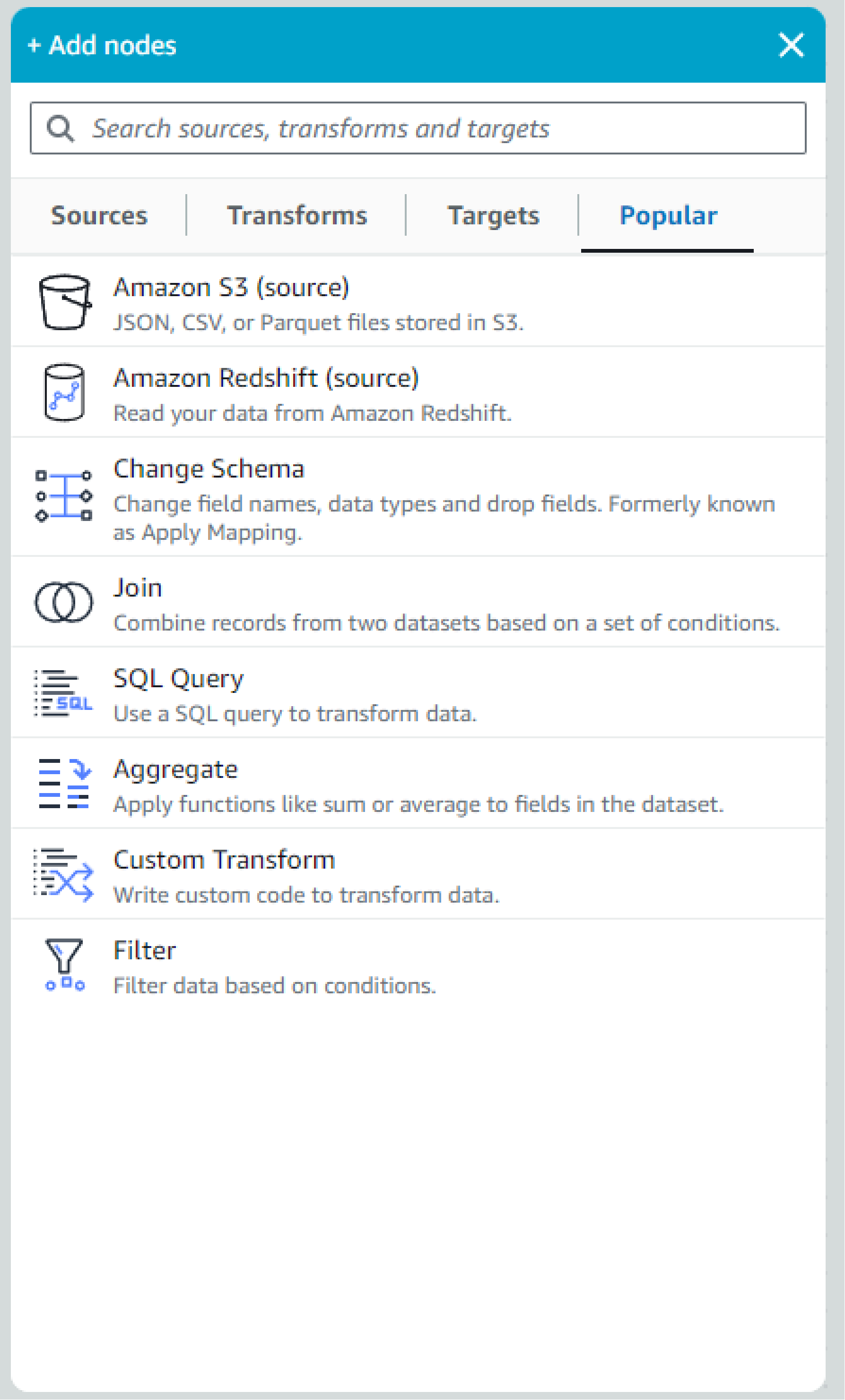
Google BigQuery
Write your data to Google BigQuery

Teradata Vantage
Write your data to Teradata Vantage

Amazon OpenSearch Service

  - Targets are used to specify the destination for writing out the transformed data. These allow you to define the location and format for the output data, such as a database table, a file in an S3 bucket, or a data stream.

- **Popular:**



Amazon S3 (source)
JSON, CSV, or Parquet files stored in S3.

Amazon Redshift (source)
Read your data from Amazon Redshift.

Change Schema
Change field names, data types and drop fields. Formerly known as Apply Mapping.

Join
Combine records from two datasets based on a set of conditions.

SQL Query
Use a SQL query to transform data.

Aggregate
Apply functions like sum or average to fields in the dataset.

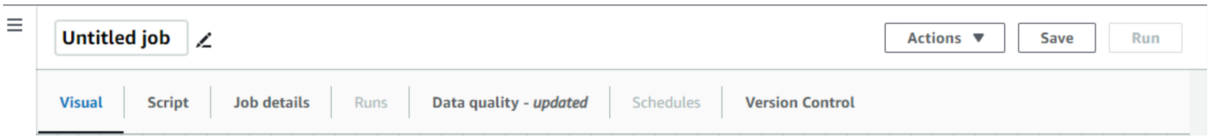
Custom Transform
Write custom code to transform data.

Filter
Filter data based on conditions.

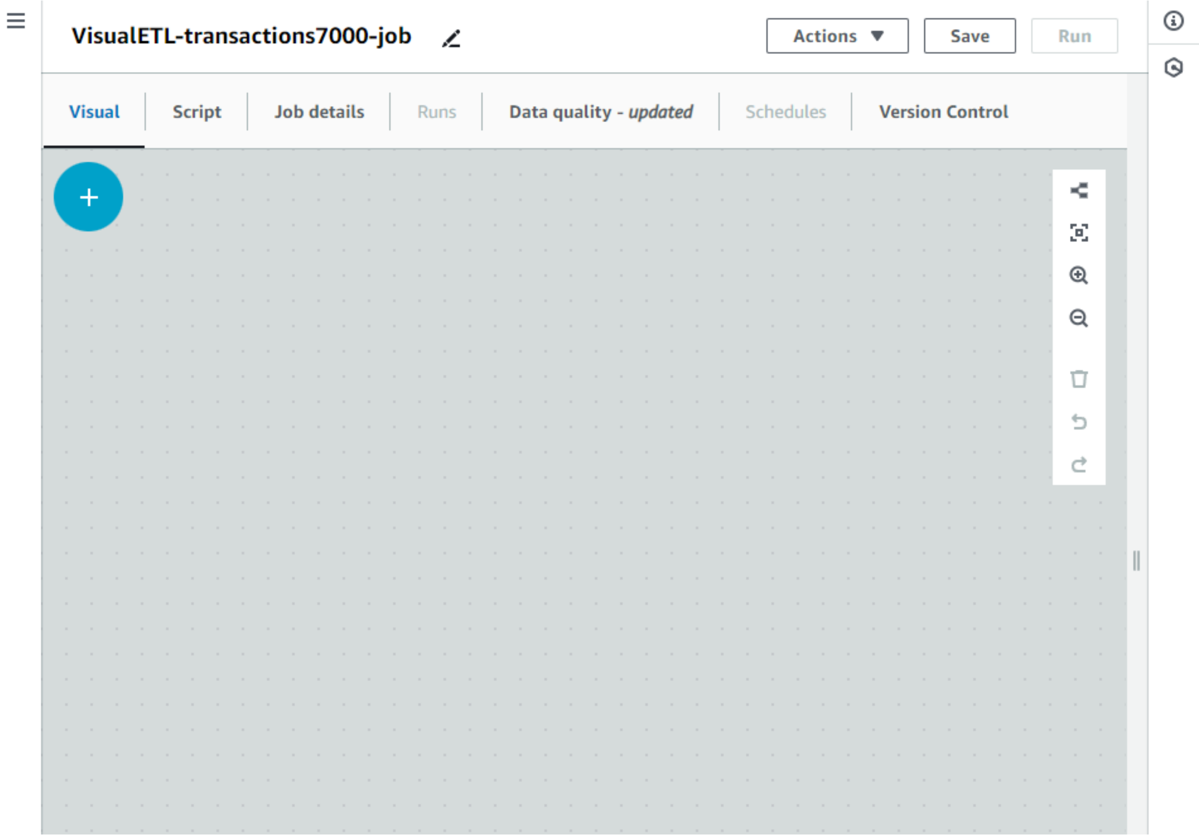
  - The **Popular** tab is designed to help you quickly build ETL pipelines by providing easy access for commonly used nodes.

5. Now, let’s **rename our job.**

- Click on the **Untitled job** in the upper left corner right beside the pencil icon.

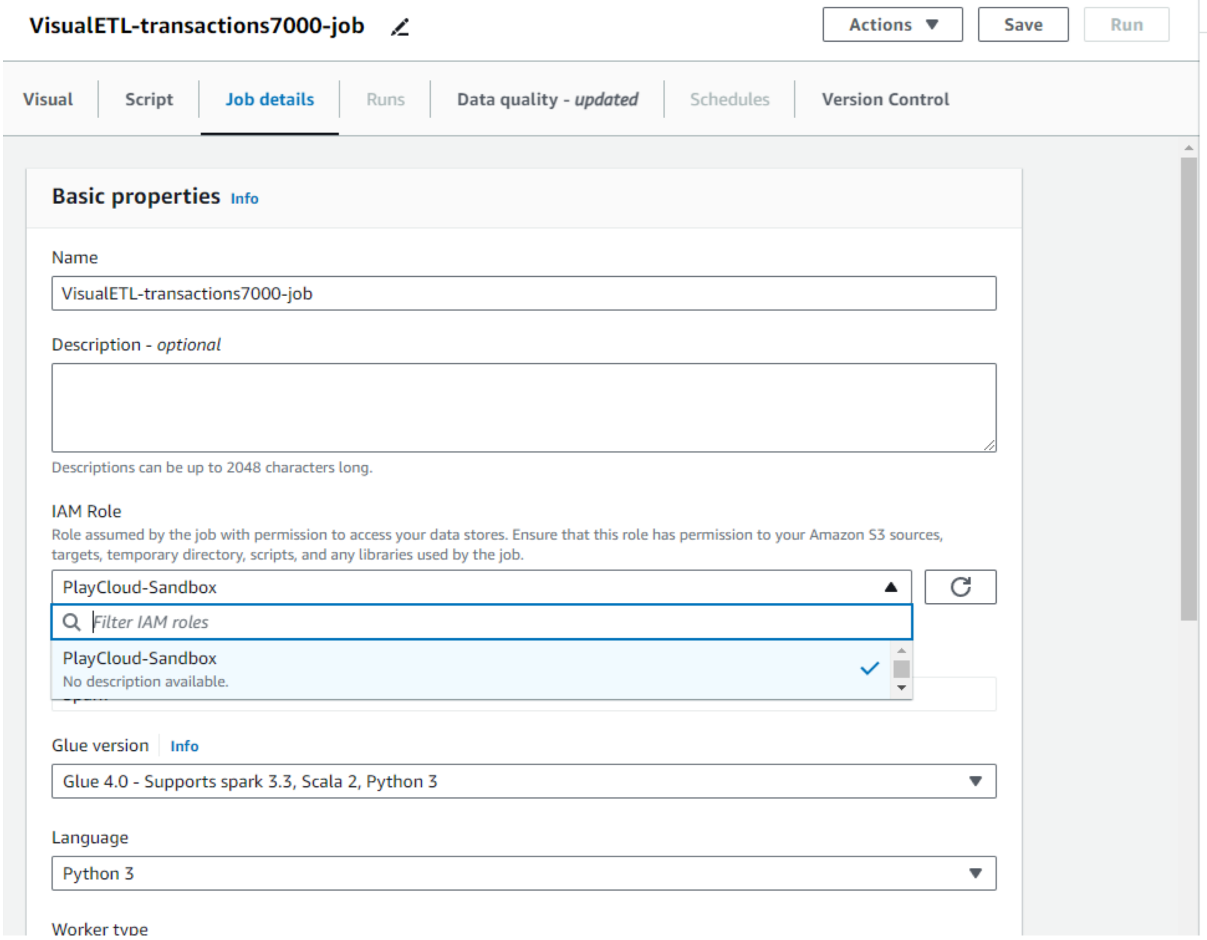


- Name it uniquely


VisualETL-transactions7000-job

6. Next, click on the Job details tab

- Select the **PlayCloud-Sandbox** IAM role in the dropdown.



VisualETL-transactions7000-job

Basic properties Info

Name

VisualETL-transactions7000-job

Description - *optional*

Descriptions can be up to 2048 characters long.

IAM Role

Role assumed by the job with permission to access your data stores. Ensure that this role has permission to your Amazon S3 sources, targets, temporary directory, scripts, and any libraries used by the job.

PlayCloud-Sandbox

Filter IAM roles

PlayCloud-Sandbox ✓
No description available.

Glue version Info

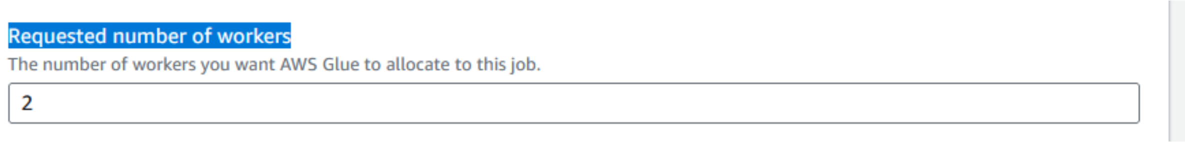
Glue 4.0 - Supports spark 3.3, Scala 2, Python 3

Language

Python 3

Worker type

- Scroll down and look for **Requested number of workers.** Type 2 for number of workers. (optional step)
__NOTE:__ The __Number of Workers__ determines the __DPU usage__ of your job, depending on your worker type. The higher your DPU usage, the higher the bill you will pay. So be mindful of your DPU usage when doing AWS GLUE experiments in your AWS Account.

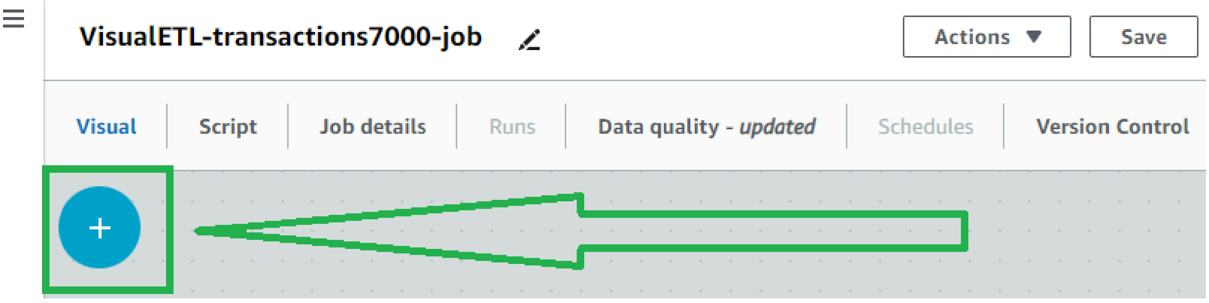


Requested number of workers

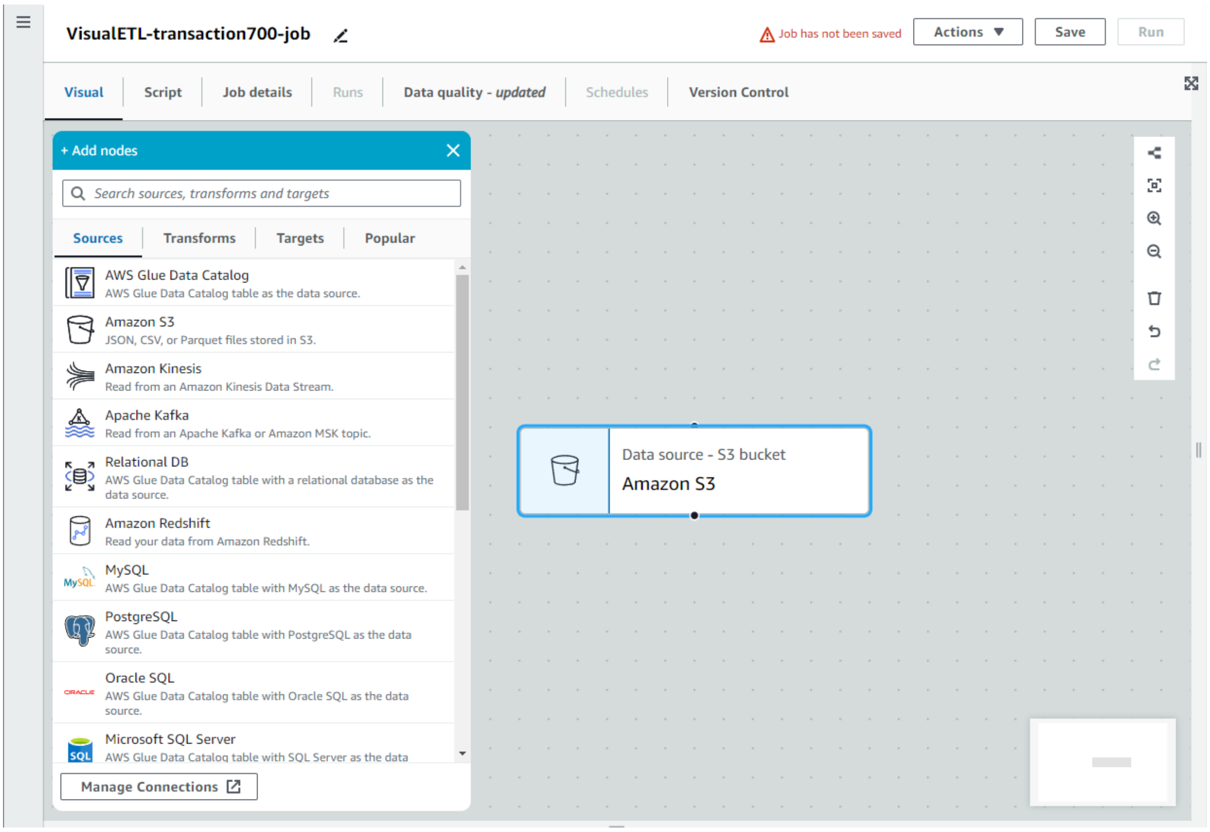
The number of workers you want AWS Glue to allocate to this job.

2

7. Return to the previous tab (Visual) and click on the **blue circle with the+ sign** to add nodes.

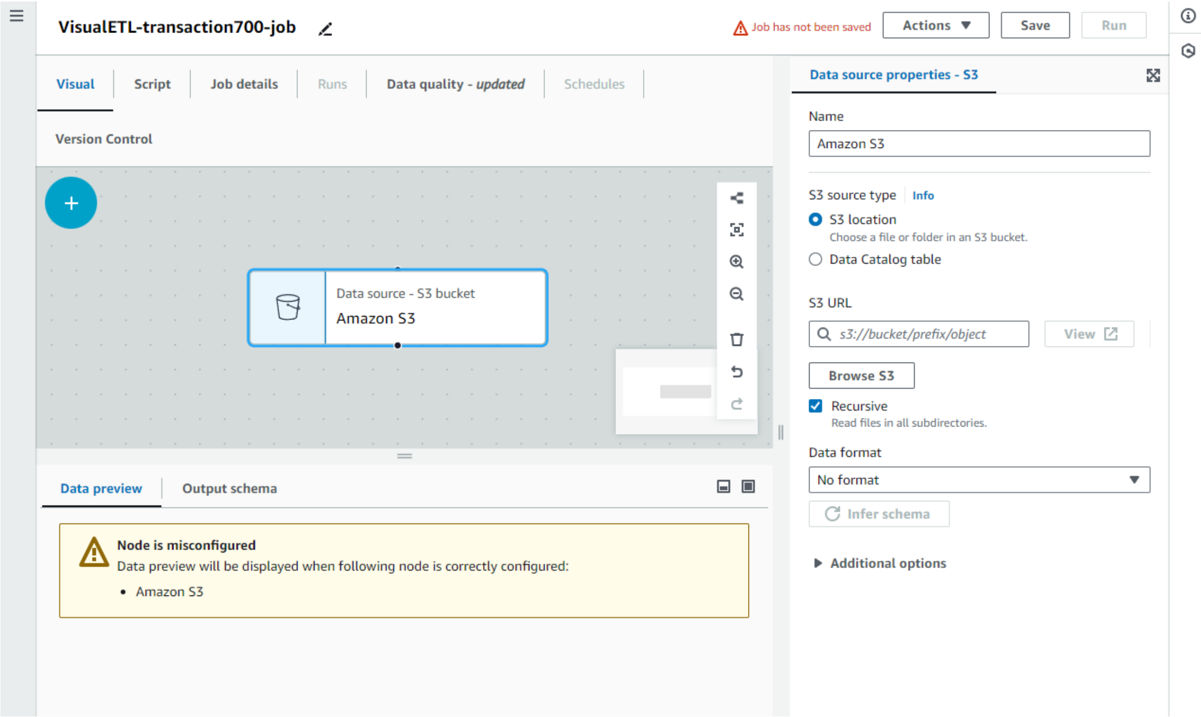


8. Let’s add the **Source node** first. In the **sources,** click on **Amazon S3.**


VisualETL-transaction700-job
Job has not been saved
Actions
Save
Run
Visual
Script
Job details
Runs
Data quality - updated
Schedules
Version Control
+ Add nodes
Search sources, transforms and targets
Sources
Transforms
Targets
Popular
AWS Glue Data Catalog
AWS Glue Data Catalog table as the data source.
Amazon S3
JSON, CSV, or Parquet files stored in S3.
Amazon Kinesis
Read from an Amazon Kinesis Data Stream.
Apache Kafka
Read from an Apache Kafka or Amazon MSK topic.
Relational DB
AWS Glue Data Catalog table with a relational database as the data source.
Amazon Redshift
Read your data from Amazon Redshift.
MySQL
AWS Glue Data Catalog table with MySQL as the data source.
PostgreSQL
AWS Glue Data Catalog table with PostgreSQL as the data source.
Oracle SQL
AWS Glue Data Catalog table with Oracle SQL as the data source.
Microsoft SQL Server
AWS Glue Data Catalog table with SQL Server as the data
Manage Connections
Data source - S3 bucket
Amazon S3

*Notice that after clicking on the source **Amazon S3**, a node appeared in our interface*

9. Click on the **Amazon S3 node** that just appeared.



Node is misconfigured
Data preview will be displayed when following node is correctly configured:
- Amazon S3

Name: Amazon S3

S3 source type Info
● S3 location
Choose a file or folder in an S3 bucket.
○ Data Catalog table

S3 URL: s3://bucket/prefix/object View

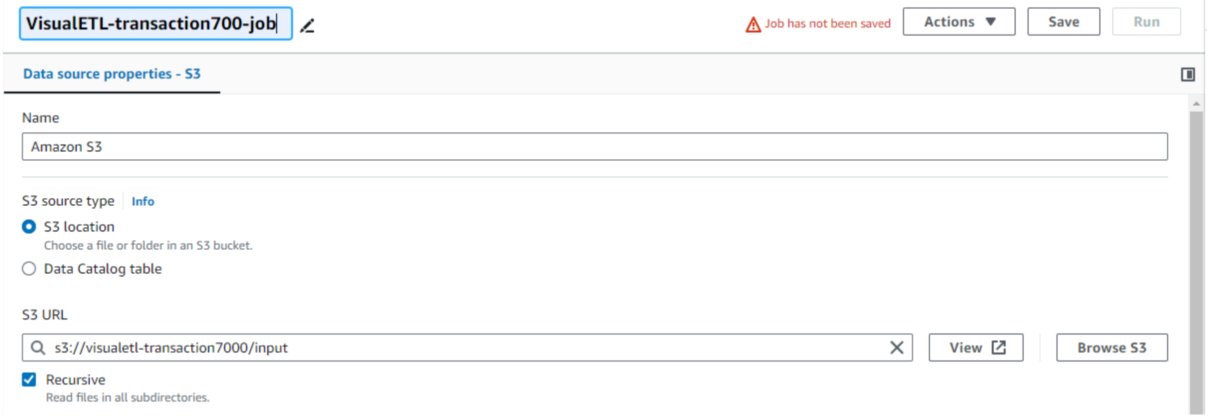
☑ Recursive
Read files in all subdirectories.

Data format: No format

- Paste the directory of your object or browse and select the bucket where you uploaded the transactions.csv file previously.

s3://{your-bucket-name}/input

DO NOT FORGET TO CHANGE THE PLACEHOLDER, {bucket_name}
DO NOT FORGET TO CHANGE THE PLACEHOLDER, {bucket_name}
DO NOT FORGET TO CHANGE THE PLACEHOLDER, {bucket_name}



Name

Amazon S3

S3 source type Info

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

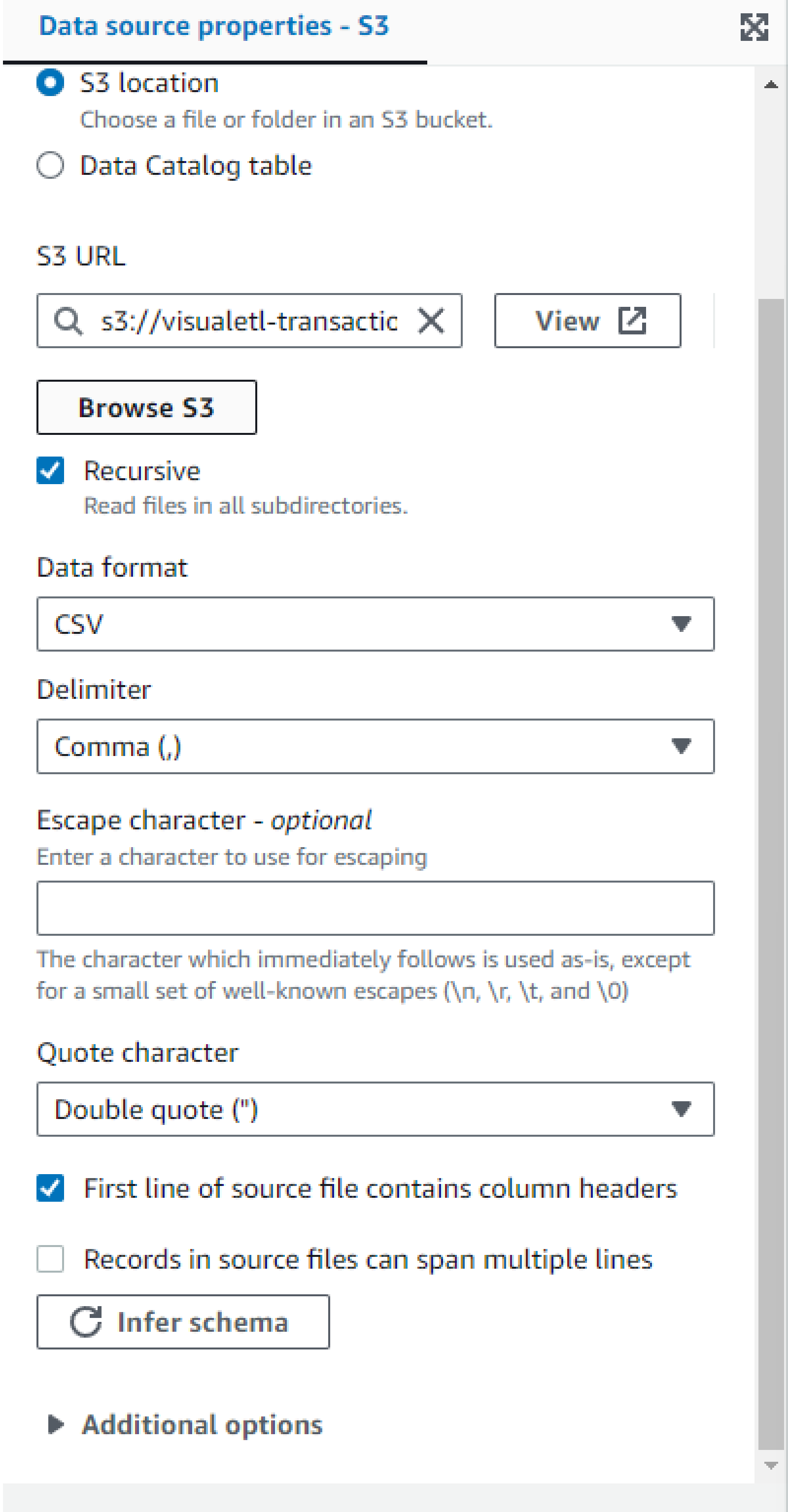
○ Data Catalog table

S3 URL

s3://visualetl-transaction7000/input

☑ Recursive
Read files in all subdirectories.

- For the **Data Forma**t, select **CSV.**



Data source properties - S3

● S3 location

Choose a file or folder in an S3 bucket.

○ Data Catalog table

S3 URL

s3://visualetl-transactic

☑ Recursive

Read files in all subdirectories.

Data format

CSV

Delimiter

Comma (,)

Escape character - *optional*

Enter a character to use for escaping

The character which immediately follows is used as-is, except for a small set of well-known escapes (\n, \r, \t, and \0)

Quote character

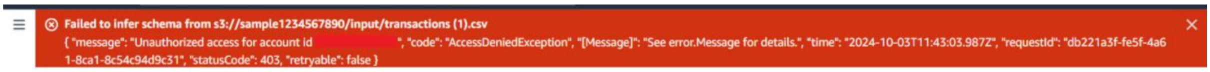
Double quote (")

☑ First line of source file contains column headers

□ Records in source files can span multiple lines

Leave the rest as default.

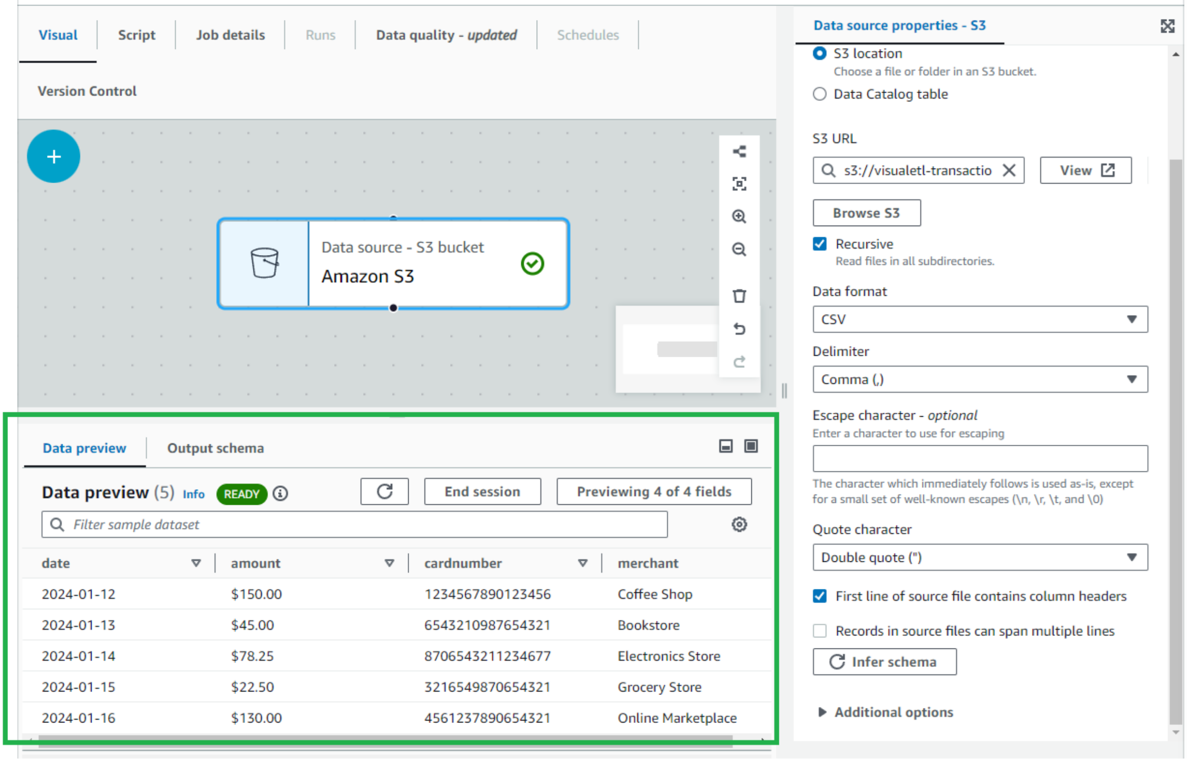
If you encounter an error similar to the example below, you can safely ignore it. You will still be able to complete this lab even if a 'failed to infer schema' error occurs.



Failed to infer schema from s3://sample1234567890/input/transactions (1).csv

```
{ "message": "Unauthorized access for account id [redacted]", "code": "AccessDeniedException", "[Message]": "See error.Message for details.", "time": "2024-10-03T11:43:03.987Z", "requestId": "db221a3f-fe5f-4a61-8ca1-8c54c94d9c31", "statusCode": 403, "retryable": false }
```

- If you wait for a while, you will see the information inside the CSV in your bucket in the Data preview.



Data preview (5)

date	amount	cardnumber	merchant
2024-01-12	$150.00	1234567890123456	Coffee Shop
2024-01-13	$45.00	6543210987654321	Bookstore
2024-01-14	$78.25	8706543211234677	Electronics Store
2024-01-15	$22.50	3216549870654321	Grocery Store
2024-01-16	$130.00	4561237890654321	Online Marketplace

Data source properties - S3

● S3 location
Choose a file or folder in an S3 bucket.
○ Data Catalog table

S3 URL: s3://visualetl-transactio

Browse S3

☑ Recursive
Read files in all subdirectories.

Data format: CSV

Delimiter: Comma (,)

Escape character - *optional*
Enter a character to use for escaping
The character which immediately follows is used as-is, except for a small set of well-known escapes (\n, \r, \t, and \0)

Quote character: Double quote (")

☑ First line of source file contains column headers

□ Records in source files can span multiple lines

Infer schema

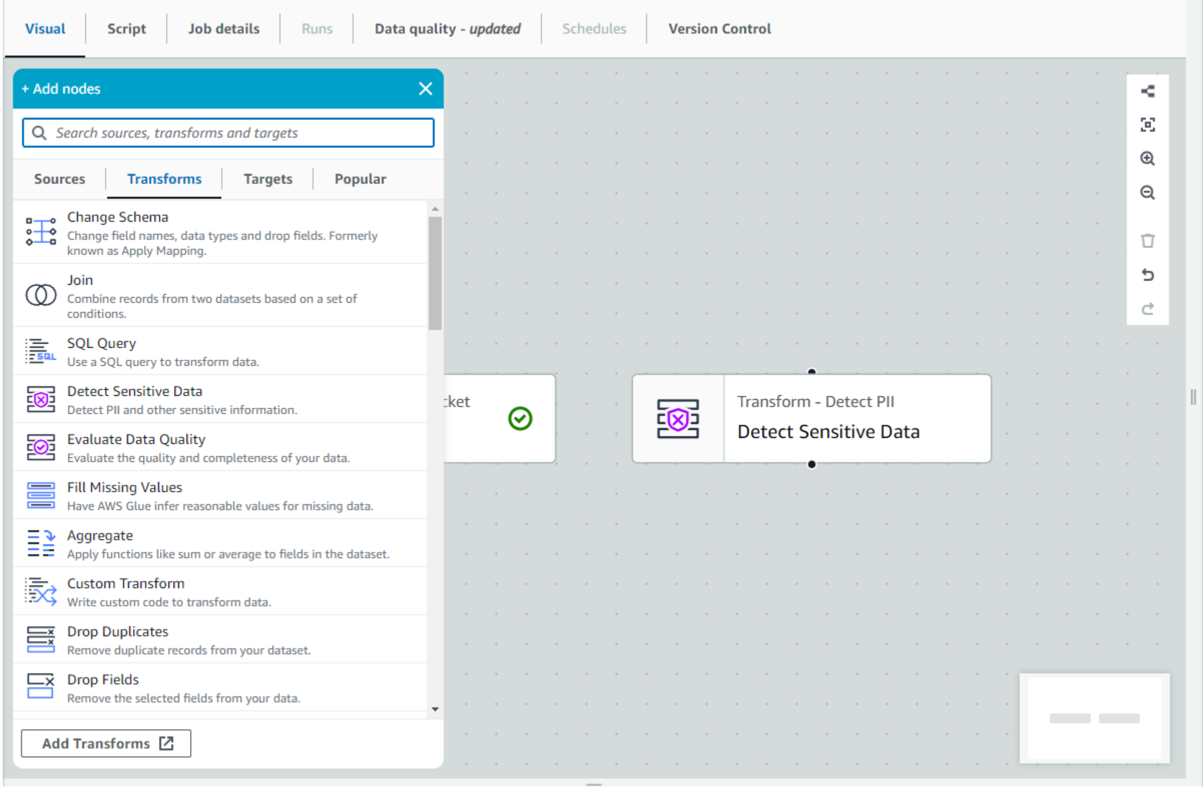
▶ Additional options

10. **Applying Detect Sensitive Data Transformations**

In this lab, we are gonna use Detect Sensitive Data Transformations first. This transformation scans the dataset for sensitive data like personal identifiable information (PII), then replace it or add a column with information on what was found and where.

- Click on the **+ sign** again (**Add Node**) and navigate to the **Transforms tab**

- Select **Detect Sensitive Data**

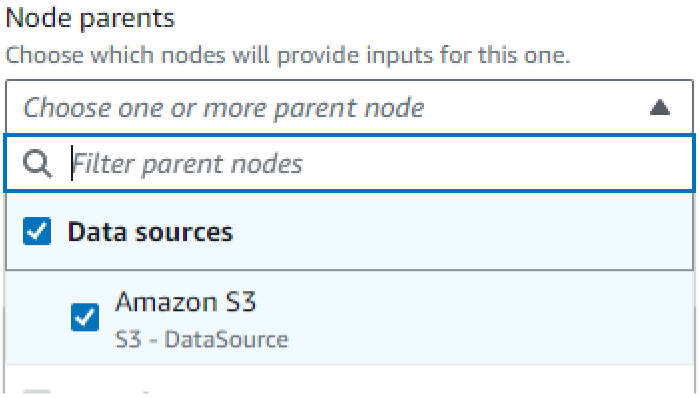


Visual
Script
Job details
Runs
Data quality - *updated*
Schedules
Version Control
+ Add nodes
Search sources, transforms and targets
Sources
Transforms
Targets
Popular
Change Schema
Change field names, data types and drop fields. Formerly known as Apply Mapping.
Join
Combine records from two datasets based on a set of conditions.
SQL Query
Use a SQL query to transform data.
Detect Sensitive Data
Detect PII and other sensitive information.
Evaluate Data Quality
Evaluate the quality and completeness of your data.
Fill Missing Values
Have AWS Glue infer reasonable values for missing data.
Aggregate
Apply functions like sum or average to fields in the dataset.
Custom Transform
Write custom code to transform data.
Drop Duplicates
Remove duplicate records from your dataset.
Drop Fields
Remove the selected fields from your data.
Add Transforms
:ket
Transform - Detect PII
Detect Sensitive Data

- Click on the **Detect Sensitive Data Transforms node**

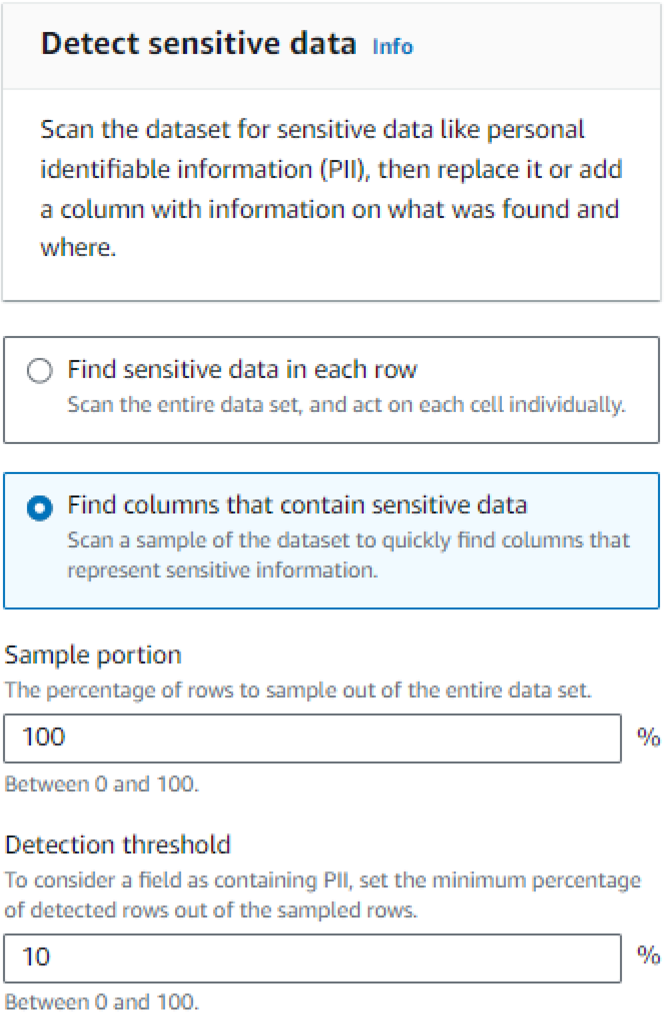
Follow the following Configurations

  - Node parents: ***Amazon S3***



Node parents
Choose which nodes will provide inputs for this one.
Choose one or more parent node
Filter parent nodes
☑ Data sources
☑ Amazon S3
S3 - DataSource

- **Detect sensitive data:** Select **Find columns that contain sensitive data.**



Detect sensitive data Info

Scan the dataset for sensitive data like personal identifiable information (PII), then replace it or add a column with information on what was found and where.

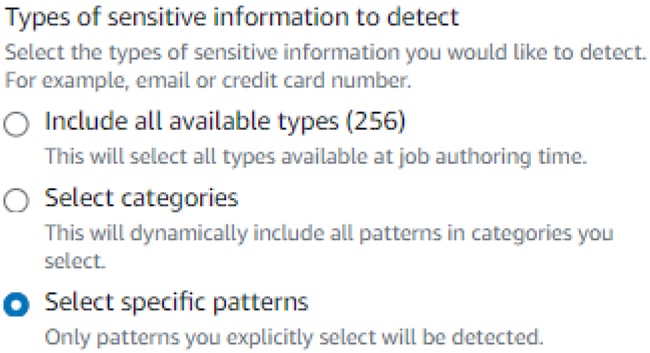
○ Find sensitive data in each row
Scan the entire data set, and act on each cell individually.

● Find columns that contain sensitive data
Scan a sample of the dataset to quickly find columns that represent sensitive information.

Sample portion
The percentage of rows to sample out of the entire data set.
100 %
Between 0 and 100.

Detection threshold
To consider a field as containing PII, set the minimum percentage of detected rows out of the sampled rows.
10 %
Between 0 and 100.

- **Types of sensitive information to detect:** Select **Select specific patterns**



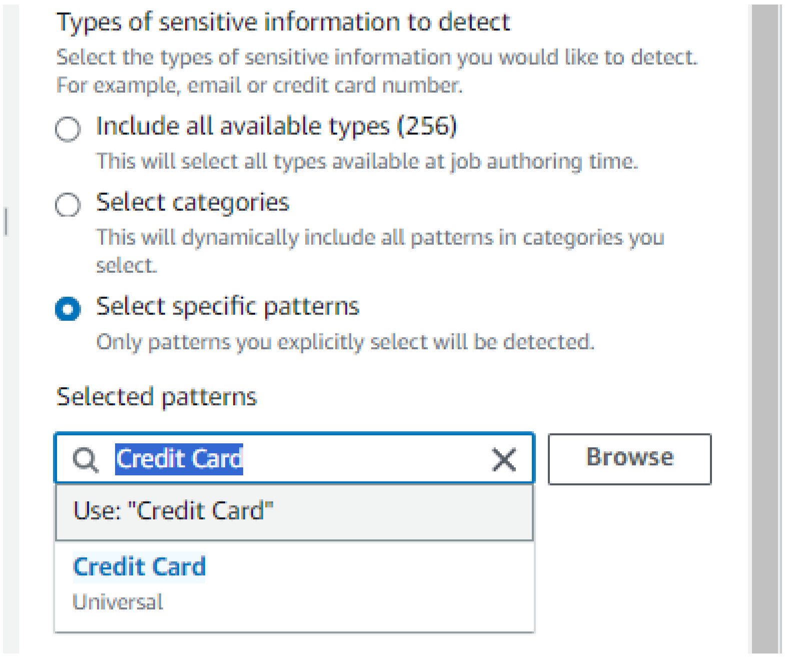
Types of sensitive information to detect
Select the types of sensitive information you would like to detect. For example, email or credit card number.

○ Include all available types (256)
This will select all types available at job authoring time.

○ Select categories
This will dynamically include all patterns in categories you select.

● Select specific patterns
Only patterns you explicitly select will be detected.

- **Select patterns: Search for Credit Card and select it**



Types of sensitive information to detect

Select the types of sensitive information you would like to detect. For example, email or credit card number.

○ Include all available types (256)
This will select all types available at job authoring time.

○ Select categories
This will dynamically include all patterns in categories you select.

● Select specific patterns
Only patterns you explicitly select will be detected.

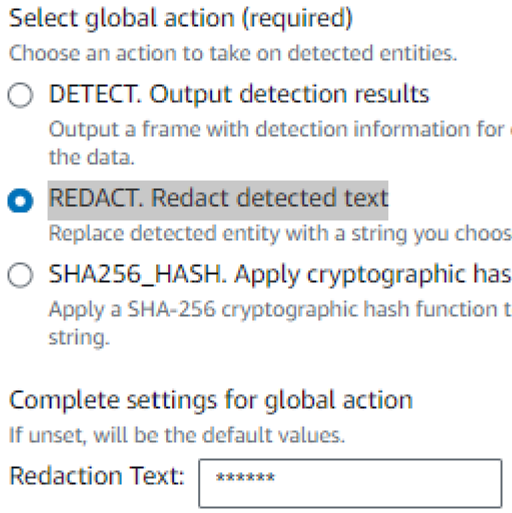
Selected patterns

Credit Card

Use: "Credit Card"

Credit Card
Universal

- Scroll down and look for:
Select global action (required): Select **REDACT. Redact detected text**



Select global action (required)

Choose an action to take on detected entities.

○ DETECT. Output detection results
Output a frame with detection information for each column in the data.

● REDACT. Redact detected text
Replace detected entity with a string you choose.

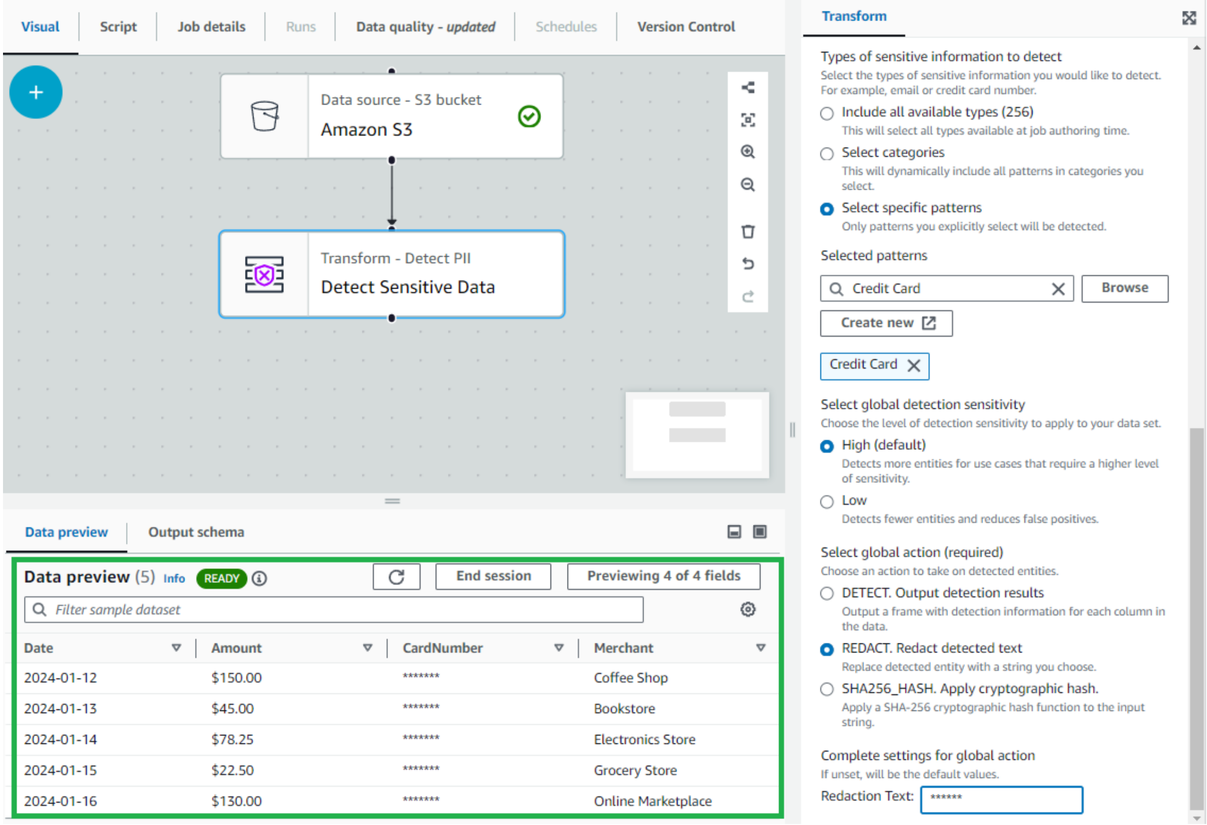
○ SHA256_HASH. Apply cryptographic hash.
Apply a SHA-256 cryptographic hash function to the input string.

Complete settings for global action

If unset, will be the default values.

Redaction Text: ******

- In the **Data Preview,** you will see that the CardNumber Column has been redacted.



Data preview (5) Info READY

End session | Previewing 4 of 4 fields

Filter sample dataset

Date	Amount	CardNumber	Merchant
2024-01-12	$150.00	*******	Coffee Shop
2024-01-13	$45.00	*******	Bookstore
2024-01-14	$78.25	*******	Electronics Store
2024-01-15	$22.50	*******	Grocery Store
2024-01-16	$130.00	*******	Online Marketplace

Types of sensitive information to detect
Select the types of sensitive information you would like to detect. For example, email or credit card number.

○ Include all available types (256)
This will select all types available at job authoring time.

○ Select categories
This will dynamically include all patterns in categories you select.

● Select specific patterns
Only patterns you explicitly select will be detected.

Selected patterns

Credit Card ✕ | Browse

Create new

Credit Card ✕

Select global detection sensitivity
Choose the level of detection sensitivity to apply to your data set.

● High (default)
Detects more entities for use cases that require a higher level of sensitivity.

○ Low
Detects fewer entities and reduces false positives.

Select global action (required)
Choose an action to take on detected entities.

○ DETECT. Output detection results
Output a frame with detection information for each column in the data.

● REDACT. Redact detected text
Replace detected entity with a string you choose.

○ SHA256_HASH. Apply cryptographic hash.
Apply a SHA-256 cryptographic hash function to the input string.

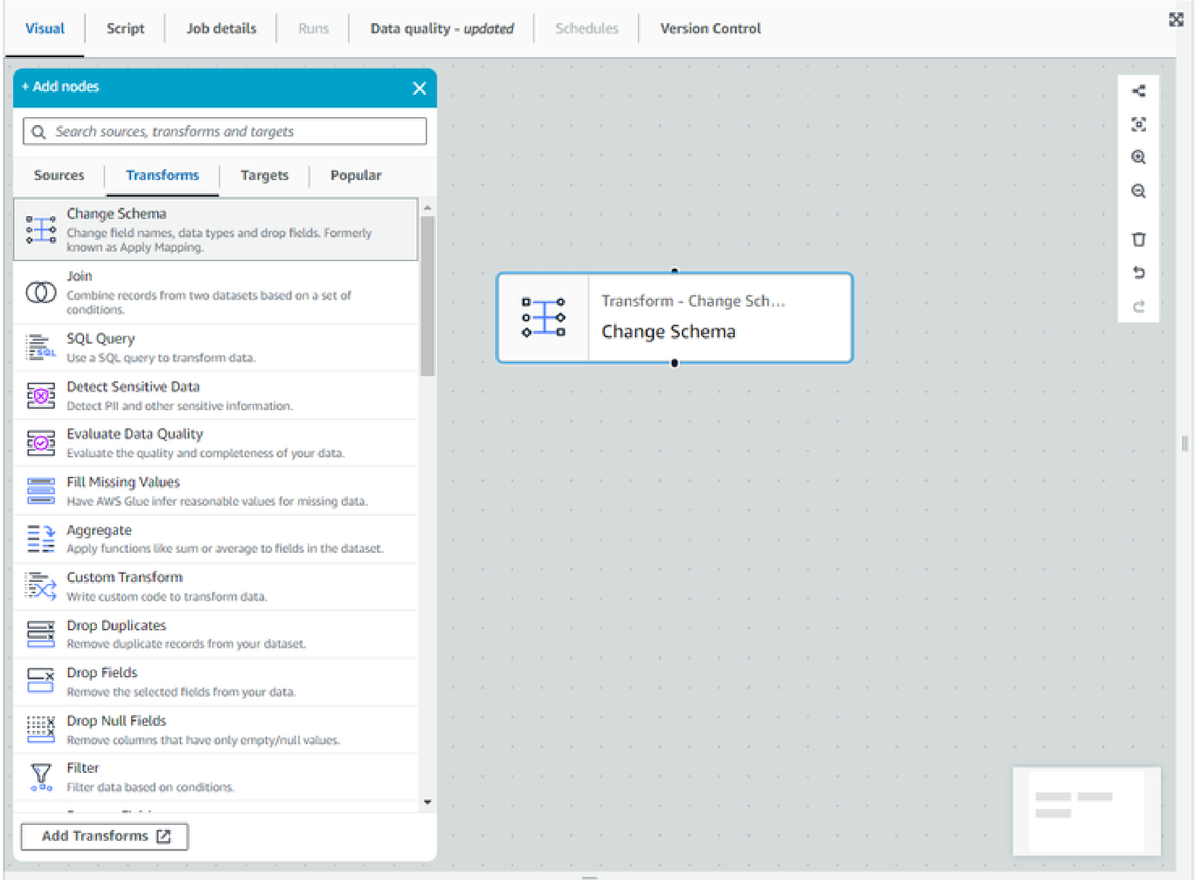
Complete settings for global action
If unset, will be the default values.

Redaction Text: ******

11. **Applying Change Schema Transformations**
Next, let's do a simple Change Schema transformation.

- Click on the **+ sign** again (**Add Node**) and navigate to the **Transforms tab**

- Select **Change Schema**



Change Schema
Change field names, data types and drop fields. Formerly known as Apply Mapping.

Join
Combine records from two datasets based on a set of conditions.

SQL Query
Use a SQL query to transform data.

Detect Sensitive Data
Detect PII and other sensitive information.

Evaluate Data Quality
Evaluate the quality and completeness of your data.

Fill Missing Values
Have AWS Glue infer reasonable values for missing data.

Aggregate
Apply functions like sum or average to fields in the dataset.

Custom Transform
Write custom code to transform data.

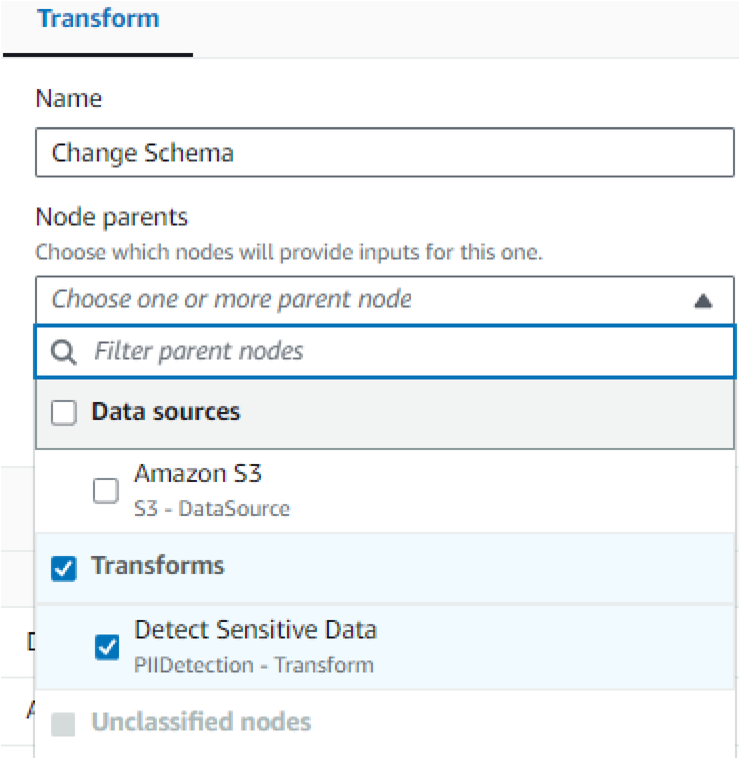
Drop Duplicates
Remove duplicate records from your dataset.

Drop Fields
Remove the selected fields from your data.

Drop Null Fields
Remove columns that have only empty/null values.

Filter
Filter data based on conditions.

- Click on the **Change Schema Transforms node**
  Follow the following Configurations
  - Node parents: **Detect Sensitive Data**



Name

Change Schema

Node parents

Choose which nodes will provide inputs for this one.

Choose one or more parent node

Filter parent nodes

□ **Data sources**

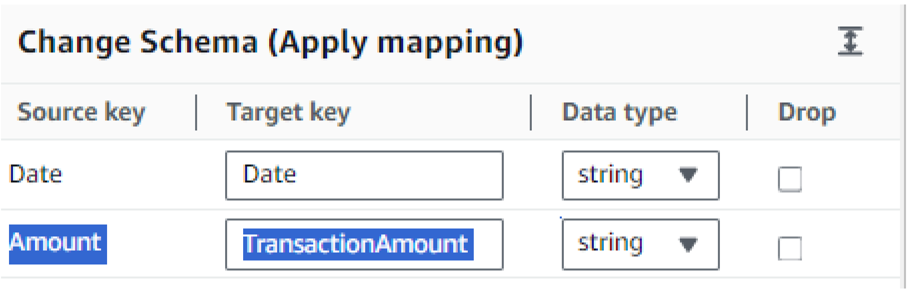
□ Amazon S3
S3 - DataSource

☑ **Transforms**

☑ Detect Sensitive Data
PIIDetection - Transform

□ **Unclassified nodes**

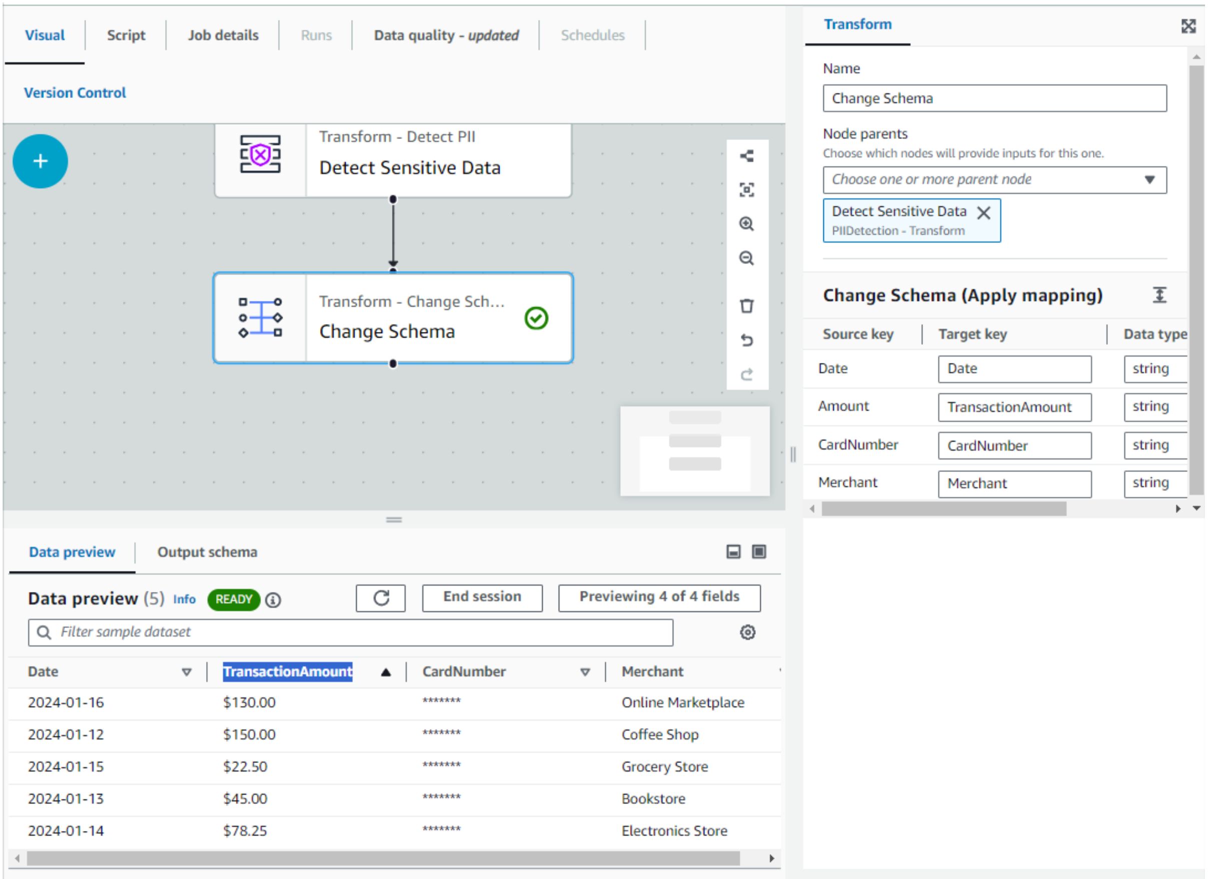
- **Change Schema (Apply mapping)**:
    - Under the Source key, look for **Amount** and change it to **TransactionAmount** under the Target Key



Change Schema (Apply mapping)

Source key	Target key	Data type	Drop
Date	Date	string	□
Amount	TransactionAmount	string	□

- You will see the changes made in the **Data preview**



Name: Change Schema

Node parents: Choose which nodes will provide inputs for this one.

Choose one or more parent node

Detect Sensitive Data ✕ PIIDetection - Transform

Change Schema (Apply mapping)

Source key	Target key	Data type
Date	Date	string
Amount	TransactionAmount	string
CardNumber	CardNumber	string
Merchant	Merchant	string

Data preview (5) Info READY

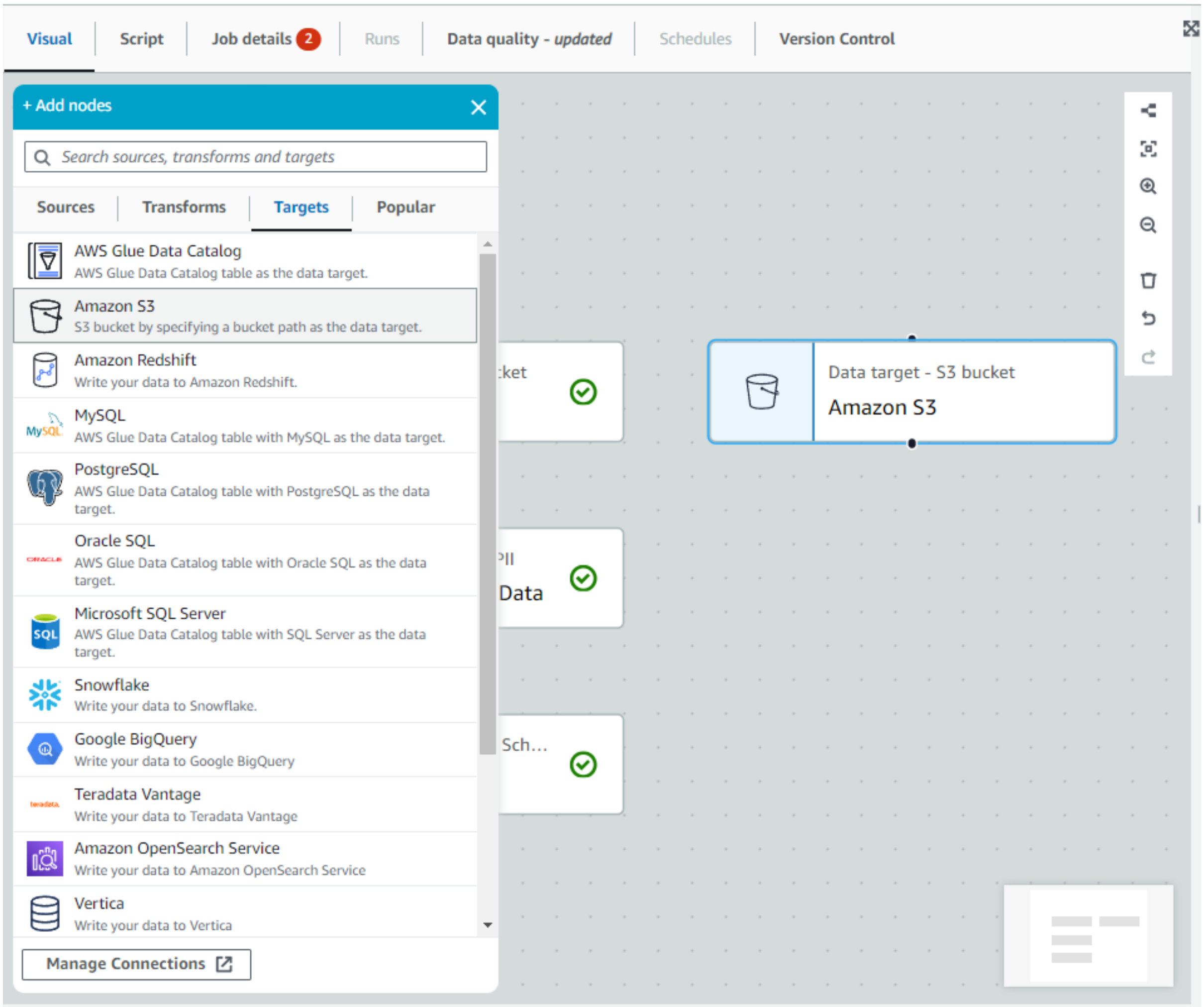
Filter sample dataset

Date	TransactionAmount	CardNumber	Merchant
2024-01-16	$130.00	*******	Online Marketplace
2024-01-12	$150.00	*******	Coffee Shop
2024-01-15	$22.50	*******	Grocery Store
2024-01-13	$45.00	*******	Bookstore
2024-01-14	$78.25	*******	Electronics Store

12. Lastly, lets **Add the Target Node** for our ETL Job.

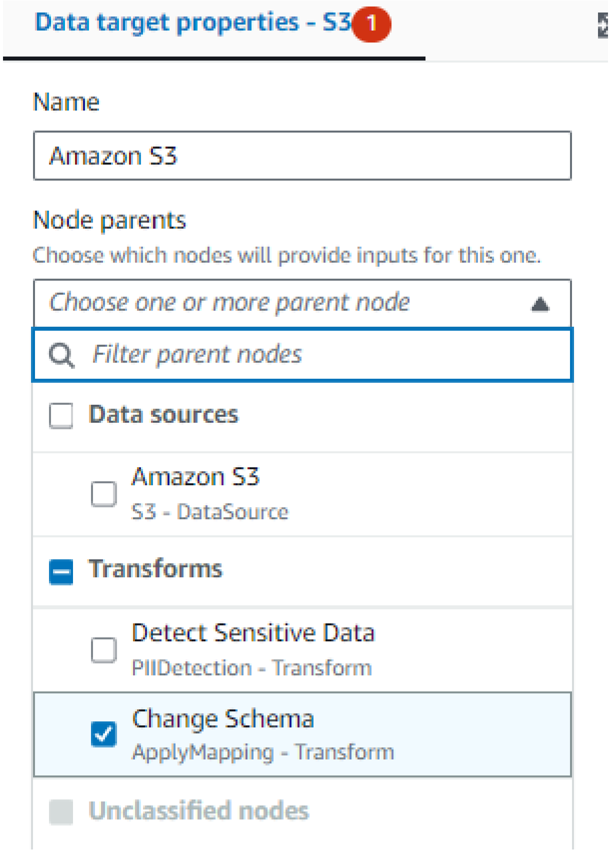
- Click on the **+ sign** again (**Add Node**) and navigate to the **Targets tab**

- Select **Amazon S3**


Visual
Script
Job details 2
Runs
Data quality - *updated*
Schedules
Version Control
+ Add nodes
Search sources, transforms and targets
Sources
Transforms
Targets
Popular
AWS Glue Data Catalog
AWS Glue Data Catalog table as the data target.
Amazon S3
S3 bucket by specifying a bucket path as the data target.
Amazon Redshift
Write your data to Amazon Redshift.
MySQL
AWS Glue Data Catalog table with MySQL as the data target.
PostgreSQL
AWS Glue Data Catalog table with PostgreSQL as the data target.
Oracle SQL
AWS Glue Data Catalog table with Oracle SQL as the data target.
Microsoft SQL Server
AWS Glue Data Catalog table with SQL Server as the data target.
Snowflake
Write your data to Snowflake.
Google BigQuery
Write your data to Google BigQuery
Teradata Vantage
Write your data to Teradata Vantage
Amazon OpenSearch Service
Write your data to Amazon OpenSearch Service
Vertica
Write your data to Vertica
Manage Connections
:ket
PII
Data
Sch…
Data target - S3 bucket
Amazon S3

- Click on the Amazon S3 **Target node**
  Follow the following Configurations

- Node parents: **Change Schema**



Data target properties - S3 1

Name

Amazon S3

Node parents

Choose which nodes will provide inputs for this one.

Choose one or more parent node

Filter parent nodes

□ Data sources

□ Amazon S3
S3 - DataSource

Transforms

□ Detect Sensitive Data
PIIDetection - Transform

☑ Change Schema
ApplyMapping - Transform

□ Unclassified nodes

- Format: **Parquet**
  Parquet is an open-source, column-oriented data file format that is designed for efficient data storage and retrieval.

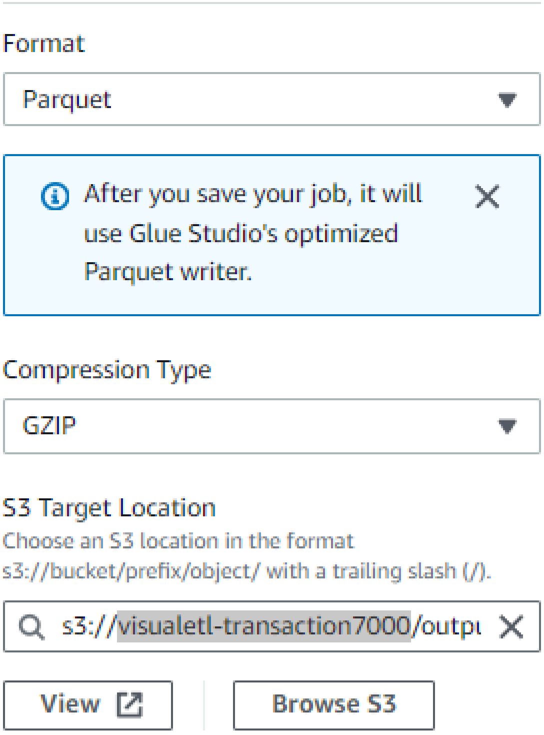
- *compression Type:* ***GZIP***

- **S3 Target Location:**

s3://{your-bucket-name}/output/

DO NOT FORGET TO CHANGE THE PLACEHOLDER, {bucket_name}
DO NOT FORGET TO CHANGE THE PLACEHOLDER, {bucket_name}

DO NOT FORGET TO CHANGE THE PLACEHOLDER, {bucket_name}



Format

Parquet

After you save your job, it will use Glue Studio's optimized Parquet writer.

Compression Type

GZIP

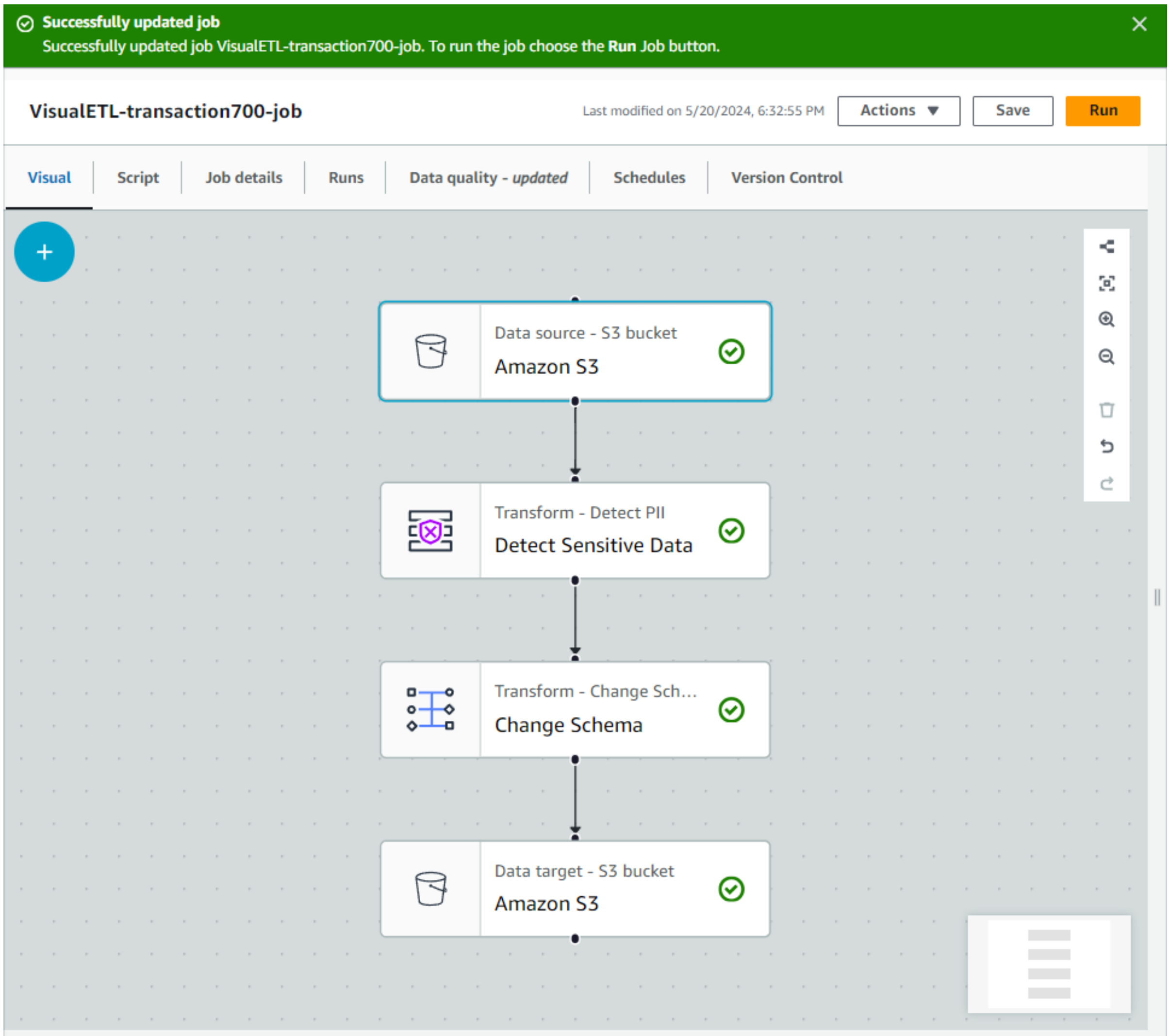
S3 Target Location

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

s3://visualetl-transaction7000/outpu

-
  - Leave the rest as Default

13. Click on **Save** in the upper right corner of the interface to save your first Visual ETL job.



Successfully updated job

Successfully updated job VisualETL-transaction700-job. To run the job choose the **Run** Job button.

NOTE: *After saving the Job, you won't be able to run it. This lab simulates creating a visual ETL job and is intended for demonstration purposes only.*

That's it congratulations on completing this lab! Through this hands-on experience, you've set up data sources, applied sensitive data transformations, and adjusted data schemas using AWS Glue's visual interface. The skills you've honed today are invaluable for efficiently managing and manipulating large datasets in real-world scenarios.

This lab has demonstrated how AWS Glue's Visual ETL can streamline your data transformation workflows, making the process more intuitive and less error-prone compared to hard coding your ETL job. As you continue to explore AWS Glue, leverage these advantages to enhance your data integration and analytics capabilities further.

One last thing! It is a good practice to clean up the resources created during this lab. Not only will it make you a better professional, but you will also become a more organized person. Happy learning!