Guided Lab: Introduction to AWS Glue Using Python Shell

Description

In this guided lab, we'll work with a CSV file containing transaction records and use Python Shell to redact sensitive information.

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.

CSV stands for **Comma Separated Values**. It is a file format used to store tabular data, such as in spreadsheets or databases. In a CSV file, each line represents a record, and the fields in each record are separated by commas. The first line of the file typically contains the names of the fields. CSV files are commonly used for data exchange between different applications and are supported by many software programs.

Prerequisites

This lab assumes you have basic knowledge of Python programming 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.

Objectives

In this lab, you will:

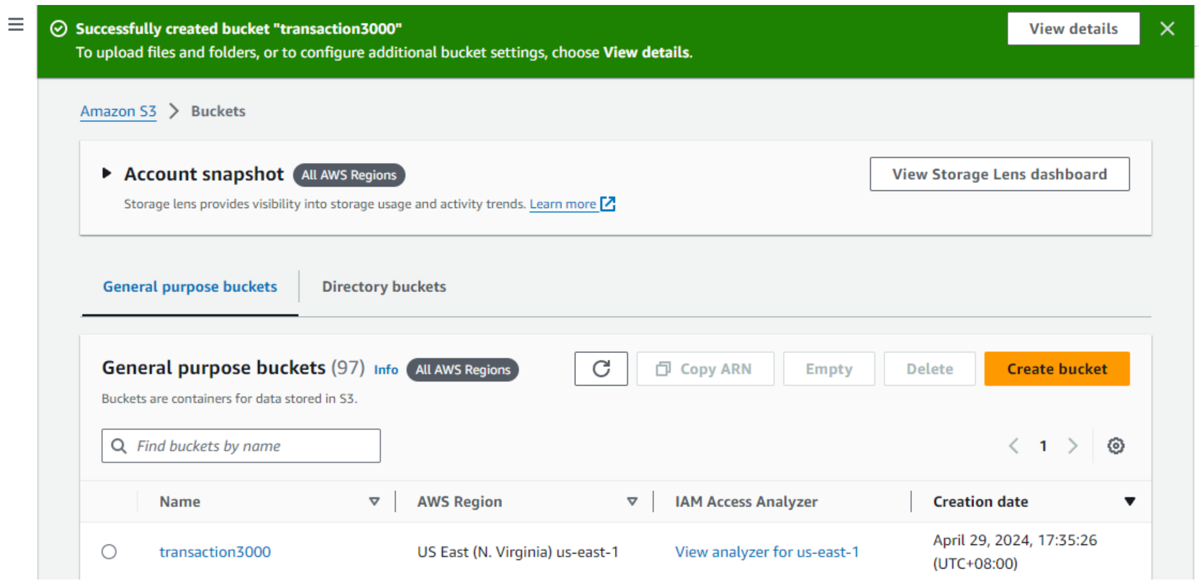
- Learn how to use AWS Glue Python Shell to process and transform data stored in an S3 bucket using Python scripts.

Subscribe to access AWS PlayCloud Labs

Lab Steps

Prepare Your Environment

1. **Log into the AWS Management Console**: Navigate to the console and sign in.
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**
    -



Successfully created bucket "transaction3000"
To upload files and folders, or to configure additional bucket settings, choose **View details**.

Amazon S3 > Buckets

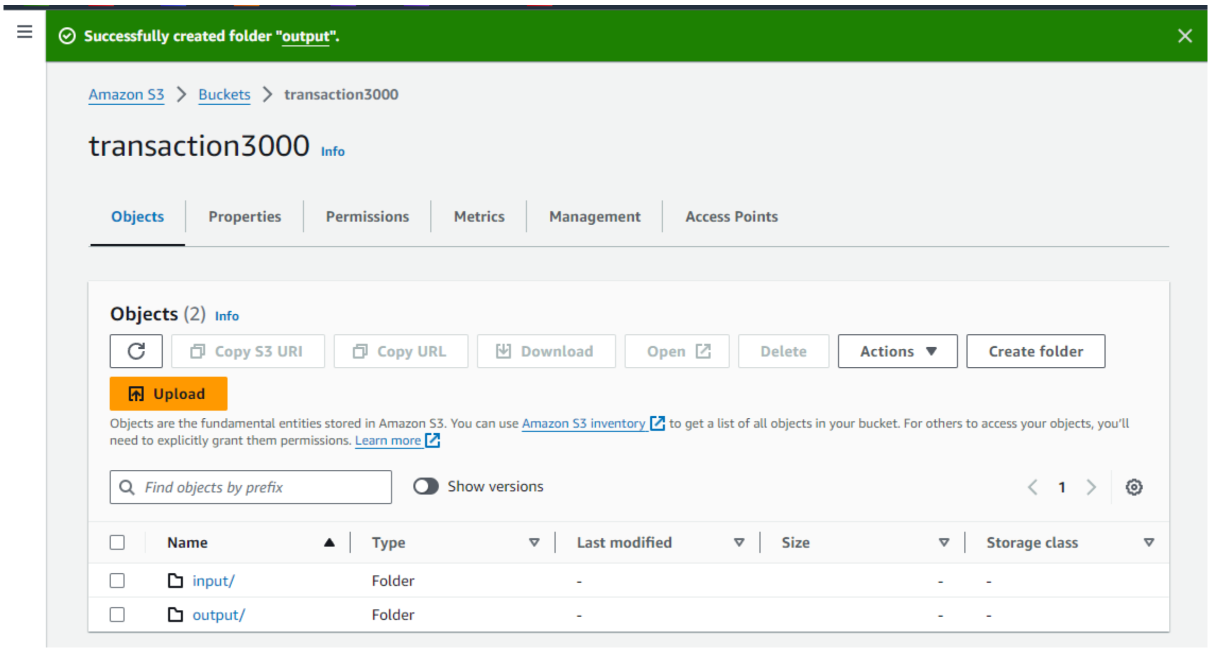
Account snapshot — All AWS Regions
Storage lens provides visibility into storage usage and activity trends. Learn more

General purpose buckets (97) — All AWS Regions
Buckets are containers for data stored in S3.

Name	AWS Region	IAM Access Analyzer	Creation date
transaction3000	US East (N. Virginia) us-east-1	View analyzer for us-east-1	April 29, 2024, 17:35:26 (UTC+08:00)

3. **Create a two folders:**

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



Successfully created folder "output".

Amazon S3 > Buckets > transaction3000

transaction3000

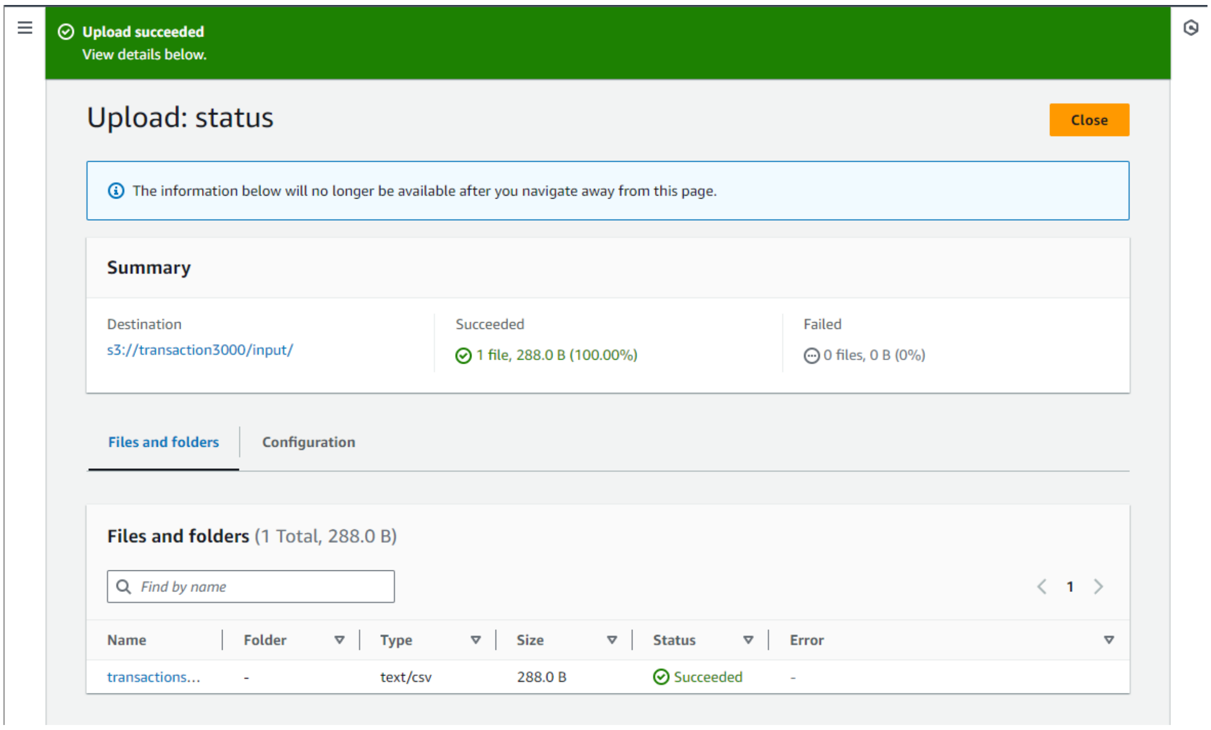
Objects (2)
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

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 and upload the file.

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


Upload succeeded
View details below.

Upload: status

Close

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

Summary

Destination	Succeeded	Failed
s3://transaction3000/input/	1 file, 288.0 B (100.00%)	0 files, 0 B (0%)

Files and folders | Configuration

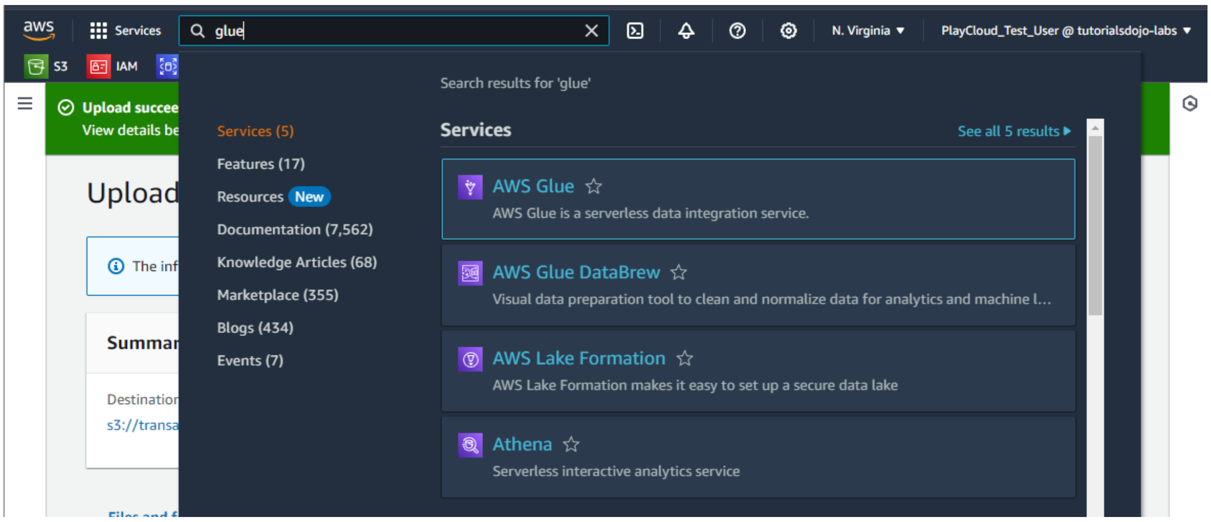
Files and folders (1 Total, 288.0 B)

Find by name

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

Create Your First Python Shell Job in AWS Glue

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



Search results for 'glue'

Services (5)
Features (17)
Resources New
Documentation (7,562)
Knowledge Articles (68)
Marketplace (355)
Blogs (434)
Events (7)

Services
See all 5 results

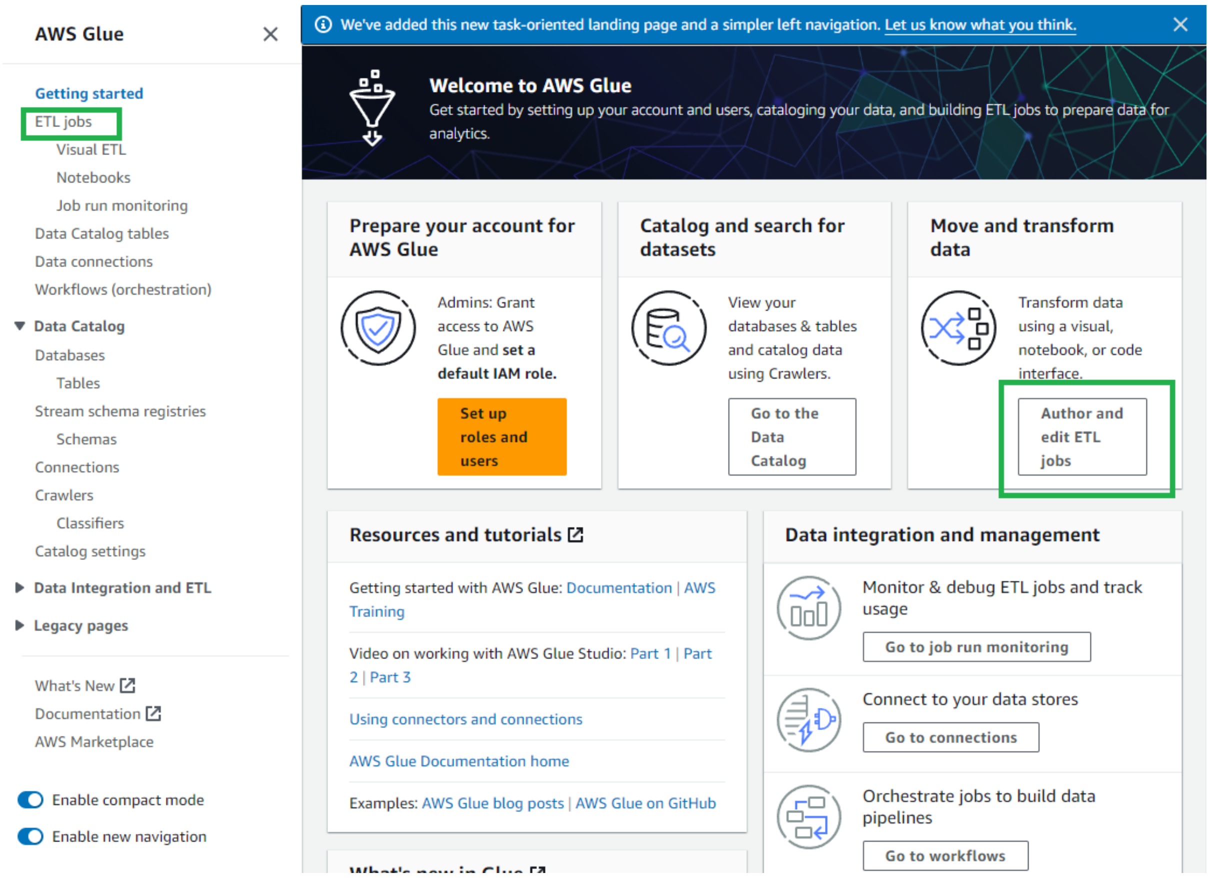
AWS Glue
AWS Glue is a serverless data integration service.

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

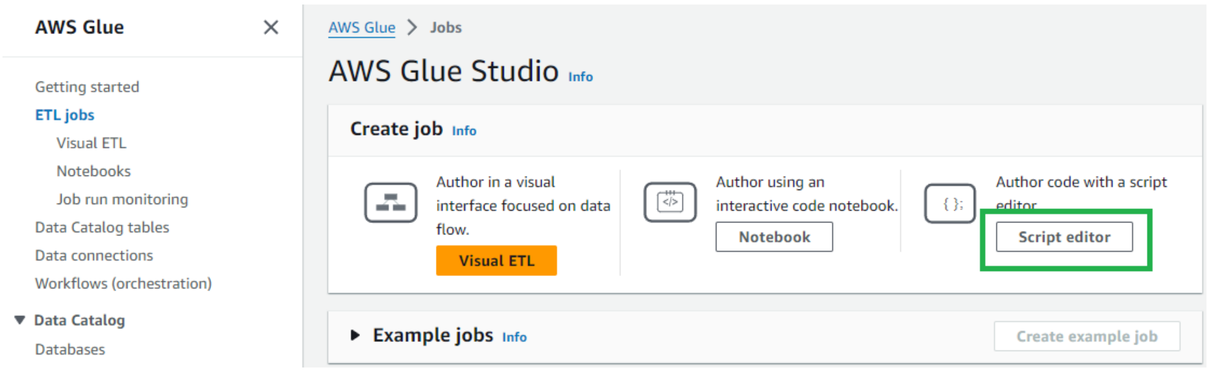
Connect to your data stores

Go to connections

Orchestrate jobs to build data pipelines

Go to workflows

3. Click **Script Editor**


AWS Glue

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

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

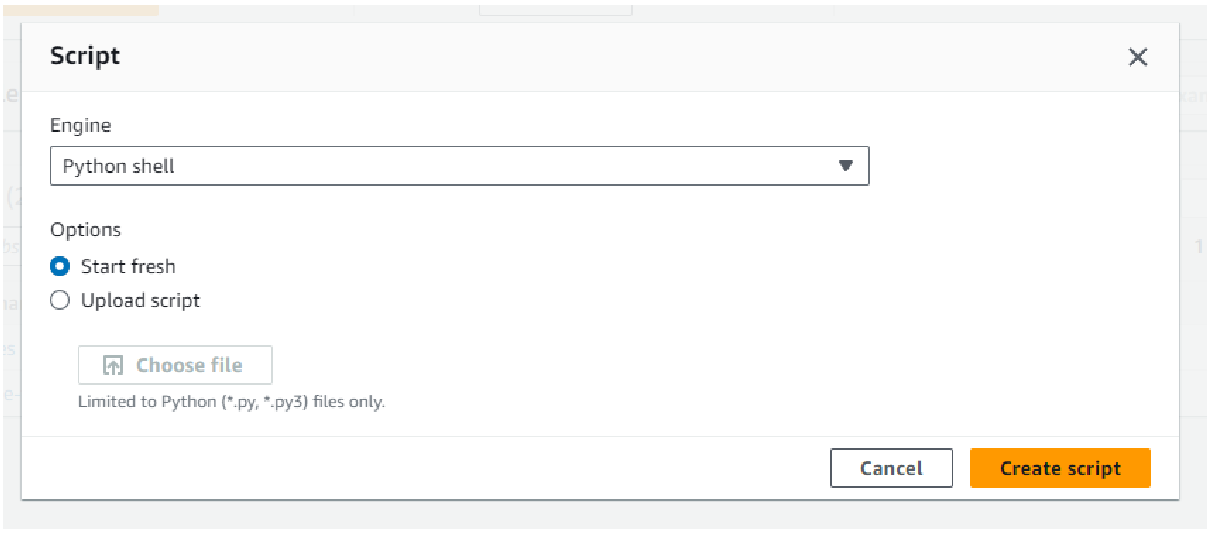
Author code with a script editor

Script editor

Example jobs Info

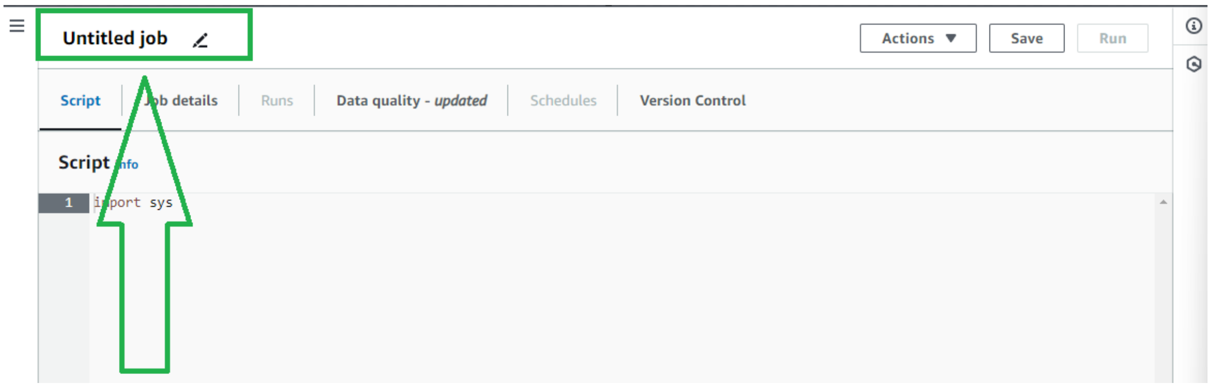
Create example job

4. Under Engine select **Python shell** and select **Start fresh** as the Options



Click **Create script**

5. Provide a name for the job, e.g., “DataProcessingJob”.



```
import sys
```

6. In the script section, paste the provided Python script.

```
import sys

import pandas as pd

from awsglue.utils import getResolvedOptions


# Fetch job parameters

args = getResolvedOptions(sys.argv, [])

input_path = "s3://{bucket_name}/input/transactions.csv"

output_path = "s3://{bucket_name}/output/"


# Load data from S3

transactions = pd.read_csv(input_path)
```

```
# Function to redact card numbers
def redact_card_number(card_number):
    # Convert the card number to a string, just in case it's in numeric format
    card_number = str(card_number)
    # Replace all but the last 4 digits with asterisks
    return '*' * (len(card_number) - 4) + card_number[-4:]

# Apply the redaction function to the CardNumber column
transactions['CardNumber'] = transactions['CardNumber'].apply(redact_card_number)

# Save processed data
transactions.to_csv(output_path + "transactions_redacted.csv", index=False)

print("Data processing complete. Card numbers redacted.")
```

DO NOT FORGET TO CHANGE THE PLACEHOLDER, {bucket_name}

- CODE EXPLANATION:
  Importing Libraries:
    - import sys: This module provides access to some variables used or maintained by the Python interpreter and to functions that interact strongly with the interpreter.
    - import pandas as pd: This imports the pandas library, which is a powerful data manipulation tool built on top of the Python programming language.
    - from awsglue.utils import getResolvedOptions: This imports a function getResolvedOptions from the awsglue.utils module. This function is used to fetch job parameters when running an AWS Glue job.

Fetching Job Parameters:

- args = getResolvedOptions(sys.argv, []): This line fetches the job parameters using the getResolvedOptions function. It retrieves the parameters passed to the Glue job at runtime and stores them in the args dictionary.

Defining Input and Output Paths:

- input_path = "s3://{bucket_name}/input/transactions.csv": This specifies the S3 path where the input CSV file (transactions.csv) is located.
- output_path = "s3://{bucket_name}/output/": This specifies the S3 path where the processed data will be saved.

Loading Data from S3:

- transactions = pd.read_csv(input_path): This line uses the pandas read_csv function to load the data from the CSV file located at input_path into a pandas DataFrame named transactions.

Redacting Card Numbers:

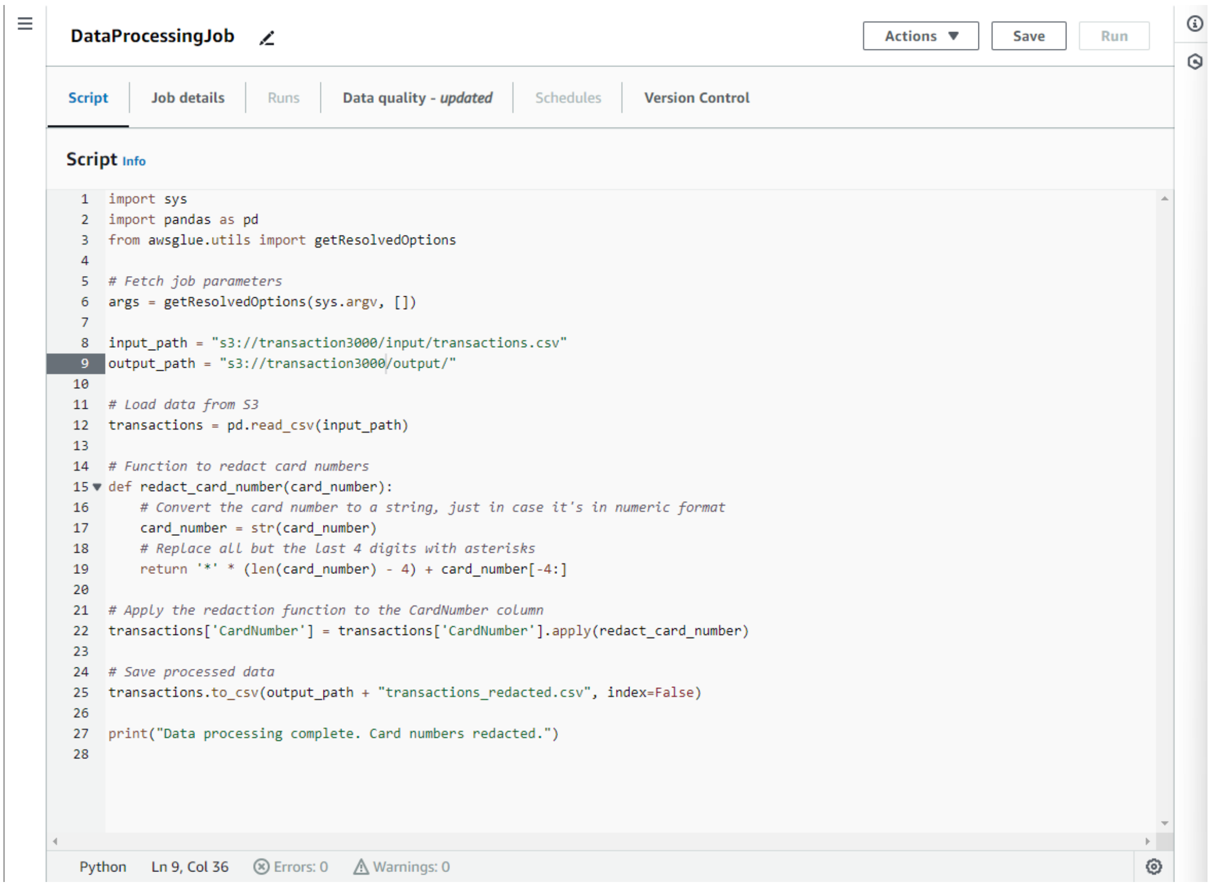
- def redact_card_number(card_number): This defines a function named redact_card_number that takes a card_number as input.
- card_number = str(card_number): This line converts the card_number to a string, ensuring it's in string format.
- return '*' * (len(card_number) - 4) + card_number[-4:]: This line replaces all characters of the card_number except the last four digits with asterisks and returns the redacted card number.
- transactions['CardNumber'] = transactions['CardNumber'].apply(redact_card_number): This applies the redact_card_number function to the 'CardNumber' column of the transactions DataFrame, replacing the original card numbers with the redacted versions.

Saving Processed Data:

- transactions.to_csv(output_path + "transactions_redacted.csv", index=False): This line saves the processed data (with redacted card numbers) to a new CSV file named transactions_redacted.csv (You can changed the name as desired) at the specified output_path.

Printing Message:

- print("Data processing complete. Card numbers redacted."): This line prints a message indicating that the data processing is complete and the card numbers have been redacted.

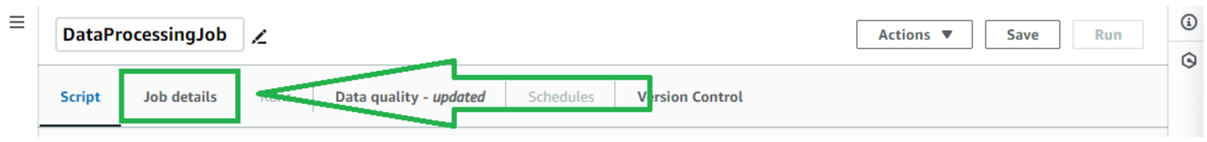


```python
import sys
import pandas as pd
from awsglue.utils import getResolvedOptions

# Fetch job parameters
args = getResolvedOptions(sys.argv, [])

input_path = "s3://transaction3000/input/transactions.csv"
output_path = "s3://transaction3000/output/"

# Load data from S3
transactions = pd.read_csv(input_path)

# Function to redact card numbers
def redact_card_number(card_number):
    # Convert the card number to a string, just in case it's in numeric format
    card_number = str(card_number)
    # Replace all but the last 4 digits with asterisks
    return '*' * (len(card_number) - 4) + card_number[-4:]

# Apply the redaction function to the CardNumber column
transactions['CardNumber'] = transactions['CardNumber'].apply(redact_card_number)

# Save processed data
transactions.to_csv(output_path + "transactions_redacted.csv", index=False)

print("Data processing complete. Card numbers redacted.")
```

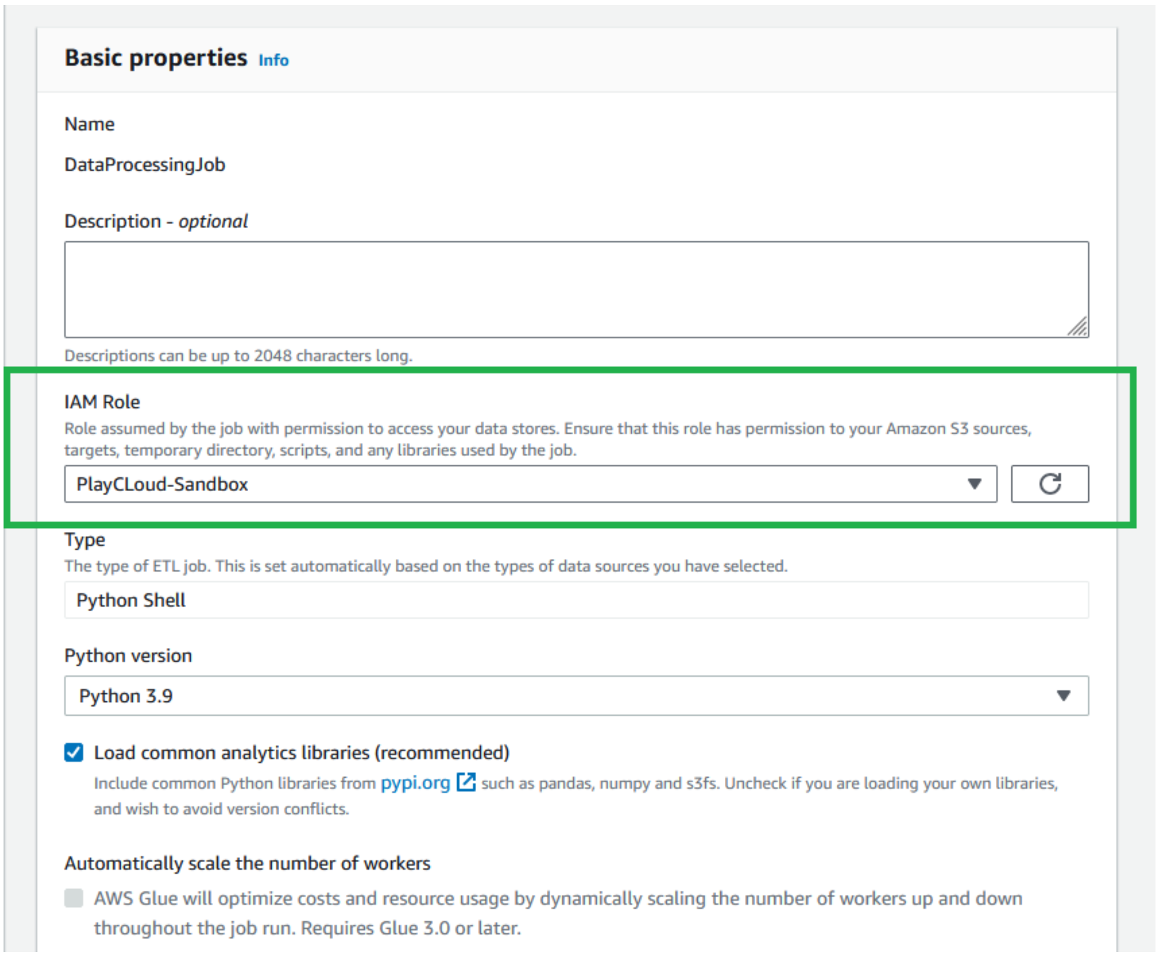
7. Review the script to ensure it performs data redaction as required.

8. Set the **IAM role** you created earlier.

- Click on Job Details



- Under IAM role, Select **PlayCLoud-Sandbox.**



Basic properties Info

Name

DataProcessingJob

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

Type

The type of ETL job. This is set automatically based on the types of data sources you have selected.

Python Shell

Python version

Python 3.9

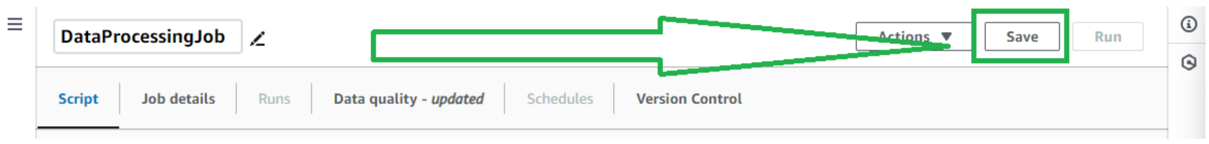
✓ Load common analytics libraries (recommended)

Include common Python libraries from pypi.org such as pandas, numpy and s3fs. Uncheck if you are loading your own libraries, and wish to avoid version conflicts.

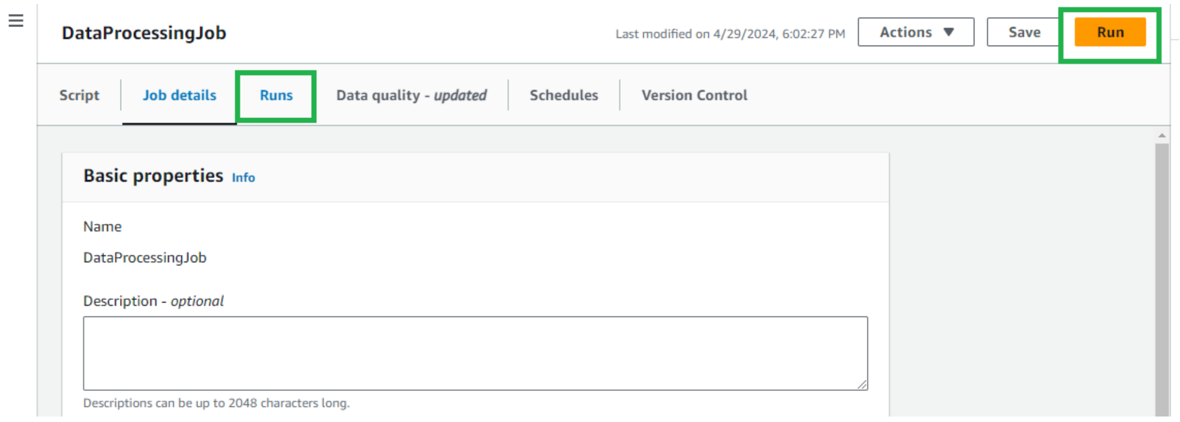
Automatically scale the number of workers

AWS Glue will optimize costs and resource usage by dynamically scaling the number of workers up and down throughout the job run. Requires Glue 3.0 or later.

- Click **Save**



9. You should be able to see the **Runs Tab** now and the **Run** button in the upper right corner.



Basic properties Info

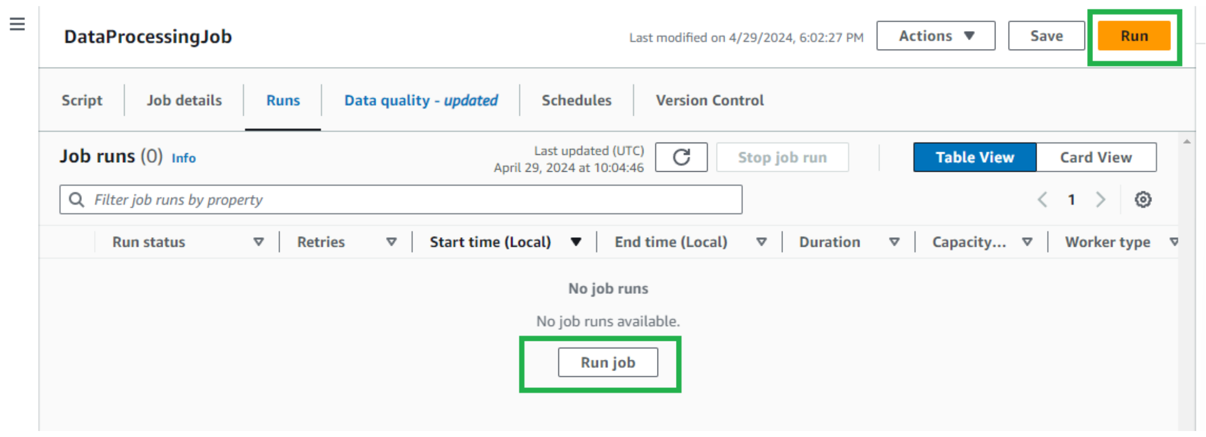
Name

DataProcessingJob

Description - *optional*

Descriptions can be up to 2048 characters long.

10. Go to the **Runs Tab,** Click either **Run job** or **Run** in the upper right corner.

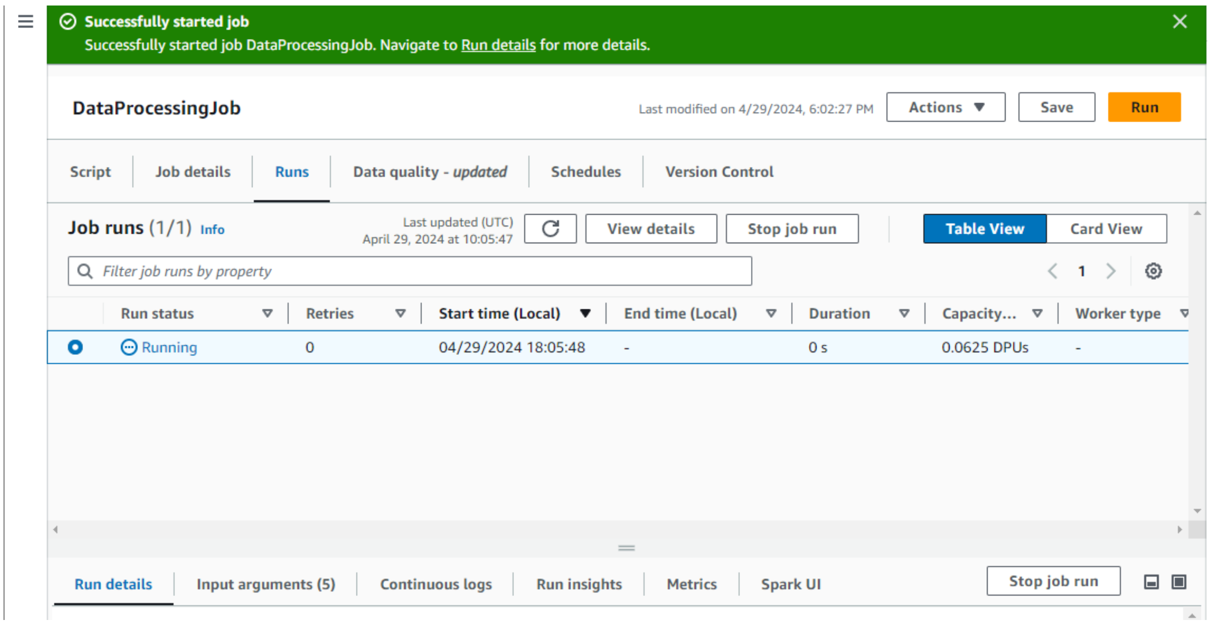


Run status	Retries	Start time (Local)	End time (Local)	Duration	Capacity...	Worker type

No job runs

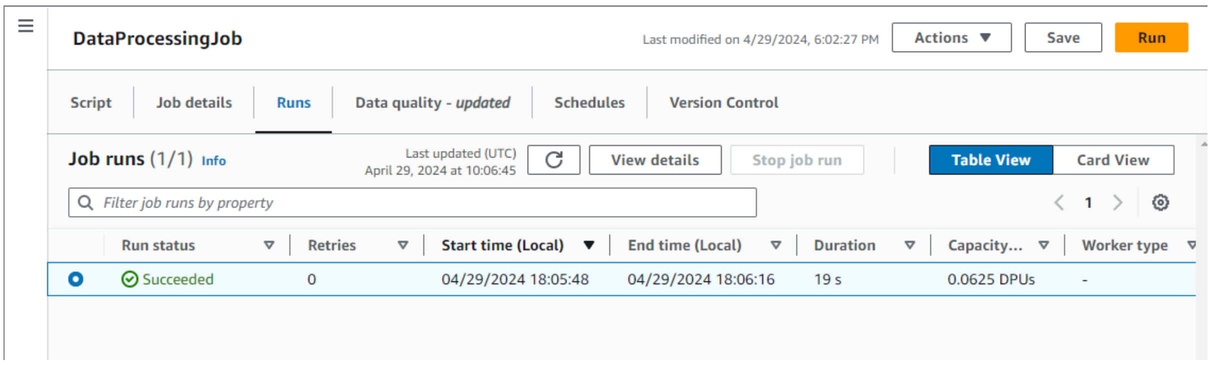
No job runs available.

11. You will be prompted with a **Successfully Started Job.**


Successfully started job
Successfully started job DataProcessingJob. Navigate to Run details for more details.

Run status	Retries	Start time (Local)	End time (Local)	Duration	Capacity...	Worker type
Running	0	04/29/2024 18:05:48	-	0 s	0.0625 DPUs	-

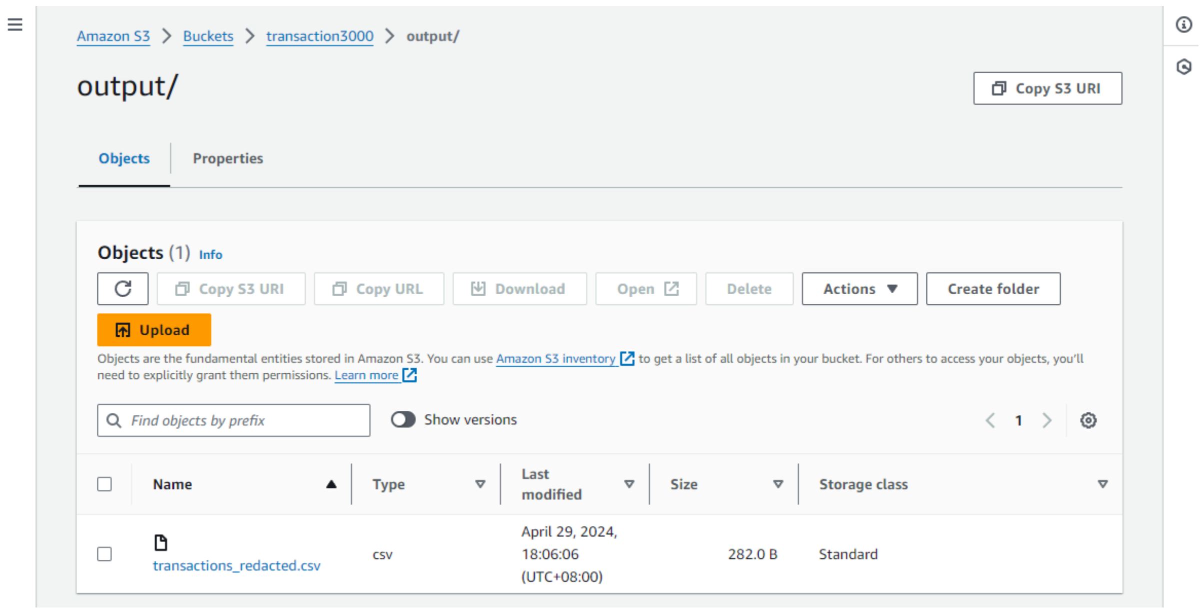
12. Wait for it to finished and you should see similar output like the image below.



Run status	Retries	Start time (Local)	End time (Local)	Duration	Capacity...	Worker type
Succeeded	0	04/29/2024 18:05:48	04/29/2024 18:06:16	19 s	0.0625 DPUs	-

13. Now, navigate back to your S3 bucket's **output/ folder**

- You should see a csv file named **transactions_redacted.csv**
  *Remember, in our code we intentionally named this **transactions_redacted.csv.** You can changed this as desired in the code editor.*


Amazon S3 > Buckets > transaction3000 > output/

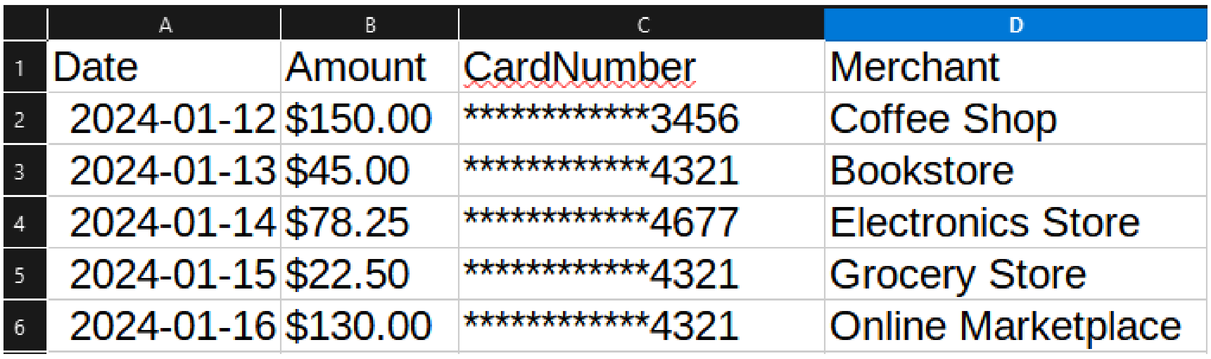
output/

Objects (1)

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
transactions_redacted.csv	csv	April 29, 2024, 18:06:06 (UTC+08:00)	282.0 B	Standard

14. Download the object and open the CSV file using Excel or any other spreadsheet application to visualize the data. When you are done you can continue and upload the file.



	A	B	C	D
1	Date	Amount	CardNumber	Merchant
2	2024-01-12	$150.00	************3456	Coffee Shop
3	2024-01-13	$45.00	************4321	Bookstore
4	2024-01-14	$78.25	************4677	Electronics Store
5	2024-01-15	$22.50	************4321	Grocery Store
6	2024-01-16	$130.00	************4321	Online Marketplace

Take your time to look at the data and observe what are the difference from the original CSV file data.

Congratulations! You just learn how to use AWS Glue Python Shell to process and transform data stored in an S3 bucket using Python scripts. This is only an introductory of AWS Glue can do. Remember redacting is just one of the data transformation you can do with AWS Glue. You can also drop columns, join and relationalize data, etc. From this point you can go and try and experiment different python shell script as you see fit.

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!