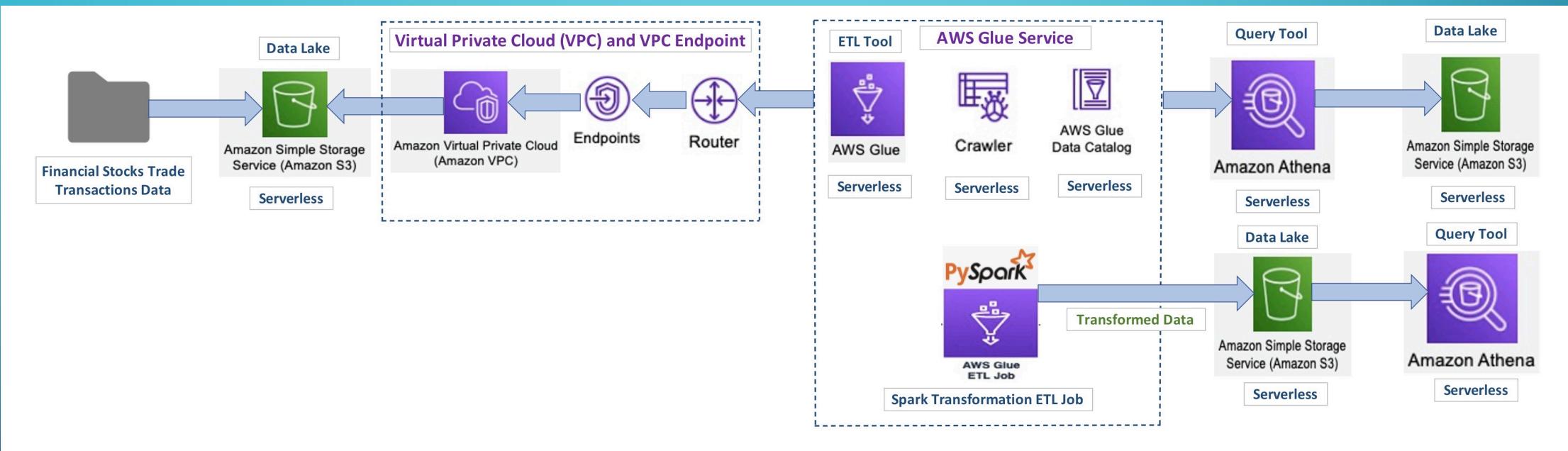




BUILDING A FINANCIAL DATA PIPELINE WITH SPARK TRANSFORMATION USING PYTHON AND AWS

FINANCIAL STOCKS TRADE TRANSACTIONS DATA

PROJECT ARCHITECTURE



TECHNOLOGY STACK

Programming/Scripting Language:

- ❖ Python

Query Language:

- ❖ SQL

Command Line Interface (CLI):

- ❖ AWS CLI

AWS Services:

- ❖ Identity and Access Management (IAM)
- ❖ Virtual Private Cloud (VPC) and VPC Endpoint
- ❖ Single Storage Service (S3)
- ❖ Glue (Glue Crawler, Glue Catalog, Glue Database & Tables, ETL Job, Spark Transformation)
- ❖ CloudWatch
- ❖ Athena
- ❖ Spark

1. CREATION OF AWS IAM USER WITH FULL ACCESS TO AWS S3 BUCKET

Summary

User ARN: arn:aws:iam::146871189787:user/VP_S3_User

Path: /

Creation time: 2022-03-28 19:14 UTC+0100

Permissions Groups (1) Tags Security credentials Access Advisor

▼ Permissions policies (1 policy applied)

Add permissions + Add inline policy

Policy name	Policy type
Attached from group	AWS managed policy from group VP_S3_Group
AmazonS3FullAccess	x

Policy summary { } JSON Simulate policy

Filter

Service	Access level	Resource	Request condition
S3	Full access	All resources	None
S3 Object Lambda	Full access	All resources	None

2. CREATION OF AWS S3 BUCKET (*CONTINUED*)

AWS S3 Bucket before executing the python scripts to create a bucket

The screenshot shows the AWS S3 Buckets page. At the top left, it says "Amazon S3 > Buckets". Below that is a section titled "Account snapshot" with a link to "View Storage Lens dashboard". Underneath is a table header for "Buckets (0) [Info](#)". The table has columns for "Name", "AWS Region", "Access", and "Creation date". To the right of the table are buttons for "Create bucket", "Copy ARN", "Empty", and "Delete". Below the table, there's a search bar with the placeholder "Find buckets by name" and a pagination area with a single page indicator "1". A message at the bottom center says "No buckets" and "You don't have any buckets.", with a "Create bucket" button below it.

2. CREATION OF AWS S3 BUCKET (CONTINUED)

AWS S3 Bucket after executing the python scripts to create a bucket

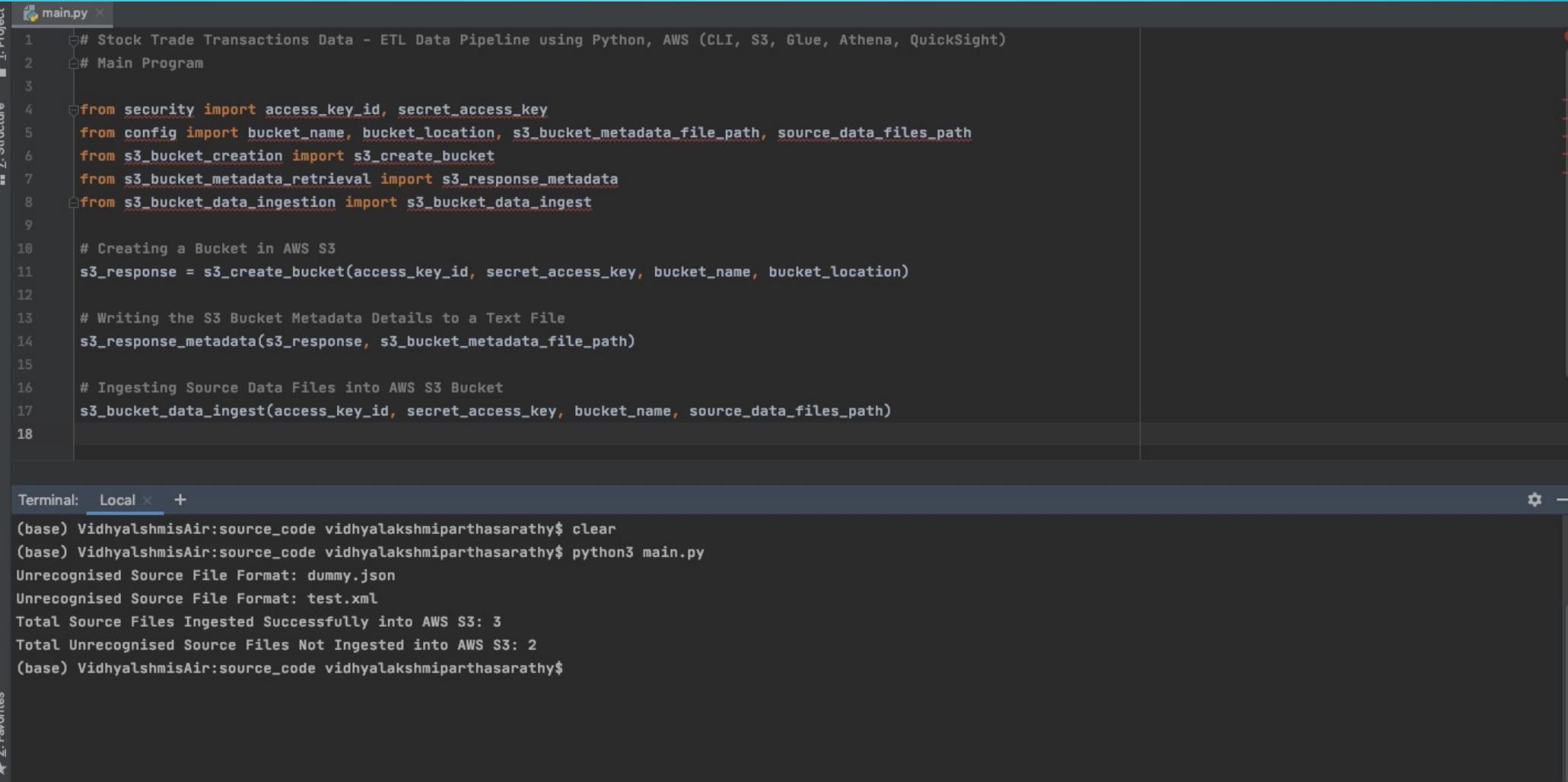
The screenshot shows the AWS S3 Buckets page. At the top left, there is a navigation bar with three horizontal lines and the text "Amazon S3 > Buckets". On the right side of the top bar, there is a small info icon. Below the top bar, there is a section titled "Account snapshot" with a sub-section "Storage lens provides visibility into storage usage and activity trends. Learn more" and a "View Storage Lens dashboard" button. In the center, there is a table titled "Buckets (1) Info" with a "Find buckets by name" search bar above it. The table has columns: Name, AWS Region, Access, and Creation date. There is one row showing a bucket named "vp-stock-trades-bucket" located in "EU (London) eu-west-2" with "Objects can be public" access and created on "March 30, 2022, 16:31:24 (UTC+01:00)". To the right of the table, there are buttons for "Copy ARN", "Empty", "Delete", and "Create bucket". Below the table, there are navigation arrows and a refresh icon.

Name	AWS Region	Access	Creation date
vp-stock-trades-bucket	EU (London) eu-west-2	Objects can be public	March 30, 2022, 16:31:24 (UTC+01:00)

2. CREATION OF AWS S3 BUCKET

The screenshot shows the AWS S3 console interface for the bucket 'vp-stock-trades-bucket'. The top navigation bar includes 'Amazon S3 > Buckets > vp-stock-trades-bucket'. The main title 'vp-stock-trades-bucket' has an 'Info' link. Below the title is a navigation bar with tabs: 'Objects' (highlighted in orange), 'Properties', 'Permissions', 'Metrics', 'Management', and 'Access Points'. The 'Objects' tab displays a section titled 'Objects (0)'. A descriptive text states: '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.' It includes a 'Learn more' link. Below this are several action buttons: 'C' (Create), 'Copy S3 URI', 'Copy URL', 'Download', 'Open', 'Delete', 'Actions', 'Create folder', and 'Upload' (highlighted in orange). A search bar with placeholder text 'Find objects by prefix' is also present. At the bottom, there is a table header with columns: 'Name', 'Type', 'Last modified', 'Size', and 'Storage class'. The main content area below the table header displays the message 'No objects' and 'You don't have any objects in this bucket.' with a central 'Upload' button.

3. INGESTING SOURCE DATA FILES IN TO AWS S3 BUCKET (CONTINUED)



```
main.py x
1 # Stock Trade Transactions Data - ETL Data Pipeline using Python, AWS (CLI, S3, Glue, Athena, QuickSight)
2 # Main Program
3
4 from security import access_key_id, secret_access_key
5 from config import bucket_name, bucket_location, s3_bucket_metadata_file_path, source_data_files_path
6 from s3_bucket_creation import s3_create_bucket
7 from s3_bucket_metadata_retrieval import s3_response_metadata
8 from s3_bucket_data_ingestion import s3_bucket_data_ingest
9
10 # Creating a Bucket in AWS S3
11 s3_response = s3_create_bucket(access_key_id, secret_access_key, bucket_name, bucket_location)
12
13 # Writing the S3 Bucket Metadata Details to a Text File
14 s3_response_metadata(s3_response, s3_bucket_metadata_file_path)
15
16 # Ingesting Source Data Files into AWS S3 Bucket
17 s3_bucket_data_ingest(access_key_id, secret_access_key, bucket_name, source_data_files_path)
18
```

Terminal: Local × +

```
(base) VidhyalshmisAir:source_code vidhyalakshmi.parthasarathy$ clear
(base) VidhyalshmisAir:source_code vidhyalakshmi.parthasarathy$ python3 main.py
Unrecognised Source File Format: dummy.json
Unrecognised Source File Format: test.xml
Total Source Files Ingested Successfully into AWS S3: 3
Total Unrecognised Source Files Not Ingested into AWS S3: 2
(base) VidhyalshmisAir:source_code vidhyalakshmi.parthasarathy$
```

1:Project
2:Structure
★ 2:Favorites

3. INGESTING SOURCE DATA FILES IN TO AWS S3 BUCKET (CONTINUED)

The screenshot shows the Amazon S3 console interface for the 'vp-stock-trades-bucket'. The top navigation bar shows 'Amazon S3 > Buckets > vp-stock-trades-bucket'. The main title is 'vp-stock-trades-bucket' with an 'Info' link. Below the title is a navigation bar with tabs: Objects (highlighted in orange), Properties, Permissions, Metrics, Management, and Access Points.

The main content area is titled 'Objects (2)'. A sub-instruction says: '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.' Below this is a 'Learn more' link with a help icon.

Below the sub-instruction is a row of buttons: 'Copy S3 URI' (white), 'Copy URL' (white), 'Download' (white), 'Open' (white), 'Delete' (white), 'Actions ▾' (white), 'Create folder' (white), and 'Upload' (orange). To the right of these buttons are navigation arrows and a refresh icon.

Below the buttons is a search bar with the placeholder text 'Find objects by prefix'.

The main table lists two objects:

	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	financial_stocks_data_csv/	Folder	-	-	-
<input type="checkbox"/>	financial_stocks_data_txt/	Folder	-	-	-

3. INGESTING SOURCE DATA FILES IN TO AWS S3 BUCKET (CONTINUED)

The screenshot shows the Amazon S3 console interface. The navigation path is: Amazon S3 > Buckets > vp-stock-trades-bucket > financial_stocks_data_csv/. The current view is the 'Objects' tab, which displays two CSV files: 'stock_sectors.csv' and 'trade_transactions.csv'. The objects are listed in a table with columns for Name, Type, Last modified, Size, and Storage class. The 'Actions' button is highlighted in orange.

financial_stocks_data_csv/

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](#)

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	stock_sectors.csv	csv	March 30, 2022, 16:45:43 (UTC+01:00)	122.7 KB	Standard
<input type="checkbox"/>	trade_transactions.csv	csv	March 30, 2022, 16:45:43 (UTC+01:00)	78.1 KB	Standard

3. INGESTING SOURCE DATA FILES IN TO AWS S3 BUCKET

The screenshot shows the Amazon S3 console interface. The navigation path is: Amazon S3 > Buckets > vp-stock-trades-bucket > financial_stocks_data_txt/. The current view is the 'Objects' tab. A single object, 'marketcap.txt', is listed. The object details are as follows:

Name	Type	Last modified	Size	Storage class
marketcap.txt	txt	March 30, 2022, 16:45:43 (UTC+01:00)	8.0 KB	Standard

At the top right of the objects list, there is a 'Copy S3 URI' button. Below the objects list, there is a search bar labeled 'Find objects by prefix' and a pagination indicator showing page 1 of 1.

4. CREATION OF A VPC ENDPOINT INTERFACE TO ENABLE AWS GLUE TO ACCESS AWS S3 (CONTINUED)

Endpoints (1/1) <small>Info</small>			
<input type="checkbox"/>	Name	VPC endpoint ID	VPC ID
<input checked="" type="checkbox"/>	vp-stock-trades-bucket-vpc-endpoint	vpce-0cef04dedf3d3e365	vpc-0c73fc06befa93a5c vp-stock-trades-bucket-vpc
Endpoint ID vpce-0cef04dedf3d3e365	Status Available	Creation time Wednesday, March 30, 2022, 17:05:46 GMT+1	Endpoint type Gateway
VPC ID vpc-0c73fc06befa93a5c (vp-stock-trades-bucket-vpc)	Status message -	Service name com.amazonaws.eu-west-2.s3	Private DNS names enabled No

4. CREATION OF A VPC ENDPOINT INTERFACE TO ENABLE AWS GLUE TO ACCESS AWS S3 (CONTINUED)

The screenshot shows the AWS VPC Endpoint interface details page for the endpoint `vpce-0cef04dedf3d3e365`. The endpoint is associated with the VPC `vpc-0c73fc06befa93a5c` and has an `Endpoint ID` of `vpce-0cef04dedf3d3e365`. It is currently `Available` and was created on `Wednesday, March 30, 2022, 17:05:46 GMT+1`. The service name is `com.amazonaws.eu-west-2.s3`, and it is a `Gateway` type endpoint. Private DNS names are not enabled. The `Route tables` tab is selected, showing one route table named `rtb-04ff62337ed83f7d0` which is associated with 3 subnets.

Details	Actions
Endpoint ID vpce-0cef04dedf3d3e365	Status Available
VPC ID vpc-0c73fc06befa93a5c (vp-stock-trades-bucket-vpc)	Status message -
Creation time Wednesday, March 30, 2022, 17:05:46 GMT+1	Endpoint type Gateway
Service name com.amazonaws.eu-west-2.s3	Private DNS names enabled No

Route tables | Policy | Tags

Route tables (1)			
Name	Route Table ID	Main	Associated Id
-	rtb-04ff62337ed83f7d0	Yes	3 subnets

Filter route tables

Manage route tables

4. CREATION OF A VPC ENDPOINT INTERFACE TO ENABLE AWS GLUE TO ACCESS AWS S3

VPC > Your VPCs > vpc-0c73fc06befa93a5c

vpc-0c73fc06befa93a5c / vp-stock-trades-bucket-vpc

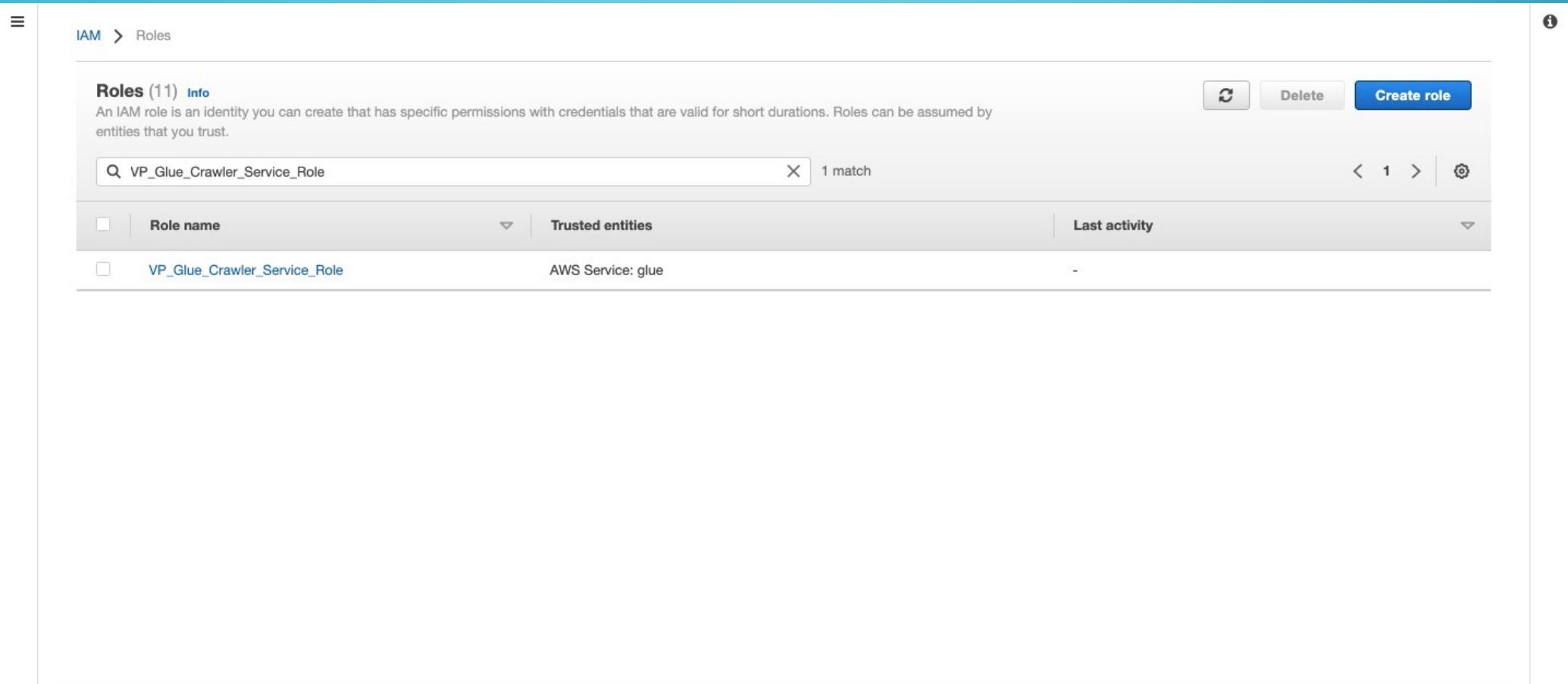
Actions ▾

Details		Info	
VPC ID	<input type="text"/> vpc-0c73fc06befa93a5c	State	Available
Tenancy	Default	DHCP options set	dopt-02273cd8ff19fd048
Default VPC	Yes	IPv4 CIDR	172.31.0.0/16
Route 53 Resolver DNS Firewall rule groups	-	Owner ID	<input type="text"/> 146871189787
DNS hostnames		DNS resolution	
Enabled		Enabled	
Main route table		Main network ACL	
rtb-04ff62337ed83f7d0		acl-0dd545e3e0f094775	
IPv6 pool		IPv6 CIDR (Network border group)	
-		-	

CIDRs | Flow logs | Tags

CIDRs		Info		
Address type	CIDR	Network Border Group	Pool	Status
IPv4	172.31.0.0/16	-	-	Associated

5. CREATION OF AN IAM ROLE TO GRANT PERMISSIONS TO AWS GLUE CRAWLER TO ACCESS AWS S3 (*CONTINUED*)



5. CREATION OF AN IAM ROLE TO GRANT PERMISSIONS TO AWS GLUE CRAWLER TO ACCESS AWS S3

The screenshot shows the AWS IAM Roles page with the following details:

Breadcrumbs: IAM > Roles > VP_Glue_Crawler_Service_Role

Role Name: VP_Glue_Crawler_Service_Role

Description: Allows Glue Service with read-only permissions to access the folders and files in the AWS S3 Bucket.

Summary:

Creation date	ARN
March 30, 2022, 17:28 (UTC+01:00)	arn:aws:iam::146871189787:role/VP_Glue_Crawler_Service_Role
Last activity	Maximum session duration
None	1 hour

Permissions: This tab is selected. Other tabs include Trust relationships, Tags, Access Advisor, and Revoke sessions.

Permissions policies (2): You can attach up to 10 managed policies.

Actions: Filter policies by property or policy name and press enter, Refresh, Simulate, Remove, Add permissions ▾, Previous, Next, and Refresh.

Policy name	Type	Description
AWSGlueServiceRole	AWS managed	Policy for AWS Glue service role which allows access to related services including EC2, S3, and Cloudwatch Logs
AmazonS3ReadOnlyAccess	AWS managed	Provides read only access to all buckets via the AWS Management Console.

6. CREATION OF A DATABASE IN THE AWS GLUE TO STORE THE GLUE CRAWLER DATA CATALOG SCHEMA RESULTS (CONTINUED)

The screenshot shows the AWS Glue Data Catalog interface. On the left, a sidebar menu lists various AWS Glue services: Data catalog, Databases, Tables, Connections, Crawlers, Classifiers, Schema registries, Schemas, Settings, ETL, AWS Glue Studio, Jobs, Jobs (legacy), ML Transforms, Blueprints, Workflows, Triggers, Dev endpoints, Notebooks, and Security. The 'Databases' section is currently selected.

The main content area is titled 'Databases' with the sub-instruction: 'A database is a set of associated table definitions, organized into a logical group.' Below this, there are three buttons: 'Add database' (highlighted in blue), 'View tables', and 'Action ▾'. To the right, a table displays one database entry:

Name	Description
<input type="checkbox"/> vp-stock-trades-bucket-database	This is the AWS Glue Database created to store the Data Catalog Schema Results retrieved by AWS Glue Crawler, by crawling the stock trade transaction raw source data files ingested in AWS S3 Bucket.

At the bottom right of the table, there are links for 'Showing: 1 - 1 < > 🔍 ⓘ'.

6. CREATION OF A DATABASE IN THE AWS GLUE TO STORE THE GLUE CRAWLER DATA CATALOG SCHEMA RESULTS

The screenshot shows the AWS Glue Data Catalog interface. On the left, there is a sidebar with the following navigation options:

- AWS Glue**
- Data catalog**
 - Databases
 - Tables
 - Connections
 - Crawlers
 - Classifiers
 - Schema registries
 - Schemas
 - Settings
- ETL**
 - AWS Glue Studio
 - Jobs
 - Jobs (legacy)
 - ML Transforms
 - Blueprints
 - Workflows
 - Triggers
 - Dev endpoints
 - Notebooks
- Security**
 - Security configurations

The main content area shows the details of a database named "vp-stock-trades-bucket-database". The database was created on "2023-09-01T10:00:00Z". It has a "Description" field stating: "This is the AWS Glue Database created to stored the Data Catalog Schema Results retrieved by AWS Glue Crawler, by crawling the stock trade transaction raw source data files ingested in AWS S3 Bucket." There are "Edit database" and "Delete database" buttons at the top of this section.

Below the database details, there is a link to "Tables in vp-stock-trades-bucket-database".

7. ESTABLISH A NETWORK CONNECTION FROM AWS GLUE CRAWLER THROUGH VPC ENDPOINT TO AWS S3 BUCKET (CONTINUED)

The screenshot shows the AWS Glue Connections page. On the left, there's a sidebar with navigation links for Data catalog, ETL, and AWS Glue Studio. The main area is titled "Connections" with a sub-instruction: "A connection contains the properties needed to connect to your data." It includes buttons for "Add connection", "Test connection", and "Action". A table lists the existing connections, showing one entry:

<input type="checkbox"/>	Name	Type	Date created	Last updated	Updated by
<input type="checkbox"/>	vp-stock-trades-bucket-glue-connection	Network	30 March 2022 6:01 PM UTC+1	30 March 2022 6:01 PM UTC+1	user/vidhyala...

At the top right of the main area, it says "Showing: 1 - 1" followed by navigation icons. At the bottom right, there are refresh and help icons.

7. ESTABLISH A NETWORK CONNECTION FROM AWS GLUE CRAWLER THROUGH VPC ENDPOINT TO AWS S3 BUCKET

The screenshot shows the AWS Glue Connections page. On the left, there's a sidebar with 'AWS Glue' and 'ETL' sections. Under 'AWS Glue', there are links for Data catalog, Databases, Tables, Connections, Crawlers, Classifiers, Schema registries, Schemas, and Settings. Under 'ETL', there are links for AWS Glue Studio, Jobs (New), Jobs (legacy), ML Transforms, Blueprints, Workflows, Triggers, Dev endpoints, and Notebooks. The main content area shows a connection named 'vp-stock-trades-bucket-glue-connection'. The 'Edit' button is highlighted with a red box. The connection details are as follows:

Setting	Value
Type	Network
VPC Id	vpc-0c73fc06befa93a5c
Subnet	subnet-018f5d5d635679cd6
Security groups	sg-06ce986cc8a1ca2ab
Require SSL connection	false
Description	Establish a network connection from AWS Glue to AWS S3 Bucket.
Created	30 March 2022 6:01 PM UTC+1
Last modified	30 March 2022 6:01 PM UTC+1

8. VERIFY/TEST THE NETWORK CONNECTION FROM AWS GLUE CRAWLER THROUGH VPC ENDPOINT TO AWS S3 BUCKET (CONTINUED)

The screenshot shows the AWS Glue Connections interface. On the left, the navigation menu includes options like Data catalog, Databases, Tables, Connections (which is selected), Crawlers, Classifiers, Schema registries, Schemas, Settings, ETL, AWS Glue Studio, Jobs, and ML Transforms. The main area displays a list of connections, with one entry visible: "vp-stock-trades" (Last updated: M UTC+1, 30 March 2022 6:01 PM UTC+1, Updated by user/vidhyala...). A modal dialog titled "Test connection" is open over the list. The dialog contains the following fields:

- IAM role**: VP_Glue_Crawler_Service_Role
- Include path**: s3://vp-stock-trades-bucket/financial_stocks_da
- S3** tree view:
 - vp-stock-trades-bucket
 - financial_stocks_data_csv
 - stock_sectors.csv
 - trade_transactions.csv**
 - financial_stocks_data_txt

At the bottom of the dialog is a blue "Test connection" button.

8. VERIFY/TEST THE NETWORK CONNECTION FROM AWS GLUE CRAWLER THROUGH VPC ENDPOINT TO AWS S3 BUCKET (CONTINUED)

AWS Glue

Data catalog

Databases

Tables

Connections

Crawlers

Classifiers

Schema registries

Schemas

Settings

ETL

AWS Glue Studio

Jobs - New

Jobs (legacy)

ML Transforms

Blueprints

Workflows

Triggers

Dev endpoints

Notebooks

Connections A connection contains the properties needed to connect to your data.

Testing vp-stock-trades-bucket-glue-connection access to your data store is in progress. This can take a few moments.

Add connection Test connection Action ▾ Showing: 1 - 1

<input checked="" type="checkbox"/> Name	Type	Date created	Last updated	Updated by
<input checked="" type="checkbox"/> vp-stock-trades-bucket-glue-connection	Network	30 March 2022 6:01 PM UTC+1	30 March 2022 6:01 PM UTC+1	user/vidhyala...

8. VERIFY/TEST THE NETWORK CONNECTION FROM AWS GLUE CRAWLER THROUGH VPC ENDPOINT TO AWS S3 BUCKET

The screenshot shows the AWS Glue Connections page. On the left sidebar, under the 'Connections' section, the 'Connections' link is highlighted in orange. The main content area displays a success message: 'vp-stock-trades-bucket-glue-connection connected successfully to your instance.' Below this message is a table listing connections. The table has columns: Name, Type, Date created, Last updated, and Updated by. One row is visible, showing the connection 'vp-stock-trades-bucket-glue-connection' which is a 'Network' type connection created and last updated on 30 March 2022 at 6:01 PM UTC+1, and was updated by 'user/vidhyala...'. The table includes pagination and action buttons for adding new connections and testing existing ones.

Name	Type	Date created	Last updated	Updated by
vp-stock-trades-bucket-glue-connection	Network	30 March 2022 6:01 PM UTC+1	30 March 2022 6:01 PM UTC+1	user/vidhyala...

9. CREATION OF AWS GLUE CRAWLER TO RETRIEVE DATA CATALOG SCHEMA RESULTS OF THE SPECIFIED SOURCE FILES IN AWS S3 BUCKET (CONTINUED)

The screenshot shows the AWS Glue Data Catalog interface. On the left, there's a sidebar with navigation links for AWS Glue, Data catalog, Databases, Tables, Connections, Crawlers, Classifiers, Schema registries, Schemas, Settings, ETL (AWS Glue Studio, Jobs, Jobs (legacy), ML Transforms, Blueprints, Workflows, Triggers, Dev endpoints, Notebooks), and a link to the AWS Glue Studio documentation.

The main content area is titled "Crawlers". It contains a brief description: "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." Below this is a search bar with placeholder text "Filter by tags and attributes".

There are three buttons at the top of the crawler list: "Add crawler" (highlighted in blue), "Run crawler", and "Action ▾". To the right of the search bar are "User preferences" and "Showing: 1 - 1 < > ⌂ ⓘ".

The crawler list table has the following columns: Name, Schedule, Status, Logs, Last runtime, Median runtime, Tables updated, and Tables added. One row is visible, showing the crawler "vp-stock-trades-bucket-glue-crawler" with a status of "Ready", last runtime of "0 secs", median runtime of "0 secs", and both "Tables updated" and "Tables added" as "0".

Name	Schedule	Status	Logs	Last runtime	Median runtime	Tables updated	Tables added
vp-stock-trades-bucket-glue-crawler		Ready		0 secs	0 secs	0	0

9. CREATION OF AWS GLUE CRAWLER TO RETRIEVE DATA CATALOG SCHEMA RESULTS OF THE SPECIFIED SOURCE FILES IN AWS S3 BUCKET

The screenshot shows the AWS Glue console interface. On the left, there's a navigation sidebar with sections like Data catalog, ETL, and Security. The main area is titled 'Crawlers > vp-stock-trades-bucket-glue-crawler'. It displays the configuration for this specific crawler.

Crawler Details:

- Name:** vp-stock-trades-bucket-glue-crawler
- Description:** This is an AWS Glue Crawler created to crawl and retrieve the Data Catalogue Schema Results of the specific source raw data files in the configured AWS S3 Bucket.
- Create a single schema for each S3 path:** false
- Table level:**
- Security configuration:**
- Tags:** -
- State:** Ready
- Schedule:**
- Last updated:** Wed Mar 30 23:56:02 GMT+100 2022
- Date created:** Wed Mar 30 19:29:49 GMT+100 2022
- Database:** vp-stock-trades-bucket-database
- Table prefix:** vp_stock_trades_bucket
- Service role:** VP_Glue_Crawler_Service_Role
- Selected classifiers:**
- Data store:** S3
- Include path:** s3://vp-stock-trades-bucket
- Connection:** vp-stock-trades-bucket-glue-connection
- Exclude patterns:**

Configuration options:

- Schema updates in the data store:** Update the table definition in the data catalog.
- Object deletion in the data store:** Mark the table as deprecated in the data catalog.

10. RUNNING THE AWS GLUE CRAWLER TO RETRIEVE DATA CATALOG SCHEMA RESULTS OF THE SPECIFIED SOURCE FILES IN AWS S3 BUCKET (CONTINUED)

The screenshot shows the AWS Glue console interface. On the left, there is a navigation sidebar with the following menu items:

- Data catalog
- Databases
- Tables
- Connections
- Crawlers** (highlighted in orange)
- Classifiers
- Schema registries
- Schemas
- Settings

Below the main content area, there is another sidebar for ETL services:

- AWS Glue Studio
- Jobs - New
- Jobs (legacy)
- ML Transforms
- Blueprints
- Workflows
- Triggers
- Dev endpoints
- Notebooks

The main content area is titled "Crawlers" and contains the following information:

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.

Crawler "vp-stock-trades-bucket-glue-crawler" is now running.

Below this message, there is a search bar with the placeholder "Filter by tags and attributes" and a "User preferences" link. The table below shows the status of the crawler:

<input type="checkbox"/>	Name	Schedule	Status	Logs	Last runtime	Median runtime	Tables updated	Tables added
<input type="checkbox"/>	vp-stock-trades-bucket-glue-crawler		⌚ Starting		0 secs	0 secs	0	0

10. RUNNING THE AWS GLUE CRAWLER TO RETRIEVE DATA CATALOG SCHEMA RESULTS OF THE SPECIFIED SOURCE FILES IN AWS S3 BUCKET

AWS Glue

Data catalog

- Databases
- Tables
- Connections

Crawlers

- Classifiers
- Schema registries
- Schemas
- Settings

ETL

- AWS Glue Studio
- Jobs - New
- Jobs (legacy)
- ML Transforms
- Blueprints
- Workflows
- Triggers
- Dev endpoints
- Notebooks

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.

Crawler "vp-stock-trades-bucket-glue-crawler" completed and made the following changes: 1 tables created, 0 tables updated. See the tables created in database [vp-stock-trades-bucket-database](#).

[User preferences](#)

Add crawler Run crawler Action Filter by tags and attributes Showing: 1 - 1

<input type="checkbox"/>	Name	Schedule	Status	Logs	Last runtime	Median runtime	Tables updated	Tables added
<input type="checkbox"/>	vp-stock-trades-bucket-glue-crawler		Ready	Logs	2 mins	2 mins	0	1

11. VIEWING THE AWS GLUE CRAWLER EXECUTION LOGS IN AWS CLOUDWATCH (CONTINUED)

The screenshot shows the AWS CloudWatch Log Events interface. The left sidebar contains navigation links for CloudWatch, Favorites, Dashboards, Alarms, Logs (Log groups, Logs Insights), Metrics, X-Ray traces, Events, Application monitoring, Insights, Settings, and Getting Started. The main content area displays the log group path: CloudWatch > Log groups > /aws-glue/crawlers > vp-stock-trades-bucket-glue-crawler. The title is "Log events". A search bar contains the query "[401653b4-4416-40e2-a222-a29dd54e74b9]". Below the search bar are filter options: Clear, 1m, 30m, 1h, 12h, Custom, and a refresh icon. The log table has columns: ▶, Timestamp, and Message. The table lists the following log entries:

▶	Timestamp	Message
▶	2022-03-30T19:34:19.411+01:00	[401653b4-4416-40e2-a222-a29dd54e74b9] BENCHMARK : Running Start Crawl for Crawler vp-stock-trades-buck...
▶	2022-03-30T19:35:01.821+01:00	[401653b4-4416-40e2-a222-a29dd54e74b9] INFO : S3 ConnectionName is vp-stock-trades-bucket-glue-connecti...
▶	2022-03-30T19:36:17.365+01:00	[401653b4-4416-40e2-a222-a29dd54e74b9] BENCHMARK : Classification complete, writing results to database...
▶	2022-03-30T19:36:17.365+01:00	[401653b4-4416-40e2-a222-a29dd54e74b9] INFO : Crawler configured with SchemaChangePolicy {"UpdateBehavi...
▶	2022-03-30T19:36:29.056+01:00	[401653b4-4416-40e2-a222-a29dd54e74b9] INFO : Created table vp_stock_trades_buckettrade_transactions_cs...
▶	2022-03-30T19:36:29.068+01:00	[401653b4-4416-40e2-a222-a29dd54e74b9] INFO : !!398!!: Optional[{"namespace":"vp-stock-trades-bucket-da...
▶	2022-03-30T19:36:31.776+01:00	[401653b4-4416-40e2-a222-a29dd54e74b9] BENCHMARK : Finished writing to Catalog
▶	2022-03-30T19:37:39.094+01:00	[401653b4-4416-40e2-a222-a29dd54e74b9] BENCHMARK : Crawler has finished running and is in state READY

11. VIEWING THE AWS GLUE CRAWLER EXECUTION LOGS IN AWS CLOUDWATCH (CONTINUED)

The screenshot shows the AWS CloudWatch Log Events interface. The left sidebar navigation includes:

- Favorites
- Dashboards
- Alarms (0 alarms, 0 events, 0 metrics)
- Logs
 - Log groups** (highlighted in orange)
 - Logs Insights
- Metrics
- X-Ray traces
- Events
- Application monitoring
- Insights
- Settings
- Getting Started

The main content area displays log events for the log group `/aws-glue/crawlers` under the crawler `vp-stock-trades-bucket-glue-crawler`. The search bar contains the ID `401653b4-4416-40e2-a222-a29dd54e74b9`. The log table has columns **Timestamp** and **Message**. The messages show the crawler's progress, including BENCHMARK logs for start, classification, and schema configuration, as well as INFO logs for S3 connection setup.

Timestamp	Message
2022-03-30T19:34:19.411+01:00	[401653b4-4416-40e2-a222-a29dd54e74b9] BENCHMARK : Running Start Crawl for Crawler vp-stock-trades-bu... [401653b4-4416-40e2-a222-a29dd54e74b9] BENCHMARK : Running Start Crawl for Crawler vp-stock-trades-bucket-glue-crawler
2022-03-30T19:35:01.821+01:00	[401653b4-4416-40e2-a222-a29dd54e74b9] INFO : S3 ConnectionName is vp-stock-trades-bucket-glue-connec... [401653b4-4416-40e2-a222-a29dd54e74b9] INFO : S3 ConnectionName is vp-stock-trades-bucket-glue-connection
2022-03-30T19:36:17.365+01:00	[401653b4-4416-40e2-a222-a29dd54e74b9] BENCHMARK : Classification complete, writing results to database... [401653b4-4416-40e2-a222-a29dd54e74b9] BENCHMARK : Classification complete, writing results to database vp-stock-trades-bucket-database
2022-03-30T19:36:17.365+01:00	[401653b4-4416-40e2-a222-a29dd54e74b9] INFO : Crawler configured with SchemaChangePolicy {"UpdateBeha... [401653b4-4416-40e2-a222-a29dd54e74b9] INFO : Crawler configured with SchemaChangePolicy { "UpdateBehavior": "UPDATE_IN_DATABASE", "DeleteBehavior": "DEPRECATE_IN_DATABASE" }

11. VIEWING THE AWS GLUE CRAWLER EXECUTION LOGS IN AWS CLOUDWATCH (CONTINUED)

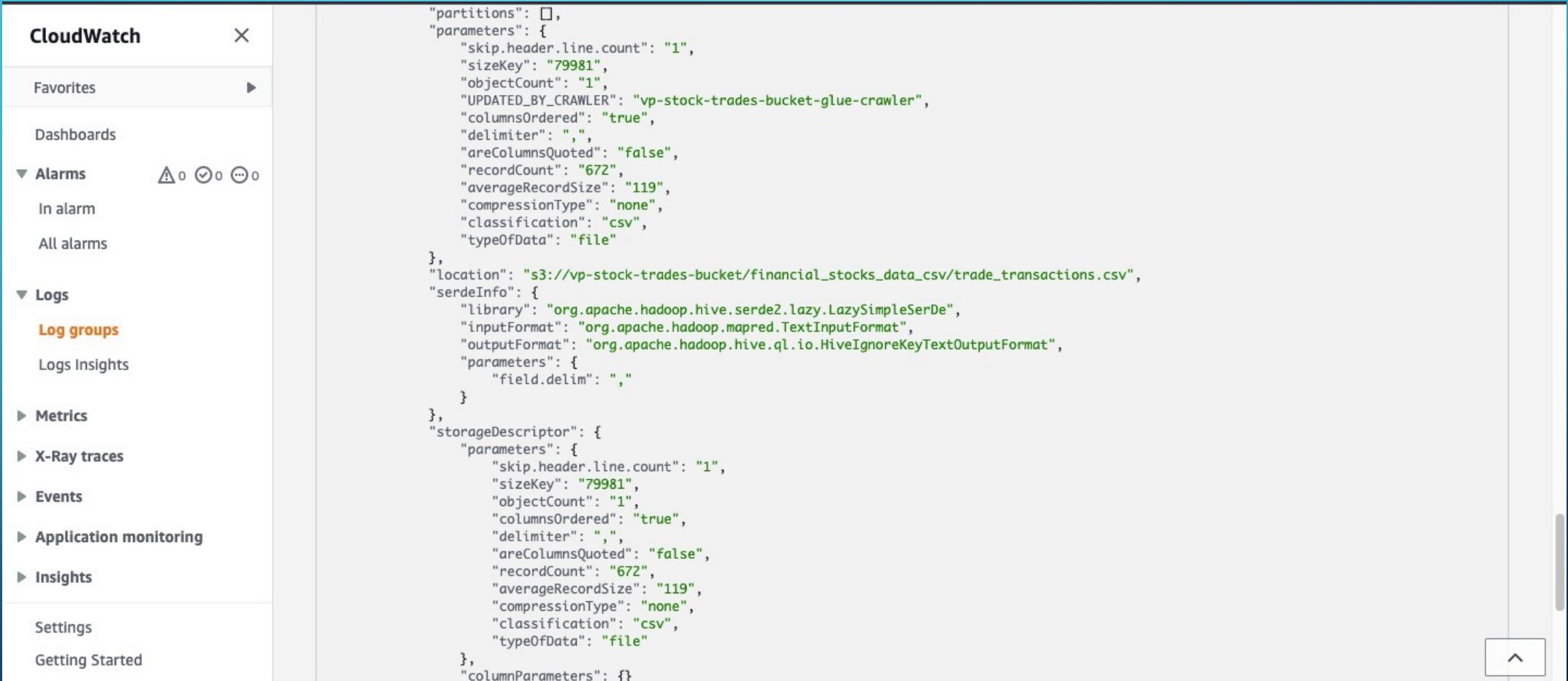
The screenshot shows the AWS CloudWatch interface with the 'Logs' section selected. On the left, a sidebar lists various monitoring options: Favorites, Dashboards, Alarms, Logs (selected), Log groups (highlighted in orange), Metrics, X-Ray traces, Events, Application monitoring, Insights, Settings, and Getting Started. The main pane displays log entries for a crawler named 'vp-stock-trades-buckettrade_transactions_crawler'. Two log entries are visible:

```
[401653b4-4416-40e2-a222-a29dd54e74b9] INFO : Created table vp_stock_trades_buckettrade_transactions...  
[401653b4-4416-40e2-a222-a29dd54e74b9] INFO : Created table vp_stock_trades_buckettrade_transactions_csv in database vp-stock-trades-bucket-database
```

```
[401653b4-4416-40e2-a222-a29dd54e74b9] INFO : !!398!!: Optional[{"namespace": "vp-stock-trades-bucket...",  
[401653b4-4416-40e2-a222-a29dd54e74b9] INFO : !!398!!: Optional[{  
    "namespace": "vp-stock-trades-bucket-database",  
    "tblName": "vp_stock_trades_buckettrade_transactions_csv",  
    "schema": {  
        "dataType": "struct",  
        "fields": [  
            {  
                "name": "Symbol",  
                "container": {  
                    "dataType": "string",  
                    "properties": {}  
                },  
                "properties": {}  
            },  
            {  
                "name": "Owner",  
                "container": {  
                    "dataType": "string",  
                    "properties": {}  
                },  
                "properties": {}  
            },  
            {  
                "name": "Relationship",  
                "container": {  
                    "dataType": "string",  
                    "properties": {}  
                },  
                "properties": {}  
            }  
        ],  
        "properties": {}  
    }],  
    "properties": {}  
}]
```

Each log entry has a 'Copy' button to its right. A scroll bar is visible on the right side of the main pane.

11. VIEWING THE AWS GLUE CRAWLER EXECUTION LOGS IN AWS CLOUDWATCH (CONTINUED)



The screenshot shows the AWS CloudWatch interface. The left sidebar contains navigation links: Favorites, Dashboards, Alarms (with 0 items), Logs (selected), Log groups, Logs Insights, Metrics, X-Ray traces, Events, Application monitoring, Insights, Settings, and Getting Started. The main content area displays the log details for a crawler named "vp-stock-trades-bucket-glue-crawler".

```
    "partitions": [],
    "parameters": {
        "skip.header.line.count": "1",
        "sizeKey": "79981",
        "objectCount": "1",
        "UPDATED_BY_CRAWLER": "vp-stock-trades-bucket-glue-crawler",
        "columnsOrdered": "true",
        "delimiter": ",",
        "areColumnsQuoted": "false",
        "recordCount": "672",
        "averageRecordSize": "119",
        "compressionType": "none",
        "classification": "csv",
        "typeOfData": "file"
    },
    "location": "s3://vp-stock-trades-bucket/financial_stocks_data_csv/trade_transactions.csv",
    "serdeInfo": {
        "library": "org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe",
        "inputFormat": "org.apache.hadoop.mapred.TextInputFormat",
        "outputFormat": "org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat",
        "parameters": {
            "field.delim": ","
        }
    },
    "storageDescriptor": {
        "parameters": {
            "skip.header.line.count": "1",
            "sizeKey": "79981",
            "objectCount": "1",
            "columnsOrdered": "true",
            "delimiter": ",",
            "areColumnsQuoted": "false",
            "recordCount": "672",
            "averageRecordSize": "119",
            "compressionType": "none",
            "classification": "csv",
            "typeOfData": "file"
        },
        "columnParameters": {}
    }
}
```

11. VIEWING THE AWS GLUE CRAWLER EXECUTION LOGS IN AWS CLOUDWATCH

The screenshot shows the AWS CloudWatch console interface. On the left, a sidebar menu lists various CloudWatch services: Favorites, Dashboards, Alarms, Logs (selected), Metrics, X-Ray traces, Events, Application monitoring, Insights, Settings, and Getting Started. The Logs section is expanded, showing Log groups and Log Insights.

The main pane displays the execution logs for a crawler. The log entries are:

```
        "sizeKey": "79981",
        "objectCount": "1",
        "columnsOrdered": "true",
        "delimiter": ",",
        "areColumnsQuoted": "false",
        "recordCount": "672",
        "averageRecordSize": "119",
        "compressionType": "none",
        "classification": "csv",
        "typeOfData": "file"
    },
    "columnParameters": {}
},
"hiveCompatible": false,
"description": null,
"columnComments": {},
"partitionComments": {},
"additionalLocations": null,
"registeredWithLakeFormation": false,
"classification": {
    "present": true
},
"deprecated": false
}
] CatalogDataForLocation(location=s3://vp-stock-trades-bucket/financial_stocks_data_csv/trade_transactions.csv,
simpleName=vp_stock_trades_buckettrade_transactions_csv,
compositeName=vp_stock_trades_buckettrade_transactions_csv_3f23fa671f8a592aab69a81b91a7038e, existingTable=Optional.empty,
tableForSimpleNamePresent=false)

```

2022-03-30T19:36:31.776+01:00 [401653b4-4416-40e2-a222-a29dd54e74b9] BENCHMARK : Finished writing to Catalog

2022-03-30T19:37:39.094+01:00 [401653b4-4416-40e2-a222-a29dd54e74b9] BENCHMARK : Finished writing to Catalog

2022-03-30T19:37:39.094+01:00 [401653b4-4416-40e2-a222-a29dd54e74b9] BENCHMARK : Crawler has finished running and is in state READY

2022-03-30T19:37:39.094+01:00 [401653b4-4416-40e2-a222-a29dd54e74b9] BENCHMARK : Crawler has finished running and is in state READY

Copy Copy ^

12. VERIFYING THE DATA CATALOG SCHEMA RESULTS STORED IN THE AWS GLUE DATABASE TABLE (CONTINUED)

AWS Glue Database Table before executing the AWS Glue Crawler

The screenshot shows the AWS Glue Data Catalog interface. On the left, there's a sidebar with sections for Data catalog, ETL, and Dev endpoints. Under Data catalog, the 'Tables' section is selected, indicated by an orange highlight. The main content area is titled 'Tables' with a sub-instruction: 'A table is the metadata definition that represents your data, including its schema. A table can be used as a source or target in a job definition.' Below this, there are buttons for 'Add tables' and 'Action', a search bar with placeholder 'Filter by attributes or search by keyword', and a 'Save view' button. The table header includes columns for Name, Database, Location, Classification, Last updated, and Deprecated. A message at the bottom states, 'You don't have any tables defined in your data catalog.', accompanied by a grid icon. A blue button labeled 'Add tables using a crawler' is also present.

12. VERIFYING THE DATA CATALOG SCHEMA RESULTS STORED IN THE AWS GLUE DATABASE TABLE (CONTINUED)

AWS Glue Database Table after executing the AWS Glue Crawler

The screenshot shows the AWS Glue Data Catalog interface. On the left, there's a sidebar with navigation links for Data catalog, ETL, and Security. Under Data catalog, the 'Tables' link is highlighted in orange. The main area is titled 'Tables' with a sub-instruction: 'A table is the metadata definition that represents your data, including its schema. A table can be used as a source or target in a job definition.' Below this is a search bar with a placeholder 'Name : vp_stock_trades_bucketfinancial_stocks...', a filter bar, and a 'Save view' button. The table itself has columns: Name, Database, Location, Classification, Last updated, and Deprecated. There is one row visible:

Name	Database	Location	Classification	Last updated	Deprecated
vp_stock_trades_bucketfinancial_stocks_data_csv	vp-stock-trades-bucket-database	s3://vp-stock-trades-bucket/fin...	csv	31 March 2022 12:00 PM UTC+1	

12. VERIFYING THE DATA CATALOG SCHEMA RESULTS STORED IN THE AWS GLUE DATABASE TABLE (CONTINUED)

AWS Glue Database Table after executing the AWS Glue Crawler

The screenshot shows the AWS Glue Data Catalog interface. On the left, there's a sidebar with navigation links for Data catalog, ETL, and other services like AWS Glue Studio, Jobs, Workflows, etc. The main area displays a table named "vp_stock_trades_bucketfinancial_stocks_data_csv". The table details are as follows:

Name	vp_stock_trades_bucketfinancial_stocks_data_csv
Description	
Database	vp-stock-trades-bucket-database
Classification	csv
Location	s3://vp-stock-trades-bucket/financial_stocks_data_csv/
Connection	
Deprecated	No
Last updated	Thu Mar 31 00:00:10 GMT+100 2022
Input format	org.apache.hadoop.mapred.TextInputFormat
Output format	org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat
Serde serialization lib	org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe
Serde parameters	field.delim , skip.header.line.count 1 sizeKey 79981 objectCount 1
Table properties	UPDATED_BY_CRAWLER vp-stock-trades-bucket-glue-crawler CrawlerSchemaSerializerVersion 1.0 recordCount 672 averageRecordSize 119 CrawlerSchemaDeserializerVersion 1.0 compressionType none columnsOrdered true areColumnsQuoted false delimiter , typeOfData file

On the right, there's a "Versions" section showing one version entry:

Version	Created	Created by
0	31 March 2022 1...	role/VP_Glue_Cr... Crawler

12. VERIFYING THE DATA CATALOG SCHEMA RESULTS STORED IN THE AWS GLUE DATABASE TABLE (CONTINUED)

AWS Glue Database Table after executing the AWS Glue Crawler

The screenshot shows the AWS Glue Data Catalog interface. On the left, there's a sidebar with navigation links for Data catalog, Databases, Tables, Connections, Crawlers, Classifiers, Schema registries, Schemas, Settings, ETL (AWS Glue Studio, Jobs, Jobs (legacy), ML Transforms, Blueprints, Workflows, Triggers, Dev endpoints, Notebooks), and Security. The main area shows a table named "vp_stock_trades_bucketfinancial_stocks_data_csv". The table properties are listed as follows:

Name	vp_stock_trades_bucketfinancial_stocks_data_csv
Description	
Database	vp-stock-trades-bucket-database
Classification	csv
Location	s3://vp-stock-trades-bucket/financial_stocks_data_csv/
Connection	
Deprecated	No
Last updated	Thu Mar 31 00:00:10 GMT+100 2022
Input format	org.apache.hadoop.mapred.TextInputFormat
Output format	org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat
Serde serialization lib	org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe
Serde parameters	field.delim : , skip.header.line.count : 1 sizeKey : 79981 objectCount : 1
Table properties	UPDATED_BY_CRAWLER : vp-stock-trades-bucket-glue-crawler CrawlerSchemaSerializerVersion : 1.0 recordCount : 672 averageRecordSize : 119 CrawlerSchemaDeserializerVersion : 1.0 compressionType : none columnsOrdered : true areColumnsQuoted : false delimiter : , typeOfData : file

At the bottom right, it says "Showing: 1 - 12 of 12".

12. VERIFYING THE DATA CATALOG SCHEMA RESULTS STORED IN THE AWS GLUE DATABASE TABLE

AWS Glue Database Table after executing the AWS Glue Crawler

The screenshot shows the AWS Glue Data Catalog interface. On the left, there's a sidebar with navigation links for AWS Glue, Data catalog, ETL, and Security. The main area displays a table named 'vp_stock_trades' with the following properties:

UPDATED_BY_CRAWLER	vp_stock_trades	bucket	glue crawler	CrawlerSchemaDeserializerVersion	1.0	compressionType	none	columnsOrdered	true
averageRecordSize	119	areColumnsQuoted	false	delimiter	,	typeOfData	file		

The table has 12 columns, each with a numerical ID from 1 to 12. The schema details are as follows:

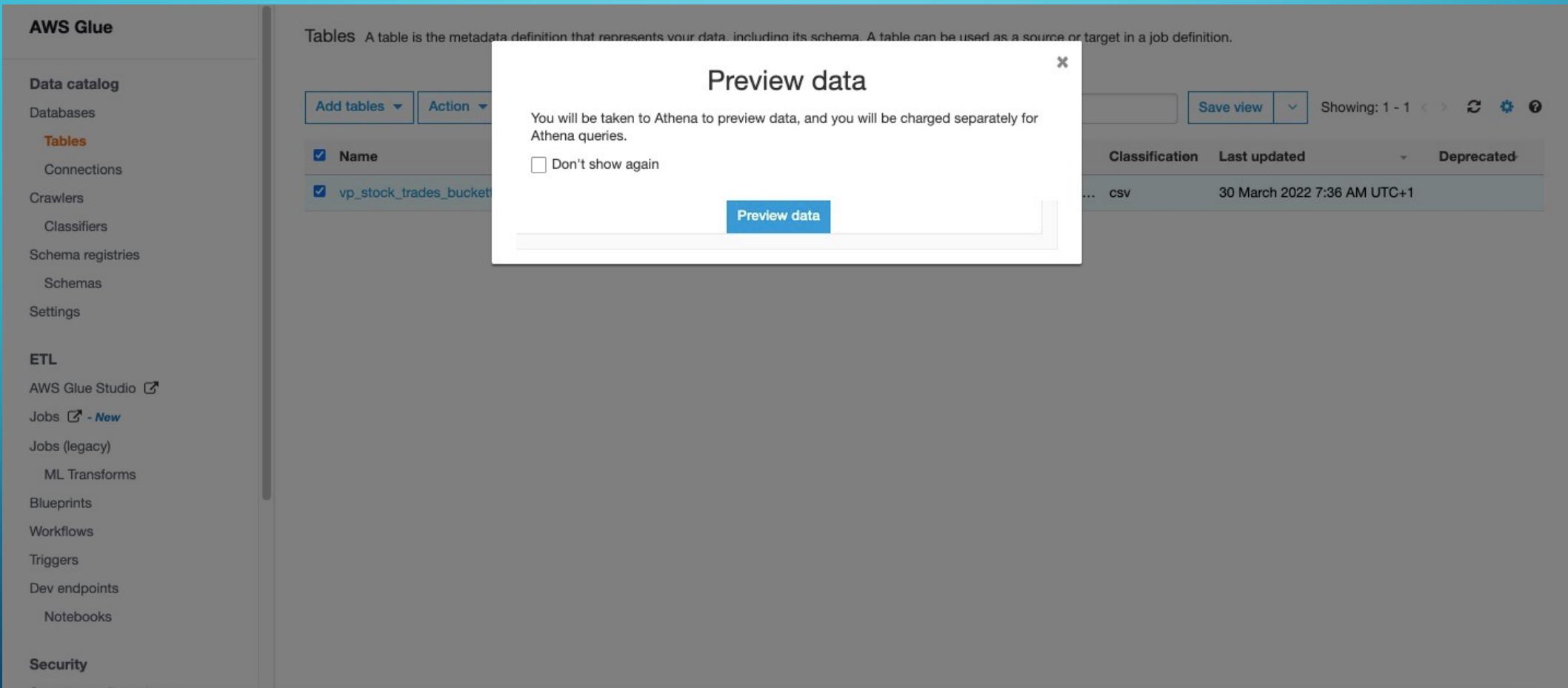
	Column name	Data type	Partition key	Comment
1	symbol	string		
2	owner	string		
3	relationship	string		
4	date	string		
5	cost	double		
6	# shares	bigint		
7	value(\$)	string		
8	total shares	bigint		
9	filing	string		
10	type	string		
11	currentprice	string		
12	movingaverage	string		

13. QUERYING THE RAW SOURCE DATA STORED IN THE AWS S3 BUCKET THROUGH AWS GLUE INTERFACE USING AWS ATHENA (CONTINUED)

The screenshot shows the AWS Glue Data Catalog interface. On the left, there's a sidebar with navigation links for Data catalog, ETL, and Security. Under Data catalog, the 'Tables' link is highlighted. The main area displays a table of tables. One row is selected, and a context menu is open over it, showing options like 'Edit table details', 'View details', 'View data' (which is highlighted with a blue background), and 'Delete table'. The table itself has columns for Name, Database, Location, Classification, Last updated, and Deprecated. The selected row shows 'actions_csv' as the name, 'vp-stock-trades-bucket-database' as the database, 's3://vp-stock-trades-bucket/fin...' as the location, 'csv' as the classification, and '30 March 2022 7:36 AM UTC+1' as the last updated time.

Name	Database	Location	Classification	Last updated	Deprecated
actions_csv	vp-stock-trades-bucket-database	s3://vp-stock-trades-bucket/fin...	csv	30 March 2022 7:36 AM UTC+1	

13. QUERYING THE RAW SOURCE DATA STORED IN THE AWS S3 BUCKET THROUGH AWS GLUE INTERFACE USING AWS ATHENA (CONTINUED)



13. QUERYING THE RAW SOURCE DATA STORED IN THE AWS S3 BUCKET THROUGH AWS GLUE INTERFACE USING AWS ATHENA (CONTINUED)

The screenshot shows the Amazon Athena Query editor interface. The top navigation bar includes 'Amazon Athena > Query editor', tabs for 'Editor' (selected), 'Recent queries', 'Saved queries', and 'Settings', and a 'Workgroup' dropdown set to 'primary'. The main area is divided into two panes: 'Data' on the left and 'Query' on the right.

Data Pane: Contains fields for 'Data Source' (set to 'AwsDataCatalog') and 'Database' (set to 'vp-stock-trades-bucket-database'). Below these are sections for 'Tables and views' (with a 'Create' button) and a table listing. The table section shows 'Tables (2)' with one entry: 'vp_stock_trades_bucketfinancial_stocks_data_csv'. This table has columns: symbol (string), owner (string), relationship (string), date (string), cost (double), # shares (int), value(\$), and total shares (int).

Query Pane: Shows 'Query 7' (disabled) and 'Query 8' (selected). The SQL query in 'Query 8' is: `1 SELECT * FROM "AwsDataCatalog"."vp-stock-trades-bucket-database"."vp_stock_trades_bucketfinancial_stocks_data_csv" limit 10;`. The status bar indicates the query is 'Completed' with a run time of 0.424 sec and data scanned of 78.11 KB.

Results: The results pane displays 10 rows of data from the query. The columns are: #, symbol, owner, relationship, date, cost, # shares, value(\$), and total shares. The data is as follows:

#	symbol	owner	relationship	date	cost	# shares	value(\$)	total shares
1	EVR	Walsh Robert B	Principal Financial Officer	"Apr 23"	50	2000	100280	
2	DSNY	Graber Mark A	10% Owner	"Apr 20"	0	13000	7800	

13. QUERYING THE RAW SOURCE DATA STORED IN THE AWS S3 BUCKET THROUGH AWS GLUE INTERFACE USING AWS ATHENA (CONTINUED)

Creating a folder in the current AWS S3 Bucket to store the AWS Athena Query Output Results

The screenshot shows the Amazon S3 console interface. On the left, the navigation pane includes 'Buckets', 'Access Points', 'Object Lambda Access Points', 'Multi-Region Access Points', 'Batch Operations', 'Access analyzer for S3', 'Block Public Access settings for this account', 'Storage Lens' (with 'Dashboards' and 'AWS Organizations settings'), 'Feature spotlight' (with a '3' notification), and 'AWS Marketplace for S3'. The main content area shows the 'vp-stock-trades-bucket' bucket details. The 'Objects' tab is selected, displaying three objects: 'athena_output_query_results/' (selected), 'financial_stocks_data_csv/', and 'financial_stocks_data_txt/'. Below the table are buttons for 'Upload', 'Find objects by prefix', and navigation controls.

Name	Type	Last modified	Size	Storage class
athena_output_query_results/	Folder	-	-	-
financial_stocks_data_csv/	Folder	-	-	-
financial_stocks_data_txt/	Folder	-	-	-

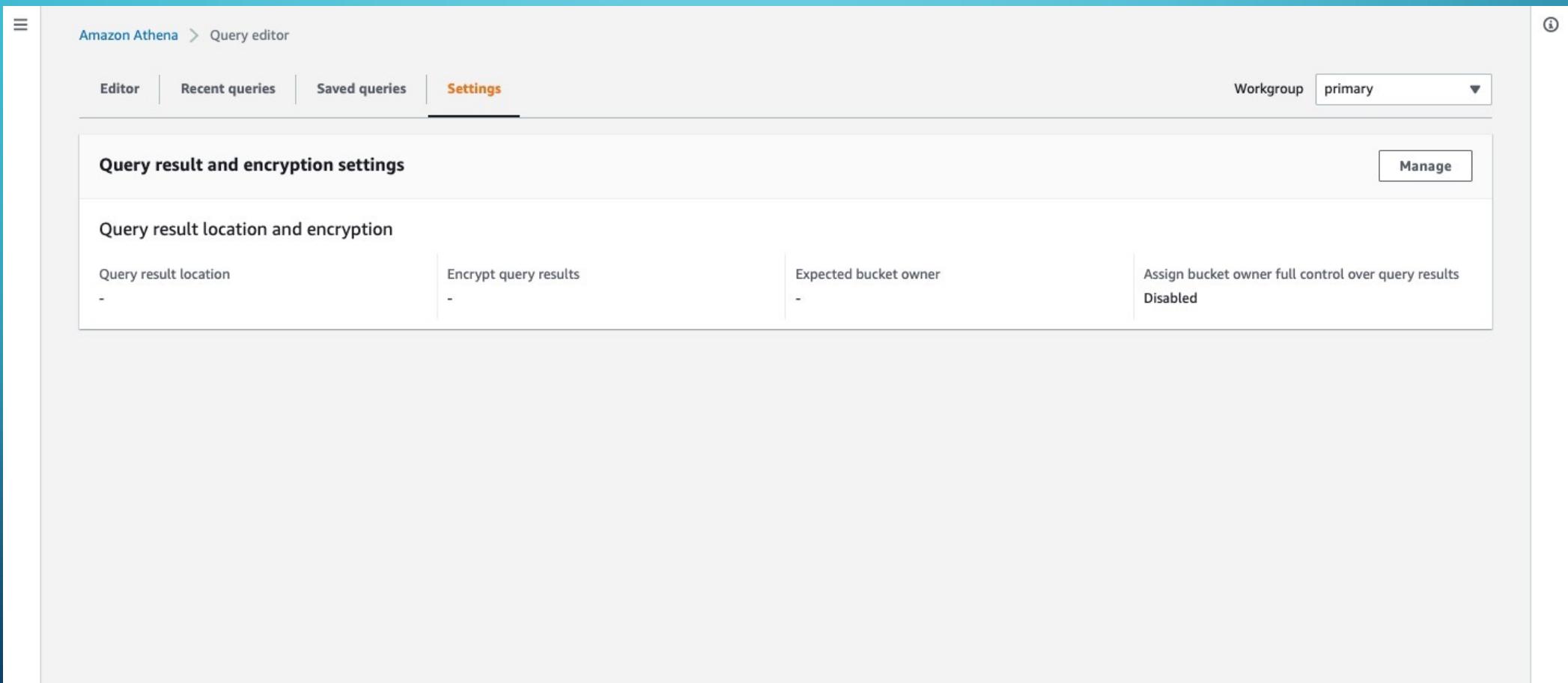
13. QUERYING THE RAW SOURCE DATA STORED IN THE AWS S3 BUCKET THROUGH AWS GLUE INTERFACE USING AWS ATHENA (CONTINUED)

Creating a folder in the current AWS S3 Bucket to store the AWS Athena Query Output Results

The screenshot shows the Amazon S3 console interface. On the left, the navigation pane includes 'Buckets', 'Access Points', 'Object Lambda Access Points', 'Multi-Region Access Points', 'Batch Operations', 'Access analyzer for S3', 'Block Public Access settings for this account', 'Storage Lens' (with 'Dashboards' and 'AWS Organizations settings' sub-options), 'Feature spotlight' (with a '3' notification), and 'AWS Marketplace for S3'. The main content area shows the path 'Amazon S3 > Buckets > vp-stock-trades-bucket > athena_output_query_results/'. The folder name 'athena_output_query_results/' is displayed prominently. A 'Copy S3 URI' button is located in the top right. Below it, tabs for 'Objects' and 'Properties' are shown, with 'Objects' selected. The 'Objects (0)' section contains instructions about objects and permissions, along with buttons for 'Upload', 'Copy S3 URI', 'Copy URL', 'Download', 'Open', 'Delete', 'Actions', and 'Create folder'. A search bar labeled 'Find objects by prefix' is also present. At the bottom, a table header for 'Name', 'Type', 'Last modified', 'Size', and 'Storage class' is shown, followed by the message 'No objects' and 'You don't have any objects in this folder.' A final 'Upload' button is at the bottom right.

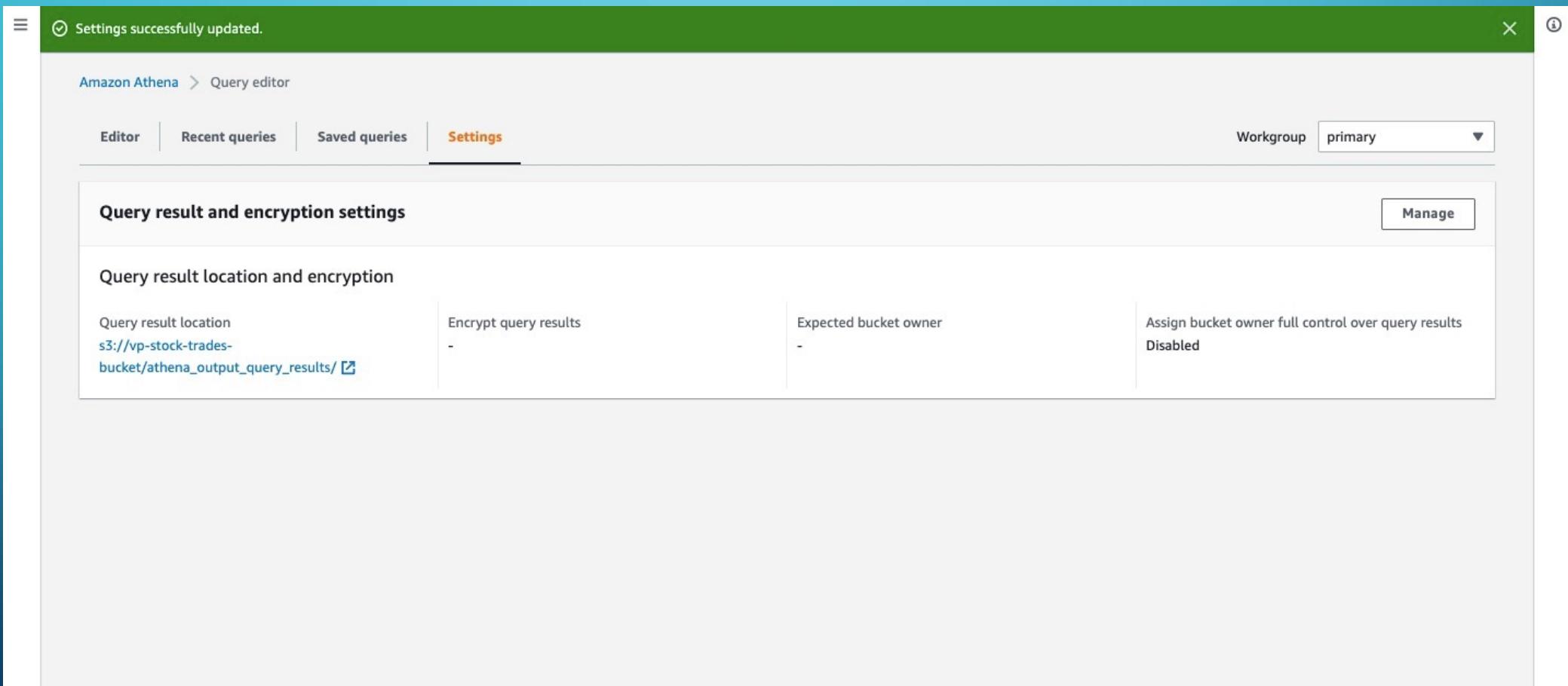
13. QUERYING THE RAW SOURCE DATA STORED IN THE AWS S3 BUCKET THROUGH AWS GLUE INTERFACE USING AWS ATHENA (CONTINUED)

Setting-up the AWS S3 Bucket location to store the AWS Athena Query Output Results in the AWS Athena Settings



13. QUERYING THE RAW SOURCE DATA STORED IN THE AWS S3 BUCKET THROUGH AWS GLUE INTERFACE USING AWS ATHENA (CONTINUED)

Setting-up the AWS S3 Bucket location to store the AWS Athena Query Output Results in the AWS Athena Settings



13. QUERYING THE RAW SOURCE DATA STORED IN THE AWS S3 BUCKET THROUGH AWS GLUE INTERFACE USING AWS ATHENA (CONTINUED)

SQL Query Execution to query and filter the trade transactions for which number of shares processed is less than 10

The screenshot shows the Amazon Athena Query editor interface. The top navigation bar includes tabs for 'Editor' (which is selected), 'Recent queries', 'Saved queries', and 'Settings'. A dropdown for 'Workgroup' is set to 'primary'. On the left, the 'Data' sidebar shows the 'Data Source' as 'AwsDataCatalog' and the 'Database' as 'vp-stock-trades-bucket-database'. Under 'Tables and views', there are two tables listed: 'vp_stock_trades_bucketfinancial_stoc' and 'ks_data_csv'. The 'vp_stock_trades_bucketfinancial_stoc' table has columns: symbol (string), owner (string), relationship (string), date (string), and cost (double). The main workspace displays 'Query 7' with the following SQL query:

```
1 SELECT * FROM "AwsDataCatalog"."vp-stock-trades-bucket-database"."vp_stock_trades_bucketfinancial_stocks_data_csv" WHERE "# shares" < 10
```

The query is currently at 'Ln 1, Col 1'. Below the query are buttons for 'Run', 'Cancel', 'Save', 'Clear', and 'Create'. The 'Results' section shows '(0)' results with a note: 'No results' and 'Run a query to view results'. There are also 'Copy' and 'Download results' buttons.

13. QUERYING THE RAW SOURCE DATA STORED IN THE AWS S3 BUCKET THROUGH AWS GLUE INTERFACE USING AWS ATHENA

SQL Query Execution to query and filter the trade transactions for which number of shares processed is less than 10

The screenshot shows the Amazon Athena Query Editor interface. The left sidebar displays the Data Source (AwsDataCatalog) and Database (vp-stock-trades-bucket-database). The Tables and views section shows two tables: vp_stock_trades_bucketfinancial_stoc ks_data_csv. The main area displays a completed query named "Query 7". The SQL code is:

```
1 SELECT * FROM "AwsDataCatalog"."vp-stock-trades-bucket-database"."vp_stock_trades_bucketfinancial_stocks_data_csv" WHERE "# shares" < 10
```

The results table shows two rows of data:

#	symbol	owner	relationship	date	cost	# shares	value(\$)
1	DSNY	Graber Mark A	10% Owner	"Apr 20		0	13000
2	RVP	SHAW THOMAS J	President and CEO	"Apr 24		3	800

14. VERIFYING THE AWS ATHENA QUERY OUTPUT RESULTS STORED IN TO AWS S3 BUCKET (CONTINUED)

Amazon S3 > Buckets > vp-stock-trades-bucket > athena_output_query_results/

athena_output_query_results/

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](#)

Actions ▾

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	Unsaved/	Folder	-	-	-

14. VERIFYING THE AWS ATHENA QUERY OUTPUT RESULTS STORED IN TO AWS S3 BUCKET (CONTINUED)

The screenshot shows the AWS S3 console interface. The navigation path is: Amazon S3 > Buckets > vp-stock-trades-bucket > athena_output_query_results/ > Unsaved/ > 2022/ > 03/ > 31/. The current view is on the '31/' page. There are two objects listed:

Name	Type	Last modified	Size	Storage class
bfe0ca20-2dfc-4d44-8c50-469783b52534.csv	csv	March 31, 2022, 00:41:53 (UTC+01:00)	25.4 KB	Standard
bfe0ca20-2dfc-4d44-8c50-469783b52534.csv.metadata	metadata	March 31, 2022, 00:41:53 (UTC+01:00)	598.0 B	Standard

14. VERIFYING THE AWS ATHENA QUERY OUTPUT RESULTS STORED IN TO AWS S3 BUCKET (CONTINUED)

The screenshot shows the AWS S3 Object Overview page for the file `bfe0ca20-2dfc-4d44-8c50-469783b52534.csv.metadata`. The page displays various metadata details such as Owner, AWS Region, Last modified, Size, Type, and Key. It also provides links for S3 URI, Amazon Resource Name (ARN), Entity tag (Etag), and Object URL.

Object overview

Key	Value
Owner	4d97ff8f9dfc260e11d18083dc29d77931845b8a284d61ae03f6323dc4a5b186
AWS Region	EU (London) eu-west-2
Last modified	March 31, 2022, 00:41:53 (UTC+01:00)
Size	598.0 B
Type	metadata
Key	athena_output_query_results/Unsaved/2022/03/31/bfe0ca20-2dfc-4d44-8c50-469783b52534.csv.metadata

S3 URI

s3://vp-stock-trades-bucket/athena_output_query_results/Unsaved/2022/03/31/bfe0ca20-2dfc-4d44-8c50-469783b52534.csv.metadata

Amazon Resource Name (ARN)

arn:aws:s3:::vp-stock-trades-bucket/athena_output_query_results/Unsaved/2022/03/31/bfe0ca20-2dfc-4d44-8c50-469783b52534.csv.metadata

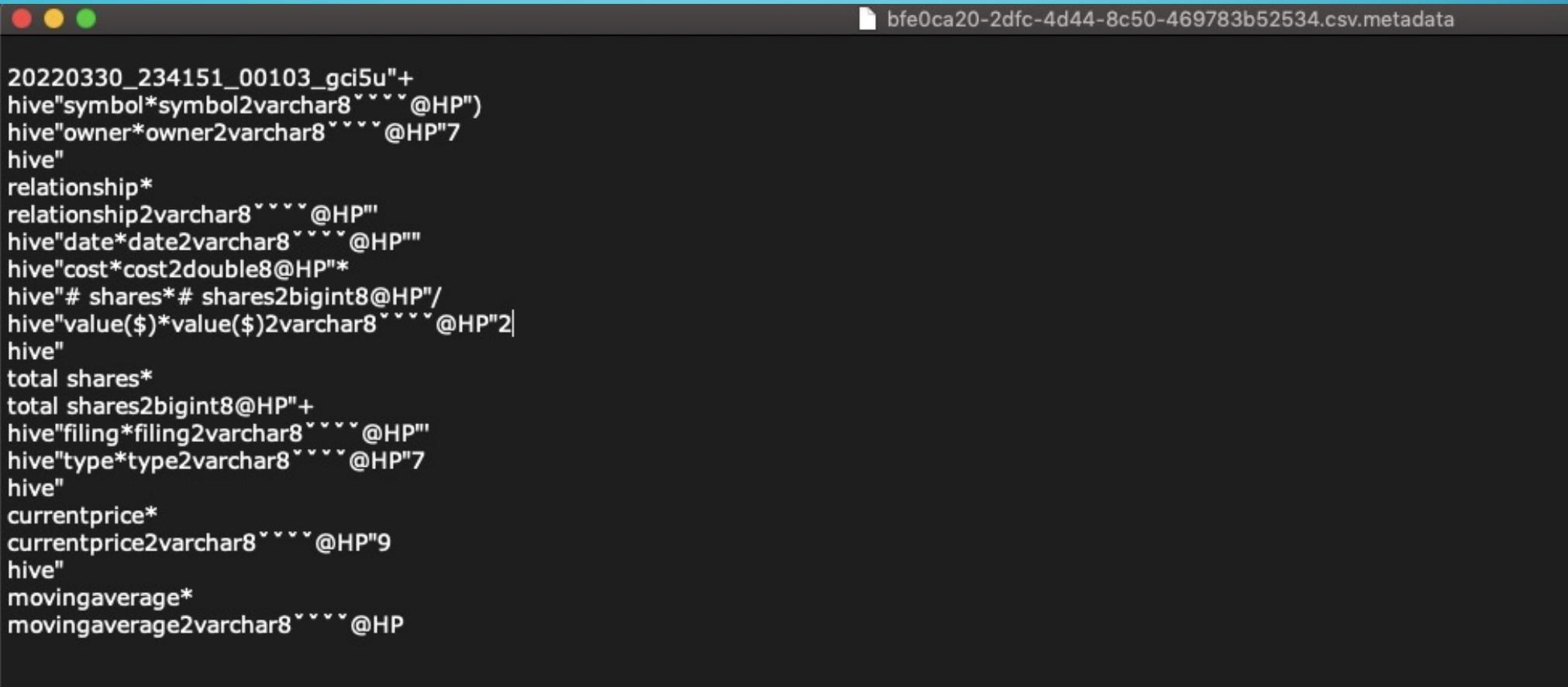
Entity tag (Etag)

[fb8f77e04d3aea7a6d0cf10230b703c9](#)

Object URL

https://vp-stock-trades-bucket.s3.eu-west-2.amazonaws.com/athena_output_query_results/Unsaved/2022/03/31/bfe0ca20-2dfc-4d44-8c50-469783b52534.csv.metadata

14. VERIFYING THE AWS ATHENA QUERY OUTPUT RESULTS STORED IN TO AWS S3 BUCKET (CONTINUED)



The screenshot shows a terminal window with a dark background and light-colored text. The title bar of the window reads "bfe0ca20-2dfc-4d44-8c50-469783b52534.csv.metadata". The window contains the following text:

```
20220330_234151_00103_gci5u"+
hive"symbol*symbol2varchar8***@HP")
hive"owner*owner2varchar8***@HP"7
hive"
relationship*
relationship2varchar8***@HP"
hive"date*date2varchar8***@HP"""
hive"cost*cost2double8@HP"*
hive"# shares*# shares2bigint8@HP"/
hive"value($)*value($)2varchar8***@HP"2
hive"
total shares*
total shares2bigint8@HP"+
hive"filing*filing2varchar8***@HP"
hive"type*type2varchar8***@HP"7
hive"
currentprice*
currentprice2varchar8***@HP"9
hive"
movingaverage*
movingaverage2varchar8***@HP
```

14. VERIFYING THE AWS ATHENA QUERY OUTPUT RESULTS STORED IN TO AWS S3 BUCKET (CONTINUED)

The screenshot shows the AWS S3 Object Properties page for a file named `bfe0ca20-2dfc-4d44-8c50-469783b52534.csv`. The file was created in the `athena_output_query_results/Unsaved/2022/03/31/` directory of the `vp-stock-trades-bucket`.

Properties tab selected.

Object overview

Attribute	Value
Owner	4d97ff8f9dfc260e11d18083dc29d77931845b8a284d61ae03f6323dc4a5b186
AWS Region	EU (London) eu-west-2
Last modified	March 31, 2022, 00:41:53 (UTC+01:00)
Size	25.4 KB
Type	csv
Key	athena_output_query_results/Unsaved/2022/03/31/bfe0ca20-2dfc-4d44-8c50-469783b52534.csv

Actions

- Copy S3 URI
- Download
- Open
- Object actions ▾

S3 URI: `s3://vp-stock-trades-bucket/athena_output_query_results/Unsaved/2022/03/31/bfe0ca20-2dfc-4d44-8c50-469783b52534.csv`

Amazon Resource Name (ARN): `arn:aws:s3:::vp-stock-trades-bucket/athena_output_query_results/Unsaved/2022/03/31/bfe0ca20-2dfc-4d44-8c50-469783b52534.csv`

Entity tag (Etag): `3f68334897281d32a7232ce975d55a43`

Object URL: `https://vp-stock-trades-bucket.s3.eu-west-2.amazonaws.com/athena_output_query_results/Unsaved/2022/03/31/bfe0ca20-2dfc-4d44-8c50-469783b52534.csv`

14. VERIFYING THE AWS ATHENA QUERY OUTPUT RESULTS STORED IN TO AWS S3 BUCKET



AWS GLUE ETL JOB WITH SPARK TRANSFORMATION

15. CREATING A NEW AWS S3 BUCKET TO STORE THE TRANSFORMED DATA (CONTINUED)

The screenshot shows the AWS S3 Buckets page. On the left, there's a sidebar with options like Buckets, Storage Lens, and Feature spotlight. The main area displays an account snapshot and a list of buckets.

Buckets

- Access Points
- Object Lambda Access Points
- Multi-Region Access Points
- Batch Operations
- Access analyzer for S3

Block Public Access settings for this account

Storage Lens

- Dashboards
- AWS Organizations settings

Feature spotlight (3)

AWS Marketplace for S3

Account snapshot

Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

Buckets (2) [Info](#)

Buckets are containers for data stored in S3. [Learn more](#)

Name	AWS Region	Access	Creation date
vp-stock-trades-bucket	EU (London) eu-west-2	Objects can be public	March 30, 2022, 16:45:42 (UTC+01:00)
vp-stock-trades-transformed-bucket	EU (London) eu-west-2	Objects can be public	March 31, 2022, 22:08:11 (UTC+01:00)

[View Storage Lens dashboard](#)

Action buttons: [C](#), [Copy ARN](#), [Empty](#), [Delete](#), [Create bucket](#)

15. CREATING A NEW AWS S3 BUCKET TO STORE THE TRANSFORMED DATA

The screenshot shows the Amazon S3 console interface. On the left, a sidebar menu includes options like 'Buckets', 'Access Points', 'Object Lambda Access Points', 'Multi-Region Access Points', 'Batch Operations', 'Access analyzer for S3', 'Block Public Access settings for this account', 'Storage Lens' (with 'Dashboards' and 'AWS Organizations settings' sub-options), 'Feature spotlight' (with a '3' badge), and 'AWS Marketplace for S3'. The main content area displays the details for the bucket 'vp-stock-trades-transformed-bucket'. The top navigation bar shows the path 'Amazon S3 > Buckets > vp-stock-trades-transformed-bucket'. Below the path, the bucket name is displayed with an 'Info' link. A tab bar at the top of the content area includes 'Objects' (which is selected and highlighted in orange), 'Properties', 'Permissions', 'Metrics', 'Management', and 'Access Points'. The 'Objects' section contains a heading 'Objects (0)', a descriptive text about objects in S3, and a toolbar with buttons for 'Copy S3 URI', 'Copy URL', 'Download', 'Open', 'Delete', 'Actions', 'Create folder', and 'Upload'. A search bar below the toolbar allows for 'Find objects by prefix'. The main table area has columns for 'Name', 'Type', 'Last modified', 'Size', and 'Storage class'. A message 'No objects' indicates there are currently no objects in the bucket. At the bottom of the table area is a large 'Upload' button.

16. CREATION OF FOLDERS IN THE AWS S3 NEW BUCKET (TRANSFORMATION DATA BUCKET)

Three directories to store the AWS Glue Spark ETL Job Python Script, to store the temporary intermediate results and to store the spark job event logs respectively

The screenshot shows the Amazon S3 console interface. On the left, there is a navigation sidebar with the following options:

- Buckets
- Access Points
- Object Lambda Access Points
- Multi-Region Access Points
- Batch Operations
- Access analyzer for S3
- Block Public Access settings for this account
- Storage Lens (expanded, showing Dashboards and AWS Organizations settings)
- Feature spotlight (with a '3' badge)
- AWS Marketplace for S3

The main content area displays the details for the bucket "vp-stock-trades-transformed-bucket". The URL is "Amazon S3 > Buckets > vp-stock-trades-transformed-bucket". The tab "Objects" is selected. The page title is "vp-stock-trades-transformed-bucket" with an "Info" link. Below the title, there are tabs for Properties, Permissions, Metrics, Management, and Access Points.

The "Objects (3)" section contains the following table:

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	glue_spark_etl_event_logs/	Folder	-	-	-
<input type="checkbox"/>	glue_spark_etl_python_scripts_temp_results/	Folder	-	-	-
<input type="checkbox"/>	glue_spark_etl_python_scripts/	Folder	-	-	-

17. CREATION OF AN IAM ROLE TO GRANT PERMISSIONS TO AWS GLUE CRAWLER TO ACCESS AWS S3 (CONTINUED)

IAM Role with Read and Write permissions to access AWS S3 Bucket

The screenshot shows the AWS Identity and Access Management (IAM) service interface. On the left, the navigation pane includes options like Dashboard, User groups, Users, Roles (which is selected and highlighted in orange), Policies, Identity providers, and Account settings. Below these are sections for Access management and Access reports, each with their own sub-options.

The main content area displays a success message: "Role VP_Glue_Crawler_Service_S3_Full_Access_Role created". The title of the page is "Roles (Selected 1/12)". A search bar at the top right contains the role name "VP_Glue_Crawler_Service_S3_Full_Access_Role", which has resulted in 1 match. The table below lists the role details:

Role name	Trusted entities	Last activity
VP_Glue_Crawler_Service_S3_Full_Access_Role	AWS Service: glue	-

At the top right of the main content area, there are buttons for "View role", "Delete", and "Create role".

17. CREATION OF AN IAM ROLE TO GRANT PERMISSIONS TO AWS GLUE CRAWLER TO ACCESS AWS S3

IAM Role with Read and Write permissions to access AWS S3 Bucket

The screenshot shows the AWS Identity and Access Management (IAM) service interface. On the left, the navigation pane is visible with sections like 'Access management' (selected), 'Roles' (highlighted in orange), and 'Permissions policies'. The main content area displays the details of a specific role:

VP_Glue_Crawler_Service_S3_Full_Access_Role
Allows Glue Service with both read and write permissions to access the folders and files in the AWS S3 Bucket.

Summary

Creation date	ARN
March 31, 2022, 22:22 (UTC+01:00)	arn:aws:iam::146871189787:role/VP_Glue_Crawler_Service_S3_Full_Access_Role
Last activity	Maximum session duration
None	1 hour

Permissions (Selected tab) | Trust relationships | Tags | Access Advisor | Revoke sessions

Permissions policies (2)
You can attach up to 10 managed policies.

Policy name	Type	Description
AmazonS3FullAccess	AWS managed policy	Provides full access to all buckets via the AWS Management Console.
AWSGlueServiceRole	AWS managed policy	Policy for AWS Glue service role which allows access to related services including EC2, S3, and ...

18. CREATION OF AWS GLUE ETL SPARK JOB WITH TRANSFORMATION LOGIC (CONTINUED)

Transformation Logic will be applied on the source raw data file “trade_transactions.csv” stored in AWS S3 Bucket ‘vp-stock-trades-bucket’

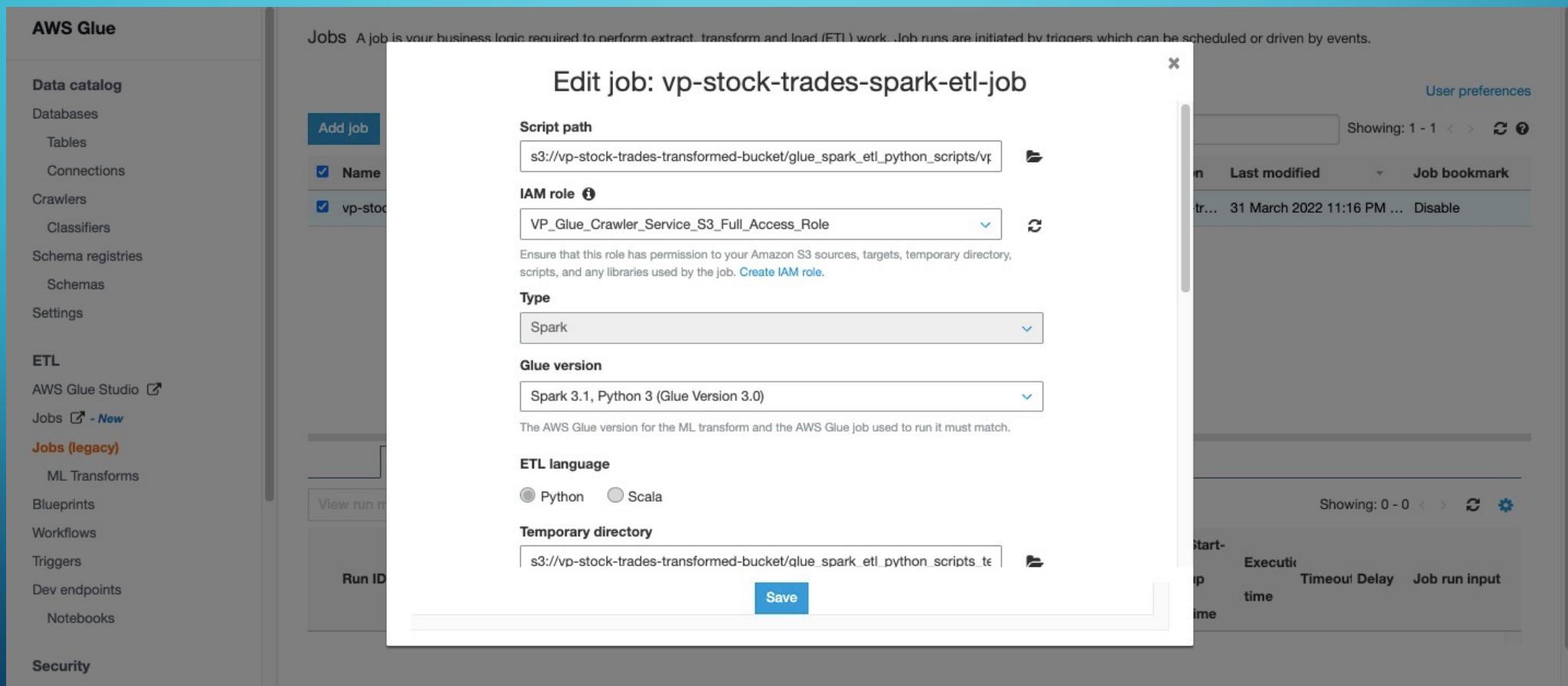
The screenshot shows the AWS Glue console interface. On the left, there is a navigation sidebar with the following sections:

- Data catalog**: Databases, Tables, Connections, Crawlers, Classifiers, Schema registries, Schemas, Settings.
- ETL**: AWS Glue Studio (checkbox), Jobs (checkbox - New), **Jobs (legacy)**: ML Transforms, Blueprints, Workflows, Triggers, Dev endpoints, Notebooks.
- Security**: IAM users, Groups, Policies.

The main content area is titled "Jobs" with a sub-instruction: "A job is your business logic required to perform extract, transform and load (ETL) work. Job runs are initiated by triggers which can be scheduled or driven by events." It includes a "User preferences" link and a search bar with filters for "Name", "Type", "ETL language", "Script location", "Last modified", and "Job bookmark". A button labeled "Add job" is prominently displayed. Below the search bar, a message states "Showing: 0 - 0 < > ⌂ ⓘ" and "You don't have any jobs defined yet." with a corresponding icon.

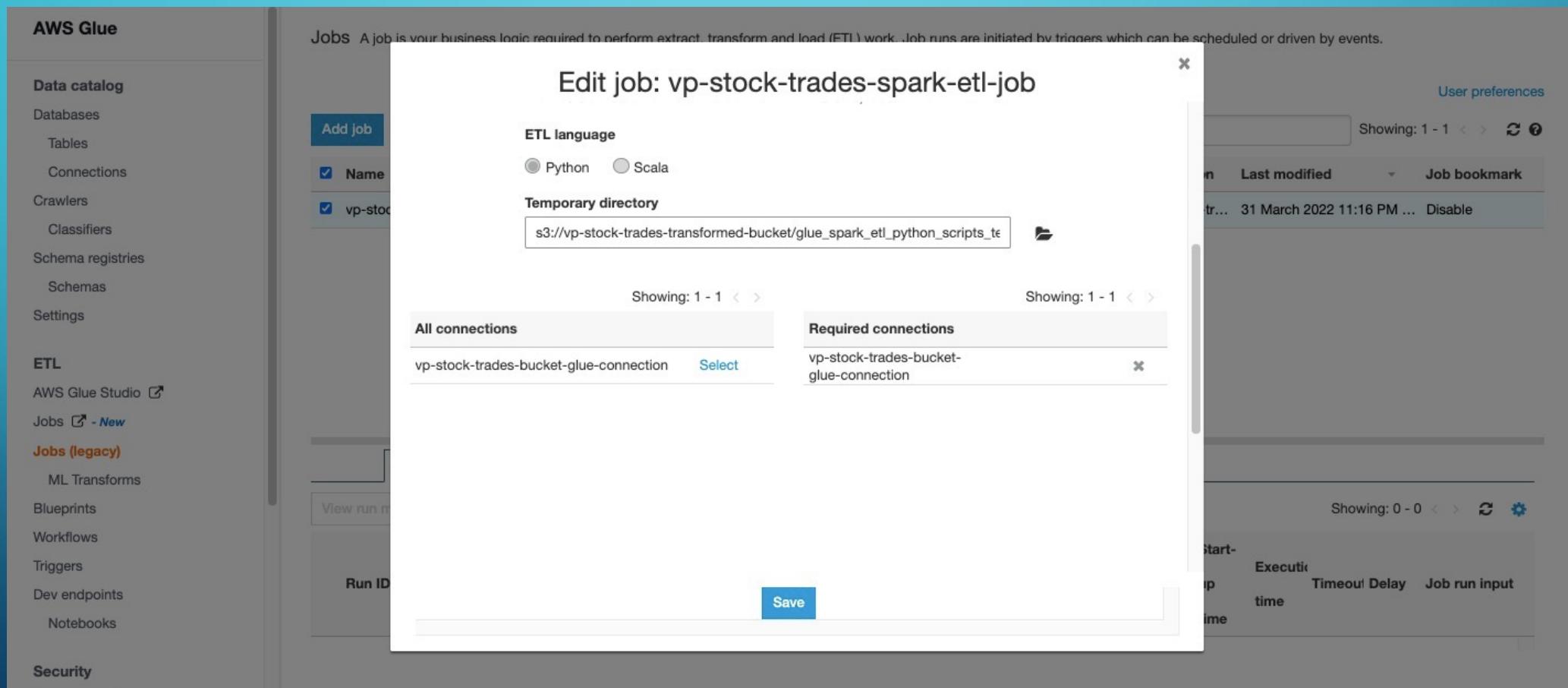
18. CREATION OF AWS GLUE ETL SPARK JOB WITH TRANSFORMATION LOGIC (CONTINUED)

Transformation Logic will be applied on the source raw data file “trade_transactions.csv” stored in AWS S3 Bucket ‘vp-stock-trades-bucket’



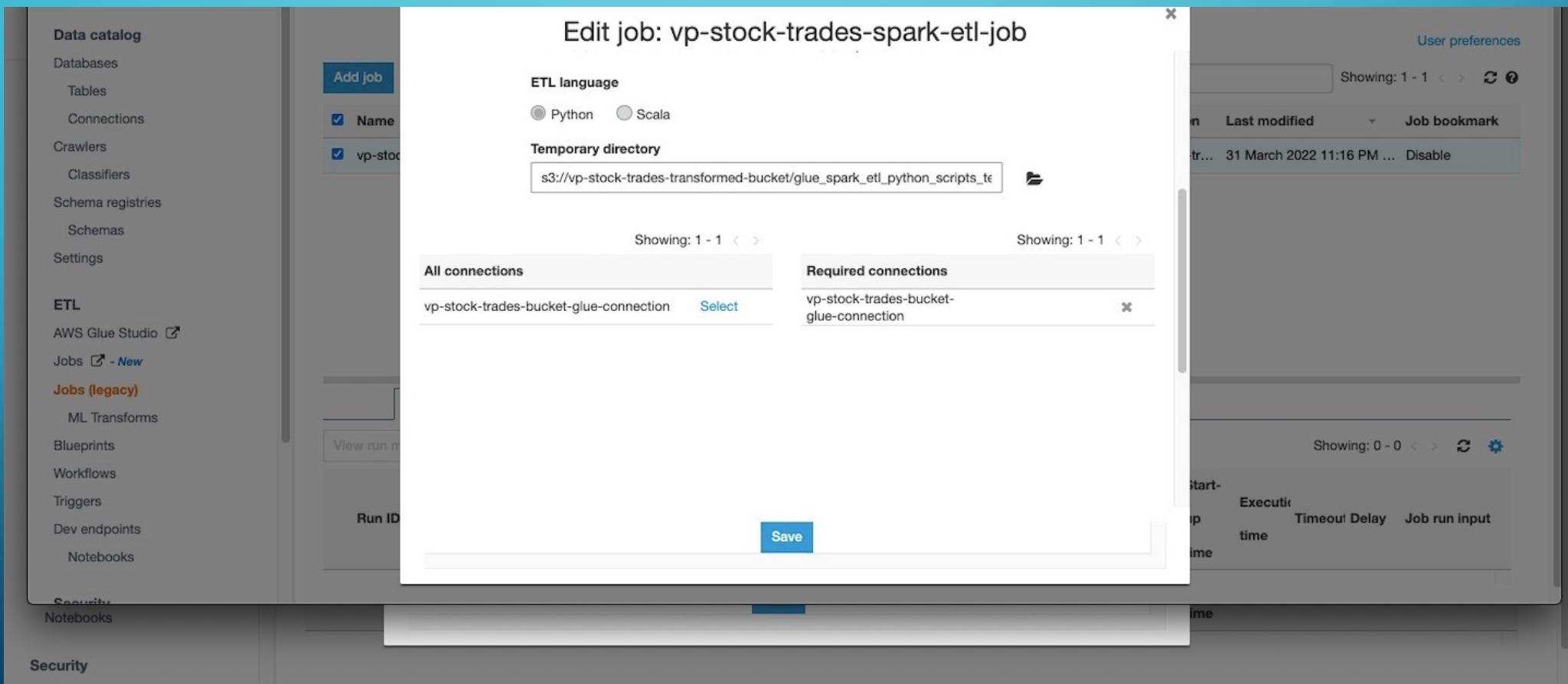
18. CREATION OF AWS GLUE ETL SPARK JOB WITH TRANSFORMATION LOGIC (CONTINUED)

Transformation Logic will be applied on the source raw data file “trade_transactions.csv” stored in AWS S3 Bucket ‘vp-stock-trades-bucket’



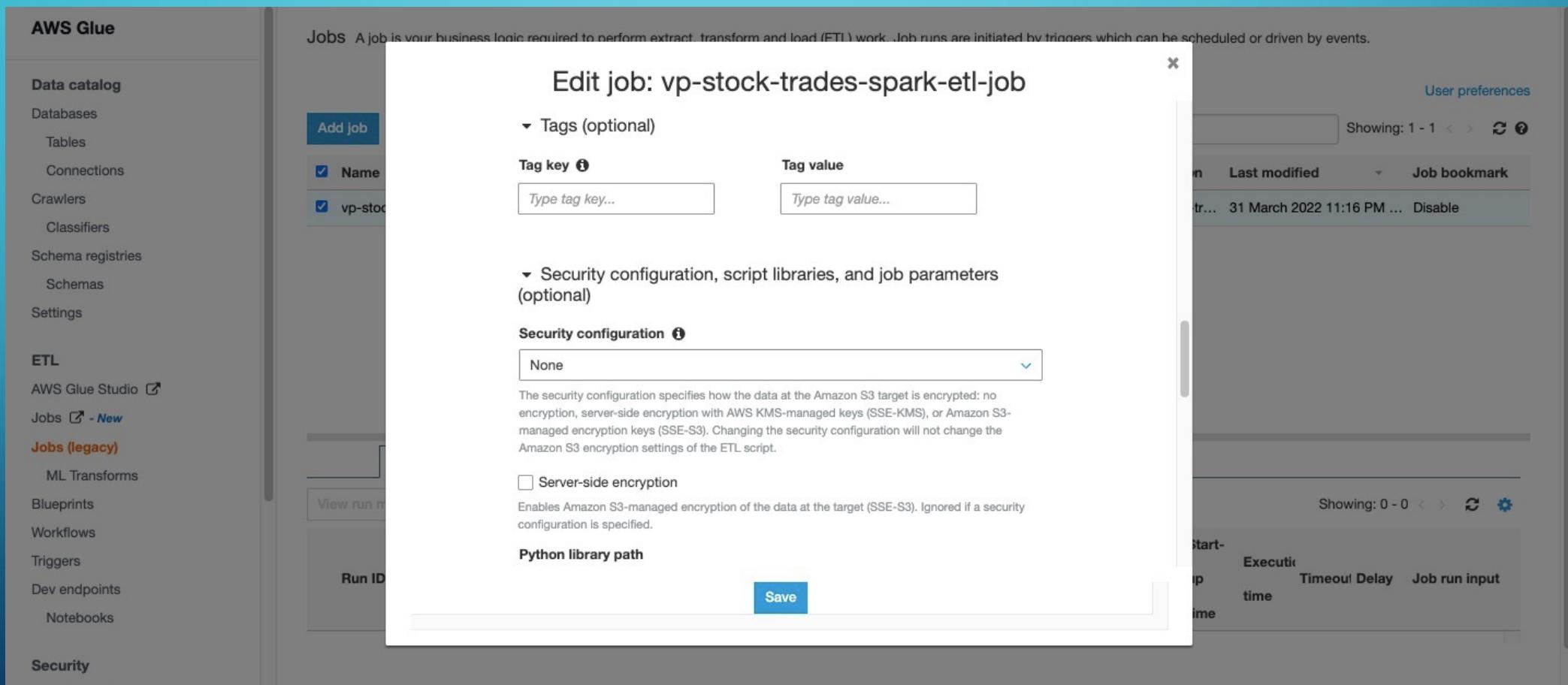
18. CREATION OF AWS GLUE ETL SPARK JOB WITH TRANSFORMATION LOGIC (CONTINUED)

Transformation Logic will be applied on the source raw data file “trade_transactions.csv” stored in AWS S3 Bucket ‘vp-stock-trades-bucket’



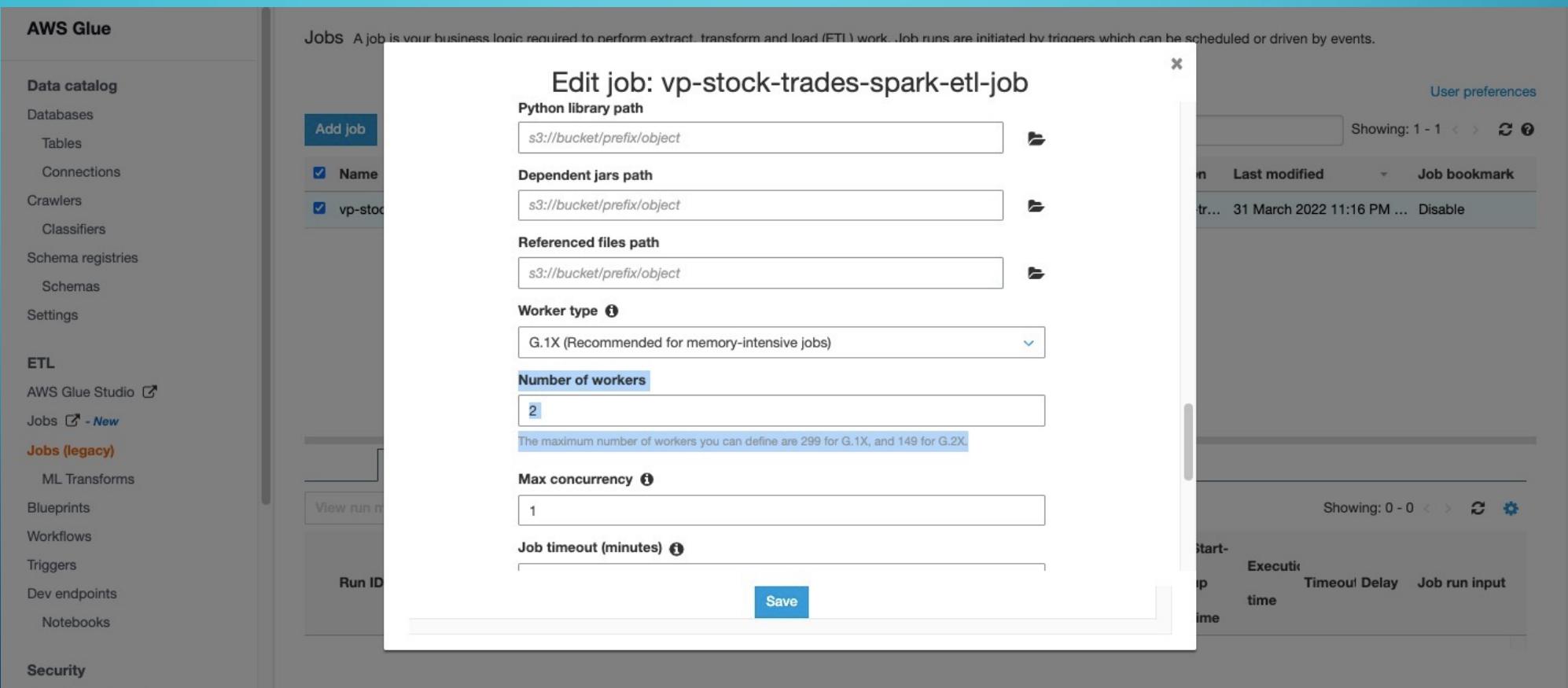
18. CREATION OF AWS GLUE ETL SPARK JOB WITH TRANSFORMATION LOGIC (CONTINUED)

Transformation Logic will be applied on the source raw data file “trade_transactions.csv” stored in AWS S3 Bucket ‘vp-stock-trades-bucket’



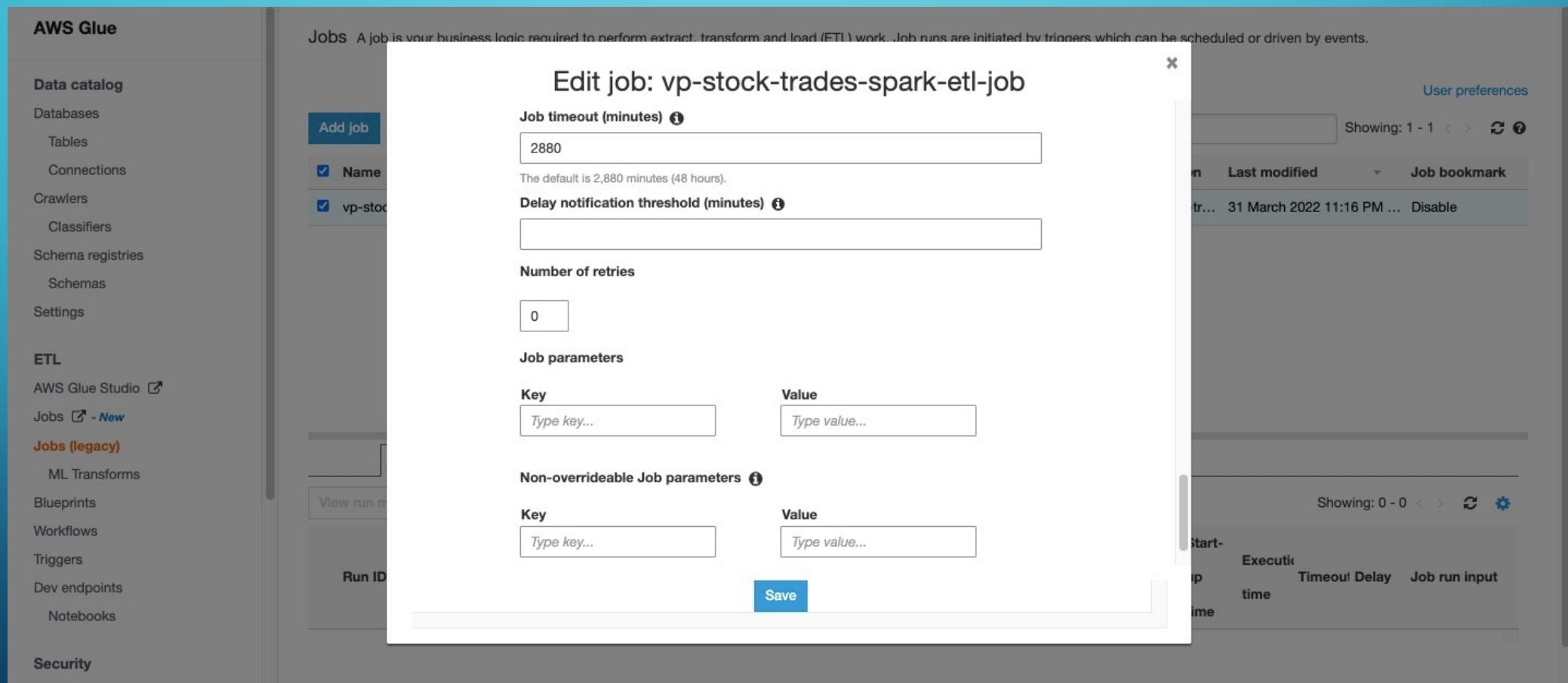
18. CREATION OF AWS GLUE ETL SPARK JOB WITH TRANSFORMATION LOGIC (CONTINUED)

Transformation Logic will be applied on the source raw data file “trade_transactions.csv” stored in AWS S3 Bucket ‘vp-stock-trades-bucket’



18. CREATION OF AWS GLUE ETL SPARK JOB WITH TRANSFORMATION LOGIC (CONTINUED)

Transformation Logic will be applied on the source raw data file “trade_transactions.csv” stored in AWS S3 Bucket ‘vp-stock-trades-bucket’



18. CREATION OF AWS GLUE ETL SPARK JOB WITH TRANSFORMATION LOGIC

Transformation Logic will be applied on the source raw data file “trade_transactions.csv” stored in AWS S3 Bucket ‘vp-stock-trades-bucket’

The screenshot shows the AWS Glue Jobs console. On the left, there's a sidebar with navigation links for Data catalog, ETL (AWS Glue Studio, Jobs - New, Jobs (legacy), ML Transforms, Blueprints, Workflows, Triggers, Dev endpoints, Notebooks, Security), and a link to AWS Lambda. The main area is titled "Jobs" with a sub-instruction: "A job is your business logic required to perform extract, transform and load (ETL) work. Job runs are initiated by triggers which can be scheduled or driven by events." It includes a search bar with "Filter by tags and attributes" and a table header with columns: Name, Type, ETL language, Script location, Last modified, and Job bookmark. A single row is listed: "vp-stock-trades-spark-etl-job" (Spark, python, s3://vp-stock-trades-bucket, 31 March 2022 11:16 PM, Disable). Below this is a "History" tab with sub-links "View run metrics" and "Rewind job bookmark". The table under "History" has columns: Run ID, Retry attempt status, Error, Output, Logs, Error logs, Glue version, Maximum capacity, Triggered by, Start time, End time, Start-up time, Executive time, Timeout delay, and Job run input. A message "No job runs found" is displayed.

19. A COPY OF AWS GLUE ETL SPARK JOB ETL SCRIPT WITH TRANSFORMATION LOGIC SAVED INTO THE SPECIFIED PATH OF THE S3 BUCKET (CONTINUED)

Amazon S3

Buckets

Access Points

Object Lambda Access Points

Multi-Region Access Points

Batch Operations

Access analyzer for S3

Block Public Access settings for this account

Storage Lens

Dashboards

AWS Organizations settings

Feature spotlight 3

AWS Marketplace for S3

Amazon S3 > Buckets > vp-stock-trades-transformed-bucket > glue_spark_etl_python_scripts/

glue_spark_etl_python_scripts/

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](#)

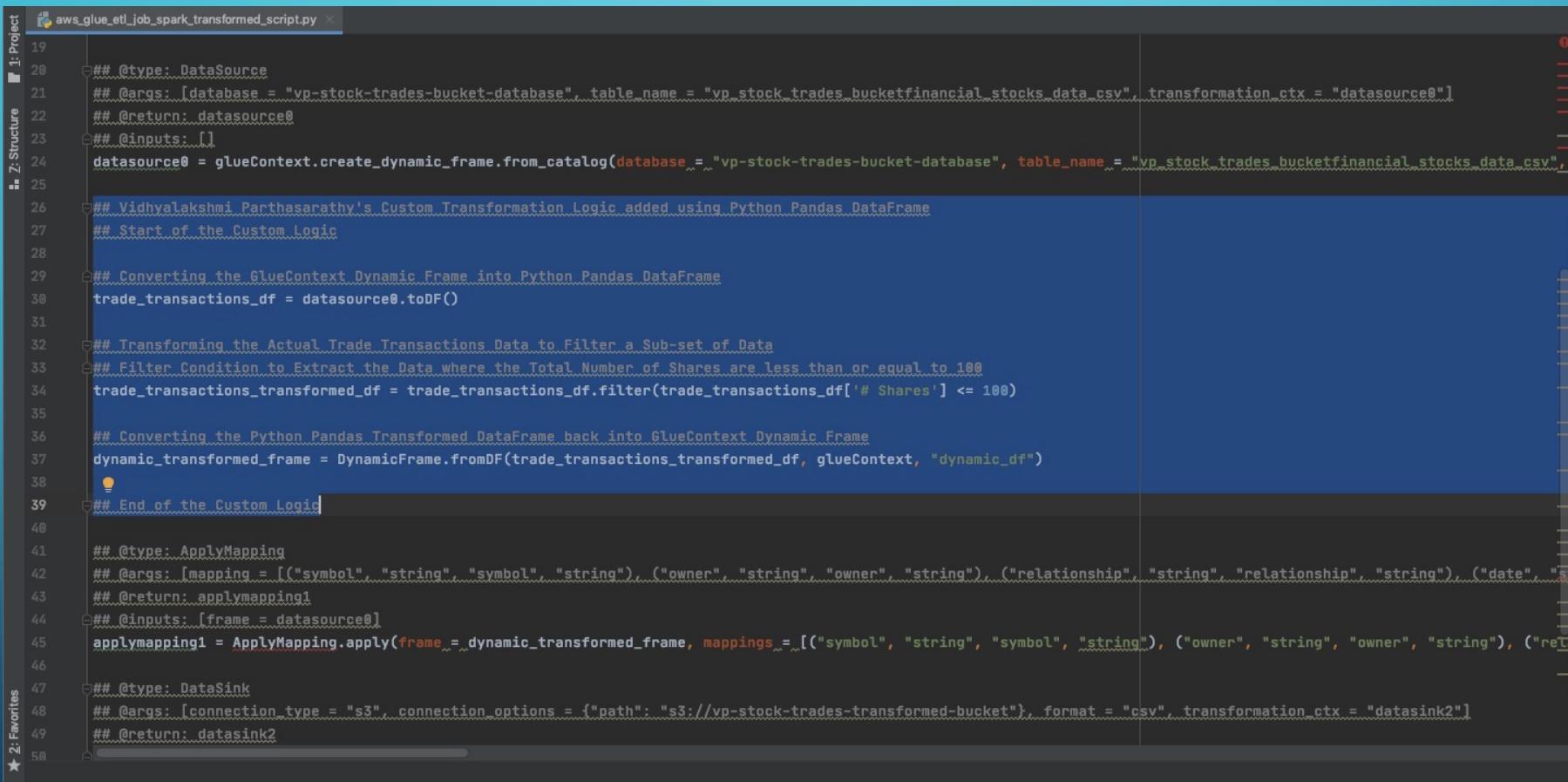
C Copy S3 URI Copy URL Download Open Delete Actions Create folder Upload

Find objects by prefix < 1 >

Name	Type	Last modified	Size	Storage class
vp-stock-trades-spark-etl-job-python-script	-	April 1, 2022, 00:04:10 (UTC+01:00)	3.5 KB	Standard
vp-stock-trades-spark-etl-job-python-script.temp	temp	April 1, 2022, 00:03:57 (UTC+01:00)	3.5 KB	Standard

19. A COPY OF AWS GLUE ETL SPARK JOB ETL SCRIPT WITH TRANSFORMATION LOGIC SAVED INTO THE SPECIFIED PATH OF THE S3 BUCKET

Transformation Logic to filter the trade transactions with the total number of shares ('# Shares') as less than or equal to 100. Expected Results as 33 Trade Transactions (Rows) in the output transformed data file.



The screenshot shows a code editor window with the file name "aws_glue_etl_job_spark_transformed_script.py". The code is a Python script for an AWS Glue ETL job. It starts by creating a dynamic frame from a database table, then applies custom transformation logic to filter trade transactions where the total number of shares is less than or equal to 100. Finally, it applies an mapping and saves the results to an S3 bucket.

```
aws_glue_etl_job_spark_transformed_script.py
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
```

```
## @type: DataSource
## @args: [database = "vp-stock-trades-bucket-database", table_name = "vp_stock_trades_bucketfinancial_stocks_data.csv", transformation_ctx = "datasource0"]
## @return: datasource0
## @inputs: []
datasource0 = glueContext.create_dynamic_frame.from_catalog(database = "vp-stock-trades-bucket-database", table_name = "vp_stock_trades_bucketfinancial_stocks_data.csv", transformation_ctx = "datasource0")
## Vidhyalakshmi Parthasarathy's Custom Transformation Logic added using Python Pandas DataFrame
## Start of the Custom Logic

## Converting the GlueContext Dynamic Frame into Python Pandas DataFrame
trade_transactions_df = datasource0.toDF()

## Transforming the Actual Trade Transactions Data to Filter a Sub-set of Data
## Filter Condition to Extract the Data where the Total Number of Shares are less than or equal to 100
trade_transactions_transformed_df = trade_transactions_df.filter(trade_transactions_df['# Shares'] <= 100)

## Converting the Python Pandas Transformed DataFrame back into GlueContext Dynamic Frame
dynamic_transformed_frame = DynamicFrame.fromDF(trade_transactions_transformed_df, glueContext, "dynamic_df")

## End of the Custom Logic

## @type: ApplyMapping
## @args: [mapping = [{"symbol": "string", "symbol": "string"}, {"owner": "string", "owner": "string"}, {"relationship": "string", "relationship": "string"}, {"date": "string", "date": "string"}], transformation_ctx = "applymapping1"]
## @return: applymapping1
## @inputs: [frame = datasource0]
applymapping1 = ApplyMapping.apply(frame = dynamic_transformed_frame, mappings = [{"symbol": "string", "symbol": "string"}, {"owner": "string", "owner": "string"}, {"relationship": "string", "relationship": "string"}, {"date": "string", "date": "string"}], transformation_ctx = "applymapping1")

## @type: DataSink
## @args: [connection_type = "s3", connection_options = {"path": "s3://vp-stock-trades-transformed-bucket"}, format = "csv", transformation_ctx = "datasink2"]
## @return: datasink2
```

20. EXECUTION OF AWS GLUE ETL SPARK JOB WITH TRANSFORMATION LOGIC (CONTINUED)

Transformation Logic will be applied on the source raw data file “trade_transactions.csv” stored in AWS S3 Bucket ‘vp-stock-trades-bucket’

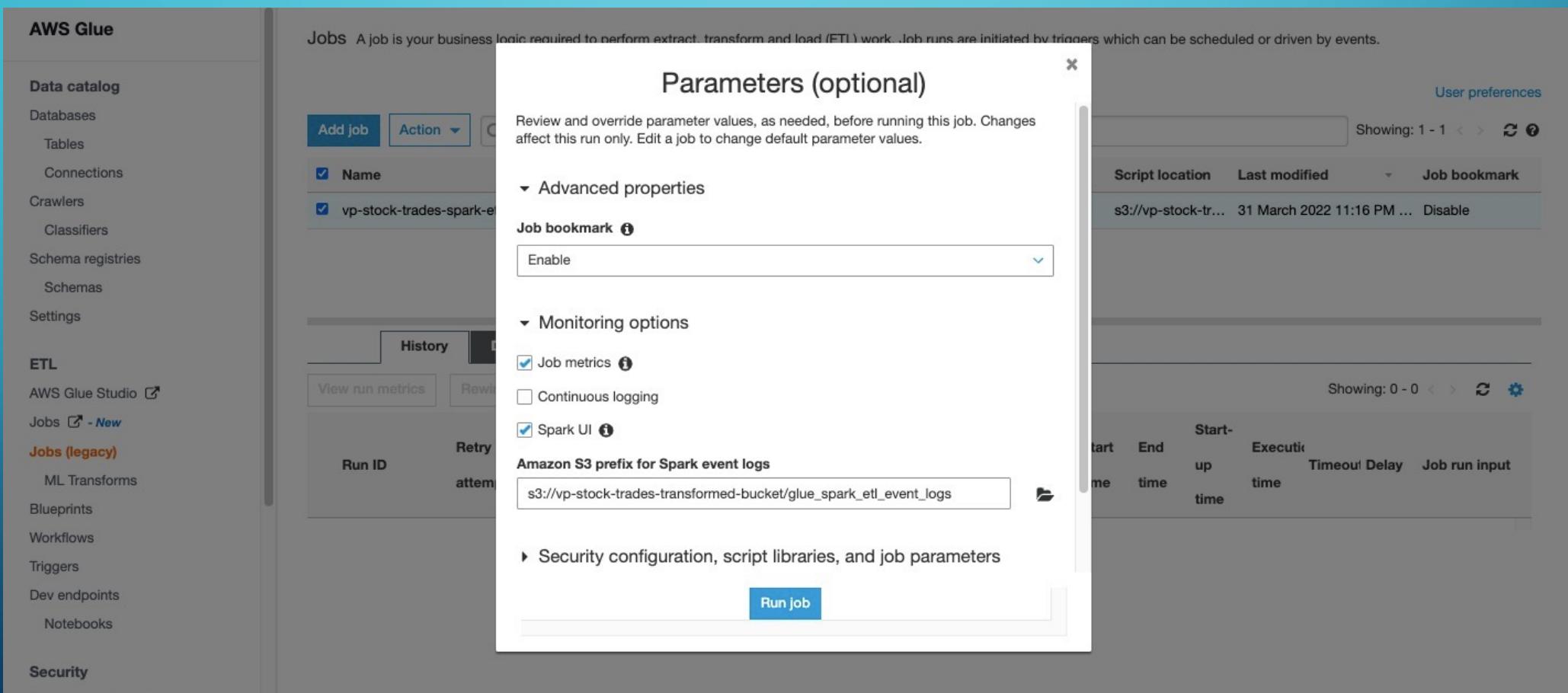
The screenshot shows the AWS Glue Jobs console. On the left, there's a sidebar with navigation links for Data catalog, ETL, and Security. Under ETL, 'Jobs - New' is selected. A context menu is open over a job named 'vp-stock'. The menu options are: Run job, Stop job run, Choose job triggers, Delete, Edit job, Edit script, Reset job bookmark, and Create development endpoint. The main table lists one job entry:

Run ID	Retry attempt status	Error	Output logs	Error logs	Glue version	Maximum capacity	Triggered by	Start time	End time	Start-up time	Execution time	Timeout delay	Job run input
													No job runs found

At the top right of the table, it says 'Showing: 1 - 1' and has navigation icons. Below the table, it says 'Showing: 0 - 0'.

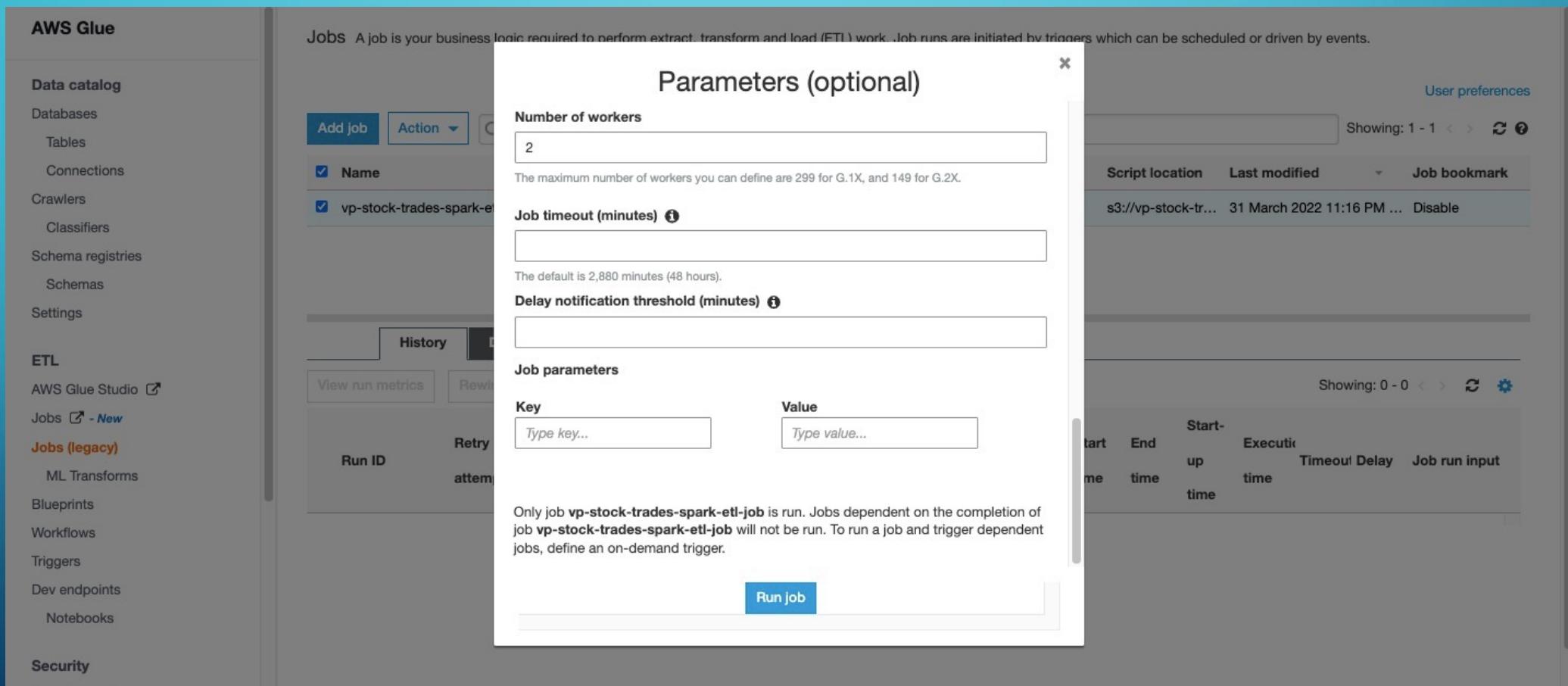
20. EXECUTION OF AWS GLUE ETL SPARK JOB WITH TRANSFORMATION LOGIC (CONTINUED)

Transformation Logic will be applied on the source raw data file “trade_transactions.csv” stored in AWS S3 Bucket ‘vp-stock-trades-bucket’



20. EXECUTION OF AWS GLUE ETL SPARK JOB WITH TRANSFORMATION LOGIC (CONTINUED)

Transformation Logic will be applied on the source raw data file “trade_transactions.csv” stored in AWS S3 Bucket ‘vp-stock-trades-bucket’



20. EXECUTION OF AWS GLUE ETL SPARK JOB WITH TRANSFORMATION LOGIC (CONTINUED)

Transformation Logic will be applied on the source raw data file “trade_transactions.csv” stored in AWS S3 Bucket ‘vp-stock-trades-bucket’

The screenshot shows the AWS Glue interface with the 'Jobs' section selected. A message at the top states "Job 'vp-stock-trades-spark-etl-job' is now running." Below this, a table lists the job details:

Name	Type	ETL language	Script location	Last modified	Job bookmark
vp-stock-trades-spark-etl-job	Spark	python	s3://vp-stock-tr...	31 March 2022 11:16 PM ...	Disable

20. EXECUTION OF AWS GLUE ETL SPARK JOB WITH TRANSFORMATION LOGIC (CONTINUED)

Job Execution is in Progress

The screenshot shows the AWS Glue console interface. On the left, the navigation menu includes sections for Data catalog, ETL (AWS Glue Studio, Jobs - New, Jobs (legacy), ML Transforms, Blueprints, Workflows, Triggers, Dev endpoints, Notebooks), and Security. The main content area displays the 'Jobs' page, which defines a job as business logic for ETL work. A search bar at the top right allows filtering by tags and attributes. The table lists one job entry:

Name	Type	ETL language	Script location	Last modified	Job bookmark
vp-stock-trades-spark-etl-job	Spark	python	s3://vp-stock-tr...	31 March 2022 11:16 PM ...	Disable

Below the table, tabs for History, Details, Script, and Metrics are visible. The History tab shows a single run entry:

Run ID	Retry attempt status	Error	Output Logs	Error logs	Glue version	Maximum capacity	Triggered by	Start time	End time	Start-up time	Execution time	Timeout Delay	Job run input
jr_0aeda887a...	-	Running	Logs	Error logs	3.0	2		1 Ap...	0 secs	45 secs	2880 mins		s3://vp-stock...

20. EXECUTION OF AWS GLUE ETL SPARK JOB WITH TRANSFORMATION LOGIC (CONTINUED)

Job Execution is Completed and Successful

The screenshot shows the AWS Glue console interface. On the left, the navigation menu includes sections for Data catalog (Databases, Tables, Connections), Crawlers, Classifiers, Schema registries, Schemas, Settings, and ETL (AWS Glue Studio, Jobs - New, Jobs (legacy), ML Transforms, Blueprints, Workflows, Triggers, Dev endpoints, Notebooks, Security). The main content area is titled 'Jobs' with a sub-instruction: 'A job is your business logic required to perform extract, transform and load (ETL) work. Job runs are initiated by triggers which can be scheduled or driven by events.' It features a search bar with filters for 'Name', 'Type' (Spark), 'ETL language' (python), 'Script location' (s3://vp-stock-tr...), 'Last modified' (31 March 2022 11:16 PM), and 'Job bookmark' (Disable). A table lists one job entry: 'vp-stock-trades-spark-etl-job'. Below the table are tabs for History, Details, Script, and Metrics. The History tab shows a single run record with columns: Run ID (jr_0aeda887a...), Retry attempt (1), Run status (Succeeded), Error (None), Output (Logs), Error logs (Logs), Glue version (3.0), Maximum capacity (2), Triggered by (None), Start time (1 Apr 2022 18:58:45), End time (1 Apr 2022 19:00:03), Start-up time (18 secs), Execution time (1 min), Timeout (2880 mins), and Job run input (s3://vp-stock-tr...).

Run ID	Retry attempt	Run status	Error	Output	Logs	Error logs	Glue version	Maximum capacity	Triggered by	Start time	End time	Start-up time	Execution time	Timeout	Delay	Job run input
jr_0aeda887a...	1	Succeeded	None	Logs	Logs	Logs	3.0	2	None	1 Apr 2022 18:58:45	1 Apr 2022 19:00:03	18 secs	1 min	2880 mins	s3://vp-stock-tr...	

20. EXECUTION OF AWS GLUE ETL SPARK JOB WITH TRANSFORMATION LOGIC (CONTINUED)

Job Execution History

The screenshot shows the AWS Glue Job Execution History interface. On the left, there's a sidebar with 'AWS Glue' navigation options: Data catalog, ETL (AWS Glue Studio, Jobs - New, Jobs (legacy), ML Transforms, Blueprints, Workflows, Triggers, Dev endpoints, Notebooks), and Security. The main area displays job details for a specific run:

Jobs > vp-stock-trades-spark-etl-job > jr_0aeda887ae9abb2c31ee3b43d5355787460409fa60d269ab8999dc4bd1930f5a

Job Run Id	jr_0aeda887ae9abb2c31ee3b43d5355787460409fa60d269ab8999dc4bd1930f5a
Job retry attempt	-
Name	vp-stock-trades-spark-etl-job
Trigger condition	
Input arguments	--enable-spark-ui true --spark-event-logs-path s3://vp-stock-trades-transformed-bucket/glue_spark_etl_event_logs
Job bookmark	Enable
Run status	Succeeded
Errors	Error logs
Logs	Logs
Continuous logging	Disable
Glue version	3.0
Start time	1 April 2022 12:26 AM UTC+1
End time	1 April 2022 12:27 AM UTC+1
Duration	1 min
Bookmark used	

20. EXECUTION OF AWS GLUE ETL SPARK JOB WITH TRANSFORMATION LOGIC (CONTINUED)

Job Run Details

AWS Glue

User preferences

Add job Action ▾ Filter by tags and attributes Showing: 1 - 1 < > ⌂ ⓘ

Name	Type	ETL language	Script location	Last modified	Job bookmark
vp-stock-trades-spark-etl-job	Spark	python	s3://vp-stock-tr... 31 March 2022 11:16 PM ... Disable		

History Details Script Metrics

Name: vp-stock-trades-spark-etl-job IAM role: VP_Glue_Crawler_Service_S3_Full_Access_Role Type: Spark Glue version: 3.0 Python version: 3 Spark version: 3.1 ETL language: python Script location: s3://vp-stock-trades-transformed-bucket/glue_spark_etl_python_scripts/vp-stock-trades-spark-etl-job-python-script Temporary directory: s3://vp-stock-trades-transformed-bucket/glue_spark_etl_python_scripts_temp_results Job bookmark: Disable Job metrics: Enable Continuous logging: Disable Server-side encryption: Disabled	Python lib path: - Jar lib path: - Other lib path: - Job parameters: --enable-spark-ui true --spark-event-logs-path s3://vp-stock-trades-transformed-bucket/glue_spark_etl_event_logs Non-overridable Job parameters: - Connections: vp-stock-trades-bucket-glue-connection Maximum capacity: 2 Worker type: G.1X Number of workers: 2 Job timeout (minutes): 2880 Delay notification: -
---	---

20. EXECUTION OF AWS GLUE ETL SPARK JOB WITH TRANSFORMATION LOGIC (CONTINUED)

Job Run Details

AWS Glue

Data catalog

Databases
Tables
Connections

Crawlers
Classifiers

Schema registries
Schemas

Settings

ETL

AWS Glue Studio

Jobs **- New**

Jobs (legacy)

ML Transforms

Blueprints

Workflows

Triggers

Dev endpoints

Notebooks

Security

Add job Action Filter by tags and attributes Showing: 1 - 1 User preferences

Name	Type	ETL language	Script location	Last modified	Job bookmark
vp-stock-trades-spark-etl-job	Spark	python	s3://vp-stock-tr... 31 March 2022 11:16 PM ... Disable		

Spark version: 3.1
ETL language: python
Script location: s3://vp-stock-trades-transformed-bucket/glue_spark_etl_python_scripts/vp-stock-trades-spark-etl-job-python-script
Temporary directory: s3://vp-stock-trades-transformed-bucket/glue_spark_etl_python_scripts_temp_results
Job bookmark: Disable
Job metrics: Enable
Continuous logging: Disable
Server-side encryption: Disabled

Job parameters: --spark-event-logs-path s3://vp-stock-trades-transformed-bucket/glue_spark_etl_event_logs

Non-overridable Job parameters:

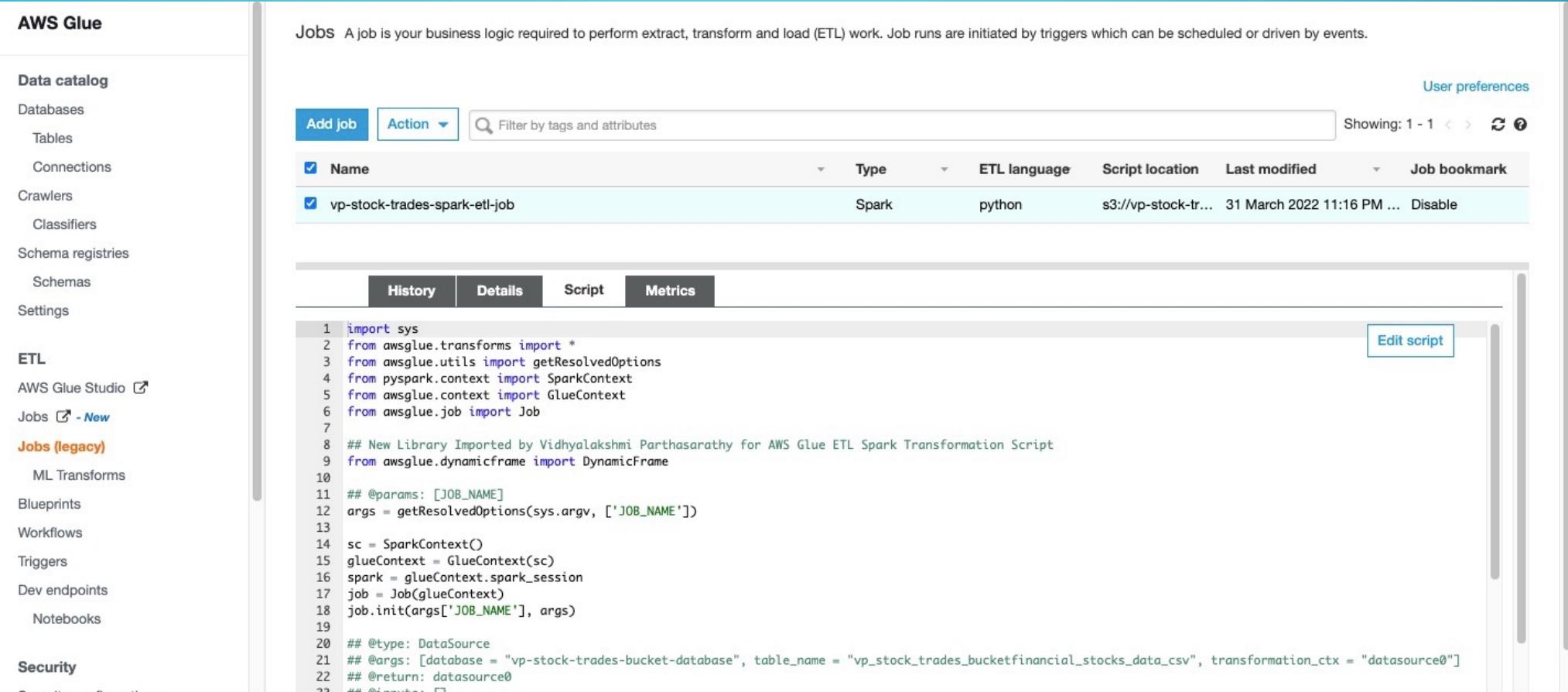
- Connections: vp-stock-trades-bucket-glue-connection
- Maximum capacity: 2
- Worker type: G.1X
- Number of workers: 2
- Job timeout (minutes): 2880
- Delay notification threshold (minutes): -
- Tags: -

Automatically run this job if any of the following triggers fire:

Trigger name	Trigger type	Trigger status	Trigger parameters	Jobs to trigger
No triggers start this job				

20. EXECUTION OF AWS GLUE ETL SPARK JOB WITH TRANSFORMATION LOGIC (CONTINUED)

Copy of the Spark ETL Job Python Script



The screenshot shows the AWS Glue Jobs console. On the left, the navigation menu includes Data catalog, ETL (AWS Glue Studio, Jobs - New, Jobs (legacy), ML Transforms, Blueprints, Workflows, Triggers, Dev endpoints, Notebooks, Security), and General configurations. The main area displays a table of jobs with one entry: "vp-stock-trades-spark-etl-job". The table columns are Name, Type, ETL language, Script location, Last modified, and Job bookmark. The "Script" tab is selected, showing the following Python code:

```
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 ## New Library Imported by Vidhyalakshmi Parthasarathy for AWS Glue ETL Spark Transformation Script
9 from awsglue.dynamicframe import DynamicFrame
10
11 ## @params: [JOB_NAME]
12 args = getResolvedOptions(sys.argv, ['JOB_NAME'])
13
14 sc = SparkContext()
15 glueContext = GlueContext(sc)
16 spark = glueContext.spark_session
17 job = Job(glueContext)
18 job.init(args['JOB_NAME'], args)
19
20 ## @type: DataSource
21 ## @args: [database = "vp-stock-trades-bucket-database", table_name = "vp_stock_trades_bucketfinancial_stocks_data_csv", transformation_ctx = "datasource0"]
22 ## @return: datasource0
23 ## @inputs: [
```

An "Edit script" button is located in the top right corner of the code editor.

20. EXECUTION OF AWS GLUE ETL SPARK JOB WITH TRANSFORMATION LOGIC

Job Run Metrics

AWS Glue

Data catalog

- Databases
- Tables
- Connections

Crawlers

- Classifiers

Schema registries

- Schemas

Settings

ETL

- AWS Glue Studio
- Jobs - New
- Jobs (legacy)**
- ML Transforms
- Blueprints
- Workflows
- Triggers
- Dev endpoints
- Notebooks

Security

User preferences

Add job Action ▾ Filter by tags and attributes Showing: 1 - 1 < >

Name	Type	ETL language	Script location	Last modified	Job bookmark
vp-stock-trades-spark-etl-job	Spark	python	s3://vp-stock-tr... 31 March 2022 11:16 PM ...	Disable	

History Details Script Metrics View additional metrics

1h 3h 12h 1d 3d 1w Add to dashboard

ETL Data Movement

No data available.
Try adjusting the dashboard time range.

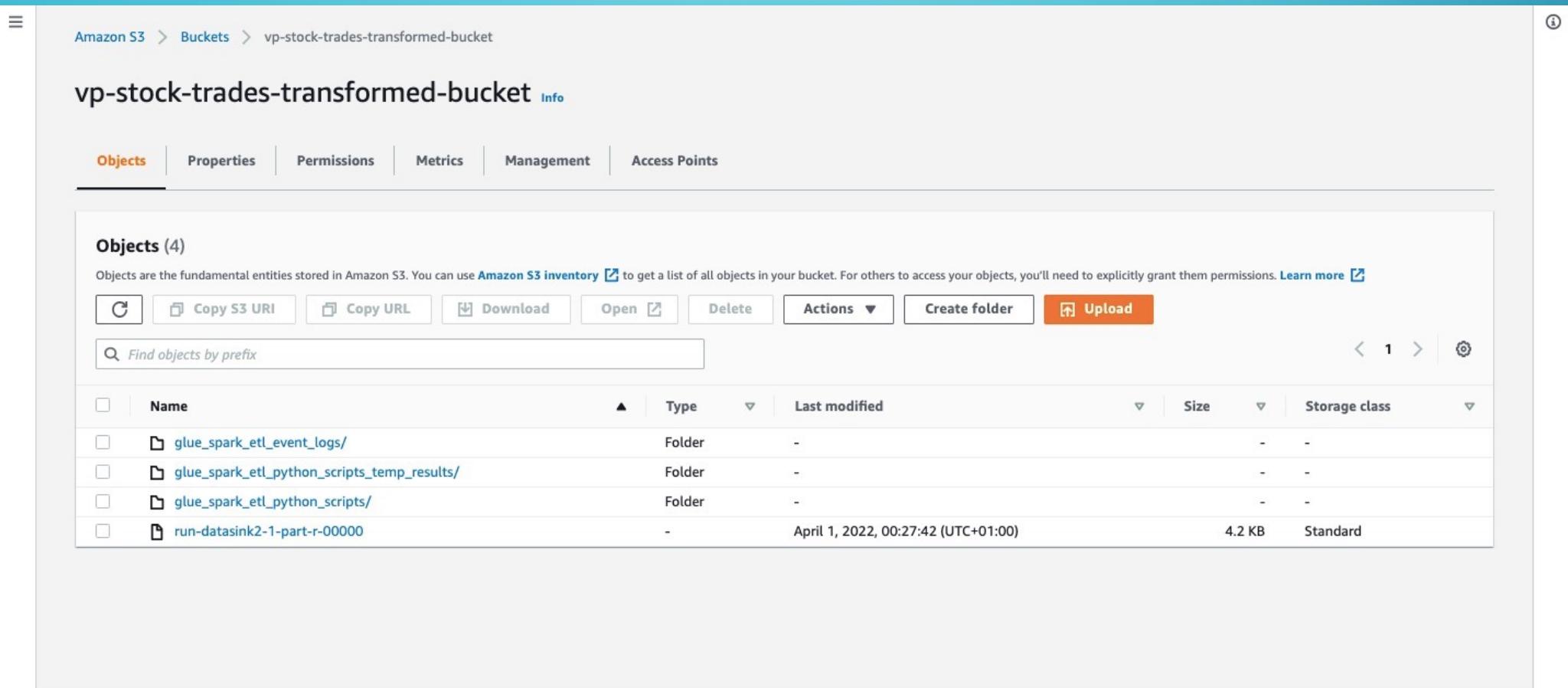
Memory Profile: Driver and Executors

50% Usage (0.5)

The screenshot shows the AWS Glue Job Run Metrics page. On the left, there's a sidebar with navigation links for Data catalog, Crawlers, Schema registries, Settings, ETL (AWS Glue Studio, Jobs, ML Transforms, Blueprints, Workflows, Triggers, Dev endpoints, Notebooks), and Security. The main area has tabs for History, Details, Script, and Metrics. The Metrics tab is active, displaying two charts: 'ETL Data Movement' (which shows 'No data available. Try adjusting the dashboard time range.') and 'Memory Profile: Driver and Executors' (which shows a blue bar at 50% usage from 23:30 to 00:00). At the top, there's a search bar, a filter for tags and attributes, and a table listing the job 'vp-stock-trades-spark-etl-job'. The job details show it's a Spark job using python, located at s3://vp-stock-tr..., last modified on 31 March 2022 at 11:16 PM, and is disabled. There are also buttons for User preferences, Add job, Action, and various time ranges (1h, 3h, 12h, 1d, 3d, 1w) and a 'Add to dashboard' button.

21. FILES SAVED IN THE AWS S3 BUCKET POST COMPLETION OF THE AWS GLUE SPARK ETL JOB (CONTINUED)

List of the folders created earlier



The screenshot shows the Amazon S3 console interface. The top navigation bar shows 'Amazon S3 > Buckets > vp-stock-trades-transformed-bucket'. The main heading is 'vp-stock-trades-transformed-bucket' with a 'Info' link. Below the heading are tabs: Objects (highlighted in orange), Properties, Permissions, Metrics, Management, and Access Points. A sub-header 'Objects (4)' is displayed. A message below it says: '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](#)'.

Below the message are several action buttons: 'Copy S3 URI', 'Copy URL', 'Download', 'Open', 'Delete', 'Actions', 'Create folder', and 'Upload'. There is also a search bar with the placeholder 'Find objects by prefix'.

A table lists the objects:

	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	glue_spark_etl_event_logs/	Folder	-	-	-
<input type="checkbox"/>	glue_spark_etl_python_scripts_temp_results/	Folder	-	-	-
<input type="checkbox"/>	glue_spark_etl_python_scripts/	Folder	-	-	-
<input type="checkbox"/>	run-datasink2-1-part-r-00000	-	April 1, 2022, 00:27:42 (UTC+01:00)	4.2 KB	Standard

21. FILES SAVED IN THE AWS S3 BUCKET POST COMPLETION OF THE AWS GLUE SPARK ETL JOB (CONTINUED)

Temporary Intermediate Results

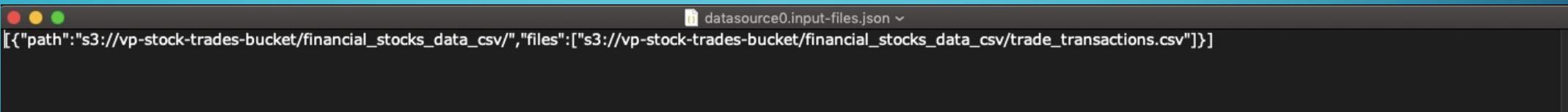
The screenshot shows the AWS S3 console interface. The path in the top navigation bar is: Amazon S3 > Buckets > vp-stock-trades-transformed-bucket > glue_spark_etl_python_scripts_temp_results/ > partitionlisting/ > vp-stock-trades-spark-etl-job/ > jr_Oaeda887ae9abb2c31ee3b43d5355787460409fa60d269ab8999dc4bd1930f5a/. A "Copy S3 URI" button is visible on the right.

The main area displays the "Objects" tab, which lists one object: "datasource0.input-files.json". The object details are as follows:

Name	Type	Last modified	Size	Storage class
datasource0.input-files.json	json	April 1, 2022, 00:27:07 (UTC+01:00)	156.0 B	Standard

21. FILES SAVED IN THE AWS S3 BUCKET POST COMPLETION OF THE AWS GLUE SPARK ETL JOB *(CONTINUED)*

Temporary Intermediate Results



```
datasource0.input-files.json ~
[{"path": "s3://vp-stock-trades-bucket/financial_stocks_data_csv/", "files": ["s3://vp-stock-trades-bucket/financial_stocks_data_csv/trade_transactions.csv"]}]
```

21. FILES SAVED IN THE AWS S3 BUCKET POST COMPLETION OF THE AWS GLUE SPARK ETL JOB (CONTINUED)

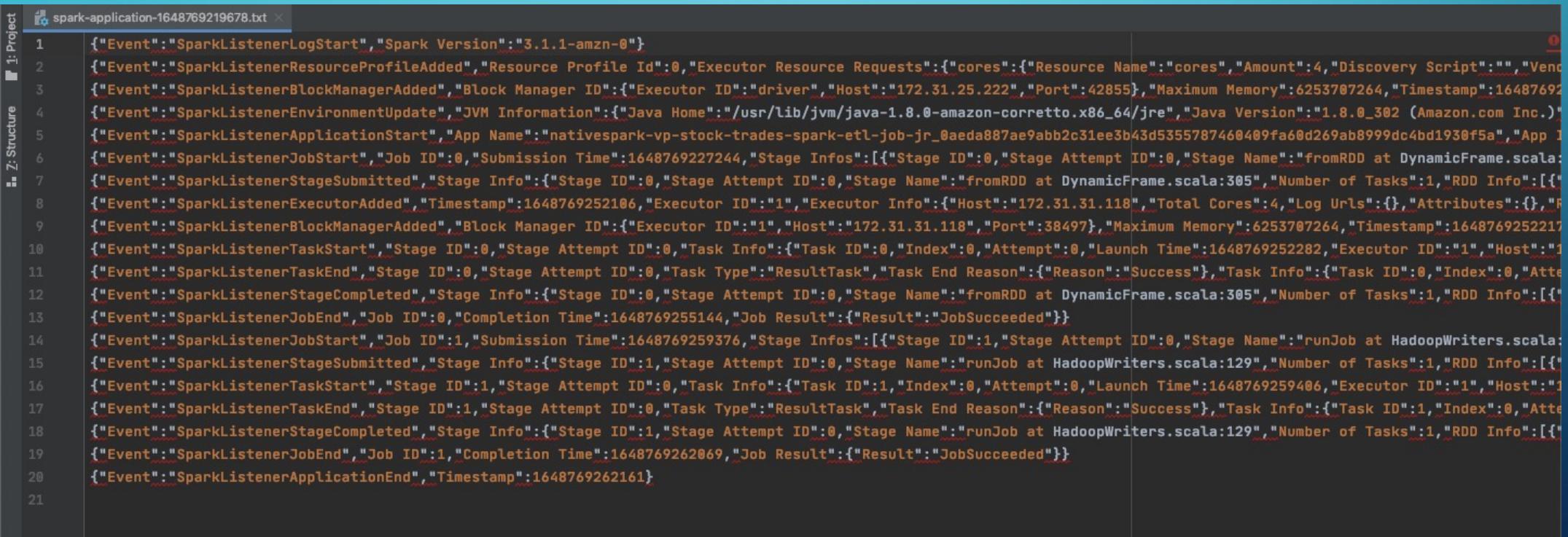
Spark Event Logs

The screenshot shows the Amazon S3 console interface. On the left, a sidebar menu includes 'Buckets', 'Access Points', 'Object Lambda Access Points', 'Multi-Region Access Points', 'Batch Operations', 'Access analyzer for S3', 'Block Public Access settings for this account', 'Storage Lens' (with 'Dashboards' and 'AWS Organizations settings' sub-options), 'Feature spotlight' (with a '3' badge), and 'AWS Marketplace for S3'. The main area displays the path 'Amazon S3 > Buckets > vp-stock-trades-transformed-bucket > glue_spark_etl_event_logs/'. The 'Objects' tab is selected, showing one object named 'spark-application-1648769219678'. The object details show it was last modified on April 1, 2022, at 00:27:43 (UTC+01:00), has a size of 95.4 KB, and is stored in the Standard storage class. A 'Copy S3 URI' button is visible above the object list.

Name	Type	Last modified	Size	Storage class
spark-application-1648769219678	-	April 1, 2022, 00:27:43 (UTC+01:00)	95.4 KB	Standard

21. FILES SAVED IN THE AWS S3 BUCKET POST COMPLETION OF THE AWS GLUE SPARK ETL JOB (CONTINUED)

Spark Event Logs



The screenshot shows a terminal window with the title "spark-application-1648769219678.txt". The log output is a JSON array of events. The events describe the lifecycle of a Spark application, starting with "SparkListenerLogStart" and ending with "SparkListenerApplicationEnd". Key details include the Spark version (3.1.1-amzn-0), resource profiles, block managers, and tasks. The log also shows successful job completion ("JobSucceeded") and application end.

```
[{"Event": "SparkListenerLogStart", "Spark Version": "3.1.1-amzn-0"}, {"Event": "SparkListenerResourceProfileAdded", "Resource Profile Id": 0, "Executor Resource Requests": {"cores": {"Resource Name": "cores", "Amount": 4, "Discovery Script": "", "Vendor": ""}}, "Timestamp": 1648769219678}, {"Event": "SparkListenerBlockManagerAdded", "Block Manager ID": {"Executor ID": "driver", "Host": "172.31.25.222", "Port": 42855}, "Maximum Memory": 6253707264, "Timestamp": 1648769219678}, {"Event": "SparkListenerEnvironmentUpdate", "JVM Information": {"Java Home": "/usr/lib/jvm/java-1.8.0-amazon-corretto.x86_64/jre", "Java Version": "1.8.0_302 (Amazon.com Inc.)"}}, {"Event": "SparkListenerApplicationStart", "App Name": "nativespark-vp-stock-trades-spark-etl-job-jr_0aeda887ae9abb2c31ee3b43d5355787460409fa60d269ab8999dc4bd1930f5a", "App ID": 1, "Timestamp": 1648769227244}, {"Event": "SparkListenerJobStart", "Job ID": 0, "Submission Time": 1648769227244, "Stage Infos": [{"Stage ID": 0, "Stage Attempt ID": 0, "Stage Name": "fromRDD at DynamicFrame.scala:305", "Number of Tasks": 1, "RDD Info": [{"Task ID": 0, "Index": 0, "Attempt": 0, "Launch Time": 1648769227244, "Executor ID": "1", "Host": "172.31.31.118", "Total Cores": 4, "Log URLs": {}, "Attributes": {}}, {"Task ID": 0, "Index": 0, "Attempt": 0, "Launch Time": 1648769227244, "Executor ID": "1", "Host": "172.31.31.118", "Total Cores": 4, "Log URLs": {}, "Attributes": {}}, {"Task ID": 0, "Index": 0, "Attempt": 0, "Launch Time": 1648769227244, "Executor ID": "1", "Host": "172.31.31.118", "Total Cores": 4, "Log URLs": {}, "Attributes": {}}, {"Task ID": 0, "Index": 0, "Attempt": 0, "Launch Time": 1648769227244, "Executor ID": "1", "Host": "172.31.31.118", "Total Cores": 4, "Log URLs": {}, "Attributes": {}}], "Timestamp": 1648769227244}, {"Event": "SparkListenerStageSubmitted", "Stage Info": {"Stage ID": 0, "Stage Attempt ID": 0, "Stage Name": "fromRDD at DynamicFrame.scala:305", "Number of Tasks": 1, "RDD Info": [{"Task ID": 0, "Index": 0, "Attempt": 0, "Launch Time": 1648769227244, "Executor ID": "1", "Host": "172.31.31.118", "Total Cores": 4, "Log URLs": {}, "Attributes": {}}, {"Task ID": 0, "Index": 0, "Attempt": 0, "Launch Time": 1648769227244, "Executor ID": "1", "Host": "172.31.31.118", "Total Cores": 4, "Log URLs": {}, "Attributes": {}}, {"Task ID": 0, "Index": 0, "Attempt": 0, "Launch Time": 1648769227244, "Executor ID": "1", "Host": "172.31.31.118", "Total Cores": 4, "Log URLs": {}, "Attributes": {}}, {"Task ID": 0, "Index": 0, "Attempt": 0, "Launch Time": 1648769227244, "Executor ID": "1", "Host": "172.31.31.118", "Total Cores": 4, "Log URLs": {}, "Attributes": {}}], "Timestamp": 1648769227244}, {"Event": "SparkListenerExecutorAdded", "Timestamp": 1648769252106, "Executor ID": "1", "Executor Info": {"Host": "172.31.31.118", "Total Cores": 4, "Log URLs": {}, "Attributes": {}}, "Timestamp": 1648769252106}, {"Event": "SparkListenerBlockManagerAdded", "Block Manager ID": {"Executor ID": "1", "Host": "172.31.31.118", "Port": 38497}, "Maximum Memory": 6253707264, "Timestamp": 1648769252217}, {"Event": "SparkListenerTaskStart", "Stage ID": 0, "Stage Attempt ID": 0, "Task Info": {"Task ID": 0, "Index": 0, "Attempt": 0, "Launch Time": 1648769252282, "Executor ID": "1", "Host": "172.31.31.118", "Total Cores": 4, "Log URLs": {}, "Attributes": {}}, "Timestamp": 1648769252282}, {"Event": "SparkListenerTaskEnd", "Stage ID": 0, "Stage Attempt ID": 0, "Task Type": "ResultTask", "Task End Reason": {"Reason": "Success"}, "Task Info": {"Task ID": 0, "Index": 0, "Attempt": 0, "Launch Time": 1648769252282, "Executor ID": "1", "Host": "172.31.31.118", "Total Cores": 4, "Log URLs": {}, "Attributes": {}}, "Timestamp": 1648769252282}, {"Event": "SparkListenerStageCompleted", "Stage Info": {"Stage ID": 0, "Stage Attempt ID": 0, "Stage Name": "fromRDD at DynamicFrame.scala:305", "Number of Tasks": 1, "RDD Info": [{"Task ID": 0, "Index": 0, "Attempt": 0, "Launch Time": 1648769227244, "Executor ID": "1", "Host": "172.31.31.118", "Total Cores": 4, "Log URLs": {}, "Attributes": {}}, {"Task ID": 0, "Index": 0, "Attempt": 0, "Launch Time": 1648769227244, "Executor ID": "1", "Host": "172.31.31.118", "Total Cores": 4, "Log URLs": {}, "Attributes": {}}, {"Task ID": 0, "Index": 0, "Attempt": 0, "Launch Time": 1648769227244, "Executor ID": "1", "Host": "172.31.31.118", "Total Cores": 4, "Log URLs": {}, "Attributes": {}}, {"Task ID": 0, "Index": 0, "Attempt": 0, "Launch Time": 1648769227244, "Executor ID": "1", "Host": "172.31.31.118", "Total Cores": 4, "Log URLs": {}, "Attributes": {}}], "Timestamp": 1648769255144}, {"Event": "SparkListenerJobEnd", "Job ID": 0, "Completion Time": 1648769255144, "Job Result": {"Result": "JobSucceeded"}, "Timestamp": 1648769255144}, {"Event": "SparkListenerJobStart", "Job ID": 1, "Submission Time": 1648769259376, "Stage Infos": [{"Stage ID": 1, "Stage Attempt ID": 0, "Stage Name": "runJob at HadoopWriters.scala:129", "Number of Tasks": 1, "RDD Info": [{"Task ID": 1, "Index": 0, "Attempt": 0, "Launch Time": 1648769259406, "Executor ID": "1", "Host": "172.31.31.118", "Total Cores": 4, "Log URLs": {}, "Attributes": {}}, {"Task ID": 1, "Index": 0, "Attempt": 0, "Launch Time": 1648769259406, "Executor ID": "1", "Host": "172.31.31.118", "Total Cores": 4, "Log URLs": {}, "Attributes": {}}, {"Task ID": 1, "Index": 0, "Attempt": 0, "Launch Time": 1648769259406, "Executor ID": "1", "Host": "172.31.31.118", "Total Cores": 4, "Log URLs": {}, "Attributes": {}}, {"Task ID": 1, "Index": 0, "Attempt": 0, "Launch Time": 1648769259406, "Executor ID": "1", "Host": "172.31.31.118", "Total Cores": 4, "Log URLs": {}, "Attributes": {}}], "Timestamp": 1648769259376}, {"Event": "SparkListenerStageSubmitted", "Stage Info": {"Stage ID": 1, "Stage Attempt ID": 0, "Stage Name": "runJob at HadoopWriters.scala:129", "Number of Tasks": 1, "RDD Info": [{"Task ID": 1, "Index": 0, "Attempt": 0, "Launch Time": 1648769259406, "Executor ID": "1", "Host": "172.31.31.118", "Total Cores": 4, "Log URLs": {}, "Attributes": {}}, {"Task ID": 1, "Index": 0, "Attempt": 0, "Launch Time": 1648769259406, "Executor ID": "1", "Host": "172.31.31.118", "Total Cores": 4, "Log URLs": {}, "Attributes": {}}, {"Task ID": 1, "Index": 0, "Attempt": 0, "Launch Time": 1648769259406, "Executor ID": "1", "Host": "172.31.31.118", "Total Cores": 4, "Log URLs": {}, "Attributes": {}}, {"Task ID": 1, "Index": 0, "Attempt": 0, "Launch Time": 1648769259406, "Executor ID": "1", "Host": "172.31.31.118", "Total Cores": 4, "Log URLs": {}, "Attributes": {}}], "Timestamp": 1648769259376}, {"Event": "SparkListenerTaskStart", "Stage ID": 1, "Stage Attempt ID": 0, "Task Info": {"Task ID": 1, "Index": 0, "Attempt": 0, "Launch Time": 1648769259406, "Executor ID": "1", "Host": "172.31.31.118", "Total Cores": 4, "Log URLs": {}, "Attributes": {}}, "Timestamp": 1648769259406}, {"Event": "SparkListenerTaskEnd", "Stage ID": 1, "Stage Attempt ID": 0, "Task Type": "ResultTask", "Task End Reason": {"Reason": "Success"}, "Task Info": {"Task ID": 1, "Index": 0, "Attempt": 0, "Launch Time": 1648769259406, "Executor ID": "1", "Host": "172.31.31.118", "Total Cores": 4, "Log URLs": {}, "Attributes": {}}, "Timestamp": 1648769259406}, {"Event": "SparkListenerStageCompleted", "Stage Info": {"Stage ID": 1, "Stage Attempt ID": 0, "Stage Name": "runJob at HadoopWriters.scala:129", "Number of Tasks": 1, "RDD Info": [{"Task ID": 1, "Index": 0, "Attempt": 0, "Launch Time": 1648769259406, "Executor ID": "1", "Host": "172.31.31.118", "Total Cores": 4, "Log URLs": {}, "Attributes": {}}, {"Task ID": 1, "Index": 0, "Attempt": 0, "Launch Time": 1648769259406, "Executor ID": "1", "Host": "172.31.31.118", "Total Cores": 4, "Log URLs": {}, "Attributes": {}}, {"Task ID": 1, "Index": 0, "Attempt": 0, "Launch Time": 1648769259406, "Executor ID": "1", "Host": "172.31.31.118", "Total Cores": 4, "Log URLs": {}, "Attributes": {}}, {"Task ID": 1, "Index": 0, "Attempt": 0, "Launch Time": 1648769259406, "Executor ID": "1", "Host": "172.31.31.118", "Total Cores": 4, "Log URLs": {}, "Attributes": {}}], "Timestamp": 1648769262069}, {"Event": "SparkListenerJobEnd", "Job ID": 1, "Completion Time": 1648769262069, "Job Result": {"Result": "JobSucceeded"}, "Timestamp": 1648769262069}, {"Event": "SparkListenerApplicationEnd", "Timestamp": 1648769262161}
```

21. FILES SAVED IN THE AWS S3 BUCKET POST COMPLETION OF THE AWS GLUE SPARK ETL JOB (CONTINUED)

Transformed Data File

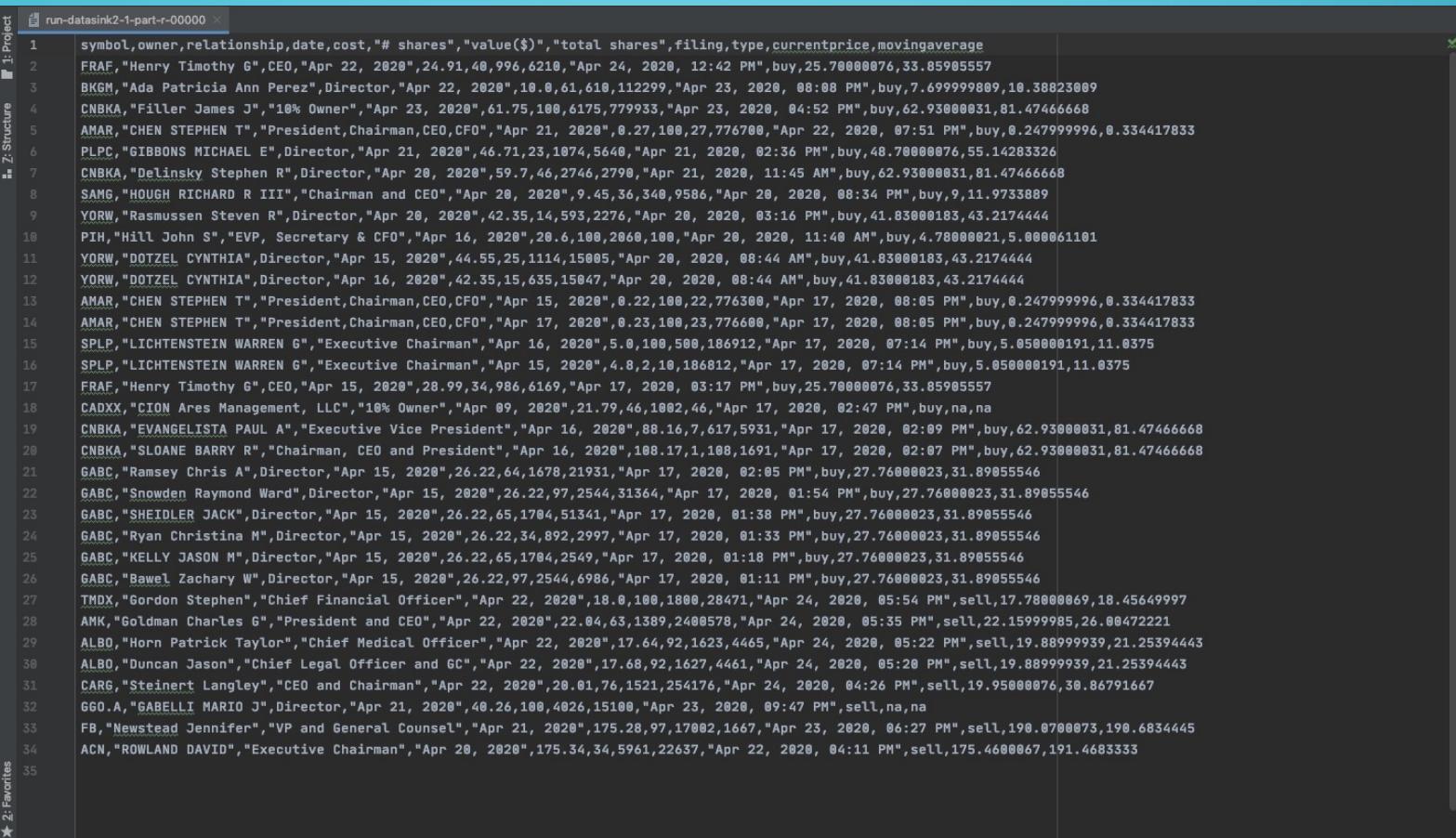
The screenshot shows the Amazon S3 Object Overview page for a file named "run-datasink2-1-part-r-00000". The left sidebar contains navigation links for Buckets, Storage Lens, Feature spotlight, and AWS Marketplace for S3. The main content area displays the object's properties, including its owner, AWS Region, Last modified date, Size, Type, Key, and various URLs.

Object overview

Property	Value
Owner	4d97ff8f9dfc260e11d18083dc29d77931845b8a284d61ae03f6323dc4a5b186
AWS Region	EU (London) eu-west-2
Last modified	April 1, 2022, 00:27:42 (UTC+01:00)
Size	4.2 KB
Type	
Key	run-datasink2-1-part-r-00000
S3 URI	s3://vp-stock-trades-transformed-bucket/run-datasink2-1-part-r-00000
Amazon Resource Name (ARN)	arn:aws:s3:::vp-stock-trades-transformed-bucket/run-datasink2-1-part-r-00000
Entity tag (Etag)	72405c8ba7d451688293321feb33f2a4
Object URL	https://vp-stock-trades-transformed-bucket.s3.eu-west-2.amazonaws.com/run-datasink2-1-part-r-00000

21. FILES SAVED IN THE AWS S3 BUCKET POST COMPLETION OF THE AWS GLUE SPARK ETL JOB

- ❖ Transformed Data File
- ❖ Transformation Logic to filter the trade transactions with the total number of shares ('# Shares') as less than or equal to 100. Expected Results achieved as 33 Trade Transactions (Rows) in the output transformed data file.



```
run-datasink2-1-part-r-00000
1 symbol,owner,relationship,date,cost,"# shares","value($)", "total shares",filing,type,currentprice,movingaverage
2 FRAF,"Henry Timothy G",CEO,"Apr 22, 2020",24.91,46,996,6210,"Apr 24, 2020, 12:42 PM",buy,25.70000076,33.85905557
3 BKGM,"Ada Patricia Ann Perez",Director,"Apr 22, 2020",10.0,61,610,112299,"Apr 23, 2020, 08:08 PM",buy,7.69999809,10.38823009
4 CNBKA,"Filler James J","10% Owner","Apr 23, 2020",61.75,100,6175,779933,"Apr 23, 2020, 04:52 PM",buy,62.93000031,81.47466668
5 AMAR,"CHEN STEPHEN T","President,Chairman,CEO,CFO","Apr 21, 2020",0.27,100,27,776700,"Apr 22, 2020, 07:51 PM",buy,0.247999996,0.334417833
6 PLPC,"GIBBONS MICHAEL E",Director,"Apr 21, 2020",46.71,23,1874,5640,"Apr 21, 2020, 02:36 PM",buy,48.70000076,55.14283326
7 CNBKA,"Delinsky Stephen R",Director,"Apr 20, 2020",59.7,46,2746,2790,"Apr 21, 2020, 11:45 AM",buy,62.93000031,81.47466668
8 SAMG,"HOUGH RICHARD R III","Chairman and CEO","Apr 20, 2020",9.45,36,340,9586,"Apr 20, 2020, 08:34 PM",buy,9,11.9733889
9 YORW,"Rasmussen Steven R",Director,"Apr 20, 2020",42.35,14,593,2276,"Apr 20, 2020, 03:16 PM",buy,41.83000183,43.2174444
10 PTH,"Hill John S","EVP, Secretary & CFO","Apr 16, 2020",20.6,100,2060,100,"Apr 20, 2020, 11:40 AM",buy,4.78000021,5.000061101
11 YORW,"DOTZEL CYNTHIA",Director,"Apr 15, 2020",44.55,25,1114,15005,"Apr 20, 2020, 08:44 AM",buy,41.83000183,43.2174444
12 YORW,"DOTZEL CYNTHIA",Director,"Apr 16, 2020",42.35,15,635,15047,"Apr 20, 2020, 08:44 AM",buy,41.83000183,43.2174444
13 AMAR,"CHEN STEPHEN T","President,Chairman,CEO,CFO","Apr 15, 2020",0.22,100,22,776300,"Apr 17, 2020, 08:05 PM",buy,0.247999996,0.334417833
14 AMAR,"CHEN STEPHEN T","President,Chairman,CEO,CFO","Apr 17, 2020",0.23,100,23,776600,"Apr 17, 2020, 08:05 PM",buy,0.247999996,0.334417833
15 SPLP,"LICHTENSTEIN WARREN G","Executive Chairman","Apr 16, 2020",5.0,100,500,186912,"Apr 17, 2020, 07:14 PM",buy,5.050000191,11.0375
16 SPLP,"LICHTENSTEIN WARREN G","Executive Chairman","Apr 15, 2020",4.8,2,10,186812,"Apr 17, 2020, 07:14 PM",buy,5.050000191,11.0375
17 FRAF,"Henry Timothy G",CEO,"Apr 15, 2020",28.99,34,986,6169,"Apr 17, 2020, 03:17 PM",buy,25.70000076,33.85905557
18 CADXX,"CION Ares Management, LLC","10% Owner","Apr 09, 2020",21.79,46,1002,46,"Apr 17, 2020, 02:47 PM",buy,na,na
19 CNBKA,"EVANGELISTA PAUL A","Executive Vice President","Apr 16, 2020",88.16,7,617,5931,"Apr 17, 2020, 02:09 PM",buy,62.93000031,81.47466668
20 CNBKA,"SLOANE BARRY R","Chairman, CEO and President","Apr 16, 2020",108.17,1,108,1691,"Apr 17, 2020, 02:07 PM",buy,62.93000031,81.47466668
21 GABC,"Ramsey Chris A",Director,"Apr 15, 2020",26.22,64,1678,21931,"Apr 17, 2020, 02:05 PM",buy,27.76000023,31.89055546
22 GABC,"Snowden Raymond Ward",Director,"Apr 15, 2020",26.22,97,2544,31364,"Apr 17, 2020, 01:54 PM",buy,27.76000023,31.89055546
23 GABC,"SHEIDLDER JACK",Director,"Apr 15, 2020",26.22,65,1704,51341,"Apr 17, 2020, 01:38 PM",buy,27.76000023,31.89055546
24 GABC,"Ryan Christine M",Director,"Apr 15, 2020",26.22,34,892,2997,"Apr 17, 2020, 01:33 PM",buy,27.76000023,31.89055546
25 GABC,"KELLY JASON M",Director,"Apr 15, 2020",26.22,65,1704,2549,"Apr 17, 2020, 01:18 PM",buy,27.76000023,31.89055546
26 GABC,"Bawel Zachary W",Director,"Apr 15, 2020",26.22,97,2544,6986,"Apr 17, 2020, 01:11 PM",buy,27.76000023,31.89055546
27 TMDX,"Gordon Stephen","Chief Financial Officer","Apr 22, 2020",18.0,100,1800,28471,"Apr 24, 2020, 05:54 PM",sell,17.78000069,18.45649997
28 AMK,"Goldman Charles G","President and CEO","Apr 22, 2020",22.04,63,1389,2400578,"Apr 24, 2020, 05:35 PM",sell,22.15999985,26.00472221
29 ALBO,"Horn Patrick Taylor","Chief Medical Officer","Apr 22, 2020",17.64,92,1623,4465,"Apr 24, 2020, 05:22 PM",sell,19.88999939,21.25394443
30 ALBO,"Duncan Jason","Chief Legal Officer and GC","Apr 22, 2020",17.68,92,1627,4461,"Apr 24, 2020, 05:20 PM",sell,19.88999939,21.25394443
31 CARG,"Steinert Langley","CEO and Chairman","Apr 22, 2020",20.01,76,1521,254176,"Apr 24, 2020, 04:26 PM",sell,19.95000076,30.86791667
32 GGO,A,"GABELLI MARIO J",Director,"Apr 21, 2020",40.26,100,4026,15100,"Apr 23, 2020, 09:47 PM",sell,na,na
33 FB,"Newstead Jennifer","VP and General Counsel","Apr 21, 2020",175.28,97,17002,1667,"Apr 23, 2020, 06:27 PM",sell,190.0700073,190.6834445
34 ACN,"ROWLAND DAVID","Executive Chairman","Apr 20, 2020",175.34,34,5961,22637,"Apr 22, 2020, 04:11 PM",sell,175.4600067,191.4683333
```

22. AWS CLOUDWATCH LOGS OF THE AWS GLUE SPARK ETL SPARK TRANSFORMATION JOB (CONTINUED)

The screenshot shows the AWS CloudWatch Log Events interface. The left sidebar navigation includes 'CloudWatch' (selected), 'Favorites', 'Dashboards', 'Alarms' (with 0 alarms), 'Logs' (selected), 'Log groups' (selected), 'Logs Insights', 'Metrics', 'X-Ray traces', 'Events', 'Application monitoring', 'Insights', 'Settings', and 'Getting Started'. The main content area shows the log group path: CloudWatch > Log groups > /aws-glue/jobs/error > jr_Oaeda887ae9abb2c31ee3b43d5355787460409fa60d269ab8999dc4bd1930f5a. The title bar indicates the specific log entry ID: jr_Oaeda887ae9abb2c31ee3b43d5355787460409fa60d269ab8999dc4bd1930f5a. The 'Log events' section contains a message: 'There are older events to load. Load more.' Below this, a table lists 20 log events from April 1, 2022, at 04:00:00 to April 31, 2022, at 23:00:00. The columns are 'Timestamp' and 'Message'. The messages describe ANTLR tool version discrepancies and various GlueContext operations like getting catalog sources and secret options from Secret Manager.

Timestamp	Message
2022-04-01T00:27:04.608+01:00	ANTLR Tool version 4.5.1 used for code generation does not match the current runtime version 4.8ANTLR Runtime version 4.7...
2022-04-01T00:27:04.644+01:00	ANTLR Tool version 4.5.1 used for code generation does not match the current runtime version 4.8ANTLR Runtime version 4.7...
2022-04-01T00:27:04.670+01:00	2022-03-31 23:27:04,670 INFO [Thread-5] glue.GlueContext (GlueContext.scala:getCatalogSource(214)): getCatalogSource: cat...
2022-04-01T00:27:04.689+01:00	2022-03-31 23:27:04,688 INFO [Thread-5] glue.GlueContext (GlueContext.scala:getCatalogSource(228)): getCatalogSource: tra...
2022-04-01T00:27:04.689+01:00	2022-03-31 23:27:04,689 INFO [Thread-5] glue.GlueContext (GlueContext.scala:getCatalogSource(323)): classification csv
2022-04-01T00:27:05.165+01:00	2022-03-31 23:27:05,165 INFO [Thread-5] glue.GlueContext (Logging.scala:logInfo(57)): No of partitions from catalog are 0...
2022-04-01T00:27:05.167+01:00	2022-03-31 23:27:05,166 INFO [Thread-5] glue.GlueContext (GlueContext.scala:getCatalogSource(383)): location s3://vp-stoc...
2022-04-01T00:27:05.174+01:00	2022-03-31 23:27:05,174 INFO [Thread-5] glue.GlueContext (GlueContext.scala:getSecretOptionsFromSecretManager(818)): Glue...
2022-04-01T00:27:05.754+01:00	2022-03-31 23:27:05,753 INFO [Thread-5] glue.HadoopDataSource (DataSource.scala:setFormat(595)): glue.etl.simcsvParser, ...
2022-04-01T00:27:05.840+01:00	2022-03-31 23:27:05,840 INFO [Thread-5] hadoop.PartitionFilesListerUsingBookmark (FileSystemBookmark.scala:partitions(326...
2022-04-01T00:27:05.840+01:00	2022-03-31 23:27:05,840 INFO [Thread-5] hadoop.PartitionFilesListerUsingBookmark (FileSystemBookmark.scala:partitions(327...
2022-04-01T00:27:05.840+01:00	2022-03-31 23:27:05,840 INFO [Thread-5] hadoop.PartitionFilesListerUsingBookmark (FileSystemBookmark.scala:partitions(328...
2022-04-01T00:27:05.842+01:00	2022-03-31 23:27:05,841 INFO [Thread-5] hadoop.PartitionFilesListerUsingBookmark (FileSystemBookmark.scala:partitions(333...
2022-04-01T00:27:06.409+01:00	2022-03-31 23:27:06,409 INFO [Thread-5] hadoop.PartitionFilesListerUsingBookmark (FileSystemBookmark.scala:\$anonfun\$parti...
2022-04-01T00:27:06.490+01:00	2022-03-31 23:27:06,489 INFO [Thread-5] hadoop.PartitionFilesListerUsingBookmark (FileSystemBookmark.scala:applyBookmarkF...

22. AWS CLOUDWATCH LOGS OF THE AWS GLUE SPARK ETL SPARK TRANSFORMATION JOB (CONTINUED)

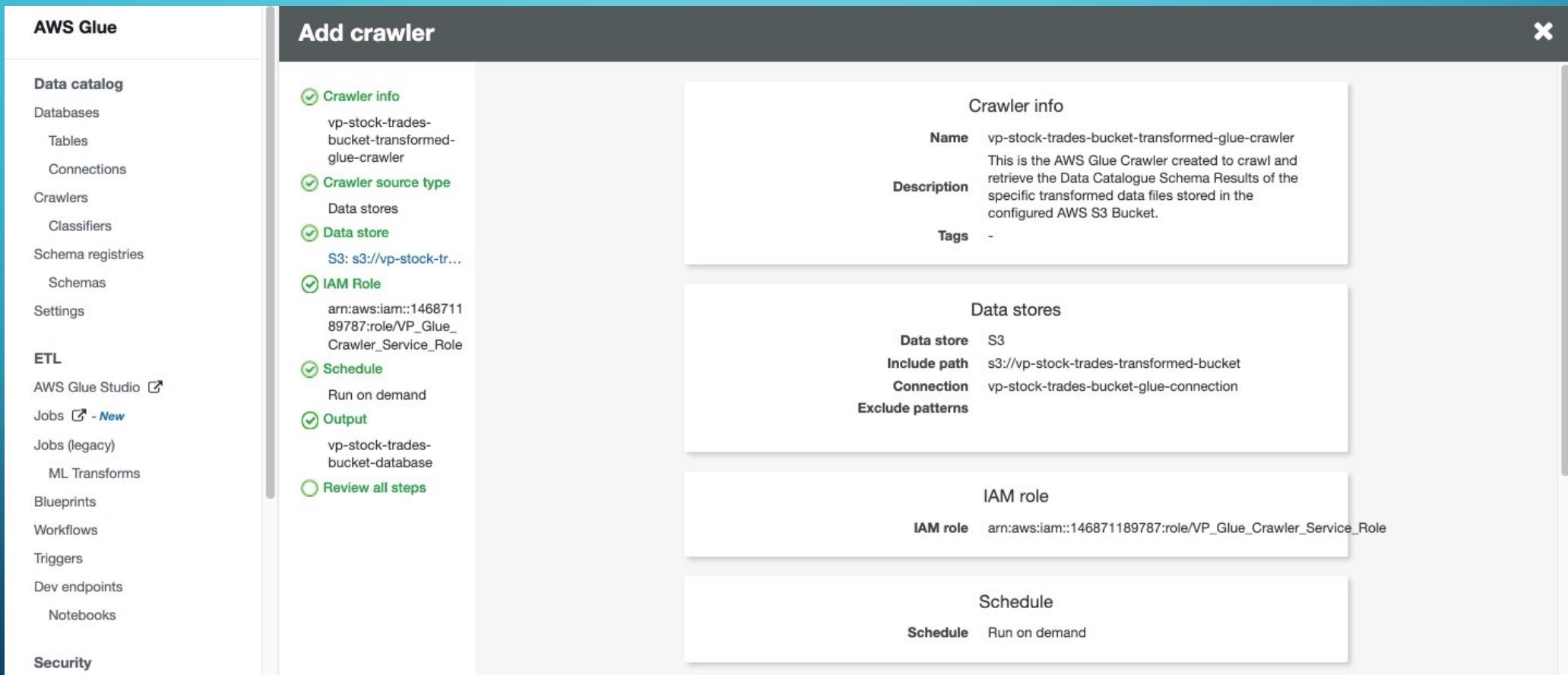
CloudWatch	X
Favorites	▶
Dashboards	
Alarms	⚠️ 0 ⓘ 0 🔍 0
In alarm	
All alarms	
Logs	
Log groups	
Logs Insights	
Metrics	
X-Ray traces	
Events	
Application monitoring	
Insights	
Settings	
Getting Started	
	▶ 2022-04-01T00:27:42.066+01:00 2022-03-31 23:27:42,065 INFO [task-result-getter-1] scheduler.TaskSetManager (Logging.scala:logInfo(57)): Finished task 0... ▶ 2022-04-01T00:27:42.066+01:00 2022-03-31 23:27:42,065 INFO [task-result-getter-1] scheduler.TaskSchedulerImpl (Logging.scala:logInfo(57)): Removed Task... ▶ 2022-04-01T00:27:42.068+01:00 2022-03-31 23:27:42,068 INFO [dag-scheduler-event-loop] scheduler.DAGScheduler (Logging.scala:logInfo(57)): ResultStage 1... ▶ 2022-04-01T00:27:42.069+01:00 2022-03-31 23:27:42,069 INFO [dag-scheduler-event-loop] scheduler.DAGScheduler (Logging.scala:logInfo(57)): Job 1 is fini... ▶ 2022-04-01T00:27:42.069+01:00 2022-03-31 23:27:42,069 INFO [dag-scheduler-event-loop] scheduler.TaskSchedulerImpl (Logging.scala:logInfo(57)): Killing ... ▶ 2022-04-01T00:27:42.071+01:00 2022-03-31 23:27:42,070 INFO [Thread-5] scheduler.DAGScheduler (Logging.scala:logInfo(57)): Job 1 finished: runJob at Had... ▶ 2022-04-01T00:27:42.072+01:00 2022-03-31 23:27:42,072 INFO [Thread-5] sinks.HadoopDataSink (HadoopDataSink.scala:\$anonfun\$writeDynamicFrame\$1(273)): en... ▶ 2022-04-01T00:27:42.072+01:00 2022-03-31 23:27:42,072 INFO [Thread-5] sinks.HadoopDataSink (HadoopDataSink.scala:\$anonfun\$writeDynamicFrame\$1(274)): pa... ▶ 2022-04-01T00:27:42.073+01:00 2022-03-31 23:27:42,073 INFO [Thread-5] sinks.HadoopDataSink (HadoopDataSink.scala:\$anonfun\$writeDynamicFrame\$1(275)): na... ▶ 2022-04-01T00:27:42.157+01:00 2022-03-31 23:27:42,156 INFO [main] glue.ProcessLauncher (Logging.scala:logInfo(57)): postprocessing ▶ 2022-04-01T00:27:42.158+01:00 2022-03-31 23:27:42,157 INFO [main] glue.LogPusher (Logging.scala:logInfo(57)): stopping ▶ 2022-04-01T00:27:42.161+01:00 2022-03-31 23:27:42,161 INFO [shutdown-hook-0] spark.SparkContext (Logging.scala:logInfo(57)): Invoking stop() from shutd... ▶ 2022-04-01T00:27:42.166+01:00 2022-03-31 23:27:42,166 INFO [shutdown-hook-0] scheduler.JESSchedulerBackend (Logging.scala:logInfo(57)): Shutting down a... ▶ 2022-04-01T00:27:42.167+01:00 2022-03-31 23:27:42,166 INFO [dispatcher-CoarseGrainedScheduler] cluster.CoarseGrainedSchedulerBackend\$DriverEndpoint (Lo... ▶ 2022-04-01T00:27:42.181+01:00 2022-03-31 23:27:42,181 INFO [dispatcher-event-loop-1] spark.MapOutputTrackerMasterEndpoint (Logging.scala:logInfo(57)): ... ▶ 2022-04-01T00:27:42.219+01:00 2022-03-31 23:27:42,219 INFO [shutdown-hook-0] memory.MemoryStore (Logging.scala:logInfo(57)): MemoryStore cleared ▶ 2022-04-01T00:27:42.220+01:00 2022-03-31 23:27:42,219 INFO [shutdown-hook-0] storage.BlockManager (Logging.scala:logInfo(57)): BlockManager stopped ▶ 2022-04-01T00:27:42.224+01:00 2022-03-31 23:27:42,224 INFO [shutdown-hook-0] storage.BlockManagerMaster (Logging.scala:logInfo(57)): BlockManagerMaster... ▶ 2022-04-01T00:27:42.229+01:00 2022-03-31 23:27:42,229 INFO [dispatcher-event-loop-2] scheduler.OutputCommitCoordinator\$OutputCommitCoordinatorEndpoint ... ▶ 2022-04-01T00:27:42.236+01:00 2022-03-31 23:27:42,236 INFO [shutdown-hook-0] spark.SparkContext (Logging.scala:logInfo(57)): Successfully stopped Spark... ▶ 2022-04-01T00:27:42.237+01:00 2022-03-31 23:27:42,237 INFO [shutdown-hook-0] glue.LogPusher (Logging.scala:logInfo(57)): uploading /tmp/spark-event-log... ▶ 2022-04-01T00:27:42.326+01:00 2022-03-31 23:27:42,325 INFO [shutdown-hook-0] s3n.MultipartUploadOutputStream (MultipartUploadOutputStream.java:close(41... ▶ 2022-04-01T00:27:42.360+01:00 2022-03-31 23:27:42,359 INFO [shutdown-hook-0] util.ShutdownHookManager (Logging.scala:logInfo(57)): Shutdown hook called ▶ 2022-04-01T00:27:42.360+01:00 2022-03-31 23:27:42,360 INFO [shutdown-hook-0] util.ShutdownHookManager (Logging.scala:logInfo(57)): Deleting directory /... ▶ 2022-04-01T00:27:42.364+01:00 2022-03-31 23:27:42,364 INFO [shutdown-hook-0] util.ShutdownHookManager (Logging.scala:logInfo(57)): Deleting directory /...
	No newer events at this moment. Auto retry paused. Resume



CATALOGUE AND QUERY
THE TRANSFORMED DATA

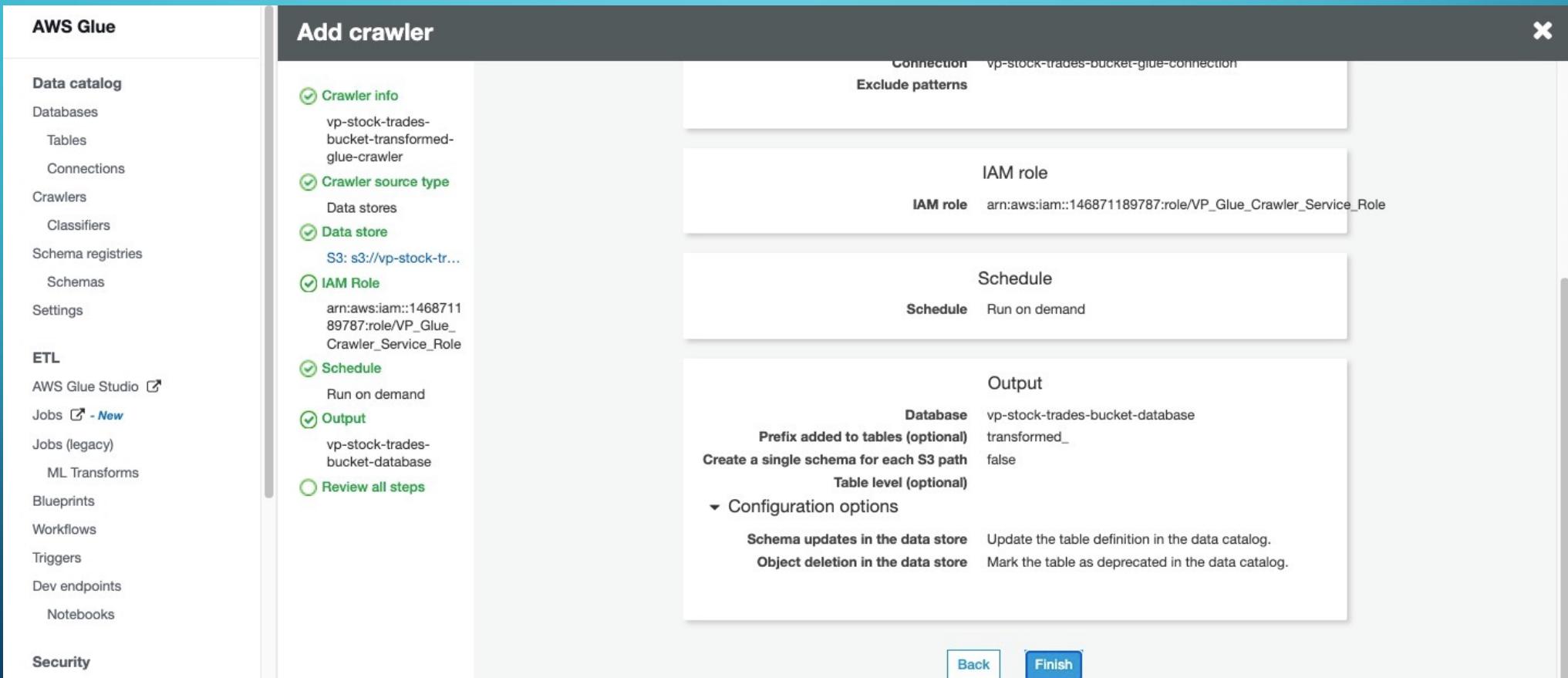
23. CREATING AWS GLUE CRAWLER TO CRAWL THE TRANSFORMED DATA STORED IN THE AWS S3 BUCKET (CONTINUED)

The transformed data file 'run-datasink2-1-part-r-00000' stored in the AWS S3 Bucket 'vp-stocks-trades-transformed-bucket' will be crawled to extract the data catalog information



23. CREATING AWS GLUE CRAWLER TO CRAWL THE TRANSFORMED DATA STORED IN THE AWS S3 BUCKET (CONTINUED)

The transformed data file 'run-datasink2-1-part-r-00000' stored in the AWS S3 Bucket 'vp-stocks-trade-transformed-bucket' will be crawled to extract the data catalog information



23. CREATING AWS GLUE CRAWLER TO CRAWL THE TRANSFORMED DATA STORED IN THE AWS S3 BUCKET (CONTINUED)

The transformed data file 'run-datasink2-1-part-r-00000' stored in the AWS S3 Bucket 'vp-stocks-trades-transformed-bucket' will be crawled to extract the data catalog information

The screenshot shows the AWS Glue Data Catalog interface. On the left, the navigation menu includes 'Data catalog' (Databases, Tables, Connections), 'Crawlers' (Classifiers, Schema registries, Schemas, Settings), and 'ETL' (AWS Glue Studio, Jobs, Jobs (legacy), ML Transforms, Blueprints, Workflows, Triggers, Dev endpoints, Notebooks, Security). The 'Crawlers' section is selected.

The main content area displays a message: "Crawler vp-stock-trades-bucket-transformed-glue-crawler was created to run on demand. Run it now?" with a green 'Run it now?' button and a red 'X' icon.

A table lists the crawlers:

<input type="checkbox"/>	Name	Schedule	Status	Logs	Last runtime	Median runtime	Tables updated	Tables added
<input type="checkbox"/>	vp-stock-trades-bucket-glue-crawler		Ready	Logs	3 mins	3 mins	0	2
<input checked="" type="checkbox"/>	vp-stock-trades-bucket-transformed-glue-cra...		Ready		0 secs	0 secs	0	0

Filtering options include 'Action' (Add crawler, Run crawler, Action dropdown), 'User preferences', and a search bar 'Filter by tags and attributes'. A note at the bottom says 'Showing: 1 - 2 < > ⌂ ⓘ'.

23. CREATING AWS GLUE CRAWLER TO CRAWL THE TRANSFORMED DATA STORED IN THE AWS S3 BUCKET

The transformed data file 'run-datasink2-1-part-r-00000' stored in the AWS S3 Bucket 'vp-stocks-trades-transformed-bucket' will be crawled to extract the data catalog information

The screenshot shows the AWS Glue console interface. On the left, there's a sidebar with 'Data catalog' and 'ETL' sections. Under 'Data catalog', 'Crawlers' is selected. Under 'ETL', 'AWS Glue Studio' is selected. The main area displays the configuration for a crawler named 'vp-stock-trades-bucket-transformed-glue-crawler'. The crawler is set to run at 'Table level' and has a 'Create a single schema for each S3 path' configuration. It uses the 'Security configuration' and 'Selected classifiers' options. The crawler is currently in a 'Ready' state. The 'Configuration options' section includes settings for 'Schema updates in the data store' and 'Object deletion in the data store'.

Crawlers > vp-stock-trades-bucket-transformed-glue-crawler

Run crawler **Edit**

Data catalog

- Databases
- Tables
- Connections
- Crawlers
- Classifiers
- Schema registries
- Schemas
- Settings

ETL

- AWS Glue Studio
- Jobs - New
- Jobs (legacy)
- ML Transforms
- Blueprints
- Workflows
- Triggers
- Dev endpoints
- Notebooks

Security

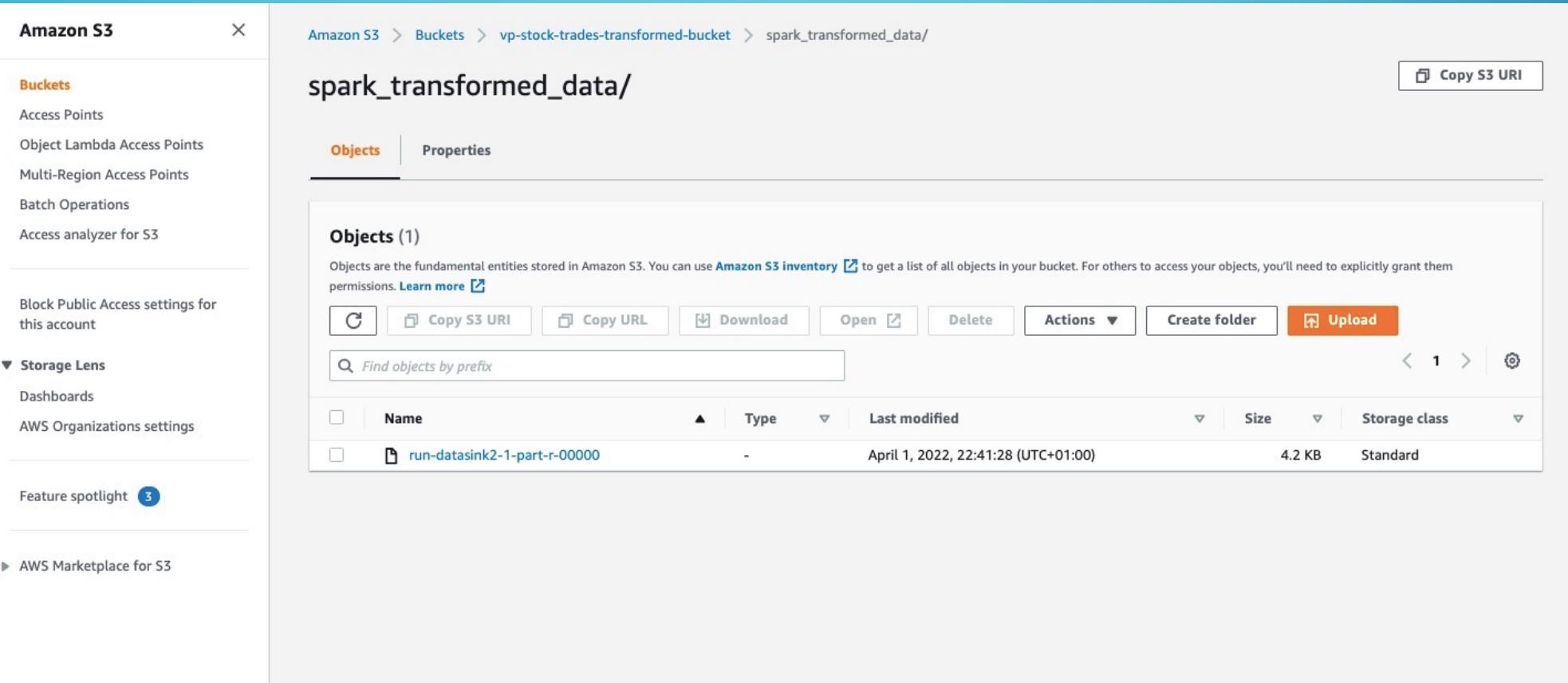
Name vp-stock-trades-bucket-transformed-glue-crawler
Description This is the AWS Glue Crawler created to crawl and retrieve the Data Catalogue Schema Results of the specific transformed data files stored in the configured AWS S3 Bucket.
Create a single schema for each S3 path
Table level
Security configuration
Tags -
State Ready
Schedule
Last updated Fri Apr 01 21:26:16 GMT+100 2022
Date created Fri Apr 01 21:26:16 GMT+100 2022
Database vp-stock-trades-bucket-database
Table prefix transformed_
Service role VP_Glue_Crawler_Service_Role
Selected classifiers
Data store S3
Include path s3://vp-stock-trades-transformed-bucket
Connection vp-stock-trades-bucket-glue-connection
Exclude patterns

Configuration options

- Schema updates in the data store** Update the table definition in the data catalog.
- Object deletion in the data store** Mark the table as deprecated in the data catalog.

24. COPIED THE TRANSFORMED DATA FILE INTO A FOLDER WITHIN THE AWS S3 TRANSFORMED BUCKET BEFORE EXECUTING THE AWS GLUE CRAWLER

This is followed as a best practice when we have multiple sub-folders within the given AWS S3 Bucket to be crawled and avoid having querying issues with AWS Athena for those files that are placed directly within the AWS S3 Bucket without any sub-folder structure



The screenshot shows the Amazon S3 console interface. On the left, the navigation pane includes options like Buckets, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, Access analyzer for S3, Block Public Access settings for this account, Storage Lens (with Dashboards and AWS Organizations settings), Feature spotlight, and AWS Marketplace for S3. The main content area displays the path: Amazon S3 > Buckets > vp-stock-trades-transformed-bucket > spark_transformed_data/. The current view is on the 'Objects' tab, which shows one object named 'run-datasink2-1-part-r-00000'. The object was last modified on April 1, 2022, at 22:41:28 (UTC+01:00) and has a size of 4.2 KB, stored in the Standard storage class. The interface includes standard S3 actions like Copy S3 URI, Copy URL, Download, Open, Delete, Actions, Create folder, and Upload.

Name	Type	Last modified	Size	Storage class
run-datasink2-1-part-r-00000	-	April 1, 2022, 22:41:28 (UTC+01:00)	4.2 KB	Standard

25. CREATING A SUB-FOLDER IN THE AWS S3 TRANSFORMED BUCKET TO STORES THE AWS ATHENA QUERY RESULTS

Sub-folder named 'athena_transformed_output_query_results'

The screenshot shows the Amazon S3 console interface. On the left, the navigation sidebar includes options like Buckets, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, Access analyzer for S3, Block Public Access settings for this account, Storage Lens (Dashboards and AWS Organizations settings), Feature spotlight (3 items), and AWS Marketplace for S3. The main content area displays the path: Amazon S3 > Buckets > vp-stock-trades-transformed-bucket > athena_transformed_output_query_results/. The title bar shows 'athena_transformed_output_query_results/'. Below the title, there are tabs for 'Objects' (selected) and 'Properties'. The 'Objects' section shows 'Objects (0)' and a message: '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](#)'. It includes buttons for Copy S3 URI, Copy URL, Download, Open, Delete, Actions (with a dropdown arrow), Create folder (highlighted in orange), and Upload. A search bar says 'Find objects by prefix'. A table header with columns Name, Type, Last modified, Size, and Storage class is shown, followed by a message: 'No objects. You don't have any objects in this folder.' with a central 'Upload' button.

26. EXECUTING AWS GLUE CRAWLER TO CRAWL THE TRANSFORMED DATA STORED IN THE AWS S3 BUCKET (CONTINUED)

AWS Glue Crawler execution started and running

The screenshot shows the AWS Glue service interface. On the left, there's a sidebar with navigation links: Data catalog, Crawlers, Schema registries, Settings, ETL (AWS Glue Studio, Jobs, Jobs (legacy), ML Transforms, Blueprints, Workflows, Triggers, Dev endpoints, Notebooks, Security). The main area is titled "Crawlers" with a sub-instruction: "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." A green banner at the top of the crawler list states: "Crawler 'vp-stock-trades-bucket-glue-crawler' is now running." Below this, there are buttons for "Add crawler", "Run crawler", "Action", and a search bar. The table lists two crawlers:

<input type="checkbox"/>	Name	Schedule	Status	Logs	Last runtime	Median runtime	Tables updated	Tables added
<input type="checkbox"/>	vp-stock-trades-bucket-glue-crawler		Ready	Logs	3 mins	3 mins	0	2
<input type="checkbox"/>	vp-stock-trades-bucket-transformed-glue-cra...		Starting		0 secs	0 secs	0	0

26. EXECUTING AWS GLUE CRAWLER TO CRAWL THE TRANSFORMED DATA STORED IN THE AWS S3 BUCKET

AWS Glue Crawler execution is completed, and the transformed Data Catalogue Tables are created and added to the AWS Glue database specified

The screenshot shows the AWS Glue console interface. On the left, there is a navigation sidebar with the following sections:

- AWS Glue** (selected)
- Data catalog**
 - Databases
 - Tables
 - Connections
- Crawlers** (selected)
 - Classifiers
 - Schema registries
 - Schemas
 - Settings
- ETL**
 - AWS Glue Studio
 - Jobs - New
 - Jobs (legacy)
 - ML Transforms
 - Blueprints
 - Workflows
 - Triggers
 - Dev endpoints
 - Notebooks
- Security

The main content area is titled "Crawlers" and contains the following information:

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.

Crawler "vp-stock-trades-bucket-transformed-glue-crawler" completed and made the following changes: 4 tables created, 0 tables updated. See the tables created in database [vp-stock-trades-bucket-database](#). X

Below this message, there is a table listing the crawlers:

<input type="checkbox"/>	Name	Schedule	Status	Logs	Last runtime	Median runtime	Tables updated	Tables added
<input type="checkbox"/>	vp-stock-trades-bucket-glue-crawler		Ready	Logs	3 mins	3 mins	0	2
<input type="checkbox"/>	vp-stock-trades-bucket-transformed-glue-cra...		Ready	Logs	3 mins	3 mins	0	4

At the top of the table area, there are buttons for "Add crawler", "Run crawler", "Action", and a search bar. To the right, there are links for "User preferences" and "Showing: 1 - 2 < >".

27. VIEWING THE DATA CATALOGUE TABLE CREATED FOR THE TRANSFORMED DATA (CONTINUED)

Data Catalogue Table named 'transformed_spark_transformed_data'

The screenshot shows the AWS Glue Data Catalog interface. On the left, there's a sidebar with navigation links for Data catalog, ETL, and Security. Under Data catalog, 'Tables' is selected, and the 'transformed_spark_transformed_data' table is highlighted. The main pane displays a list of tables with columns for Name, Database, Location, Classification, Last updated, and Deprecated. The 'transformed_spark_transformed_data' table is the second item in the list.

Name	Database	Location	Classification	Last updated	Deprecated
transformed_run_datasink2_1_part_r_00000	vp-stock-trades-bucket-database	s3://vp-stock-trades-transform...	csv	1 April 2022 9:32 AM UTC+1	
transformed_spark_transformed_data	vp-stock-trades-bucket-database	s3://vp-stock-trades-transform...	csv	1 April 2022 11:00 AM UTC+1	
vp_stock_trades_bucketfinancial_stocks_data_txt	vp-stock-trades-bucket-database	s3://vp-stock-trades-bucket/fin...	csv	31 March 2022 12:00 PM UTC+1	
transformed_glue_spark_etl_python_scripts	vp-stock-trades-bucket-database	s3://vp-stock-trades-transform...	Unknown	1 April 2022 9:32 AM UTC+1	
vp_stock_trades_bucketfinancial_stocks_data_csv	vp-stock-trades-bucket-database	s3://vp-stock-trades-bucket/fin...	csv	31 March 2022 12:00 PM UTC+1	
transformed_glue_spark_etl_python_scripts_tem...	vp-stock-trades-bucket-database	s3://vp-stock-trades-transform...	json	1 April 2022 9:32 AM UTC+1	

27. VIEWING THE DATA CATALOGUE TABLE CREATED FOR THE TRANSFORMED DATA (CONTINUED)

Data Catalogue Table named 'transformed_spark_transformed_data'

The screenshot shows the AWS Glue Data Catalog interface. On the left, a sidebar lists various AWS Glue services: Data catalog, Databases, Tables, Connections, Crawlers, Classifiers, Schema registries, Schemas, Settings, ETL (AWS Glue Studio, Jobs, Jobs (legacy), ML Transforms, Blueprints, Workflows, Triggers, Dev endpoints, Notebooks), and Security (Security configurations). The main panel displays the details for the 'transformed_spark_transformed_data' table. The table properties include:

- Name: transformed_spark_transformed_data
- Description: vp-stock-trades-bucket-database
- Database: vp-stock-trades-bucket-database
- Classification: csv
- Location: s3://vp-stock-trades-transformed-bucket/spark_transformed_data/
- Connection: (not specified)
- Deprecated: No
- Last updated: Fri Apr 01 23:00:34 GMT+100 2022
- Input format: org.apache.hadoop.mapred.TextInputFormat
- Output format: org.apache.hadoop.hive.io.HiveIgnoreKeyTextOutputFormat
- Serde serialization lib: org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe
- Serde parameters: field.delim , skip.header.line.count 1 sizeKey 4273 objectCount 1
- Table properties: UPDATED_BY_CRAWLER vp-stock-trades-bucket-transformed-glue-crawler CrawlerSchemaSerializerVersion 1.0 recordCount 36 averageRecordSize 116 CrawlerSchemaDeserializerVersion 1.0 compressionType none columnsOrdered true areColumnsQuoted false delimiter , typeOfData file

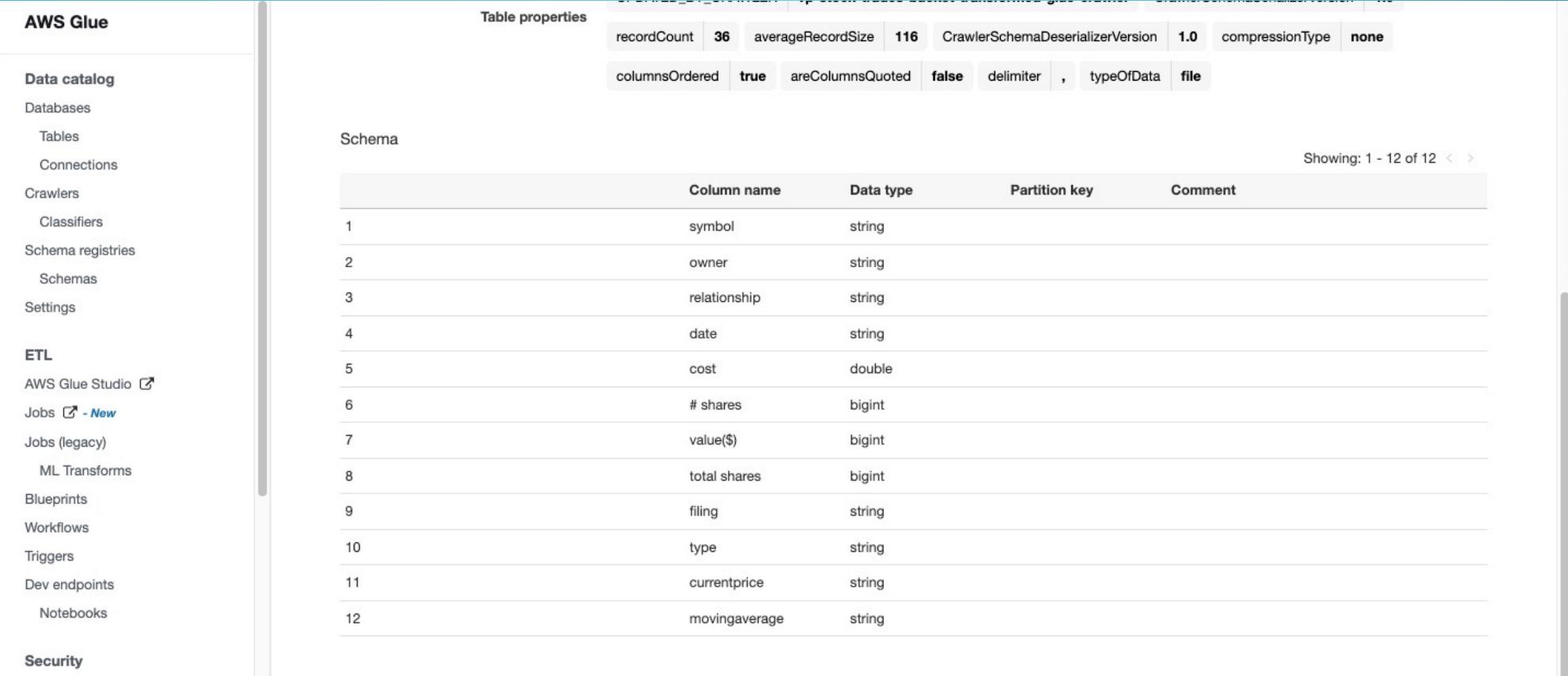
A modal window titled 'Version (Current version)' is open, showing one version entry:

Version	Created	Created by
0	1 April 2022 11:0...	arn:aws:sts::146... role/VP_Glue_Cr...

At the bottom of the main panel, there are two pagination controls: 'Showing: 1 - 12 of 12'.

27. VIEWING THE DATA CATALOGUE TABLE CREATED FOR THE TRANSFORMED DATA

As we have not performed any modifications/transformations to the data schema for the transformed data in the Spark Transformation Python Job; hence the data schema remains same as the source data schema



The screenshot shows the AWS Glue Data Catalog interface. On the left, there's a sidebar with navigation links for AWS Glue, Data catalog, Databases, Tables, Connections, Crawlers, Classifiers, Schema registries, Schemas, Settings, ETL (AWS Glue Studio, Jobs, Jobs (legacy), ML Transforms, Blueprints, Workflows, Triggers, Dev endpoints, Notebooks), and Security. The main area is titled "Table properties" and displays the following details:

recordCount	36	averageRecordSize	116	CrawlerSchemaDeserializerVersion	1.0	compressionType	none
columnsOrdered	true	areColumnsQuoted	false	delimiter	,	typeOfData	file

Below this, the "Schema" section shows a table with 12 columns:

	Column name	Data type	Partition key	Comment
1	symbol	string		
2	owner	string		
3	relationship	string		
4	date	string		
5	cost	double		
6	# shares	bigint		
7	value(\$)	bigint		
8	total shares	bigint		
9	filing	string		
10	type	string		
11	currentprice	string		
12	movingaverage	string		

At the bottom right of the schema table, it says "Showing: 1 - 12 of 12 < >".

28. QUERYING THE TRANSFORMED DATA IN AWS ATHENA (CONTINUED)

AWS Glue

Data catalog

- Databases
- Tables**
- Connections
- Crawlers
- Classifiers
- Schema registries
- Schemas
- Settings

ETL

- AWS Glue Studio
- Jobs - **New**
- Jobs (legacy)
- ML Transforms
- Blueprints
- Workflows
- Triggers
- Dev endpoints
- Notebooks

Security

Tables A table is the metadata definition that represents your data, including its schema. A table can be used as a source or target in a job definition.

Action	Filter by attributes or search by keyword	Save view	Showing: 1 - 6	More
<input type="checkbox"/> Name	Edit table details			
<input type="checkbox"/> transformed	View details			
<input checked="" type="checkbox"/> transformed	View data			
<input type="checkbox"/> vp_stock_trades_bucketfinancial_stocks_data_txt	ata	vp-stock-trades-bucket-database	s3://vp-stock-trades-transform...	csv
<input type="checkbox"/> transformed_glue_spark_etl_python_scripts		vp-stock-trades-bucket-database	s3://vp-stock-trades-transform...	Unknown
<input type="checkbox"/> vp_stock_trades_bucketfinancial_stocks_data_csv		vp-stock-trades-bucket-database	s3://vp-stock-trades-bucket/fin...	csv
<input type="checkbox"/> transformed_glue_spark_etl_python_scripts_tem...		vp-stock-trades-bucket-database	s3://vp-stock-trades-transform...	json

28. QUERYING THE TRANSFORMED DATA IN AWS ATHENA

After executing the SQL query to display all the trade transactions from the transformed data file. The query displays a total of 33 trade transactions as held in the transformed data file as per the spark transformation logic implemented

The screenshot shows the Amazon Athena Query editor interface. The top navigation bar includes tabs for 'Editor' (which is selected), 'Recent queries', 'Saved queries', and 'Settings'. A dropdown for 'Workgroup' is set to 'primary'. On the left, the 'Data' sidebar shows the 'Data Source' as 'AwsDataCatalog' and the 'Database' as 'vp-stock-trades-bucket-database'. Below this, the 'Tables and views' section lists 'transformed_spark_transformed_data' with columns: symbol (string), owner (string), relationship (string), date (string), cost (string), # shares (string), and value(\$) (string). The main area displays 'Query 15' with the SQL command: `1 SELECT * FROM "AwsDataCatalog"."vp-stock-trades-bucket-database"."transformed_spark_transformed_data";`. The status bar indicates the query was completed successfully with 33 results, a run time of 0.467 sec, and 4.17 KB scanned data. The results table shows two rows of data.

#	symbol	owner	relationship	date	cost	# shares	value(\$)
1	FRAF	"Henry Timothy G"	CEO	"Apr 22"	24	40	
2	BKGM	"Ada Patricia Ann Perez"	Director	"Apr 22"	10	61	

29. VIEWING THE AWS ATHENA TRANSFORMED DATA QUERY RESULTS STORED IN AWS S3 TRANSFORMED DATA BUCKET (CONTINUED)

AWS Athena SQL query results are saved within a new folder 'Unsaved' created automatically within the 'athena_transformed_output_query_results' sub-folder

The screenshot shows the Amazon S3 console interface. On the left, the navigation pane is visible with sections like 'Buckets', 'Storage Lens', and 'AWS Marketplace for S3'. The main area shows the path 'Amazon S3 > Buckets > vp-stock-trades-transformed-bucket > athena_transformed_output_query_results/'. The 'athena_transformed_output_query_results/' folder is selected. The 'Objects' tab is active, displaying 'Objects (1)'. A table lists one object: 'Unsaved/' which is a Folder. The table columns include Name, Type, Last modified, Size, and Storage class. Action buttons for Copy S3 URI, Copy URL, Download, Open, Delete, Actions, Create folder, and Upload are available at the top of the object list. A search bar 'Find objects by prefix' is also present.

Name	Type	Last modified	Size	Storage class
Unsaved/	Folder	-	-	-

29. VIEWING THE AWS ATHENA TRANSFORMED DATA QUERY RESULTS STORED IN AWS S3 TRANSFORMED DATA BUCKET (CONTINUED)

AWS Athena SQL query results are saved within a new folder 'Unsaved' created automatically within the 'athena_transformed_output_query_results' sub-folder

Amazon S3 > Buckets > vp-stock-trades-transformed-bucket > athena_transformed_output_query_results/ > Unsaved/ > 2022/ > 04/ > 01/

01/

Copy S3 URI

Objects Properties

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](#)

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	77ef4bb6-6a33-464c-aafe-02af5345286a.csv	csv	April 1, 2022, 23:19:17 (UTC+01:00)	3.8 KB	Standard
<input type="checkbox"/>	77ef4bb6-6a33-464c-aafe-02af5345286a.csv.metadata	metadata	April 1, 2022, 23:19:18 (UTC+01:00)	593.0 B	Standard

29. VIEWING THE AWS ATHENA TRANSFORMED DATA QUERY RESULTS STORED IN AWS S3 TRANSFORMED DATA BUCKET (CONTINUED)

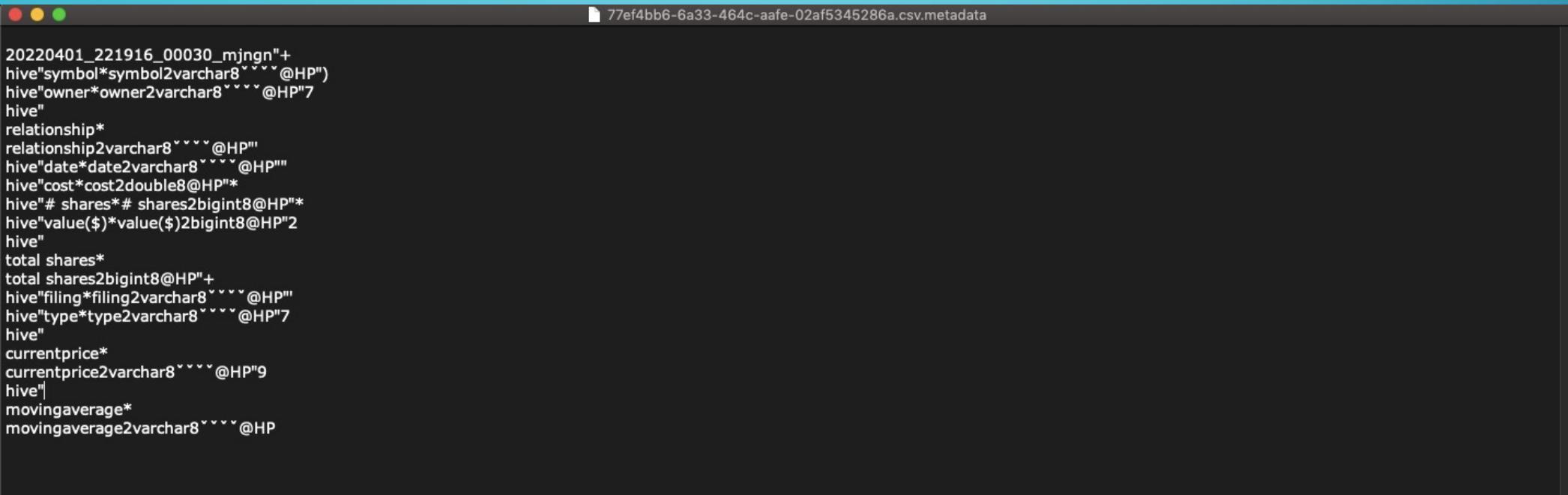
Metadata file

The screenshot shows the AWS S3 Object Details page for a file named `77ef4bb6-6a33-464c-aafe-02af5345286a.csv.metadata`. The page is titled "77ef4bb6-6a33-464c-aafe-02af5345286a.csv.metadata" and includes a "Properties" tab selected, "Permissions" and "Versions" tabs, and "Object actions" buttons for "Copy S3 URI", "Download", "Open", and "Object actions". The "Object overview" section displays the following details:

Property	Value
Owner	4d97ff8f9dfc260e11d18083dc29d77931845b8a284d61ae03f6323dc4a5b186
AWS Region	EU (London) eu-west-2
Last modified	April 1, 2022, 23:19:18 (UTC+01:00)
Size	593.0 B
Type	metadata
Key	athena_transformed_output_query_results/Unsaved/2022/04/01/77ef4bb6-6a33-464c-aafe-02af5345286a.csv.metadata
S3 URI	s3://vp-stock-trades-transformed-bucket/athena_transformed_output_query_results/Unsaved/2022/04/01/77ef4bb6-6a33-464c-aafe-02af5345286a.csv.metadata
Amazon Resource Name (ARN)	arn:aws:s3:::vp-stock-trades-transformed-bucket/athena_transformed_output_query_results/Unsaved/2022/04/01/77ef4bb6-6a33-464c-aafe-02af5345286a.csv.metadata
Entity tag (Etag)	53391138feb432bbce9e4174eae883a7
Object URL	https://vp-stock-trades-transformed-bucket.s3.eu-west-2.amazonaws.com/athena_transformed_output_query_results/Unsaved/2022/04/01/77ef4bb6-6a33-464c-aafe-02af5345286a.csv.metadata

29. VIEWING THE AWS ATHENA TRANSFORMED DATA QUERY RESULTS STORED IN AWS S3 TRANSFORMED DATA BUCKET (*CONTINUED*)

Metadata file



A screenshot of a terminal window displaying the contents of a CSV metadata file named "77ef4bb6-6a33-464c-aafe-02af5345286a.csv.metadata". The file contains the following schema definition:

```
20220401_221916_00030_mjngn"+  
hive"symbol*symbol2varchar8***@HP")  
hive"owner*owner2varchar8***@HP"7  
hive"  
relationship*  
relationship2varchar8****@HP""  
hive"date*date2varchar8***@HP""  
hive"cost*cost2double8@HP"**  
hive"# shares*# shares2bigint8@HP"**  
hive"value($)*value($)2bigint8@HP"2  
hive"  
total shares*  
total shares2bigint8@HP"+  
hive"filings*filing2varchar8***@HP""  
hive"type*type2varchar8***@HP"7  
hive"  
currentprice*  
currentprice2varchar8****@HP"9  
hive"  
movingaverage*  
movingaverage2varchar8***@HP
```

29. VIEWING THE AWS ATHENA TRANSFORMED DATA QUERY RESULTS STORED IN AWS S3 TRANSFORMED DATA BUCKET (*CONTINUED*)

Transformed Data SQL query output results csv file

The screenshot shows the Amazon S3 Object Overview page for a file named `77ef4bb6-6a33-464c-aafe-02af5345286a.csv`. The file was generated by AWS Athena and is located in the `athena_transformed_output_query_results/Unsaved/2022/04/01/` directory of the `vp-stock-trades-transformed-bucket`.

Properties tab selected.

Object overview

Attribute	Value
Owner	4d97ff8f9dfc260e11d18083dc29d77931845b8a284d61ae03f6323dc4a5b186
AWS Region	EU (London) eu-west-2
Last modified	April 1, 2022, 23:19:17 (UTC+01:00)
Size	3.8 KB
Type	csv
Key	athena_transformed_output_query_results/Unsaved/2022/04/01/77ef4bb6-6a33-464c-aafe-02af5345286a.csv

S3 URI: s3://vp-stock-trades-transformed-bucket/athena_transformed_output_query_results/Unsaved/2022/04/01/77ef4bb6-6a33-464c-aafe-02af5345286a.csv

Amazon Resource Name (ARN): [arn:aws:s3:::vp-stock-trades-transformed-bucket/athena_transformed_output_query_results/Unsaved/2022/04/01/77ef4bb6-6a33-464c-aafe-02af5345286a.csv](#)

Entity tag (Etag): [78a9647d656fc02e265059135349b3e5](#)

Object URL: https://vp-stock-trades-transformed-bucket.s3.eu-west-2.amazonaws.com/athena_transformed_output_query_results/Unsaved/2022/04/01/77ef4bb6-6a33-464c-aafe-02af5345286a.csv

Buttons at the top right: Copy S3 URI, Download, Open, Object actions.

29. VIEWING THE AWS ATHENA TRANSFORMED DATA QUERY RESULTS STORED IN AWS S3 TRANSFORMED DATA BUCKET

Transformed Data SQL query output results csv file

symbol	owner	relationship	date	cost	# shares	value(\$)	total shares	filing	type	currentprice	movingaverage
2 FRAF	"Henry Timo"	CEO	"Apr 22		24	40	996	6210	"Apr 24	2020	12:42 PM"
3 BKGM	"Ada Patricia"	Director	"Apr 22		10	61	610	112299	"Apr 23	2020	08:08 PM"
4 CNBKA	"Filler James"	"10% Owner"	"Apr 23		61	100	6175	779933	"Apr 23	2020	04:52 PM"
5 AMAR	"CHEN STEPI"	"President"	Chairman					0.27	100	27	776700
6 PLPC	"GIBBONS N"	Director	"Apr 21		46	23	1074	5640	"Apr 21	2020	02:36 PM"
7 CNBKA	"Delinsky Ste"	Director	"Apr 20		59	46	2746	2790	"Apr 21	2020	11:45 AM"
8 SAMG	"HOUGH RIC"	"Chairman a"	"Apr 20		9	36	340	9586	"Apr 20	2020	08:34 PM"
9 YORW	"Rasmussen Director"	"Apr 20			42	14	593	2276	"Apr 20	2020	03:16 PM"
10 PIH	"Hill John S"	"EVF"	Secretary & CFO		20	100	2060	100	"Apr 20	2020	
11 YORW	"DOTZEL CY"	Director	"Apr 15		44	25	1114	15005	"Apr 20	2020	08:44 AM"
12 YORW	"DOTZEL CY"	Director	"Apr 16		42	15	635	15047	"Apr 20	2020	08:44 AM"
13 AMAR	"CHEN STEPI"	"President"	Chairman					0.22	100	22	776300
14 AMAR	"CHEN STEPI"	"President"	Chairman					0.23	100	23	776600
15 SPLP	"LICHENSTE"	Executive CI	"Apr 16		5	100	500	186912	"Apr 17	2020	07:14 PM"
16 SPLP	"LICHENSTE"	Executive CI	"Apr 15		4	2	10	186812	"Apr 17	2020	07:14 PM"
17 FRAF	"Henry Timo"	CEO	"Apr 15		28	34	986	6169	"Apr 17	2020	03:17 PM"
18 CADXX	"CION Ares M"	LLC"	"10% Owner"			21	46	1002	46	"Apr 17	2020
19 CNBKA	"EVANGELIS"	Executive V	"Apr 16		88	7	617	5931	"Apr 17	2020	02:09 PM"
20 CNBKA	"SLOANE BA"	Chairman	CEO and President"		108	1	108	1691	"Apr 17	2020	
21 GABC	"Ramsey Chr"	Director	"Apr 15		26	64	1678	21931	"Apr 17	2020	02:05 PM"
22 GABC	"Snowden R"	Director	"Apr 15		26	97	2544	31364	"Apr 17	2020	01:54 PM"
23 GABC	"SHEIDLER JJ"	Director	"Apr 15		26	65	1704	51341	"Apr 17	2020	01:38 PM"
24 GABC	"Ryan Christi"	Director	"Apr 15		26	34	892	2997	"Apr 17	2020	01:33 PM"
25 GABC	"KELLY JASO"	Director	"Apr 15		26	65	1704	2549	"Apr 17	2020	01:18 PM"
26 GABC	"Bawel Zach"	Director	"Apr 15		26	97	2544	6986	"Apr 17	2020	01:11 PM"
27 TMDX	"Gordon Stej"	Chief Finan	"Apr 22		18	100	1800	28471	"Apr 24	2020	05:54 PM"
28 AMK	"Goldman CI"	President ai	"Apr 22		22	63	1389	2400578	"Apr 24	2020	05:35 PM"
29 ALBO	"Horn Patrick"	Chief Medic	"Apr 22		17	92	1623	4465	"Apr 24	2020	05:22 PM"
30 ALBO	"Duncan Jasc"	Chief Legal	"Apr 22		17	92	1627	4461	"Apr 24	2020	05:20 PM"
31 CARG	"Steinert Lar"	CEO and Ch	"Apr 22		20	76	1521	254176	"Apr 24	2020	04:26 PM"
32 GGO.A	"GABELLI M"	Director	"Apr 21		40	100	4026	15100	"Apr 23	2020	09:47 PM"
33 FB	"Newstead J"	"VP and Gen"	"Apr 21		100	97	17002	1667	"Apr 23	2020	06:27 PM"
34 ACN	"ROWLAND"	Executive CI	"Apr 20		100	34	5961	22637	"Apr 22	2020	04:11 PM"

30. VIEWING THE AWS CLOUDWATCH LOGS FOR THE TRANSFORMATION AWS GLUE CRAWLER EXECUTED

Transformation AWS Glue Crawler named 'vp-stock-trades-bucket-transformed-glue-crawler'

The screenshot shows the AWS CloudWatch Management Console interface. The left sidebar is titled 'CloudWatch' and includes sections for Favorites, Dashboards, Alarms, Logs (with Log groups selected), Metrics, X-Ray traces, Events, Application monitoring, Insights, Settings, and Getting Started. The main content area shows the log events for the crawler 'vp-stock-trades-bucket-transformed-glue-crawler'. The log events table has columns for Timestamp and Message. The first few log entries are:

Timestamp	Message
2022-04-01T22:57:10.261+01:00	[0f9413bf-b8c8-434d-926f-8aa4d787949b] BENCHMARK : Running Start Crawl for Crawler vp-stock-trades-bucket-transformed-glue...
2022-04-01T22:57:51.999+01:00	[0f9413bf-b8c8-434d-926f-8aa4d787949b] INFO : S3 ConnectionName is vp-stock-trades-bucket-glue-connection
2022-04-01T23:00:11.653+01:00	[0f9413bf-b8c8-434d-926f-8aa4d787949b] BENCHMARK : Classification complete, writing results to database vp-stock-trades-buc...
2022-04-01T23:00:11.654+01:00	[0f9413bf-b8c8-434d-926f-8aa4d787949b] INFO : Crawler configured with SchemaChangePolicy {"UpdateBehavior": "UPDATE_IN_DATAB...
2022-04-01T23:00:34.304+01:00	[0f9413bf-b8c8-434d-926f-8aa4d787949b] INFO : Table transformed_glue_spark_etl_event_logs in database vp-stock-trades-buc...
2022-04-01T23:00:34.690+01:00	[0f9413bf-b8c8-434d-926f-8aa4d787949b] INFO : Created table transformed_spark_transformed_data in database vp-stock-trades...
2022-04-01T23:00:47.080+01:00	[0f9413bf-b8c8-434d-926f-8aa4d787949b] BENCHMARK : Finished writing to Catalog
2022-04-01T23:01:55.724+01:00	[0f9413bf-b8c8-434d-926f-8aa4d787949b] BENCHMARK : Crawler has finished running and is in state READY



END OF THE PROJECT