

• SIMPLIFY DATA TRANSFORMATION FOR ANALYTICS / ML TO UNLOCK INSIGHTS

Wendy Wong
AWS Data Community Builder



AGENDA

- Introduction
- Synopsis
- Use Cases
- High Level Architecture
- Getting Started
- What's New?

Lesson Objectives

In this lesson you will:

- Create IAM permissions for AWS services
- Use AWS Glue and AWS Glue Studio to prepare your data for analytics
- Create an external table using AWS Athena
- Unlock business insights using Amazon QuickSight



INTRODUCTION

+

•

◦

+

•

◦

About Me

- Data Scientist, Senior Data Analyst, Business Analyst, AWS Community Builder
- Lead instructor data analytics at General Assembly
- Former data analytics instructor at Academy Xi
- Former Senior Consultant and Digital Accelerator at PwC Digital Academy
- Stanford University Women in Data Science Sydney Ambassador in 2018 and 2020
- Director of Women in Big Data Sydney Chapter in 2020
- Editor at Towards Data Science 2017-2019



Synopsis

- Real world data is very dirty - from startups, consulting, finance, higher education and government. What's your first use case?
- The journey to cloud, a long transformation project demonstrating the different data analytics life cycle maturity of customers from **descriptive, prescriptive and predictive**.
- If an organization does not have FTE for data engineers, data scientists and business analysts may complete ETL jobs. E.g. Alteryx, Power Query.
- The AWS services used to build a data solution included:



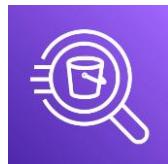
Amazon Simple Storage Service
(Amazon S3)



AWS Identity and Access
Management (IAM)



Amazon QuickSight



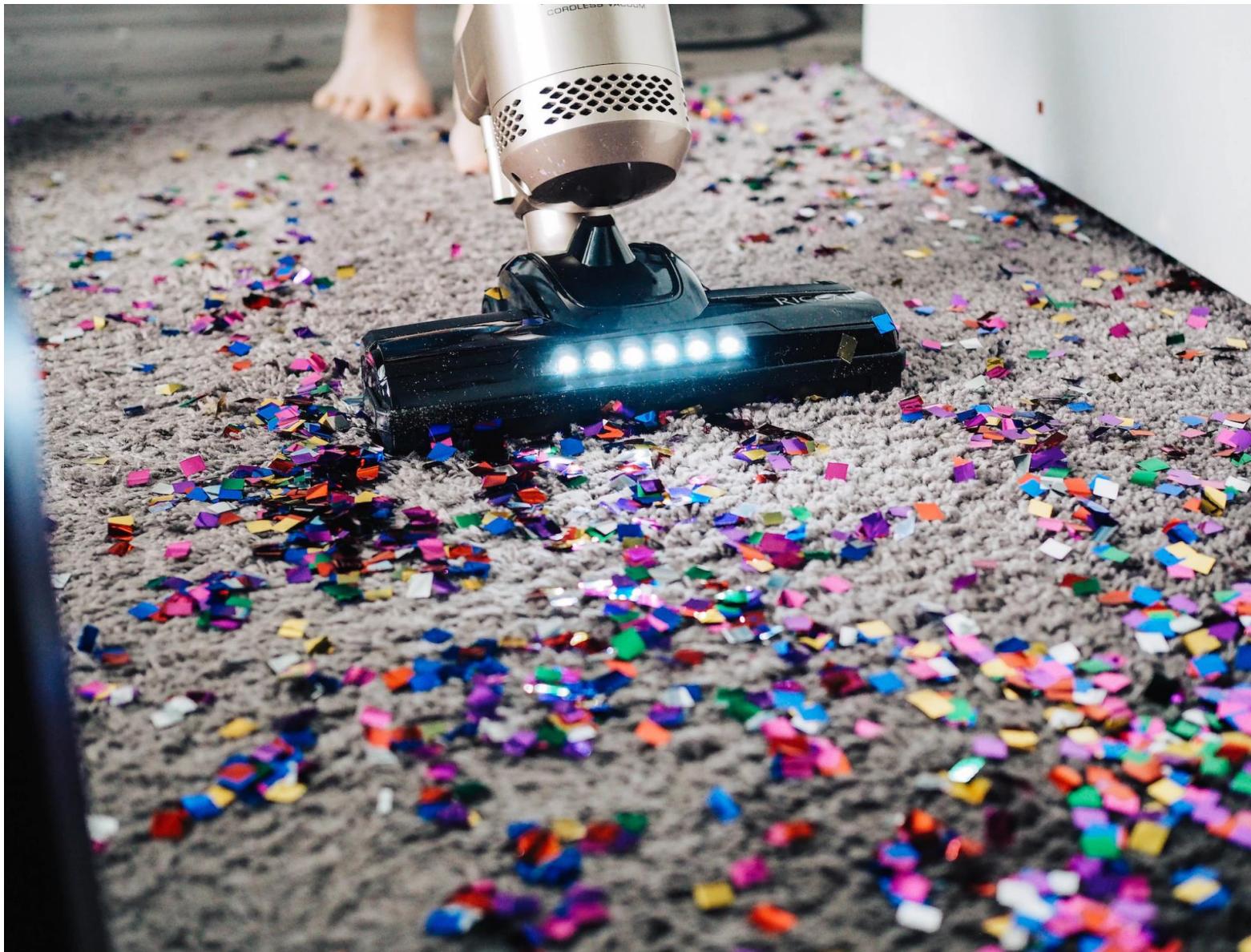
Amazon Athena



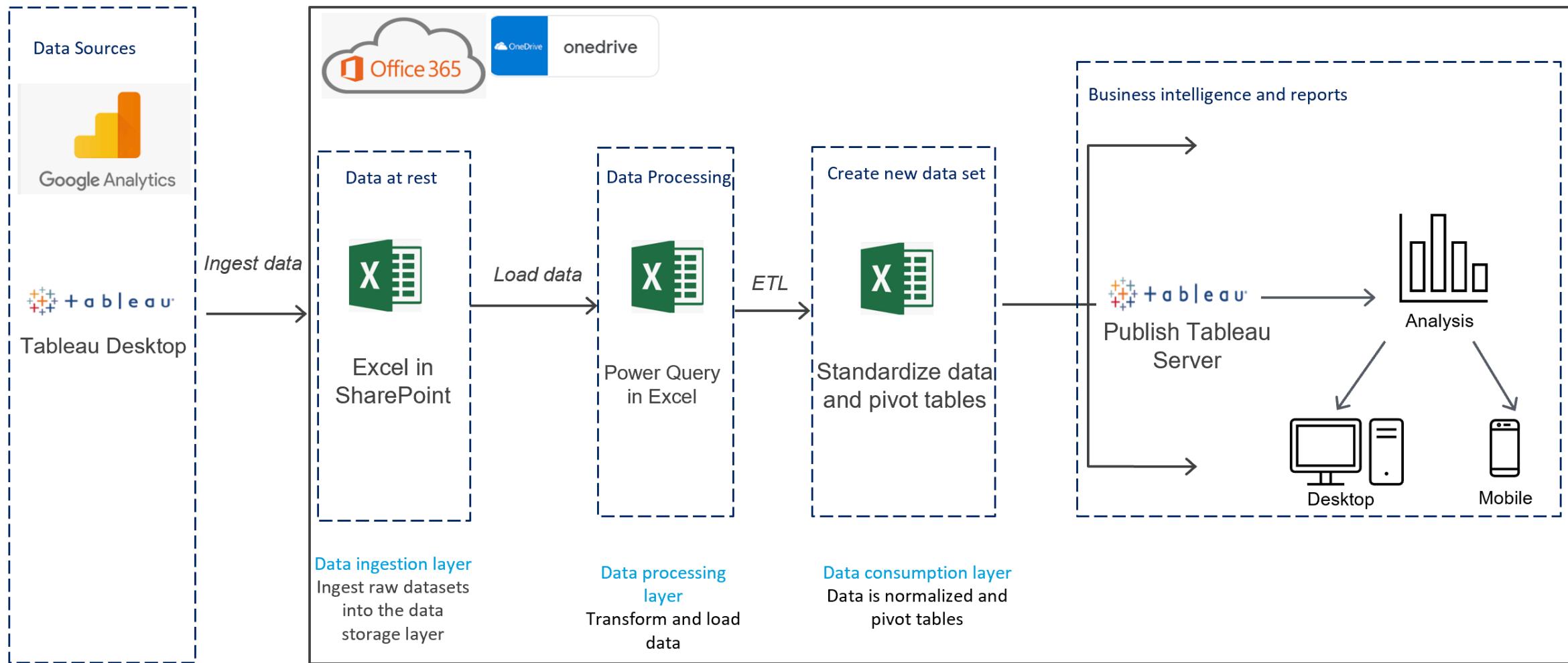
AWS Glue



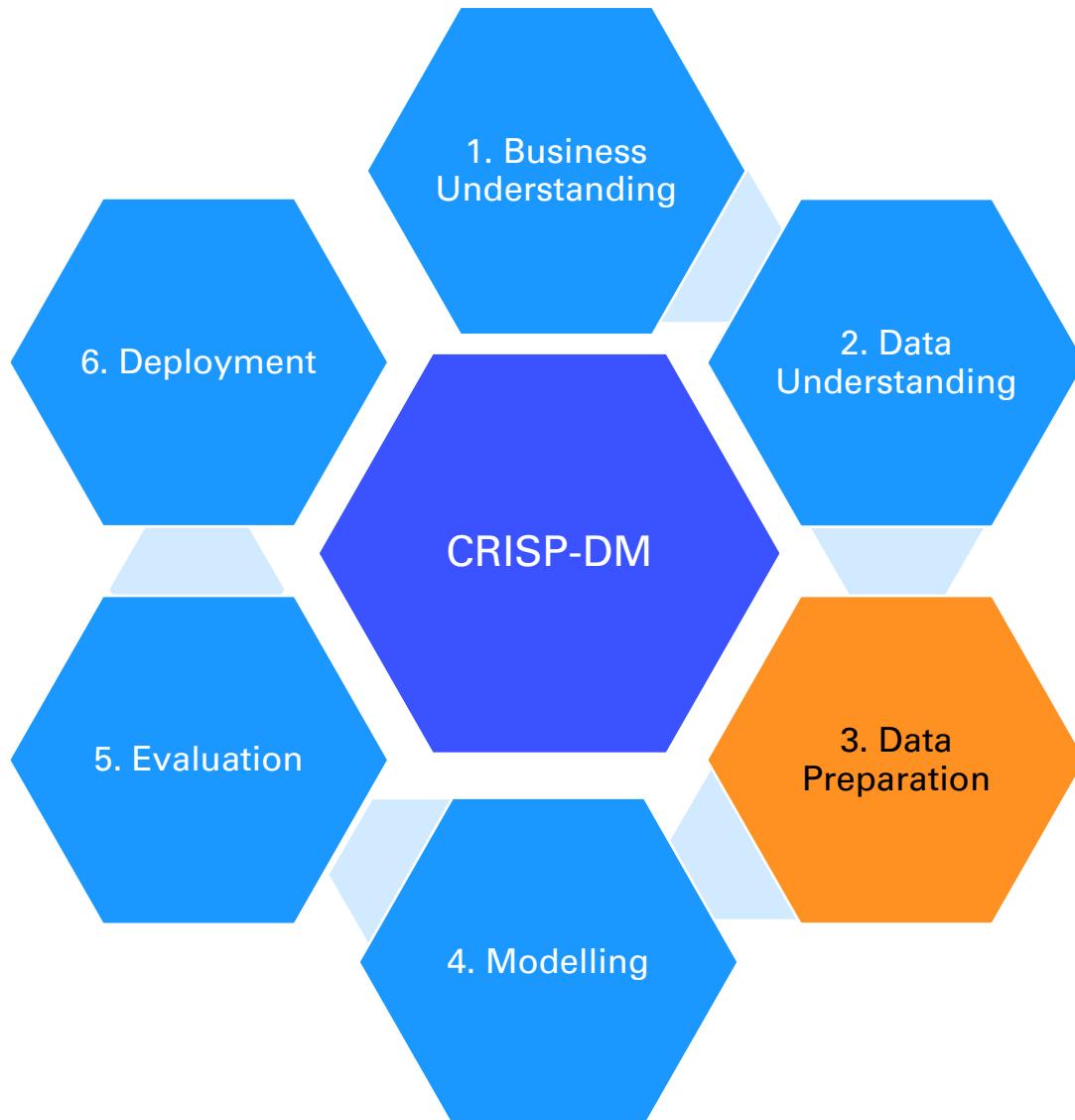
So you want to be a data scientist



What is ETL?



Data Analytics Workflow (CRISP - DM)



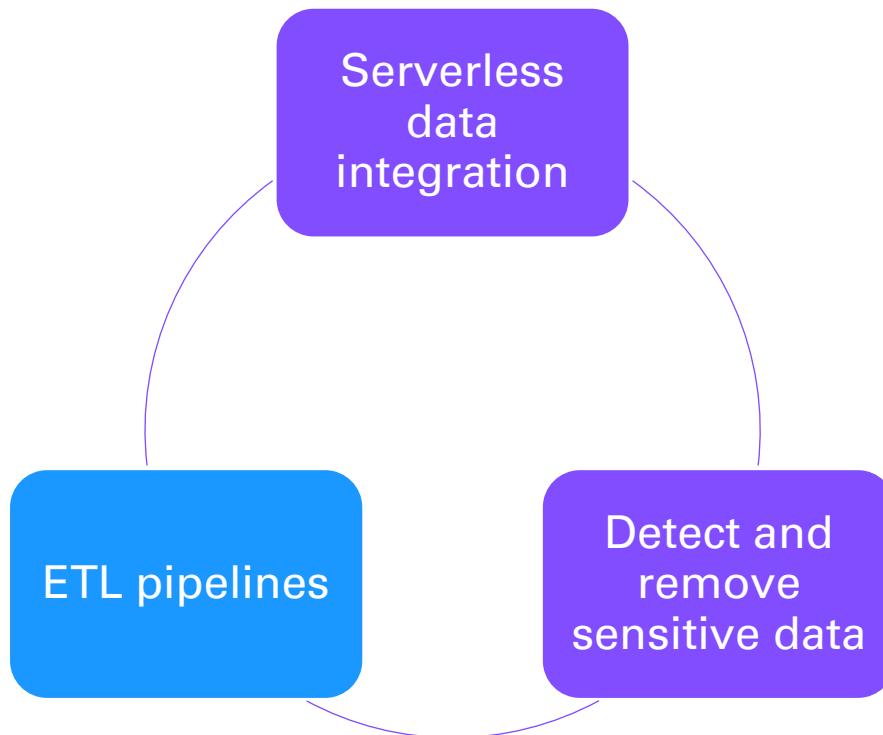
Data Preparation:

- ETL considerations
- Clean and normalize the data

AWS GLUE



What is Amazon Glue?



Languages

- Python, Scala

Benefits

- Catalog data for use in data lake and data warehouse
- Connect to multiple data sources
- You do not need to provision hardware
- **Schedule job** based on trigger event
- Data integration for analytics and ML
- Edit job scripts in AWS Glue Studio

Use Cases



AWS Glue

- Schedule jobs (on **demand** / trigger event)
- Glue Data Catalog
- Discover data properties
- S3, Redshift, AWS databases

Data Engineer

AWS Glue Studio

- Visual editor to create ETL workflows
- **No-code ETL jobs**
- Job monitoring
- Retries
- Generate scripts

Business Analyst,
Data Engineer

AWS Glue DataBrew

- Normalize data
- Inspect outliers
- Format data
- Create new variables

Data Scientist

HIGH LEVEL ARCHITECTURE

+

•

◦

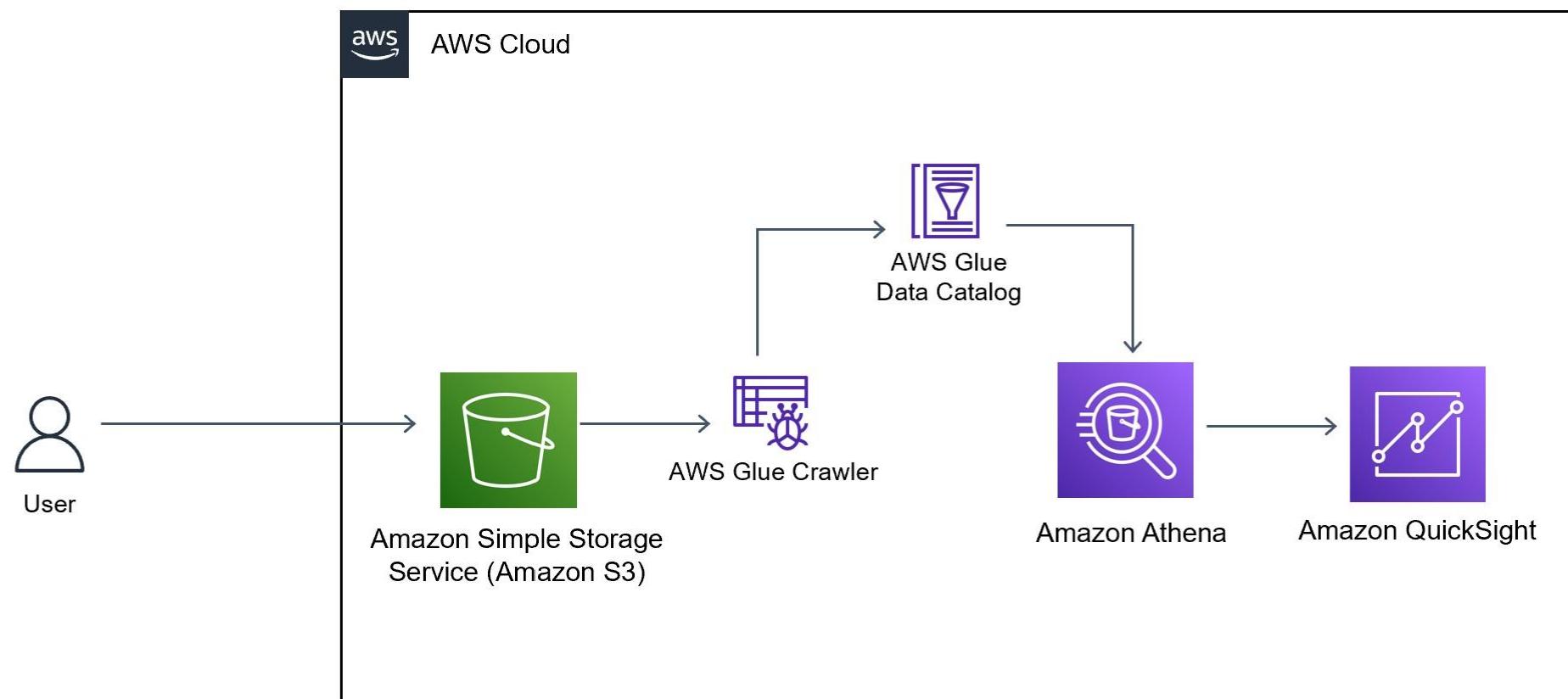
+

•

◦

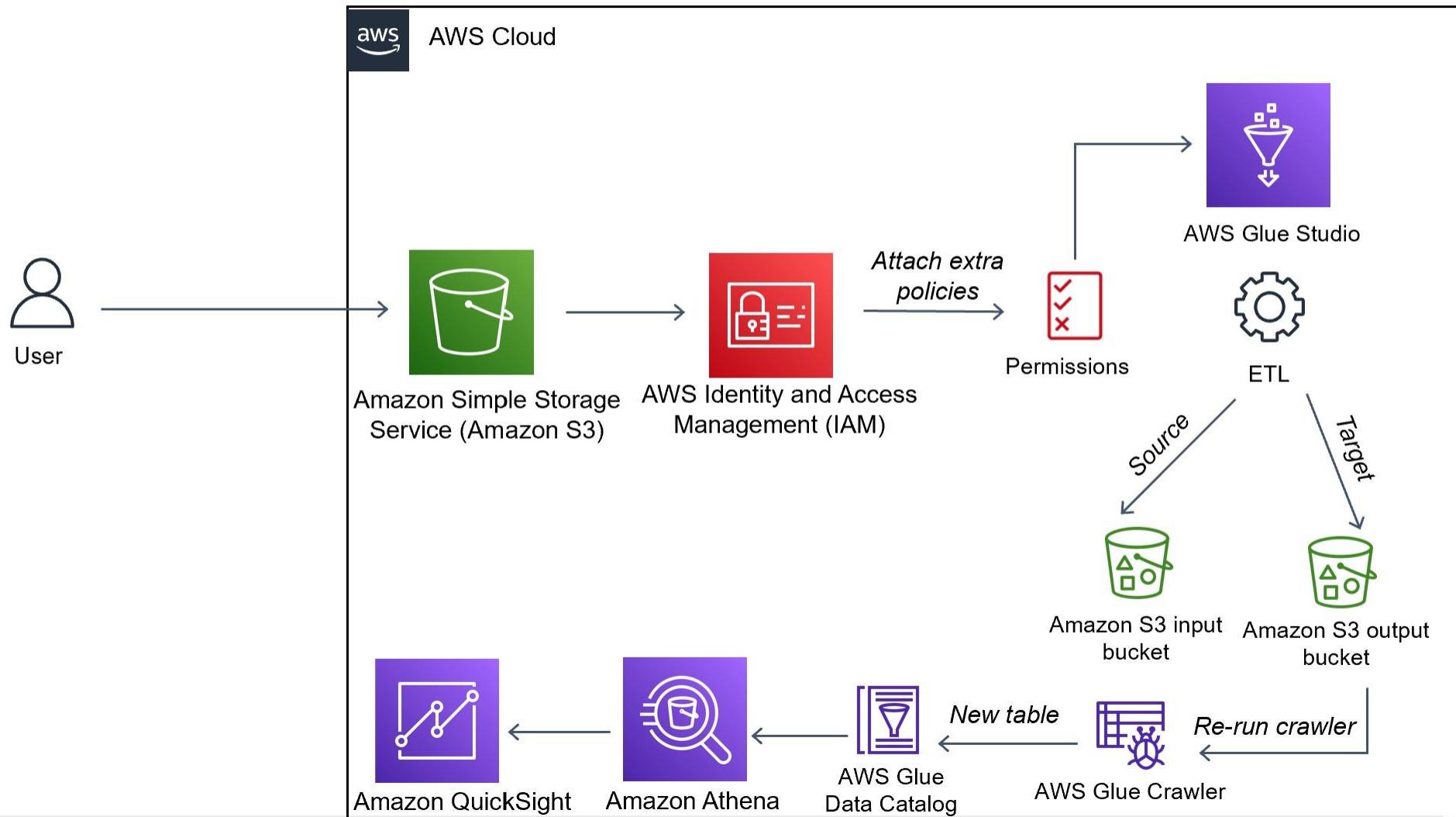
SCHEDULE AWS GLUE JOB

High Level Architecture – Use AWS Glue to crawl and catalog data



AWS GLUE INTERACTIVE

High Level Architecture – Use AWS Glue Studio to perform data ETL



+
•
○

1. ETL – AWS GLUE

+
•
○

Getting started - IAM Permissions

The screenshot shows the AWS IAM Policies page. The left sidebar has 'Identity and Access Management (IAM)' selected. The main area shows a search bar with 'glue' and a results table with 16 matches. The table columns are Policy name, Type, Used as, and Description. The last row, 'AWSGlueConsoleFullAccess', is highlighted.

Policy name	Type	Used as	Description
AWSGlueServiceRole-AWSGlueServiceRole-	Customer managed	Permissions policy (1)	This policy will be u
AWSGlueServiceRole-AWSGlueServiceRole-CrawlerTutorial	Customer managed	Permissions policy (1)	This policy will be u
AWSGlueServiceRole-learn-glue-role	Customer managed	Permissions policy (1)	This policy will be u
AWSGlueServiceNotebookRole	AWS managed	None	Policy for AWS Glu
AWSGlueServiceRole	AWS managed	Permissions policy (4)	Policy for AWS Glu
AWSGlueConsoleSageMakerNotebookFullAccess	AWS managed	None	Provides full access
AWSGlueConsoleFullAccess	AWS managed	Permissions policy (4)	Provides full access

- [AWSGlueConsoleFullAccess](#) and [AmazonS3FullAccess](#)

Upload data into Amazon S3

1	CAMIS	DBA	BORO	BUILDING	STREET	ZIPCODE	PHONE	CUISINE	DE INSPECTION ACTION	VIOLATION	VIOLAT
2	41011076 YAKITORI S	Manhattan		2707 BROADWA		10025	2.13E+09	Japanese	02/14/201 Violations \04K	Eviden	
3	41316819 HIDEAWAY	Manhattan		185 DUANE STF		10013	2.12E+09	American	03/15/202 Violations \04K	Eviden	
4	50050712 PIG BEACH	Brooklyn		480 UNION STR		11231	9.17E+09	Barbecue	10/26/201 Violations \06D	Food c	
5	41420319 OCEANA	Manhattan		120 WEST 49 S		10019	2.13E+09	Seafood	09/13/201 Violations \02G	Cold fo	
6	50040939 WASAN BR	Brooklyn		440 BERGEN ST		11217	3.48E+09	Japanese	03/20/201 Violations \10B	Plumbi	
7	50118108	Brooklyn		398 KNICKERBC		11237	9.18E+09		1/01/1900		
8	41227925 GOOD EAT	Queens	69-32	GRAND AV		11378	7.18E+09	American	5/09/2019 Violations \10F	Non-fo	
9	50077587 WYCKOFF F	Brooklyn		250 WYCKOFF A		11237	3.47E+09	Mexican	05/30/201 Violations \10B	Plumbi	
10	41706711 NOT GUILT	Staten Islar		19 HYATT STR		10301	7.18E+09	Sandwiche	10/17/201 Establishm\06C	Food n	
11	41210703 LOS POTRIL	Staten Islar		150 PORT RICHI		10302	7.19E+09	Mexican	07/19/202 Violations \08A	Facility	
12	50003047 ELI ZABAR	Manhattan		922 MADISON /		10021	6.47E+09	American	03/17/202 Violations \02G	Cold fo	
13	50076675 ELIM BISTR	Manhattan		11 PARK PLAC		10007	2.13E+09	Sandwiche	07/23/201 Violations \10B	Plumbi	
14	50101465 NEW CHAN	Bronx		2570 BRONXWOO		10469	7.18E+09	Pakistani	8/03/2022 Violations \10B	Anti-sij	
15	50101465 NEW CHAN	Bronx		2570 BRONXWOO		10469	7.18E+09	Pakistani	8/03/2022 Violations \10B	Anti-sij	
16	41579144 SALT AND F	Manhattan	139 WEST 33 S			10001	2.12E+09	Soups	08/16/202 Violations \04N	Filth fli	
17	50059509 BARN JOO	Manhattan		35 UNION SQU		10003	6.46E+09	Korean	2/12/2018 Violations \10B	Plumbi	
18	50104728 El Gallo Ta	Manhattan		369 BROOME S		10013	9.17E+09	Mexican	01/26/202 Violations \02B	Hot foo	
19	50080649 TURKO'S G	Brooklyn		110 MOORE ST		11206	9.18E+09	Mediterran	6/05/2019 No violatio	Bulb nc	
20	50004456 JUSTINO'S I	Manhattan		77 PEARL STR		10004	2.13E+09	Pizza	1/04/2019 Violations \10F	Non-fo	
21	41168300 ORIGINAL I	Brooklyn		594 CRESCENT :		11208	7.19E+09	Pizza	08/17/202 Violations \02G	Cold fo	
22	50044246 FLEET BAKE	Manhattan		24 BOWERY		10013	6.46E+09	Bakery Pro	05/14/201 Violations \06B	Tobacc	
23	50048697 DYCKMAN	Manhattan		100 DYCKMAN		10040	2.13E+09	American	08/29/201 Violations \02B	Hot foo	
24	DOHMH_NYC_R	DATA NYC		11100	7.18E+09	000	2.13E+09	000	01/01/2019 00:00:00	000	

Amazon S3 > Buckets > new-york-restaurant-inspections > raw/ > Upload

Upload Info

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. [Learn more](#)

Drag and drop files and folders you want to upload here, or choose [Add files](#), or [Add folders](#).

Files and folders (1 Total, 106.2 MB)					
<input type="checkbox"/>	Name	Folder	Type	Size	
<input type="checkbox"/>	DOHMH_New_York_City_Restaurant_Inspection_Results.csv	-	text/csv	106.2 MB	

Destination

Destination

<s3://new-york-restaurant-inspections/raw/>

Destination details

Bucket settings that impact new objects stored in the specified destination.

AWS Glue Crawler

Add crawler X

Crawler info Add information about your crawler

Crawler source type

Data store

IAM Role

Schedule

Output

Review all steps

Crawler name
new-york-restaurant-inspection-crawler

▶ Tags, description, security configuration, and classifiers (optional)

Next

AWS Glue Crawler

Add crawler

Crawler info
new-york-restaurant-inspection-crawler

Crawler source type
Data stores

Data store
S3: s3://new-york-re...

IAM Role

Schedule

Output

Review all steps

Add a data store

Choose a data store

S3

Connection

Select a connection

Optional: include a Network connection to use with this S3 target. Note that each crawler is limited to one Network connection so any future S3 targets will also use the same connection (or none, if left blank).

Add connection

Crawl data in

Specified path in my account

Specified path in another account

Include path

s3://new-york-restaurant-inspections/processed

All folders and files contained in the include path are crawled. For example, type s3://MyBucket/MyFolder/ to crawl all objects in MyFolder within MyBucket.

Sample size (optional)

Enter a number between 1 and 249

This field sets the number of files in each leaf folder to be crawled. If not set, all the files are crawled.

Exclude patterns (optional)

Back

Next

AWS Glue Crawler

Add crawler

Crawler info
new-york-restaurant-inspection-crawler

Crawler source type
Data stores

Data store
S3: s3://new-york-re...
S3: s3://new-york-re...

IAM Role

role/AWSGlueServiceRole-learn-glue-role

Schedule
Run on demand

Output
restaurant-inspections

Review all steps

Crawler info

Name new-york-restaurant-inspection-crawler
Tags -

Data stores

Data store S3
Include path s3://new-york-restaurant-inspections/raw/DOHMH_New_York_City_Restaurant_Inspection_Results.csv
Connection
Exclude patterns

Data store S3
Include path s3://new-york-restaurant-inspections/processed
Connection
Exclude patterns

IAM role

IAM role 
role/AWSGlueServiceRole-learn-glue-role

Schedule

Schedule Run on demand

Output

Database restaurant-inspections

Check crawler details

AWS Glue Data Catalog

AWS Glue

Data catalog

Databases

Tables

Connections

Crawlers

Classifiers

...

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.

Add tables Action Filter by attributes or search by keyword Save view Showing: 1 - 2 ?

Name	Database	Location	Classification	Last updated	Deprecated
<input type="checkbox"/> rawdohmh_new_york_city_restaurant_inspection_...	restaurant-inspections	s3://new-york-restaurant-inspecti...	csv	9 August 2022 8:40 AM UTC+10	
<input type="checkbox"/> processed	restaurant-inspections	s3://new-york-restaurant-inspecti...	parquet	9 August 2022 11:06 AM UTC+10	

AWS Glue Data Catalog

Schema

	Column name	Data type	Partition key	Comment
1	camis	bigint		
2	dba	string		
3	boro	string		
4	building	string		
5	street	string		
6	zipcode	bigint		
7	phone	string		
8	cuisine description	string		
9	inspection date	string		
10	action	string		
11	violation code	string		
12	violation description	string		
13	critical flag	string		
14	score	bigint		
15	grade	string		
16	grade date	string		
17	record date	string		
18	inspection type	string		

Showing: 1 - 26 of 26 < >

Edit Schema



2. ETL – AWS GLUE STUDIO

AWS Glue Studio



AWS Glue Studio > Jobs

Jobs Info

Create job Info

Visual with a source and target
Start with a source, ApplyMapping transform, and target.

Visual with a blank canvas
Author using an interactive visual interface.

Spark script editor
Write or upload your own Spark code.

Python Shell script editor
Write or upload your own Python shell script.

Jupyter Notebook
Write your own code in a Jupyter Notebook for interactive development.

Ray script editor New
Write your own code to run on Ray.

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

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

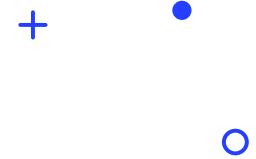
Your jobs (0) Info

Actions ▾ Run job

Find jobs

Job name	Type	Last modified	AWS Glue version
No jobs			
You have not created a job yet.			
<input type="button" value="Create job from a blank graph"/>			

AWS Glue Studio



restaurant inspection job 1

⚠ Job has not been saved Save Delete Actions ▾ Run

Visual 1 Script Job details 1 Runs Schedules

Source Transform Target Undo Redo Remove

Node properties Data source properties - S3 Output schema Data preview

S3 source type [Info](#)

Data Catalog table

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

Database
Choose a database.
 C

Table
 C

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

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

```
graph TD; A[Data source - S3 bucket] --> B[Transform - ApplyMapping];
```

AWS Glue Studio

restaurant inspection job

⚠ Job has not been saved

Visual Script Job details Runs Schedules

Source Transform Target Undo Redo Remove

Node properties Data preview

Apply mapping

Source key	Target key	Data type	Drop
camis	camis	long ▾	<input type="checkbox"/>
dba	dba	string ▾	<input type="checkbox"/>
boro	boro	string ▾	<input type="checkbox"/>
building	building	string ▾	<input type="checkbox"/>
street	street	string ▾	<input type="checkbox"/>
zipcode	zipcode	long ▾	<input type="checkbox"/>
phone	phone	string ▾	<input type="checkbox"/>
cuisine description	cuisine description	string ▾	<input type="checkbox"/>
inspection date	inspection date	string ▾	<input type="checkbox"/>

```
graph TD; S[S3 bucket] --> T[Transform - ApplyMapping]; style T fill:#0072BD,color:#fff
```

Edit Schema

AWS Glue Studio

restaurant inspection job

Job has not been saved

Save Delete Actions Run

Visual 1 Script Job details 1 Runs Schedules

Source Transform Target Undo Redo Remove Node properties Transform Output schema Data preview

Data source - S3 bucket
S3 bucket

Transform - ApplyMapping
ApplyMapping

Data target - S3 bucket
S3 bucket

Schema

Key Data type

Key	Data type
camis	long
dba	string
boro	string
building	string
street	string
zipcode	long
phone	string
cuisine description	string
inspection date	date
action	string
violation code	string
violation description	string

```
graph TD; Source[Data source - S3 bucket<br/>S3 bucket] --> Transform[Transform - ApplyMapping<br/>ApplyMapping]; Transform --> Target[Data target - S3 bucket<br/>S3 bucket]
```

AWS Glue Studio



restaurant inspection job

Job has not been saved Actions Run

Visual Script Job details Runs Schedules

Source Transform Target Undo Redo Remove

Data target properties - S3

Format Parquet

Compression Type Snappy

53 Target Location Choose an S3 location in the format s3://bucket/prefix/object/ with a trailing slash (/).
s3://new-york-restaurant-inspections/processed/

Data Catalog update options:
Choose how you want to update the Data Catalog table's schema and partitions. These options will only apply if the Data Catalog table is an S3 backed source.
 Do not update the Data Catalog
 Create a table in the Data Catalog and on subsequent runs, update the schema and add new partitions
 Create a table in the Data Catalog and on subsequent runs, keep existing schema and add new partitions

Partition keys - optional
Add partition keys

```
graph TD; S1[Data source - S3 bucket<br/>S3 bucket] --> T1[Transform - ApplyMapping<br/>ApplyMapping]; T1 --> S2[Data target - S3 bucket<br/>S3 bucket]
```

AWS Glue Studio



restaurant inspection job

⚠ Job has not been saved

Save

End session

Delete

Actions ▾

Run

Visual

Script

Job details

Runs

Schedules

Script (Locked) Info

Generate classic script

Download script

Edit script

```
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 args = getResolvedOptions(sys.argv, ["JOB_NAME"])
9 sc = SparkContext()
10 glueContext = GlueContext(sc)
11 spark = glueContext.spark_session
12 job = Job(glueContext)
13 job.init(args["JOB_NAME"], args)
14
15 # Script generated for node S3 bucket
16 S3bucket_node1 = glueContext.create_dynamic_frame.from_catalog(
17     database="restaurant-inspections",
18     table_name="rawdohnh_new_york_city_restaurant_inspection_results_csv",
19     transformation_ctx="S3bucket_node1",
20 )
21
22 # Script generated for node ApplyMapping
23 ApplyMapping_node2 = ApplyMapping.apply(
24     frame=S3bucket_node1,
25     mappings=[
26         ("camis", "long", "camis", "long"),
27         ("dba", "string", "dba", "string"),
28         ("boro", "string", "boro", "string"),
29         ("building", "string", "building", "string"),
30         ("street", "string", "street", "string"),
31         ("zipcode", "long", "zipcode", "long"),
32         ("phone", "string", "phone", "string"),
33         ("cuisine description", "string", "cuisine description", "string"),
34         ("inspection date", "string", "inspection date", "date"),
35         ("action", "string", "action", "string"),
36         ("violation code", "string", "violation code", "string"),
37         ("violation description", "string", "violation description", "string"),
38         ("critical flag", "string", "critical flag", "string"),
39     ],
40     transformation_ctx="ApplyMapping_node2"
41 )
42
43 # Script generated for node Drop Fields
44 DropFields_node3 = DropFields.apply(
45     frame=ApplyMapping_node2,
46     fields=["camis", "dba", "boro", "building", "street", "zipcode", "phone", "cuisine description", "inspection date", "action", "violation code", "violation description", "critical flag"],
47     transformation_ctx="DropFields_node3"
48 )
49
50 # Script generated for node Sink
51 Sink_node4 = Sink.apply(
52     frame=DropFields_node3,
53     connection_type="s3",
54     connection_options={
55         "path": "s3://gluestudio-123456789012-us-east-1/restaurant-inspections/rawdohnh_new_york_city_restaurant_inspection_results_csv/_SUCCESS"
56     },
57     format="json",
58     transformation_ctx="Sink_node4"
59 )
60
61 job.commit()
```

3.CREATE AN EXTERNAL TABLE WITH AMAZON ATHENA

Getting started - IAM Permissions

Identity and Access Management (IAM) X

Search IAM

Dashboard

▼ Access management

- User groups
- Users
- Roles

Policies

- Identity providers
- Account settings

▼ Access reports

i Introducing the new Policies list experience
We've redesigned the Policies list experience to make it easier to use.

IAM > Policies

Policies (959) Info
A policy is an object in AWS that defines permissions.

Filter policies by property or policy name and press enter

"athena" X Clear filters

Policy name
<input type="radio"/> + AWSQuicksightAthenaAccess
<input type="radio"/> + AmazonAthenaFullAccess

Check S3 can access Amazon Athena

The screenshot shows the AWS Athena Settings page. At the top, there is a navigation bar with various services: IAM, S3, Amazon Redshift, AWS Glue, AWS Glue DataBrew, QuickSight, Athena, Step Functions, and CloudFormation. Below the navigation bar, a green success message box displays the text "Settings successfully updated." with a checkmark icon. To the right of the message box is a close button (X). The main content area has a breadcrumb navigation path: "Amazon Athena > Query editor". Below the path, there are tabs: "Editor", "Recent queries", "Saved queries", and "Settings", with "Settings" being the active tab. On the right side, there is a "Workgroup" dropdown set to "primary". The "Query result and encryption settings" section contains four items: "Query result location and encryption", "Query result location" (with a redacted value), "Encrypt query results" (set to "-"), "Expected bucket owner" (with a redacted value), and "Assign bucket owner full control over query results" (set to "Enabled"). A "Manage" button is located in the top right corner of this section.

Create external table

The screenshot shows the Amazon Athena Query editor interface. At the top, there's a green success message: "Query saved Create_table_from_S3 was successfully saved." Below the header, the navigation bar includes "Amazon Athena > Query editor". The tabs at the top are "Editor" (which is selected), "Recent queries", "Saved queries", and "Settings". The "Workgroup" dropdown is set to "prim".

The main area is divided into two sections: "Data" on the left and the "Query editor" on the right.

Data Section:

- Data source:** AwsDataCatalog
- Database:** sales_data
- Tables and views:** A list showing three tables: processed, processed_sales, and processed_table.

Query Editor Section:

- Query Name:** Create_table_from_S3
- SQL Query:** The code is as follows:

```
1 CREATE EXTERNAL TABLE IF NOT EXISTS `sales_data`.`Processed_table` ()  
2   `region` string,  
3   `ship_date` date,  
4   `total_profit` double,  
5   `total_revenue` double,  
6   `total_cost` double,  
7   `id` string,  
8   `country` string,  
9   `item_type` string,  
10  `sales_channel` string,  
11  `order_priority` string,  
12  `order_date` date,  
13  `order_id` string,  
14  `units_sold` bigint,  
15  `unit_price` double,
```
- Status:** Completed
- Metrics:** Time in queue: 106 ms, Run time: 1.155 sec



4. CREATE VISUALIZATION WITH AMAZON QUICKSIGHT

Upload data from Amazon Athena

The screenshot shows the AWS QuickSight console interface. On the left, there's a sidebar with a 'Datasets' tab selected. Below it, a 'Create a Dataset' section is visible, with a 'FROM NEW DATA SOURCES' button and a 'Upload a file' button. The main area displays a grid of data source icons, including Athena, RDS, MySQL, Aurora, Teradata, Exasol, SalesForce Connect, Redshift (Auto-discovered), Redshift (Manual connect), PostgreSQL, ORACLE, MariaDB, Presto, Snowflake, AWS IoT Analytics, Amazon OpenSearch Service, GitHub, Twitter, and Jira.

A modal dialog box titled 'Finish dataset creation' is open in the center. It contains the following information:

- Table: processed_table
- Data source: Processed_data
- Schema: sales_data

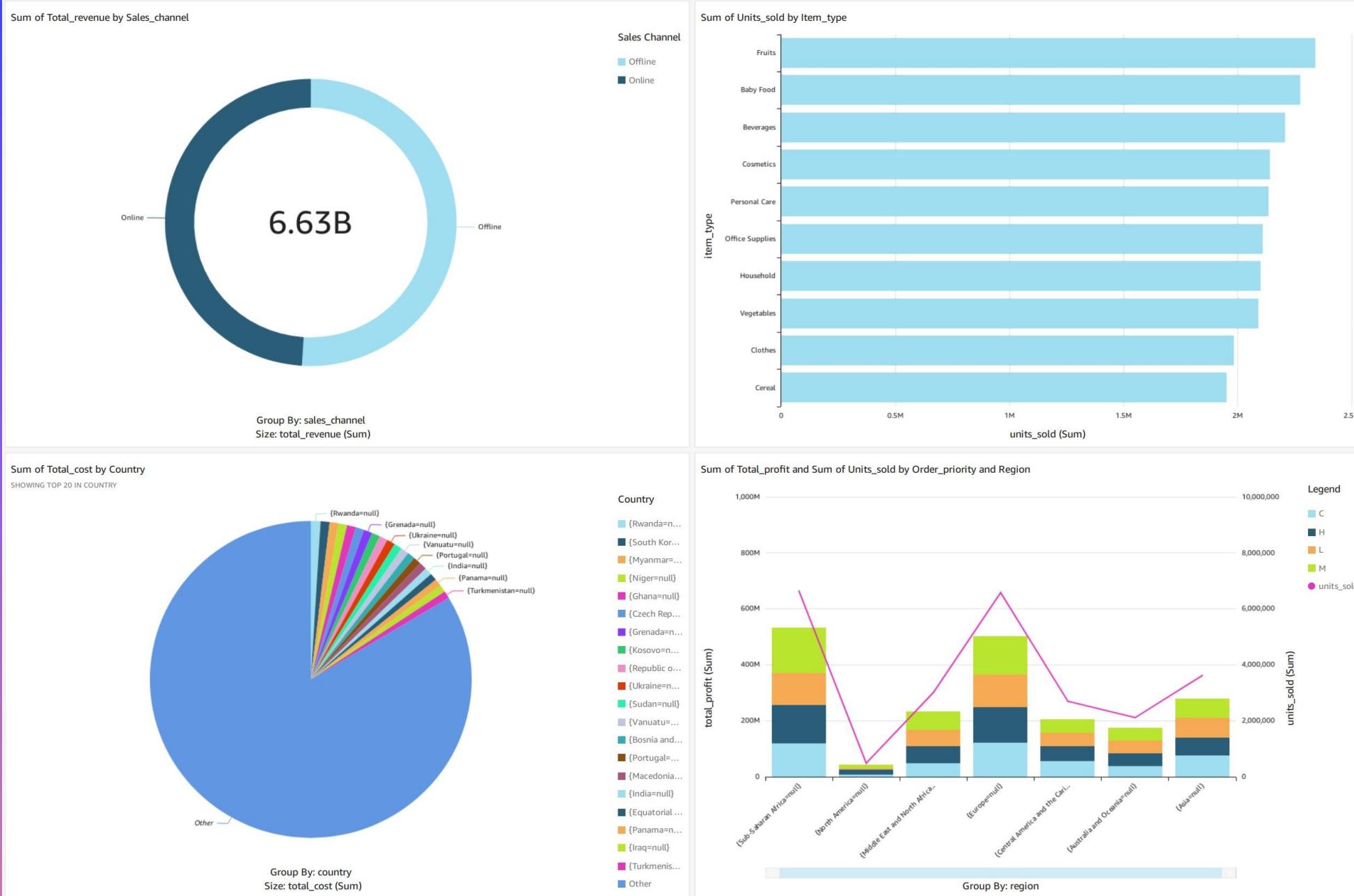
Below this, there are two radio button options:

- Import to SPICE for quicker analytics ✓ 10GB available SPICE
- Directly query your data

There is also a checkbox: Email owners when a refresh fails.

At the bottom of the dialog are two buttons: 'Edit/Preview data' (highlighted with a blue border) and 'Visualize'.

AMAZON QUICKSIGHT





WHAT'S NEW?



Amazon re:Invent 2022



AWS Re:invent 2022

- Keynotes, Leadership sessions
- Data, Analytics
- AI/Machine Learning

<https://www.youtube.com/user/amazonwebservices>

Preview: AWS Glue Data Quality

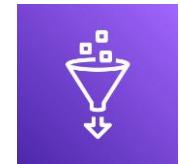
Preview

- Automatic data quality recommendations on your data, based on rules

The screenshot shows the AWS Glue Data Quality interface in visual mode. At the top, there are tabs for Visual, Script, Job details, Runs, Data quality, Schedules, and Version Control. Below the tabs is a toolbar with icons for Source, Action, Target, Undo, Redo, Remove, and search. A search bar is positioned above a list of data quality actions. The actions listed are:

- Change Schema (Apply Mapping)
- Join
- SQL Query
- Detect Sensitive Data
- Evaluate Data Quality
- Fill Missing Values
- Aggregate
- Custom Transform
- Drop Duplicates

Each action has a brief description below it. To the right of the list, there are three rectangular boxes representing datasets: "Source - S3 bucket", "Transform - ApplyMapping", and "Target - S3 bucket". Arrows indicate a flow from the source to the transform step and then to the target.



Source: AWS Glue Documentation

+



THANK YOU

https://dev.to/abc_wendsss