Laboratory Practice Report

Practice 12

November 22, 2023

Computer Systems Engineering

Cloud Architecture

Prof. M.S. Rodolfo Luthe Ríos

Marco Ricardo Cordero Hernández

is727272@iteso.mx

ITESO

Universidad Jesuita de Guadalajara

Abstract

The current development has the goal of following a step-by-step tutorial on how to handle big volumes of data in order to provide insightful presentations that could eventually lead to a business success case.

Inside the present contents, cloud usage advantages will also be demonstrated, as several remote services are used to achieve the stated goals in a seamless and direct way, reducing significantly configuration time to jump straight to action.

As the custom follows, a well known public cloud provider will be used to not only demonstrate the implementation, but to understand this same implementation, as it is possible by a workshop hosted by the same entity.

State of the Art

It is not strange to heard nowadays that data is the new oil, even stated before in the contents of the current course. Greed always makes a fragile person to want more, to desire quantity with preferably quality, so it’s not strange to find concepts such as data mining, scrapping, and more benevolent, **data lakes**.

As defined by AWS [1], a data lake is a centralized repository that allows to store all structured and unstructured data at any scale. In fact, data can be stored without having to first structure it, and that’s when the concept of ETL comes into play; ETL means Extract, Transform, and Load [2], an abstract maneuver of processing data at different stages, being them at start, progress, and end, as a general rule.

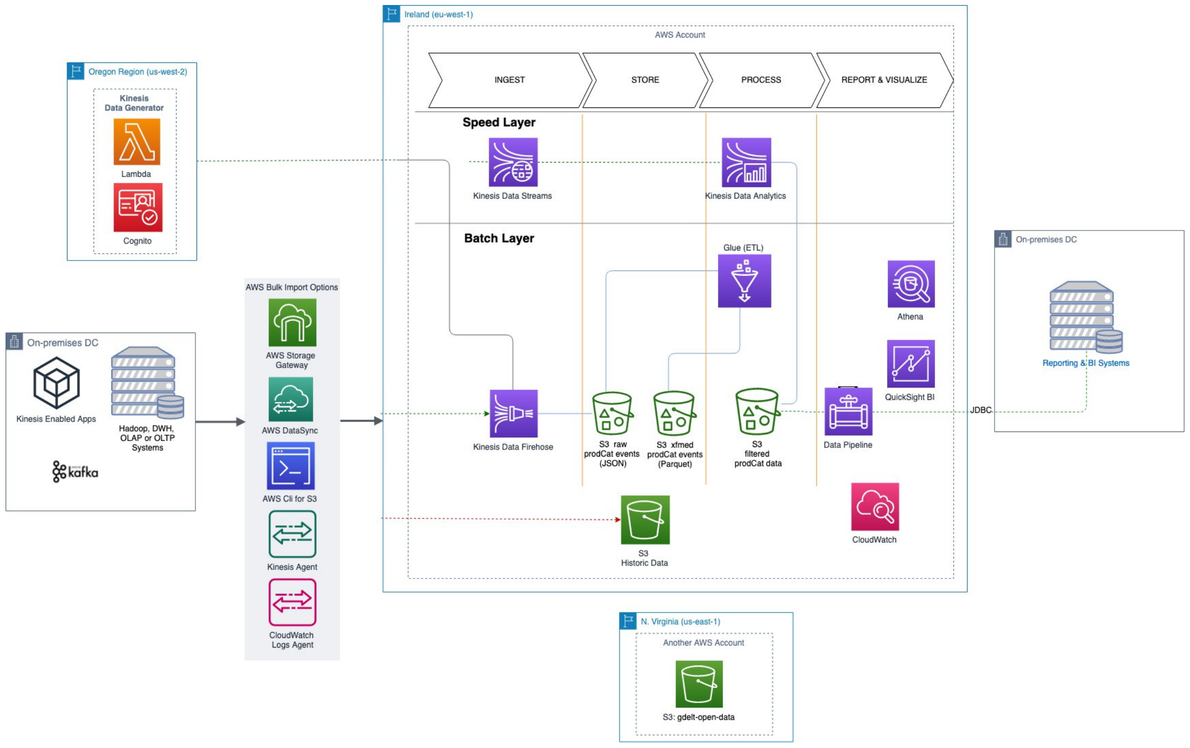
Both data lakes and ETL concepts are strongly linked, as having sparse data contained in the same location can result troublesome for analytics matters, although beneficial for ease of access. When combined together, business intelligence [3] solutions can be made in matter of mere hours having a well structured data flow or at least big amounts of information.

The current development has the goal of demonstrating several service usage through AWS provided resources in a public workshop made available exclusively for these matters. Although the complete implementation is a really heavy duty, each individual step demonstrates the platform’s ability to link resources across several domains of use cases to address a common problematic.

The impact of doing this kind of developments resides in the importance of data, as in its pure state of simple bytes stored in disk arrays doesn’t mean nothing, but, when taken through an appropriate process of handlers, it can even give support to enterprises devoted solemnly to provide other business with insightful reports to control their operations better, being this for sale optimization matters or decision taking acts.

Diagram

In this extremely rare occasion, the solution's diagram is provided. It looks like the following:



Although convenient, some discrepancies can be spotted, mainly the regions, as this proposal states the usage of European Union regions. This can be overlooked, but surely would make performance impacts in a real scenario, due to geographical circumstances.

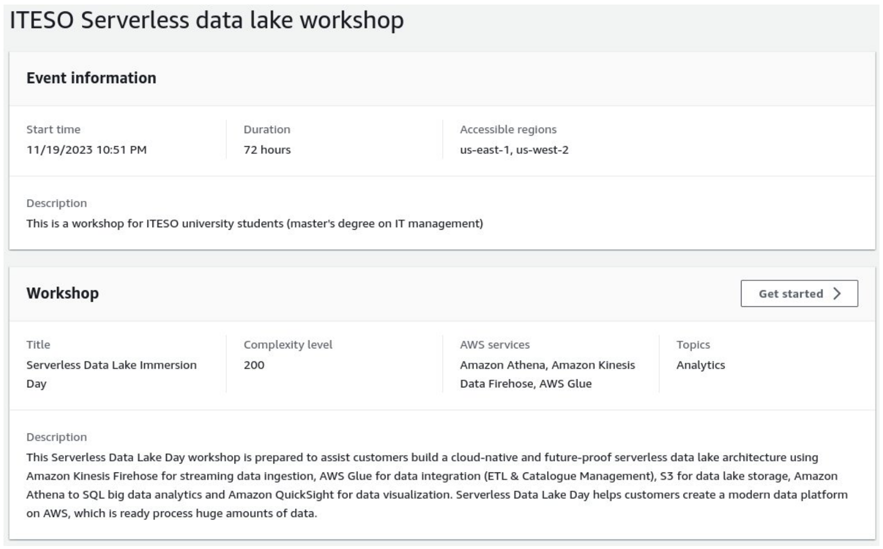
Practice Development

Introduction and prerequisites

Before digging into formal development, some considerations have to be taken, these primarily being AWS IAM [4] usage for IAM roles or OTP access. This step is basic, but if it isn't done properly, none of the following can be done.

Exploration

Similarly to a previously showed practice, the way of completing the current development is through an AWS hosted event. The details of accessing the *Serverless Data Lake Day* are spared. Once the workshop has been opened, a familiar panel will greet the user, stating the purpose of its contents.


ITESO Serverless data lake workshop

Event information

Start time	Duration	Accessible regions
11/19/2023 10:51 PM	72 hours	us-east-1, us-west-2

Description
This is a workshop for ITESO university students (master's degree on IT management)

Workshop

Title	Complexity level	AWS services	Topics
Serverless Data Lake Immersion Day	200	Amazon Athena, Amazon Kinesis Data Firehose, AWS Glue	Analytics

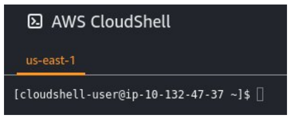
Description
This Serverless Data Lake Day workshop is prepared to assist customers build a cloud-native and future-proof serverless data lake architecture using Amazon Kinesis Firehose for streaming data ingestion, AWS Glue for data integration (ETL & Catalogue Management), S3 for data lake storage, Amazon Athena to SQL big data analytics and Amazon QuickSight for data visualization. Serverless Data Lake Day helps customers create a modern data platform on AWS, which is ready process huge amounts of data.

Several services can be spotted, such as Athena [5], Kinesis Data Firehose [6], and Glue [7]. Each of these will be used in next sections.

Lab1: Data Ingestion & Storage

In this first part of the workshop, the user is presented with two options that will ultimately lead to the seam results: Batch Ingestion, and Real Time Streaming Data Ingestion. The main difference is Kinesis usage, with the first option lacking it altogether. For time constraint purposes, this is preferred, as the implementation is pretty straightforward.

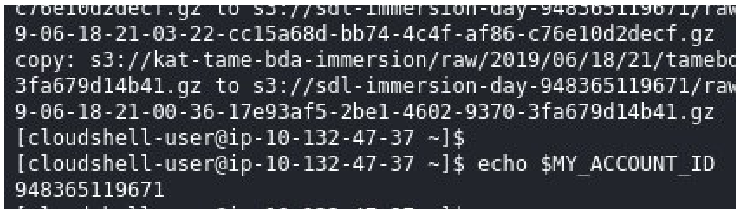
The first step to this will be accessing CloudShell [8], this can be done inside AWS' console (located at the bottom left of the workshop, inside the panel). **Note**: Before doing so, change the region of the console from default value Oregon (us-west-2) to North Virginia (us-west-1), otherwise, errors will be prompt in subsequent steps.



```
[cloudshell-user@ip-10-132-47-37 ~]$
```

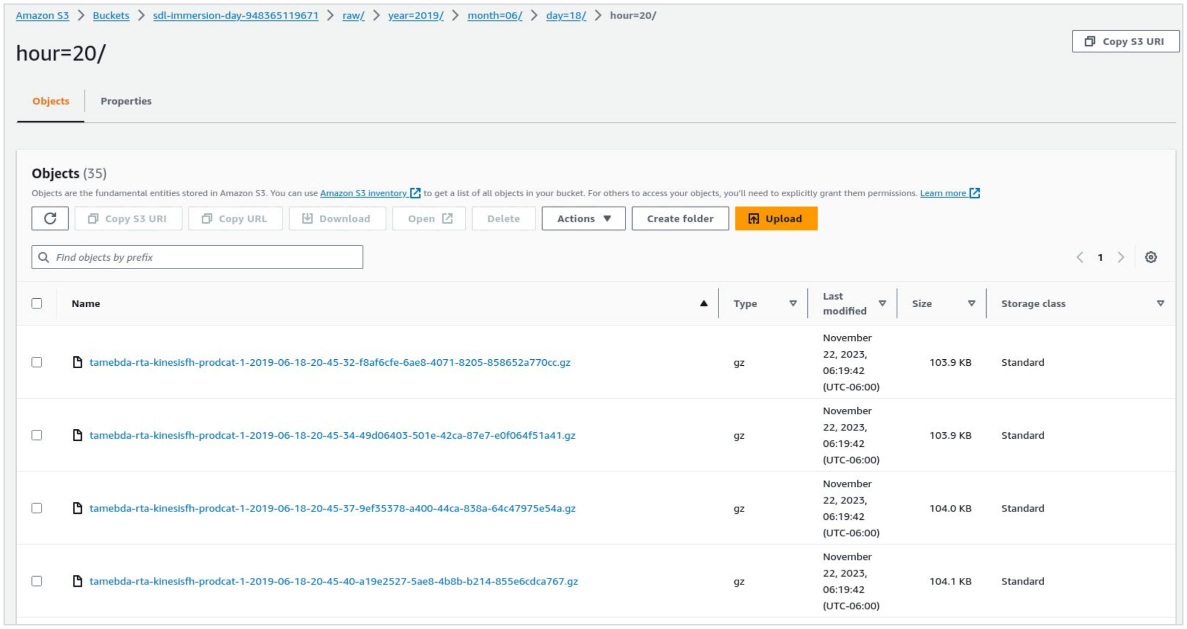
AWS provides the following commands, which should be entered inside the console:

```
export MY_ACCOUNT_ID="$(aws sts get-caller-identity --query Account --output text)"

aws s3 cp s3://kat-tame-bda-immersion/raw/2019/06/18/20/ s3://sdl-immersion-day-
$MY_ACCOUNT_ID/raw/year=2019/month=06/day=18/hour=20/ --recursive
aws s3 cp s3://kat-tame-bda-immersion/raw/2019/06/18/21/ s3://sdl-immersion-day-
$MY_ACCOUNT_ID/raw/year=2019/month=06/day=18/hour=21/ --recursive
```



```
9-06-18-21-03-22-cc15a68d-bb74-4c4f-af86-c76e10d2decf.gz
copy: s3://kat-tame-bda-immersion/raw/2019/06/18/21/tameb
3fa679d14b41.gz to s3://sdl-immersion-day-948365119671/raw
9-06-18-21-00-36-17e93af5-2be1-4602-9370-3fa679d14b41.gz
[cloudshell-user@ip-10-132-47-37 ~]$
[cloudshell-user@ip-10-132-47-37 ~]$ echo $MY_ACCOUNT_ID
948365119671
```

Multiple output lines will be displayed, and although it might seem cryptic, the presence of all these means that all has been set correctly. In fact, to verify this, search for the S3 service. A single bucket should be displayed with its contents being several directories with .gz files inside.



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

Name	Type	Last modified	Size	Storage class
tamebda-rta-kinesisfh-prodcat-1-2019-06-18-20-45-32-f8af6cfe-6ae8-4071-8205-858652a770cc.gz	gz	November 22, 2023, 06:19:42 (UTC-06:00)	103.9 KB	Standard
tamebda-rta-kinesisfh-prodcat-1-2019-06-18-20-45-34-49d06403-501e-42ca-87e7-e0f064f51a41.gz	gz	November 22, 2023, 06:19:42 (UTC-06:00)	103.9 KB	Standard
tamebda-rta-kinesisfh-prodcat-1-2019-06-18-20-45-37-9ef35378-a400-44ca-838a-64c47975e54a.gz	gz	November 22, 2023, 06:19:42 (UTC-06:00)	104.0 KB	Standard
tamebda-rta-kinesisfh-prodcat-1-2019-06-18-20-45-40-a19e2527-5ae8-4b8b-b214-855e6cdca767.gz	gz	November 22, 2023, 06:19:42 (UTC-06:00)	104.1 KB	Standard

This verification marks the end of the workshop's first part. Easy, right? At least for now. After this, CloudShell can be closed.

Lab2: Data Cataloging and ETL

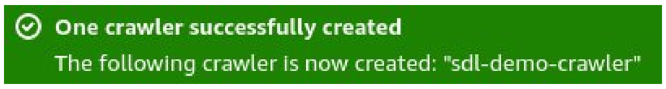
As stated before, ETL stands for Extract, Transform, and Load; this term is often used in data streams where raw information is emitted and modifications are needed before passing it to third parties or another even another ETL process. There also exists the ELT (same terms, different order of execution) variant, in which transformation is made within the same workflow. In the current section, as it names suggest, previously imported data will be cataloged and processed.

The first step is to create a *crawler* for the S3 bucket containing all data. This is done through the Glue service's console. **Note**: several access to the AWS console will probably change it's region, just ensure that all actions made inside it are done trough us-east-1.

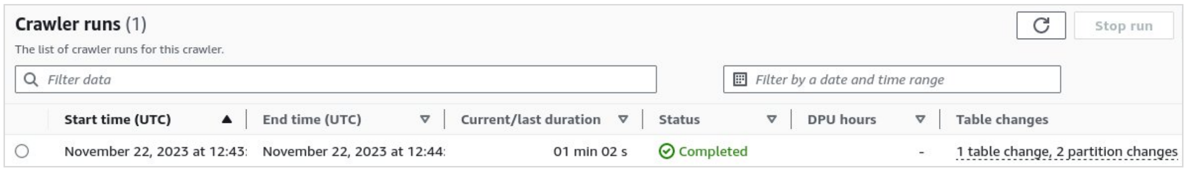
The specifications for this new crawler are the following:

- Name: sdl-demo-crawler
- Default data source configuration
    - Add a data source of type S3
        - Browse for the immersion day bucket and its *raw* directory
        - Select *Crawl all sub-folders* option
- Security settings
    - Select existing *SDL-GlueRole* IAM role
- Output and scheduling settings
    - Add database (this will take to another tab)
        - Name: sdl-demo-data
    - Refresh and select the created database

After all configurations are made, the crawler is created.

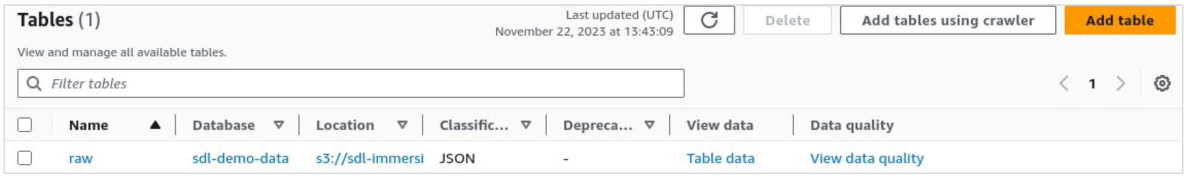

One crawler successfully created
The following crawler is now created: "sdl-demo-crawler"

At this point, it's appropriate to mention that crawlers are components inside a data workflow that classify said data schema and then create metadata tables inside the same container catalog. A crawler run is the application of this concept.


Crawler runs (1)
The list of crawler runs for this crawler.
Stop run
Filter data
Filter by a date and time range

Start time (UTC)	End time (UTC)	Current/last duration	Status	DPU hours	Table changes
November 22, 2023 at 12:43:	November 22, 2023 at 12:44:	01 min 02 s	Completed	-	1 table change, 2 partition changes

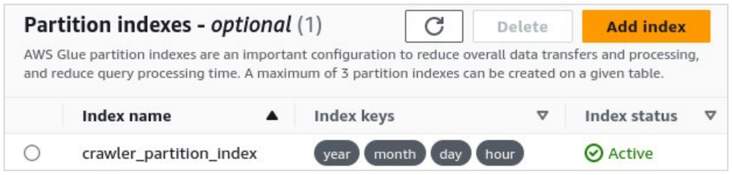
Once the crawl it's complete, on the same service panel, the tables created from the crawl can be seen.



Name	Database	Location	Classific...	Depreca...	View data	Data quality
raw	sdl-demo-data	s3://sdl-immersi	JSON	-	Table data	View data quality

The next step is to modify this created table schema, as the crawler might've made some sort of mistake. First, locate the bottom of the table overview section an delete the existing partition index.

Once the previous indexes were modified, new ones will be created. To do so, search for *Actions* button and then *Edit schema* action. Select individually all rows from *year* to *hour* and assign partition indexes from 10 to 13 respectively, then save table as new version. After that, create a new one with an arbitrary name or the previous one (*crawler_partition_index*).



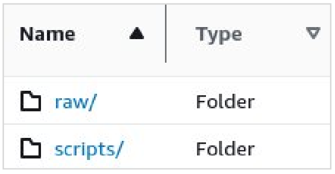
Partition indexes - *optional* (1)

AWS Glue partition indexes are an important configuration to reduce overall data transfers and processing, and reduce query processing time. A maximum of 3 partition indexes can be created on a given table.

Index name	Index keys	Index status
crawler_partition_index	year month day hour	Active

Each change done to the table creates a new version, so, if several changes were made, several new versions will be listed inside the version tab.

The next big step in this section will be the creation of a script directory inside the same S3 bucket, this will mimic a central repository for ETL. Having open the Glue service, open a new tab with the S3 service listing, then, select the bucket and create a new *scripts* directory inside it.

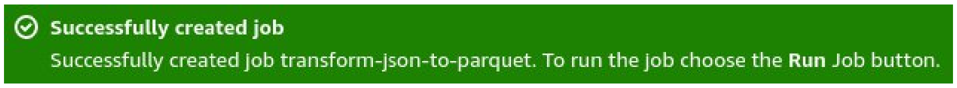


Name	Type
raw/	Folder
scripts/	Folder

Now, in the open *Glue* tab, search for *ETL Jobs* in the side panel, where *Glue Studio* can be found. Once inside, create a new job through the visual ETL option. After this selection, a graphical interface will be displayed. Before digging deeper into this, switch to the job details tab and input the following parameters:

- Name: transform-json-to-parquet
- IAM Role: SDL-GlueRole
- Glue version: Glue 3.0
- Language: Python 3
- Worker type: G 1X
- Advanced properties
    - Script filename: transform-json-to-parquet-py
    - Script path: Previously set script directory inside S3 path

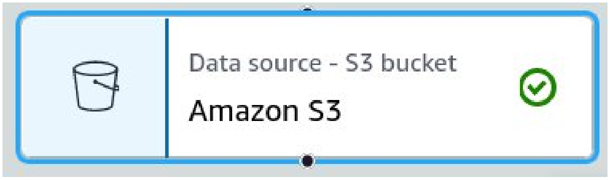
Leave everything else as default and then **save** the details.



Successfully created job

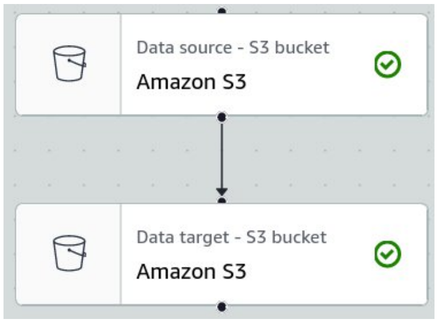
Successfully created job transform-json-to-parquet. To run the job choose the **Run** Job button.

After this, return to visual tab to configure source & targets for the ETL transformation. Click on the add symbol located at the top left corner and select the sources tab, then, select S3. A new visual component inside the canvas will be shown; click it to bring up the data source properties. Inside this section, switch from default "S3 location" option to "Data Catalog table" and select the previously created Glue database (sdl-demo-data) with corresponding table (raw). This source status should now be in an *okay* status.

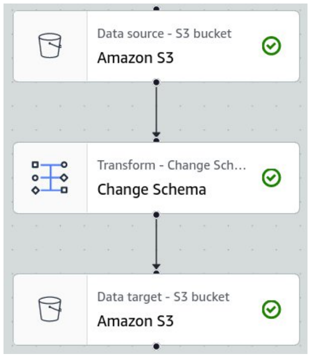


Add another S3 node but now from the targets tab. Both nodes should be connected automatically. Configure it with the follow:

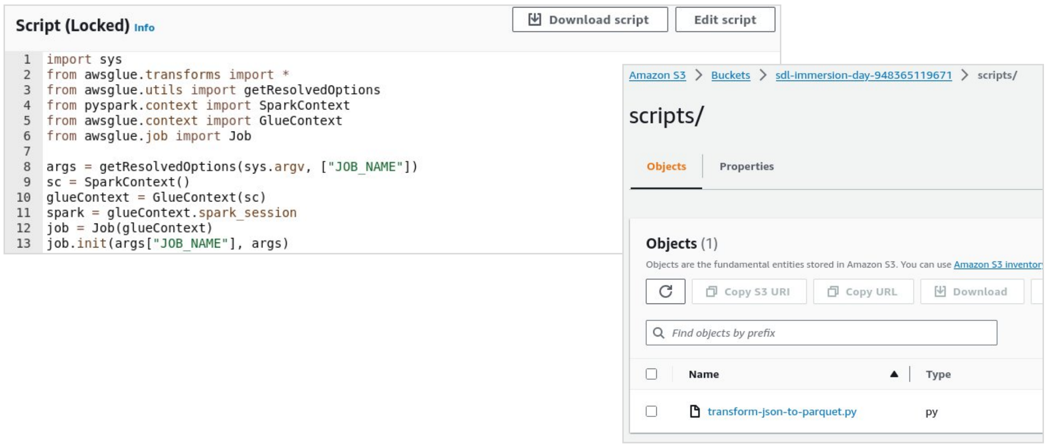
- Format: Parquet
- Compression type: Snappy
- Browse S3 and choose datalake bucket (immersion day)
- S3 location target: add /compressed-parquet/ to the bucket name (this will create a new directory)



Next, select transforms tab and then “Change Schema” to apply a new mapping. In “Node parents” box, select “Amazon S3” option. Mark *color* field to drop and rename date fields to *date_start* and *date_until* (this may lag the panel a little bit). After that, select the target node, deselect the S3 data source and swap it to change schema, it’ll link the recently configured node to the previous ones.



Once all this is done, **save** the job again. The job’s generated script can be seen in the script tab or directly inside the S3 scripts directory.



```
import sys
from awsglue.transforms import *
from awsglue.utils import getResolvedOptions
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.job import Job

args = getResolvedOptions(sys.argv, ["JOB_NAME"])
sc = SparkContext()
glueContext = GlueContext(sc)
spark = glueContext.spark_session
job = Job(glueContext)
job.init(args["JOB_NAME"], args)
```

Amazon S3 > Buckets > sdl-immersion-day-948365119671 > scripts/

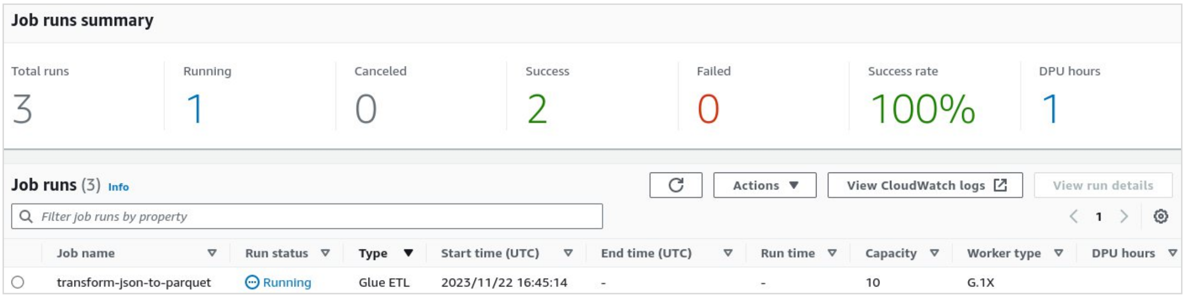
scripts/

Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventor

Name	Type
transform-json-to-parquet.py	py

With the job fully configured, it can be run and monitored. To run it, simply press the button from the same section as before. To monitor it, inside Glue Studio, search for “Job run monitoring”. This management console shows details about all jobs runs.

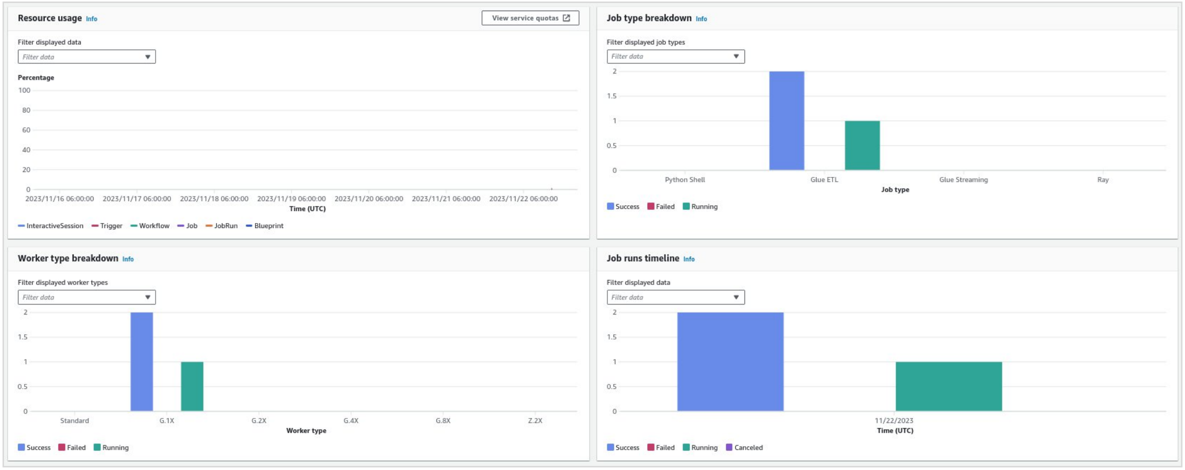

Job runs summary

Total runs	Running	Canceled	Success	Failed	Success rate	DPU hours
3	1	0	2	0	100%	1

Job runs (3) Info

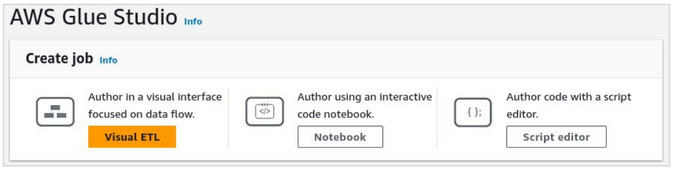
Job name	Run status	Type	Start time (UTC)	End time (UTC)	Run time	Capacity	Worker type	DPU hours
transform-json-to-parquet	Running	Glue ETL	2023/11/22 16:45:14	-	-	10	G.1X	

By selecting an individual run, more specific metrics can be seen at the inferior section of this console.



After each run, the S3 bucket will be updated inside the corresponding parquet path with several parquet files.

The next and final part of this workshop section covers Glue’s interactive sessions through Jupyter notebooks. AWS provides this resource to save the hassle of a new notebook creation. Having the file downloaded from the source, it has to be uploaded as a new job inside the seen Glue Studio section.

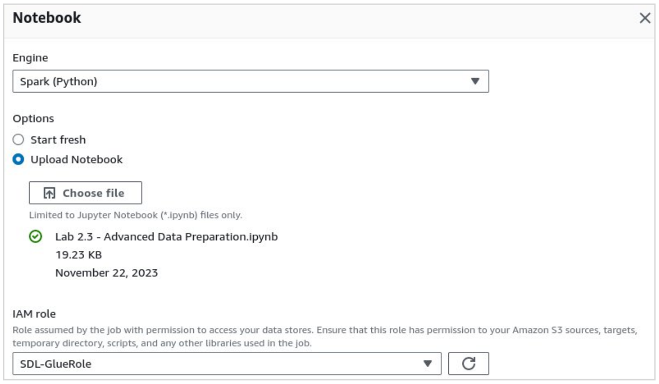

AWS Glue Studio Info

Create job Info

Author in a visual interface focused on data flow. Visual ETL

Author using an interactive code notebook. Notebook

Author code with a script editor. Script editor


Notebook

Engine

Spark (Python)

Options

○ Start fresh

● Upload Notebook

Choose file

Limited to Jupyter Notebook (*.ipynb) files only.

✓ Lab 2.3 - Advanced Data Preparation.ipynb
19.23 KB
November 22, 2023

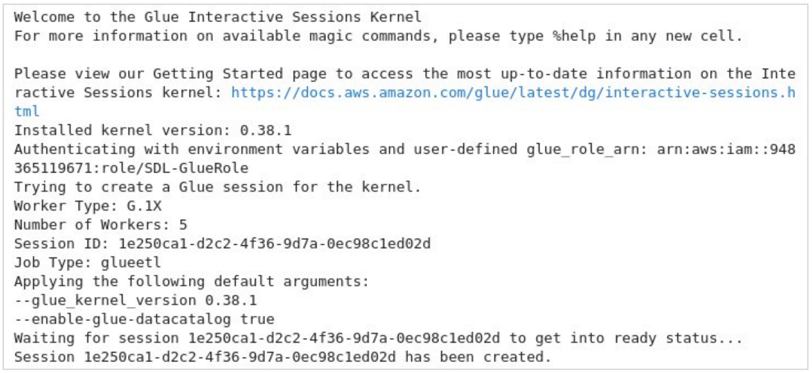
IAM role

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

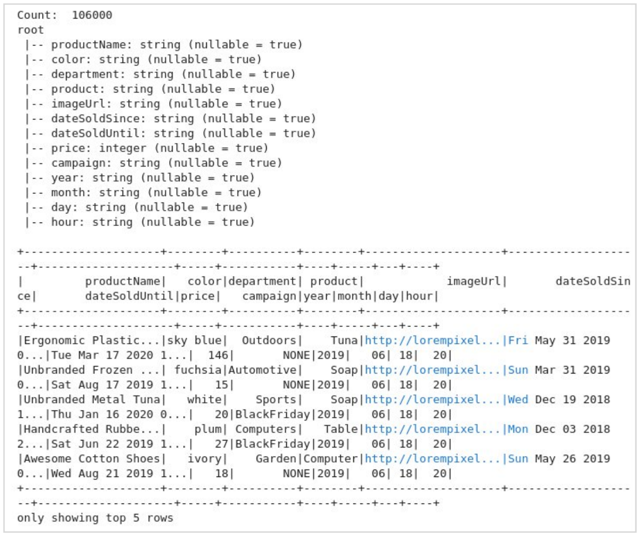
SDL-GlueRole

After some time, an interactive Jupyter environment will be displayed. This workspace takes code snippets and runs them in an enclosed environment. The current file loaded has particular instructions that aid with connection and interfacing with AWS.

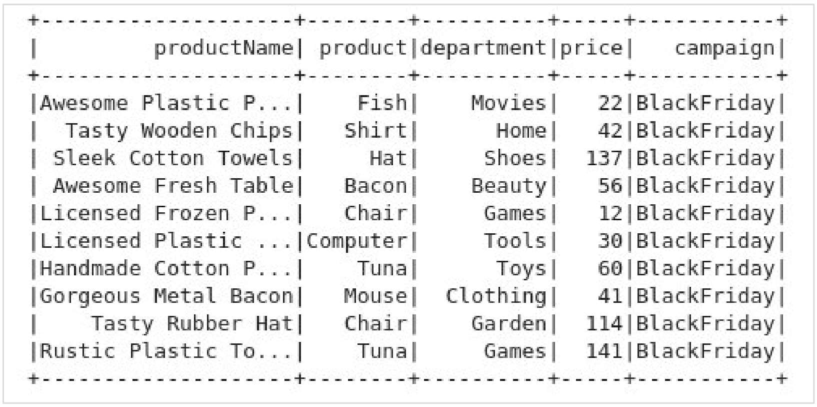
One of the first things prompted to do is changing the account number variable found in code snippet 1. To get this done, click on the top right of AWS' console and copy the Account ID field (there should be a copy button there), then replace the placeholder string without dashes. After the replacement is made, all code cells can be executed normally.



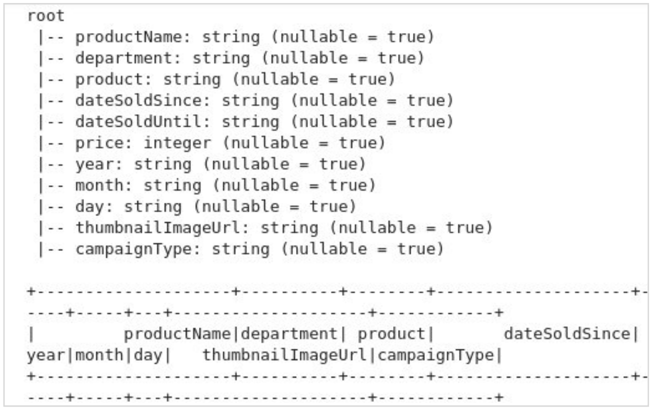
```
Welcome to the Glue Interactive Sessions Kernel
For more information on available magic commands, please type %help in any new cell.

Please view our Getting Started page to access the most up-to-date information on the Inte
ractive Sessions kernel: https://docs.aws.amazon.com/glue/latest/dg/interactive-sessions.h
tml
Installed kernel version: 0.38.1
Authenticating with environment variables and user-defined glue_role_arn: arn:aws:iam::948
365119671:role/SDL-GlueRole
Trying to create a Glue session for the kernel.
Worker Type: G.1X
Number of Workers: 5
Session ID: 1e250ca1-d2c2-4f36-9d7a-0ec98c1ed02d
Job Type: glueetl
Applying the following default arguments:
--glue_kernel_version 0.38.1
--enable-glue-datacatalog true
Waiting for session 1e250ca1-d2c2-4f36-9d7a-0ec98c1ed02d to get into ready status...
Session 1e250ca1-d2c2-4f36-9d7a-0ec98c1ed02d has been created.
```



```
Count:  106000
root
 |-- productName: string (nullable = true)
 |-- color: string (nullable = true)
 |-- department: string (nullable = true)
 |-- product: string (nullable = true)
 |-- imageUrl: string (nullable = true)
 |-- dateSoldSince: string (nullable = true)
 |-- dateSoldUntil: string (nullable = true)
 |-- price: integer (nullable = true)
 |-- campaign: string (nullable = true)
 |-- year: string (nullable = true)
 |-- month: string (nullable = true)
 |-- day: string (nullable = true)
 |-- hour: string (nullable = true)

+--------------------+--------+----------+--------+--------------------+-----------------
--+--------------------+-----+----------+----+-----+---+----+
|         productName|   color|department| product|            imageUrl|       dateSoldSin
ce|       dateSoldUntil|price|   campaign|year|month|day|hour|
+--------------------+--------+----------+--------+--------------------+-----------------
--+--------------------+-----+----------+----+-----+---+----+
|Ergonomic Plastic...|sky blue|  Outdoors|    Tuna|http://lorempixel...|Fri May 31 2019
0...|Tue Mar 17 2020 1...|  146|      NONE|2019|   06| 18|  20|
|Unbranded Frozen ...| fuchsia|Automotive|    Soap|http://lorempixel...|Sun Mar 31 2019
0...|Sat Aug 17 2019 1...|   15|      NONE|2019|   06| 18|  20|
|Unbranded Metal Tuna|   white|    Sports|    Soap|http://lorempixel...|Wed Dec 19 2018
1...|Thu Jan 16 2020 0...|   20|BlackFriday|2019|   06| 18|  20|
|Handcrafted Rubbe...|    plum| Computers|   Table|http://lorempixel...|Mon Dec 03 2018
2...|Sat Jun 22 2019 1...|   27|BlackFriday|2019|   06| 18|  20|
|Awesome Cotton Shoes|   ivory|    Garden|Computer|http://lorempixel...|Sun May 26 2019
0...|Wed Aug 21 2019 1...|   18|      NONE|2019|   06| 18|  20|
+--------------------+--------+----------+--------+--------------------+-----------------
--+--------------------+-----+----------+----+-----+---+----+
only showing top 5 rows
```



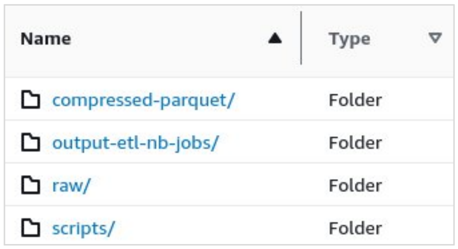
```
+--------------------+--------+----------+-----+-----------+
|         productName| product|department|price|   campaign|
+--------------------+--------+----------+-----+-----------+
|Awesome Plastic P...|    Fish|    Movies|   22|BlackFriday|
|  Tasty Wooden Chips|   Shirt|      Home|   42|BlackFriday|
| Sleek Cotton Towels|     Hat|     Shoes|  137|BlackFriday|
| Awesome Fresh Table|   Bacon|    Beauty|   56|BlackFriday|
|Licensed Frozen P...|   Chair|     Games|   12|BlackFriday|
|Licensed Plastic ...|Computer|     Tools|   30|BlackFriday|
|Handmade Cotton P...|    Tuna|      Toys|   60|BlackFriday|
|Gorgeous Metal Bacon|   Mouse|  Clothing|   41|BlackFriday|
|    Tasty Rubber Hat|   Chair|    Garden|  114|BlackFriday|
|Rustic Plastic To...|    Tuna|     Games|  141|BlackFriday|
+--------------------+--------+----------+-----+-----------+
```



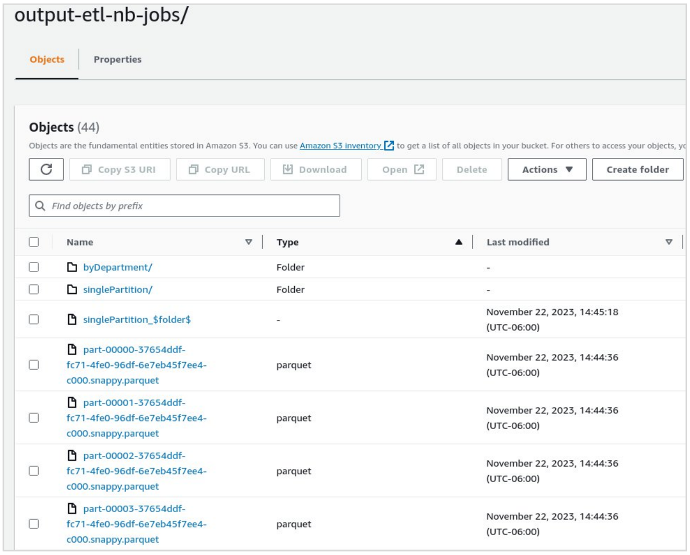
```
root
 |-- productName: string (nullable = true)
 |-- department: string (nullable = true)
 |-- product: string (nullable = true)
 |-- dateSoldSince: string (nullable = true)
 |-- dateSoldUntil: string (nullable = true)
 |-- price: integer (nullable = true)
 |-- year: string (nullable = true)
 |-- month: string (nullable = true)
 |-- day: string (nullable = true)
 |-- thumbnailImageUrl: string (nullable = true)
 |-- campaignType: string (nullable = true)

+--------------------+----------+--------+--------------------+-
----+-----+---+--------------------+------------+
|         productName|department| product|       dateSoldSince|
year|month|day|   thumbnailImageUrl|campaignType|
+--------------------+----------+--------+--------------------+-
----+-----+---+--------------------+------------+
```

```
<awsglue.dynamicframe.DynamicFrame object at 0x7f69c9f4bc50>
```

The previous code run should leave the original S3 bucket with an additional directory called "output-etl-nb-jobs" with several contents inside.



Name	Type
compressed-parquet/	Folder
output-etl-nb-jobs/	Folder
raw/	Folder
scripts/	Folder



output-etl-nb-jobs/

Objects (44)

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, yo

Name	Type	Last modified
byDepartment/	Folder	-
singlePartition/	Folder	-
singlePartition_$folder$	-	November 22, 2023, 14:45:18 (UTC-06:00)
part-00000-37654ddf-fc71-4fe0-96df-6e7eb45f7ee4-c000.snappy.parquet	parquet	November 22, 2023, 14:44:36 (UTC-06:00)
part-00001-37654ddf-fc71-4fe0-96df-6e7eb45f7ee4-c000.snappy.parquet	parquet	November 22, 2023, 14:44:36 (UTC-06:00)
part-00002-37654ddf-fc71-4fe0-96df-6e7eb45f7ee4-c000.snappy.parquet	parquet	November 22, 2023, 14:44:36 (UTC-06:00)
part-00003-37654ddf-fc71-4fe0-96df-6e7eb45f7ee4-c000.snappy.parquet	parquet	November 22, 2023, 14:44:36 (UTC-06:00)

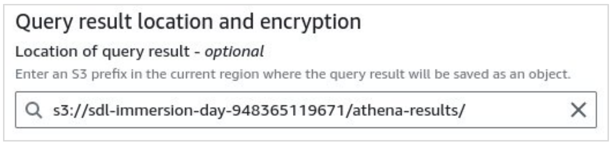
Although an statement for section completion has already been issued, an additional exercise involving ETL automation through triggers is proposed. For time constraints, this demonstration will be skipped, but its core concept is that of creating a job that will be executed once a day to handle specified data through a ruleset.

This marks the end of the second section of the workshop.

Lab3: Data Analytics & Visualization

The final section of this workshops will handle bigger datasets than those seen before and will show them graphically in order to provide a deeper sense of understanding and an alternative way of presenting results.

For the first part, Athena service will be used. Inside it's own console section, search for "Launch query editor" button or "Query editor". Once inside, head for the settings tab and then click on manage to set query result location. Search for the immersion day S3 bucket and add /athena-results/ at the end after selecting it.



Query result location and encryption

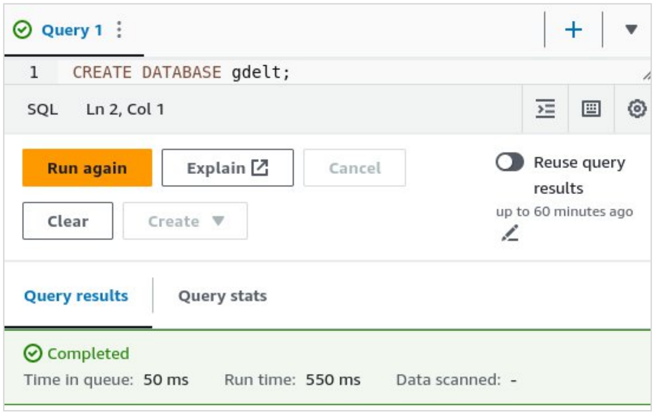
Location of query result - *optional*

Enter an S3 prefix in the current region where the query result will be saved as an object.

s3://sdl-immersion-day-948365119671/athena-results/

After that, locate the query editor box and input the following SQL statement:

```
CREATE DATABASE gdelt;
```

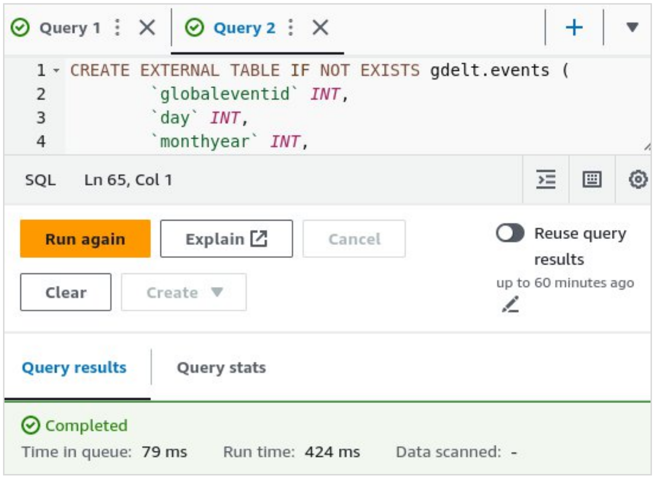


```
CREATE DATABASE gdelt;
```

SQL Ln 2, Col 1

Completed
Time in queue: 50 ms Run time: 550 ms Data scanned: -

After that, a long creation query (omitted for brevity reasons) will also be executed.



```
CREATE EXTERNAL TABLE IF NOT EXISTS gdelt.events (
        `globaleventid` INT,
        `day` INT,
        `monthyear` INT,
```

SQL Ln 65, Col 1

Completed
Time in queue: 79 ms Run time: 424 ms Data scanned: -

After the query completes successfully, four (1) (2) (3) (4) files have to been locally downloaded and then uploaded to the same S3 bucket inside a new directory named gdelt containing 4 folders with names *countries*, *eventcodes*, *groups* and *types*.

Once that is done, run the following SQL statements, one for each directory:

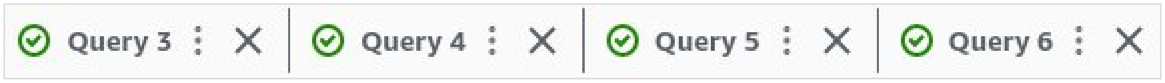
```
CREATE EXTERNAL TABLE IF NOT EXISTS gdelt.eventcodes (
       `code` string,
       `description` string
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe'
WITH SERDEPROPERTIES (
       'serialization.format' = '\t','field.delim' = '\t')
LOCATION 's3://<YOUR_BUCKET_NAME>/gdelt/eventcodes/'
 TBLPROPERTIES ( "skip.header.line.count"="1")
```

```
CREATE EXTERNAL TABLE IF NOT EXISTS gdelt.types (
       `type` string,
       `description` string
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe'
WITH SERDEPROPERTIES (
       'serialization.format' = '\t','field.delim' = '\t')
LOCATION 's3://<YOUR_BUCKET_NAME>/gdelt/types/'
TBLPROPERTIES ( "skip.header.line.count"="1");
```

```
CREATE EXTERNAL TABLE IF NOT EXISTS gdelt.groups (
       `group` string,
       `description` string
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe'
WITH SERDEPROPERTIES (
       'serialization.format' = '\t','field.delim' = '\t')
LOCATION 's3://<YOUR_BUCKET_NAME>/gdelt/groups/'
TBLPROPERTIES ( "skip.header.line.count"="1");
```

```
CREATE EXTERNAL TABLE IF NOT EXISTS gdelt.countries (
       `code` string,
       `country` string
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe'
WITH SERDEPROPERTIES (
       'serialization.format' = '\t','field.delim' = '\t')
LOCATION 's3://<YOUR_BUCKET_NAME>/gdelt/countries/'
TBLPROPERTIES ( "skip.header.line.count"="1");
```

Evidently, replace the <YOUR_BUCKET_NAME> placeholder with the immersion day bucket name.

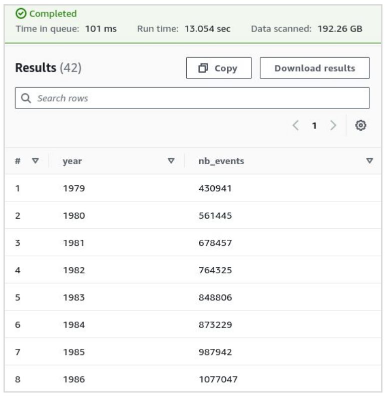


After each table creation has been made, run the next query to verify results (it should return number of events per year):

```
SELECT year,
    COUNT(globaleventid) AS nb_events
FROM gdelt.events
GROUP BY year
ORDER BY year ASC;
```

As seen on the left, results are being displayed after some processing time, this means that previous queries were correct and successful.

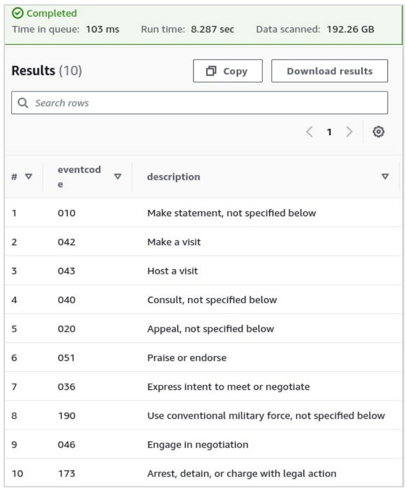
The following parts suggest more queries, which will be omitted, but results are shown below.



Completed
Time in queue: 101 ms Run time: 13.054 sec Data scanned: 192.26 GB

Results (42)

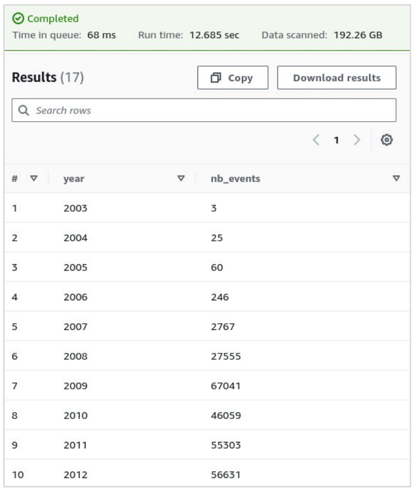
#	year	nb_events
1	1979	430941
2	1980	561445
3	1981	678457
4	1982	764325
5	1983	848806
6	1984	873229
7	1985	987942
8	1986	1077047



Completed
Time in queue: 103 ms Run time: 8.287 sec Data scanned: 192.26 GB

Results (10)

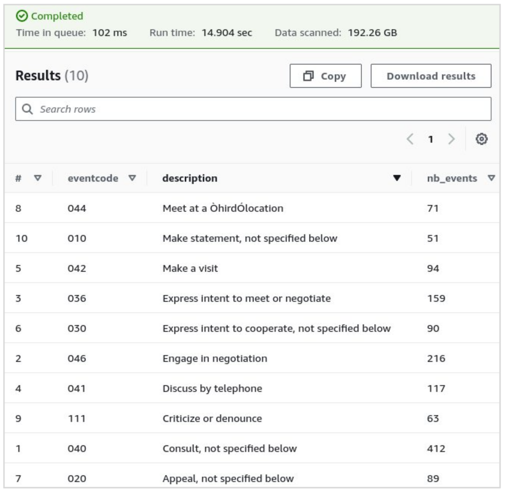
#	eventcode	description
1	010	Make statement, not specified below
2	042	Make a visit
3	043	Host a visit
4	040	Consult, not specified below
5	020	Appeal, not specified below
6	051	Praise or endorse
7	036	Express intent to meet or negotiate
8	190	Use conventional military force, not specified below
9	046	Engage in negotiation
10	173	Arrest, detain, or charge with legal action



Completed
Time in queue: 68 ms Run time: 12.685 sec Data scanned: 192.26 GB

Results (17)

#	year	nb_events
1	2003	3
2	2004	25
3	2005	60
4	2006	246
5	2007	2767
6	2008	27555
7	2009	67041
8	2010	46059
9	2011	55303
10	2012	56631

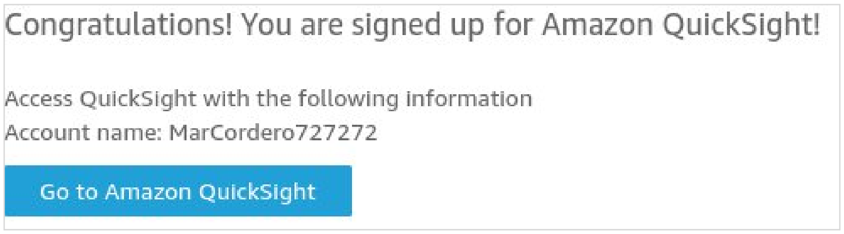


Completed
Time in queue: 102 ms Run time: 14.904 sec Data scanned: 192.26 GB

Results (10)

#	eventcode	description	nb_events
8	044	Meet at a ÒhirdÓlocation	71
10	010	Make statement, not specified below	51
5	042	Make a visit	94
3	036	Express intent to meet or negotiate	159
6	030	Express intent to cooperate, not specified below	90
2	046	Engage in negotiation	216
4	041	Discuss by telephone	117
9	111	Criticize or denounce	63
1	040	Consult, not specified below	412
7	020	Appeal, not specified below	89

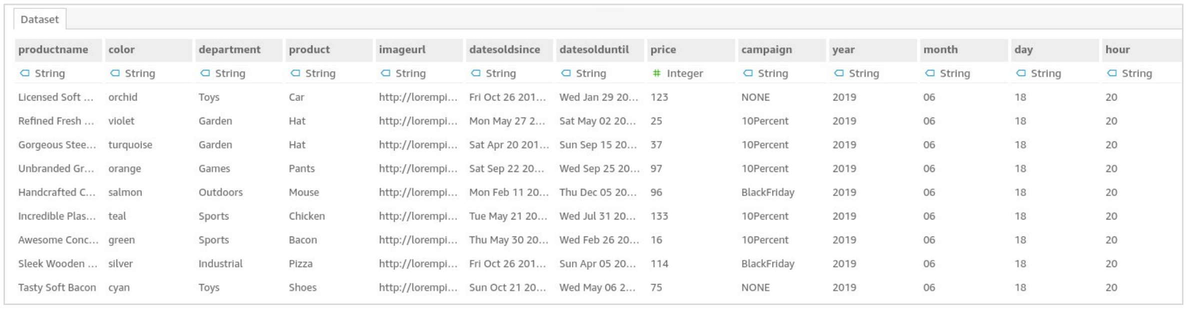
The next part involves QuickSight [9], another service for business intelligence (BI). When accessing it's panel for the first time, the user will be prompted with a sign up screen, as it requires an external account. Leave the default enterprise option selected and continue, skip the paginated add-on ad, and leave the rest as default except for account information, in which unique values are needed. Below account info, select Amazon Athena if it's not selected and then Amazon S3, a new window will be prompted, select the immersion day bucket; in the alternate tab inside the same popup window, select the different bucket option and input gdelt-open-data as the name. After this, two buckets should be selected. The account creation may take some time.



Congratulations! You are signed up for Amazon QuickSight!

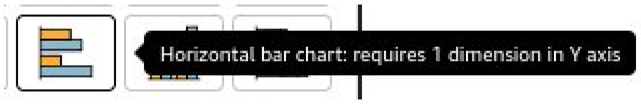
Access QuickSight with the following information
Account name: MarCordero727272

Having the newly created account, head to Amazon QuickSight and search for Data Sets on the left menu and then create a new one. The next screen will contain several data sources, select Athena. Give it a symbolic name and then create it. If everything was done correctly, the next screen should contain the previously created *sdl-demo-data* inside table sets, which has to be selected. Click on preview data to verify the table contents.



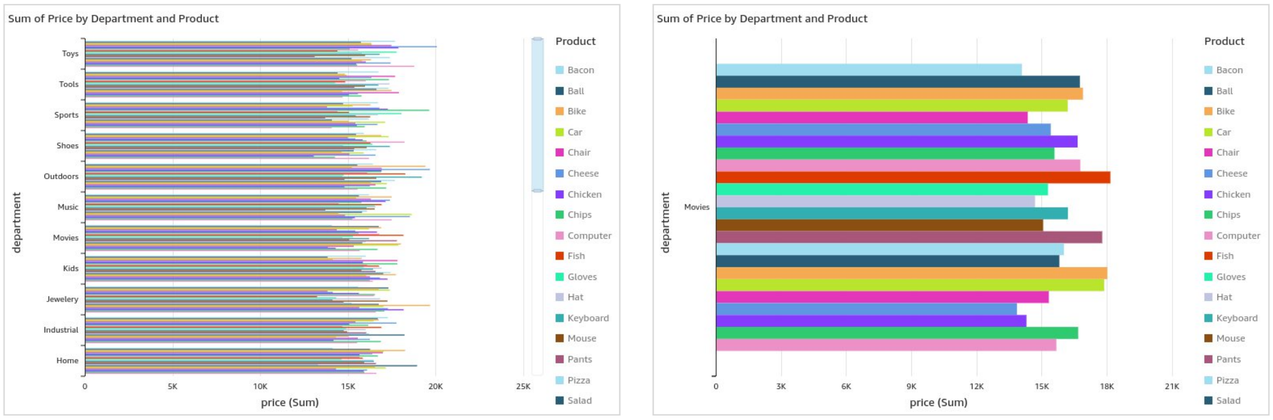
productname	color	department	product	imageurl	datesoldsince	datesolduntil	price	campaign	year	month	day	hour
String	String	String	String	String	String	String	# Integer	String	String	String	String	String
Licensed Soft ...	orchid	Toys	Car	http://lorempi...	Fri Oct 26 201...	Wed Jan 29 20...	123	NONE	2019	06	18	20
Refined Fresh ...	violet	Garden	Hat	http://lorempi...	Mon May 27 2...	Sat May 02 20...	25	10Percent	2019	06	18	20
Gorgeous Stee...	turquoise	Garden	Hat	http://lorempi...	Sat Apr 20 201...	Sun Sep 15 20...	37	10Percent	2019	06	18	20
Unbranded Gr...	orange	Games	Pants	http://lorempi...	Sat Sep 22 20...	Wed Sep 25 20...	97	10Percent	2019	06	18	20
Handcrafted C...	salmon	Outdoors	Mouse	http://lorempi...	Mon Feb 11 20...	Thu Dec 05 20...	96	BlackFriday	2019	06	18	20
Incredible Plas...	teal	Sports	Chicken	http://lorempi...	Tue May 21 20...	Wed Jul 31 20...	133	10Percent	2019	06	18	20
Awesome Conc...	green	Sports	Bacon	http://lorempi...	Thu May 30 20...	Wed Feb 26 20...	16	10Percent	2019	06	18	20
Sleek Wooden ...	silver	Industrial	Pizza	http://lorempi...	Fri Oct 26 201...	Sun Apr 05 20...	114	BlackFriday	2019	06	18	20
Tasty Soft Bacon	cyan	Toys	Shoes	http://lorempi...	Sun Oct 21 20...	Wed May 06 2...	75	NONE	2019	06	18	20

After everything has been verified, click on publish &visualize. An automatic redirection will be made to the new sheet screen. To demonstrate the current service, create an interactive sheet and search for the *horizontal bar chart* widget.

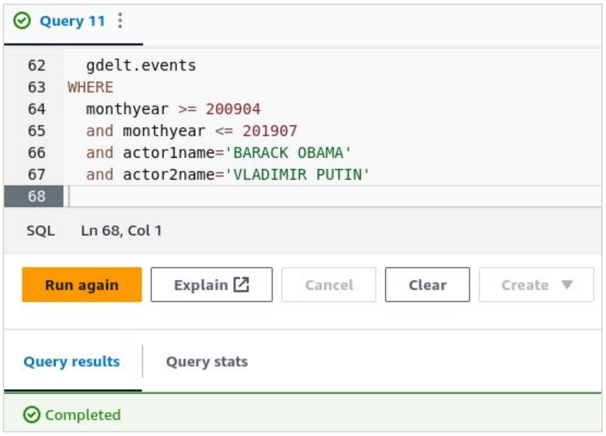


Horizontal bar chart: requires 1 dimension in Y axis

Set *department* as the column inside Y-Axis box, *price* in the Value box, and *product* in the Group/Color box. At first, multiple diminute colored lines will be displayed, but, as with any other dashboard, drill-down features are present to gaze upon better details of any selection.



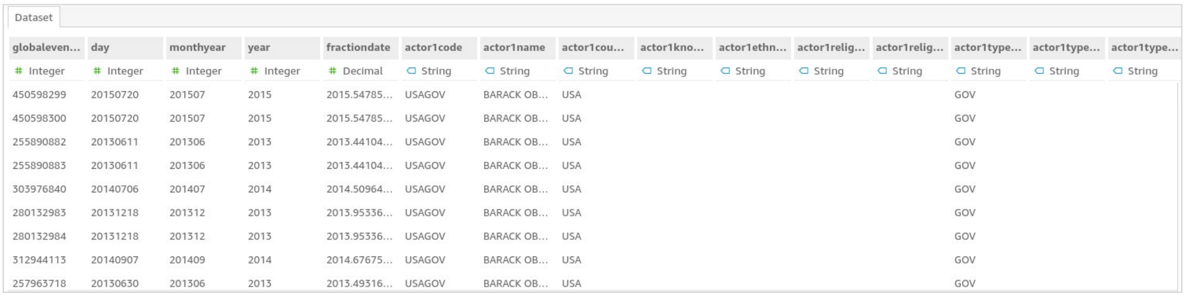
The next section will also use QuickSight, but now, it'll be used for S3 data visualization and dataset joining abilities. Inside the Athena service, the previously used *gdelt* table will be used again. Yet another long query will be executed, and yet again, for the same reasons, it's omitted here.



```
  gdelt.events
WHERE
  monthyear >= 200904
  and monthyear <= 201907
  and actor1name='BARACK OBAMA'
  and actor2name='VLADIMIR PUTIN'

```

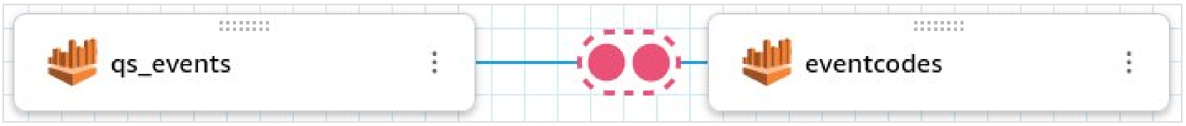
SQL Ln 68, Col 1

Similarly to the previous step, create a new dataset inside QuickSight with *existing* Athena data source and *gdelt* tableset and *qs_events* table. Preview the data.

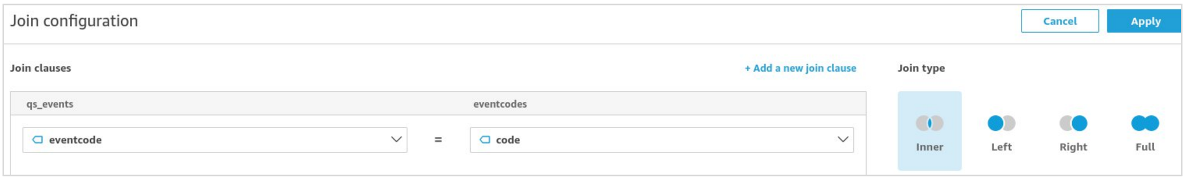


globaleven...	day	monthyear	year	fractiondate	actor1code	actor1name	actor1cou...	actor1kno...	actor1ethn...	actor1relig...	actor1relig...	actor1type...	actor1type...	actor1type...
# Integer	# Integer	# Integer	# Integer	# Decimal	String	String	String	String	String	String	String	String	String	String
450598299	20150720	201507	2015	2015.54785...	USAGOV	BARACK OB...	USA					GOV		
450598300	20150720	201507	2015	2015.54785...	USAGOV	BARACK OB...	USA					GOV		
255890882	20130611	201306	2013	2013.44104...	USAGOV	BARACK OB...	USA					GOV		
255890883	20130611	201306	2013	2013.44104...	USAGOV	BARACK OB...	USA					GOV		
303976840	20140706	201407	2014	2014.50964...	USAGOV	BARACK OB...	USA					GOV		
280132983	20131218	201312	2013	2013.95336...	USAGOV	BARACK OB...	USA					GOV		
280132984	20131218	201312	2013	2013.95336...	USAGOV	BARACK OB...	USA					GOV		
312944113	20140907	201409	2014	2014.67675...	USAGOV	BARACK OB...	USA					GOV		
257963718	20130630	201306	2013	2013.49316...	USAGOV	BARACK OB...	USA					GOV		

After that, locate the *Add data* button on the top right corner and click it: data source > athena > table *eventcodes*. The visual editor now shows two tables with a join function between them.



Apply the following configuration and proceed.



Join configuration

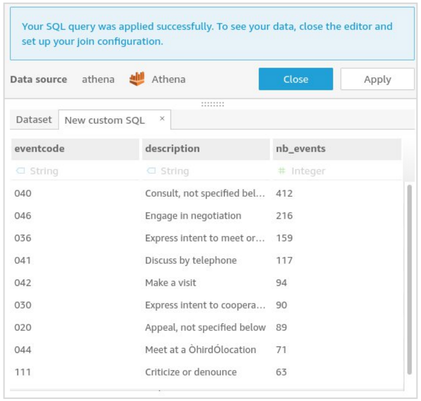
Join clauses

qs_events		eventcodes
eventcode	=	code

Join type: [✓ Inner] Left Right Full

Repeat the same *Add data* process, but instead of manually selecting tables, go for the *Use custom SQL* function. Once inside the new screen, input the following query:

```
-- Count Obama/Putin events per category

SELECT eventcode,
    gdelt.eventcodes.description,
    nb_events
FROM (SELECT gdelt.events.eventcode,
            COUNT(gdelt.events.globaleventid) AS nb_events
    FROM gdelt.events
    WHERE actor1name='BARACK OBAMA' and actor2name='VLADIMIR PUTIN'
    GROUP BY gdelt.events.eventcode
    ORDER BY nb_events DESC)
JOIN gdelt.eventcodes ON eventcode = gdelt.eventcodes.code
WHERE nb_events ≥ 50
ORDER BY nb_events DESC;
```

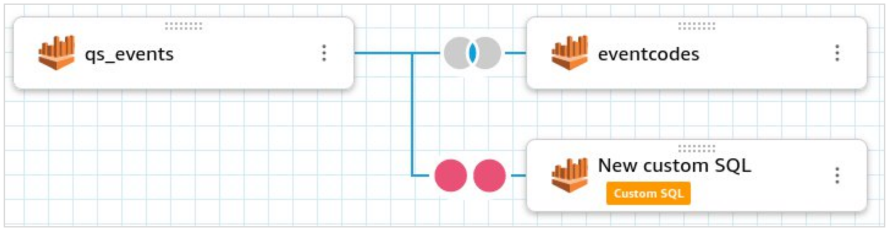


Your SQL query was applied successfully. To see your data, close the editor and set up your join configuration.

Data source athena Athena

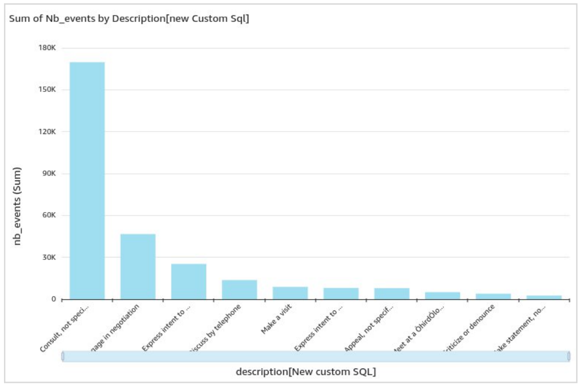
eventcode	description	nb_events
String	String	Integer
040	Consult, not specified bel…	412
046	Engage in negotiation	216
036	Express intent to meet or…	159
041	Discuss by telephone	117
042	Make a visit	94
030	Express intent to coopera…	90
020	Appeal, not specified below	89
044	Meet at a ÒhirdÓlocation	71
111	Criticize or denounce	63

Once the query terminates, close the editor to see a new join operation available.



Set the join with the same configurations as before (use eventcode for both keys). After this, delete the eventcodes dataset, then publish & visualize.

To review these results, follow the same steps as before to create a dashboard view, but now use a vertical bart chart with description as X axis and nb_events for Value.



As in another previous demonstration, the next final part will be purely theoretical, as the actual implementation requires more complex and heavier downloads, altogether with some configuration overkill for Linux.

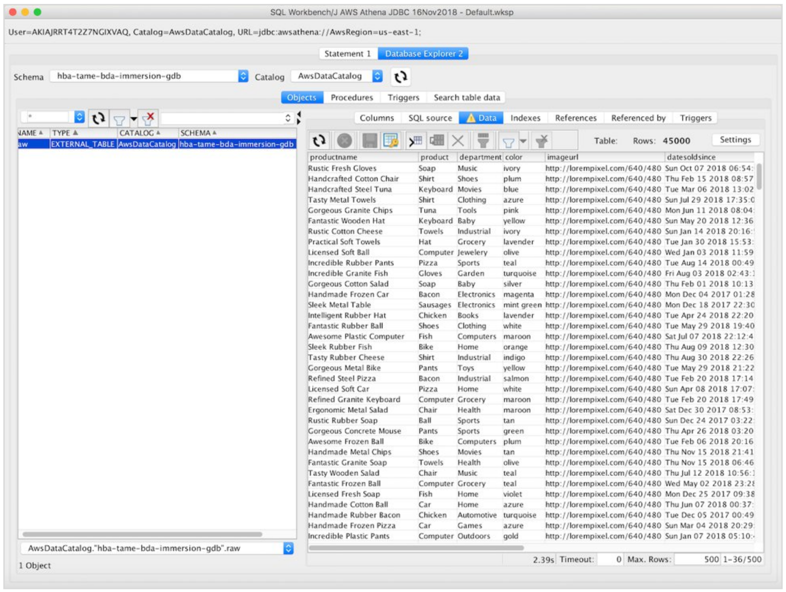
The first supposed step is to create long-lived IAM credentials inside AWS console, something thoroughly advised against by the same AWS, as it supposes a security risk. For this event, this is not necessary, as proper keys are provided.

```
export AWS_ACCESS_KEY_ID="ASIA5ZTXXKC3S2G24M5B"
export AWS_SECRET_ACCESS_KEY="M/dySgNcOZ4myYnN1jwa0ecaaxbkMxmJ5amVjs/b"
export AWS_SESSION_TOKEN="IQoJb3JpZ2luX2VjEN//////////wEaCXVzLWVhc3QtMSJGMEQCIC//ucxQF8DOqAyz9QH
```

Having these keys and SQL Workbench installed, follow these configuration steps:

1. Select File > Manage Drivers
2. Manage Drivers dialog box
    a. Name = Athena JDBC Driver
    b. Version = AthenaJDBC42.jar
    c. Classname = com.simba.athena.jdbc.Driver
    d. Sample URL = jdbc\:awsathena://AwsRegion=us-east-1
    e. Username = AWS_ACCESS_KEY
    f. Password = AWS_SECRET_ACCESS_KEY
3. Extended properties
    a. New property *S3OutputLocation* with value of the URL given for the immersion day S3 bucket and */Athena* at the end

After this, open SQL Workbench and connect through the recently made configuration. After connection it's made, select Database Explorer inside the application, and then select the *sdl-demo-data* schema. Preview the data. This will return S3 stored data.



User=AKIAJRRT4T2Z7NGIXVAQ, Catalog=AwsDataCatalog, URL=jdbc:awsathena://AwsRegion=us-east-1;

Schema: hba-tame-bda-immersion-gdb
Catalog: AwsDataCatalog

NAME	TYPE	CATALOG	SCHEMA
aw	EXTERNAL_TABLE	AwsDataCatalog	hba-tame-bda-immersion-gdb

AwsDataCatalog."hba-tame-bda-immersion-gdb".raw
1 Object

Table: Rows: 45000

productname	product	department	color	imageurl	datesoldsince
Rustic Fresh Gloves	Soap	Music	ivory	http://lorempixel.com/640/480	Sun Oct 07 2018 06:54:
Handcrafted Cotton Chair	Shirt	Shoes	plum	http://lorempixel.com/640/480	Thu Feb 15 2018 08:57
Handcrafted Steel Tuna	Keyboard	Movies	blue	http://lorempixel.com/640/480	Tue Mar 06 2018 13:02
Tasty Metal Towels	Shirt	Clothing	azure	http://lorempixel.com/640/480	Sun Jul 29 2018 17:35:0
Gorgeous Granite Chips	Tuna	Tools	pink	http://lorempixel.com/640/480	Mon Jun 11 2018 08:04:
Fantastic Wooden Hat	Keyboard	Baby	yellow	http://lorempixel.com/640/480	Sun May 20 2018 12:36
Rustic Cotton Cheese	Towels	Industrial	ivory	http://lorempixel.com/640/480	Sun Jan 14 2018 20:16:
Practical Soft Towels	Hat	Grocery	lavender	http://lorempixel.com/640/480	Tue Jan 30 2018 15:53:
Licensed Soft Ball	Computer	Jewelery	olive	http://lorempixel.com/640/480	Wed Jan 03 2018 11:59
Incredible Rubber Pants	Pizza	Sports	teal	http://lorempixel.com/640/480	Tue Aug 14 2018 00:49
Incredible Granite Fish	Gloves	Garden	turquoise	http://lorempixel.com/640/480	Fri Aug 03 2018 02:43:1
Gorgeous Cotton Salad	Soap	Baby	silver	http://lorempixel.com/640/480	Thu Feb 01 2018 10:13
Handmade Frozen Car	Bacon	Electronics	magenta	http://lorempixel.com/640/480	Mon Dec 04 2017 01:28
Sleek Metal Table	Sausages	Electronics	mint green	http://lorempixel.com/640/480	Mon Dec 18 2017 22:30
Intelligent Rubber Hat	Chicken	Books	lavender	http://lorempixel.com/640/480	Tue Apr 24 2018 22:20
Fantastic Rubber Ball	Shoes	Clothing	white	http://lorempixel.com/640/480	Tue May 29 2018 19:40
Awesome Plastic Computer	Fish	Computers	maroon	http://lorempixel.com/640/480	Sat Jul 07 2018 22:12:4
Sleek Rubber Fish	Bike	Home	orange	http://lorempixel.com/640/480	Thu Aug 09 2018 12:30
Tasty Rubber Cheese	Shirt	Industrial	indigo	http://lorempixel.com/640/480	Thu Aug 30 2018 22:26
Gorgeous Metal Bike	Pants	Toys	yellow	http://lorempixel.com/640/480	Tue May 29 2018 21:22
Refined Steel Pizza	Bacon	Industrial	salmon	http://lorempixel.com/640/480	Tue Feb 20 2018 17:14
Licensed Soft Car	Pizza	Home	white	http://lorempixel.com/640/480	Sun Apr 08 2018 17:07:
Refined Granite Keyboard	Computer	Grocery	maroon	http://lorempixel.com/640/480	Tue Feb 20 2018 17:49
Ergonomic Metal Salad	Chair	Health	maroon	http://lorempixel.com/640/480	Sat Dec 30 2017 08:53:
Rustic Rubber Soap	Ball	Sports	tan	http://lorempixel.com/640/480	Sun Dec 24 2017 03:22:
Gorgeous Concrete Mouse	Pants	Sports	green	http://lorempixel.com/640/480	Thu Apr 26 2018 03:20
Awesome Frozen Ball	Bike	Computers	plum	http://lorempixel.com/640/480	Tue Feb 06 2018 20:16
Handmade Metal Chips	Shoes	Movies	tan	http://lorempixel.com/640/480	Thu Nov 15 2018 21:41
Fantastic Granite Soap	Towels	Health	olive	http://lorempixel.com/640/480	Thu Nov 15 2018 06:46
Tasty Wooden Salad	Chair	Music	teal	http://lorempixel.com/640/480	Thu Jul 12 2018 10:56:1
Fantastic Frozen Ball	Computer	Grocery	teal	http://lorempixel.com/640/480	Wed May 02 2018 23:28
Licensed Fresh Soap	Fish	Home	violet	http://lorempixel.com/640/480	Mon Dec 25 2017 09:38
Handmade Cotton Ball	Car	Home	azure	http://lorempixel.com/640/480	Thu Jun 07 2018 00:37:
Handmade Rubber Bacon	Chicken	Automotive	turquoise	http://lorempixel.com/640/480	Tue Dec 05 2017 00:49
Handmade Frozen Pizza	Car	Games	azure	http://lorempixel.com/640/480	Sun Mar 04 2018 20:29:
Incredible Plastic Pants	Computer	Outdoors	gold	http://lorempixel.com/640/480	Sun Jan 07 2018 05:10:

2.39s Timeout: 0 Max. Rows: 500 1-36/500

This marks the end of this section and the workshop. Remember to end every resource created.

Problems and Solutions

No problems, and consequently, no solutions in this practice, as it was a step-by-step well document tutorial.

Experiments and Results

No experiments conducted this time, too archaic.

Budget Justification

Similarly to the diagram provided, another instance of extreme luck is that of cost information provision. This being said, the cost of monthly operating the seen implementation would be the following:

- Kinesis Firehose: < $0.1
- Athena: < $5
- Glue: < $1
- S3: < $0.1
- QuickSight: < $10

Total: ≈ $16.2 USD (as of today)

This of course can be corroborated here.

Conclusions

This exhaustive demonstration was made possible only by the sheer fact that it’s the last one of them all, at least for now. Although this might seem unflattering, rest assured that the knowledge gathered through the shown development it’s definitely impactful and leaves behind a latent curiosity about data transformation.

In past activities, there were an opportunity to process raw data directly from memory latency checkers adapted inside some hardware motherboard, interfaced through a tool called *mlc*. Said process consisted in invoking the command line tool (extracting), placing the results inside Windows directories through dark magic made with bash (still extracting), creating a very case specific parsers with Python 3.8 and (loading) data through them, and then applying some modifications (transform) to finally pass the converted data to another unfortunate person. As it can be seen, this experience followed an ELT pattern implemented through very rudimentary methods. Based in this experience, the final conclusion to be stated is that AWS can be a blessing from time to time, saving tons of hours and countless headaches trying to implement something natively when it can be done in matter of seconds and just a few clicks away.

Bibliography

[1] Aws.amazon.com. 'What is a Data Lake?'. [Online]. Available: https://aws.amazon.com/what-is/data-lake/.

[2] Aws.amazon.com. 'What is ETL (Extract Transform Load)'. [Online]. Available: https://aws.amazon.com/what-is/etl/.

[3] Aws.amazon.com. 'What is business intelligence?'. [Online]. Available: https://www.ibm.com/topics/business-intelligence.

[4] Aws.amazon.com. 'AWS Identity and Access Management'. [Online]. Available: https://aws.amazon.com/iam/.

[5] Aws.amazon.com. 'Amazon Athena'. [Online]. Available: https://aws.amazon.com/athena/.

[6] Aws.amazon.com. 'Amazon Kinesis Data Firehose'. [Online]. Available: https://aws.amazon.com/kinesis/data-firehose/.

[7] Aws.amazon.com. 'AWS Glue'. [Online]. Available: https://aws.amazon.com/glue/.

[8] Aws.amazon.com. 'AWS CloudShell'. [Online]. Available: https://aws.amazon.com/cloudshell/.

[9] Aws.amazon.com. 'Amazon QuickSight'. [Online]. Available: https://aws.amazon.com/quicksight/.