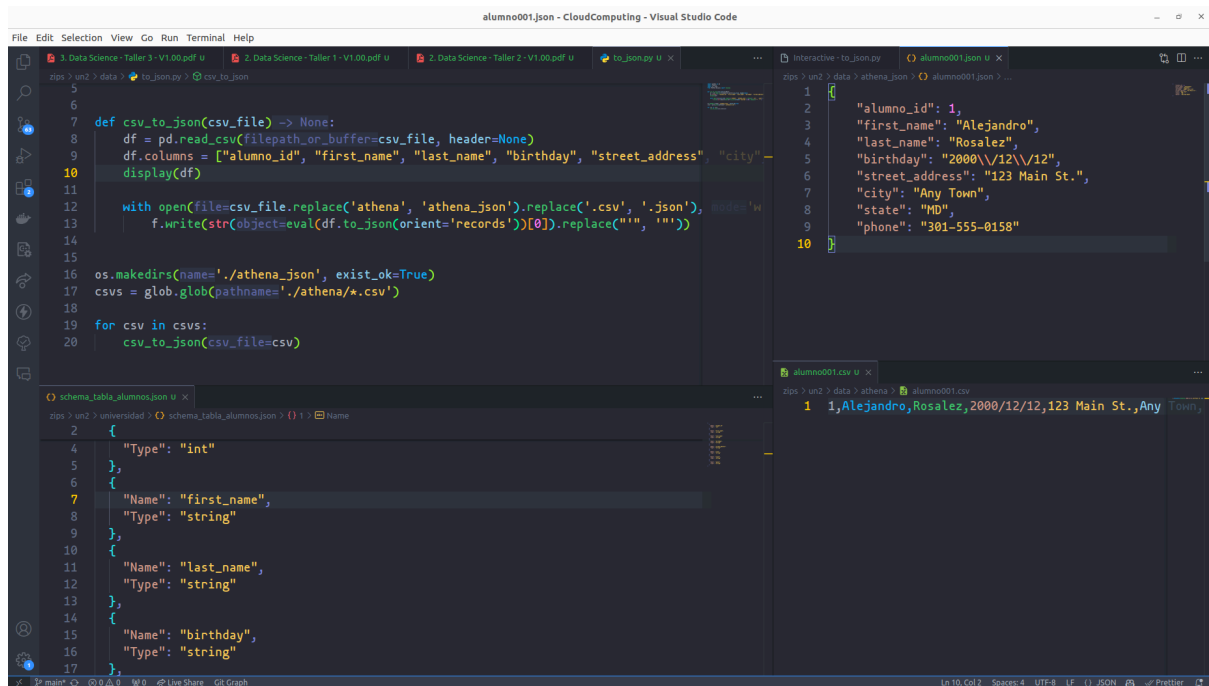


Creamos un script de Python para pasar los archivos .csv a formato .json.



```
def csv_to_json(csv_file) -> None:
    df = pd.read_csv(filepath_or_buffer=csv_file, header=None)
    df.columns = ["alumno_id", "first_name", "last_name", "birthday", "street_address", "city"]
    display(df)

    with open(files=csv_file.replace('athena', 'athena_json').replace('.csv', '.json'), mode='w') as f:
        f.write(str(object=eval(df.to_json(orient='records'))[0]).replace("'", ""))

os.makedirs(name='./athena_json', exist_ok=True)
csvs = glob.glob(pathname='./athena/*.csv')

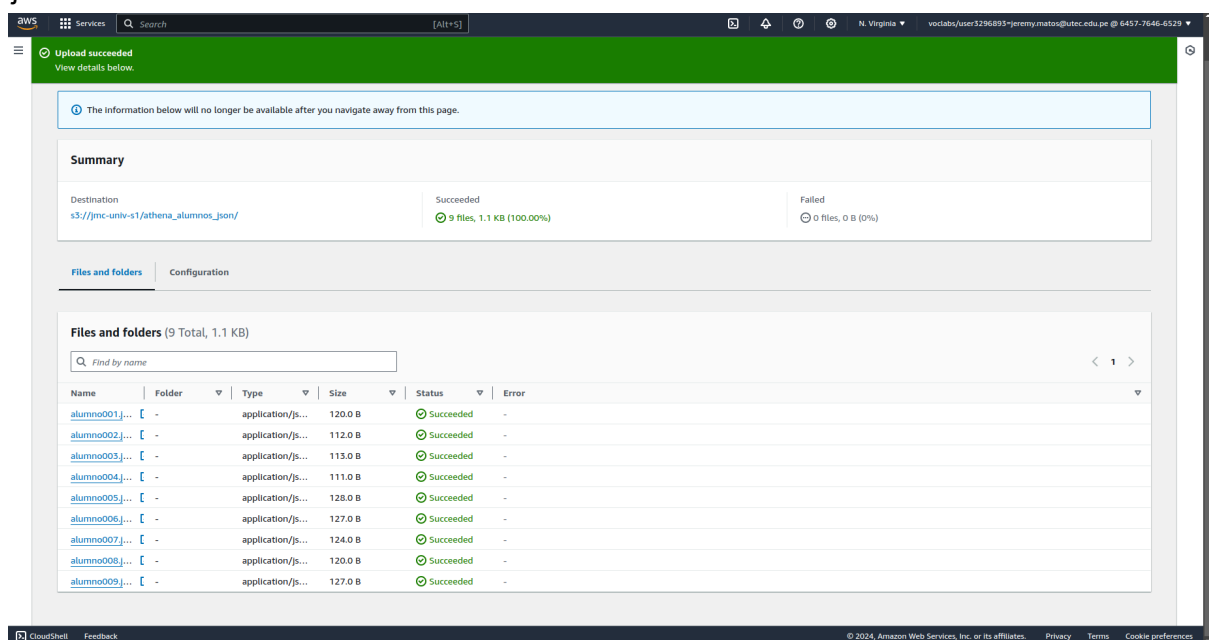
for csv in csvs:
    csv_to_json(csv_file=csv)
```

```
1
2 "alumno_id": 1,
3 "first_name": "Alejandro",
4 "last_name": "Rosalez",
5 "birthday": "2000\12\12",
6 "street_address": "123 Main St.",
7 "city": "Any Town",
8 "state": "MD",
9 "phone": "301-555-0158"
10 ]
```

```
1
2 {
3   "Type": "int"
4 },
5 {
6   "Name": "first_name",
7   "Type": "string"
8 },
9 {
10  "Name": "last_name",
11  "Type": "string"
12 },
13 {
14  "Name": "birthday",
15  "Type": "string"
16 },
17 }
```

```
1 1,Alejandro,Rosalez,2000/12/12,123 Main St.,Any Town,
```

Creamos un folder en el bucket S3 llamado “athena_alumnos_json” y subimos los archivos .json transformados.



Upload succeeded
View details below.

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

Summary

Destination	Succeeded	Failed
s3://jmc-univ-s1/athena_alumnos_json/	9 files, 1.1 KB (100.00%)	0 files, 0 B (0%)

Files and folders | Configuration

Files and folders (9 Total, 1.1 KB)

Find by name

Name	Folder	Type	Size	Status	Error
alumno001.j...	-	application/js...	120.0 B	Succeeded	-
alumno002.j...	-	application/js...	112.0 B	Succeeded	-
alumno003.j...	-	application/js...	113.0 B	Succeeded	-
alumno004.j...	-	application/js...	111.0 B	Succeeded	-
alumno005.j...	-	application/js...	128.0 B	Succeeded	-
alumno006.j...	-	application/js...	127.0 B	Succeeded	-
alumno007.j...	-	application/js...	124.0 B	Succeeded	-
alumno008.j...	-	application/js...	120.0 B	Succeeded	-
alumno009.j...	-	application/js...	127.0 B	Succeeded	-

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Validamos la subida de los archivos.

Amazon S3 > Buckets > jmc-univ-s1 > athena_alumnos_json/

athena_alumnos_json/ Copy S3 URI

Objects Properties

Objects (9) Info Refresh Copy S3 URI Copy URL Download Open Delete Actions Create folder Upload

Find objects by prefix

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	alumno001.json	json	September 21, 2024, 20:15:04 (UTC-05:00)	120.0 B	Standard
<input type="checkbox"/>	alumno002.json	json	September 21, 2024, 20:15:04 (UTC-05:00)	112.0 B	Standard
<input type="checkbox"/>	alumno003.json	json	September 21, 2024, 20:15:05 (UTC-05:00)	113.0 B	Standard
<input type="checkbox"/>	alumno004.json	json	September 21, 2024, 20:15:05 (UTC-05:00)	111.0 B	Standard
<input type="checkbox"/>	alumno005.json	json	September 21, 2024, 20:15:06 (UTC-05:00)	128.0 B	Standard
<input type="checkbox"/>	alumno006.json	json	September 21, 2024, 20:15:06 (UTC-05:00)	127.0 B	Standard
<input type="checkbox"/>	alumno007.json	json	September 21, 2024, 20:15:07 (UTC-05:00)	124.0 B	Standard
<input type="checkbox"/>	alumno008.json	json	September 21, 2024, 20:15:08 (UTC-05:00)	120.0 B	Standard
<input type="checkbox"/>	alumno009.json	json	September 21, 2024, 20:15:08 (UTC-05:00)	127.0 B	Standard

Creamos una tabla llamada “alumnos_json” referenciando al folder del bucket donde se encuentran los archivos “.json” .

AWS Glue

Getting started
ETL Jobs
Visual ETL
Notebooks
Job run monitoring
Data Catalog tables
Data connections
Workflows (orchestration)
Data Catalog
Data Integration and ETL
Legacy pages

What's New
Documentation
AWS Marketplace

☒ Enable compact mode
☒ Enable new navigation

Table format
Data Catalog managed tables support data compaction for Apache Iceberg table type. [Learn more](#)

☒ Standard AWS Glue table (default)
Create a standard AWS Glue table.

☐ Apache Iceberg table
Create a table in Apache Iceberg table format.

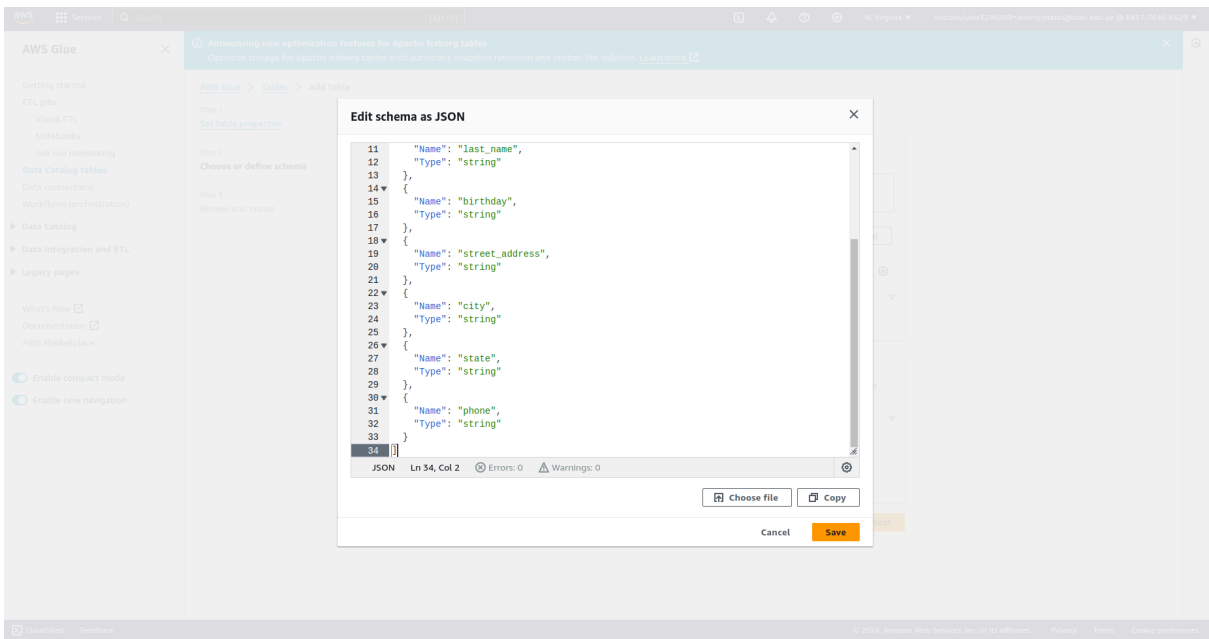
Data store
Select the type of source
☒ S3
☐ Kinesis
☐ Kafka

Data location is specified in
☒ my account
☐ another account

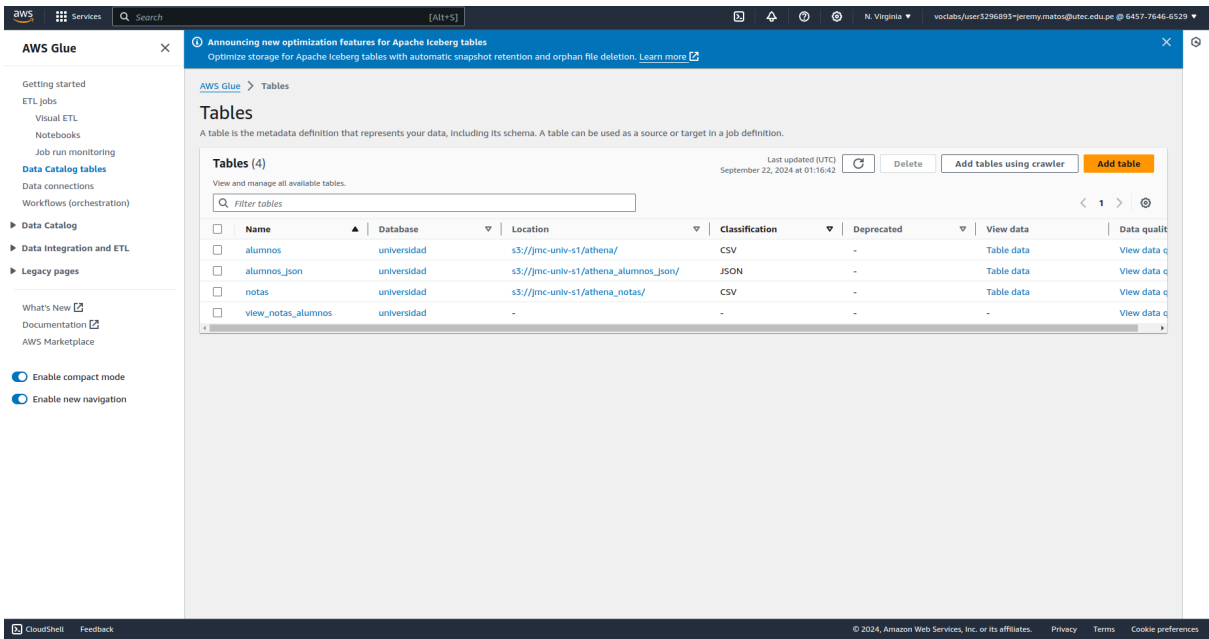
Include path
 View Browse S3
Path must be in the form s3://bucket/prefix/. It must end with a slash (/) and not include any files.

Data format
Classification
Choose the format of the data in your table.
☐ Avro
☐ CSV
☒ JSON
☐ XML
☐ Parquet
☐ ORC

Agregamos el esquema JSON que usamos en el alumnos .csv.



Validamos la creación de la tabla alumnos_json.



Validamos la creación de la tabla haciendo una query simple.

The screenshot displays the Amazon Redshift console interface. At the top, there's a navigation bar with the AWS logo, 'Services' link, a search bar, and user information (N. Virginia, vortabz/user3236093-jeremy.matos@utec.edu.pe). The main panel is divided into several sections:

- Data Source Configuration:** Includes fields for 'Data source' (set to 'AwsDataCatalog'), 'Database' (set to 'universidad'), and a 'Tables and views' section with a 'Create' button.
- Query Editor:** Shows a single query: `SELECT * FROM "AwsDataCatalog"."universidad"."alumnos_json" limit 10;`
- Execution Controls:** Buttons for 'Run again', 'Explain', 'Cancel', 'Clear', and 'Create' are visible.
- Results Summary:** A green banner indicates the query is 'Completed'. It shows performance metrics: 'Time in queue: 105 ms', 'Run time: 469 ms', and 'Data scanned: 1.68 KB'.
- Results Table:** A table titled 'Results (9)' contains 9 rows of student data. The columns are: #, alumno_id, first_name, last_name, birthday, street_address, city, state, and phone.

#	alumno_id	first_name	last_name	birthday	street_address	city	state	phone
1	3	John	Stiles	2006/09/20	1980 8th St.	Nowhere	NY	914-555-0122
2	1	Alejandro	Rosalez	2000/12/12	123 Main St.	Any Town	MD	301-555-0158
3	6	David	Lyons	2005/06/23	69512 King Road	West Kellytown	CA	001-066-838-4285
4	2	Jane	Doe	2004/10/05	456 State St.	Anywhere	WA	360-555-0163