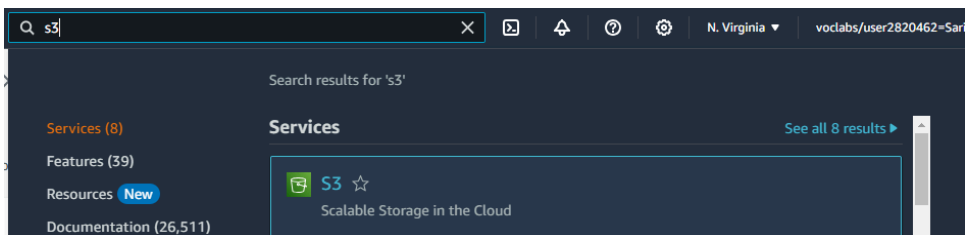


**Sebastián Arias Usma.**  
**C.C 1017932811**  
**Evidencias laboratorios 3 big data**  
**st0263-241 Tópicos de telemática**

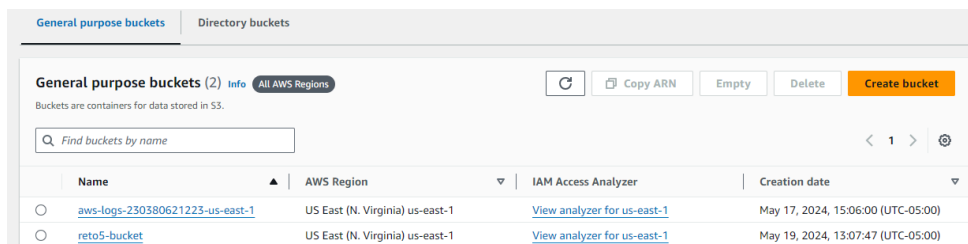
**Github:** <https://github.com/sebasarias17/sariasu-st0263/tree/main/Reto%206>  
**Reto#6 IMPLEMENTACIÓN DE UN DATA WAREHOUSE SENCILLO CON AWS S3, GLUE y ATHENA.**

**Paso # 1:** Creamos un Bucket de S3 en nuestro AWS de la siguiente manera.

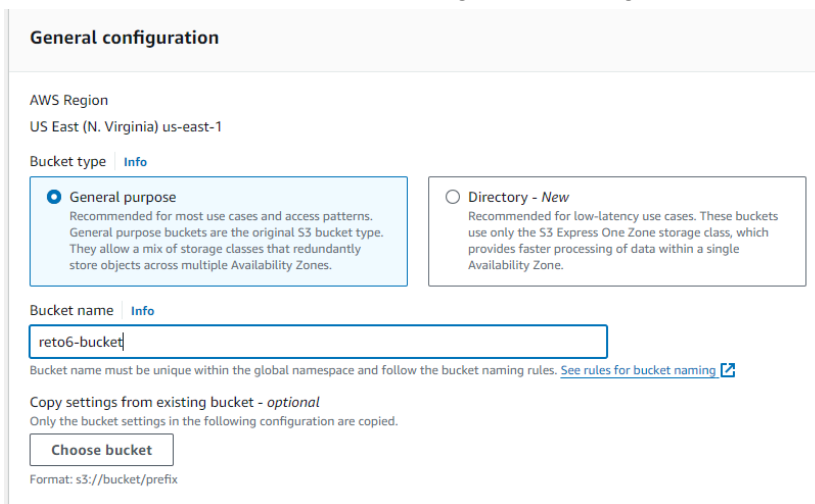
- Nos vamos al apartado de S3 en AWS



- Ahora le damos click al botón de crear bucket



- Creamos el Bucket de S3 Con la siguiente configuración.



- Modificamos el Object Ownership y lo habilitamos.

### Object Ownership [Info](#)


Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

☐ **ACLs disabled (recommended)**

All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

☒ **ACLs enabled**

Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

 We recommend disabling ACLs, unless you need to control access for each object individually or to have the object writer own the data they upload. Using a bucket policy instead of ACLs to share data with users outside of your account simplifies permissions management and auditing.


#### Object Ownership

☒ **Bucket owner preferred**

If new objects written to this bucket specify the bucket-owner-full-control canned ACL, they are owned by the bucket owner. Otherwise, they are owned by the object writer.

☐ **Object writer**

The object writer remains the object owner.

 If you want to enforce object ownership for new objects only, your bucket policy must specify that the bucket-owner-full-control canned ACL is required for object uploads. [Learn more](#)

- Ahora modificamos el acceso para que sea público.

### Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

☐ **Block all public access**

Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

☐ **Block public access to buckets and objects granted through new access control lists (ACLs)**

S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.

☐ **Block public access to buckets and objects granted through any access control lists (ACLs)**

S3 will ignore all ACLs that grant public access to buckets and objects.

☐ **Block public access to buckets and objects granted through new public bucket or access point policies**

S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.

☐ **Block public and cross-account access to buckets and objects through any public bucket or access point policies**

S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.



**Turning off block all public access might result in this bucket and the objects within becoming public**

AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

☒ I acknowledge that the current settings might result in this bucket and the objects within becoming public.

- Y estos ultimos ajustes los dejamos por defecto

### Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

#### Bucket Versioning

☒ **Disable**

☐ **Enable**

**Tags - optional** (0)

You can use bucket tags to track storage costs and organize buckets. [Learn more](#)

No tags associated with this bucket.

[Add tag](#)

---

**Default encryption** [Info](#)

Server-side encryption is automatically applied to new objects stored in this bucket.

**Encryption type** [Info](#)

☒ Server-side encryption with Amazon S3 managed keys (SSE-S3)  
☐ Server-side encryption with AWS Key Management Service keys (SSE-KMS)  
☐ Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)  
 Secure your objects with two separate layers of encryption. For details on pricing, see [DSSE-KMS pricing](#) on the [Storage](#) tab of the [Amazon S3 pricing page](#).

**Bucket Key**

Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. [Learn more](#)

☐ Disable  
☒ Enable

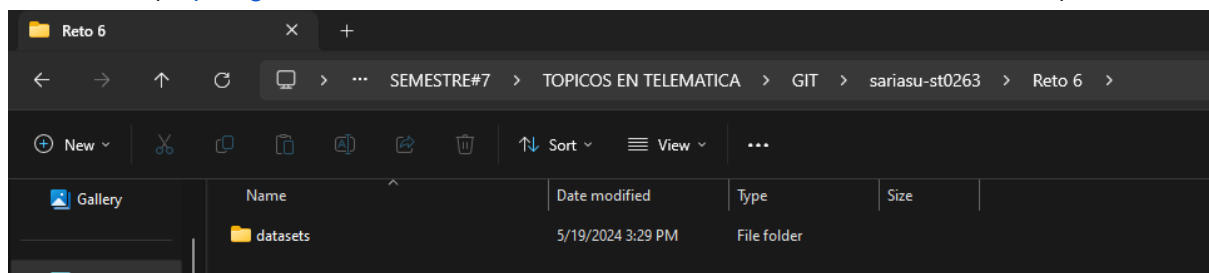
- Ahora le damos click en crear el cluster.

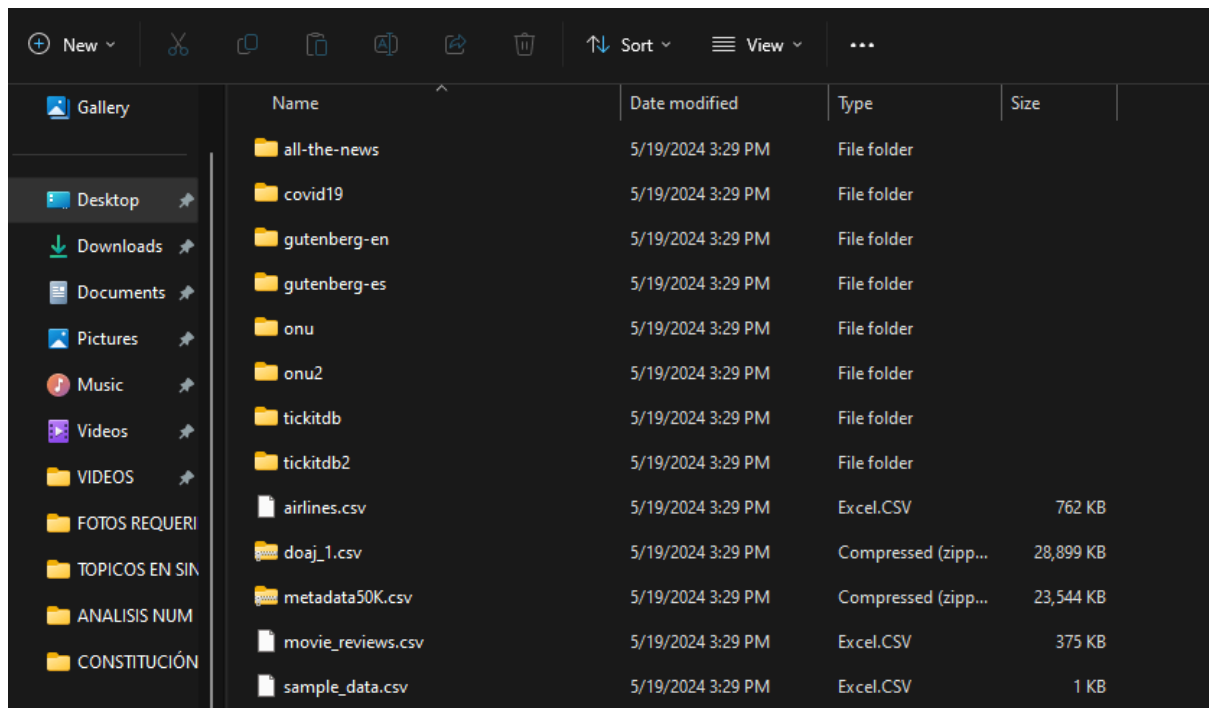
**► Advanced settings**

[i](#) After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.

[Cancel](#) [Create bucket](#)

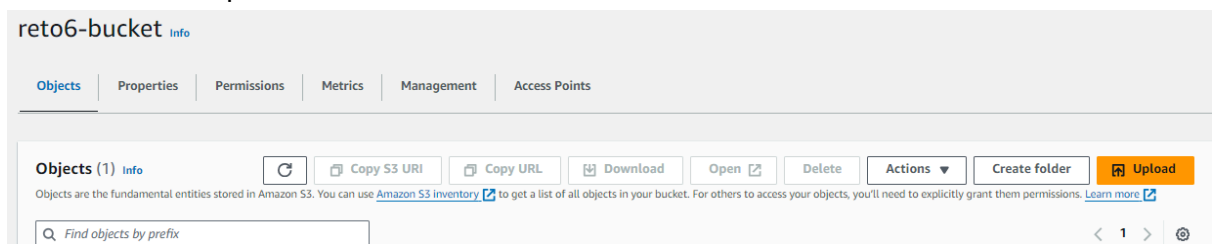
**Paso # 2:** Ahora clonaremos este github, el cual contiene los datasets necesarios para este laboratorio (<https://github.com/sebasarias17/sariasu-st0263/tree/main/Reto%206> ) .



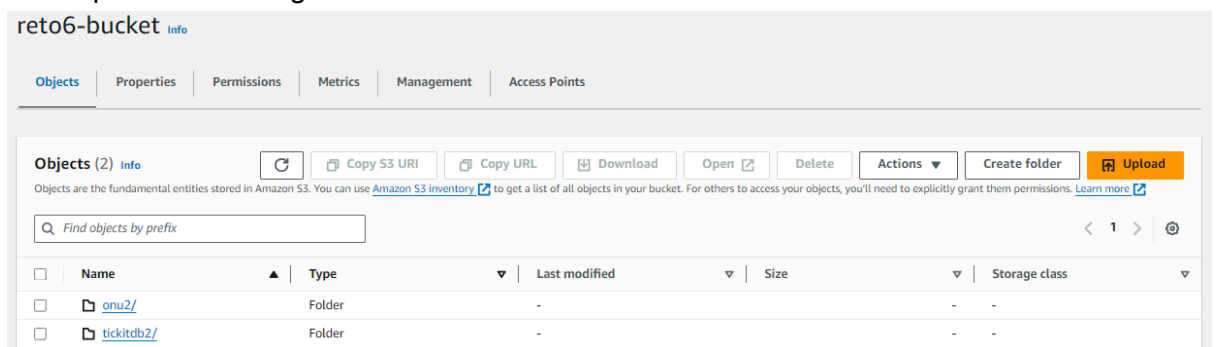


**Paso #3:** Ahora subiremos de manera manualmente a nuestro bucket las carpetas (onu2 y tickitdb2)

- Damos click en upload.

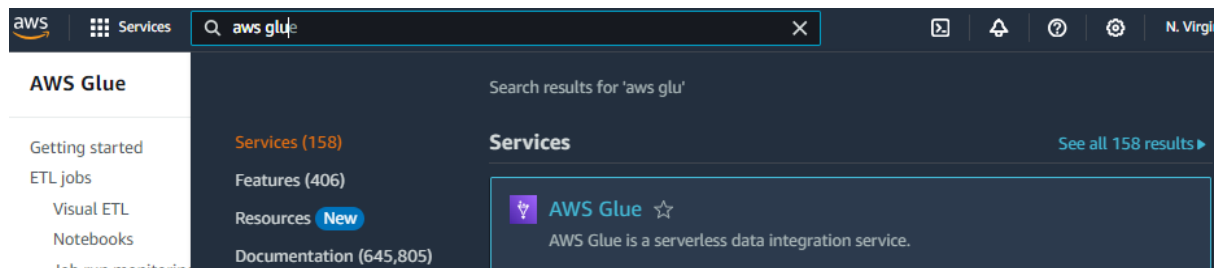


- Ahora subiremos las dos carpetas anteriormente mencionadas, primero una y luego la otra para obtener algo así

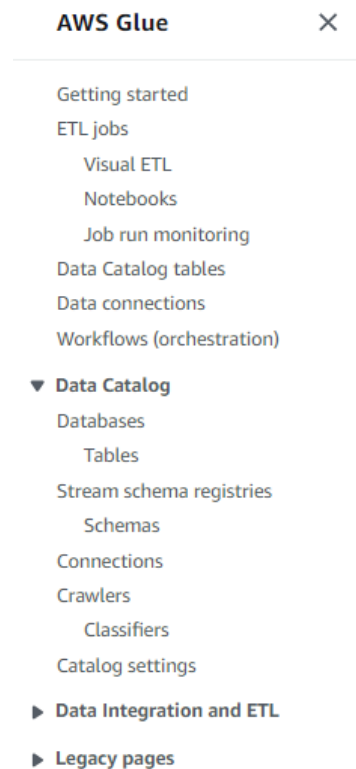


**Paso #4:** Ahora pasaremos a AWS Glue para catalogar y crear las tablas.

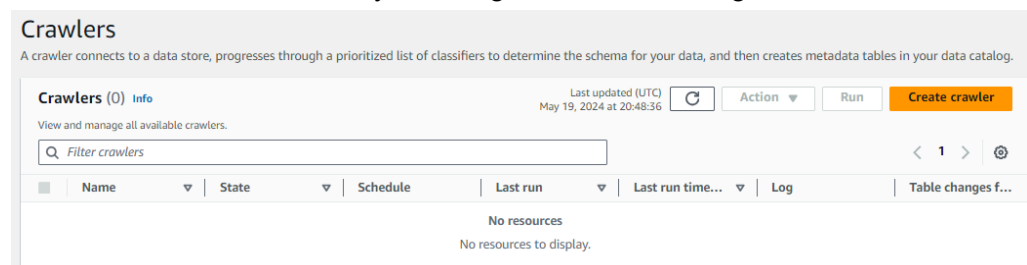
- Accedemos al apartado de AWS Glue en AWS



- Ahora en el menú de la izquierda iremos al apartado de Data Catalog y haremos la configuración en la opción de (crawlers).



- Ahora crearemos el crawler y lo configuraremos de la siguiente manera.



- Le daremos el nombre que deseemos

AWS Glue > Crawlers > Add crawler

Step 1  
**Set crawler properties**

Step 2  
Choose data sources and classifiers

Step 3  
Configure security settings

Step 4  
Set output and scheduling

Step 5  
Review and create

## Set crawler properties

**Crawler details** [Info](#)

Name

catalogaronu

Name can be up to 255 characters long. Some character set including control characters are prohibited.

Description - *optional*

Enter a description

Descriptions can be up to 2048 characters long.

► **Tags - optional**

Use tags to organize and identify your resources.

Cancel **Next**

- Adicionamos una fuente de datos en el apartado de Add data source

## Choose data sources and classifiers

**Data source configuration**

Is your data already mapped to Glue tables?

☒ **Not yet**  
Select one or more data sources to be crawled.

☐ **Yes**  
Select existing tables from your Glue Data Catalog.

**Data sources (0)** [Info](#)

The list of data sources to be scanned by the crawler.

Edit Remove **Add a data source**

Type	Data source	Parameters
You don't have any data sources.		

**Add a data source**

► **Custom classifiers - optional**

A classifier checks whether a given file is in a format the crawler can handle. If it is, the classifier creates a schema in the form of a StructType object that matches that data format.

Cancel **Previous** **Next**

- Nos iremos a modificar el path donde buscará los archivos, haremos click en browse S3

### S3 path

Browse for or enter an existing S3 path.

**View** **Browse S3**

All folders and files contained in the S3 path are crawled. For example, type s3://MyBucket/MyFolder/ to crawl all objects in MyFolder within MyBucket.

- Ahora ingresamos el bucket que hemos creado, en mi caso es reto6-bucket, y daremos click en su nombre

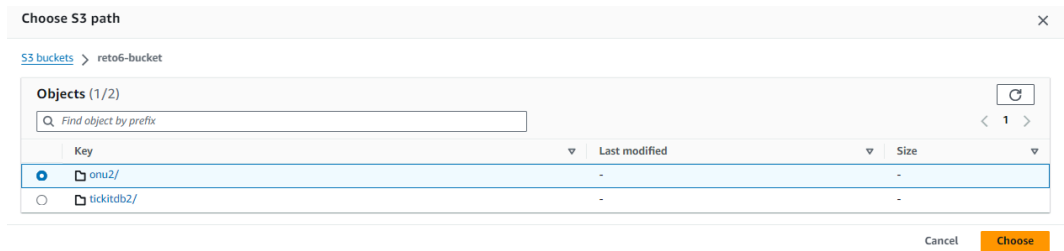
### Choose S3 path

S3 buckets

**Buckets (3)**

Name	Creation date
<input type="radio"/> aws-logs-230380621223-us-east-1	May 17, 2024 at 20:06:00
<input type="radio"/> reto5-bucket	May 19, 2024 at 18:07:47
<input type="radio"/> reto6-bucket	May 19, 2024 at 20:25:02

- Ahora seleccionaremos la carpeta de “onu2” y le daremos choose



- Borraremos el ultimo slash del path para obtener algo así

#### S3 path

Browse for or enter an existing S3 path.

Q s3://reto6-bucket/onu2 X View Browse S3

All folders and files contained in the S3 path are crawled. For example, type s3://MyBucket/MyFolder/ to crawl all objects in MyFolder within MyBucket.

- Ya teniendo la configuración de esta manera le daremos click al boton de “Add an S3 data source”

#### Add data source

Data source  
Choose the source of data to be crawled.

S3

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

Clear selection Add new connection

Location of S3 data  
☒ In this account  
☐ In a different account

S3 path  
Browse for or enter an existing S3 path.

Q s3://reto6-bucket/onu2 X View Browse S3

All folders and files contained in the S3 path are crawled. For example, type s3://MyBucket/MyFolder/ to crawl all objects in MyFolder within MyBucket.

Subsequent crawler runs  
This field is a global field that affects all S3 data sources.

☒ Crawl all sub-folders  
 Crawl all folders again with every subsequent crawl.

☐ Crawl new sub-folders only  
 Only Amazon S3 folders that were added since the last crawl will be crawled. If the schemas are compatible, new partitions will be added to existing tables.

☐ Crawl based on events  
 Rely on Amazon S3 events to control what folders to crawl.

Cancel Add an S3 data source

- Ahora le daremos next a este paso

## Choose data sources and classifiers

**Data source configuration**

Is your data already mapped to Glue tables?

☒ Not yet  
Select one or more data sources to be crawled.

☐ Yes  
Select existing tables from your Glue Data Catalog.

**Data sources (1)** [Info](#)
Edit Remove Add a data source

The list of data sources to be scanned by the crawler.

	Type	Data source	Parameters
<input type="radio"/>	S3	s3://reto6-bucket/onu2	Recrawl all

► **Custom classifiers - optional**

A classifier checks whether a given file is in a format the crawler can handle. If it is, the classifier creates a schema in the form of a StructType object that matches that data format.

Cancel Previous Next

- Configuraremos las opciones de seguridad y damos next

## Configure security settings

**IAM role** [Info](#)

Existing IAM role

LabRole

▼

↻

View

Create new IAM role Update chosen IAM role

Only IAM roles created by the AWS Glue console and have the prefix "AWSGlueServiceRole-" can be updated.

**Lake Formation configuration - optional**

Allow the crawler to use Lake Formation credentials for crawling the data source. [Learn more.](#)

☐ Use Lake Formation credentials for crawling S3 data source  
Checking this box will allow the crawler to use Lake Formation credentials for crawling the data source. If the data source is registered in another account, you must provide the registered account ID. Otherwise, the crawler will crawl only those data sources associated to the account. Only applicable to S3, Glue Catalog, Iceberg, and Hudi data sources.

► **Security configuration - optional**

Enable at-rest encryption with a security configuration.

Cancel Previous Next

- Ahora para crear una nueva base de datos la adicionamos en el botón "Add database"

**Output configuration** [Info](#)

Target database

Choose a database

▼

↻

Clear selection Add database

⚠ Target database is required

Table name prefix - optional

Type a prefix added to table names

Maximum table threshold - optional

This field sets the maximum number of tables the crawler is allowed to generate. In the event that this number is surpassed, the crawl will fail with an error. If not set, the crawler will automatically generate the number of tables depending on the data schema.

Type a number greater than 0

► **Advanced options**

- La crearemos así y le daremos "Next"



## Create a database

Create a database in the AWS Glue Data Catalog.

### Database details

Name

Database name is required, in lowercase characters, and no longer than 255 characters.

Description - *optional*

Descriptions can be up to 2048 characters long.

### Database settings

Location - *optional*

Set the URI location for use by clients of the Data Catalog.

Cancel

Create database

- Habiendo creado la base de datos la asignaremos y daremos "Next"

## Set output and scheduling

### Output configuration [Info](#)

Target database



Clear selection

Add database [↗](#)

Table name prefix - *optional*

Maximum table threshold - *optional*

This field sets the maximum number of tables the crawler is allowed to generate. In the event that this number is surpassed, the crawl will fail with an error. If not set, the crawler will automatically generate the number of tables depending on the data schema.

► Advanced options

### Crawler schedule

You can define a time-based schedule for your crawlers and jobs in AWS Glue. The definition of these schedules uses the Unix-like [cron](#) syntax. [Learn more](#) [↗](#)

Frequency

Cancel

Previous

Next

- Por ultimo le daremos a "create crawler"

### Step 4: Set output and scheduling

Edit

#### Set output and scheduling

Database

onur6db

Table prefix - *optional*

-

Maximum table threshold -

*optional*

-

Schedule

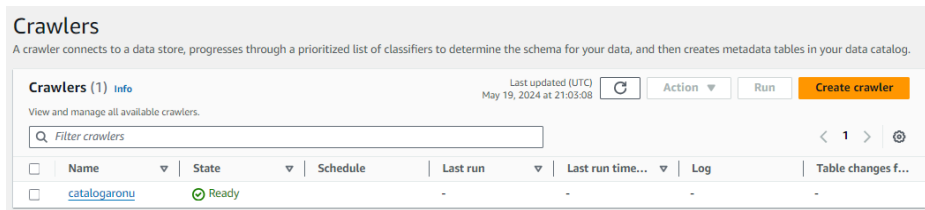
On demand

Cancel

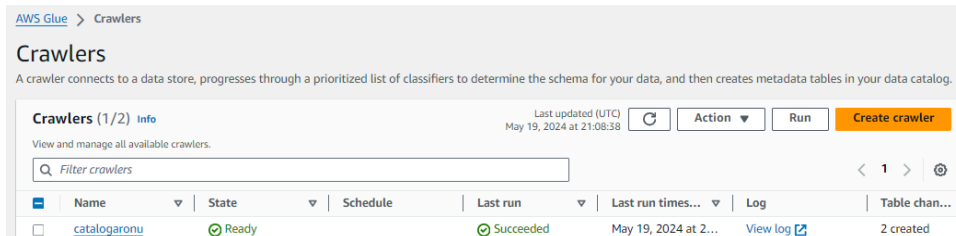
Previous

Create crawler

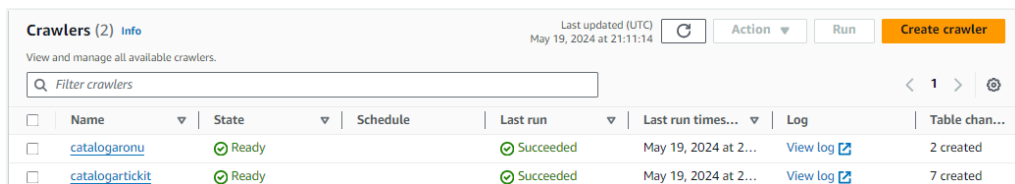
- Así nos debería salir el crawler que hicimos



- Ahora lo ejecutamos seleccionando el crawler y seleccionando “Run”

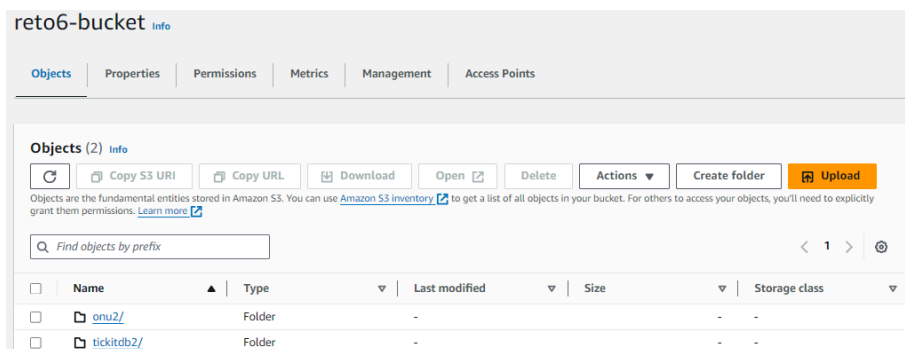


- Por ultimo seguiremos los mismos pasos para crear el crawler de tickitdb, para obtener esto en los crawlers.

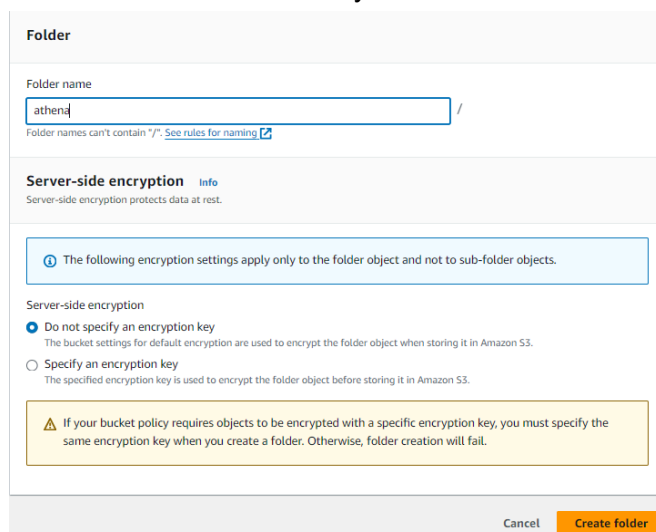


## Paso #5: Ahora crearemos el directorio de salida de Athena en S3

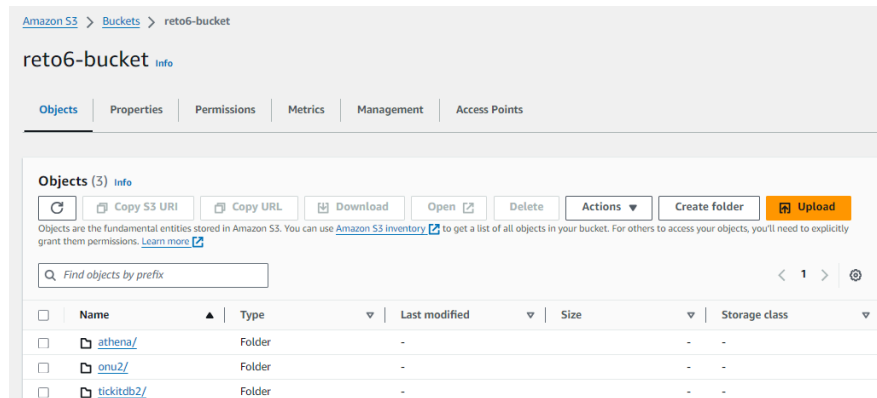
- Haremos click en create Folder



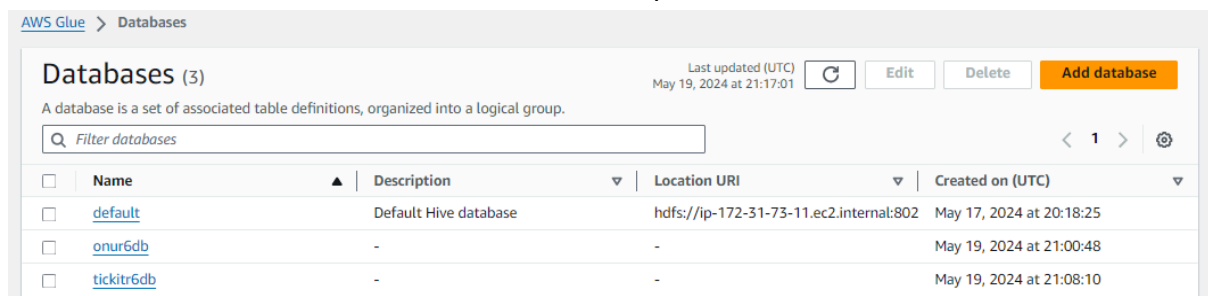
- Creamos la carpeta con estas credenciales y daremos click en “create folder”



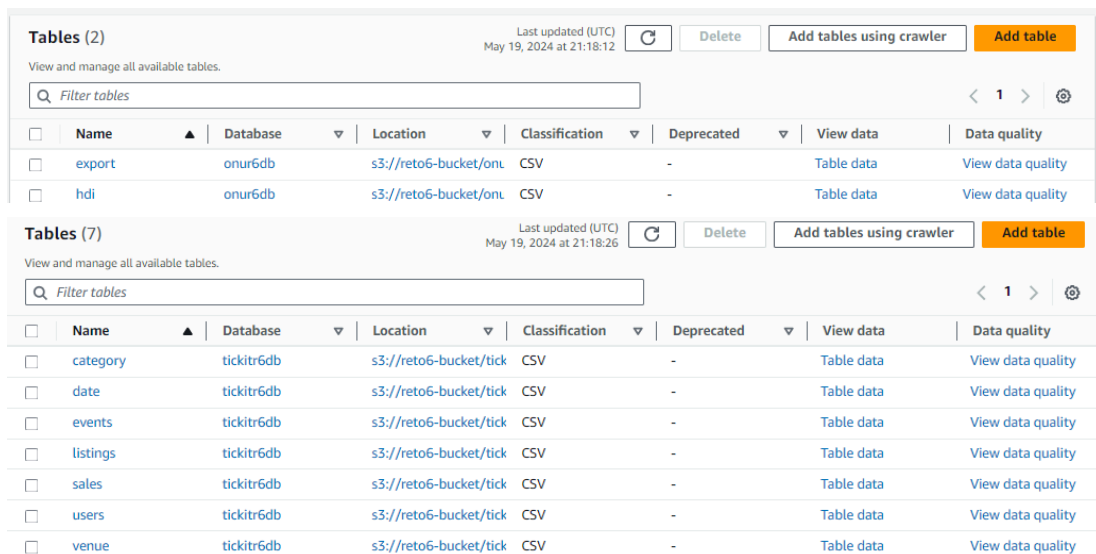
- Asi obtendremos estas 3 carpetas.



## Paso #6: Ahora en nuestro AWS Glue iremos al apartado de bases de datos



- Ahora cuando accedemos a cualquier base de datos que creamos anteriormente nos deberan salir las tablas que se crearon, de esta manera



- De la misma manera si accedemos a cualquier tabla podremos ver los datos y columnas que tiene cada una, en este caso yo accedí a la carpeta "hdi" en la db de onu2

Schema (9)

View and manage the table schema.


Q

Filter schemas

<

1

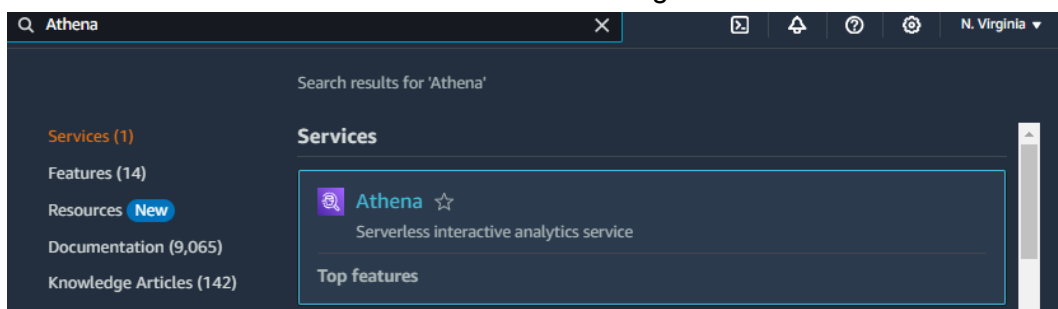
>



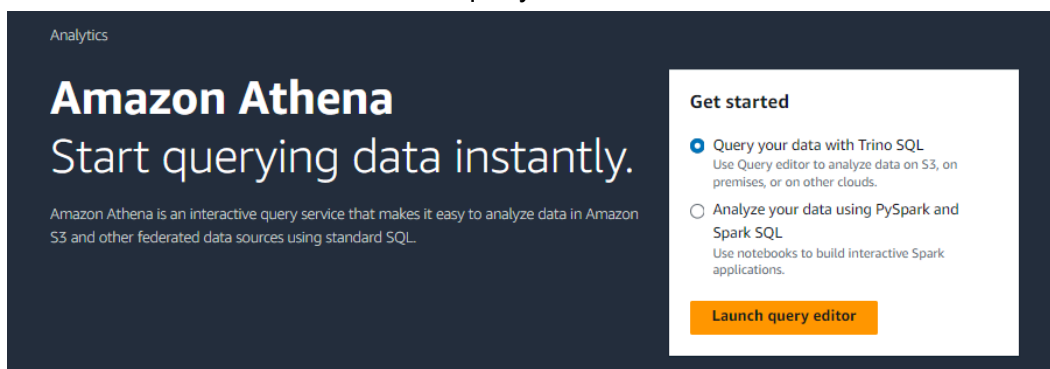
#	Column name	Data type	Partition key	Comment
1	id	bigint	-	-
2	country	string	-	-
3	human development index (...)	double	-	-
4	life expectancy at birth	double	-	-
5	mean years of schooling	double	-	-
6	expected years of schooling	double	-	-
7	gross national income (gni) ...	bigint	-	-
8	gni per capita rank minus h...	bigint	-	-
9	nonincome hdi	double	-	-

**Paso #7:** Ahora podremos realizar consultas SQL mediante athena de la siguiente manera

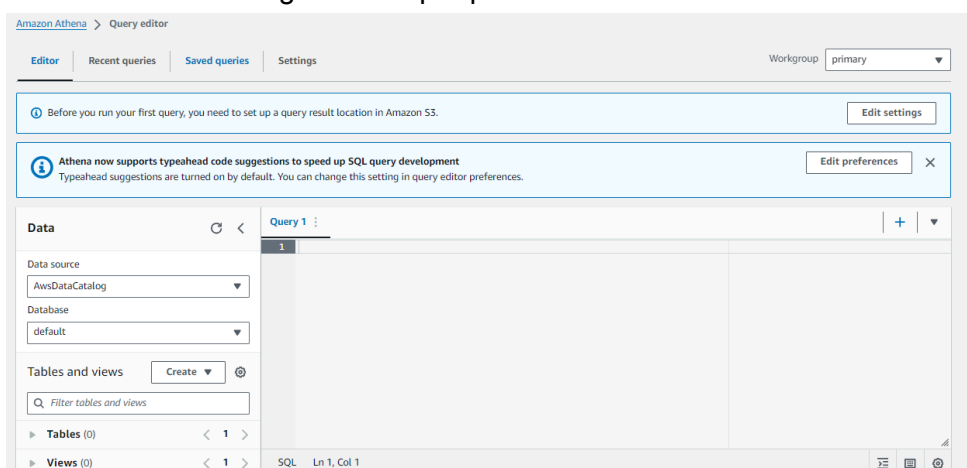
- Accedemos al servicio de Athena en AWS de la siguiente manera.



- Accederemos dando click a “Launch query editor”



- Ahora debemos configurar esto por primera vez en el botón de “Edit Settings”



- Ahora seleccionamos el directorio de salida que creamos anteriormente en nuestro bucket llamado “Athena”

#### Location of query result - optional

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

Q s3://bucket/prefix/object/

View

Browse S3

- Aca seleccionamos nuestro bucket con nuestro nombre en este caso “reto6-bucket”

Choose S3 data set

S3 buckets

Bucket (3)

Filter bucket

	Name	Creation date
<input type="radio"/>	aws-logs-230380621223-us-east-1	2024-05-17T15:06:00.000-05:00
<input type="radio"/>	reto5-bucket	2024-05-19T13:07:47.000-05:00
<input type="radio"/>	reto6-bucket	2024-05-19T15:25:02.000-05:00

Cancel Choose

- Ahora seleccionamos la carpeta que creamos con el nombre “Athena” y daremos click en “choose”

Choose S3 data set

S3 buckets > reto6-bucket

Objects (1/3)

Find object by prefix

	Key	Last modified	Size
<input checked="" type="radio"/>	athena	-	-
<input type="radio"/>	onu2	-	-
<input type="radio"/>	ticketdb2	-	-

Cancel Choose

- Por último daremos click en “Save” cuando nuestras configuraciones de vean así

Manage settings

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.

Q s3://reto6-bucket/athena

View Browse S3

**You can create and manage lifecycle rules for this bucket**

Use Amazon S3 lifecycle rules to store your query results and metadata cost effectively or to delete them after a period of time.

[Learn more](#)

[Lifecycle configuration](#)

Expected bucket owner - optional

Specify the AWS account ID that you expect to be the owner of your query results output location bucket.

Enter AWS account ID

☐ Assign bucket owner full control over query results

Enabling this option grants the owner of the S3 query results bucket full control over the query results. This means that if your query result location is owned by another account, you grant full control over your query results to the other account.

☐ Encrypt query results

Cancel Save

- Ahora volvemos a nuestro editor y seleccionamos la base de datos en la que haremos consultas y podremos empezar a realizar consultas con comandos SQL

**Data**

Data source: AwsDataCatalog

Database: onur6db

Tables and views: **Create**

Filter tables and views

**Tables (2)**

- export
- hdi

**Query 1**

```
1 SELECT * FROM "onur6db"."hdi" limit 10;
```

SQL Ln 1, Col 40

**Run again** **Explain** **Cancel** **Clear** **Create**

Reuse query results up to 60 minutes ago

El comando que utilice fue “SELECT \* FROM "onur6db"."hdi" limit 10;”

**Results (10)** **Copy** **Download results**

Search rows

#	id	country	human development index (hdi)	life expectancy at birth	mean years of schooling	expected years of schooling	gross national income (gni) per capita	gni per capita a rank minus hdi rank
1	1	Norway	0.943	81.1	12.6	17.3	47557	6
2	2	Australia	0.929	81.9	12.0	18.0	34431	16
3	3	Netherlands	0.91	80.7	11.6	16.8	36402	9
4	4	United States	0.91	78.5	12.4	16.0	43017	6
5	5	New Zealand	0.908	80.7	12.5	18.0	23737	30
6	6	Canada	0.908	81.0	12.1	16.0	35166	10
7	7	Ireland	0.908	80.6	11.6	18.0	29322	19
8	8	Liechtenstein	0.905	79.6	10.3	14.7	83717	-6
9	9	Germany	0.905	80.4	12.2	15.9	34854	8
10	10	Sweden	0.904	81.4	11.7	15.7	35837	4

Y podremos hacer lo mismo con nuestra otra base de datos de tickit

