



Workshop Analytics

Guía Práctica de Laboratorio

AWS Solutions Architecture Iberia

Table of Contents

Introducción.....	3
Pre-requisitos.....	3
GUÍA DE PRÁCTICAS.....	4
Práctica 1. Construye una solución de analítica serverless con Amazon S3 y AWS Glue.....	4
Crear bucket en Amazon S3.....	4
Crear rol con permisos para utilizar los servicios	5
Crear database.....	7
Modificar script de Spark con la transformación	8
Crear Job de transformación con AWS Glue	10
Creación de crawlers en AWS Glue.....	12
Ejecución de queries con Amazon Athena.....	15

Introducción

AWS Glue es un servicio de extracción, transformación y carga de datos (ETL) completamente administrado que ayuda a los clientes a preparar y cargar los datos para su análisis. Puede crear y ejecutar un trabajo de ETL con tan solo unos clics en la consola de administración de AWS. Simplemente debe apuntar AWS Glue a sus datos almacenados en AWS y AWS Glue encontrará sus datos y almacenará los metadatos asociados (p. ej., esquemas y definiciones de tablas) en el catálogo de datos de AWS Glue. Una vez catalogados, puede realizar búsquedas y consultas inmediatamente en sus datos, que están disponibles para operaciones de ETL.

Funcionamiento

Seleccionando un origen y un destino para los datos, AWS Glue genera código ETL en Scala o Python para extraer datos del origen, transformar los datos de manera que se correspondan con los esquemas de destino y cargarlos en el destino. Puede editar y probar el código y depurar errores mediante la consola, en su IDE favorito o en cualquier bloc de notas.

Beneficios

- **Menos complicaciones:** AWS Glue se integra en una amplia variedad de servicios de AWS, lo que simplifica el proceso de incorporación. AWS Glue es compatible de manera nativa con datos almacenados en Amazon Aurora y con los demás motores de Amazon RDS, Amazon Redshift y Amazon S3, así como también con los motores de bases de datos comunes y las bases de datos de su nube virtual privada (Amazon VPC) que se ejecutan en Amazon EC2.
- **Rentabilidad:** AWS Glue es un servicio sin servidor. No es necesario aprovisionar ni administrar infraestructura. AWS Glue administra el aprovisionamiento, la configuración y el escalado de los recursos necesarios para ejecutar sus trabajos de ETL en un entorno Apache Spark totalmente administrado y escalable. Solo paga por los recursos utilizados mientras se ejecutan los trabajos.
- **Mayor eficacia:** AWS Glue automatiza gran parte del proceso de creación, mantenimiento y ejecución de trabajos de ETL. AWS Glue rastrea sus orígenes de datos, identifica formatos de datos y sugiere esquemas y transformaciones. AWS Glue genera automáticamente el código para ejecutar sus transformaciones de datos y procesos de carga.

<https://aws.amazon.com/es/glue/>

Pre-requisitos

Cuenta de AWS: Se debe haber asignado una cuenta a cada usuario del workshop.

Navegador: Se recomienda utilizar una versión actualizada de **Chrome** o **Firefox**.

GUÍA DE PRÁCTICAS

Práctica 1. Construye una solución de analítica serverless con Amazon S3 y AWS Glue

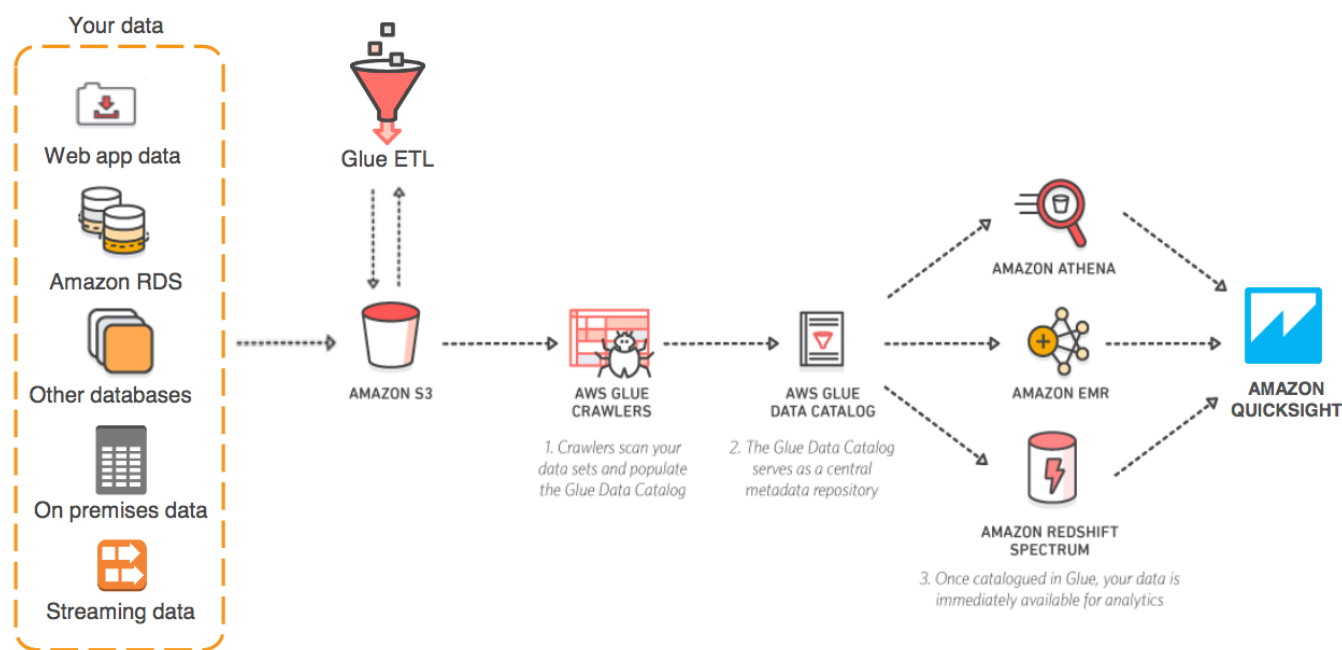
En esta práctica, seguiremos los pasos para crear una solución de ejemplo usando servicios de analítica de AWS totalmente serverless, es decir sin necesidad de provisionar infraestructura o instancias sino apoyándonos exclusivamente en servicios gestionados por AWS.

En el ejercicio partiremos de un fichero con datos en formato JSON y distintos niveles en árbol o nested, que está alojado en un bucket de Amazon S3 que os proporcionaremos.

A partir de allí, usaremos AWS Glue para:

- Descubrir automáticamente el esquema de los datos en el fichero
- Realizar una transformación de ejemplo para convertir el árbol de datos en un formato de árbol más plano
- Descubrir el esquema de los datos en los nuevos ficheros generados

Finalmente usaremos Amazon Athena para realizar queries directamente apoyándonos en el catálogo de datos.



Ejemplo de arquitectura de referencia para una solución de analítica serverless en AWS

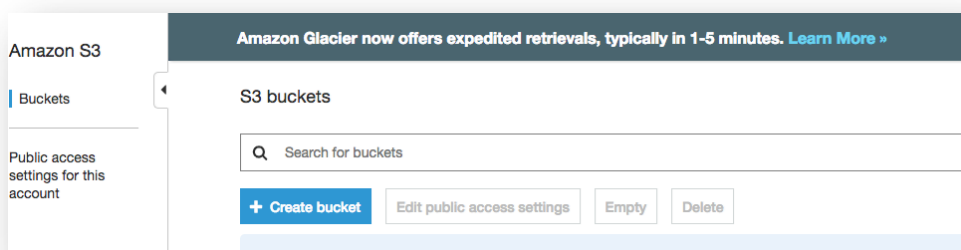
Crear bucket en Amazon S3

1. Accede a la consola de AWS con el URL de cuenta que se te ha entregado. Verifica que el ID de cuenta en la esquina superior derecha coincide con el que se te ha asignado para estas prácticas de laboratorio.
2. En la esquina superior-derecha de la consola de AWS, verifica:
 - Que el ID de cuenta coincide con el que se te ha asignado para estas prácticas de laboratorio.

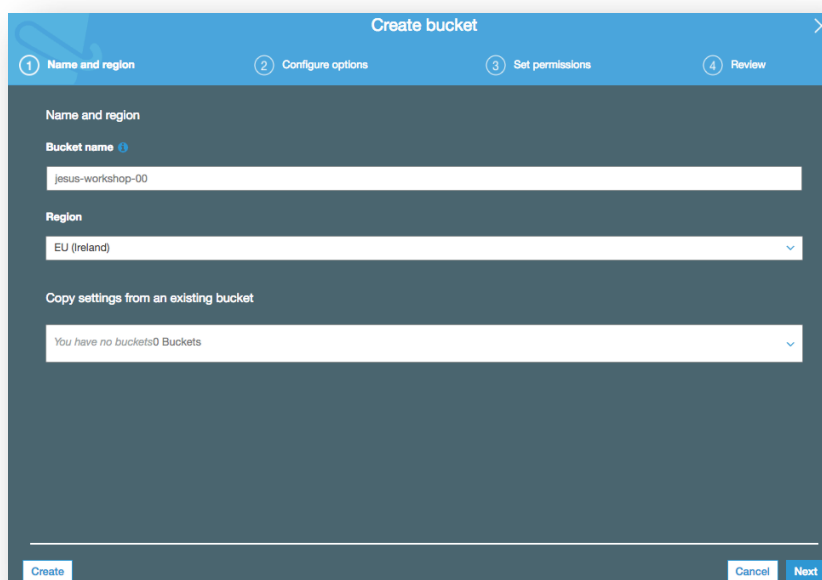
- Que la región seleccionada es "N. Virginia". Si no es así cámbiala haciendo click sobre la región actual.

3. Busca y haz click en "S3" en la lista de servicios, o escríbelo en la caja de texto de "Find Services" como "S3"

4. Haz click en "Create bucket"



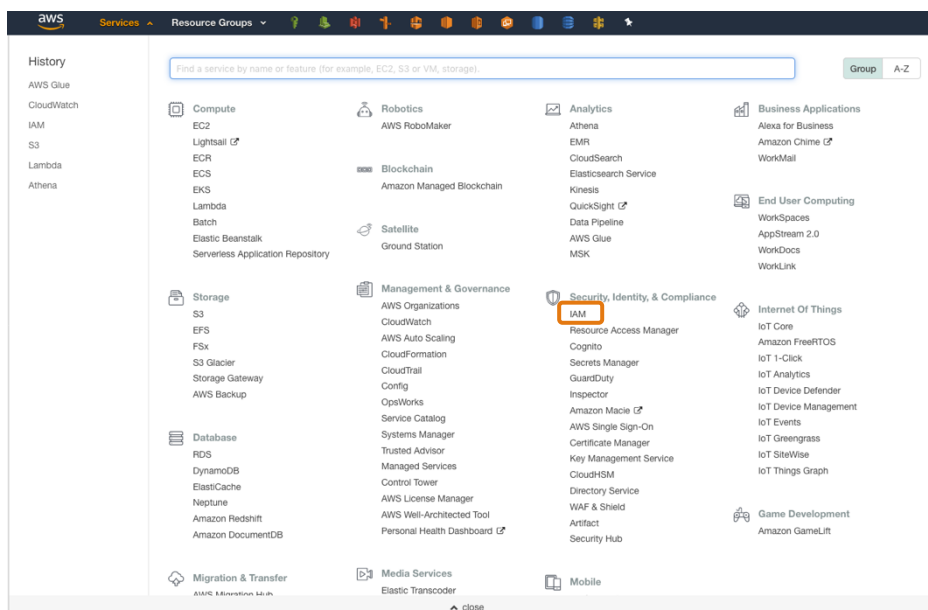
5. Como nombre del bucket, introduce "<nombre>-workshop-<XX>", siendo XX un número aleatorio del 00 al 99. Asegúrate que la región es (N. Virginia) y haz click en Create:



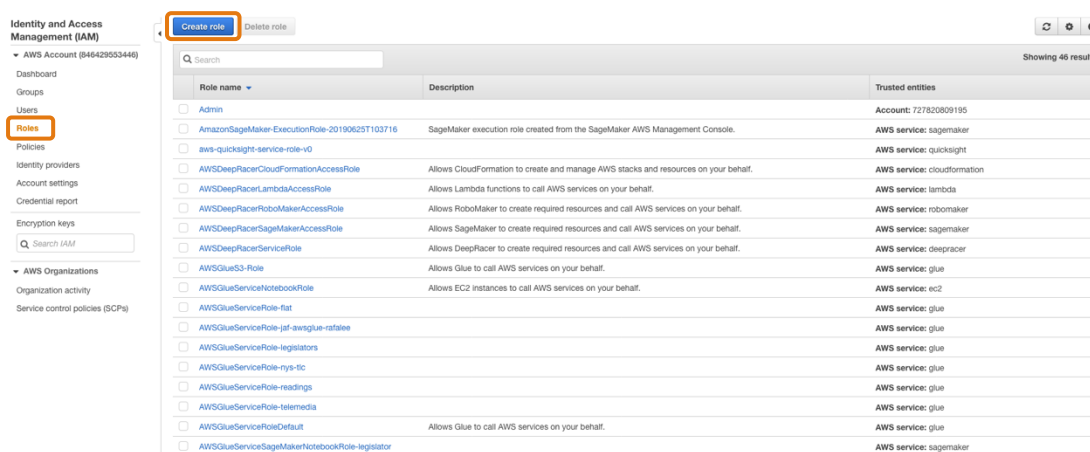
6. Haz click en el bucket creado, y luego en "Create folder" para crear dos carpetas. Crea una llamada "tmp" y otra carpeta llamada "output" dentro del bucket. Estas carpetas nos servirán para guardar los ficheros resultantes de la ETL que hagamos en nuestra solución de analítica

Crear rol con permisos para utilizar los servicios

1. Crear un role que permita al job que crearemos después acceder a S3 y a AWS Glue. Se podrían restringir más los permisos para ajustarlos sólo a los recursos necesarios, pero para simplificar, utilizaremos unas políticas predefinidas. Para ello, en la consola de AWS entraremos en el servicio IAM:



2. Entrad en la sección de Roles para crear uno nuevo:



3. Seleccionar crear un nuevo role para un servicio de AWS que será AWS Glue y pulsar en "Siguiete":

Create role

Select type of trusted entity

AWS service
EC2, Lambda and others

Another AWS account
Belonging to you or 3rd party

Web identity
Cognito or any OpenID provider

SAML 2.0 federation
Your corporate directory

Allows AWS services to perform actions on your behalf. [Learn more](#)

Choose the service that will use this role

EC2
Allows EC2 instances to call AWS services on your behalf.

Lambda
Allows Lambda functions to call AWS services on your behalf.

API Gateway	Comprehend	EMR	Kinesis	S3
AWS Backup	Config	ElastiCache	Lambda	SMS
AWS Support	Connect	Elastic Beanstalk	Lex	SNS
Amplify	DMS	Elastic Container Service	License Manager	SWF
AppSync	Data Lifecycle Manager	Elastic Transcoder	Machine Learning	SageMaker
Application Auto Scaling	Data Pipeline	ElasticLoadBalancing	Macie	Security Hub
Application Discovery Service	DataSync	Forecast	MediaConvert	Service Catalog
Batch	DeepLens	Glue	OpsWorks	Step Functions
CloudFormation	Directory Service	Greengrass	Personalize	Storage Gateway
CloudHSM	DynamoDB	GuardDuty	RAM	Transfer
CloudTrail	EC2	Inspector	RDS	Trusted Advisor
CloudWatch Application Insights	EC2 - Fleet	IoT	Redshift	VPC
CloudWatch Events	EC2 Auto Scaling	IoT Things Graph	Rekognition	WorkLink
CloudWatch Logs	EKS	KMS	RoboMaker	WorkMail

* Required

Cancel Next: Permissions

4. En el siguiente paso seleccionaremos las políticas: **AmazonS3FullAccess** y **AWSGlueConsoleFullAccess**. Para ello nos ayudaremos del buscador superior y de las cajitas con check que hay al lado de cada política. Una vez seleccionadas ambas pasaremos al paso siguiente.

5. En este caso no nos falta añadir ninguna etiqueta.

6. En el nombre del role poned **AWSGlueS3-Role**

Create role

Review

Provide the required information below and review this role before you create it.

Role name*
Use alphanumeric and "+=,@-_" characters. Maximum 64 characters.

Role description
Maximum 1000 characters. Use alphanumeric and "+=,@-_" characters.

Trusted entities AWS service: glue.amazonaws.com

Policies

- AWSGlueConsoleFullAccess**
- AmazonS3FullAccess**

Permissions boundary Permissions boundary is not set

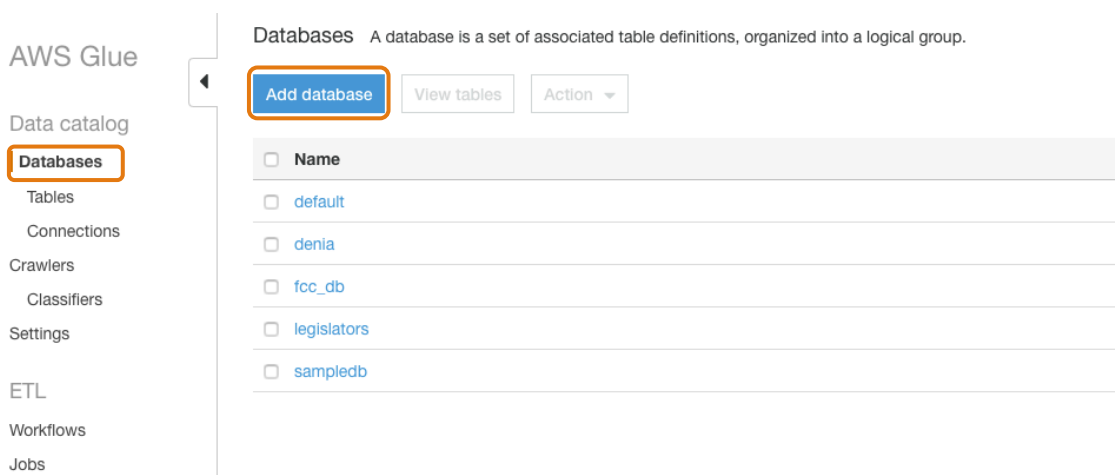
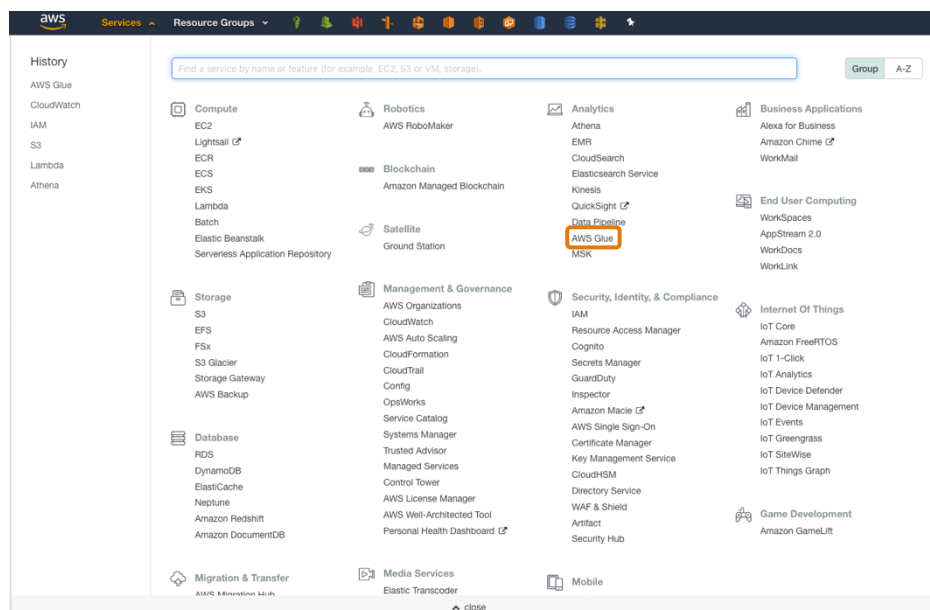
No tags were added.

* Required

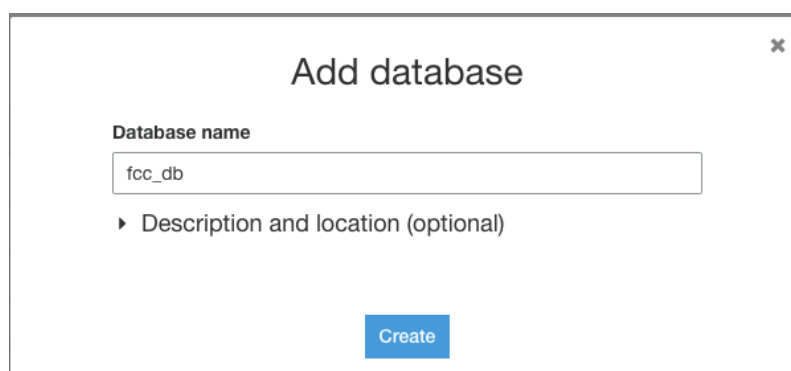
Cancel Previous Create role

Crear database

1. En la consola de AWS Glue, crear una nueva database (no es una database convencional, sino un lugar donde se guardará el esquema del fichero JSON que se utilizará como base para parsear):



2. Poner como nombre de database **fcc_db** y pulsar el botón “Crear”.



Modificar script de Spark con la transformación

1. Descargar el script proporcionado en el repositorio para poder modificarlo.
2. Sustituir los “placeholders” marcados en naranja por los datos correspondientes:


```

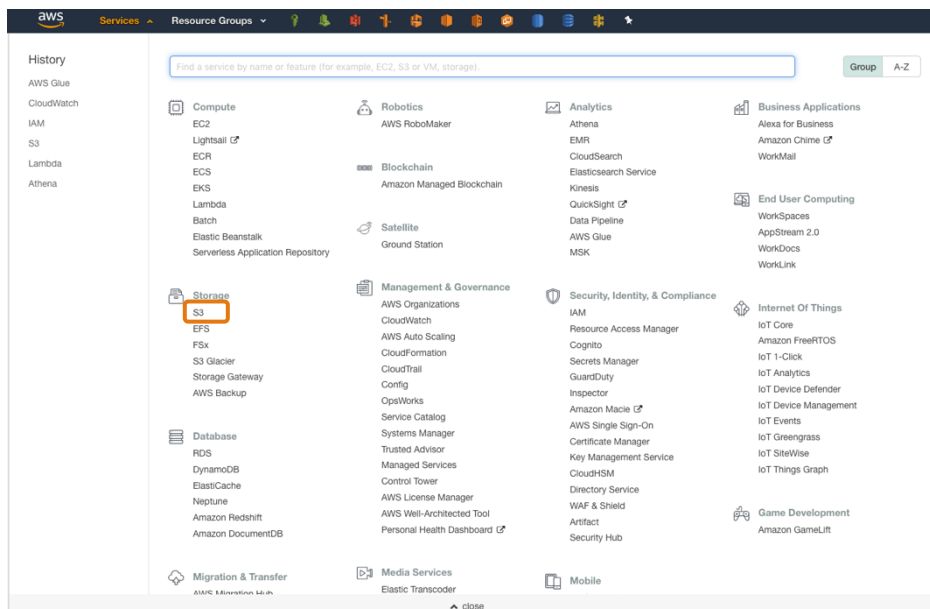
import sys
from awsglue.transforms import *
from awsglue.utils import getResolvedOptions
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.job import Job
#from awsglue.transforms import Relationalize

# Begin variables to customize with your information
glue_source_database = "<database>"
glue_source_table = "<table>"
glue_temp_storage = "<path output>" # example "s3://jaf-awsglue/output"
glue_relationalize_output_s3_path = "<path flat>" # example "s3://jaf-awsglue/flat"
dfc_root_table_name = "root" # default value is "roottable"
# End variables to customize with your information

glueContext = GlueContext(SparkContext.getOrCreate())
datasource0 = glueContext.create_dynamic_frame.from_catalog(database=glue_source_database,
table_name=glue_source_table, transformation_ctx="datasource0")
dfc = Relationalize.apply(frame=datasource0, staging_path=glue_temp_storage,
name=dfc_root_table_name, transformation_ctx="dfc")
blogdata = dfc.select(dfc_root_table_name)
blogdataoutput = glueContext.write_dynamic_frame.from_options(frame=blogdata,
connection_type="s3", connection_options={"path": glue_relationalize_output_s3_path}, format="orc",
transformation_ctx="blogdataoutput")

```

3. Subir el script resultante a S3 (en el root del bucket que hemos creado) para, en el siguiente paso, hacerle referencia.



S3 buckets [Discover the console](#)

Search for buckets All access types

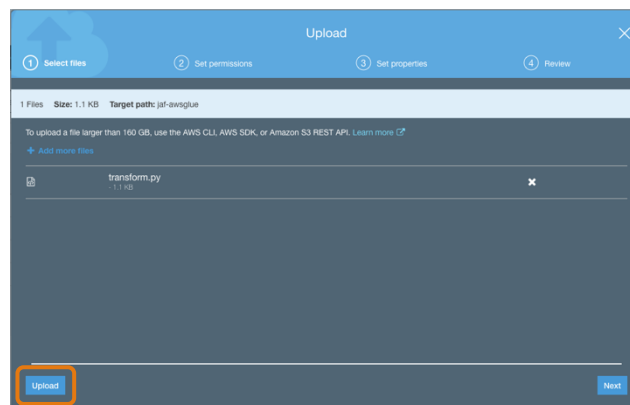
[+ Create bucket](#) [Edit public access settings](#) [Empty](#) [Delete](#)

12 Buckets 3 Regions

Bucket name	Access	Region	Date created
20190715telemedica	Objects can be public	EU (Ireland)	Jul 15, 2019 10:11:19 AM GMT+0200
aws-athena-query-results-846429553446-eu-west-1	Objects can be public	EU (Ireland)	May 29, 2019 1:56:50 PM GMT+0200
aws-athena-query-results-eu-west-1-846429553446	Objects can be public	EU (Ireland)	Jul 15, 2019 3:50:43 PM GMT+0200
aws-glue-notebooks-846429553446-eu-west-1	Objects can be public	EU (Ireland)	Jul 15, 2019 4:52:03 PM GMT+0200
aws-glue-scripts-846429553446-eu-west-1	Objects can be public	EU (Ireland)	Jul 15, 2019 3:34:41 PM GMT+0200
aws-glue-temporary-846429553446-eu-west-1	Objects can be public	EU (Ireland)	Jul 15, 2019 3:34:42 PM GMT+0200
ct-templates-1s5yk2fqaby42-eu-west-1	Objects can be public	EU (Ireland)	Jul 5, 2019 9:44:59 AM GMT+0200
cloudtrail-awslogs-846429553446-p6xqwgsh-isengard-do-not-delete	Objects can be public	US East (N. Virginia)	May 27, 2019 11:15:15 AM GMT+0200
do-not-delete-gatedgarden-audit-846429553446	Objects can be public	US West (Oregon)	May 27, 2019 11:30:08 AM GMT+0200
jaf-awsglue	Bucket and objects not public	EU (Ireland)	Jul 16, 2019 3:07:37 PM GMT+0200
rafafextest	Bucket and objects not public	EU (Ireland)	Jul 5, 2019 4:38:28 PM GMT+0200
rafalee.com	Objects can be public	EU (Ireland)	Jun 28, 2019 3:11:05 PM GMT+0200

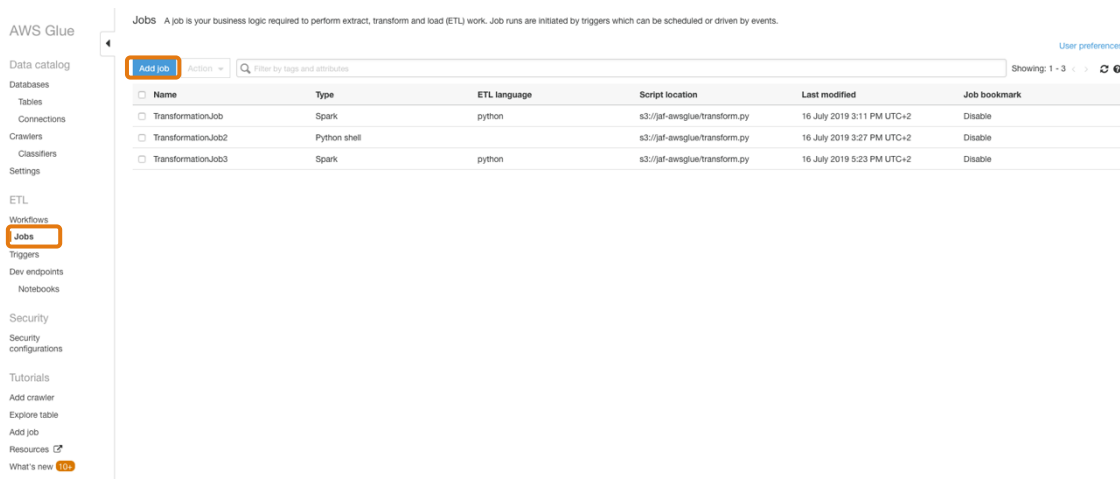
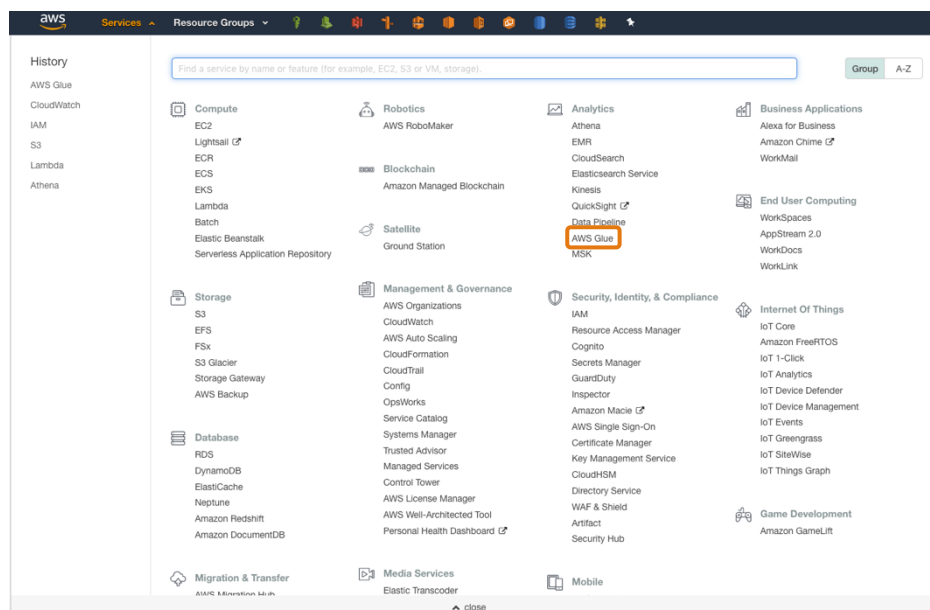
Q Type a prefix and press Enter to search. Press ESC to clear.

[Upload](#) [+ Create folder](#) [Download](#) [Actions](#)



Crear Job de transformación con AWS Glue

1. En la consola de AWS Glue, crear un nuevo job:



2. En el primer paso del wizard:

- Ponerle un nombre (**TransformationJob**)
- Asignarle el role creado previamente (**AWSGlueS3-Role**)
- Seleccionar el tipo **"Spark"**
- Seleccionar la segunda opción, la que dice que **le proporcionaremos el script**
- Como lenguaje ETL dejaremos la opción **"Python"**
- **Rutas de los scripts**

Configure the job properties

Name

TransformationJob

IAM role ⓘ

AWSGlueS3-Role

Ensure that this role has permission to your Amazon S3 sources, targets, temporary directory, scripts, and any libraries used by the job. [Create IAM role.](#)

Type

Spark

This job runs

☐ A proposed script generated by AWS Glue ⓘ

☒ An existing script that you provide

☐ A new script to be authored by you

ETL language

☒ Python ☐ Scala

S3 path where the script is stored

s3://jaf-awsglue/transform.py

Temporary directory ⓘ

s3://jaf-awsglue/temp

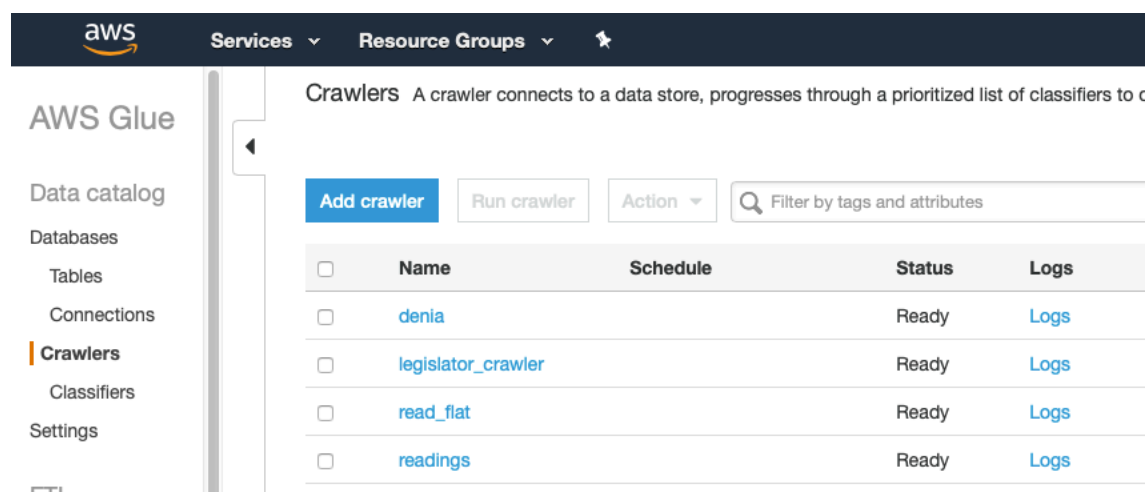
- ▶ Advanced properties
- ▶ Monitoring options
- ▶ Tags (optional)
- ▶ Security configuration, script libraries, and job parameters (optional)

3. No añadimos ninguna conexión en el segundo paso y guardamos el job.

4. A continuación, el wizard nos llevará a una ventana que nos permitirá editar el script del job en el caso que lo necesitémos. Basta con obviar esa ventana y cerrarla, no realizaremos ninguna modificación en ella

Creación de crawlers en AWS Glue

Ahora vamos a crear 2 crawlers en AWS Glue, uno para el output del flattening (o aplanado de la estructura) del fichero JSON, otro para los ficheros temporales del flattening extraídos que representaban los arrays.



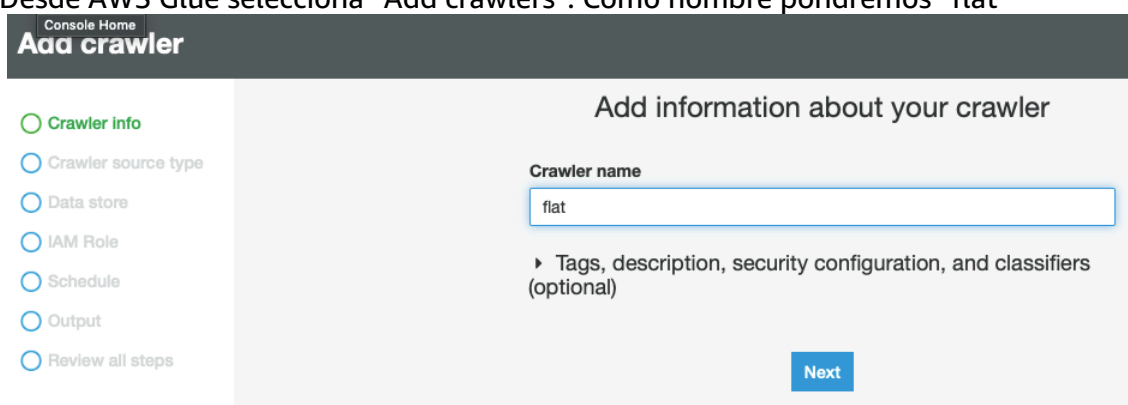
AWS Glue

Crawlers A crawler connects to a data store, progresses through a prioritized list of classifiers to c

[Add crawler](#) [Run crawler](#) [Action](#)

<input type="checkbox"/>	Name	Schedule	Status	Logs
<input type="checkbox"/>	denia		Ready	Logs
<input type="checkbox"/>	legislator_crawler		Ready	Logs
<input type="checkbox"/>	read_flat		Ready	Logs
<input type="checkbox"/>	readings		Ready	Logs

1. Desde AWS Glue selecciona “Add crawlers”. Como nombre pondremos “flat”



Add crawler

Add information about your crawler

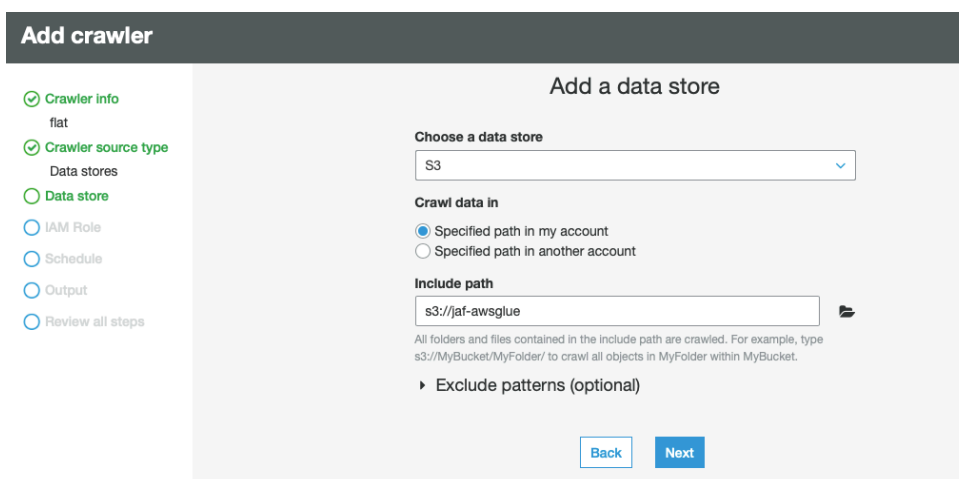
Crawler name

flat

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

[Next](#)

2. En Specify crawler source type deja Data stores y pulsa “Next”. En “Add a data store” deja “Specified path in my account”
3. Selecciona el path al bucket de S3 donde se encuentra la carpeta “output”, por ejemplo: “s3://<nombre de tu bucket>/output/”



Add crawler

Add a data store

Choose a data store

S3

Crawl data in

☒ Specified path in my account
☐ Specified path in another account

Include path

s3://jaf-awsglue

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.

► Exclude patterns (optional)

[Back](#) [Next](#)

4. Pulsa “Next” a continuación

Add crawler

☒ Crawler info

flat

☒ Crawler source type

Data stores

☐ Data store

S3: s3://jaf-awsglue

☐ IAM Role

Add another data store

☐ Yes

☒ No

Back

Next

5. Selecciona "Create an IAM role" y especifica el nombre "flat"

Add crawler

☒ Crawler info

flat

☒ Crawler source type

Data stores

☒ Data store

S3: s3://jaf-awsglue

☐ IAM Role

☐ Schedule

☐ Output

☐ Review all steps

Choose an IAM role

The IAM role allows the crawler to run and access your Amazon S3 data stores. [Learn more](#)

☐ Update a policy in an IAM role

☐ Choose an existing IAM role

☒ Create an IAM role

IAM role ⓘ

AWSGlueServiceRole-

To create an IAM role, you must have **CreateRole**, **CreatePolicy**, and **AttachRolePolicy** permissions.

Create an IAM role named **"AWSGlueServiceRole-rolename"** and attach the AWS managed policy, **AWSGlueServiceRole**, plus an inline policy that allows read access to:

- s3://jaf-awsglue

You can also create an IAM role on the [IAM console](#).

Back

Next

6. Selecciona "Run on demand"

Add crawler

☒ Crawler info

flat

☒ Crawler source type

Data stores

☒ Data store

S3: s3://jaf-awsglue

☒ IAM Role

☒ Schedule

☐ Output

☐ Review all steps

Create a schedule for this crawler

Frequency

Back

Next

7. Vamos a seleccionar la "base de datos" en el catálogo de metadatos para guardar las tablas resultantes. Presiona en el combo y busca la base de datos "fcc_db"

The screenshot shows the 'Add crawler' wizard in AWS Glue. The left sidebar lists the steps: Crawler info, Crawler source type, Data store, IAM Role, Schedule, Output, and Review all steps. The main panel is titled 'Configure the crawler's output'. It contains a 'Database' dropdown menu with 'fcc_db' selected, an 'Add database' button, a 'Prefix added to tables (optional)' text input field, and two expandable sections: 'Grouping behavior for S3 data (optional)' and 'Configuration options (optional)'. At the bottom are 'Back' and 'Next' buttons.

The screenshot shows the 'Add database' dialog box. It has a title bar with a close button (X). The 'Database name' field contains 'FCC_DB'. Below it is a 'Description and location (optional)' section. At the bottom is a 'Create' button.

8. A continuación haz click en “Next”, revisa todos los pasos y presiona “Finish”

Esto va a crear la tabla con la cual vamos a poder ejecutar los queries y obtener los datos de: type, la clave de readings, serialnumber, la clave de alarm, manufacturer, y volumeunit

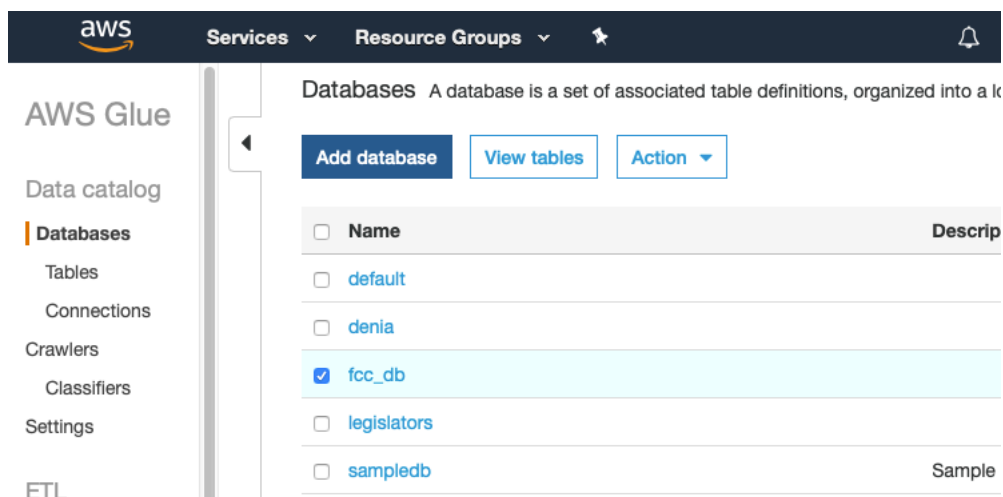
9. A continuación repetiremos los pasos desde el número 1, pero en el paso 4 vamos a usar la carpeta “tmp” en lugar de la carpeta “output”

Esto nos va a crear dos tablas, una para los datos de “alarm”, y otra para los datos de “readings”, que usaremos desde Amazon Athena para ejecutar un join en sql y juntar las tablas flat y de la tabla readings.

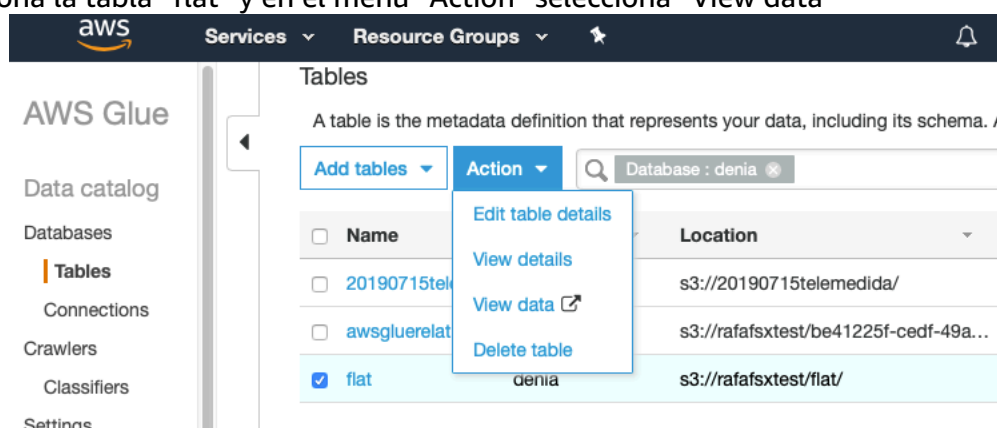
Ejecución de queries con Amazon Athena

Ahora vamos a ver los datos que encontró el crawler usando Amazon Athena:

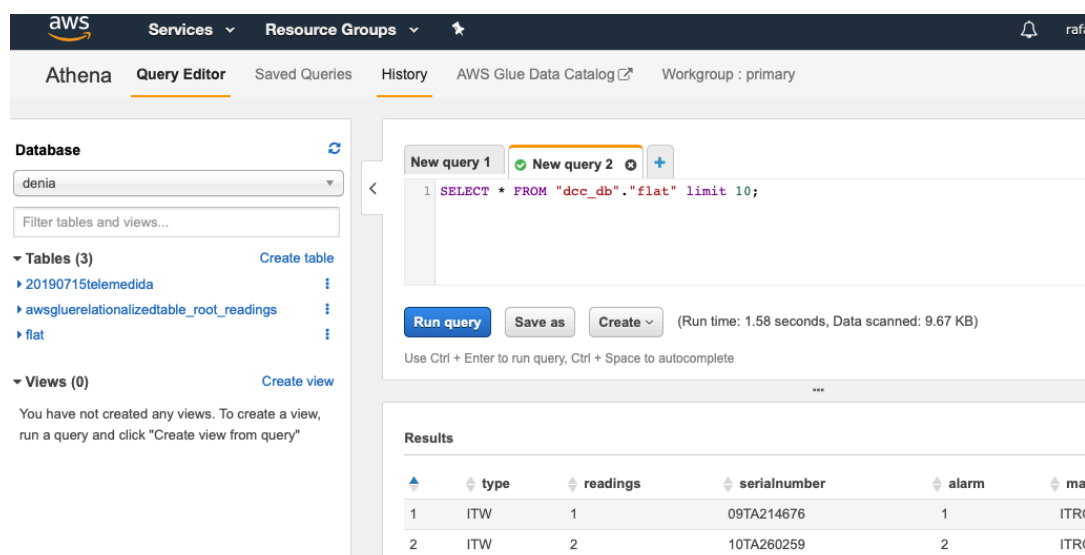
1. Desde AWS Glue selecciona en el menú de la izquierda “Databases” la base de datos “fcc_db”



2. Haz click en el botón "View tables"
3. Selecciona la tabla "flat" y en el menú "Action" selecciona "View data"



4. Esto te abrirá Amazon Athena y mostrará los datos en la tabla



A continuación hacemos un join para unir las dos tablas, y poder hacer queries de forma relacional de los datos extraídos del archivo original.

Por ejemplo (notar que los nombres de tus tablas podrían ser diferentes):

```
SELECT * FROM "dcc_db"."awsgluerelationalizedtable_root_readings" a, "dcc_db"."flat" b
where a.col0 = b.readings;
```

The screenshot shows the AWS Glue Query Editor interface. The top navigation bar includes 'Query Editor', 'Saved Queries', 'History', 'AWS Glue Data Catalog', 'Workgroup: primary', 'Settings', 'Tutorial', 'Help', and 'What's new'. The left sidebar shows a tree view with 'lamedida' and 'lionalizedtable_root_readings'. The main editor area displays a SQL query in 'New query 2':

```
1 SELECT * FROM "dcc_db"."awsgluerelationalizedtable_root_readings" a, "dcc_db"."flat" b where a.col0 = b.readings;
```

Below the query editor are buttons for 'Run query', 'Save as', 'Create', 'Format query', and 'Clear'. A status bar indicates '(Run time: 2.13 seconds, Data scanned: 1.24 MB)'. Below the query editor, a 'Results' section shows a table with 11 columns: col0, col1, col2, col3, type, readings, serialnumber, alarm, manufacturer, and volumeunit. The table contains 4 rows of data.

	col0	col1	col2	col3	type	readings	serialnumber	alarm	manufacturer	volumeunit
1	3	0	"01/05/2019 05:00:00"	997886	ITW	3	10TA260261	3	ITRON	L
2	3	1	"01/05/2019 04:00:00"	997886	ITW	3	10TA260261	3	ITRON	L
3	3	2	"01/05/2019 03:00:00"	997886	ITW	3	10TA260261	3	ITRON	L
4	3	3	"01/05/2019 02:00:00"	997886	ITW	3	10TA260261	3	ITRON	L