

# **PROCESO DE SUBIDA DE DATOS A UN DATA LAKE**

**GUIA PASO A PASO PARA SUBIR DATOS A DATALAKE EN AWS**

AUTOR: EDUARDO PADRON

# Índice

01

## INTRODUCCIÓN

Advertencia de Acciones tareas y servicios en AWS	02
Definiciones y servicios correspondientes	03
Diagrama de flujo de creación del Data Lake	04

02

## Setup del data lake

Creación del bucket	05
Creación de una database	08
Creación de una tabla	09
Creación de un role	12

03

## Creación de una función Lambda para crear la capa cruda

Función Lambda	15
Agregar variables de entorno en Lambda	20
Configuración Athena	24

01

# ADVERTENCIA DE ACCIONES TAREAS Y SERVICIOS EN AWS

## S3

Al generar un nuevo flujo, hacer pruebas en el bucket de nombre: sensorsdatav1. Y seguir nomenclatura así como la arquitectura de carpetas.



## LAMBDA

Al generar una nueva función revisar el rol y la configuración de memoria CPU así como de tiempo máximo 5 minutos y máximo de CPU 512 en caso de ser archivos grandes consultar.



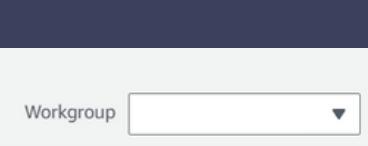
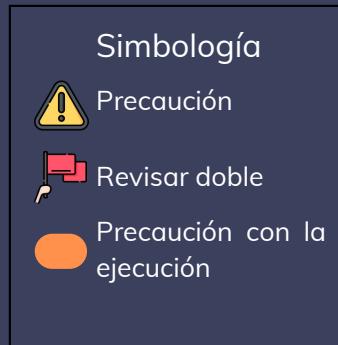
## GLUE DATA CATALOG

Al crear una tabla nueva para algún flujo en la base de datos probar primero en la base de datos de prueba antes de poner en producción así como seguir la nomenclatura st1-nombre y revisar los tipos de datos por campos (Columnas) para que no existan errores en caso de error cambiar tipos hasta encontrar el que corresponde.



## ATHENA

Revisar que el workgroup y la ruta de salida de las consultas antes de ejecutar queries



# DEFINICIONES Y SERVICIOS CORRESPONDIENTES

## DEFINICIONES

## SERVICIOS

### ST0 O FUENTE

Espacio de memoria o carpeta en datalake (S3) donde se suben o almacenan los archivos Excel o csv a procesar para la creación de capa cruda



### S3

### ST1 O CAPA CRUDA

Espacio de memoria o carpeta en datalake (S3) y Glue data catalog database donde se almacenan los archivos parquet procesados en Lambda para capa limpia y que seran procesados por un ETL Glue Job



### AWS LAMBDA

### ST2 O CAPA LIMPIA

Espacio de memoria o carpeta en datalake (S3) y Glue data catalog database donde se almacenan los archivos parquet que seran procesados por un ETL para capa semántica



### AWS GLUE JOBS

### AWS ATHENA

Servicio para realizar consultas SQL sobre las bases de datos en Glue data catalog databases. Nos ayuda a verificar los datos y que las tablas y bases de datos han sido creadas de manera exitosa

# DATA LAKE WORKFLOW PROCESS

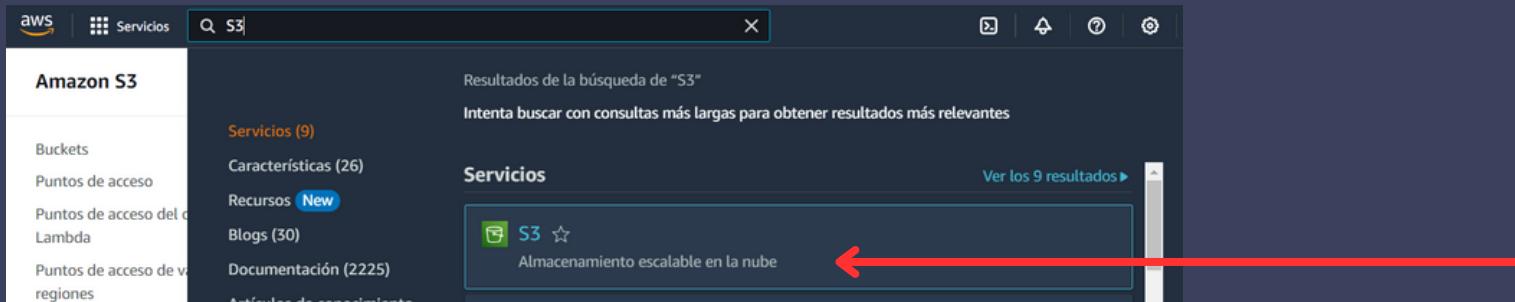


# SETUP DEL DATA LAKE

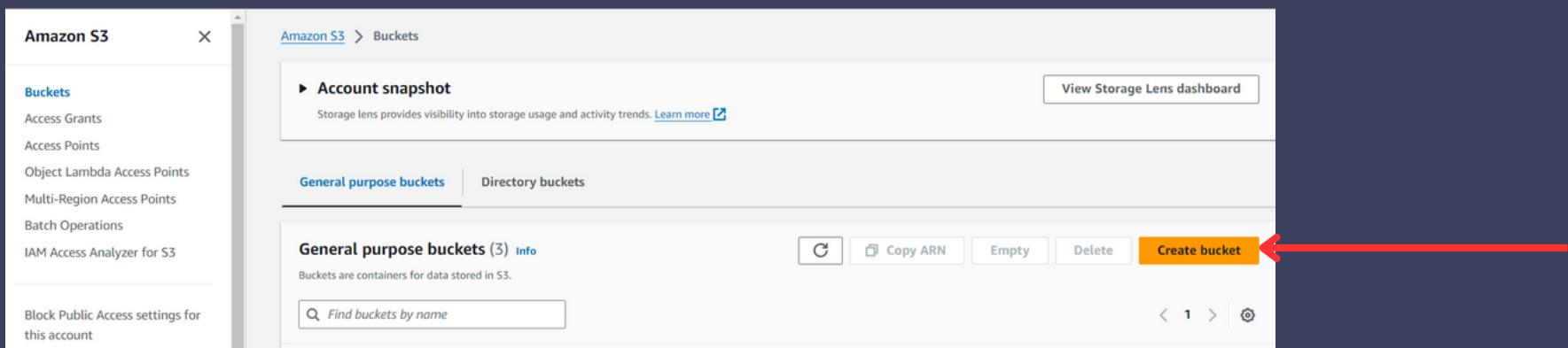
## CREACION DEL BUCKET

Antes de crear cualquier herramienta en AWS debemos tener en cuenta, el numero de pasos, buckets, databases y servicios que vamos a utilizar en base a eso como primer paso se crean los buckets, databases y los roles para cada herramienta AWS, para otorgar permisos y que se puedan ejecutar de manera exitosa. Por lo que comenzaremos con:

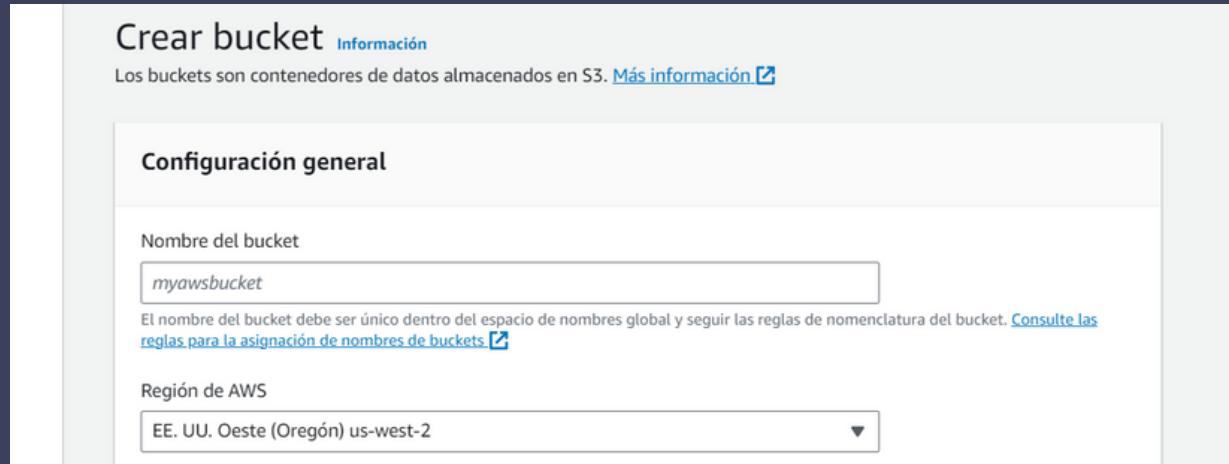
El primer paso es crear los bucket que permite almacenar archivos, por lo que debemos de entrar a la URL de la consola. Una vez dentro buscaremos S3 y seleccionaremos la opción mostrada en la imagen.



Dentro de Amazon S3 seleccionamos bucket en el panel izquierdo y podremos ver la opción para crear un bucket. Daremos clic.



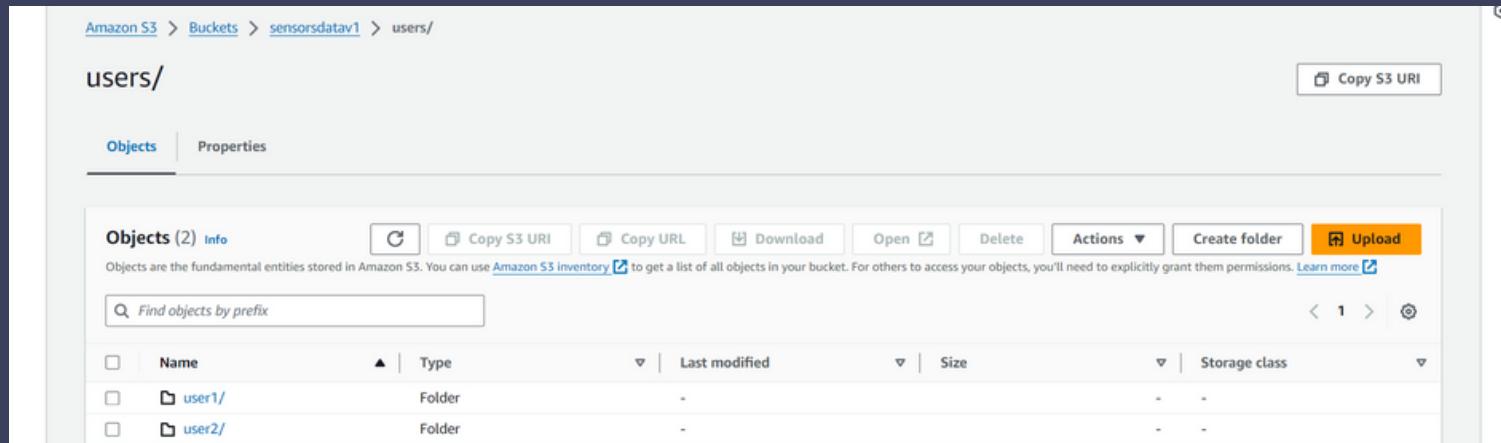
Al crear un bucket para este proceso no tenemos que especificar ningún cambio o configuración especial solo necesitamos el nombre y asegurarnos de que la región coincida con la usada por la cuenta en este caso es us-west-2, de preferencia el bucket debe contener aws-(servicio a usar)-(ID) es recomendable que después de aws pongamos el tipo de servicio usado y después un identificador.



Ahora necesitamos crear una estructura de carpetas dependiendo de los archivos es recomendable tener una carpeta por archivo diferente, con el motivo de que posteriormente se leerán todos los archivos dentro de la carpeta. En caso de no estar en carpeta no se podrá leer.

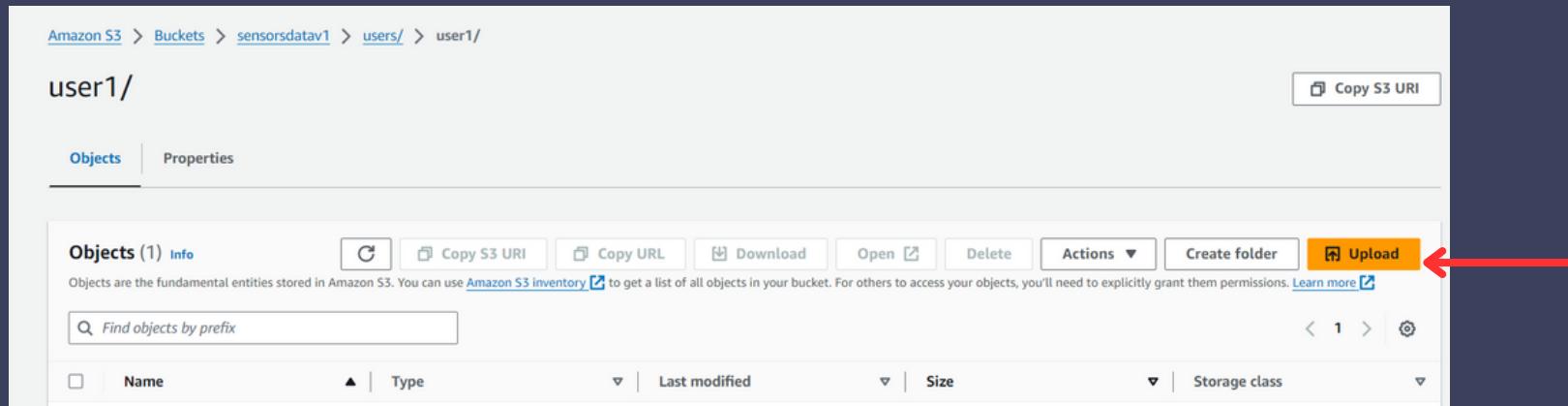
The screenshot shows the 'Amazon S3' console. On the left is the navigation pane with 'Buckets' selected. The main area shows the 'sensorsdatav1' bucket. The 'Objects' tab is active, displaying 'Objects (1)'. Below the toolbar, there is a search bar and a table header with columns: Name, Type, Last modified, Size, and Storage class. A red arrow points to the 'Create folder' button in the toolbar.

Dependiendo del caso, lo mejor es tener una carpeta por usuario en este ejemplo uso dos usuarios, por lo que se crearon dos carpetas, una de ellas contendrá nuestro archivo con datos para usuario 1 y la otra el usuario 2 en otra ubicación.



The screenshot shows the Amazon S3 console interface. The path in the top navigation bar is: Amazon S3 > Buckets > sensorsdatav1 > users/. Below the path, the folder name "users/" is displayed. On the right, there is a "Copy S3 URI" button. Underneath the path, there are tabs for "Objects" and "Properties", with "Objects" being the active tab. The "Objects" section shows two items: "user1/" and "user2/", both listed as "Folder". Above the object list is a toolbar with various actions: Copy S3 URI, Copy URL, Download, Open, Delete, Actions (with a dropdown arrow), Create folder, and Upload. The "Upload" button is highlighted with a yellow background. Below the toolbar is a search bar labeled "Find objects by prefix".

Finalmente damos clic en Cargar para subir el archivo en la carpeta correspondiente desde nuestro computador.

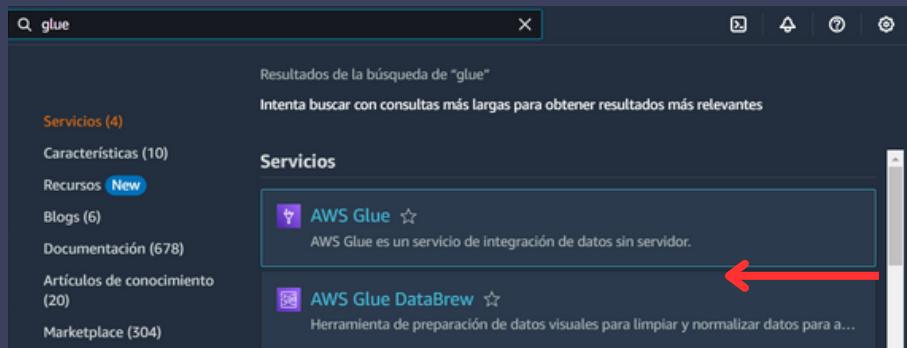


The screenshot shows the Amazon S3 console interface. The path in the top navigation bar is: Amazon S3 > Buckets > sensorsdatav1 > users/ > user1/. Below the path, the folder name "user1/" is displayed. On the right, there is a "Copy S3 URI" button. Underneath the path, there are tabs for "Objects" and "Properties", with "Objects" being the active tab. The "Objects" section shows one item: "user1/", listed as "Folder". Above the object list is a toolbar with various actions: Copy S3 URI, Copy URL, Download, Open, Delete, Actions (with a dropdown arrow), Create folder, and Upload. The "Upload" button is highlighted with a yellow background and has a red arrow pointing to it. Below the toolbar is a search bar labeled "Find objects by prefix".

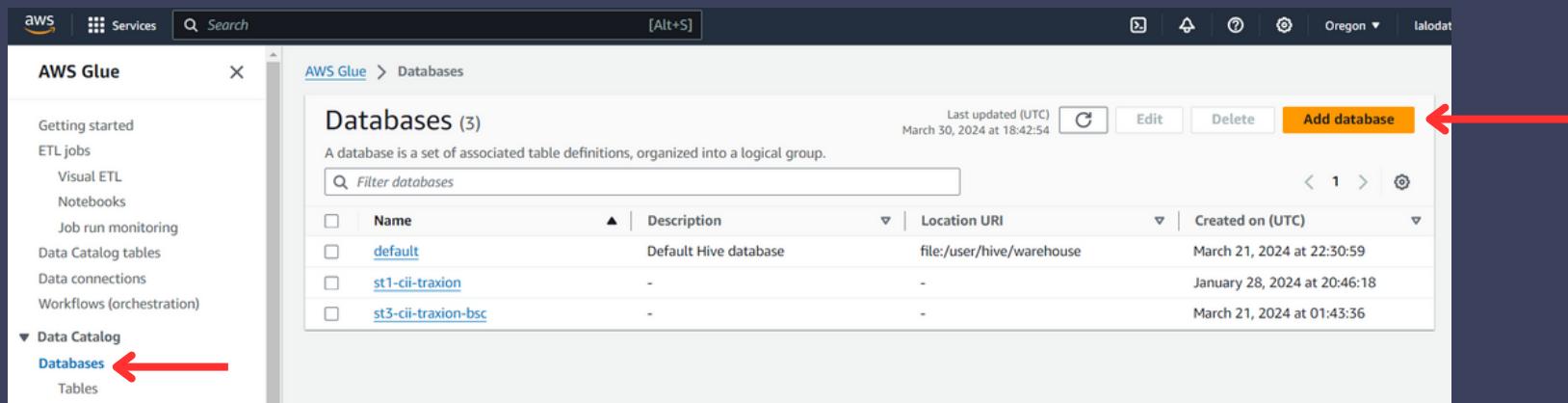
## CREACION DE UN DATABASE EN AWS GLUE CATALOG

Dependiendo de el tipo de servicio que utilices necesitara o no un database en AWS Glue que nos permitirá crear tablas que contendrán información. Asegúrate de crear una base de datos o las necesarias para la tarea a realizar.

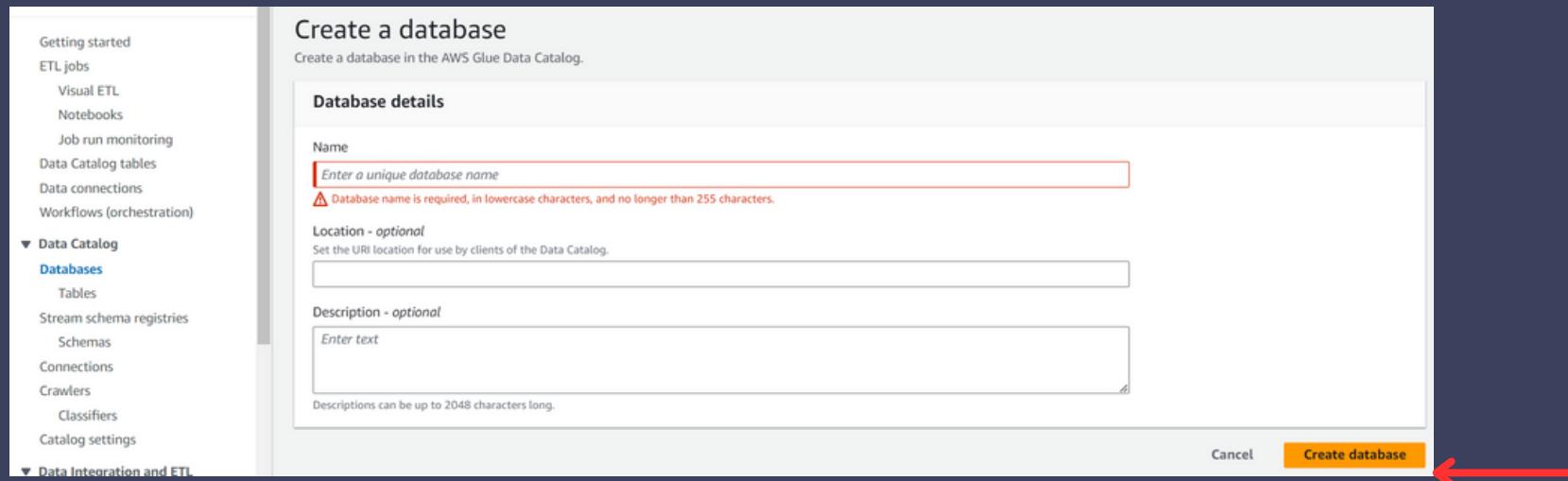
Ahora tenemos que buscar AWS Glue para entrar a la consola.



Una vez en la consola damos clic en la barra izquierda a la opción Databases, entonces nos podrá dar la opción de Add database.



Solo tenemos que agregar el nombre, de preferencia como con los bucket usar el formato aws-servicio-ID



## CREACION DE UNA TABLA

Para tener los datos en una estructura necesitamos una tabla, que es un conjunto estructurado de datos organizados en filas y columnas, donde cada fila representa un registro y cada columna representa un atributo o campo específico. facilitando la organización eficiente y la recuperación de información dentro de un sistema de base de datos.

The screenshot shows the 'sensorsdata1' database page in the AWS Glue Data Catalog. At the top, it shows the database name 'sensorsdata1' and its last update time 'March 30, 2024 at 18:45:56'. There are 'Edit' and 'Delete' buttons. Below this is the 'Database properties' section, which lists the database name 'sensorsdata1', an empty 'Description' field, an empty 'Location' field, and a 'Created on (UTC)' timestamp 'March 30, 2024 at 18:45:54'. The main area is titled 'Tables (0)' and includes a 'Filter tables' search bar and a table header with columns: Name, Database, Location, Classification, Deprecated, View data, and Data quality. A note below the table says 'No available tables'. At the top right of this section are 'Last updated (UTC)', a 'C' icon, 'Delete', 'Add tables using crawler', and a 'Add table' button, which is highlighted by a red arrow.

Tenemos que agregar el nombre, se debe usar un nombre representativo, descripción y seleccionar una database correspondiente.

AWS Glue > Tables > Add table

Step 1  
Set table properties

Step 2  
Choose or define schema

Step 3  
Review and create

### Set table properties

#### Table details

Name  Enter a unique name

If you plan to access the table from Amazon Athena, then the name should be under 256 characters and contain only lowercase letters (a-z), numbers (0-9), and underscore (\_). For more information, see [Athena names](#).

Database

Description - optional  Enter a description

Descriptions can be up to 2048 characters long.

#### Table format

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

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

Apache Iceberg table - New  
Create an Iceberg table that supports automatic data compaction.

## SELECCIONAR FORMATO Y ORIGEN DE DATOS

Tenemos que seleccionar de que servicio, el path y el tipo de formato de dato a leer. Despues dar next

Select the type of source

S3

Kinesis

Kafka

Data location is specified in

my account

another account

Include path

s3://bucket/prefix/

⚠ This is a required field.

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

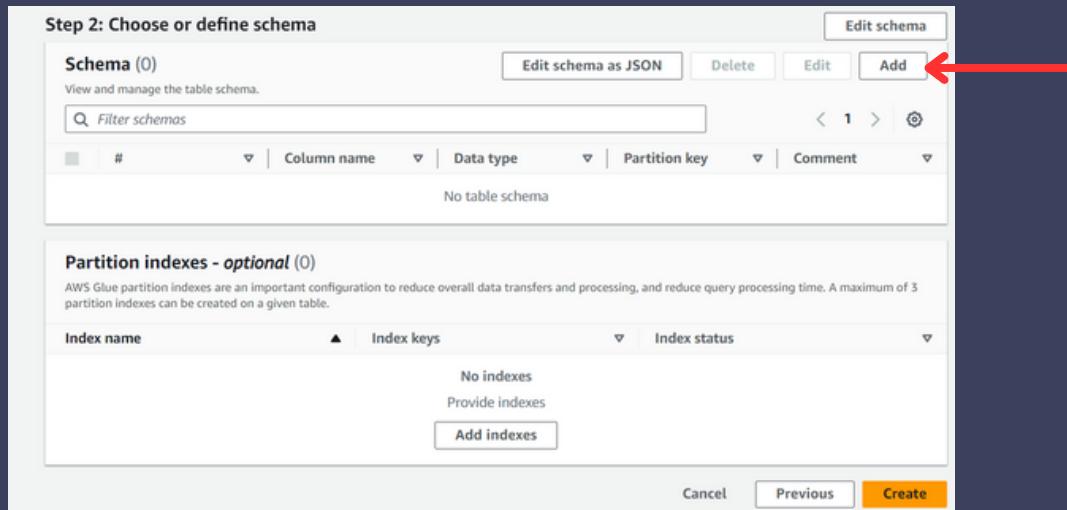
CSV

JSON

Parquet

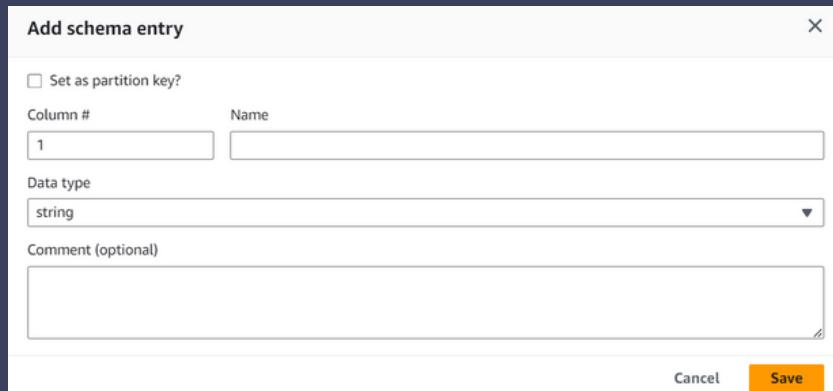
XML

El Schema es donde agregaremos los campos o columnas de nuestra db, tenemos que recordar que debe coincidir con el tipo de datos y columnas de nuestro archivo Parquet. Damos click en Add.



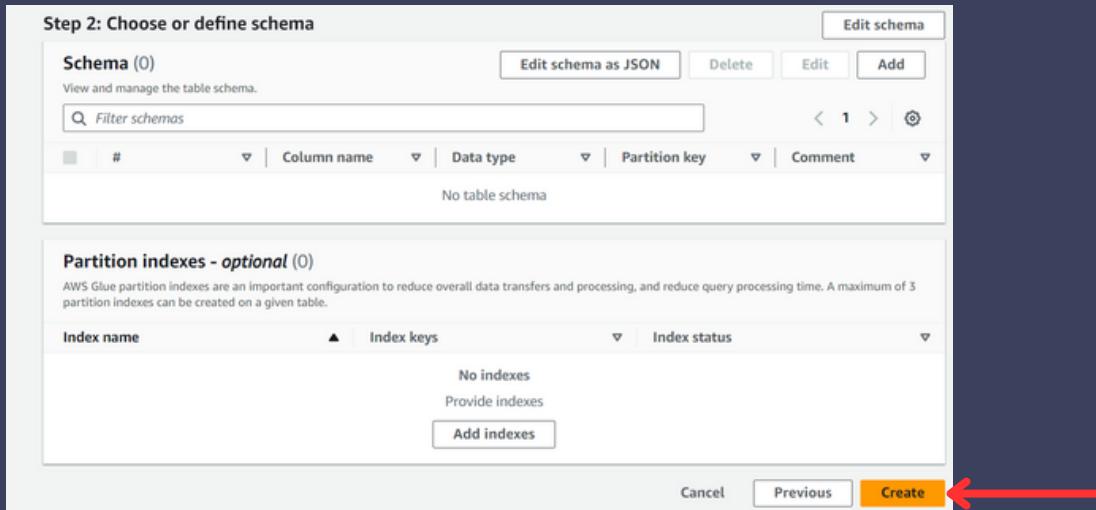
The screenshot shows the 'Step 2: Choose or define schema' interface. At the top right, there are buttons for 'Edit schema', 'Edit schema as JSON', 'Delete', 'Edit', and 'Add'. A red arrow points to the 'Add' button. Below these buttons is a search bar labeled 'Filter schemas'. Underneath is a table header with columns: #, Column name, Data type, Partition key, and Comment. The table body displays 'No table schema'. Below the table, there is a section titled 'Partition indexes - optional (0)' with a note about partition indexes. It contains a table with columns: Index name, Index keys, and Index status. The table body shows 'No indexes' and a 'Provide indexes' link. At the bottom of the screen are 'Cancel', 'Previous', and 'Create' buttons, with 'Create' being orange.

El nombre tiene que coincidir con el del archivo de origen, es importante definir nombres en columnas para mantener un estándar, y el tipo de dato si no lo que pasara es que marque un error por que no coincide el tipo de dato en tabla con el de la fuente de datos. Después de tener esto dar click en next.



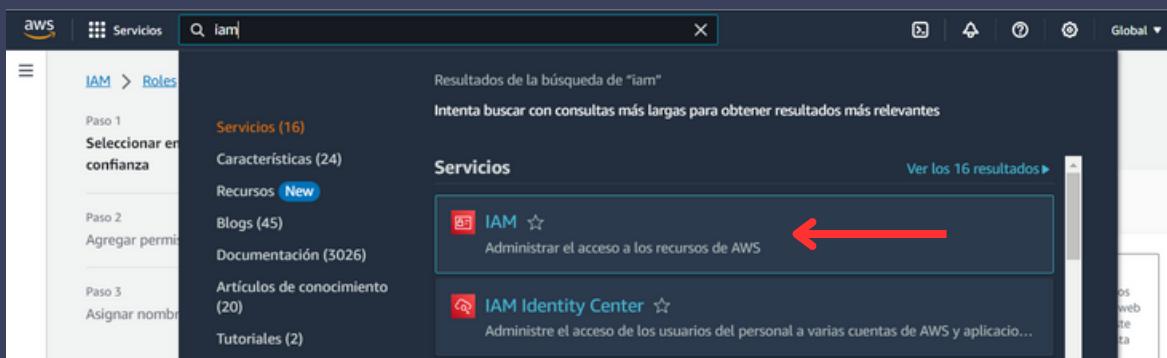
The screenshot shows the 'Add schema entry' dialog box. It has fields for 'Column #' (set to 1), 'Name' (empty), 'Data type' (set to 'string'), and a 'Comment (optional)' text area. There is also a checkbox for 'Set as partition key?'. At the bottom are 'Cancel' and 'Save' buttons, with 'Save' being orange.

Como paso final verificamos que todo este configurado de manera correcta



## CREACION DE UN ROLE

Dependiendo de el tipo de servicio que utilices necesitará un role que es un grupo de permisos para acceder a fuentes de información en AWS que nos permitirá ejecutar las herramientas creadas. Asegúrate de crear un rol y asignar permisos necesarios para la tarea a realizar. Ahora tenemos que buscar IAM para entrar a la consola y crear un role para asignar permisos.



Una vez en la consola damos clic en la barra izquierda en roles para que nos aparezca el menú y seleccionar crear rol.

The screenshot shows the AWS Identity and Access Management (IAM) service. On the left, there's a sidebar with 'Identity and Access Management (IAM)' at the top, followed by 'Search IAM', 'Dashboard', 'Access management' (which is expanded to show 'User groups' and 'Users'), and 'Roles' (which is also highlighted with a red arrow). The main area is titled 'Roles (7) Info' and contains a table with seven rows. The first row is collapsed. The second row shows 'Role name: AWSServiceRoleForTrustedAdvisor' with 'Trusted entities: AWS Service: support (aws-support-service)' and 'Last activity: 2023-09-12'. The third row shows 'Role name: Glue\_Full' with 'Trusted entities: AWS Service: glue' and 'Last activity: 2023-09-12'. The fourth row shows 'Role name: psycoprueba-role-x253rws7' with 'Trusted entities: AWS Service: lambda' and 'Last activity: 2023-09-12'. At the top right of the table, there are 'Delete' and 'Create role' buttons, with the 'Create role' button highlighted with a red arrow.

Dejamos seleccionado Servicio de AWS, buscamos en la opción Caso de uso la palabra Lambda y la seleccionamos como se muestra en la imagen, y damos clic en siguiente.

This screenshot shows the 'Step 1 Select trusted entity' screen of the IAM Role creation wizard. On the left, there are three steps: 'Step 1 Select trusted entity' (which is active), 'Step 2 Add permissions', and 'Step 3 Name, review, and create'. The main area has a search bar labeled 'Filter service or use case' and a list of services under 'Commonly used services'. The 'Lambda' service is selected, indicated by a blue checkmark and a dropdown arrow. To the right of the service list, there's a tooltip for 'Web identity' which says: 'Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.' Below the service list, there's another list of services under 'Other services' including Amazon EMR Serverless, Amazon OpenSearch Service, Amazon Grafana, Amplify, API Gateway, AppFabric, Application Auto Scaling, Application Discovery Service, Application Migration Service, AppStream 2.0, and AppSync. At the bottom, there's a section for choosing a use case for the specified service, with 'Use case' and 'Lambda' selected. A red arrow points to the 'Lambda' service in the list, and another red arrow points to the 'Lambda' use case selection.

Una vez en agregar permisos necesitamos 2 permisos para Admin y uno es AmazonS3FullAccess para acceder al bucket.

Screenshot of the AWS IAM 'Add permissions' step. The search bar shows 's3'. The results table has columns for Policy name, Type, and Description. The 'AmazonS3FullAccess' policy is selected (indicated by a checked checkbox) and highlighted with a red arrow.

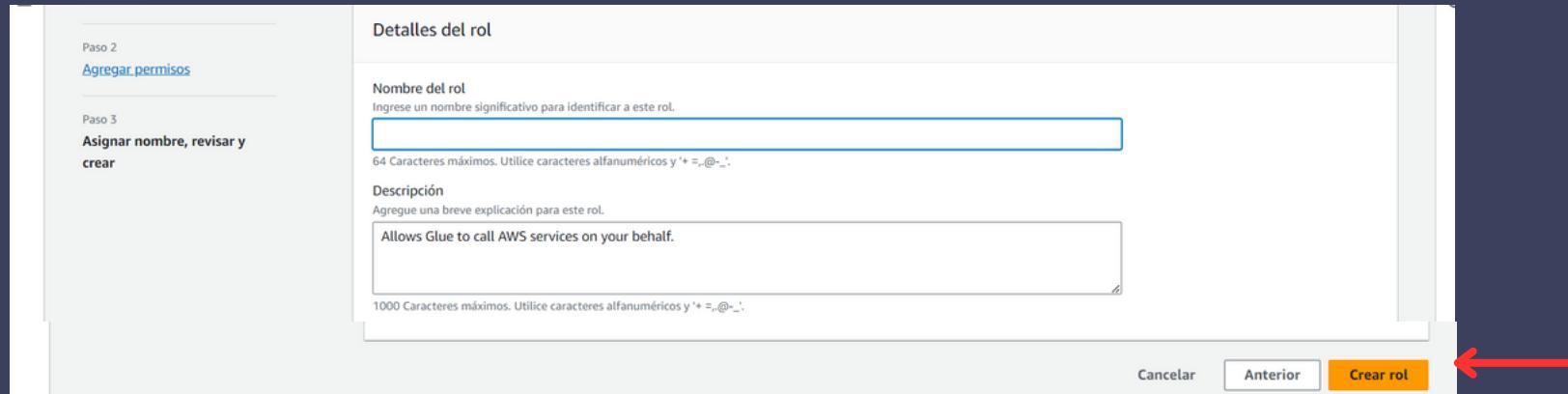
Policy name	Type	Description
AmazonDMSRedshiftS3Role	AWS managed	
<b>AmazonS3FullAccess</b>	AWS managed	
AmazonS3ObjectLambdaExecutionRole...	AWS managed	

Buscamos la palabra Lambda y seleccionamos AWSLambdaFullAccess, Administrator access para nuestro Lambda , ahora terminando de seleccionar los 3 damos en siguiente en la parte del fondo del menú

The image contains two screenshots of the AWS IAM 'Add permissions' step, showing the selection of three policies:

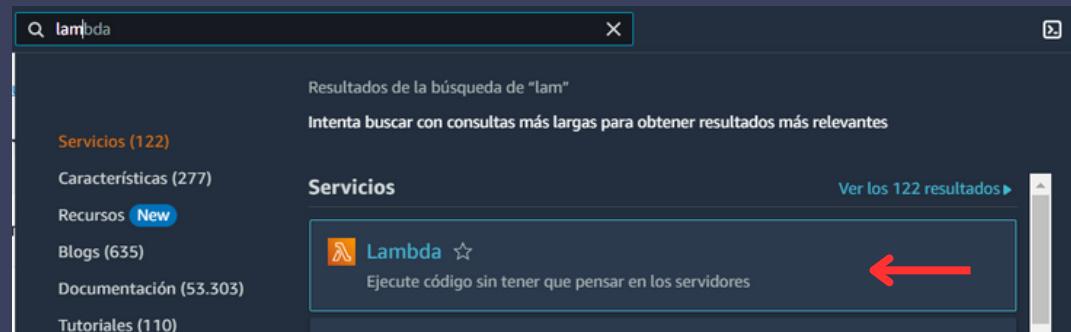
- Top Screenshot:** The search bar shows 'lambda'. The results table has columns for Policy name, Type, and Description. The 'AWSLambda\_FullAccess' policy is selected (indicated by a checked checkbox) and highlighted with a red arrow.
- Bottom Screenshot:** The search bar shows 'AdministratorAccess'. The results table has columns for Policy name, Type, and Description. The 'AdministratorAccess' policy is selected (indicated by a checked checkbox) and highlighted with a red arrow.

Ahora necesitamos seleccionar el nombre, siguiendo el estándar, aws-servicio-ID y en caso opcional puedes agregar una descripción para que es el permiso una vez agregado el nombre no necesitamos hacer ningún cambio podemos dar en Crear rol.

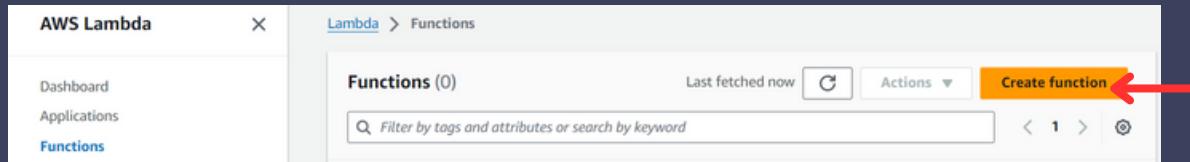


## CREACION DE UNA FUNCION LAMBDA PARA CREAR LA CAPA CRUDA

AWS Lambda permite ejecutar código sin la necesidad de gestionar servidores y ser activado por varios servicios de AWS o eventos. Usaremos Lambda para crear un solo archivo que contenga la información que necesitamos así como pasar a un formato aceptado por AWS Glue como pueden ser csv, xlxs, json, parquet entre otros. Esto será la capa cruda que nos permite almacenar archivos que estan listos para ETL y aplicar las reglas de negocio que necesitemos.



Una vez en la consola de Lambda seleccionamos en el panel izquierdo Funciones y posteriormente cuando aparezca Crear una función.



Para crear una función desde cero, tenemos que darle un nombre, para seguir el estándar podemos usar aws-servicio-ID, después tenemos que seleccionar el tiempo de ejecución que es el lenguaje y versión que se usara. Para evitar problemas de compatibilidad de versiones usa 3.8, mas adelante se añadirá la configuración para otras versiones. Faltaría seleccionar la arquitectura que es recomendable usar x86\_64. Recordemos que se ejecuta en Linux.

Lambda > Funciones > Crear una función

**Crear una función** Información

Las aplicaciones de Repertorio de aplicaciones sin servidor de AWS se han trasladado a [Crear una aplicación](#).

**Crear desde cero**  
Empiece con un sencillo ejemplo "Hello World".

**Utilizar un proyecto**  
Cree una aplicación Lambda utilizando un código de muestra y los ajustes de configuración predefinidos de casos de uso comunes.

**Imagen del contenedor**  
Seleccione una imagen de contenedor para implementar para la función.

**Información básica**

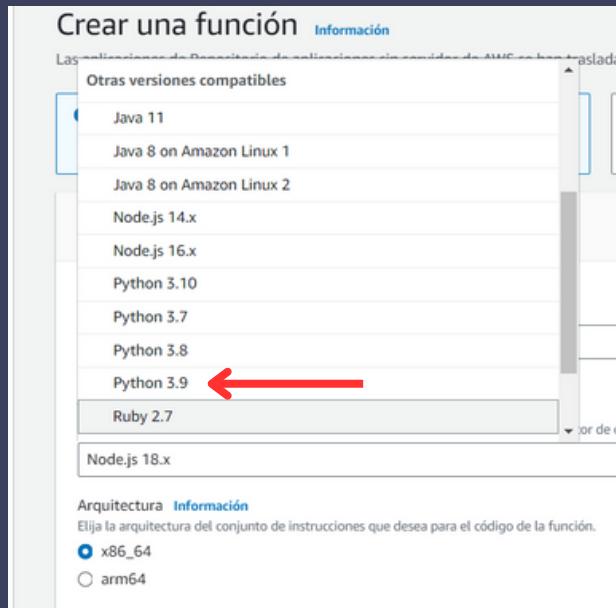
Nombre de la función  
Escriba un nombre para describir el propósito de la función.

Utilice exclusivamente letras, números, guiones o guiones bajos. No incluya espacios.

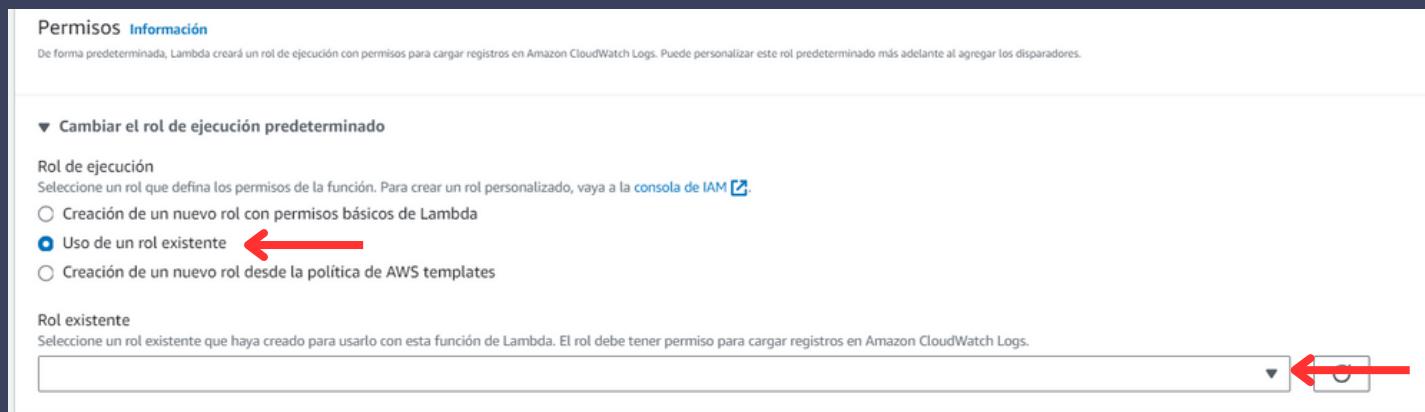
Tiempo de ejecución Información  
Elija el lenguaje que desea utilizar para escribir la función. Tenga en cuenta que el editor de código de la consola solo admite Node.js, Python y Ruby.  
 

Arquitectura Información  
Elija la arquitectura del conjunto de instrucciones que desea para el código de la función.  
 **x86\_64**  
 arm64

Selecciona la versión de Python 3.8 que prefieras



Selecciona en permisos el rol que creamos para Lambda que solo usa S3FullAccess de nombre y en caso de no existir se crea uno.



Una vez en el menú de la función creada damos scroll hacia la parte inferior y podremos ver el menú de capas.

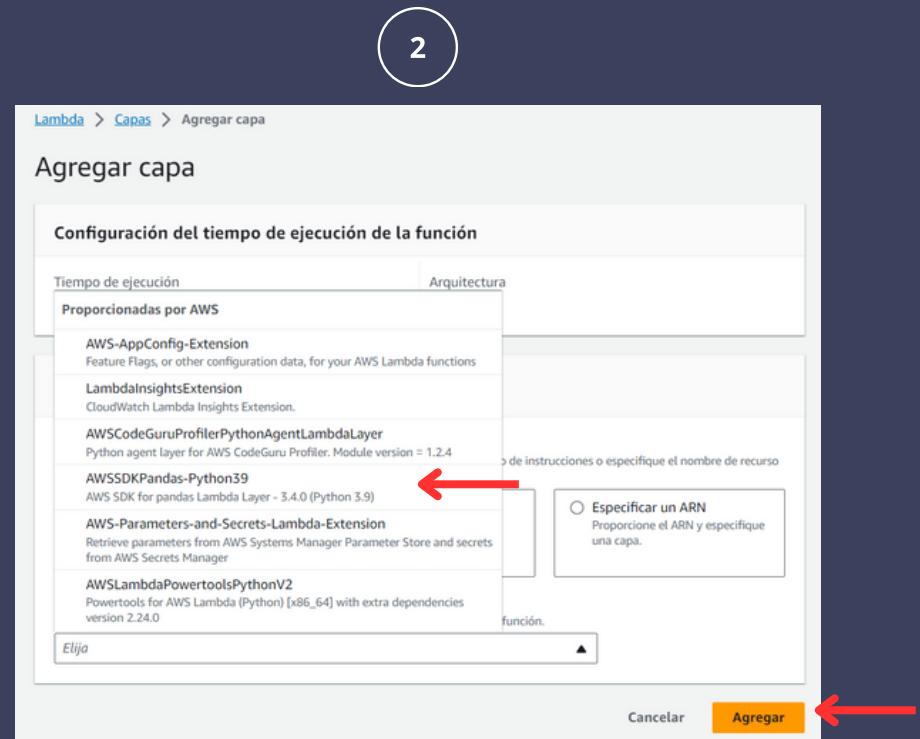
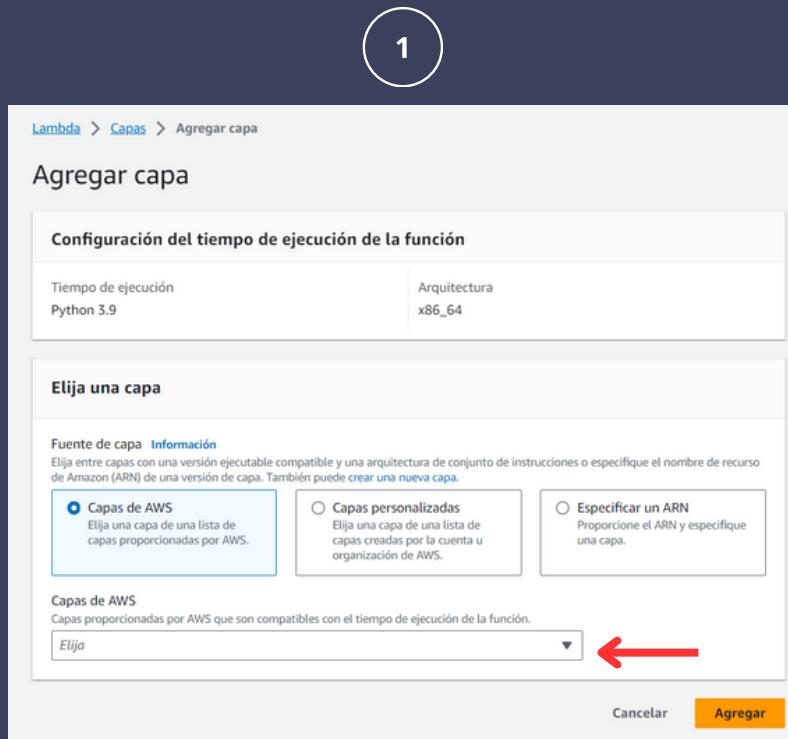
The screenshot shows the AWS Lambda Function Overview page for a function named 'sensordatav1'. The 'Layers' section indicates 0 layers. There are buttons for adding triggers and destinations. A sidebar on the right provides details like 'Last modified' (31 seconds ago) and 'Function ARN'. The 'Code source' tab is active, showing an 'Upload from' button. Navigation tabs at the bottom include 'Code', 'Test', 'Monitor', 'Configuration', 'Aliases', and 'Versions'.

Daremos clic en Añadir una capa. Las capas nos permiten usar librerías que necesitamos, algunas ya están disponibles en AWS.

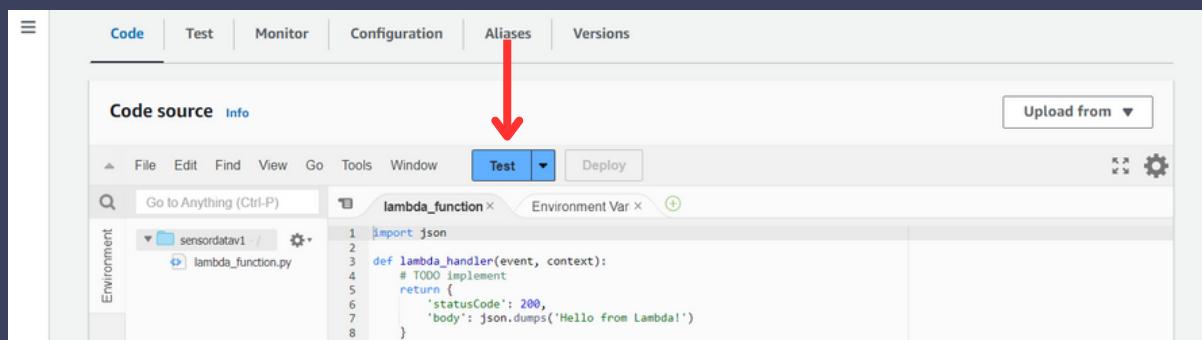
The screenshot shows the 'Layers' section in the AWS Lambda function configuration. It displays a single layer entry: 'AWSSDKPandas-Python38' (Version 19, Compatible runtime: python3.8, Compatible architecture: x86\_64). An 'Edit' button is also visible. A red arrow points to the 'Add a layer' button, which is highlighted in the interface.

Merge order	Name	Layer version	Compatible runtimes	Compatible architectures	Version ARN
1	AWSSDKPandas-Python38	19	python3.8	x86_64	

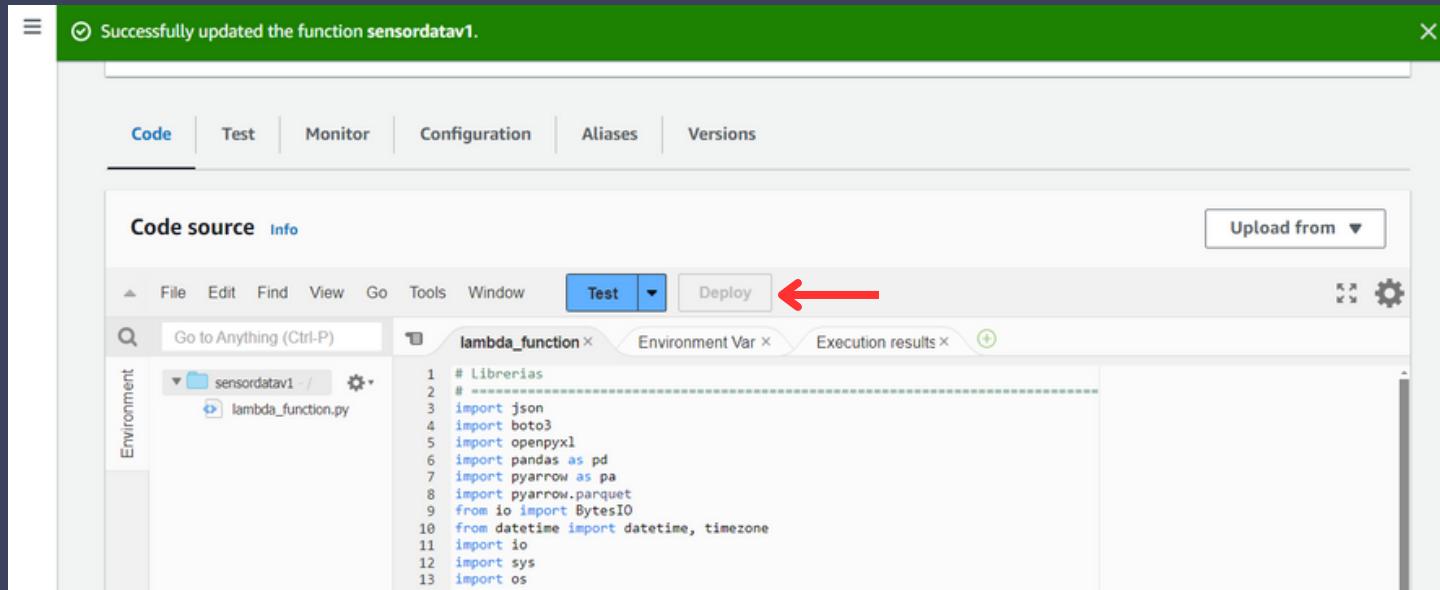
Seleccionamos Capas de AWS. Elegiremos la que necesitamos en este caso Pandas 3.8 Una vez seleccionada damos clic en Agregar.



Ahora ir a la pestaña de Código Fuente. Agregamos el código que queremos ejecutar. Guardamos y creamos un evento de prueba.

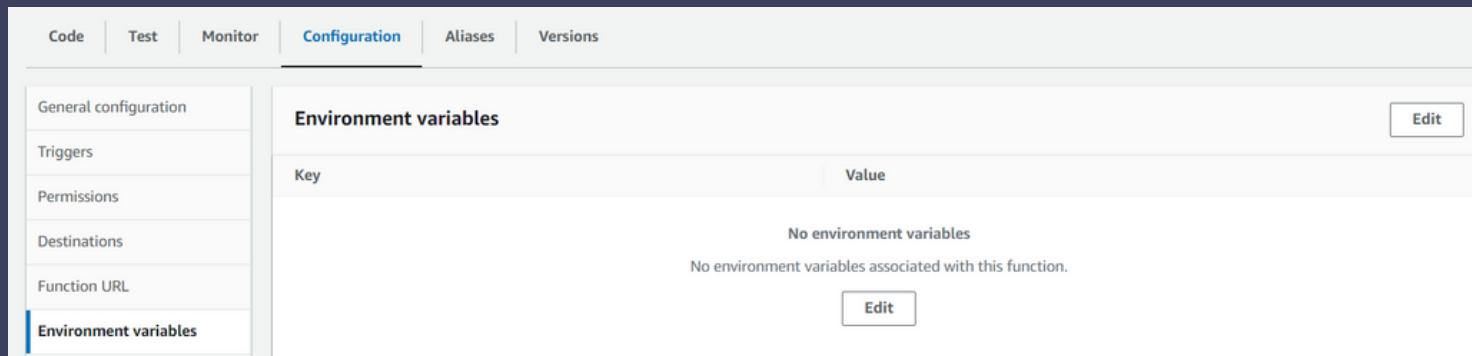


Una vez seleccionada la configuracion correcta podemos ingresar codigo para procesar nuestro csv con pandas.



## AGREGAR VARIABLES DE ENTORNO EN LAMBDA

Ahora en configuration seleccionamos la opcion Environment variables. Dar clic en Edit



Dar clic en Add enviroment variable para ir agregando una por cada variable que necesitemos

Lambda > Functions > preformateador-isk-cii-prod > Edit environment variables

## Edit environment variables

### Environment variables

You can define environment variables as key-value pairs that are accessible from your function code. These are useful to store configuration settings without the need to change function code. [Learn more](#)

There are no environment variables on this function.

Add environment variable

▶ Encryption configuration

Cancel Save

Utiliza los mismos nombres de variables, para evitar modificaciones en la extracción de sus valores y la modificación de el código de uso.

Ahora ir a la pestaña de Código Fuente. Agregamos el código que queremos ejecutar. Guardamos y creamos un evento de prueba.

Código Probar Monitorear Configuración Alias Versiones

### Código fuente

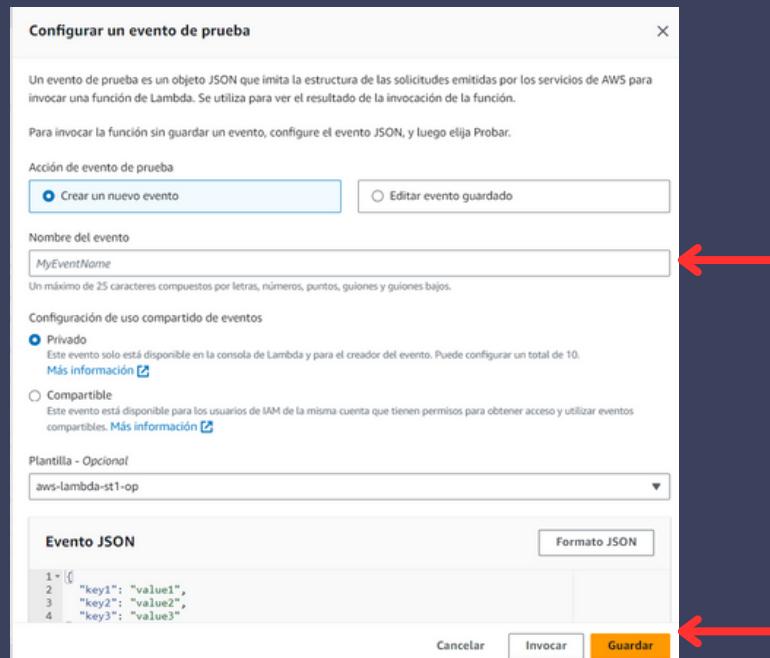
Información

File Edit Find View Go Tools Window Test Deploy Cargar desde

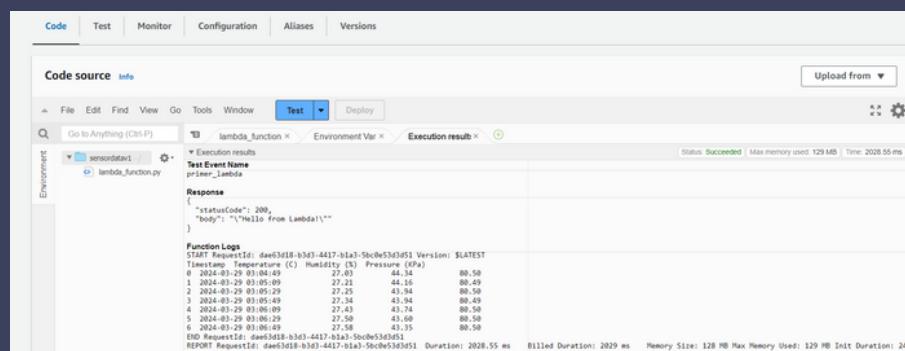
lambda\_function Environment

```
1 import boto3
2 import pandas as pd
3 from io import BytesIO
4 import openpyxl
5
6 source_bucket = 'aws-lambda-st0-op' # Replace with your source bucket name
7 destination_bucket = 'aws-glue-st0-op' # Replace with your destination bucket name
8
9 def lambda_handler(event, context):
10     # List of file keys to process
11     file_keys = [
```

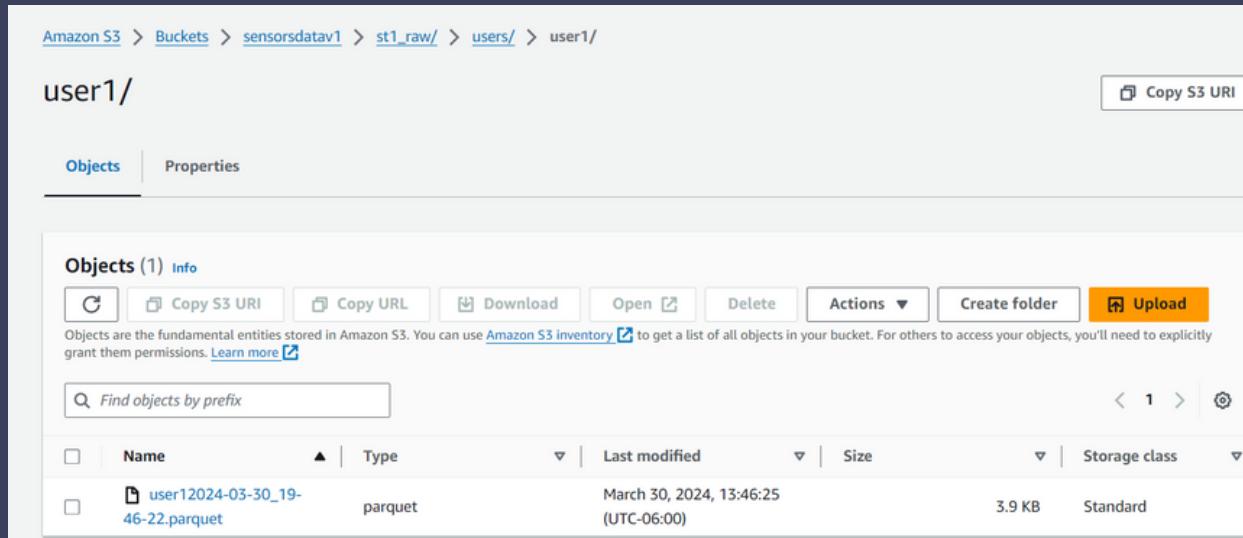
Solamente necesitamos agregar un nombre al evento y guardar. Después ponemos hacer un deploy al código y finalmente test. En caso de errores saldrá en la pestaña de Execution results que sería como la terminal(solo funciona como output de mensajes). Terminando de ejecutar encontraremos nuestros nuevos archivos CSV generados en un bucket de destino.



Una vez ejecutado podremos ver dos cosas un print y un mensaje de que se ejecuto de manera exitosa.

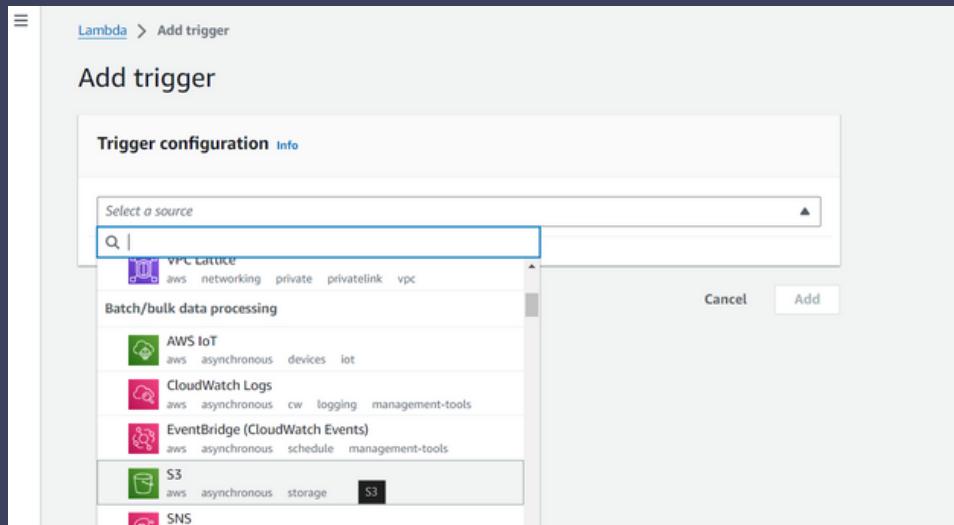


Y la segunda que en nuestro bucket tendremos nuestro archivo generado con la fecha del dia que se proceso.



The screenshot shows the Amazon S3 console interface. The path is: Amazon S3 > Buckets > sensorsdatav1 > st1\_raw/ > users/ > user1/. The 'Objects' tab is selected. A single object, 'user12024-03-30\_19-46-22.parquet', is listed. The object is a parquet file, last modified on March 30, 2024, at 13:46:25 (UTC-06:00), with a size of 3.9 KB and a storage class of Standard. The 'Actions' menu is visible above the object list, with 'Upload' highlighted.

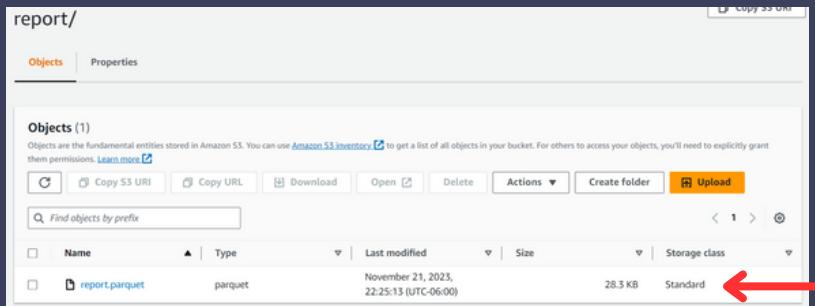
Faltaria crear un trigger para cada que se suba un archivo se ejecute de manera automatica.



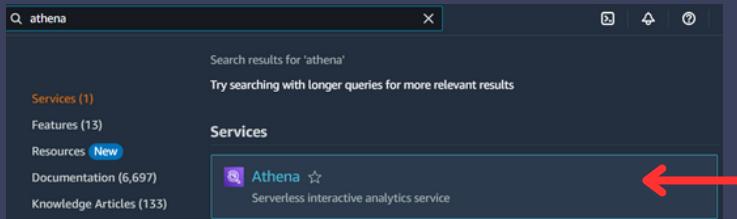
The screenshot shows the AWS Lambda 'Add trigger' configuration page. The 'Trigger configuration' section is active. In the 'Select a source' dropdown, 'S3' is selected. Other options like VPC, Lambda, and CloudWatch Logs are also listed. Below the dropdown, there are sections for 'Batch/bulk data processing' (AWS IoT, CloudWatch Logs, EventBridge) and 'SNS'. A 'Cancel' button and an 'Add' button are at the bottom right of the configuration panel.

# CONFIGURACION ATHENA

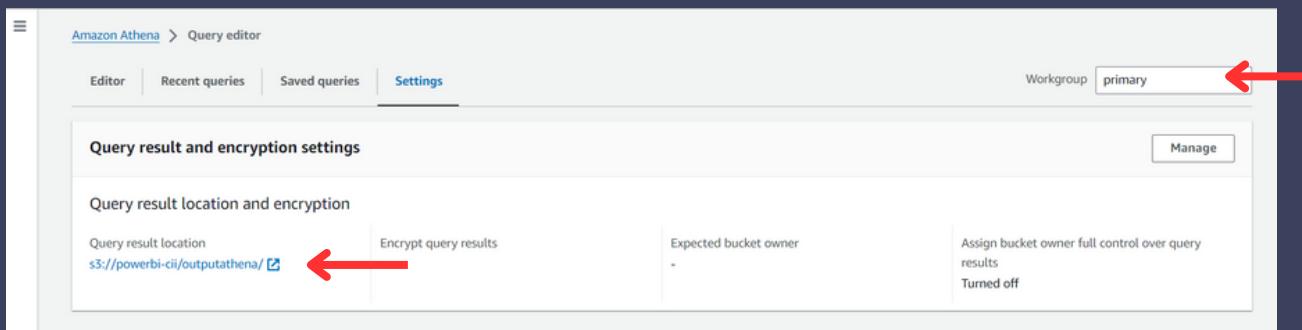
Para revisar que se creo nuestro archivo parquet tenemos que ir al S3 en la carpeta que seleccionamos.



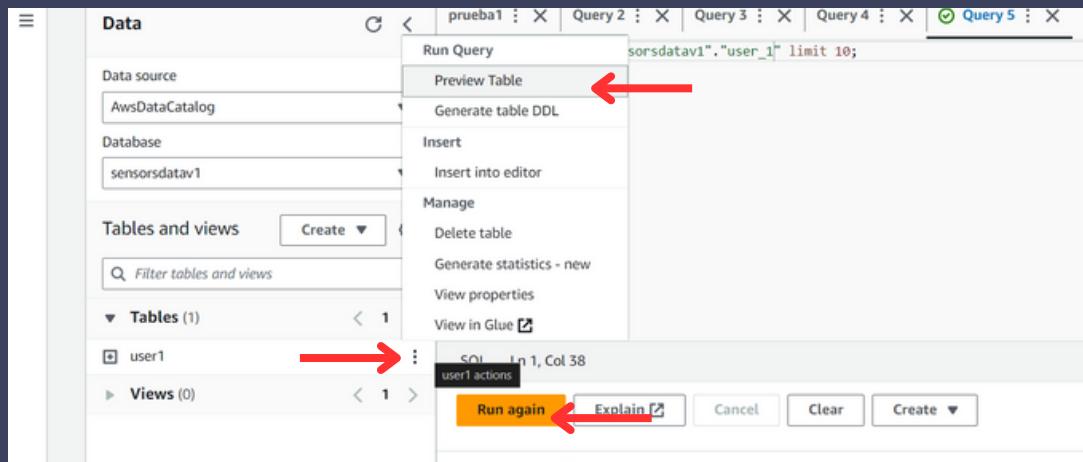
Solamente necesitamos confirmar en AWS Athena que la base de datos funciona. Seleccionar db y dar click en los 3 puntos de una tabla. Ver el preview de una tabla como una consulta SQL verificar que los datos aparezcan como se necesita. Seleccionar traxion



Seleccionar el grupo correcto es vital para evitar dañar otros procesos, así como seleccionar la carpeta de salida de los resultados de las consultas. Selecciona la ruta correcta



Dar clic en los tres puntos de la tabla que quieras visualizar y dar en la opción preview table. Esto ejecutara una sentencia SELECT en SQL para esa tabla.



The screenshot shows the 'Query results' page for the completed query. It displays the following information:

- Completed status
- Time in queue: 62 ms
- Run time: 662 ms
- Data scanned: 2.08 KB

The results table has 10 rows, each containing a timestamp and three sensor readings (temperature, humidity, pressure). The columns are labeled '#', 'timestamp', 'temperature (c)', 'humidity (%)', and 'pressure (kpa)'. The data is as follows:

#	timestamp	temperature (c)	humidity (%)	pressure (kpa)
1	2024-03-29 03:04:49	27.03	44.34	80.5
2	2024-03-29 03:05:09	27.21	44.16	80.49
3	2024-03-29 03:05:29	27.25	43.94	80.5
4	2024-03-29 03:05:49	27.34	43.94	80.49
5	2024-03-29 03:06:09	27.43	43.74	80.5
6	2024-03-29 03:06:29	27.5	43.6	80.5
7	2024-03-29 03:06:49	27.58	43.35	80.5
8	2024-03-29 03:04:49	27.03	44.34	80.5