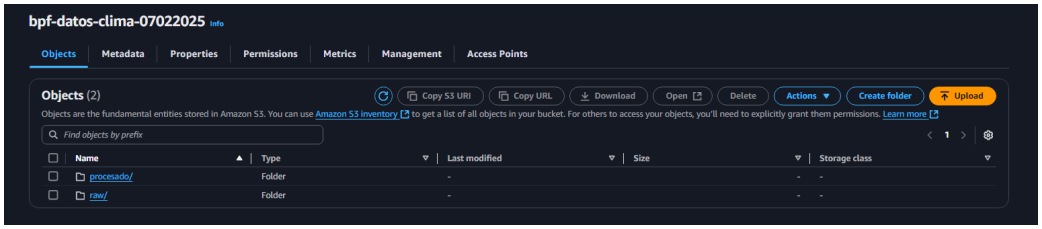
Proceso ETL usando AWS

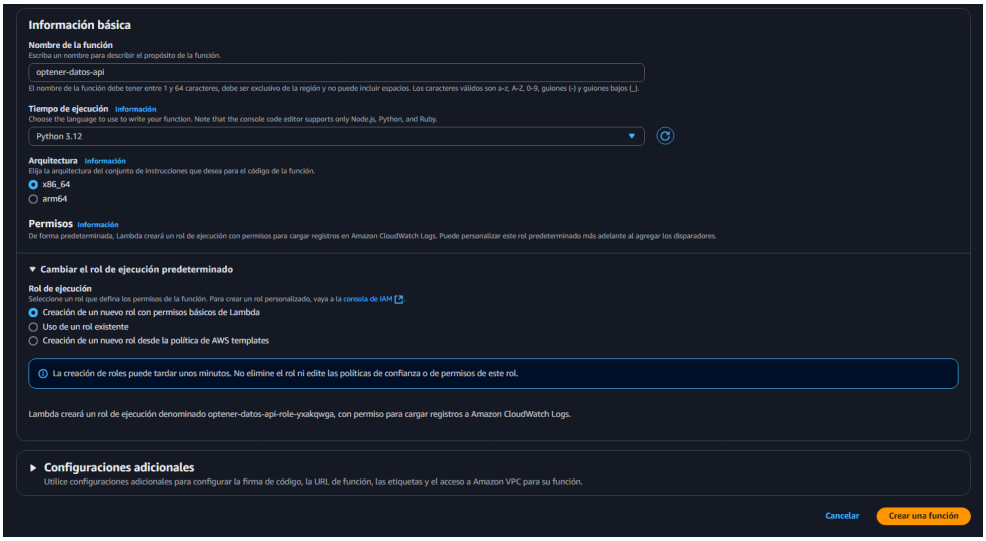
S3

1. Iniciamos sesión en la consola de aws.
2. Abrimos el servicio de S3.
3. Creamos el bucket con el nombre bpf-datos-clima-070222052.
    a. Dejamos las configuraciones por defecto y damos en crear.
4. Dentro de nuestro bucker creamos dos carpetas con los nombres “raw/” y “procesado/”.

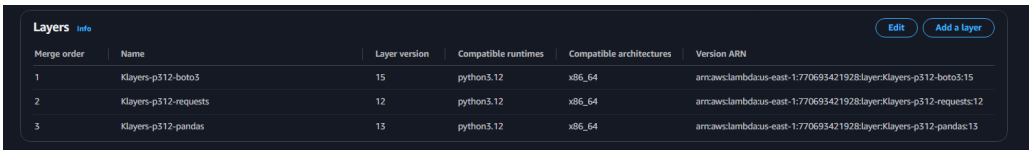


LAMBDA

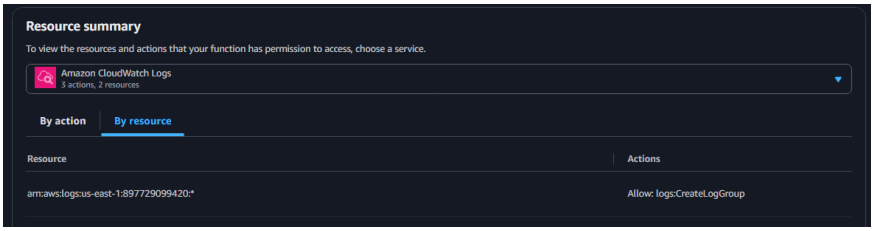
1. Abrimos nuestro servicio de lambda y damo en crear funcion
    a. Damos el nombre “obtener-datos-api”
    b. Seleccionamos un entorno de ejecución python 3.12
    c. Creamos un nuevo rol de ejecución



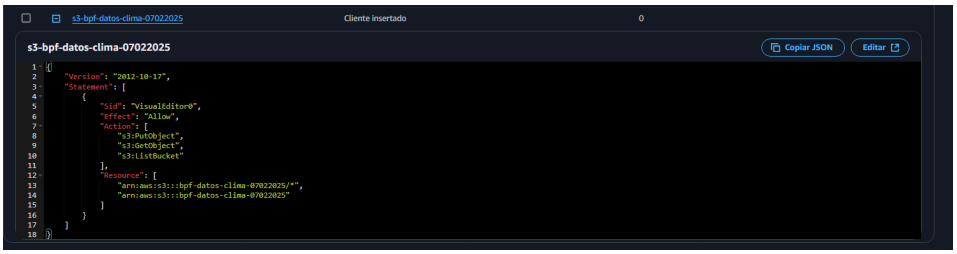
    d. Seleccionamos crear funcion
2. Agregamos capas:
    a. En la pestaña código hasta la parte de abajo seleccionamos agregar capa
    b. Obtenemos los arn de request , pandas y boto3 desde la pagina :https://github.com/keithroza
    c. Agregamos los arn de cada capa , nos aseguramos que sean de el entorno 3.12



3. Modificamos lo permisos del Rol

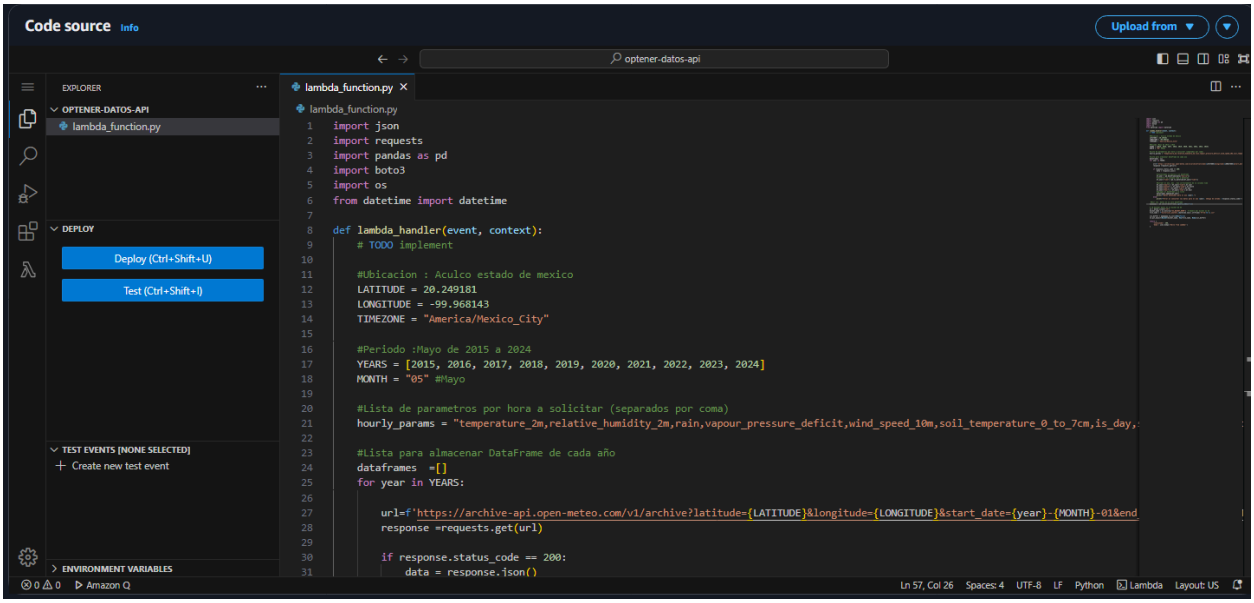


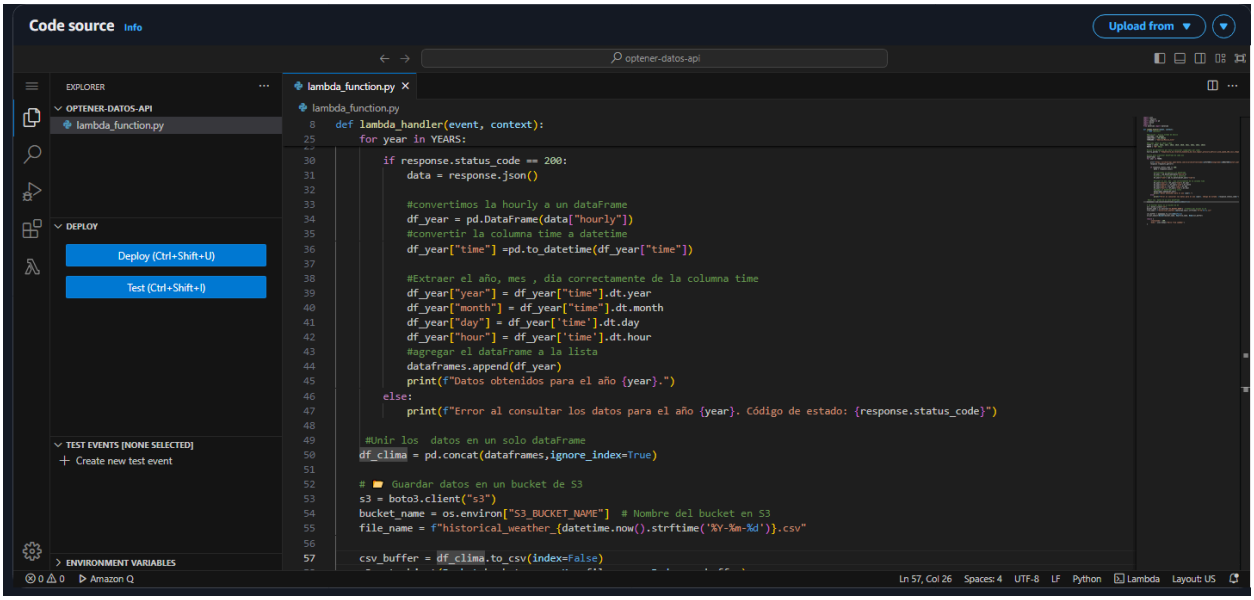
   a. En la pestaña configuración seleccionamos permisos en nombre del rol nos abrirá IAM
   b. Creamos una nueva política para darle acceso de **put get** y **list** a todos los objetos de nuestro bucket “bpf-datos-clima-070222052”.

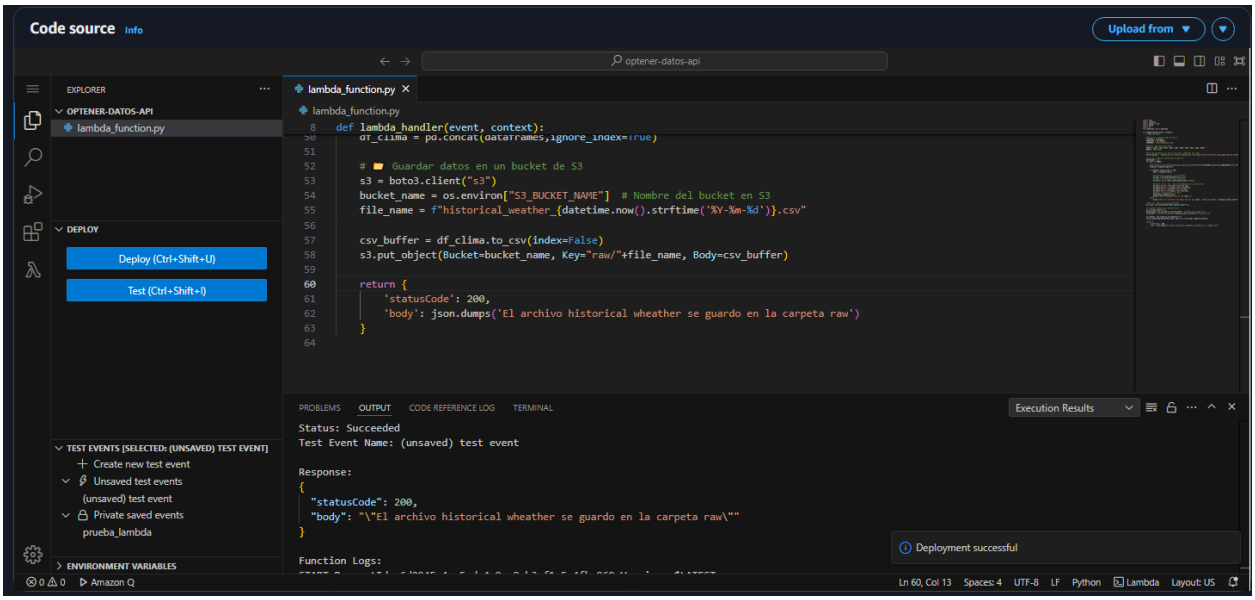


   c. Adjuntamos la política al rol
   d. Damos en guardar.
   e. Regresamos a lambda y actualizamos para ver que ya contamos con los permisos.

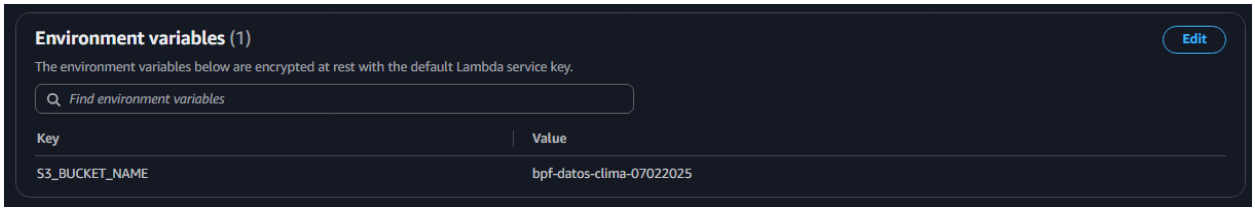
4. Agregamos el código que descarga los datos por medio de la api y los guardará en un archivo csv en el bucket bpf-datos-clima-070222052 en la carpeta raw





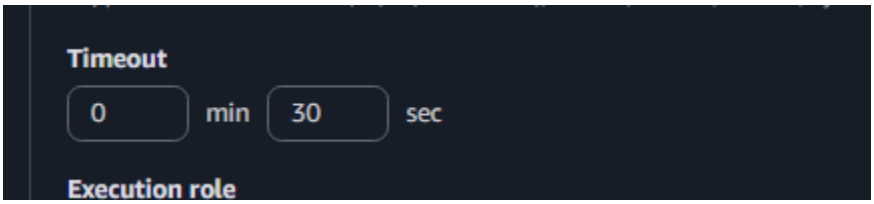


5. Agregamos nuestra variable de entorno que en este caso es el nombre de nuestro Bucket “bpf-datos-clima-070222052”. Este valor se asignará para la variable S3_BUCKET_NAME.

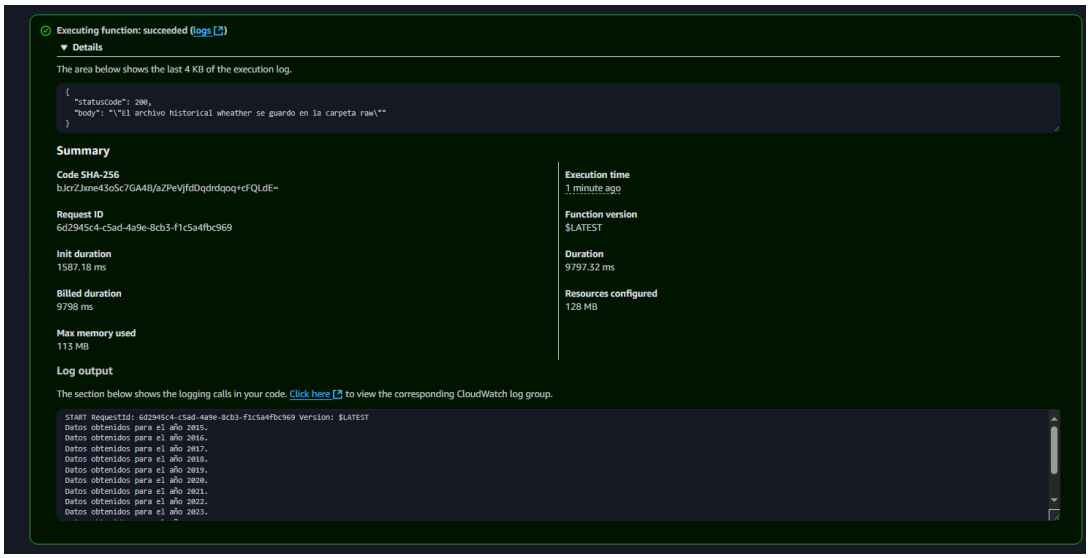


6. Al finalizar este proceso regresamos a la pestaña de código y en el entorno de desarrollo damos en deploy para guardar todos los cambios hechos en el código
7. Vamos a la pestaña configuración y en configuración general:

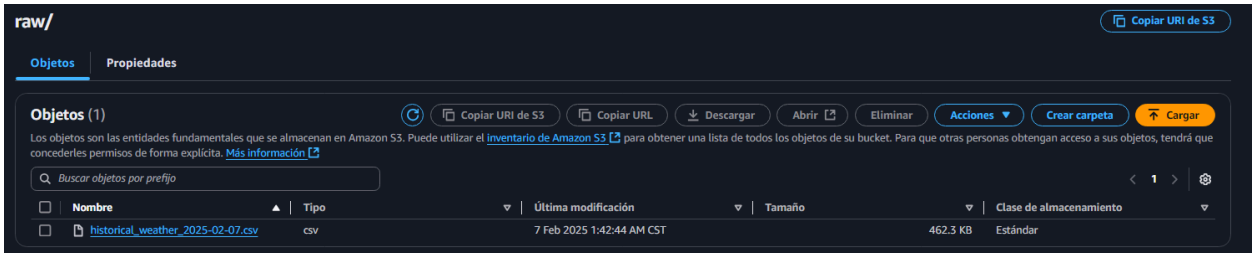
a. Editamos y en el timeout aumentamos el tiempo un valor adecuado para que se haga la descarga de datos desde la API.



8. Testeamos el código:
   a. Vamos a la carpeta test y usamos el test que viene por defecto , ya que no tenemos un disparador de la función , está ejecutara y descarga los archivos cuando presionemos test
   b. Damos click en Test



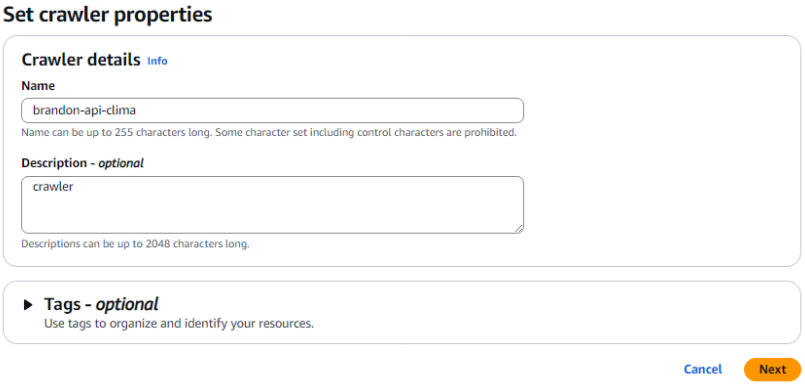
9. Revisamos nuestra carpeta raw en nuestro bucket “bpf-datos-clima-070222052”, para confirmar que nuestro archivo se generó.



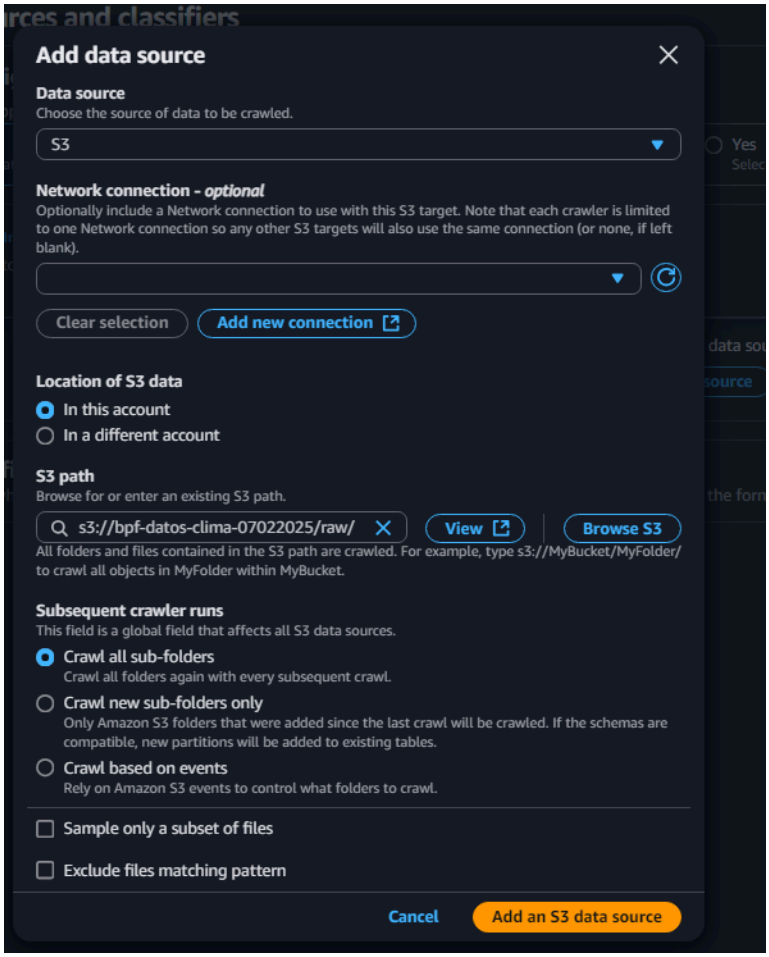
AWS GLUE

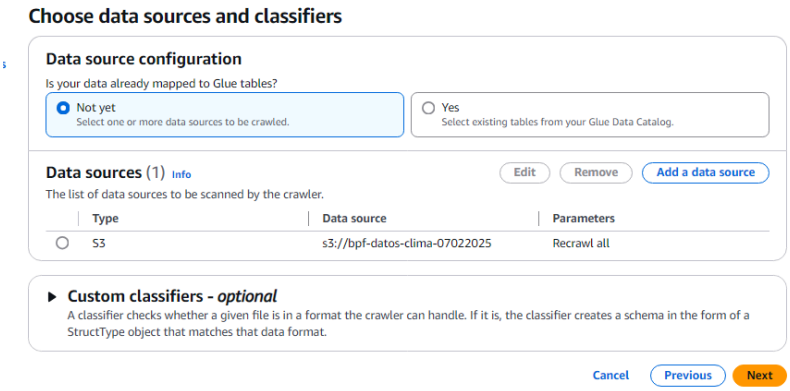
Como nuestros archivos no tienen un id y queremos cambiar los nombres de las columnas , realizamos la transformación en Glue

1. Abrimos el servicio de aws Glue
2. Seleccionamos la opción Crear Crawler y le damos un nombre

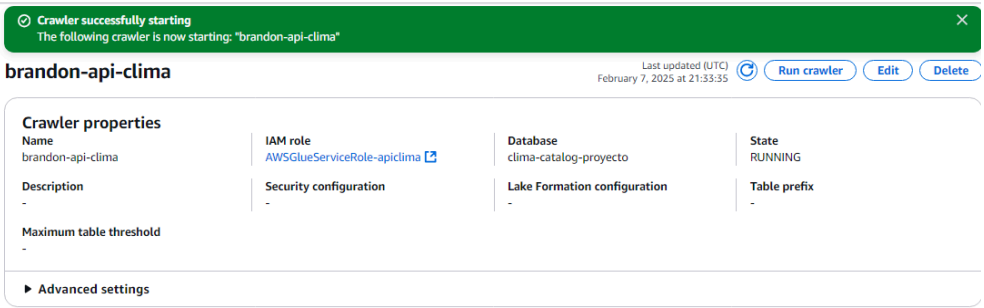


3. Añadimos una fuente de datos , en este caso nuestra carpeta en el bucket de s3

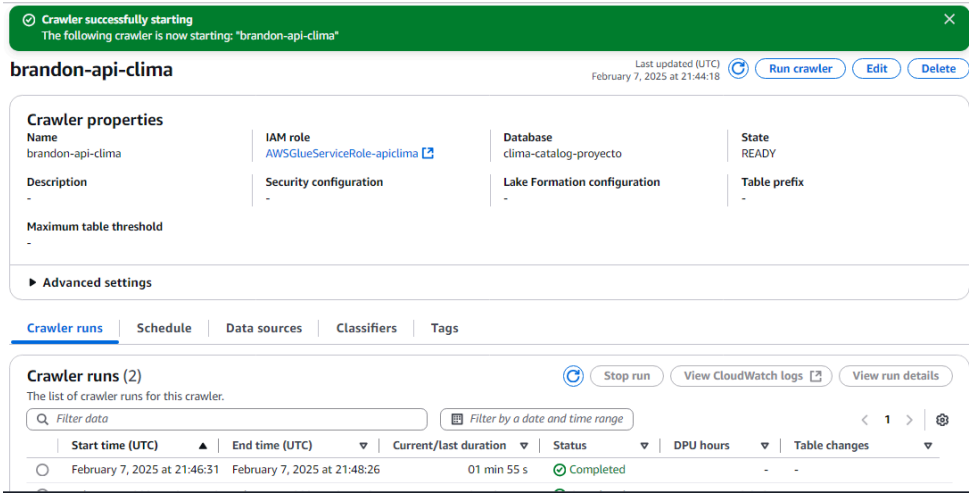




4. Creamos o seleccionamos un Rol que tenga acceso de creación y descargar de objetos de S3.
5. Añadimos un nueva base de datos destino , le damos nombre y la creamos
6. Revisamos que todo esté bien y creamos el Crawler



7. Una vez creando el crawler seleccionamos el crawler y damos en Run Crawler
    a. Esperamos que se ejecute y una vez terminado , revisamos los resultados



    b.

8. En la pestaña base de datos , seleccionamos Tablas y verificamos que se haya carga los metadatos de nuestro archivo csv que se proceso en nuestro crawler

Schema (13)

View and manage the table schema.

Edit schema as JSON | Edit schema

Filter schemas

1

#	Column name	Data type	Partition key	Comment
1	time	string	-	-
2	temperature_2m	double	-	-
3	relative_humidity_2m	bigint	-	-
4	rain	double	-	-
5	vapour_pressure_deficit	double	-	-
6	wind_speed_10m	double	-	-
7	soil_temperature_0_to_7cm	double	-	-
8	is_day	bigint	-	-
9	shortwave_radiation	double	-	-
10	year	bigint	-	-
11	month	bigint	-	-
12	day	bigint	-	-
13	hour	bigint	-	-

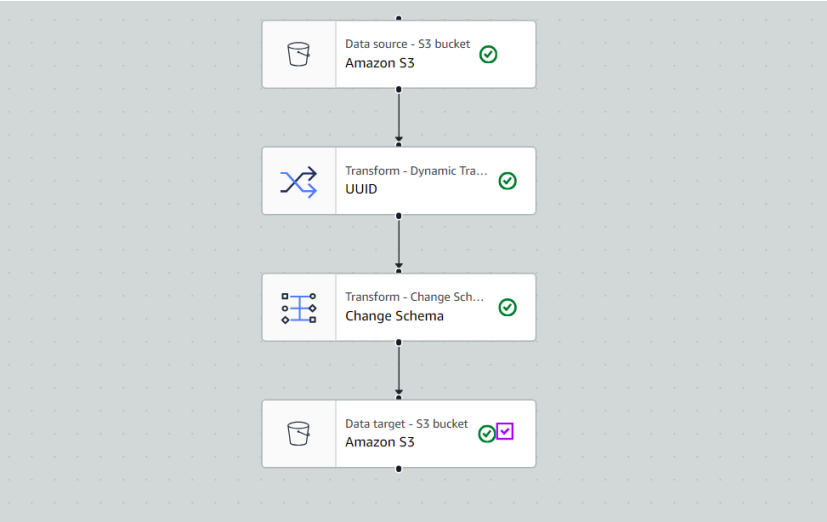
GLUE VISUAL JOB

1. Dentro de nuestro entorno para crear el trabajo seleccionamos Fuente para añadir nuestro tabla que se encuentra en el data Catalog
2. En la pestaña job Details colocamos un nombre y añadimos el rol que puede acceder escritura de S3
3. Añadimos un bloque de transformación UUID para añadirle un id a nuestro schema
4. Añadimos un bloque más de transformacion Change Schema y cambiamos los nombres de nuestras filas por los deseados, así como podemos observar que se ha añadido la columna de id que agregamos en paso anterior , también eliminamos la columna de time que contiene un formato dateTime

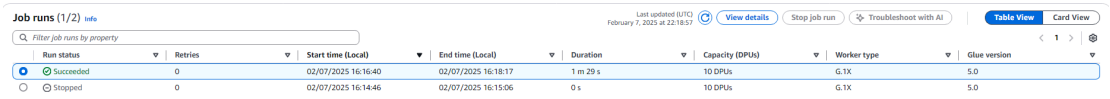
Change Schema (Apply mapping)

Source key	Target key	Data type	Drop
:ime			☑
:emperature_2m	temperatura	double	☐
'elative_humidity_2m	humedad_relativa	long	☐
'ain	precipitacion	double	☐
/apour_pressure_deficit	deficit_presion_vapo	double	☐
wind_speed_10m	velocidad_viento	double	☐
;oil_temperature_0_to_7cm	temperatura_suelo	double	☐
s_day	es_dia	int	☐
;hortwave_radiation	radiacion	double	☐
/ear	year	long	☐
nonth	month	long	☐
day	day	long	☐
nour	hour	long	☐
d	id	string	☐

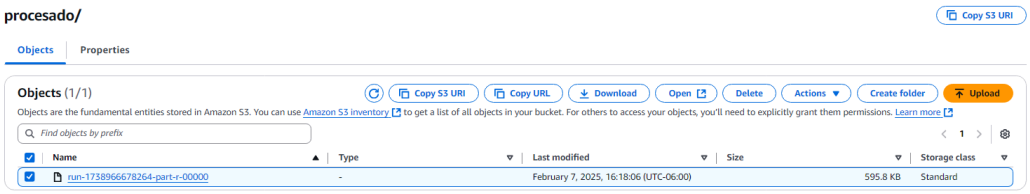
5. Por último añadimos un bloque de destino hacia nuestro carpeta procesado de nuestro bucket s3 en que le indicaremos el que formato de salida será un archivo csv sin ningún tipo de compresión
   Tu diagram deberá verse como este :



6. Una vez que se finalice , da en guardar y ejecuta tu job
   a. Espera a que se termine de ejecutar.



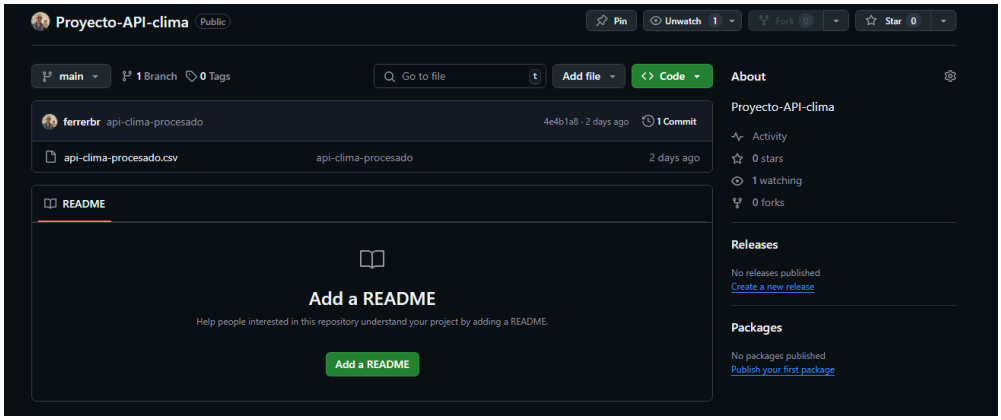
7. Revisa tu bucket en la carpeta de procesado para verificar que tu documento se encuentra ahí.



8. Descargamos el archivo y en el explorador editamos y colocamos el .csv que faltaba

GitHub

1. Crea un repositorio en GitHub, sube tu archivo .



Listo ya puedes consumir tu archivo para continuar en colab.