

Tarea M54 – Bryan Alberto Coronado García

- Creación del DataLake en AWS Glue

The screenshot shows the AWS Glue Database properties page for 'datalake-ebac'. The left sidebar lists various AWS Glue services: Getting started, ETL jobs (Visual ETL, Notebooks, Job run monitoring), Data Catalog tables, Data connections, Workflows (orchestration), Zero-ETL integrations (New), and Data Catalog (Databases, Tables, Stream schema registries, Schemas, Connections). The main content area displays the database properties: Name (datalake-ebac), Description (Datalake de prueba para Tarea M54), Location (-), and Created on (UTC) (August 6, 2025 at 03:34:30). Below this, the 'Tables (0)' section shows a table header with columns: Name, Database, Location, Classification, Deprecated, View data, Data quality, and Column statistics. A search bar at the top of the table header contains the placeholder 'Filter tables'. The status bar at the bottom indicates 'Last updated (UTC) August 6, 2025 at 03:34:38'.

- Carga de datos a un nuevo bucket S3

Objects		Metadata	Properties	Permissions	Metrics	Management	Access Points
Objects (3)							
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more							
<input type="text"/> Find objects by prefix							
<input type="checkbox"/> Name	Type	Last modified	Size	Storage class			
Analista de datos M54 - kc_house_data.csv	csv	August 6, 2025, 15:43:39 (UTC-06:00)	2.4 MB	Standard			
Analista de datos M54 - supermarket_sales.csv	csv	August 6, 2025, 15:43:39 (UTC-06:00)	128.4 KB	Standard			
Analista de datos M54 - wine-clustering.csv	csv	August 6, 2025, 15:43:39 (UTC-06:00)	10.8 KB	Standard			

- Error al generar el generador crawler por falta de permisos (verificado en el Log) y modificación del rol IAM

Log events		Actions		Start tailing		Create metric filter					
You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns											
Timestamp		Message									
<input type="text" value="d6382e9d-c07f-42af-bf20-c43939c2ef96"/>		<button>Clear</button> <button>1m</button> <button>30m</button> <button>1h</button> <button>12h</button> <button>Custom</button> <button>UTC timezone</button> <button>Display</button>									
▶	Timestamp	Message									
▶	2025-08-06T21:53:12.389Z	[d6382e9d-c07f-42af-bf20-c43939c2ef96] BENCHMARK : Running Start Crawl for Crawler ebacdata									
▼	2025-08-06T21:53:20.493Z	[d6382e9d-c07f-42af-bf20-c43939c2ef96] ERROR : Not all read errors will be logged. com.amazonaws.services.s3.model.AmazonS3Exception: Service Principal: glue...									
[d6382e9d-c07f-42af-bf20-c43939c2ef96]	ERROR : Not all read errors will be logged. com.amazonaws.services.s3.model.AmazonS3Exception: Service Principal: glue.amazonaws.com is not authorized to perform: s3:GetObject on resource: "arn:aws:s3:::datam54/Analista de datos M54 - supermarket_sales.csv" because no identity-based policy allows the s3:GetObject action (Service: Amazon S3; Status Code: 403; Error Code: AccessDenied; Request ID: 87Pf09ApBENFY652; S3 Extended Request ID: khf06P/qrmTfhl8lvj7oQsT5AsbFhrh/SaTWEBG9MBrNtztWCpCXFrP8Pcowb6mOhgJ90b1SS2A=; Proxy: null), S3 Extended Request ID: khf06P/qrmTfhl8lvj7oQsT5AsbFhrh/SaTWEBG9MBrNtztWCpCXFrP8Pcowb6mOhgJ90b1SS2A=	Copy									
▼	2025-08-06T21:53:20.493Z	[d6382e9d-c07f-42af-bf20-c43939c2ef96] ERROR : Not all read errors will be logged. com.amazonaws.services.s3.model.AmazonS3Exception: Service Principal: glue...									
[d6382e9d-c07f-42af-bf20-c43939c2ef96]	ERROR : Not all read errors will be logged. com.amazonaws.services.s3.model.AmazonS3Exception: Service Principal: glue.amazonaws.com is not authorized to perform: s3:GetObject on resource: "arn:aws:s3:::datam54/Wine-clustering.csv" because no identity-based policy allows the s3:GetObject action (Service: Amazon S3; Status Code: 403; Error Code: AccessDenied; Request ID: 87PA29WMQ2ZWP6; S3 Extended Request ID: DBc5WG2SxQAKdjYd/Bgy9DRClnI0YP0umKexprF+ODUHwWMMH8CCJCMIsOmtyKj4EkxADTsLJw=; Proxy: null), S3 Extended Request ID: DBc5WG2SxQAKdjYd/Bgy9DRClnI0YP0umKexprF+ODUHwWMMH8CCJCMIsOmtyKj4EkxADTsLJw=	Copy									
▼	2025-08-06T21:53:20.494Z	[d6382e9d-c07f-42af-bf20-c43939c2ef96] ERROR : Not all read errors will be logged. com.amazonaws.services.s3.model.AmazonS3Exception: Service Principal: glue...									
[d6382e9d-c07f-42af-bf20-c43939c2ef96]	ERROR : Not all read errors will be logged. com.amazonaws.services.s3.model.AmazonS3Exception: Service Principal: glue.amazonaws.com is not authorized to perform: s3:GetObject on resource: "arn:aws:s3:::datam54/kc_house_data.csv" because no identity-based policy allows the s3:GetObject action (Service: Amazon S3; Status Code: 403; Error Code: AccessDenied; Request ID: 87P4KN5XBZDER0Z; S3 Extended Request ID: TXw1ImlJaLtm5tsinSBuvanJnLnE975xFTB0X30Ll1l7r3/y2Pp9npnwMH1DZdAtQsaoONv0m8=; Proxy: null), S3 Extended Request ID: TXw1ImlJaLtm5tsinSBuvanJnLnE975xFTB0X30Ll1l7r3/y2Pp9npnwMH1DZdAtQsaoONv0m8=	Copy									
▼	2025-08-06T21:53:27.637Z	[d6382e9d-c07f-42af-bf20-c43939c2ef96] BENCHMARK : Classification complete, writing results to database datalake-ebac									
[d6382e9d-c07f-42af-bf20-c43939c2ef96]	BENCHMARK : Classification complete, writing results to database datalake-ebac	Copy									
▶	2025-08-06T21:53:27.643Z	[d6382e9d-c07f-42af-bf20-c43939c2ef96] INFO : Crawler configured with Configuration {"Version":1.0,"CreatePartitionIndex":true} and SchemaChangePolicy {"Updat...									
▶	2025-08-06T21:53:48.716Z	[d6382e9d-c07f-42af-bf20-c43939c2ef96] BENCHMARK : Finished writing to Catalog									
▶	2025-08-06T21:53:48.871Z	[d6382e9d-c07f-42af-bf20-c43939c2ef96] BENCHMARK : Crawler has finished running and is in state READY									

Policies (1381) [Info](#)

A policy is an object in AWS that defines permissions.

[Actions](#) [Delete](#) [Create policy](#)

Policy name		Type	Used as	Description
AWSGlueServiceRole-housing-EZCRC-s3Policy	CUSTOMER MANAGED		Permissions policy (1)	This policy will be used for Glue Crawl...

AWSGlueServiceRole-housing-EZCRC-s3Policy

This policy will be used for Glue Crawler and Job execution. Please do NOT delete!

```
1 - {  
2     "Version": "2012-10-17",  
3     "Statement": [  
4         {  
5             "Effect": "Allow",  
6             "Action": [  
7                 "s3:GetObject",  
8                 "s3:PutObject"  
9             ],  
10            "Resource": [  
11                "arn:aws:s3:::datalake-m54/*"  
12            ],  
13            "Condition": {  
14                "StringEquals": {  
15                    "aws:ResourceAccount": "651746472594"  
16                }  
17            }  
18        }  
19    ]  
20}
```

[Copy JSON](#) [Edit](#)

- Crawler procesado correctamente para la generación de tablas en el DataLake

ebacdata

Crawler successfully starting
The following crawler is now starting: "ebacdata"

Last updated (UTC) August 6, 2025 at 22:53:50 [Run crawler](#) [Edit](#) [Delete](#)

Crawler properties			
Name ebacdata	IAM role AWSGlueServiceRole-housing	Database datalake-ebac	State READY
Description -	Security configuration -	Lake Formation configuration -	Table prefix -
Maximum table threshold -			
Advanced settings			
Crawler runs Schedule Data sources Classifiers Tags			

Crawler runs (2)

The list of crawler runs for this crawler.

Last updated (UTC) August 6, 2025 at 22:57:50 [Stop run](#) [View CloudWatch logs](#) [View run details](#)

[Filter data](#) [Filter by a date and time range](#)

Start time (UTC)	End time (UTC)	Current/last duration	Status	DPU hours	Table changes
August 6, 2025 at 22:06:42	August 6, 2025 at 22:07:29	46 s	Completed	0.130	3 table changes, 0 partition changes

datalake-ebac

Last updated (UTC) August 6, 2025 at 22:57:50 [Edit](#) [Delete](#)

Database properties

Name datalake-ebac	Description Datalake de prueba para Tarea M54	Location -	Created on (UTC) August 6, 2025 at 03:34:30
-----------------------	--	---------------	--

Tables (3)

View and manage all available tables.

Last updated (UTC) August 6, 2025 at 22:57:51 [Delete](#) [Add tables using crawler](#) [Add table](#)

[Filter tables](#)

Name	Database	Location	Classification	Deprecated	View data	Data quality	Column statistics
analista_de_datos_m54_	datalake-ebac	s3://datam54/Analista de	CSV	-	Table data	View data quality	View statistics
analista_de_datos_m54_	datalake-ebac	s3://datam54/Analista de	CSV	-	Table data	View data quality	View statistics
analista_de_datos_m54_	datalake-ebac	s3://datam54/Analista de	CSV	-	Table data	View data quality	View statistics

- Se generó una conexión a AWS dese VsCode usando la herramienta “AWS Toolkit” y generando un rol de usuario para su acceso

The screenshot shows the AWS IAM User 'BryanVsCode' configuration page. The top navigation bar includes 'BryanVsCode' and 'Info' buttons, and a 'Delete' button. The main section is titled 'Summary' and contains the following details:

ARN	arn:aws:iam::651746472594:user/BryanVsCode	Console access	Access key 1
		Disabled	AKIAZPPY56KJBMQOWUD - Active Used today. Created today.
Created	August 07, 2025, 21:38 (UTC-06:00)	Last console sign-in	Access key 2
		-	Create access key

Below the summary, there are tabs for 'Permissions', 'Groups', 'Tags', 'Security credentials', and 'Last Accessed'. The 'Permissions' tab is selected. The 'Permissions policies' section shows one policy named 'S3Permisos' attached via 'Customer inline' in 'Inline' mode. There are buttons for 'Remove' and 'Add permissions'.

The screenshot shows the AWS CloudWatch Metrics Insights interface. The left sidebar displays various AWS services like AWS Lambda, Amazon S3, and Amazon Kinesis. The main area shows a query for 'Analista de datos M54 - kc_house_data.csv' which imports data from S3 and creates a DataFrame. The resulting DataFrame has 10 rows and 21 columns, showing house prices and features. The bottom status bar indicates 'Spaces: 4 LF () ⚡ Cell 2 of 3 ✨ Prettier'.

```
1 import pandas as pd
2 import boto3
3 from io import StringIO
4
5 session = boto3.Session(profile_name='BryanVsCode')
6 s3 = session.client('s3')
7
8 # Se extraer el CSV desde el bucket S3
9 csv_obj = s3.get_object(Bucket='data-ebac', Key='Analista de datos M54 - kc_house_data.csv')
10 body = csv_obj['Body'].read().decode('utf-8')
11
12 # Lectura del CSV en un DataFrame de pandas
13 df = pd.read_csv(StringIO(body))
14 df.head(10)
```

10 rows x 21 columns

- Exploración de la información de “kc_house_data” en Python

Conexión a la base de datos e importación del archivo de housing / Exploración de la información de housing utilizando Python

- Nota 1: El archivo CSV en el que se valora la data de las casas a detalle es el nombrado ‘kc_house_data’, el cual era usado para esta practica y no el ‘housing.csv’ como anuncian las instrucciones.
- Nota 2: Se utiliza la librería “pandasql” para consultas SQL en dataframes de pandas para mejor legibilidad optando por esta opción como factible para la realización de la practica.

```

1 import pandas as pd
2 import pandasql as psql # Para realizar consultas SQL sobre DataFrames de pandas
3 import numpy as np
4 import boto3
5 from io import StringIO
6
7 session = boto3.Session(profile_name='BryanVsCode')
8 s3 = session.client('s3')
9
10 # Se extraer el CSV desde el bucket S3
11 csv_obj = s3.get_object(Bucket='data-ebac', Key='Analista de datos M54 - kc_house_data.csv')
12 body = csv_obj['Body'].read().decode('utf-8')
13
14 # Lectura del CSV en un DataFrame de pandas
15 df = pd.read_csv(StringIO(body))
16 df.head(10)

```

[14] ✓ 3.9s

...	id	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	...	grade	sqft_above	sqft_basement	yr_built	yr_renovated	zipcode	lat	long	sqft_living15	sqft_lot15	Python
100	221900.000	3	1.000	1180	5650	1.000	0	0	...	7	1180	0	1955	0	98178	47.511	-122.257	1340	5650		
100	538000.000	3	2.250	2570	7242	2.000	0	0	...	7	2170	400	1951	1991	98125	47.721	-122.319	1690	7639		
100	180000.000	2	1.000	770	10000	1.000	0	0	...	6	770	0	1933	0	98028	47.738	-122.233	2720	8062		
100	604000.000	4	3.000	1960	5000	1.000	0	0	...	7	1050	910	1965	0	98136	47.521	-122.393	1360	5000		
100	510000.000	3	2.000	1680	8080	1.000	0	0	...	8	1680	0	1987	0	98074	47.617	-122.045	1800	7503		
100	1225000.000	4	4.500	5420	101930	1.000	0	0	...	11	3890	1530	2001	0	98053	47.656	-122.005	4760	101930		
100	257500.000	3	2.250	1715	6819	2.000	0	0	...	7	1715	0	1995	0	98003	47.310	-122.327	2238	6819		
100	291850.000	3	1.500	1060	9711	1.000	0	0	...	7	1060	0	1963	0	98198	47.410	-122.315	1650	9711		
100	229500.000	3	1.000	1780	7470	1.000	0	0	...	7	1050	730	1960	0	98146	47.512	-122.337	1780	8113		
100	323000.000	3	2.500	1890	6560	2.000	0	0	...	7	1890	0	2003	0	98038	47.368	-122.031	2390	7570		

```

1 # Obtenemos la descripción del DataFrame
2 pd.set_option('display.float_format', lambda x: '%.2f' % x)
3 df.describe().T

```

[21] ✓ 0.0s

...	count	mean	std	min	25%	50%	75%	max
id	21613.00	4580301520.86	2876565571.31	1000102.00	2123049194.00	3904930410.00	7308900445.00	9900000190.00
price	21613.00	540088.14	367127.20	75000.00	321950.00	450000.00	645000.00	7700000.00
bedrooms	21613.00	3.37	0.93	0.00	3.00	3.00	4.00	33.00
bathrooms	21613.00	2.11	0.77	0.00	1.75	2.25	2.50	8.00
sqft_living	21613.00	2079.90	918.44	290.00	1427.00	1910.00	2550.00	13540.00
sqft_lot	21613.00	15106.97	41420.51	520.00	5040.00	7618.00	10688.00	1651359.00
floors	21613.00	1.49	0.54	1.00	1.00	1.50	2.00	3.50
waterfront	21613.00	0.01	0.09	0.00	0.00	0.00	0.00	1.00
view	21613.00	0.23	0.77	0.00	0.00	0.00	0.00	4.00
condition	21613.00	3.41	0.65	1.00	3.00	3.00	4.00	5.00
grade	21613.00	7.66	1.18	1.00	7.00	7.00	8.00	13.00
sqft_above	21613.00	1788.39	828.09	290.00	1190.00	1560.00	2210.00	9410.00
sqft_basement	21613.00	291.51	442.58	0.00	0.00	0.00	560.00	4820.00
yr_built	21613.00	1971.01	29.37	1900.00	1951.00	1975.00	1997.00	2015.00
yr_renovated	21613.00	84.40	401.68	0.00	0.00	0.00	0.00	2015.00
zipcode	21613.00	98077.94	53.51	98001.00	98033.00	98065.00	98118.00	98199.00
lat	21613.00	47.56	0.14	47.16	47.47	47.57	47.68	47.78
long	21613.00	-122.21	0.14	-122.52	-122.33	-122.23	-122.12	-121.31
sqft_living15	21613.00	1986.55	685.39	399.00	1490.00	1840.00	2360.00	6210.00
sqft_lot15	21613.00	12768.46	27304.18	651.00	5100.00	7620.00	10083.00	871200.00

- Incluir tres análisis adicionales seleccionados por el estudiante que respondan a preguntas que el negocio quisiera hacer (Incluyendo KPIs)

```

1 # -----
2 # Cuáles son los 10 códigos postales con el promedio más alto en precios?
3 #
4
5 psql.sqlrf("SELECT zipcode, \
6             AVG(price) as PrecioPromedio \
7             FROM df \
8             GROUP BY zipcode \
9             ORDER BY price \
10            limit 10")
11
12 # Esta consulta nos ayuda a comprender y valorar los zipcodes mas valiosos para posibles inversiones
13 # ante los clientes, así como para la empresa, ya que estos datos pueden ser utilizados
14 # para la toma de decisiones estratégicas en el negocio inmobiliario.
15 # Adicionalmente, se puede observar que los precios promedio de las casas en estos códigos postales
16 # son significativamente más altos que en otros códigos postales, lo que indica una alta demanda
17 # y un mercado inmobiliario activo en estas áreas. Lo que nos puede ayudar a entender alguna posible
18 # centralización de la zona, o el crecimiento de la misma.

[7] ✓ 0.3s
```

	zipcode	PrecioPromedio
0	98168	240328.37
1	98028	462480.04
2	98106	319581.39
3	98002	234284.04
4	98023	286732.79
5	98178	310612.76
6	98148	284908.60
7	98146	359483.24
8	98155	423725.70
9	98031	300539.89

```

1 # -----
2 # Cuánto aumenta el precio por cada piso en una casa?
3 #
4
5 psql.sqlrf("SELECT floors, \
6             AVG(price) as PrecioPromedio \
7             FROM df \
8             GROUP BY floors \
9             ORDER BY floors")
10
11 # Esta consulta nos permite confirmar el aumento estimado del precio por cada piso en una casa.
12 # Podemos observar que el precio promedio de las casas aumenta a medida que aumenta el número de pisos
13 # en la casa. Esto indica que las casas con más pisos tienden a tener un precio más alto, lo que puede ser
14 # un factor importante a considerar al evaluar el valor de una propiedad.
15 # También puede ser útil para la empresa al evaluar el valor de las propiedades y establecer precios
16 # competitivos en el mercado inmobiliario.

[56] ✓ 0.4s
```

	floors	PrecioPromedio
0	1.00	442180.63
1	1.50	558980.64
2	2.00	648891.16
3	2.50	1060346.49
4	3.00	582526.04
5	3.50	933312.50

```

1 # -----
2 # Las casas más antiguas siguen siendo rentables?
3 # -----
4 psql.sql("SELECT yr_built as AñoDeConstrucción, \
5           AVG(price) as PrecioPromedio, \
6           price/yr_built as PrecioPorAñoDeConstrucción \
7           FROM df \
8           GROUP BY yr_built \
9           ORDER BY yr_built ASC \
10          limit 21")
11
12 # Esta consulta nos ayuda a entender la relación entre la antigüedad de las casas y su precio.
13 # Podemos verificar si las casas más antiguas tienden a tener precios más bajos,
14 # lo que podría indicar una menor demanda o una percepción de menor valor en el mercado.
15 # También, al observar el precio por año de construcción, podemos identificar si hay una tendencia
16 # a que las casas más antiguas tengan un precio por año de construcción más bajo,
17 # lo que podría indicar una menor rentabilidad en el mercado inmobiliario para estas propiedades, a su vez
18 # valorar alguna posible renovación o remodelación de las mismas para aumentar su valor.

```

[43] ✓ 0.4s

	AñoDeConstrucción	PrecioPromedio	PrecioPorAñoDeConstrucción
0	1900	581387.21	278.95
1	1901	556935.93	202.52
2	1902	673007.41	341.75
3	1903	480958.20	336.31
4	1904	583756.64	336.13
5	1905	752977.99	299.21
6	1906	669799.40	142.35
7	1907	676257.25	229.42
8	1908	564348.69	212.26
9	1909	696135.16	349.40
10	1910	671536.31	445.46
11	1911	632488.36	408.16
12	1912	612990.70	41.84
13	1913	585683.22	378.99
14	1914	615153.48	757.58
15	1915	584896.30	489.30
16	1916	600915.04	253.13
17	1917	528108.93	108.48
18	1918	492246.88	250.26
19	1919	537779.60	334.78
20	1920	477804.40	220.31