

- Creación del DataLake en AWS Glue

AWS Glue > Databases > datalake-ebac
Last updated (UTC)
August 6, 2025 at 03:34:37 [🔄](#) [Edit](#) [Delete](#)

AWS Glue

- Getting started
- ETL jobs
 - Visual ETL
 - Notebooks
 - Job run monitoring
- Data Catalog tables
- Data connections
- Workflows (orchestration)
- Zero-ETL Integrations [New](#)

Data Catalog

[Databases](#)

- Tables
- Stream schema registries
- Schemas
- Connections

datalake-ebac

Database properties

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

Tables (0)

View and manage all available tables.

Last updated (UTC) August 6, 2025 at 03:34:38 [🔄](#) [Delete](#) [Add tables using crawler](#) [Add table](#)

<input type="checkbox"/>	Name	▲	Database ▼	Location ▼	Classification ▼	Deprecated ▼	View data	Data quality	Column statistics
No available tables									

- Carga de datos a un nuevo bucket S3

datam54

info

Objects

Metadata

Properties

Permissions

Metrics

Management

Access Points

Objects (3)

Copy S3 URL

Copy URL

Download

Open

Delete

Actions

Create folder

Upload

Q

Find objects by prefix

<

1

>

	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	Analista de datos M54 - kc_house_data.csv	csv	August 6, 2025, 15:43:39 (UTC-06:00)	2.4 MB	Standard
<input type="checkbox"/>	Analista de datos M54 - supermarket_sales.csv	csv	August 6, 2025, 15:43:39 (UTC-06:00)	128.4 KB	Standard
<input type="checkbox"/>	Analista de datos M54 - wine-clustering.csv	csv	August 6, 2025, 15:43:39 (UTC-06:00)	10.8 KB	Standard

- Error al generar al generar 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](#)

d6382e9d-c07f-42af-bf20-c43939c2ef96

Clear

1m

30m

1h

12h

Custom

UTC timezone

Display

Timestamp	Message
2025-08-06T12:53:12.389Z	[d6382e9d-c07f-42af-bf20-c43939c2ef96] BENCHMARK : Running Start Crawl for Crawler ebacdata
2025-08-06T12:53:20.493Z	[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 (Service: Amazon S3; Status Code: 403; Error Code: AccessDenied; Request ID: 87PF09APBENFY6S2; S3 Extended Request ID: khf06P/qRmTfh18lvj7oQsT5AsbFhrh/SaTWBEG9MbRNTztWcPCXfRp8Cpwb6m0hgJ90b1SS2A=; Proxy: null), S3 Extended Request ID: khf06P/qRmTfh18lvj7oQsT5AsbFhrh/SaTWBEG9MbRNTztWcPCXfRp8Cpwb6m0hgJ90b1SS2A=
2025-08-06T12:53:20.493Z	[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 - wine-clustering.csv" because no identity-based policy allows the s3:GetObject action (Service: Amazon S3; Status Code: 403; Error Code: AccessDenied; Request ID: 87PA29WQ2MF2WFP6; S3 Extended Request ID: DBc5WG25xQAKdjYd/Bgy9DRcInI0YP0umKexpFr+ODUhiYwMh8CJCMIS0mYKj4EkxADTSLJw=; Proxy: null), S3 Extended Request ID: DBc5WG25xQAKdjYd/Bgy9DRcInI0YP0umKexpFr+ODUhiYwMh8CJCMIS0mYKj4EkxADTSLJw=
2025-08-06T12:53:20.494Z	[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/Analista de datos M54 - 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: 87P4KN58XBZDERD2; S3 Extended Request ID: TXwI1mJalTm5tsin5BuavUnLnE975fTBOX30LiL7r3/y2Pp9nnpMH1IDZatQsaoNouV0mB=; Proxy: null), S3 Extended Request ID: TXwI1mJalTm5tsin5BuavUnLnE975fTBOX30LiL7r3/y2Pp9nnpMH1IDZatQsaoNouV0mB=
2025-08-06T12:53:27.637Z	[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
2025-08-06T12:53:27.643Z	[d6382e9d-c07f-42af-bf20-c43939c2ef96] INFO : Crawler configured with Configuration {"Version":1.0,"CreatePartitionIndex":true} and SchemaChangePolicy {"Updat...
2025-08-06T12:53:48.716Z	[d6382e9d-c07f-42af-bf20-c43939c2ef96] BENCHMARK : Finished writing to Catalog
2025-08-06T12:53:48.871Z	[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.


Q Search

Filter by Type

Customer managed

1 match

< 1 >



Policy name	Type	Used as	Description
<div><div><input type="checkbox"/></div><div>AWSGlueServiceRole-housing-EZCRC-s3Policy</div></div>	Customer managed	Permissions policy (1)	This policy will be used for Glue Crawl...

AWSGlueServiceRole-housing-EZCRC-s3Policy

Copy JSON

Edit


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:::datam54/*"
12-      ],
13-      "Condition": {
14-        "StringEquals": {
15-          "aws:ResourceAccount": "651746472594"
16-        }
17-      }
18-    }
19-  ]
20- }
```

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


 Crawler successfully starting

The following crawler is now starting: "ebacdata"



Last updated (UTC)

August 6, 2025 at 22:53:50



Run crawler

Edit

Delete

ebacdata

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)

Stop run

View CloudWatch logs


View run details

The list of crawler runs for this crawler.

Q Filter data

Filter by a date and time range

< 1 >



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


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)

Last updated (UTC)

August 6, 2025 at 22:57:51



Delete


Add tables using crawler

Add table

View and manage all available tables.

Q Filter tables

< 1 >



Name	Database	Location	Classification	Deprecated	View data	Data quality	Column statistics
<input type="checkbox"/> analista_de_datos_m54_	datalake-ebac	s3://datam54/Analista de	CSV	-	Table data	View data quality	View statistics
<input type="checkbox"/> analista_de_datos_m54_	datalake-ebac	s3://datam54/Analista de	CSV	-	Table data	View data quality	View statistics
<input type="checkbox"/> analista_de_datos_m54_	datalake-ebac	s3://datam54/Analista de	CSV	-	Table data	View data quality	View statistics

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

BryanVsCode Info Delete

Summary

ARN
arn:aws:iam::651746472594:user/BryanVsCode

Console access
Disabled

Access key 1
AKIAZPPY56KJBMCQWUD - Active
Used today. Created today.

Created
August 07, 2025, 21:38 (UTC-06:00)

Last console sign-in
-

Access key 2
[Create access key](#)

Permissions

Groups

Tags

Security credentials

Last Accessed

Permissions policies (1) Remove Add permissions

Permissions are defined by policies attached to the user directly or through groups.

Filter by Type
All types

< 1 >

<input type="checkbox"/>	Policy name	Type	Attached via
<input type="checkbox"/>	S3Permisos	Customer inline	Inline

AWSS3

AMAZON Q

EXPLORER

Connected with profile: BryanVsCode

US East (N. Virginia)

API Gateway

App Runner

CloudFormation

CloudWatch Logs

DocumentDB

EC2

ECR

ECS

IoT

Lambda

Redshift

S3

data-ebac

Analista de datos M54 - k...

Analista de datos M54 - superm...

Analista de datos M54 - wine-cl...

SageMaker AI

Schemas

Step Functions

Systems Manager

Resources

CDK

APPLICATION BUILDER

CODECATALYST

Tarea M54.ipynb

Analista de datos M54 - kc_house_data.csv

Tarea M54.ipynb

Tarea M54 - Bryan Alberto Coronado García

import pandas as pd

Generate

Code

Markdown

Run All

Restart

Clear All Outputs

Jupyter Variables

Outline

base (Python 3.13.2)

Tarea M54 - Bryan Alberto Coronado García

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)

[4] ✓ 1.7s

Python

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	...	grade	sqft_above	sqft_basement	yr_built	yr_renovated	zipcode	lat
0	7129300520	20141013T000000	2219000	3	1.00	1180	5650	1.0	0	0	...	7	1180	0	1955	0	98178	47.5112
1	6414100192	20141209T000000	5380000	3	2.25	2570	7242	2.0	0	0	...	7	2170	400	1951	1991	98125	47.7210
2	5631500400	20150225T000000	1800000	2	1.00	770	10000	1.0	0	0	...	6	770	0	1933	0	98028	47.7379
3	2487200875	20141209T000000	6040000	4	3.00	1960	5000	1.0	0	0	...	7	1050	910	1965	0	98136	47.5208
4	1954400510	20150218T000000	5100000	3	2.00	1680	8080	1.0	0	0	...	8	1680	0	1987	0	98074	47.6168
5	7237550310	20140512T000000	12250000	4	4.50	5420	101930	1.0	0	0	...	11	3890	1530	2001	0	98053	47.6561
6	1321400060	20140627T000000	2575000	3	2.25	1715	6819	2.0	0	0	...	7	1715	0	1995	0	98003	47.3097
7	2008000270	20150115T000000	2918500	3	1.50	1080	9711	1.0	0	0	...	7	1060	0	1963	0	98198	47.4095
8	2414600126	20150415T000000	2295000	3	1.00	1780	7470	1.0	0	0	...	7	1050	730	1960	0	98146	47.5123
9	3793500160	20150312T000000	3230000	3	2.50	1890	6560	2.0	0	0	...	7	1890	0	2003	0	98038	47.3684

10 rows x 21 columns

[]

Python

Launchpad

0 0 0

✓ AWS: profile: BryanVsCode

Spaces: 4

LF

()

Cell 2 of 3

✓ Prettier

- 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)
```

3.9s																				Python
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	
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
```

...		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.sqldf("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, asi 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 # centrificación de la zona, o el crecimiento de la misma.

```

✓ 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.sqldf("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.

```

✓ 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.sqldf("SELECT yr_built as AñoDeConstruccion, \
5             AVG(price) as PrecioPromedio, \
6             price/yr_built as PrecioPorAñoDeConstruccion \
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

...

	AñoDeConstruccion	PrecioPromedio	PrecioPorAñoDeConstruccion
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