

Basic Environment Setup:

Two VPCs created with one Glue Service in one VPC and the Database in another.

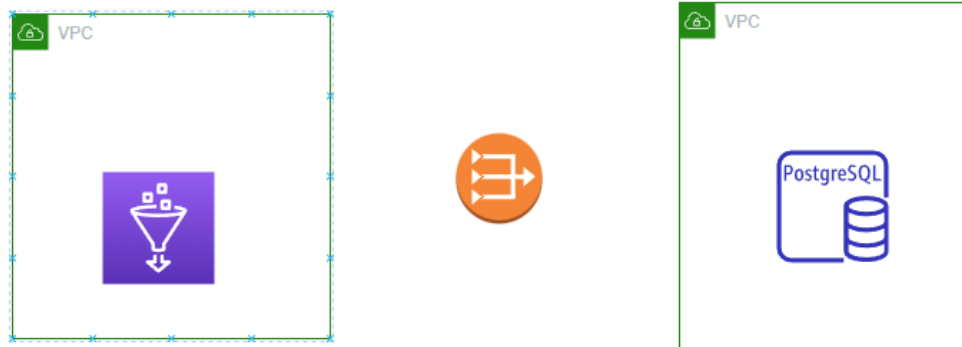
(Had trouble with using just one vpc)

Your VPCs (1/2) [Info](#)

[Refresh](#) [Actions](#) [Create VPC](#)

	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP option set	Main route table	M
<input type="checkbox"/>	-	vpc-0f807cda476542b22	Available	172.30.0.0/16	-	dopt-002ba652f0bab74...	rtb-04e6a5be363159cb2	ac
<input checked="" type="checkbox"/>	-	vpc-0842cd62b8c86c859	Available	172.31.0.0/16	-	dopt-002ba652f0bab74...	rtb-06f414a480d97e43	ac

Link used to get the 2 connected: <https://repost.aws/knowledge-center/glue-s3-endpoint-validation-failed>



VPC > Your VPCs > vpc-0f807cda476542b22

vpc-0f807cda476542b22

Details Info

VPC ID vpc-0f807cda476542b22	State Available	DNS hostnames Enabled	DNS resolution Enabled
Tenancy Default	DHCP option set dopt-002ba652f0bab7423	Main route table rtb-04e6a5be363159cb2	Main network ACL acl-057a6f42fc5458033
Default VPC No	IPv4 CIDR 172.30.0.0/16	IPv6 pool -	IPv6 CIDR (Network border group) -
Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups -	Owner ID 028191280483	

Resource map New

CIDRs | Flow logs | Tags

Resource map Info

VPC Show details
Your AWS virtual network
vpc-0f807cda476542b22

Subnets (6)
Subnets within this VPC
us-east-1a
subnet-0ca8e9a94103a7089
us-east-1b
subnet-0fe58beef31004d4d
us-east-1c
subnet-09e95a9f61f62da1a

Route tables (2)
Route network traffic to resources
db route table
rtb-04e6a5be363159cb2

Network connections (2)
Connections to other networks
igw-01e95ed7543c8cefe
db-gateway

Introducing the VPC resource map

Solid lines represent relationships between resources in your VPC. Dotted lines represent network traffic to resources.

VPC > Your VPCs > vpc-0842cd62b8c86c859

vpc-0842cd62b8c86c859

Details Info

VPC ID vpc-0842cd62b8c86c859	State Available	DNS hostnames Enabled	DNS resolution Enabled
Tenancy Default	DHCP option set dopt-002ba652f0bab7423	Main route table rtb-06f414a480df97e43	Main network ACL acl-0e4cf1f050ae3092d
Default VPC Yes	IPv4 CIDR 172.31.0.0/16	IPv6 pool -	IPv6 CIDR (Network border group) -
Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups -	Owner ID 028191280483	

Resource map New

CIDRs | Flow logs | Tags

Resource map Info

VPC Show details
Your AWS virtual network
vpc-0842cd62b8c86c859

Subnets (16)
Subnets within this VPC
us-east-1a
RDS-Pvt-subnet-5
subnet-0fd13194758461039
RDS-Pvt-subnet-8
us-east-1b
RDS-Pvt-subnet-9

Route tables (4)
Route network traffic to resources
rtb-0ab3ed0ab78bd05da
rtb-06f414a480df97e43
RDS-Pvt-rt
RDS-Pvt-rt

Network connections (2)
Connections to other networks
igw-044ad987eb79bcef9
test-nat

Introducing the VPC resource map

Solid lines represent relationships between resources in your VPC. Dotted lines represent network traffic to resources.

Database Setup:

Create one RDS Postgres DB. This will mimic the sara sync db

database-13e

Summary

DB identifier database-13e	CPU <div><div></div></div> 4.57%	Status Available	Class db.t3.micro
Role Instance	Current activity <div><div></div></div> 0 Connections	Engine PostgreSQL	Region & AZ us-east-1a

Connectivity & security

Monitoring

Logs & events

Configuration

Maintenance & backups

Tags

Connectivity & security

Endpoint & port

Endpoint
database-13e.czhqbccm3aqa.us-east-1.rds.amazonaws.com

Port
5432

Networking

Availability Zone
us-east-1a

VPC
[vpc-0f807cda476542b22](#)

Subnet group
default-vpc-0f807cda476542b22

Subnets
[subnet-0fe58beef31004d4d](#)
[subnet-0bebc40291688c2ef](#)
[subnet-09e95a9f61f62da1a](#)
[subnet-07e06ac276b1e7a76](#)

Security

VPC security groups
[default \(sg-02e85989bae844b19\)](#)
Active

Publicly accessible
Yes

Certificate authority [Info](#)
rds-ca-2019

Certificate authority date
August 22, 2024, 11:08 (UTC-06:00)

DB instance certificate expiration date

Using a DB client, Created a Veteran table and populated, schema simplified for spike purposes.

The screenshot shows a database client interface with a SQL editor and a results grid. The SQL editor contains the following commands:

```
CREATE TABLE Veteran (  
  id int,  
  LastName varchar(255),  
  FirstName varchar(255),  
  Address varchar(255),  
  Source varchar(255),  
  City varchar(255)  
);  
  
SELECT * FROM Veteran;  
  
INSERT INTO Veteran (id, LastName, FirstName, Address, Source, City) values (1, 'Doe', 'John', 'A', 'CWINRS', 'test');  
INSERT INTO Veteran (id, LastName, FirstName, Address, Source, City) values (2, 'RES Doe', 'RES John', 'A', 'RES', 'RES test');
```

The results grid shows the following data:

	id	last name	first name	address	source	city
1	1	Doe	John	A	CWINRS	test
2	2	RES Doe	RES John	A	RES	RES test

Security Setup:

Security Groups (1/3) [Info](#)

Q

Filter security groups

↺

Actions

Export security groups to CSV

	Name	Security group ID	Security group name	VPC ID	Description	Owner	Inbound rules count
<input checked="" type="checkbox"/>	-	sg-0edb9a785b069c8ea	postgres-public-access	vpc-0842cd62b8c86c859	postgres-public-access	028191280483	3 Permission entries
<input type="checkbox"/>	-	sg-07f56fc4be3197b89	default	vpc-0842cd62b8c86c859	default VPC security gr...	028191280483	3 Permission entries
<input type="checkbox"/>	-	sg-02e85989bae844b19	default	vpc-0f807cda476542b22	default VPC security gr...	028191280483	2 Permission entries

aws

Services

Q Search

[Alt+S]

VPC

>

Security Groups

>

sg-0edb9a785b069c8ea - postgres-public-access

>

Edit inbound rules

Edit inbound rules

[Info](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules

[Info](#)

Security group rule ID	Type	Protocol	Port range	Source	Descr
sgr-083fb04b4a7ba2af6	All traffic	All	All	Custom	<div><div>Q</div><div>0.0.0.0/0</div><div>×</div></div>
sgr-0037b33febfa441b0	All TCP	TCP	0 - 65535	Custom	<div><div>Q</div><div>0.0.0.0/0</div><div>×</div></div>
sgr-0fa7e7789f68e7ad3	PostgreSQL	TCP	5432	Custom	<div><div>Q</div><div>0.0.0.0/0</div><div>×</div></div>

Add rule

VPC

>

Security Groups

>

sg-07f56fc4be3197b89 - default

>

Edit inbound rules

Edit inbound rules

[Info](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules

[Info](#)

Security group rule ID	Type	Protocol	Port range	Source	D
sgr-0e35d0d876f5ae5bc	PostgreSQL	TCP	5432	Custom	<div><div>Q</div><div>0.0.0.0/0</div><div>×</div></div>
sgr-0f6b601755eed08c2	All TCP	TCP	0 - 65535	Custom	<div><div>Q</div><div>sg-07f56fc4be3197b89</div><div>×</div></div>
sgr-026ad8476a619bd31	All traffic	All	All	Custom	<div><div>Q</div><div>0.0.0.0/0</div><div>×</div></div>

Add rule

glueadmin

Allows Glue to call AWS services on your behalf.

Summary

Creation date	ARN
August 10, 2023, 13:09 (UTC-06:00)	<code>arn:aws:iam::028191280483:role/glueadmin</code>
Last activity	Maximum session duration
17 hours ago	1 hour

- Permissions
- Trust relationships
- Tags
- Access Advisor
- Revoke sessions

Permissions policies (9) [Info](#)

You can attach up to 10 managed policies.

Simulate

Remove

Filter policies by property or policy name and press enter.

<input type="checkbox"/>	Policy name	Type
<input type="checkbox"/>	AdministratorAccess	AWS managed - job function
<input type="checkbox"/>	AmazonEC2FullAccess	AWS managed
<input type="checkbox"/>	AmazonRDSReadOnlyAccess	AWS managed
<input type="checkbox"/>	AmazonS3FullAccess	AWS managed
<input type="checkbox"/>	AmazonS3ReadOnlyAccess	AWS managed
<input type="checkbox"/>	AWSGlueServiceRole	AWS managed

Glue Setup:

Create a Connector using db connections, security groups, VPC from earlier steps

AWS Glue > Connectors > pgConnv13

pgConnv13 Edit Delete Create job

Connection details [Info](#)

Connector type	JDBC	Connection URL	jdbc:postgresql://database-13e.czhqbccm3aqo.us-east-1.rds.amazonaws.com:5432/postgres
Driver class name	-	Driver path	-
Username	postgres	Require SSL connection	false
Subnet	subnet-0ca4a3874a44d2d1b	Security groups	sg-02e85989bae844b19
Description	-	Created on	2023-08-17 12:17:43.589000
Last modified	2023-08-17 12:17:43.589000	Class name	-

Test the connector

Connections (4) [Info](#)

You can manage your connections or use a connection in a job.

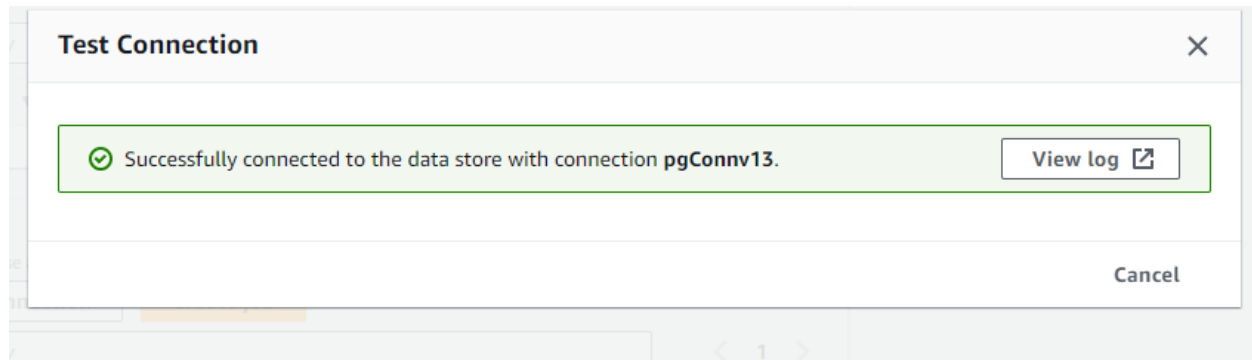
Actions ▲ Create connection Create job

View details
Delete
Edit
Test connection

property

< 1 >

	Type	Last modified
pgConnv13	JDBC	Aug 17, 2023
differentVpcPg	JDBC	Aug 16, 2023
pq2	JDBC	Aug 15, 2023



Create a Crawler using data sources from earlier steps

AWS Glue > Crawlers > Edit crawler

Step 1
Set crawler properties

Step 2
Choose data sources and classifiers

Step 3
Configure security settings

Step 4
Set output and scheduling

Step 5
Review and update

Review and update

Step 1: Set crawler properties Edit

Set crawler properties

Name	Description	Tags
craw	-	-

Step 2: Choose data sources and classifiers Edit

Data sources (1) [Info](#)
The list of data sources to be scanned by the crawler.

Type	Data source	Parameters
JDBC	postgres/public/%	-

Step 3: Configure security settings Edit

Configure security settings

IAM role	Security configuration	Lake Formation configuration
glueadmin	-	-

Step 4: Set output and scheduling Edit

Set output and scheduling

Database	Table prefix - optional	Schedule
pgtarget	-	On demand

Cancel Previous Update

AWS Glue > Crawlers

Crawlers

A crawler connects to a data store, progresses through a prioritized list of classifiers to determine the schema for your data, and then creates metadata tables in your data catalog.

Crawlers (1) [Info](#)
View and manage all available crawlers.

Filter crawlers

Last updated (UTC)
August 18, 2023 at 21:57:14 [Refresh](#) [Action](#) [Run](#) [Create crawler](#)

<input type="checkbox"/>	Name	State	Schedule	Last run	Last run timestamp	Log	Table changes from last r...
<input type="checkbox"/>	craw	Ready		Succeeded	August 17, 2023 at 22:31:43	View log	1 created

Create a Catalog, (this will mimic the CWINRS in/out tables)

AWS Glue

Getting started
ETL jobs
Visual ETL
Notebooks
Job run monitoring
Data Catalog tables
Data connections
Workflows (orchestration)
Data Catalog
Databases
Tables
Stream schema registries
Schemas
Connections
Crawlers
Classifiers
Catalog settings

One table successfully deleted
The following table is now deleted: "a (db: pgtarget)"

AWS Glue > Tables

Tables

A table is the metadata definition that represents your data, including its schema. A table can be used as a source or target in a job definition.

Tables (1)
View and manage all available tables.

Last updated (UTC)
August 18, 2023 at 19:51:29

Filter tables

<input type="checkbox"/>	Name	Database	Location	Classification	Deprecated	View data
<input type="checkbox"/>	postgres_public_veteran	pgtarget	postgres.public.veteran	postgresql	-	-

Clicking into the category table

AWS Glue > Tables > postgres_public_veteran

postgres_public_veteran

Last updated (UTC)
August 20, 2023 at 06:17:05

Version 5 (Current version)

Table overview | Data quality New

Table details | Advanced properties

Name postgres_public_veteran	Description -	Database pgtarget	Classification postgresql
Location postgres.public.veteran	Connection pgConnv13	Deprecated -	Last updated August 20, 2023 at 06:17:05
Input format -	Output format -	Serde serialization lib -	

Schema | Partitions | Indexes

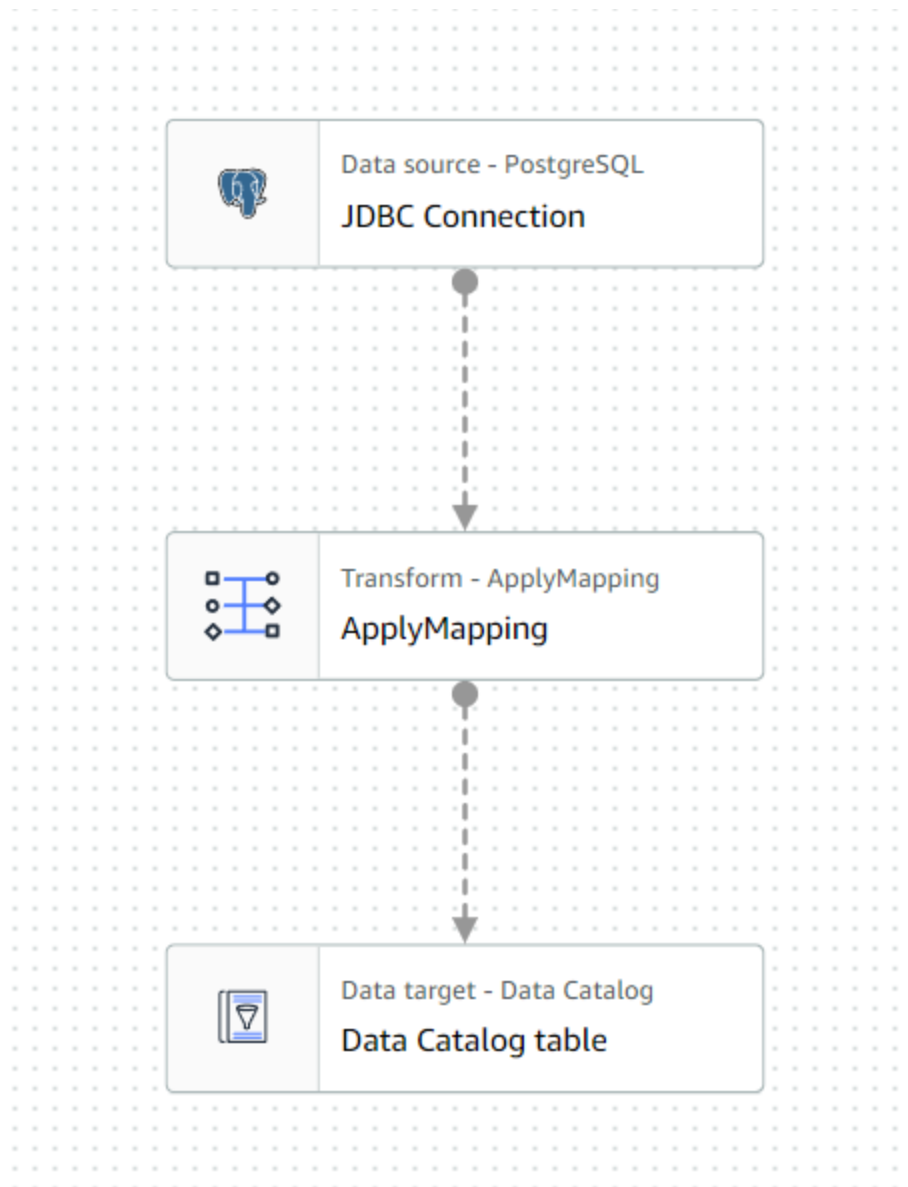
Schema (6)

View and manage the table schema.

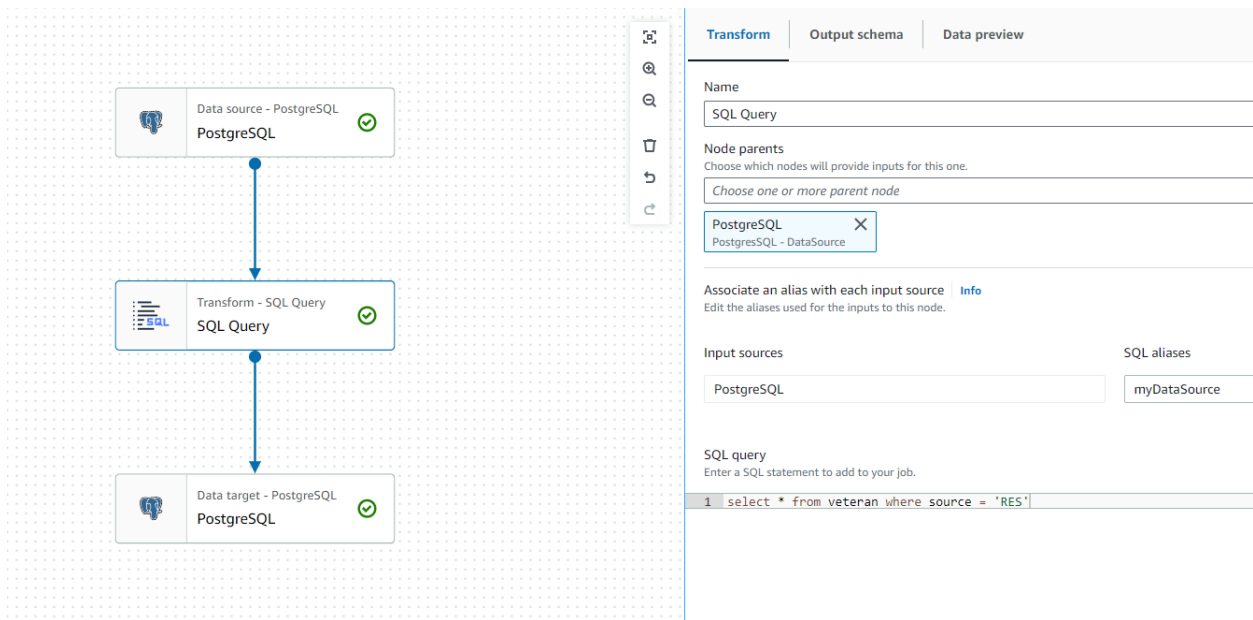
Filter schemas

#	Column name	Data type	Partition key	Comment
1	id	int	-	-
2	lastname	varchar	-	-
3	firstname	varchar	-	-
4	address	varchar	-	-
5	source	varchar	-	-
6	city	varchar	-	-

Create a ETL job using the connectors, crawlers, data catalog from earlier steps



Another ETL job version where we are filtering with the Source column



ETL job list

AWS Glue

ETL jobs

- Visual ETL
- Notebooks
- Job run monitoring
- Data Catalog tables
- Workflows (orchestration)

Data Catalog

- Databases
- Tables
- Stream schema registries
- Schemas
- Connections
- Crawlers
- Classifiers
- Catalog settings

Data Integration and ETL

- ETL jobs**
- Visual ETL
- Notebooks
- Job run monitoring
- Interactive Sessions
- Data classification tools
- Sensitive data detection

AWS Glue Studio

Create job

- ☒ Visual with a source and target
Start with a source, ApplyMapping transform, and target.
- ☐ Visual with a blank canvas
Author using an interactive visual interface.
- ☐ Spark script editor
Write or upload your own Spark code.
- ☐ Python Shell script editor
Write or upload your own Python shell script.
- ☐ Jupyter Notebook
Write your own code in a Jupyter Notebook for interactive development.
- ☐ Ray script editor
Write your own code to run on Ray.

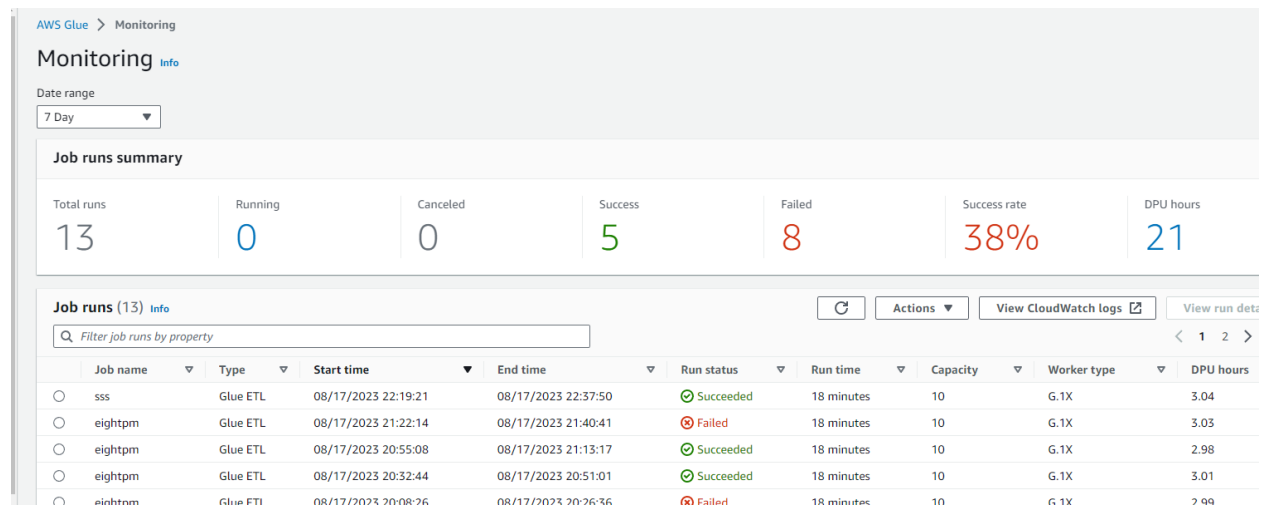
Source: Amazon S3 (JSON, CSV, or Parquet files stored in S3)

Target: Amazon S3 (S3 bucket by specifying a bucket path as the data target.)

Your jobs (3)

Job name	Type	Last modified	AWS Glue version
eightpm	Glue ETL	8/17/2023, 10:47:23 PM	4.0
pg13	Glue ETL	8/17/2023, 12:19:45 PM	4.0
test	Glue ETL	8/16/2023, 12:40:39 PM	4.0

ETL Job runs view



This demonstrates a spiked ETL job getting data from the eVA Sara Sync tables, filtering it out and sending the records into respective CWINRS/RES systems