

# Seamless Data Integration between AWS RDS Aurora MySQL and Amazon Redshift

Here's a combined and refined document summarizing the steps for setting up Zero ETL integration between Amazon Aurora MySQL and Amazon Redshift.

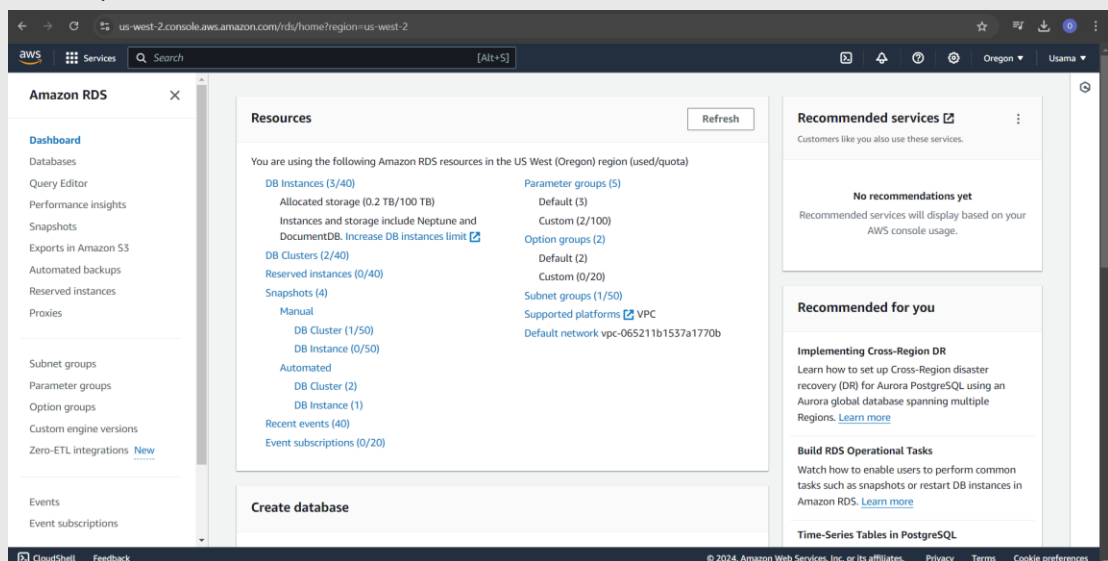
## Introduction to Zero ETL

AWS's Zero ETL feature enables seamless data integration between Amazon Aurora MySQL and Amazon Redshift without traditional ETL configurations, streamlining data management.

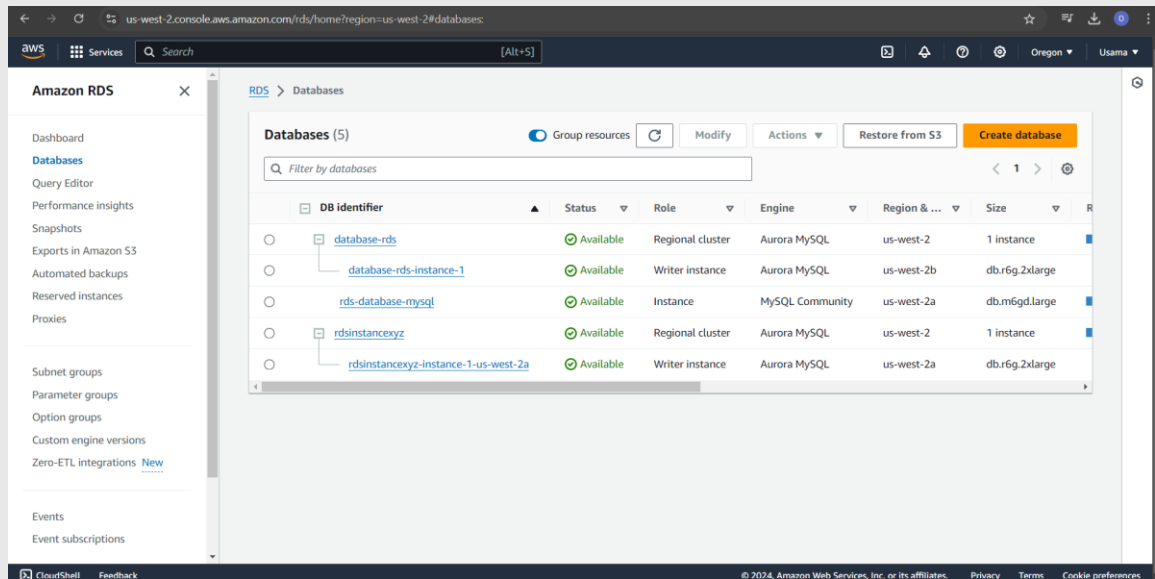
## Setting Up Aurora MySQL

### # Create Aurora MySQL Instance

Go to the RDS console. (Make sure you have to select the US West (Oregon)us-west-2)

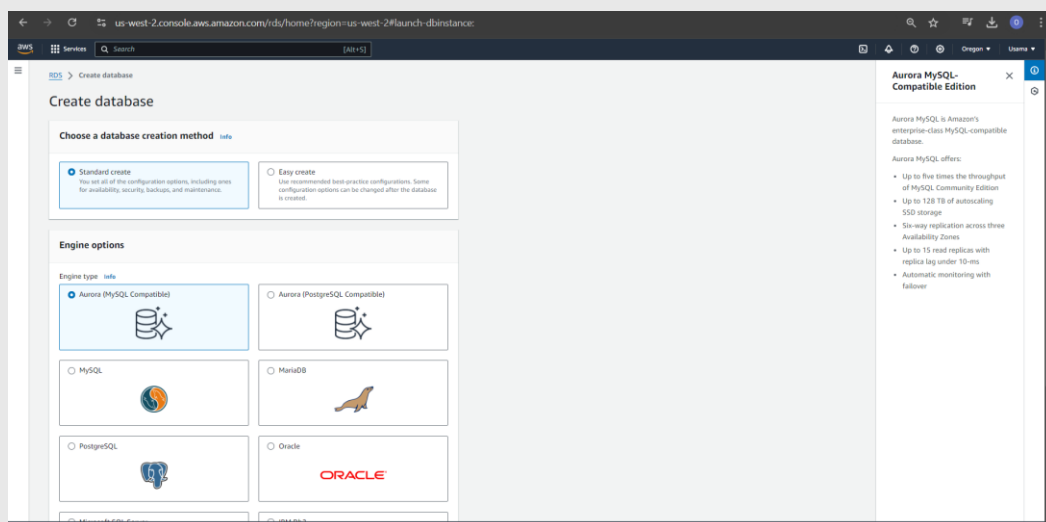


Click; "Database"



Click; Create Database

- Choose: Standard Create
- Choose: Engine option (Aurora (MySQL Compatible))



Choose Engine Version: Aurora MySQL 3.05.2 (compatible with MySQL 8.0.32) - default for major version 8.0

- Choose template, Production

Engine Version

Aurora MySQL 3.05.2 (compatible with MySQL 8.0.32) - default for major version 8.0

⚠ Parallel query is off by default. To enable it, use a DB instance parameter group with the `aurora_parallel_query` parameter enabled. [Learn more](#)

**Templates**

Choose a sample template to meet your use case.

☒ **Production**  
Use defaults for high availability and fast, consistent performance.

☐ **Dev/Test**  
This instance is intended for development use outside of a production environment.

**Settings**

**DB cluster identifier** [Info](#)  
Enter a name for your DB cluster. The name must be unique across all DB clusters owned by your AWS account in the current AWS Region.  
database-1  
The DB cluster identifier is case-insensitive, but is stored as all lowercase (as in "mydbcluster"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

**▼ Credentials Settings**

**Master username** [Info](#)  
Type a login ID for the master user of your DB instance.  
admin  
1 to 32 alphanumeric characters. The first character must be a letter.

**Credentials management**  
You can use AWS Secrets Manager or manage your master user credentials.

☒ **Managed in AWS Secrets Manager - most secure**  
RDS generates a password for you and manages it throughout its lifecycle using AWS Secrets Manager.

☐ **Self managed**  
Create your own password or have RDS create a password that you manage.

- Enter a name for your DB cluster.

**Settings**

**DB cluster identifier** [Info](#)  
Enter a name for your DB cluster. The name must be unique across all DB clusters owned by your AWS account in the current AWS Region.  
database-1  
The DB cluster identifier is case-insensitive, but is stored as all lowercase (as in "mydbcluster"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

**▼ Credentials Settings**

**Master username** [Info](#)  
Type a login ID for the master user of your DB instance.  
admin  
1 to 32 alphanumeric characters. The first character must be a letter.

**Credentials management**  
You can use AWS Secrets Manager or manage your master user credentials.

☐ **Managed in AWS Secrets Manager - most secure**  
RDS generates a password for you and manages it throughout its lifecycle using AWS Secrets Manager.

☒ **Self managed**  
Create your own password or have RDS create a password that you manage.

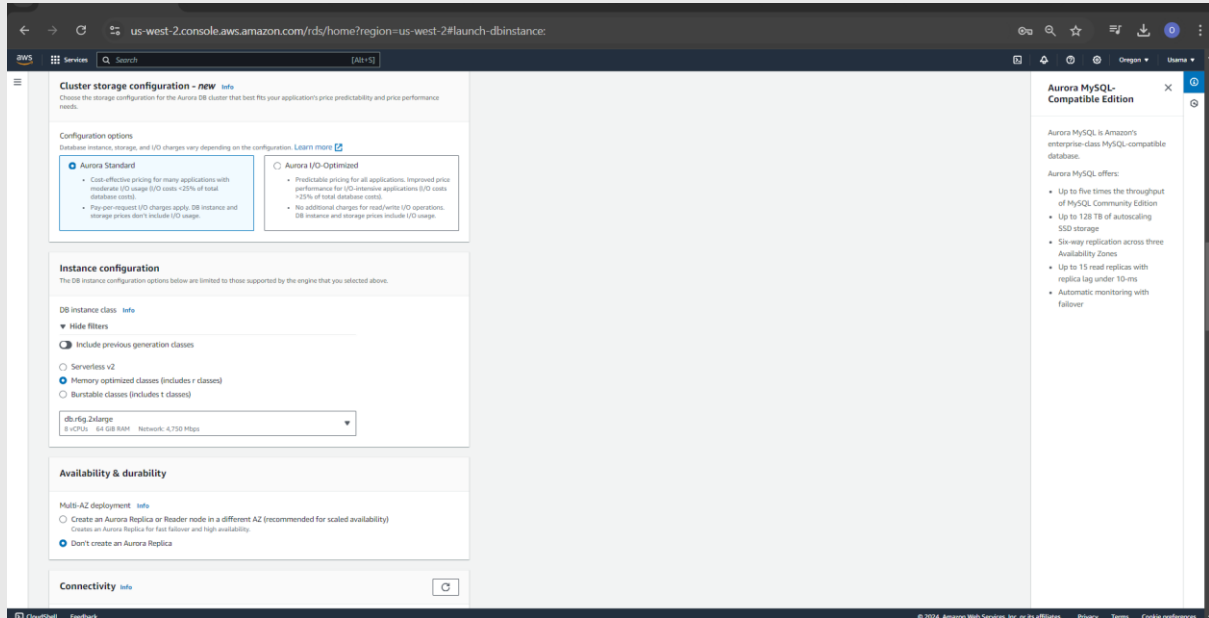
☐ **Auto generate password**  
Amazon RDS can generate a password for you, or you can specify your own password.

**Master password** [Info](#)  
Password strength: [Details](#)  
Minimum constraints: At least 8 printable ASCII characters. Can't contain any of the following symbols: / " ' @  
Confirm master password [Info](#)

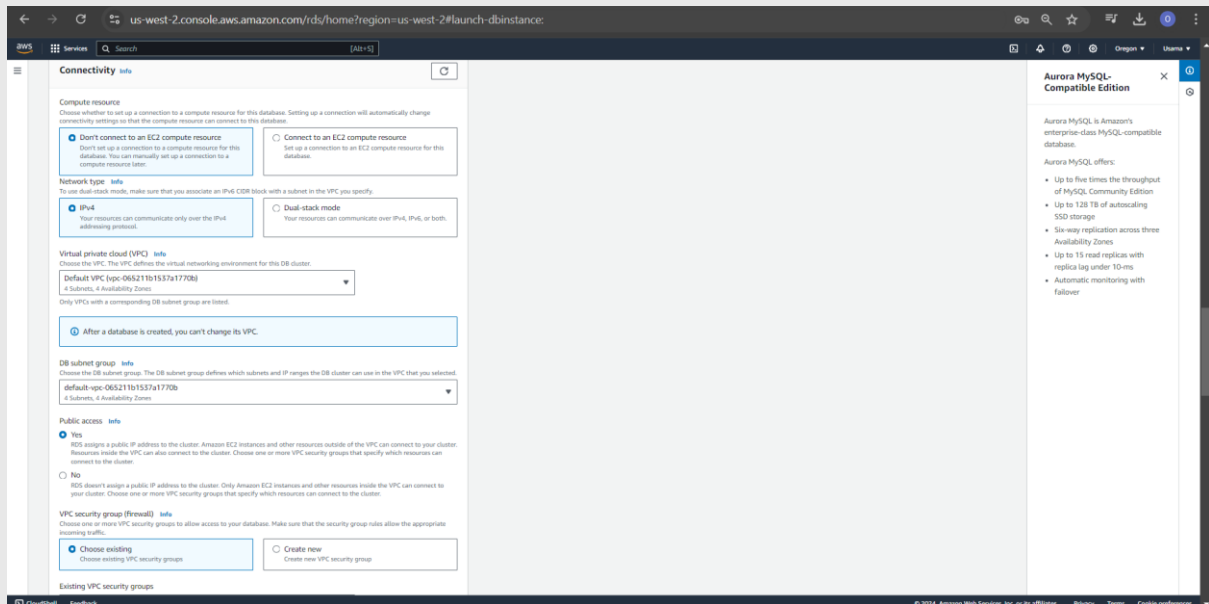
**Cluster storage configuration - new** [Info  
Choose the storage configuration for the Aurora DB cluster that best fits your application's price predictability and price performance needs.](#)

**Configuration options**  
Database instance, storage, and I/O charges vary depending on the configuration. [Learn more](#)

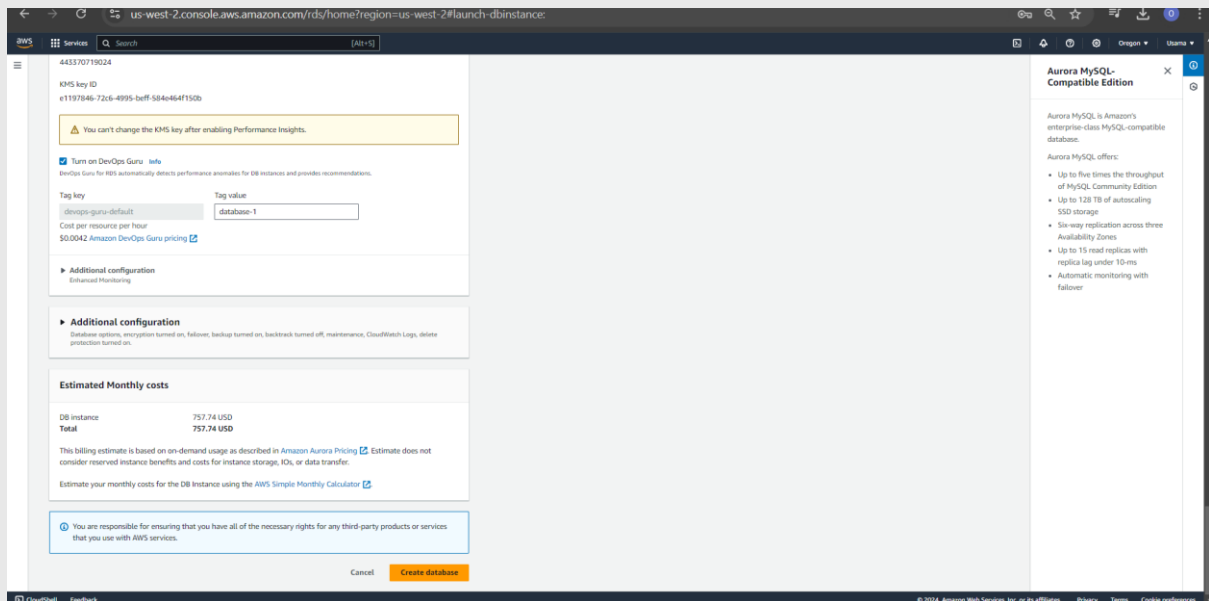
- In credentials management select self managed so you can create your own password.



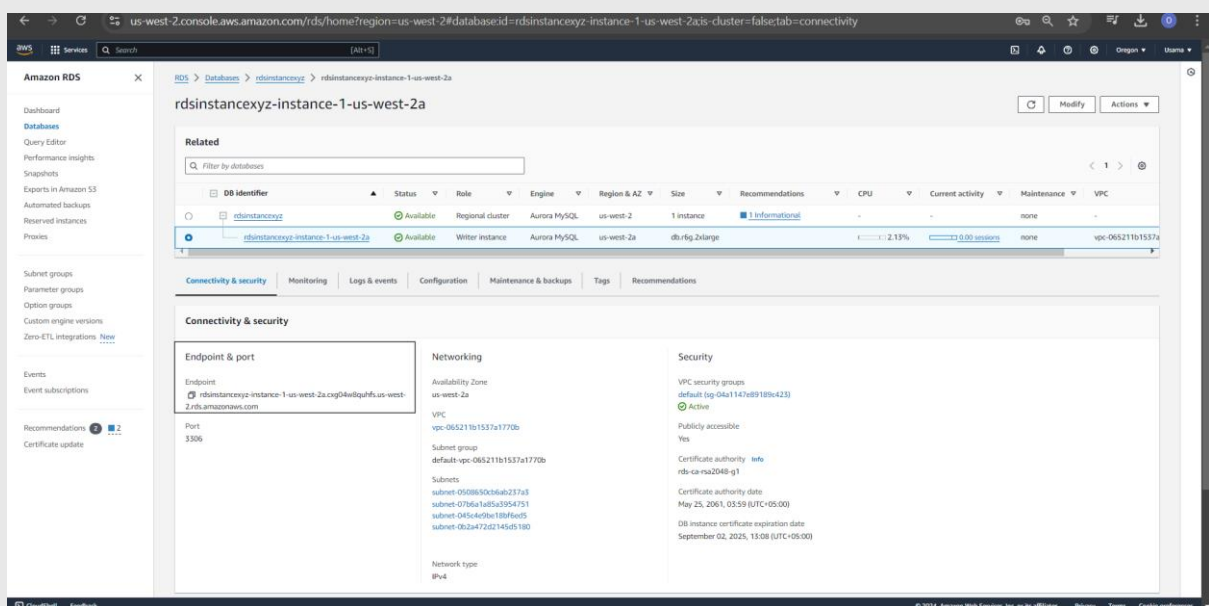
- Connectivity Set as by default



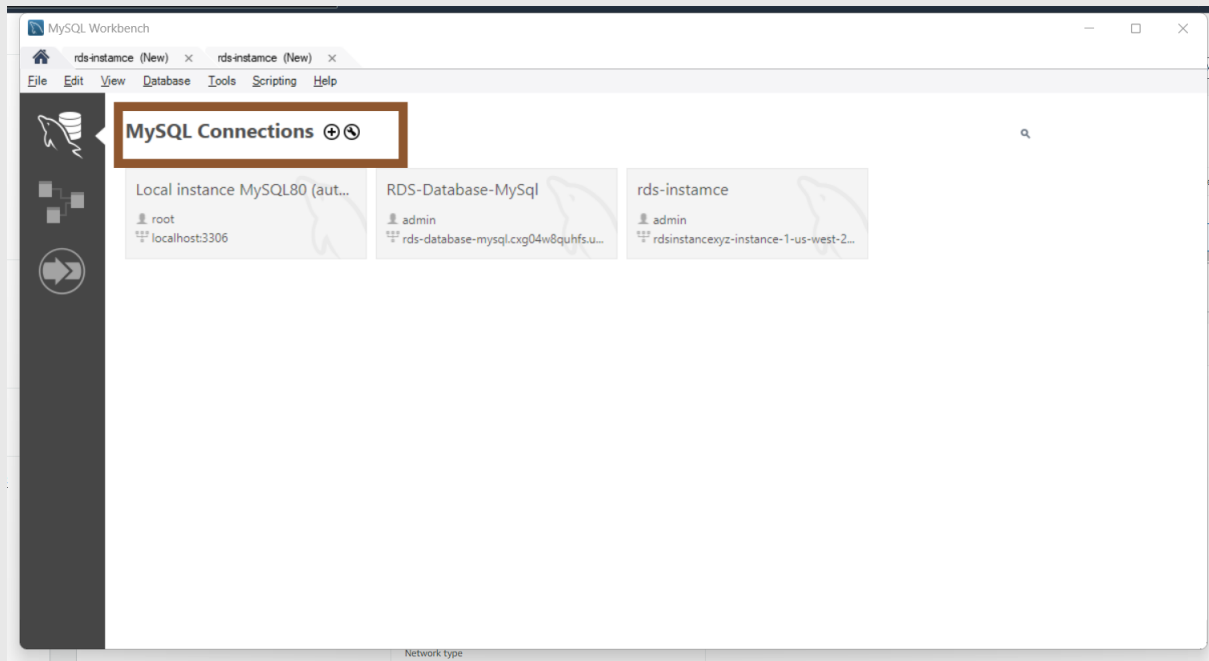
- Configuration options; By Default
- Instance configuration; By Default
- Availability & durability; Choose > Don't create an Aurora Replica
- Public access chooses; Yes
- Remaining setting are as default



- After Successfully created the database copy the Endpoint



- Open Mysql Workbench and click on plus icon

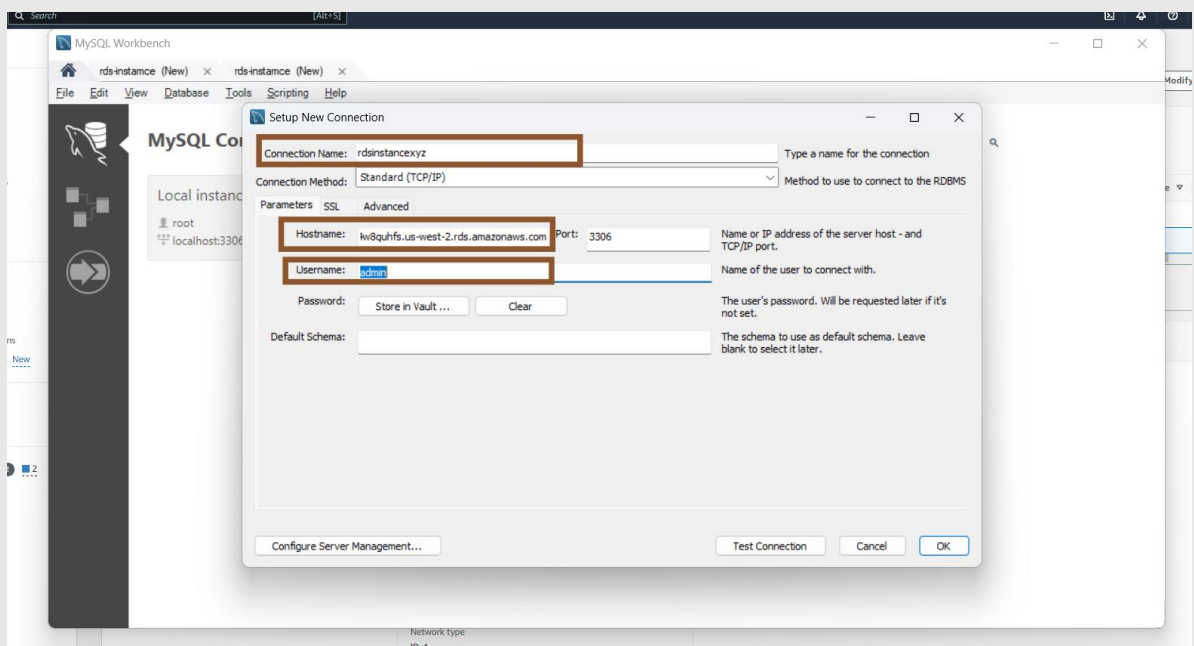


Enter Connection name which you have been created in the RDS.

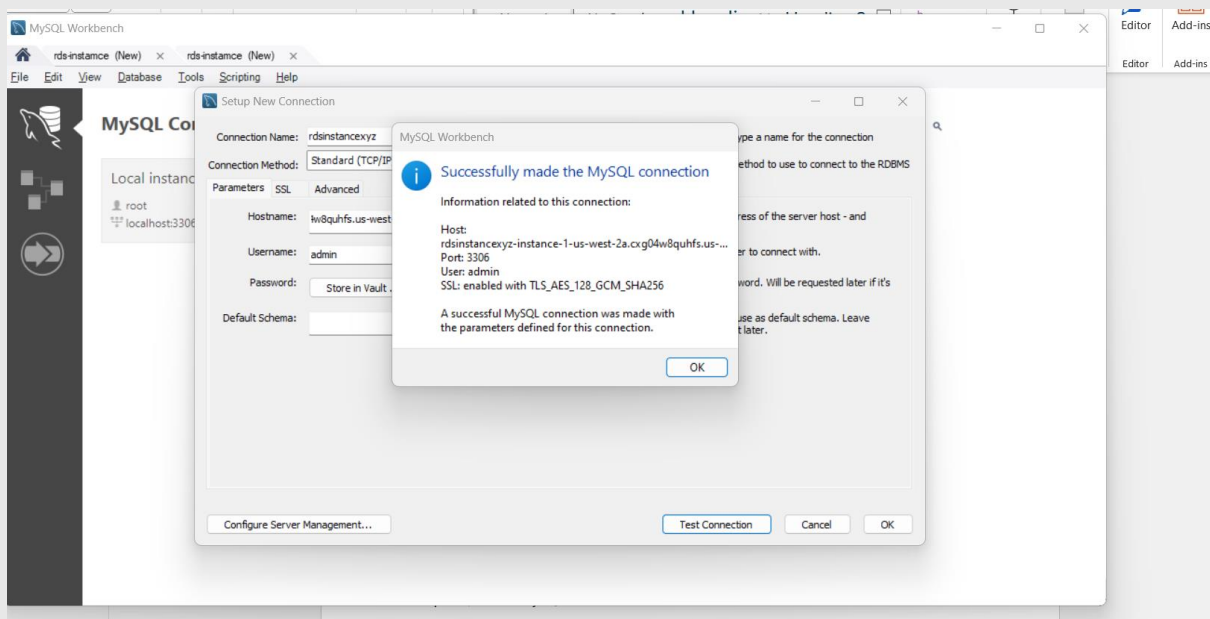
In hostname field enter the RDS endpoint

User name: Add RDS user name and password

Click on test connection

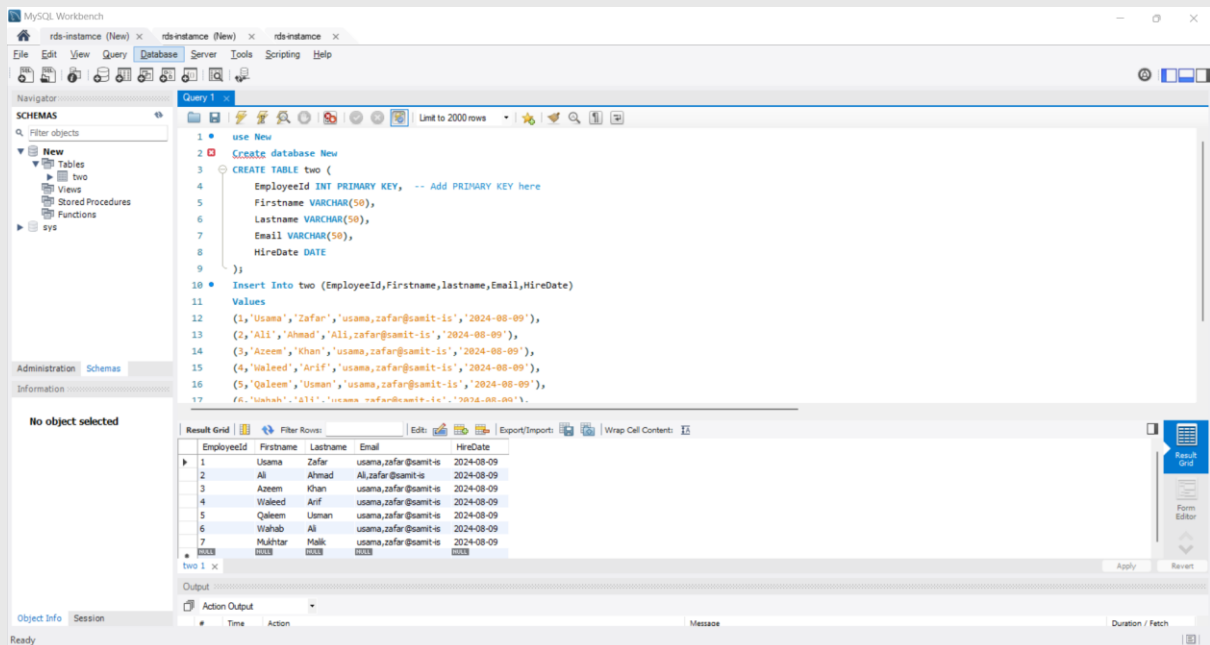


After clicking on test connection, it will show a pop up screen successfully made the MySQL connection



## Create Database in MySQL

- Create Table
- Add data in table



## Setup Amazon Redshift.

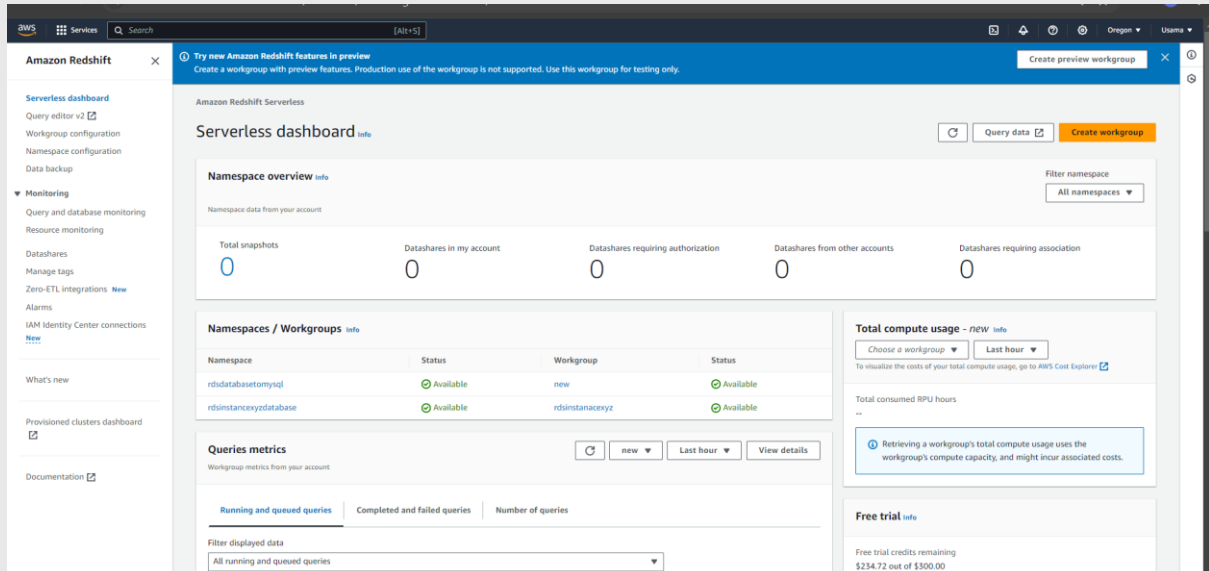
## # Create a Workgroup.

Go to the Amazon Redshift console.

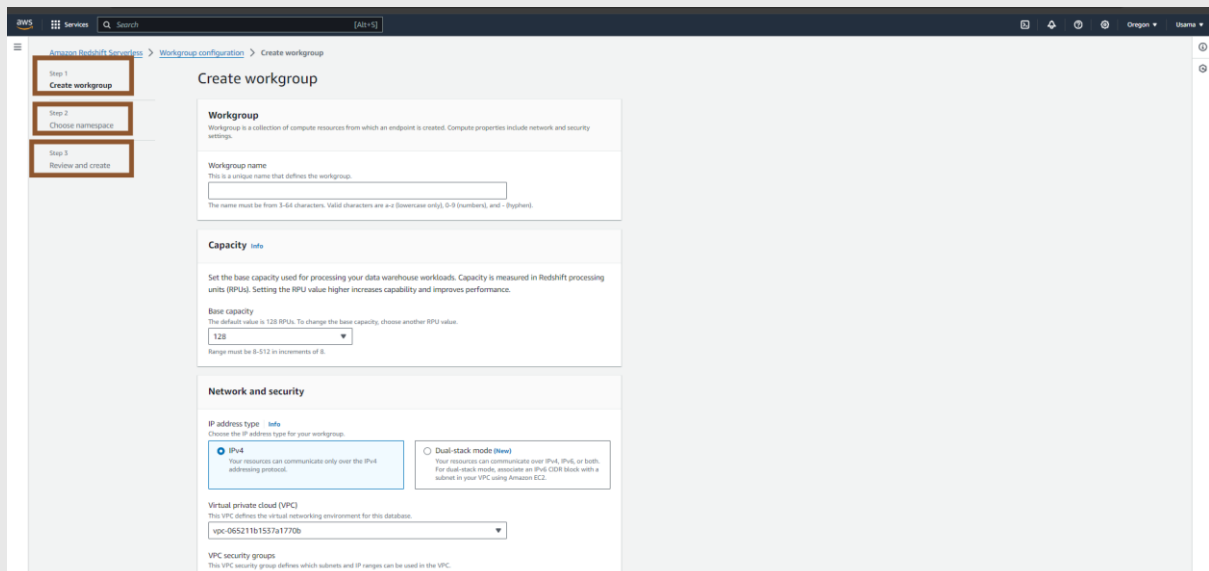
Create a new workgroup.

Configure the workgroup with appropriate settings.

Create namespaces

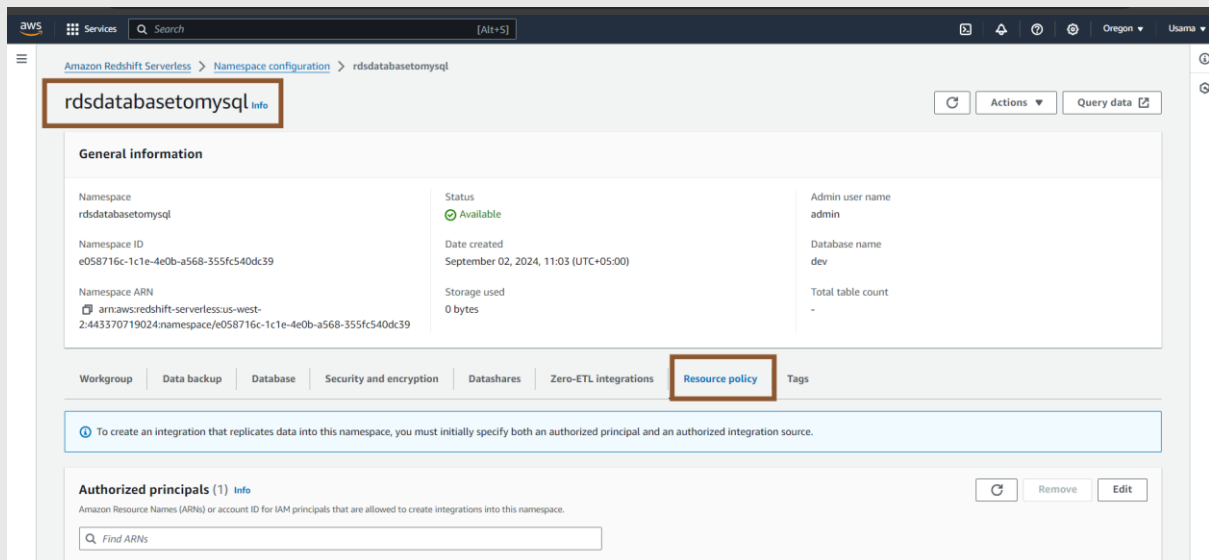


After creating the workgroup this screen will appear



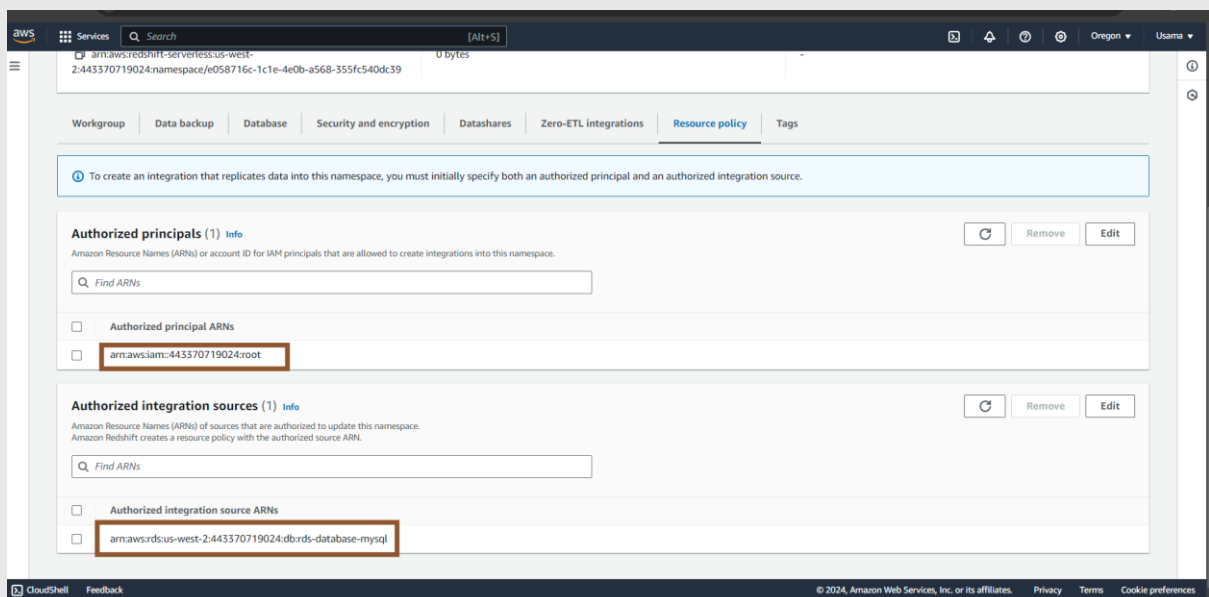
Click on Resource policy





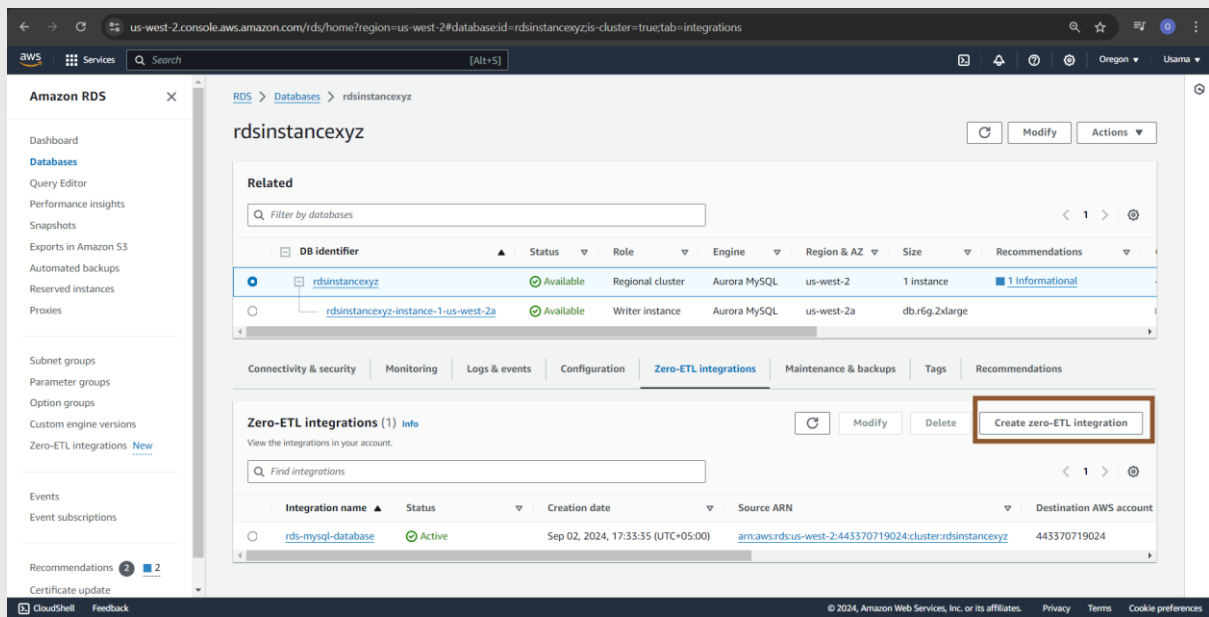
Add Authorized integration source ARNs (From RDS Amazon Resource Name (ARN))

Add Authorized principal ARNs

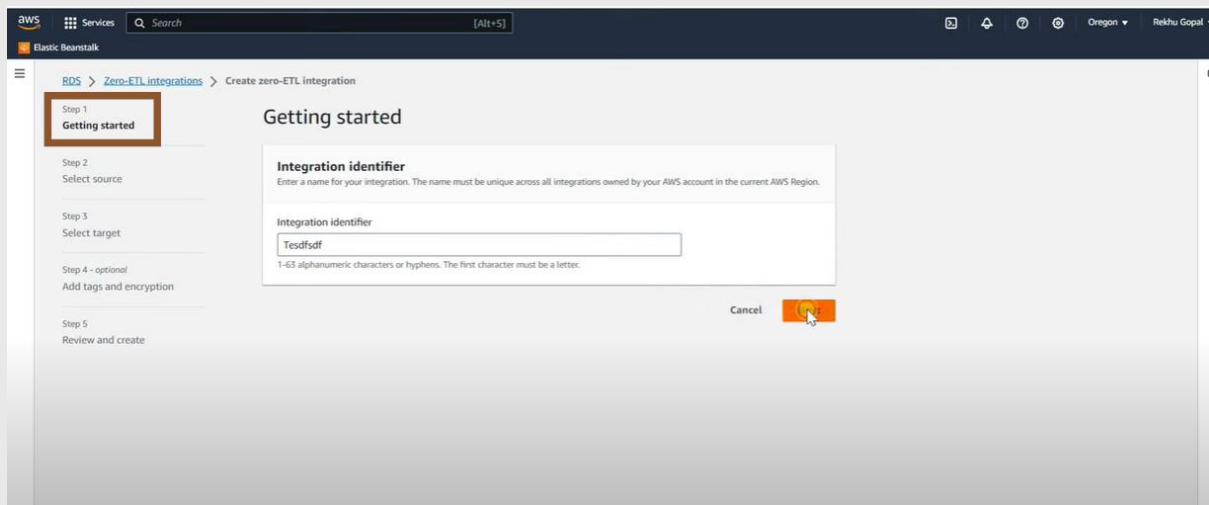


## Enable Zero ETL Integration.

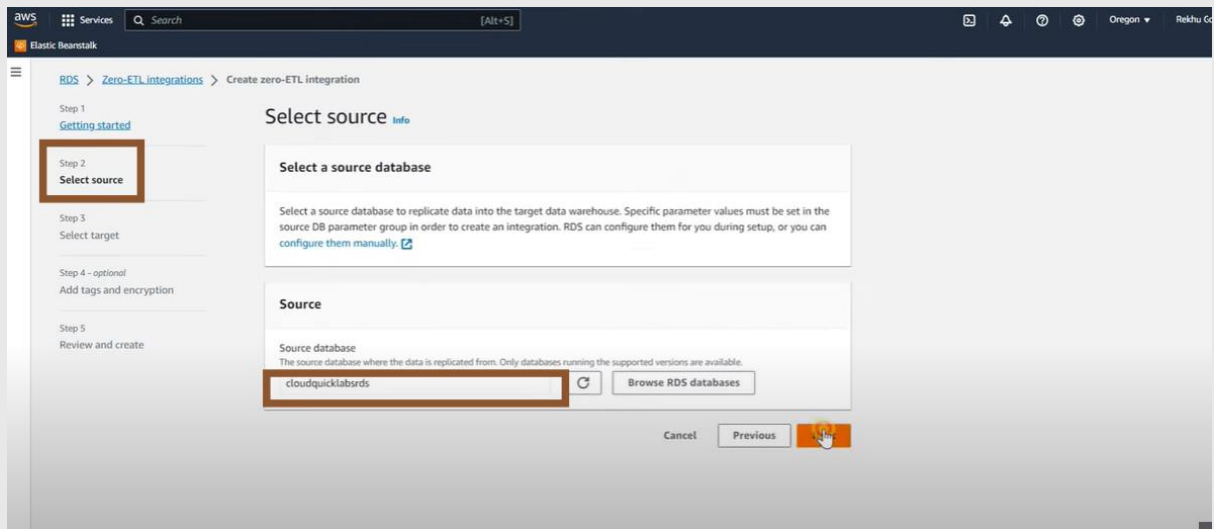
- Go to the ETL configuration page.
- Click on; "Create Zero ETL Integration"



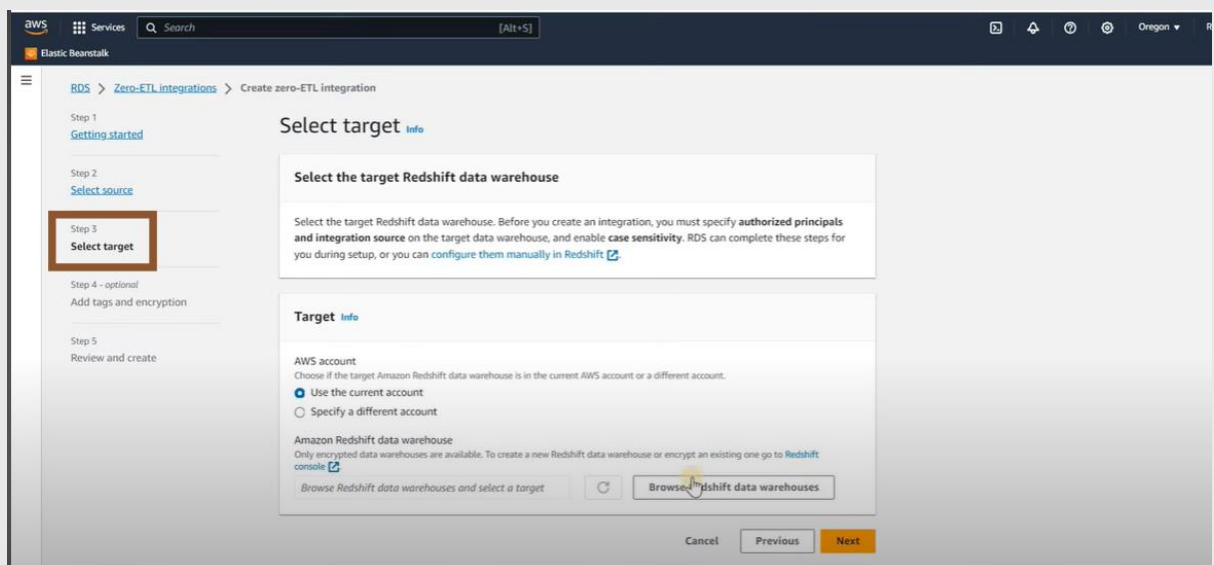
## Getting Started: Add Integration identifier name



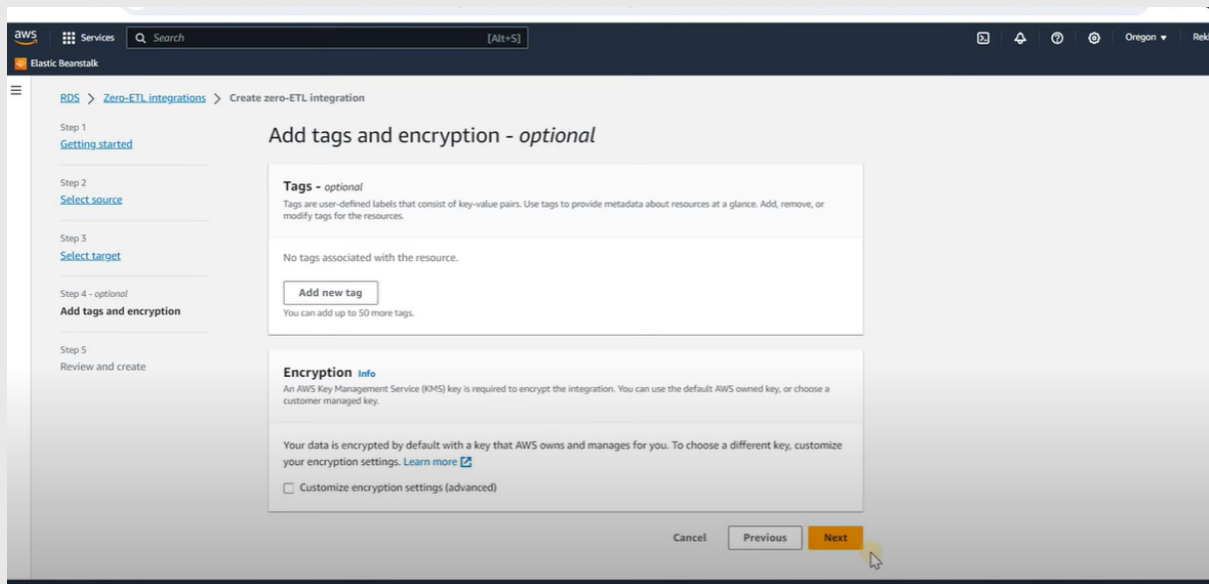
## Select Source: Brows RDS source database



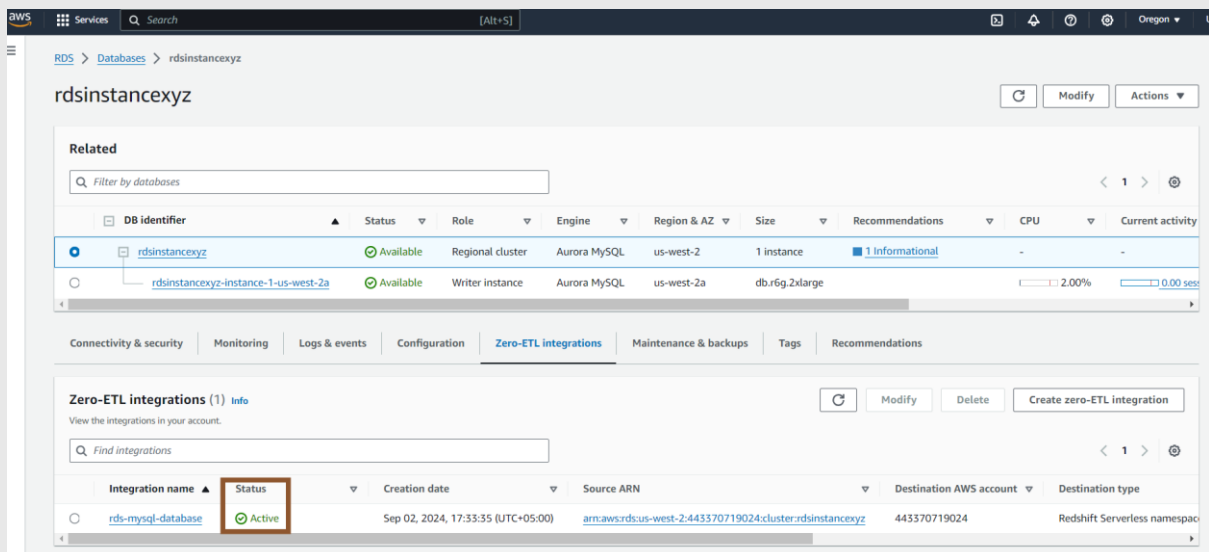
## Select Target: Add targeted Redshift data source



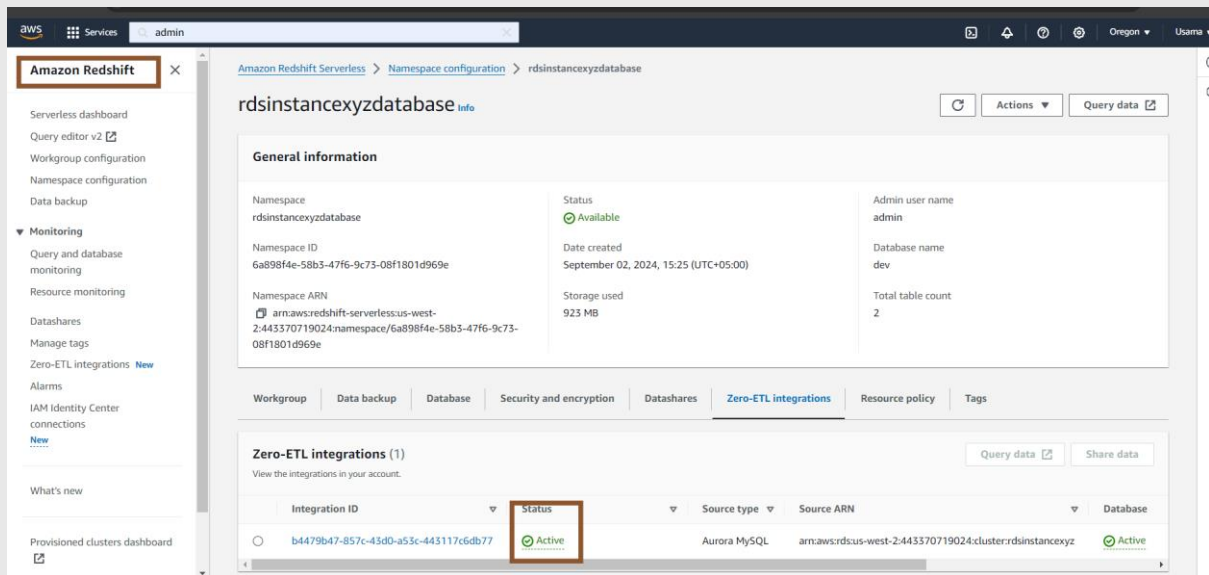
## Add tags and encryption (optional)



After successfully created it will show the active status  
Confirm the configuration status changes to; "Available"; before proceeding.

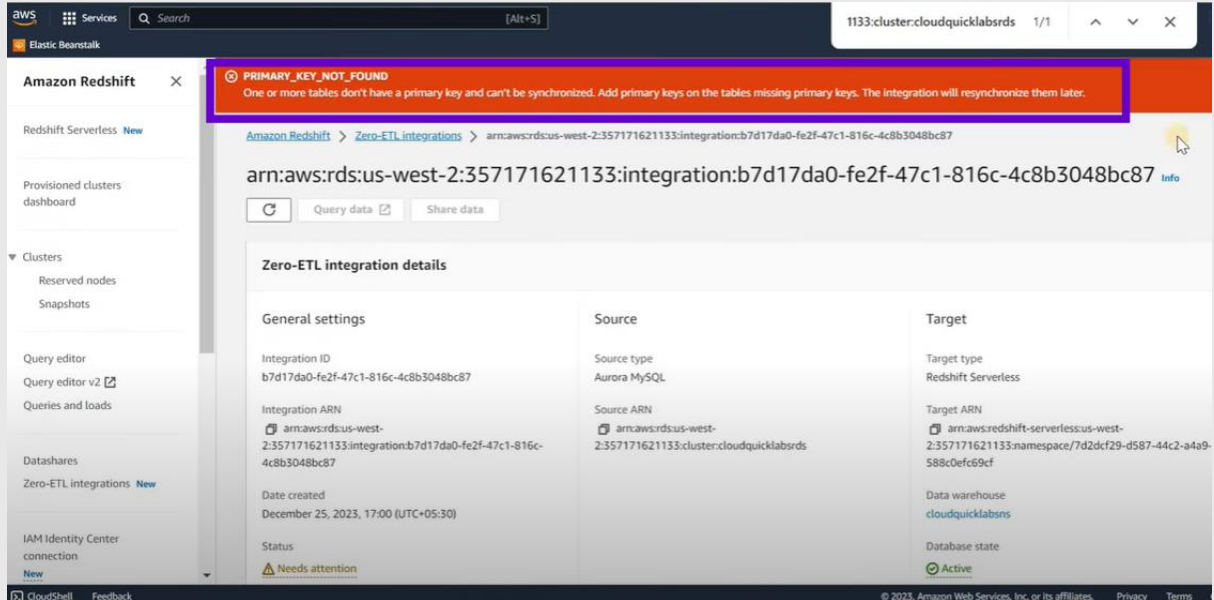


So, the integration we have created in the RDS it will appear in the Redshift.



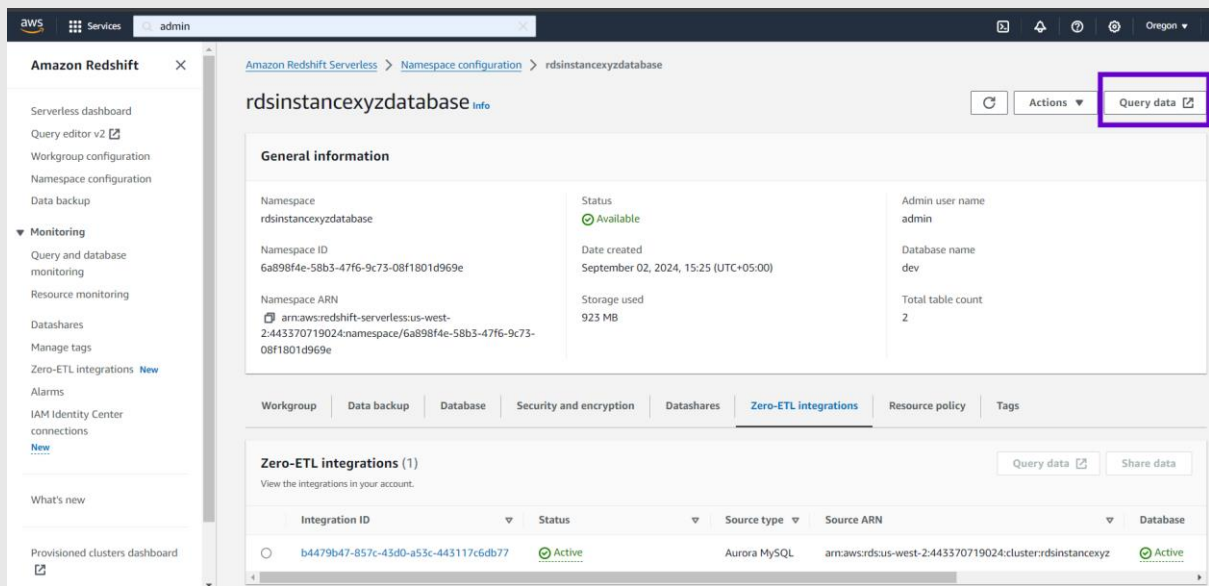
## Handle Primary Key Requirement.

When creating tables, ensure each table has a primary key to avoid errors.



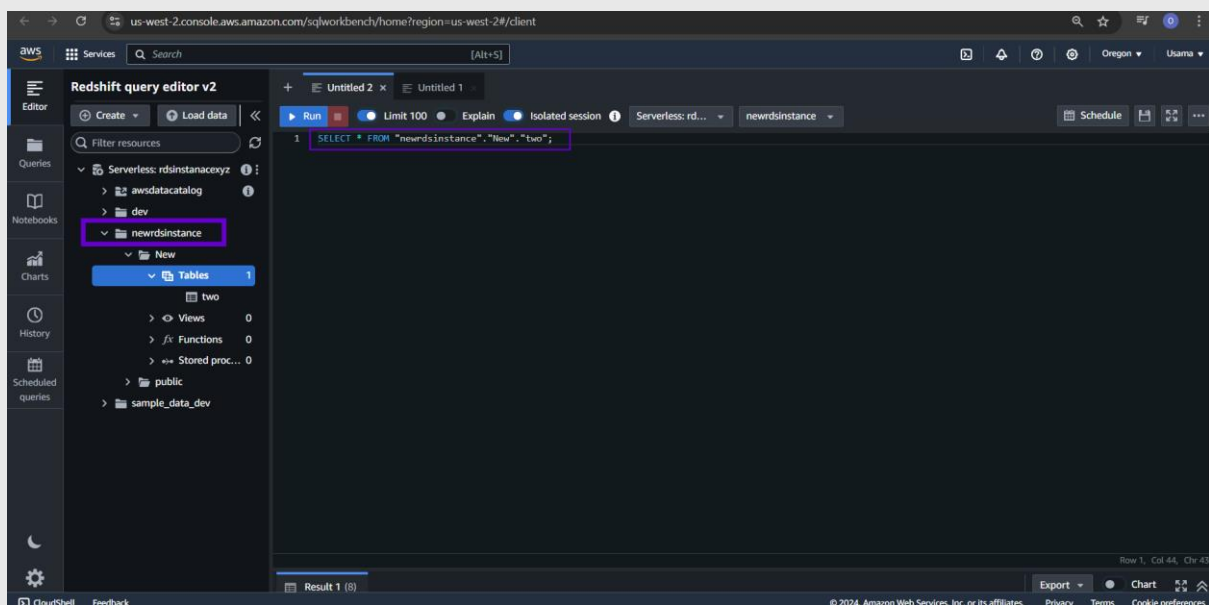
## Query Editor

When you click on query data it will appear on next tab.

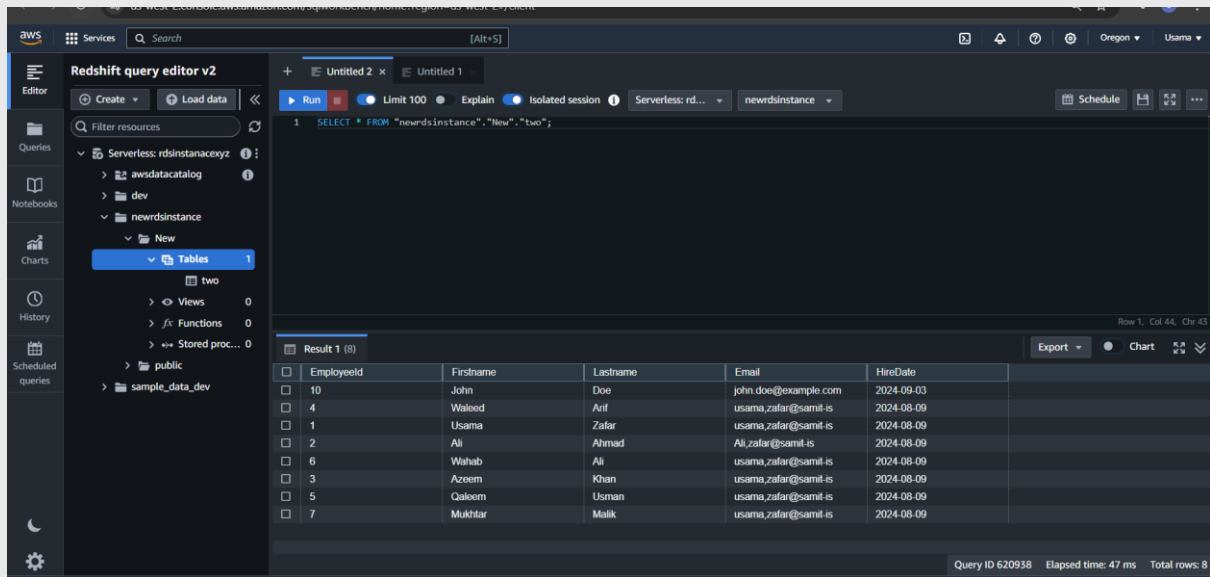


## Show Data base in Query Editor:

In the left panel, the database is displayed. Click on the dropdown arrow next to the database, and the tables created in MySQL Workbench will appear.



To view the table run query in RDS query editor



## Final Verification.

Monitor data synchronization and ensure it reflects in real-time or near real-time.

