Setting up a Data Warehouse using Amazon Redshift Serverless

Use case

Any Company Events is ready for you to create an Amazon Redshift Serverless data warehouse. After you create a namespace and workgroup, you will create a schema and tables that will loaded with TICKIT data. Finally, you will create a view and query a sample of the data.’

In this lab, you create an Amazon Redshift Serverless namespace and workgroup, schema, and tables. You load the tables with sample data and then query the data.

Objectives

By the end of this lab, you should be able to do the following:

- Create a data warehouse with Amazon Redshift Serverless.
- Create a schema.
- Create a table.
- Load the table with sample data.

On the Get started with Amazon Redshift Serverless page, configure the following.

In the Configuration section, select Customize settings.

In the Namespace section, configure:

- For Target namespace, enter `lab-namespace`.
- For Admin user credentials, select Customize admin user credentials.
  - For Admin user name, enter `dbadmin`.
  - For Admin password, select Manage admin credentials in AWS Secrets Manager.

In the Associated IAM roles section, configure:

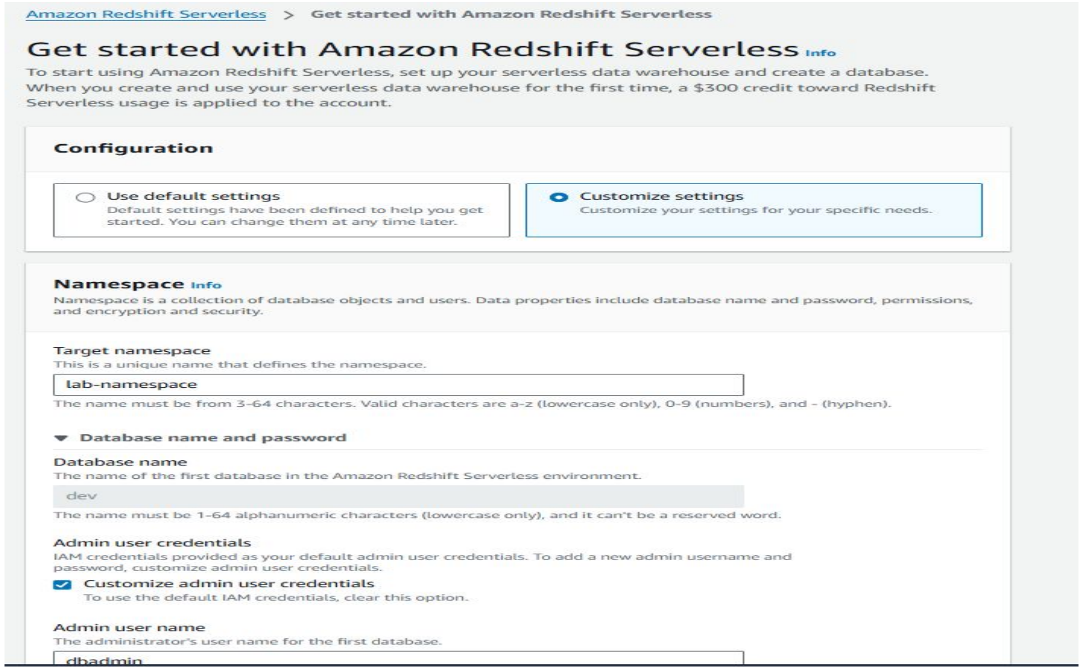
- Choose **Associate IAM role**.
- Select the IAM Role that contains the string RedshiftAccessRole in name.
- Choose **Associate IAM roles**.
- Select RedshiftAccessRole to make default IAM role.
- Choose **Set default** .
- Select Make default.
- In the Make default IAM role prompt, choose **Confirm**.

In the Workgroup section, configure:

- For Workgroup name, enter `lab-workgroup`.
- For Virtual private cloud (VPC), select `LabVPC`.
- For VPC security groups, select the security group with Enable access to redshift in description.
- For Subnet, select Private Subnet 1, Private Subnet 2, Private Subnet 3.

Choose **Save configuration**.
A new window is displayed with a message *It may take a few minutes to complete. After completing the setup, you can work with your data*.
Note: Wait for the status to display Completed.

Choose **Continue**.



Amazon Redshift Serverless > Get started with Amazon Redshift Serverless

Get started with Amazon Redshift Serverless Info

To start using Amazon Redshift Serverless, set up your serverless data warehouse and create a database. When you create and use your serverless data warehouse for the first time, a $300 credit toward Redshift Serverless usage is applied to the account.

Configuration

○ Use default settings
Default settings have been defined to help you get started. You can change them at any time later.

● Customize settings
Customize your settings for your specific needs.

Namespace Info

Namespace is a collection of database objects and users. Data properties include database name and password, permissions, and encryption and security.

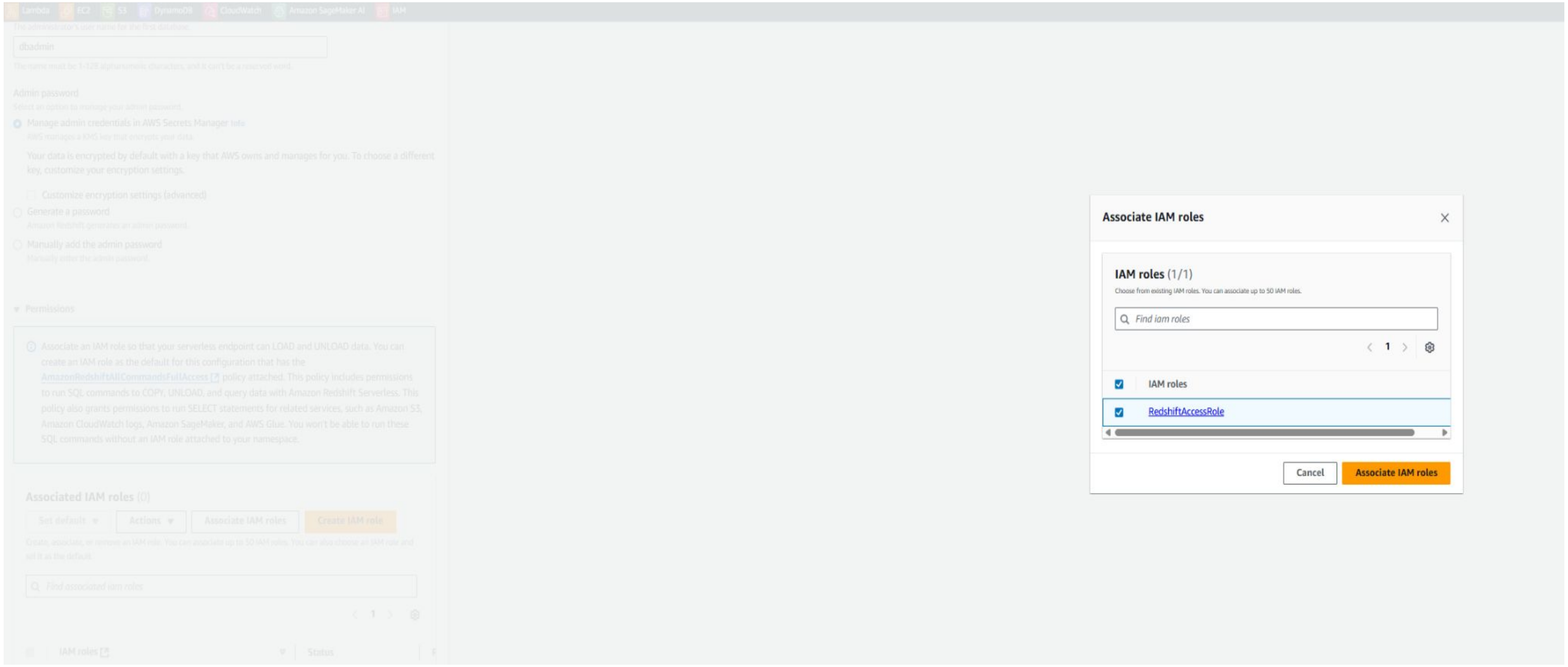
Target namespace
This is a unique name that defines the namespace.
lab-namespace
The name must be from 3-64 characters. Valid characters are a-z (lowercase only), 0-9 (numbers), and - (hyphen).

▼ Database name and password

Database name
The name of the first database in the Amazon Redshift Serverless environment.
dev
The name must be 1-64 alphanumeric characters (lowercase only), and it can't be a reserved word.

Admin user credentials
IAM credentials provided as your default admin user credentials. To add a new admin username and password, customize admin user credentials.
✓ Customize admin user credentials
To use the default IAM credentials, clear this option.

Admin user name
The administrator's user name for the first database.
dbadmin



The administrator's user name for the first database.

dbadmin

The name must be 1-128 alphanumeric characters, and it can't be a reserved word.

Admin password

Select an option to manage your admin password.

- [✓ Manage admin credentials in AWS Secrets Manager] Info
  AWS manages a KMS key that encrypts your data.
  Your data is encrypted by default with a key that AWS owns and manages for you. To choose a different key, customize your encryption settings.
  - ☐ Customize encryption settings (advanced)
- Generate a password
  Amazon Redshift generates an admin password.
- Manually add the admin password
  Manually enter the admin password.

▼ Permissions

Associate an IAM role so that your serverless endpoint can LOAD and UNLOAD data. You can create an IAM role as the default for this configuration that has the AmazonRedshiftAllCommandsFullAccess policy attached. This policy includes permissions to run SQL commands to COPY, UNLOAD, and query data with Amazon Redshift Serverless. This policy also grants permissions to run SELECT statements for related services, such as Amazon S3, Amazon CloudWatch logs, Amazon SageMaker, and AWS Glue. You won't be able to run these SQL commands without an IAM role attached to your namespace.

Associated IAM roles (0)

Create, associate, or remove an IAM role. You can associate up to 50 IAM roles. You can also choose an IAM role and set it as the default.

	IAM roles	Status

Associate IAM roles

IAM roles (1/1)

Choose from existing IAM roles. You can associate up to 50 IAM roles.

	IAM roles
✓	RedshiftAccessRole

Create a schema

In this task, you create a schema in the current database.

A schema is a namespace that contains named database objects such as tables, views, and user-defined functions (UDFs). A database can contain one or multiple schemas, and each schema belongs to only one database. Two schemas can have different objects that share the same name.

Note: Amazon Redshift automatically creates a schema called public for every new database. When you don’t specify the schema name while creating database objects, the objects go into the public schema.

Connect to workgroup

13. On the Serverless dashboard page, choose **Query data**.
    The query editor now opens in a new browser tab.
14. In the Redshift query editor v2 pane, choose the vertical ellipses (three dots) at Serverless:lab-workgroup.
15. Choose Create connection.
16. On the Connect to lab-workgroup window prompt, choose AWS Secrets Manager.
17. Select Choose a secret and choose redshift!lab-namespace-dbadmin.
18. Choose **Create connection**.
19. Command: To create a new schema named US_SALES, run the following query:
    The following command either creates the *US_SALES* schema for the dev database, or does nothing and returns a message if it already exists.
    This sql command defines a new schema for the *dev* database.

```
Create schema if not exists us_sales;
```

To create a table in *us_sales* schema, run the following command:

CREATE TABLE us_sales.supplier (

SupplierID int,

City varchar (255)

);

To Insert

INSERT INTO us_sales.supplier VALUES (781, 'San Jose'), (990, 'Palo Alto');

To view the data inserted into a table, run the following command:

```
SELECT * from us_sales.supplier;
```

Load sample data from Amazon S3

In this task, you use the COPY command to load data into the table.

The COPY command uses the Amazon Redshift massively parallel processing (MPP) architecture to read and load data in parallel from files in Amazon S3, from an Amazon DynamoDB table, or from text output from one or more remote hosts.

Note: The COPY command appends the new input data to any existing rows in the table. The maximum size of a single input row from any source is 4 MB.

Copy edit: To load data into the table from the Amazon S3 bucket, replace the S3_BUCKET_PATH placeholder value with the S3BucketPath value listed to the left of these instructions. Then, run the following query:

```
COPY us_sales.supplier

FROM 'S3_BUCKET_PATH/supplier.csv'
```

24.

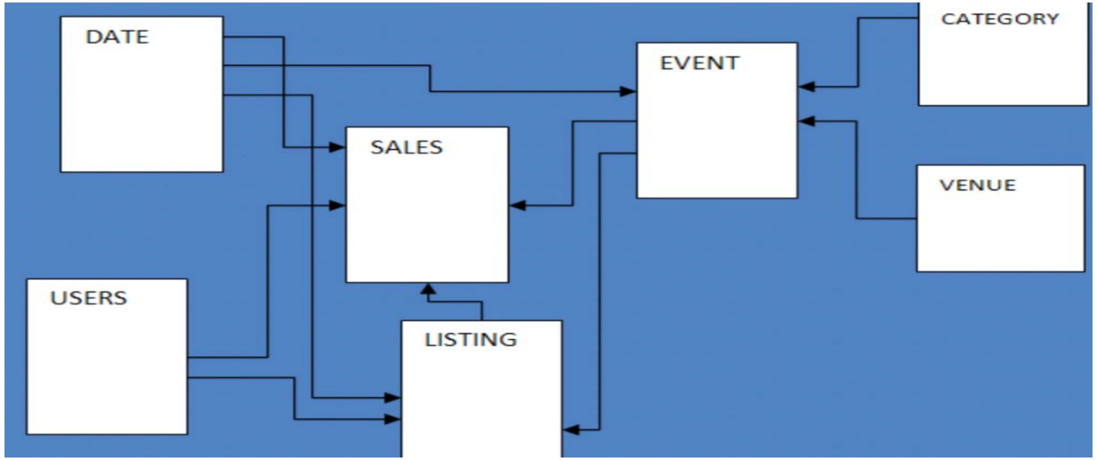
```
iam_role default csv;

SELECT * FROM us_sales.supplier;
```

Ingest and query TICKIT data

AnyCompany Events wants to build database application that helps analysts track sales activity from TICKIT website where users buy and sell tickets online for sporting events, shows, and concerts.

In this task, you load the TICKIT dataset from Amazon S3 to Redshift Serverless database and build queries to help analysts to use this information to provide incentives to buyers and sellers who frequent the site, to attract new users, and to drive advertising and promotions.



To create tables in the dev database, individually copy and run the following create table statements:

```
CREATE TABLE us_sales.users (userid INTEGER NOT NULL DISTKEY SORTKEY, username CHAR(8), firstname VARCHAR(30), lastname VARCHAR(30), city VARCHAR(30), state CHAR(2), email VARCHAR(100), phone CHAR(14), likesports BOOLEAN, liketheatre BOOLEAN, likeconcerts BOOLEAN, likejazz BOOLEAN, likeclassical BOOLEAN, likeopera BOOLEAN, likerock BOOLEAN, likevegas BOOLEAN, likebroadway BOOLEAN, likemusicals BOOLEAN);
```

Create all the other tables

```
CREATE TABLE us_sales.venue (venueid SMALLINT NOT NULL DISTKEY SORTKEY, venuename VARCHAR(100), venuecity VARCHAR(30), venuestate CHAR(2), venueseats INTEGER);
```

```
CREATE TABLE us_sales.category (catid SMALLINT NOT NULL DISTKEY SORTKEY, catgroup VARCHAR(10), catname VARCHAR(10), catdesc VARCHAR(50));
```

```
CREATE TABLE us_sales.date (dateid SMALLINT NOT NULL DISTKEY SORTKEY, caldate DATE NOT NULL, day CHAR(3) NOT NULL, week SMALLINT NOT NULL, month CHAR(5) NOT NULL,
qtr CHAR(5) NOT NULL, year SMALLINT NOT NULL, holiday BOOLEAN DEFAULT('N'));
```

```
CREATE TABLE us_sales.event (eventid INTEGER NOT NULL DISTKEY, venueid SMALLINT NOT NULL, catid SMALLINT NOT NULL, dateid SMALLINT NOT NULL SORTKEY, eventname
VARCHAR(200), starttime TIMESTAMP);
```

```
CREATE TABLE us_sales.event (eventid INTEGER NOT NULL DISTKEY, venueid SMALLINT NOT NULL, catid SMALLINT NOT NULL, dateid SMALLINT NOT NULL SORTKEY, eventname
VARCHAR(200), starttime TIMESTAMP);
```

```
CREATE TABLE us_sales.listing (listid INTEGER NOT NULL DISTKEY, sellerid INTEGER NOT NULL, eventid INTEGER NOT NULL, dateid SMALLINT NOT NULL SORTKEY, numtickets
SMALLINT NOT NULL, priceperticket DECIMAL(8,2), totalprice DECIMAL(8,2), listtime TIMESTAMP);
```

```
CREATE TABLE us_sales.sales (salesid INTEGER NOT NULL, listid INTEGER NOT NULL DISTKEY, sellerid INTEGER NOT NULL, buyerid INTEGER NOT NULL, eventid INTEGER NOT
NULL, dateid SMALLINT NOT NULL SORTKEY, qtysold SMALLINT NOT NULL, pricepaid DECIMAL(8,2), commission DECIMAL(8,2), saletime TIMESTAMP);
```

To load TICKIT dataset from the Amazon S3 bucket, replace the S3_BUCKET_PATH placeholder value with the S3BucketPath created ex s3://labstack-502883f7-b4ce-49fe-bc70-c4f0be-databucket-afjnmqcyqoxv/tickitdb. Then, run the following query:

```
copy us_sales.users  from 'S3_BUCKET_PATH/allusers.csv'  iam_role default csv;

copy us_sales.venue  from 'S3_BUCKET_PATH/venue_pipe.txt'  iam_role default delimiter '|';

copy us_sales.category  from 'S3_BUCKET_PATH/category_pipe.txt'  iam_role default delimiter '|';

copy us_sales.date  from 'S3_BUCKET_PATH/date2008.csv'  iam_role default csv;

copy us_sales.event  from 'S3_BUCKET_PATH/allevents_pipe.txt'  iam_role default delimiter '|' timeformat 'YYYY-MM-DD HH:MI:SS' ;

copy us_sales.listing  from 'S3_BUCKET_PATH/listings_pipe.txt'  iam_role default delimiter '|';

   copy us_sales.sales  from 'S3_BUCKET_PATH/sales_tab.txt'  iam_role default delimiter '\t' timeformat 'MM/DD/YYYY HH:MI:SS' ;
```