Introduction to Amazon Redshift

SPL-86 - Version 2.1.1

Introduction

This lab provides a basic understanding of Amazon Redshift. It demonstrates the basic steps required to get started with Redshift including:

- Creating a Redshift cluster
- Loading data into a Redshift cluster
- Performing queries against the data in cluster

Topics covered

By the end of this lab you will be able to:

- Launch a Redshift cluster
- Connect an SQL client to the Amazon Redshift cluster

- Load data from an S3 bucket into the Amazon Redshift cluster
- Run queries against data stored in Amazon Redshift

Amazon Redshift

Amazon Redshift is a fast, fully managed data warehouse that makes it simple and cost-effective to analyze all your data using standard SQL and your existing Business Intelligence (BI) tools.

Amazon S3

Amazon Simple Storage Service (Amazon S3) makes it simple and practical to collect, store, and analyze data - regardless of format - all at massive scale. S3 is object storage built to store and retrieve any amount of data from anywhere - web sites and mobile apps, corporate applications, and data from IoT sensors.

Other AWS Services

During this lab, you may receive error messages when performing actions beyond the steps in this lab guide. These messages will not impact your ability to complete the lab. We recommend you remain within the steps provided by these lab instructions.

Prerequisites

Familiarity with relational databases and SQL concepts would be beneficial.

Task 1: Launch an Amazon Redshift Cluster

In this task, you will launch an Amazon Redshift cluster. A cluster is a fully managed **data warehouse** that consists of a set of compute nodes. Each cluster runs an Amazon Redshift engine and contains one or more databases.
When you launch a cluster, one of the options you specify is the **node type**. The node type determines the CPU, RAM, storage capacity, and storage drive type for

each node. Node types are available in different sizes. Node size and the number of nodes determine the total storage for a cluster.

- In the **AWS Management Console**, on the **Services** menu, click **Amazon Redshift**.

You can also type in the search box to select the AWS Service (eg Redshift) that you wish to use.

- Click **Launch cluster** to open the Redshift Cluster Creation Wizard.
- On the **CLUSTER DETAILS** page, configure:
  - **Cluster identifier**:
  - **Database name**:
  - **Database port:**
  - **Master user name**:
  - **Master user password**:
  - **Confirm password**:
- Click **Continue**.

The **NODE CONFIGURATION** page is displayed. This is where you can select the **Node type** and the number of nodes in your cluster. For this lab, you will be using the default configuration of a single-node cluster.

- Click **Continue**.

The **ADDITIONAL CONFIGURATION** page will be displayed. This is where you can configure cluster settings to control encryption, security and networking.

- At the **ADDITIONAL CONFIGURATION** page, configure:
  - **Choose a VPC:** Select the other VPC (*not* the Default VPC)
  - **VPC security groups:** *Redshift Security Group*
  - **Available roles:** *Redshift-Role*

The role grants permission for Amazon Redshift to read data from Amazon S3.

- Leave all other settings at their default value and click **Continue**.

The **REVIEW** page displays information about the cluster that you are about to launch.

You may ignore the warning about the free trial.

- Click **Launch cluster** at the bottom of the screen.
- Click **Close**.

The cluster will take a few minutes to launch. Please continue with the labs steps. There is no need to wait.

- Click the name of your cluster (**lab**).

The cluster configuration will be displayed. Spend a few minutes looking at the properties, but do not click any links.

- **Cluster Properties**: Contains information about the Cluster including: Name, Type, Node Type, number of Nodes, Zone location, Time and version of the creation as well as other information.
- **Cluster Status**: Allows you to see the current status of the cluster whether it is available or not and also whether it is currently **In Maintenance Mode**.
- **Cluster Database Properties**: Contains information on the Endpoint, which is the DNS address of the cluster, and the port number on which the database accepts connections. These are required when you want to

create SQL connections. It also lets you know whether the cluster has a public IP address that can be accessed from the public internet. The JDBC URL and ODBC URL contain the URLs to connect to the cluster via a java database connection or an Oracle database connection client.

- **Backup, Audit Logging and Maintenance**: Contains information on how many days the automated snapshots are retained, whether they are automatically copied to another region, and whether logging is enabled on the cluster.
- **Capacity Details**: Contains information about the data warehouse node type, number of EC2 Compute Units per node, memory, disk storage, I/O performance as well as the processor architecture of the node type.
- **SSH Ingestion Settings**: Contains information about the Public Key of the cluster as well as the Public and Private IP addresses of the node.

Task 2: Launch Pgweb to Communicate with your Redshift Cluster

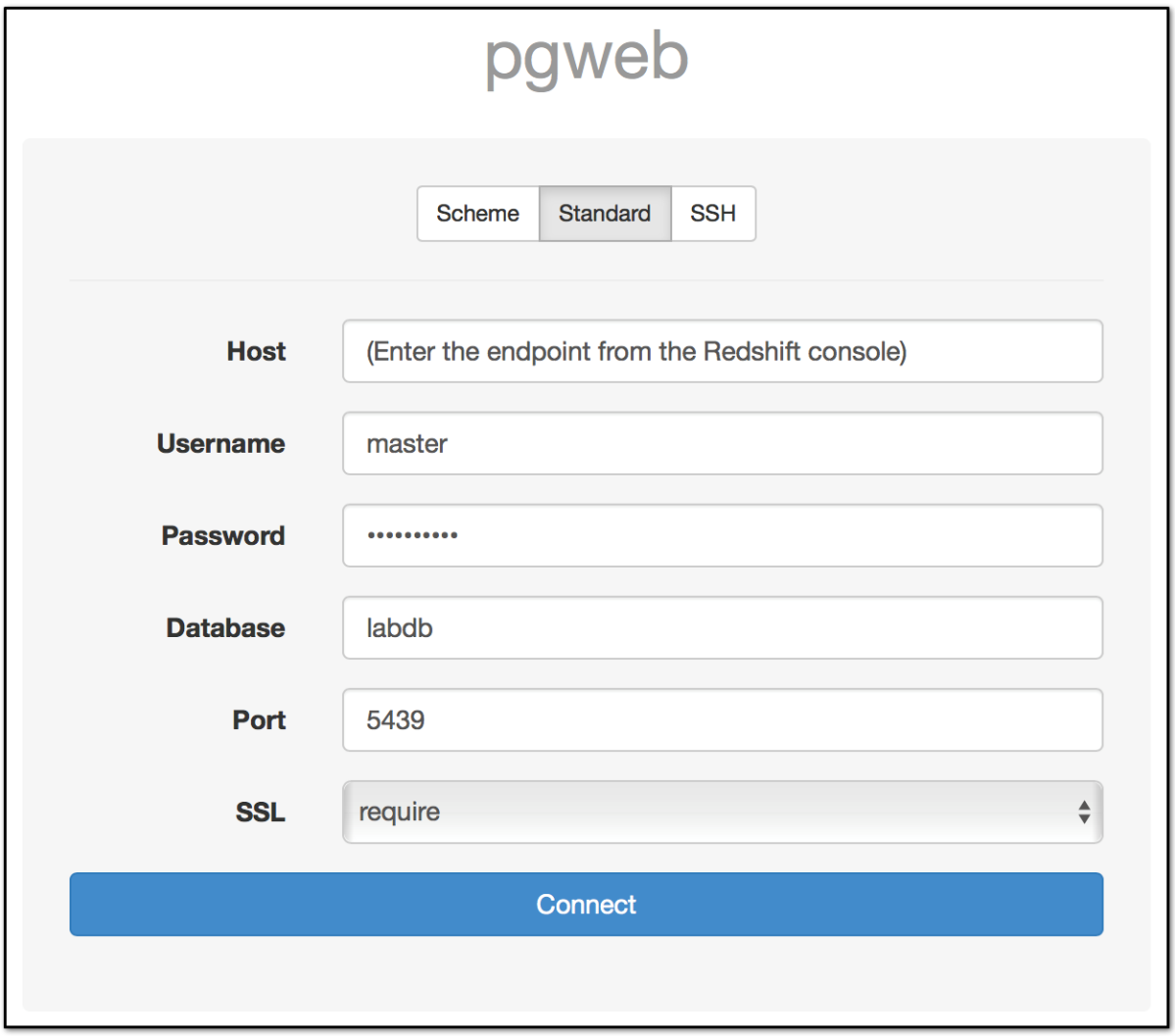
Amazon Redshift can be used via industry-standard SQL. To use Redshift, you require an **SQL Client**that provides a user interface to type SQL. Any SQL client that supports JDBC or ODBC can be used with Redshift.
For this lab, you will use a web application called **Pgweb**, which provides a friendly SQL interface to Redshift. The application is written for PostgreSQL, which is why it is called *Pgweb*. The SQL used in Amazon Redshift is based upon PostgreSQL.

- To the left of the instructions you are currently reading, copy the IP Address shown as **pgweb**.

The blue button will also copy values shown in this side panel.

- Open a **new tab** in your web browser, paste the IP Address and hit Enter.

You will be presented with a login page for *pgweb*:



- Return to your browser tab showing the Redshift console.

- Click the **Endpoint** displayed at the top of the window and Copy it to your clipboard. It will be similar to: *lab.c3uxk4gpn3xo.us-west-2.redshift.amazonaws.com:5439*

If the endpoint is not yet displayed, it is because your cluster is still being created. Click the refresh icon every 30 seconds until the *Endpoint* appears.

- Return to the web browser tab with **pgweb**.
- Enter the following information:

  - **Host:** Paste the Endpoint that you copied, but **remove from the end**
  - **Username**:
  - **Password**:
  - **Database**:
  - **Port**: (Note, this is different to the default)
    If cannot change the **Port** value, set **SSL** to **disable** and try again.
- Click **Connect**.

If you receive a **too many colons** error, double-check your **Host**. You will need to remove from the end of the Host name.

Pgweb will connect to your Amazon Redshift database. If you are not able to connect to the database, double-check the above parameters.

Task 3: Create a Table

In this task, you will execute SQL commands to create a table in Redshift.

- Copy this SQL command and paste it into Pgweb.

```
CREATE TABLE users (
  userid INTEGER NOT NULL,
  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
);
```

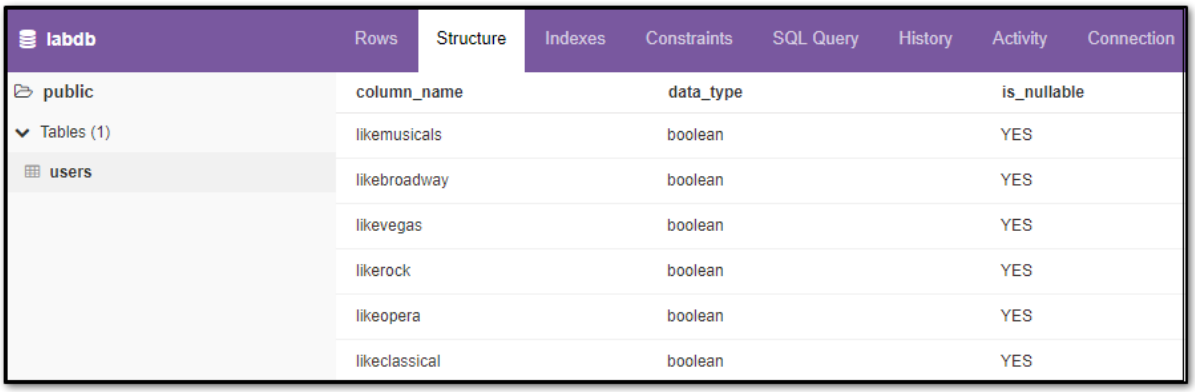
This command will create a table called **users**. It contains name, address and details about the type of music that the user likes.

- Click the **Run Query** button to execute the SQL script.

In the left navigation pane, below **Tables**, you should now see a **users** table.
You may ignore the *No records found* message because no records were queried.

- Click the **users** table, then click the **Structure** tab.

The structure of the table is displayed:



Task 4: Load Sample Data from Amazon S3

Amazon Redshift can import data from Amazon S3. Various file formats are supported, fixed-length fields, comma-separated values (CSV) and custom delimiters. The data for this lab is pipe-separated (|).

- In Pgweb, click the **SQL Query** tab at the top of the window.
- Delete the existing query, then paste this SQL command into Pgweb:

```
COPY users FROM 's3://awssampledbuswest2/tickit/allusers_pipe.txt'
CREDENTIALS 'aws_iam_role=YOUR-ROLE'
DELIMITER '|';
```

Before running this command, you will need to insert the ROLE that Redshift will use to access Amazon S3.

- To the left of the instructions you are currently reading, copy the value for **Role**. It will start with: *arn:aws:iam::*
- Paste the Role into your Pgweb query, **replacing the text *YOUR-ROLE***.

This 2nd line should now look like: *CREDENTIALS 'aws_iam_role=arn:aws:iam...'*

- Click the **Run Query** button to execute the SQL command.

The command will take approximately 10 seconds to load **49,990 rows of data**.

- Click the **users** table and then click the **Rows** tab.

You should now see the data that was loaded into Redshift.

Task 5: Query Data

Now that you have data in your Redshift database you can query the data using SQL select statements and queries. If you are familiar with SQL, feel free to try additional commands to query the data.

- Return to the **SQL Query** tab.
- Run this query to count the number of rows in the *users* table:

```
SELECT COUNT(*) FROM users;
```

The result shows that there are almost 50,000 rows in the table.

- Run this query:

```
SELECT userid, firstname, lastname, city, state
FROM users
WHERE likesports AND NOT likeopera AND state = 'OH'
ORDER BY firstname;
```

This query displays users in Ohio (OH) who like sports but do not like opera. The list is sorted by their first name.

- Run this query:

```
SELECT
  city,
  COUNT(*) AS count
FROM users
WHERE likejazz
GROUP BY city
ORDER BY count DESC
LIMIT 10;
```

This query shows the Top 10 cities where Jazz-loving users live.