Amazon Redshift

Getting Started Guide

aws

Amazon Redshift: Getting Started Guide

Table of Contents

Getting Started with Amazon Redshift

Welcome to the *Amazon Redshift Getting Started Guide*. Amazon Redshift is a fully managed, petabyte-scale data warehouse service in the cloud. An Amazon Redshift data warehouse is a collection of computing resources called *nodes*, which are organized into a group called a *cluster*. Each cluster runs an Amazon Redshift engine and contains one or more databases.

If you are a first-time user of Amazon Redshift, we recommend that you begin by reading the following sections:

- Amazon Redshift Management Overview – This topic provides an overview of Amazon Redshift.
- Service Highlights and Pricing – This product detail page provides the Amazon Redshift value proposition, service highlights, and pricing.
- Amazon Redshift Getting Started (*this guide*) – This guide provides a tutorial of using Amazon Redshift to create a sample cluster and work with sample data.

If you are building a proof-of-concept solution with Amazon Redshift, we recommend that you read Building a Proof of Concept for Amazon Redshift.

This guide is a tutorial designed to walk you through the process of creating a sample Amazon Redshift cluster. You can use this sample cluster to evaluate the Amazon Redshift service. In this tutorial, you perform the following steps:

After you complete this tutorial, you can find more information about Amazon Redshift and next steps in Where Do I Go From Here? (p. 18)

> **Important**
> The sample cluster that you create runs in a live environment. The on-demand rate is $0.25 per hour for using the sample cluster that is designed in this tutorial until you delete it. For more pricing information, go to the Amazon Redshift pricing page. If you have questions or get stuck, you can reach out to the Amazon Redshift team by posting on our Discussion Forum.

This tutorial is not meant for production environments, and does not discuss options in depth. After you complete the steps in this tutorial, you can use the Additional Resources (p. 18) section to locate more in-depth information. This information can help you plan, deploy, and maintain your clusters, and work with the data in your data warehouse.

Step 1: Set Up Prerequisites

Before you begin setting up an Amazon Redshift cluster, make sure that you complete the following prerequisites in this section:

Sign Up for AWS

If you don't already have an AWS account, you must sign up for one. If you already have an account, you can skip this prerequisite and use your existing account.

1. Open https://portal.aws.amazon.com/billing/signup.
2. Follow the online instructions.

   Part of the sign-up procedure involves receiving a phone call and entering a verification code on the phone keypad.

Determine Firewall Rules

As part of this tutorial, you specify a port when you launch your Amazon Redshift cluster. You also create an inbound ingress rule in a security group to allow access through the port to your cluster.

If your client computer is behind a firewall, you need to know an open port that you can use. This open port enables you to connect to the cluster from a SQL client tool and run queries. If you do not know this, you should work with someone who understands your network firewall rules to determine an open port in your firewall. Though Amazon Redshift uses port 5439 by default, the connection doesn't work if that port is not open in your firewall. You can't change the port number for your Amazon Redshift cluster after it is created. Thus, make sure that you specify an open port that works in your environment during the launch process.

Step 2: Create an IAM Role

For any operation that accesses data on another AWS resource, your cluster needs permission to access the resource and the data on the resource on your behalf. An example is using a COPY command to load data from Amazon S3. You provide those permissions by using AWS Identity and Access Management (IAM). You do so either through an IAM role that is attached to your cluster or by providing the AWS access key for an IAM user that has the necessary permissions.

To best protect your sensitive data and safeguard your AWS access credentials, we recommend creating an IAM role and attaching it to your cluster. For more information about providing access permissions, see Permissions to Access Other AWS Resources.

In this step, you create a new IAM role that enables Amazon Redshift to load data from Amazon S3 buckets. In the next step, you attach the role to your cluster.

To Create an IAM Role for Amazon Redshift

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Roles**.
3. Choose **Create role**.
4. In the **AWS Service** group, choose **Redshift.**
5. Under **Select your use case**, choose **Redshift - Customizable** then choose **Next: Permissions**.
6. On the **Attach permissions policies** page, choose **AmazonS3ReadOnlyAccess**. You can leave the default setting for **Set permissions boundary**. Then choose **Next: Tags**.

7. The **Add tags** page appears. You can optionally add tags. Choose **Next: Review**.

8. For **Role name**, enter a name for your role. For this tutorial, enter `myRedshiftRole`.

9. Review the information, and then choose **Create Role**.

10. Choose the role name of the role you just created.

11. Copy the **Role ARN** to your clipboard—this value is the Amazon Resource Name (ARN) for the role
    that you just created. You use that value when you use the COPY command to load data in Step 6:
    Load Sample Data from Amazon S3 (p. 15).

Now that you have created the new role, your next step is to attach it to your cluster. You can attach the
role when you launch a new cluster or you can attach it to an existing cluster. In the next step, you attach
the role to a new cluster.

Step 3: Create a Sample Amazon Redshift Cluster

Now that you have the prerequisites completed, you can launch your Amazon Redshift cluster.

Important
*The cluster that you are about to create is live (and not running in a sandbox). You incur the
standard Amazon Redshift usage fees for the cluster until you delete it.* If you complete the
tutorial described here in one sitting and delete the cluster when you are finished, the total
charges are minimal.

Note
A new console is available for Amazon Redshift. Choose either the **New Console** or the **Original
Console** instructions based on the console that you are using. The **New Console** instructions are
open by default.

New Console

To create an Amazon Redshift cluster

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://
    console.aws.amazon.com/redshift/.

    Important
    If you use IAM user credentials, ensure that you have the necessary permissions to perform
    the cluster operations. For more information, see Controlling Access to IAM Users in the
    Amazon Redshift Cluster Management Guide.

2. At top right, choose the AWS Region in which you want to create the cluster.

3. On the navigation menu, choose **CLUSTERS**, then choose **Create cluster**. The **Create cluster** page
    appears.

4. Choose **dc2.large** for the node type in the **Compute optimized** section. For **Nodes**, choose **2**.

    Note
    When you are creating a cluster for your data, you can use the Amazon Redshift console to
    help configure your cluster. To do this, choose **Calculate the best configuration for your
    needs**. When you do, a section of the page is replaced with controls for you to enter the size
    of your compressed data. Confirm the defaults and then choose **Select this configuration**.
    Your cluster configuration is displayed.

5. In the **Cluster details** section, specify values for **Cluster identifier**, **Database port**, **Master user
    name**, and **Master user password**. This tutorial uses these values:

    - **Cluster identifier**: Enter `examplecluster`.
    - **Database port**: Enter `5439`.
    - **Master user name**: Enter `awsuser`.

- **Master user password**: Enter a value for the password.

6. In the **Cluster permissions** section, for **Available IAM roles** choose the IAM role that you previously created, `myRedshiftRole`. Then choose **Add IAM role**.

7. Choose **Create cluster**.

Original Console

To launch an Amazon Redshift cluster

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift/.

   Important
   If you use IAM user credentials, ensure that the user has the necessary permissions to perform the cluster operations. For more information, go to Controlling Access to IAM Users in the *Amazon Redshift Cluster Management Guide*.

2. In the main menu, select the AWS Region in which you want to create the cluster. For the purposes of this tutorial, select **US West (Oregon)**.

3. On the Amazon Redshift Dashboard, choose **Quick launch cluster**.

   The Amazon Redshift Dashboard looks similar to the following.
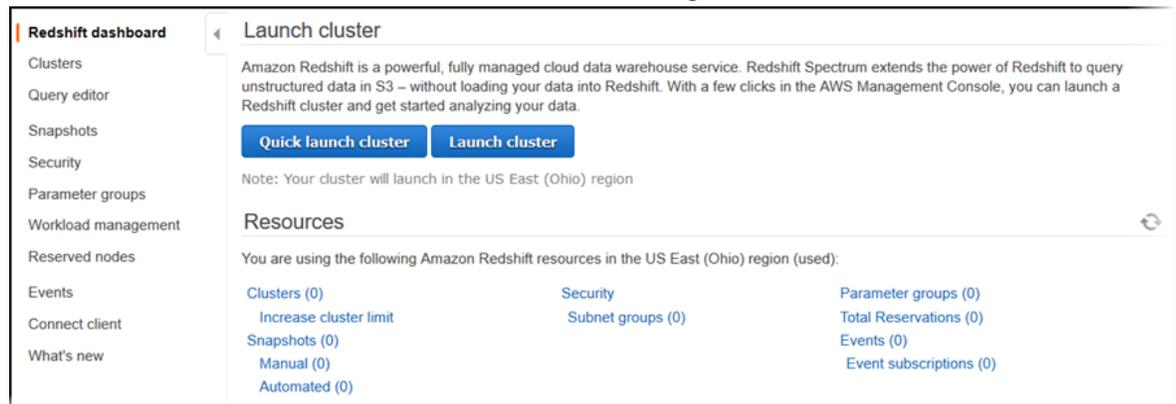
4. On the Cluster specifications page, enter the following values and then choose **Launch cluster**:

   - **Node type**: Choose **dc2.large**.

   - **Number of compute nodes**: Keep the default value of **2**.

   - **Cluster identifier**: Enter the value **examplecluster**.

   - **Master user name**: Keep the default value of **awsuser**.

   - **Master user password** and **Confirm password**: Enter a password for the master user account.

   - **Database port**: Accept the default value of **5439**.

   - **Available IAM roles**: Choose **myRedshiftRole**.

Quick launch automatically creates a default database named **dev**.

Launch your Amazon Redshift cluster - Quick launch | Switch to advanced settings

Amazon Redshift Pricing offers on-demand and reserved instance pricing options. **Save up to 75%** through reserved instances.

Node type* | dc2.large | Storage type: SSD | Storage: 0.16 TB/node | Compute optimized | 0.25 USD/node

Nodes* 2 x 0.16 TB/node = 0.32 TB storage available

Cluster identifier* examplecluster

Database name dev | Database port* 5439

Master user name* awsuser

Master user password* •••••••• | Confirm password* ••••••••

Cluster permissions - *optional*
Your cluster needs permission to access other AWS services on your behalf. For the required permissions, add an IAM role now or after you launch the cluster.
Learn more

Available IAM roles | Choose a role(s)

myRedshiftRole ✕
arn:aws:iam::799504075249:role/myRedshiftRole

▼ Default settings | Switch to advanced settings
We'll apply some default settings for network, security, backup, and maintenance to get you started. Switch to advance settings if you want to change the defaults.

Network | Using **default VPC (vpc-**████████**)** and **default** subnet. Learn more
Security | A **default** security group will be created when this cluster is launched
Configuration | Using **default** parameter group with **no** encryption
Backup | Using **current** maintenance track
Maintenance | Automated snapshots every **8 hours** retained for **2 days**

Cancel | **Launch cluster**

Note
Quick launch uses the default virtual private cloud (VPC) for your AWS Region. If a default VPC doesn't exist, Quick launch returns an error. If you don't have a default VPC, you can use the standard Launch Cluster wizard to use a different VPC. For more information, see Creating a Cluster by Using Launch Cluster.

5. A confirmation page appears and the cluster takes a few minutes to finish. Choose **Close** to return to the list of clusters.

☑ Cluster **examplecluster** is being created. Your cluster may take a few minutes to launch.

You will start accruing charges as soon as your cluster is active.
Applicable charges
The on-demand hourly rate for this cluster will be $0.25 , or $0.25 /node. If you have purchased reserved nodes in this region for this node type that are active, your costs will be discounted. Additional nodes will be billed at the on-demand rate.
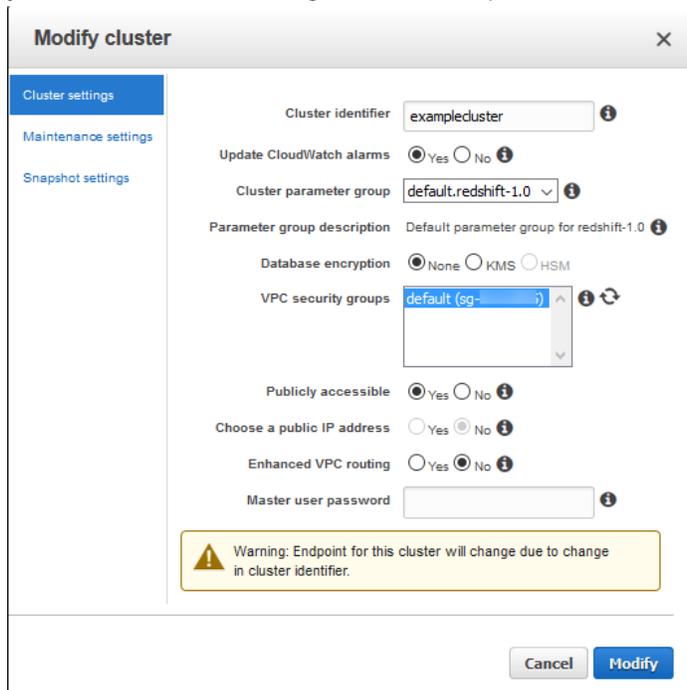
For more information, see Amazon Redshift Pricing and Reserved Nodes Documentation

6. On the Clusters page, choose the cluster that you just launched and review the **Cluster Status** information. Make sure that the **Cluster Status** is **available** and the **Database Health** is **healthy** before you try to connect to the database later in this tutorial.

			Cluster	Cluster Status	DB Health		Release Status		In Maintenance		Recent Events
☐	▶	🔍	examplecluster	available	healthy		Up to date		no		1
☐	▶	🔍	redshift-cluster-1	available	healthy		Up to date		no		10

7. On the Clusters page, choose the cluster that you just launched, choose the **Cluster** button, then **Modify cluster**. Choose the **VPC security groups** to associate with this cluster, then choose **Modify** to make the association. Make sure that the **Cluster Properties** displays the **VPC security groups** you chose before continuing to the next step.

Step 4: Authorize Access to the Cluster

Note
A new console is available for Amazon Redshift. Choose either the **New Console** or the **Original Console** instructions based on the console that you are using. The **New Console** instructions are open by default.

New Console

Later in this tutorial, you access your cluster from within a VPC. However, if you use an SQL client from outside your firewall to access the cluster, you must grant inbound access.

You can skip this step if you plan to access the cluster with the Amazon Redshift query editor from within your VPC.

**To check your firewall and grant inbound access to your cluster**

1. Check your firewall rules if your cluster needs to be accessed from outside a firewall. For example, your client might be an Amazon EC2 instance or an external computer.

2. To access from an Amazon EC2 external client, add an ingress rule to the security group attached to your cluster that allows inbound traffic. You add Amazon EC2 security group rules in the Amazon EC2 console. For example, a CIDR/IP of 192.0.2.0/24 allows clients in that IP address range to connect to your cluster. You need to find out the correct CIDR/IP for your environment.

Original Console

In the previous step, you launched your Amazon Redshift cluster. Before you can connect to the cluster, you need to configure a security group to authorize access.

To configure the VPC security group (EC2-VPC platform)

1. In the Amazon Redshift console, in the navigation pane, choose **Clusters**.

2. Choose `examplecluster` to open it, and make sure that you are on the **Configuration** tab.

3. Under **Cluster Properties**, for **VPC Security Groups**, choose your security group.
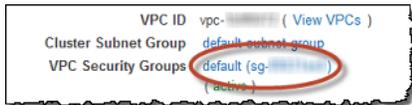

4. After your security group opens in the Amazon EC2 console, choose the **Inbound** tab.
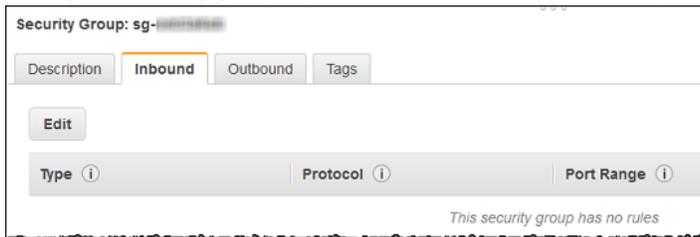
5. Choose **Edit**, **Add Rule**, and enter the following, then choose **Save**:

- **Type**: **Custom TCP Rule**.
- **Protocol**: **TCP**.
- **Port Range**: Enter the same port number that you used when you launched the cluster. The default port for Amazon Redshift is `5439`, but your port might be different.
- **Source**: Select **Custom**, then enter `0.0.0.0/0`.

    Important
    Using 0.0.0.0/0 is not recommended for anything other than demonstration purposes because it allows access from any computer on the internet. In a real environment, you create inbound rules based on your own network settings.
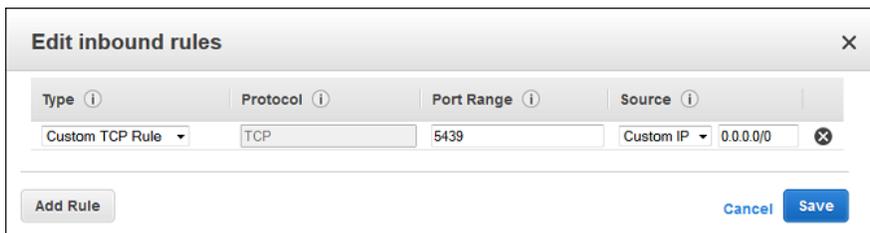
Step 5: Connect to the Sample Cluster and Run Queries

To query databases hosted by your Amazon Redshift cluster, you have two options:

- Connect to your cluster and run queries on the AWS Management Console with the query editor.

  If you use the query editor, you don't have to download and set up an SQL client application.
- Connect to your cluster through an SQL client tool, such as SQL Workbench/J.

Topics

Querying a Database Using the Query Editor

Using the query editor is the easiest way to run queries on databases hosted by your Amazon Redshift cluster. After creating your cluster, you can immediately run queries using the console.

The following cluster node types support the query editor:

- DC1.8xlarge
- DC2.large
- DC2.8xlarge
- DS2.8xlarge

Using the Amazon Redshift console query editor, you can do the following:

- Run single SQL statement queries.
- Download result sets as large as 100 MB to a comma-separated value (CSV) file.
- Save queries for reuse. You can't save queries in the EU (Paris) Region or the Asia Pacific (Osaka-Local) Region.
- View query execution details for user-defined tables.

Query Editor Considerations

For details about considerations when using the query editor, see Querying a Database Using the Query Editor in the Amazon Redshift Cluster Management Guide.

Enabling Access to the Query Editor

To access the query editor, you need permission. To enable access, attach the `AmazonRedshiftQueryEditor` and `AmazonRedshiftReadOnlyAccess` policies for AWS Identity and Access Management (IAM) to the AWS IAM user that you use to access your cluster.

If you have already created an IAM user to access Amazon Redshift, you can attach the `AmazonRedshiftQueryEditor` and `AmazonRedshiftReadOnlyAccess` policies to that user. If you haven't created an IAM user yet, create one and attach the policies to the IAM user.

To attach the required IAM policies for the Query Editor

1. Sign in to the AWS Management Console and open the IAM console at https://
   console.aws.amazon.com/iam/.
2. Choose **Users**.
3. Choose the user that needs access to the Query Editor.
4. Choose **Add permissions**.
5. Choose **Attach existing policies directly**.
6. For **Policy names**, choose **AmazonRedshiftQueryEditor** and **AmazonRedshiftReadOnlyAccess**.
7. Choose **Next: Review**.
8. Choose **Add permissions**.

Using the Query Editor

In the following example, you use the query editor to perform the following tasks:

- Run SQL commands.
- View query execution details.
- Save a query.
- Download a query result set.

> **Note**
> A new console is available for Amazon Redshift. Choose either the **New Console** or the **Original
> Console** instructions based on the console that you are using. The **New Console** instructions are
> open by default.

New Console

To use the query editor

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://
   console.aws.amazon.com/redshift/.
2. On the navigation menu, choose **EDITOR**, then connect to a database in your cluster.
3. For **Schema**, choose **public** to create a new table based on that schema.
4. Enter the following in the query editor window, and choose **Run query** to create a new table.

```
create table shoes(
            shoetype varchar (10),
            color varchar(10));
```

5. Choose **Clear**.
6. Enter the following command in the query editor window, and choose **Run** to add rows to the table.

```
insert into shoes values
('loafers', 'brown'),
('sandals', 'black');
```

7. Choose **Clear**.
8. Enter the following command in the query editor window, and choose **Run** to query the new table.

```
select * from shoes;
```

The **Query results** displays the results.

shoetype	color
sandals	black
loafers	brown

9. Choose **Execution** to view the run details.
10. Choose **Export** to download the query results as a file.

Original Console

To use the query editor

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift/.
2. In the navigation pane, choose **Query Editor**.
3. In the **Credentials** dialog box, enter the following values and then choose **Connect**:

   - **Cluster**: Choose **examplecluster**.
   - **Database**: **dev**.
   - **Database user**: **awsuser**.
   - **Password**: Enter the password that you specified when you launched the cluster.
4. For **Schema**, choose **public** to create a new table based on that schema.
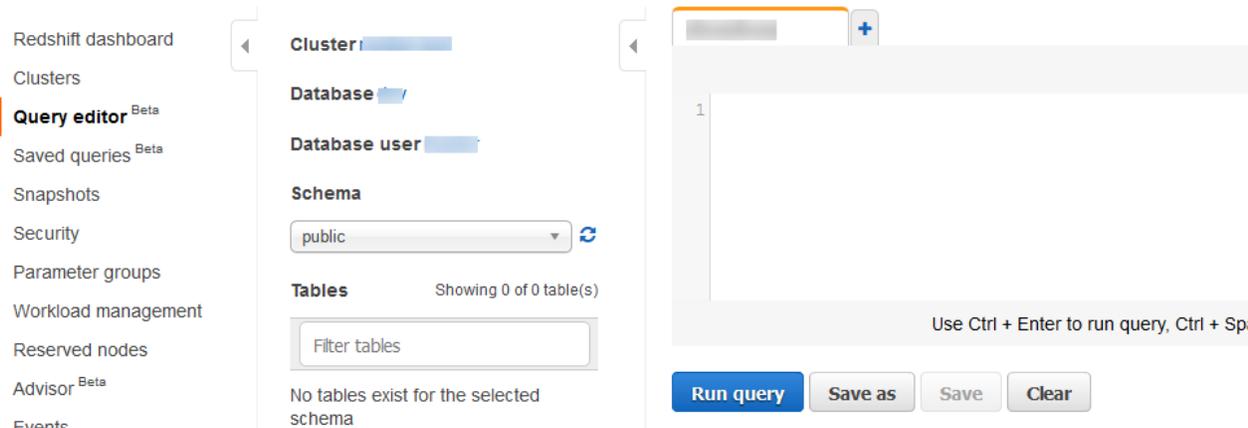
5. Enter the following in the Query Editor window and choose **Run query** to create a new table.

```
create table shoes(
shoetype varchar (10),
color varchar(10));
```
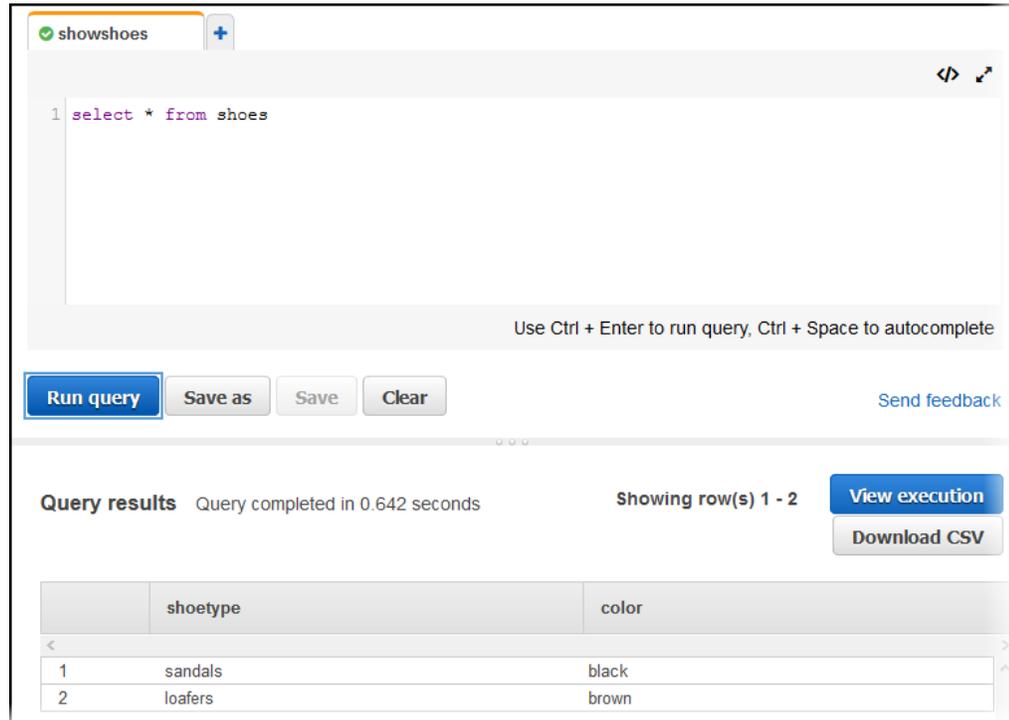
6. Choose **Clear**.
7. Enter the following command in the Query Editor window and choose **Run query** to add rows to the table.

```
insert into shoes values
('loafers', 'brown'),
('sandals', 'black');
```

8. Choose **Clear**.

9. Enter the following command in the Query Editor window and choose **Run query** to query the new table.

```
select * from shoes;
```

You should see the following results.

Querying a Database Using a SQL Client

Next, you connect to your cluster by using a SQL client tool and run a simple query to test the connection. You can use most SQL client tools that are compatible with PostgreSQL. For this tutorial, you use the SQL Workbench/J client. Complete this section by performing the following steps:

After you complete this step, you can determine whether you want to load sample data from Amazon S3 in or find more information about Amazon Redshift and reset your environment at .

Install SQL Client Drivers and Tools

You can use most SQL client tools with Amazon Redshift JDBC or ODBC drivers to connect to an Amazon Redshift cluster. In this tutorial, you connect using SQL Workbench/J, a free, DBMS-independent, cross-platform SQL query tool. If you plan to use SQL Workbench/J to complete this tutorial, use the steps following to set up the Amazon Redshift JDBC driver and SQL Workbench/J. For more complete

instructions for installing SQL Workbench/J, go to Setting Up the SQL Workbench/J Client in the *Amazon Redshift Cluster Management Guide.* If you use an Amazon EC2 instance as your client computer, install SQL Workbench/J and the required drivers on the instance.

> **Note**
> Install any third-party database tools that you want to use with your clusters yourself. Amazon Redshift doesn't provide or install any third-party tools or libraries.

To Install SQL Workbench/J on Your Client Computer

1. Review the SQL Workbench/J software license.

2. Go to the SQL Workbench/J website and download the appropriate package for your operating system.

3. Go to the Installing and starting SQL Workbench/J page and install SQL Workbench/J.

   > **Important**
   > Note the Java runtime version prerequisites for SQL Workbench/J and ensure you are using that version. Otherwise, the client application doesn't run.

4. Go to Configure a JDBC Connection and download an Amazon Redshift JDBC driver to enable SQL Workbench/J to connect to your cluster.

For more information about using the Amazon Redshift JDBC or ODBC drivers, see Configuring Connections in Amazon Redshift.

To Get Your Connection String

To connect to your cluster with your SQL client tool, you need the cluster connection string. You can find the cluster connection string in the Amazon Redshift console, on a cluster's details page.

> **Note**
> A new console is available for Amazon Redshift. Choose either the **New Console** or the **Original Console** instructions based on the console that you are using. The **New Console** instructions are open by default.

New Console

To find the connection string for a cluster

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift/.

2. On the navigation menu, choose **CLUSTERS**, then choose the cluster name from the list to open its details.

3. Choose the **Properties** tab for the cluster, and view the **Connection details** to see the values for **JDBC URL** and **ODBC URL**. The connection string is based on the AWS Region where the cluster runs.

4. Choose **Copy** to copy the string on this page.

Original Console

To find the connection string for a cluster

1. In the Amazon Redshift console, in the navigation pane, choose **Clusters**.

2. Choose `examplecluster` to open it, and make sure that you are on the **Configuration** tab.

3. On the **Configuration** tab, under **Cluster Database Properties**, copy the JDBC URL of the cluster.

Note
The endpoint for your cluster is not available until the cluster is created and in the available state.
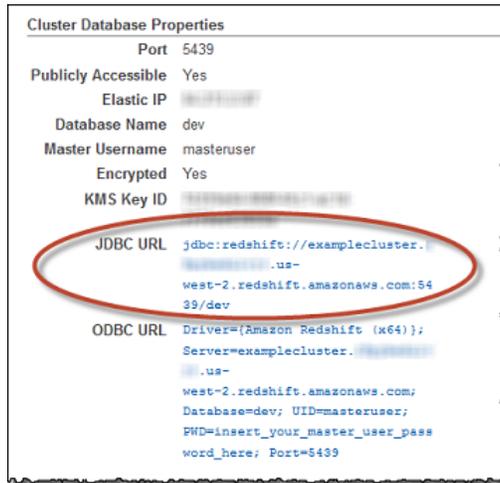
To Connect from SQL Workbench/J to Your Cluster

This step assumes you installed SQL Workbench/J.

1. Open SQL Workbench/J.
2. Choose **File**, and then choose **Connect window**.
3. Choose **Create a new connection profile**.
4. For **New profile**, enter a name for the profile.
5. Choose **Manage Drivers**. The **Manage Drivers** dialog box opens.
6. Choose **Create a new entry**. For **Name**, enter a name for the driver.
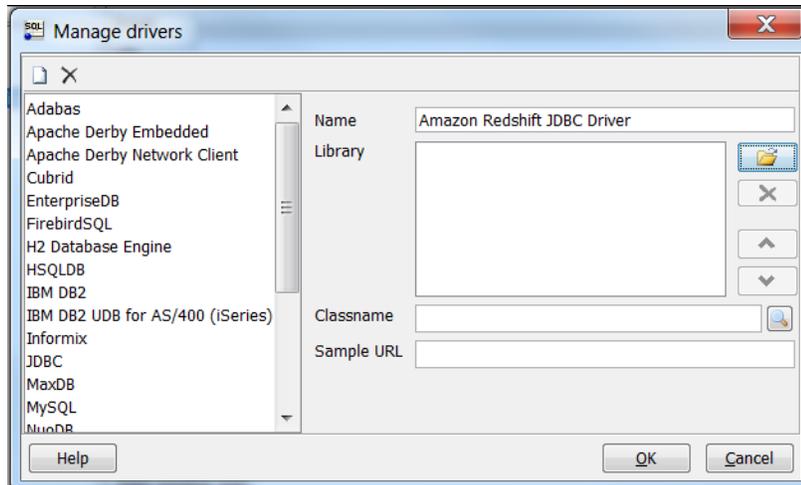
Choose the folder icon next to the **Library** box, navigate to the location of the driver, choose it, and then choose **Open**.
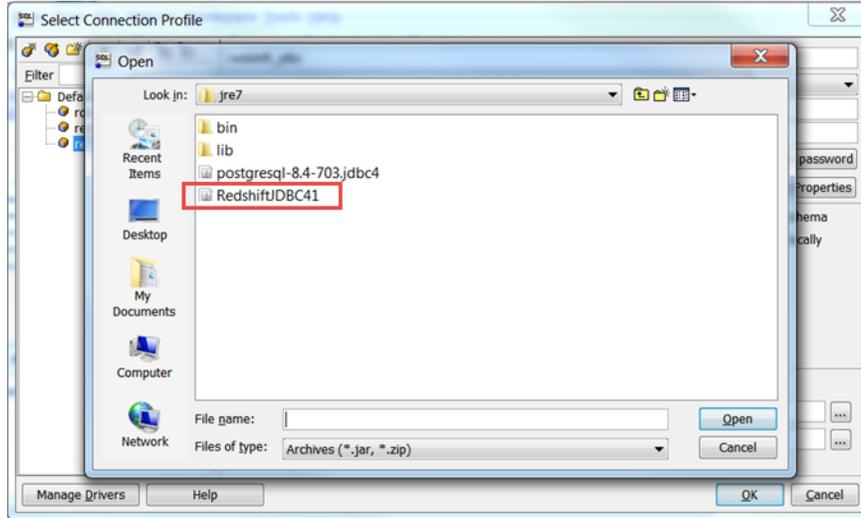
If the **Please select one driver** dialog box displays, choose **com.amazon.redshift.jdbc4.Driver** or **com.amazon.redshift.jdbc41.Driver** and then choose **OK**. SQL Workbench/J automatically completes the **Classname** box. Keep **Sample URL** blank, and choose **OK**.

7. For **Driver**, choose the driver that you just added.

8. For **URL**, copy the JDBC URL from the Amazon Redshift console and paste it here.

9. For **Username**, enter `awsuser` for the master user.

10. For **Password**, enter the password associated with the master user account.

11. Choose **Autocommit**.

12. Choose the **Save profile list** icon, as shown following.
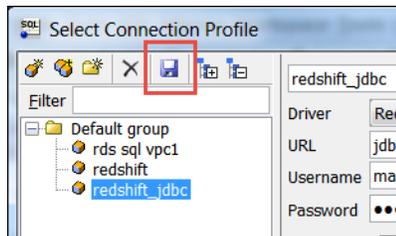
13. Choose **OK**.
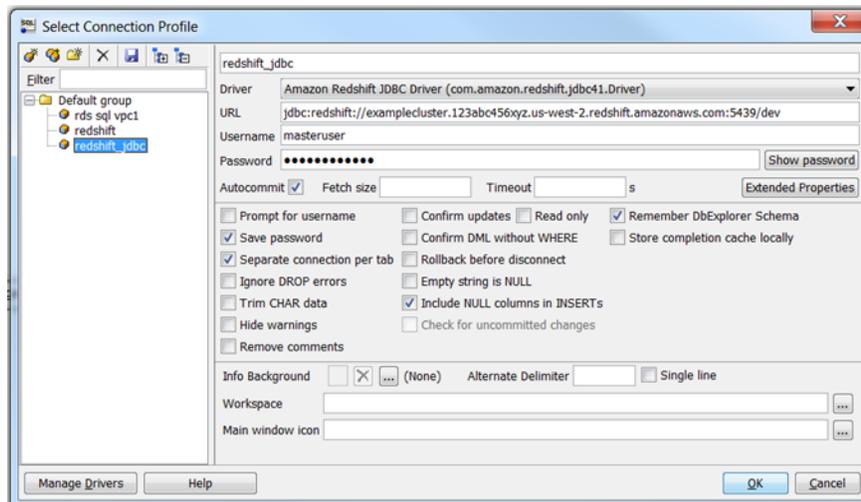
14. Enter the following command in the query window and choose **SQL**, **Execute Current** to add rows to the table.

```
create table shoes(
shoetype varchar (10),
color varchar(10));
```

15. Run the following command to add rows to the table.

```
insert into shoes values
('loafers', 'brown'),
('sandals', 'black');
```

16. Run the following command to query the new table.

```
select * from shoes;
```

Step 6: Load Sample Data from Amazon S3

At this point, you have a database called `dev` and you are connected to it. Next, you create some tables in the database, upload data to the tables, and try a query. For your convenience, the sample data you load is available in an Amazon S3 bucket.

> **Note**
> If you're using a SQL client tool, ensure that your SQL client is connected to the cluster.

After you complete this step, you can find more information about Amazon Redshift and reset your environment at Where Do I Go From Here? (p. 18).

To load sample data

1. Create tables.

   Individually copy and run the following create table statements to create tables in the `dev` database. For more information about the syntax, see CREATE TABLE in the *Amazon Redshift Database Developer Guide*.

```
create table 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 table venue(
```

```
 venueid smallint not null distkey sortkey,
 venuename varchar(100),
 venuecity varchar(30),
 venuestate char(2),
 venueseats integer);

create table category(
 catid smallint not null distkey sortkey,
 catgroup varchar(10),
 catname varchar(10),
 catdesc varchar(50));

create table date(
 dateid smallint not null distkey sortkey,
 caldate date not null,
 day character(3) not null,
 week smallint not null,
 month character(5) not null,
 qtr character(5) not null,
 year smallint not null,
 holiday boolean default('N'));

create table 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 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 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);
```

2. Load sample data from Amazon S3 by using the COPY command.

    Note
    We recommend using the COPY command to load large datasets into Amazon Redshift
    from Amazon S3 or DynamoDB. For more information about COPY syntax, see COPY in the
    Amazon Redshift Database Developer Guide.

    Download file tickitdb.zip that contains individual sample data files. Unzip and load the individual
    files to a `tickit` folder in your Amazon S3 bucket in your AWS Region. Edit the COPY commands in
    this tutorial to point to the files in your Amazon S3 bucket. For information about how to manage
    files with Amazon S3, see Creating and Configuring an S3 Bucket in the *Amazon Simple Storage
    Service Console User Guide*.

To load the sample data, you must provide authentication for your cluster to access Amazon S3 on your behalf. You can provide either role-based authentication or key-based authentication. We recommend using role-based authentication. For more information about both types of authentication, see CREDENTIALS in the Amazon Redshift Database Developer Guide.

For this step, you provide authentication by referencing the IAM role that you created and then attached to your cluster in previous steps.

> **Note**
> If you don't have proper permissions to access Amazon S3, you receive the following error message when running the COPY command: `S3ServiceException: Access Denied`.

The COPY commands include a placeholder for the Amazon Resource Name (ARN) for the IAM role, your bucket name, and an AWS Region, as shown in the following example.

```
copy users from 's3://<myBucket>/tickit/allusers_pipe.txt'
credentials 'aws_iam_role=<iam-role-arn>'
delimiter '|' region '<aws-region>';
```

To authorize access using an IAM role, replace `<iam-role-arn>` in the CREDENTIALS parameter string with the role ARN for the IAM role that you created in Step 2: Create an IAM Role (p. 2).

Your COPY command looks similar to the following example.

```
copy users from 's3://<myBucket>/tickit/allusers_pipe.txt'
credentials 'aws_iam_role=arn:aws:iam::123456789012:role/myRedshiftRole'
delimiter '|' region '<aws-region>';
```

To load the sample data, replace `<myBucket>`, `<iam-role-arn>`, and `<aws-region>` in the following COPY commands with your values. Then run the commands individually in your SQL client tool.

```
copy users from 's3://<myBucket>/tickit/allusers_pipe.txt'
credentials 'aws_iam_role=<iam-role-arn>'
delimiter '|' region '<aws-region>';

copy venue from 's3://<myBucket>/tickit/venue_pipe.txt'
credentials 'aws_iam_role=<iam-role-arn>'
delimiter '|' region '<aws-region>';

copy category from 's3://<myBucket>/tickit/category_pipe.txt'
credentials 'aws_iam_role=<iam-role-arn>'
delimiter '|' region '<aws-region>';

copy date from 's3://<myBucket>/tickit/date2008_pipe.txt'
credentials 'aws_iam_role=<iam-role-arn>'
delimiter '|' region '<aws-region>';

copy event from 's3://<myBucket>/tickit/allevents_pipe.txt'
credentials 'aws_iam_role=<iam-role-arn>'
delimiter '|' timeformat 'YYYY-MM-DD HH:MI:SS' region '<aws-region>';

copy listing from 's3://<myBucket>/tickit/listings_pipe.txt'
credentials 'aws_iam_role=<iam-role-arn>'
delimiter '|' region '<aws-region>';

copy sales from 's3://<myBucket>/tickit/sales_tab.txt'
credentials 'aws_iam_role=<iam-role-arn>'
delimiter '\t' timeformat 'MM/DD/YYYY HH:MI:SS' region '<aws-region>';
```

3. Now try the example queries. For more information, see SELECT in the *Amazon Redshift Developer Guide*.

```
-- Get definition for the sales table.
SELECT *
FROM pg_table_def
WHERE tablename = 'sales';

-- Find total sales on a given calendar date.
SELECT sum(qtysold)
FROM    sales, date
WHERE   sales.dateid = date.dateid
AND     caldate = '2008-01-05';

-- Find top 10 buyers by quantity.
SELECT firstname, lastname, total_quantity
FROM    (SELECT buyerid, sum(qtysold) total_quantity
         FROM  sales
         GROUP BY buyerid
         ORDER BY total_quantity desc limit 10) Q, users
WHERE Q.buyerid = userid
ORDER BY Q.total_quantity desc;

-- Find events in the 99.9 percentile in terms of all time gross sales.
SELECT eventname, total_price
FROM  (SELECT eventid, total_price, ntile(1000) over(order by total_price desc) as
 percentile
       FROM (SELECT eventid, sum(pricepaid) total_price
             FROM   sales
             GROUP BY eventid)) Q, event E
       WHERE Q.eventid = E.eventid
       AND percentile = 1
ORDER BY total_price desc;
```

Step 7: Find Additional Resources and Reset Your Environment

When you have completed this tutorial, you can go to other Amazon Redshift resources to learn more about the concepts introduced in this guide. You can also reset your environment to the previous state. You might want to keep the sample cluster running if you intend to try tasks in other Amazon Redshift guides. However, remember that *you continue to be charged for your cluster as long as it is running*. To stop incurring charges, revoke access to the cluster and delete it when you no longer need it.

Where Do I Go From Here?

Additional Resources

We recommend that you continue to learn more about the concepts introduced in this guide with the following resources:

- Amazon Redshift Management Overview: This topic provides an overview of Amazon Redshift.
- Amazon Redshift Cluster Management Guide: This guide builds upon this *Amazon Redshift Getting Started* and provides in-depth information about the concepts and tasks for creating, managing, and monitoring clusters.

- Amazon Redshift Database Developer Guide: This guide builds upon this *Amazon Redshift Getting Started* by providing in-depth information for database developers about designing, building, querying, and maintaining the databases that make up your data warehouse.

Resetting Your Environment

Note
A new console is available for Amazon Redshift. Choose either the **New Console** or the **Original Console** instructions based on the console that you are using. The **New Console** instructions are open by default.

New Console

When you have completed this tutorial, reset your environment to the previous state by deleting your sample cluster. *You continue to incur charges for the Amazon Redshift service until you delete the cluster*.

To delete a cluster

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift/.
2. On the navigation menu, choose **CLUSTERS** to display your list of clusters.
3. Choose the `examplecluster` cluster. For **Actions**, choose **Delete**. The **Delete cluster** page appears.
4. Confirm the cluster to be deleted, then choose **Delete cluster**.

On the cluster list page, the cluster status is updated as the cluster is deleted.

Original Console

When you have completed this tutorial, you should reset your environment to the previous state by doing the following:

- Revoke access to the port and CIDR/IP address for which you authorized access:

  If you used the EC2-VPC platform to launch your cluster, perform the steps in To Revoke Access from the VPC Security Group (p. 19).

- Delete your sample cluster. *You continue to incur charges for the Amazon Redshift service until you delete the cluster*. Perform the steps in To Delete the Sample Cluster (p. 20).

To Revoke Access from the VPC Security Group

1. In the Amazon Redshift console, in the navigation pane, choose **Clusters**.
2. Choose **examplecluster** to open it, and make sure that you are on the **Configuration** tab.
3. Under **Cluster Properties**, choose the VPC security group.
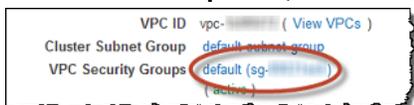
4. With the default security group selected, choose the **Inbound** tab and then choose **Edit**.
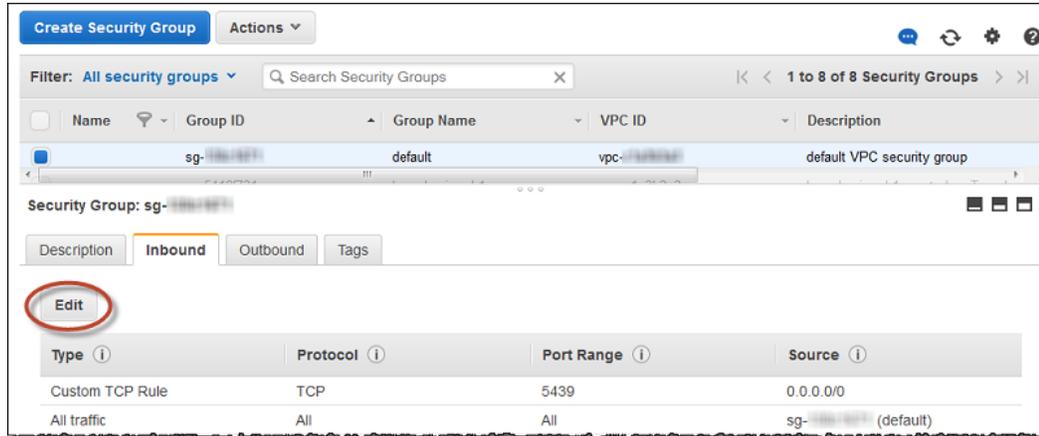
5. Delete the custom TCP/IP ingress rule that you created for your port and CIDR/IP address 0.0.0.0/0. Do not remove any other rules, such as the **All traffic** rule that was created for the security group by default. Choose **Save**.
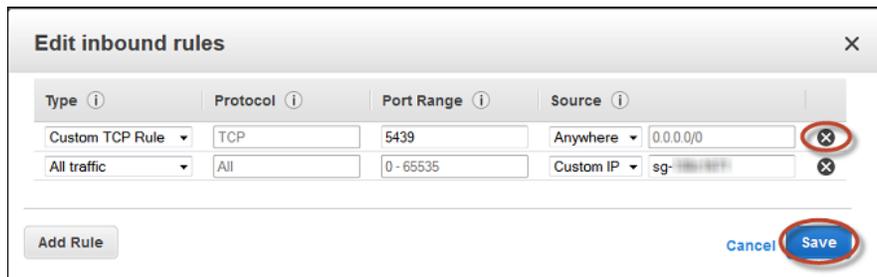

To Delete the Sample Cluster

1. In the Amazon Redshift console, in the navigation pane, choose **Clusters**.
2. Choose `examplecluster` to open it, and make sure that you are on the **Configuration** tab.
3. In the **Cluster** menu, choose **Delete**.

4. In the **Delete Cluster** window, for **Create snapshot**, choose **No** and then choose **Delete**.

Delete Cluster   ✕

If you want to restore your cluster later, you must create a manual snapshot. You can restore the cluster from the snapshot later and resume querying in minutes. You will be charged the Amazon S3 storage rate for a manual snapshot only if you don't have other clusters running, or if you exceed the allocated storage for active clusters.
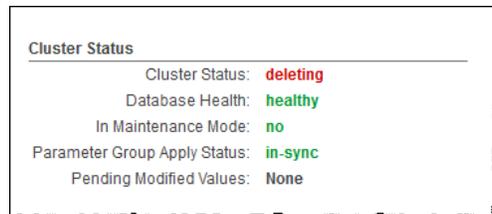
Cluster	examplecluster ℹ
Create snapshot	⦿ Yes ◯ No ℹ
Snapshot name	[ ] ℹ

Cancel   **Delete**

5. On the cluster details window, the **Cluster Status** displays that the cluster is being deleted.

Cluster Status

Cluster Status:	deleting
Database Health:	healthy
In Maintenance Mode:	no
Parameter Group Apply Status:	in-sync
Pending Modified Values:	None

Document History

The following table describes the important changes since the last release of the *Amazon Redshift Getting Started Guide*.

Latest documentation update: November 11, 2019

Change	Description	Release Date
New Console	Updated the guide to describe the new Amazon Redshift console.	November 11, 2019
New Feature	Updated the guide to describe the quick-launch cluster procedure.	August 10, 2018
New Feature	Updated the guide to launch clusters from the Amazon Redshift dashboard.	July 28, 2015
New Feature	Updated the guide to use new node type names.	June 9, 2015
Documentation Update	Updated screenshots and procedure for configuring VPC security groups.	April 30, 2015
Documentation Update	Updated screenshots and procedures to match the current console.	November 12, 2014
Documentation Update	Moved loading data from Amazon S3 information into its own section and moved next steps section into the final step for better discoverability.	May 13, 2014
Documentation Update	Removed the Welcome page and incorporated the content into the main Getting Started page.	March 14, 2014
Documentation Update	This is a new release of the *Amazon Redshift Getting Started Guide* that addresses customer feedback and service updates.	March 14, 2014
New Guide	This is the first release of the *Amazon Redshift Getting Started Guide*.	February 14, 2013