**Launching and Using Redshift Cluster**

1. Download SQL Workbench and Redshift JDBC Drivers.
2. Go to the [SQL Workbench/J website](http://www.sql-workbench.net/" \t "_blank) and download the appropriate package for your operating system.
3. Go to the [Installing and starting SQL Workbench/J page](http://www.sql-workbench.net/manual/install.html" \t "_blank) and install SQL Workbench/J.
4. Go to [Configure a JDBC Connection](http://docs.aws.amazon.com/redshift/latest/mgmt/configure-jdbc-connection.html" \t "_blank) and download an Amazon Redshift JDBC driver to enable SQL Workbench/J to connect to your cluster.

Create an IAM Role

1. Sign in to the AWS Management Console and open the IAM console at [https://console.aws.amazon.com/iam/](https://console.aws.amazon.com/iam/" \t "_blank).
2. In the left 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**, and then choose **Next: Review**.
7. For **Role name**, type a name for your role. For this tutorial, type myRedshiftRole.
8. Review the information, and then choose **Create Role**.
9. Choose the role name for new role.
10. Copy the **Role ARN** to your clipboard—this value is the Amazon Resource Name (ARN) for the role that you just created. You will use that value when you use the COPY command to load data.

# Launch a Sample Amazon Redshift Cluster

## To Launch an Amazon Redshift Cluster

1. Go to Redshift console.
2. In the main menu, select the region in which you want to create the cluster.
3. On the Amazon Redshift Dashboard, choose **Launch Cluster**.
4. On the Cluster Details page, enter the following values and then choose **Continue**:
   * **Cluster Identifier**: type examplecluster.
   * **Database Name**: leave this box blank. Amazon Redshift will create a default database named dev.
   * **Database Port**: type the port number on which the database will accept connections. You should have determined the port number in the prerequisite step of this tutorial. You cannot change the port after launching the cluster, so make sure that you have an open port number in your firewall so that you can connect from SQL client tools to the database in the cluster.
   * **Master User Name**: type masteruser. You will use this username and password to connect to your database after the cluster is available.
   * **Master User Password** and **Confirm Password**: type a password for the master user account.
5. On the Node Configuration page, select the following values and then choose **Continue**:
   * **Node Type**: **dc2.large**
   * **Cluster Type**: **Single Node**
6. On the Additional Configuration page, choose **EC2-VPC**

If you have a default VPC in the region you’ve selected, you will use the EC2-VPC platform to launch your cluster.

Use the following values if you are launching your cluster in the EC2-VPC platform:

* + **Cluster Parameter Group**: select the default parameter group.
  + **Encrypt Database**: **None**.
  + **Choose a VPC**: **Default VPC (vpc-xxxxxxxx)**
  + **Cluster Subnet Group**: **default**
  + **Publicly Accessible**: **Yes**
  + **Choose a Public IP Address**: **No**
  + **Enhanced VPC Routing**: **No**
  + **Availability Zone**: **No Preference**
  + **VPC Security Groups**: **default (sg-xxxxxxxx)**
  + **Create CloudWatch Alarm**: **No**

1. Associate an IAM role with the cluster.

For **AvailableRoles**, choose **myRedshiftRole** and then choose **Continue**.

1. On the Review page, review the selections that you’ve made and then choose **Launch Cluster**.
2. A confirmation page appears and the cluster will take a few minutes to finish. Choose **Close** to return to the list of clusters.
3. 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 lab.

## 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 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**, and enter the following, then choose **Save**:
   * **Type**: **Custom TCP Rule**.
   * **Protocol**: **TCP**.
   * **Port Range**: type 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 IP**, then type 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 would create inbound rules based on your own network settings.

## To Configure the Amazon Redshift Security Group

1. In the Amazon Redshift console, in the navigation pane, choose **Clusters**.
2. Choose examplecluster to open it, and make sure you are on the **Configuration** tab.
3. Under **Cluster Properties**, for **Cluster Security Groups**, choose **default** to open the default security group.
4. On the **Security Groups** tab, in the cluster security group list, choose the cluster security group whose rules you want to manage.
5. On the **Security Group Connections** tab, choose **Add Connection Type**.
6. In the **Connection Type** box, choose **CIDR/IP**.

In **CIDR/IP to Authorize**, type 0.0.0.0/0 and choose **Authorize**.

# Connect to the Sample Cluster

## To Get Your Connection String

1. In the Amazon Redshift console, in the navigation pane, choose **Clusters**.
2. Choose examplecluster to open it, and make sure 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. In the **New profile** text box, type a name for the profile.
5. Choose **Manage Drivers**. The **Manage Drivers** dialog opens.
6. Choose the **Create a new entry** button. In the **Name** text box, type a name for the driver.

Choose the folder icon next to the **Library** box, navigate to the location of the driver, select it, and then choose **Open**.

If the **Please select one driver** dialog box displays, select **com.amazon.redshift.jdbc4.Driver** or **com.amazon.redshift.jdbc41.Driver** and choose **OK**. SQL Workbench/J automatically completes the **Classname** box. Leave the **Sample URL** box blank, and then choose **OK**.

1. In the **Driver** box, choose the driver you just added.
2. In **URL**, copy the JDBC URL from the Amazon Redshift console and paste it here.
3. In **Username**, type *masteruser*.
4. In **Password**, type the password associated with the master user account.
5. Choose the **Autocommit** box.
6. Choose the **Save profile list** icon:
7. Choose **OK**.

# Load Sample Data from Amazon S3

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

**Note**

Before you proceed, ensure that your SQL Workbench/J client is connected to the cluster.

1. Create tables.

Copy and execute the following create table statements to create tables in the dev database.

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);

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

The sample data for this tutorial is provided in an Amazon S3 bucket that is owned by Amazon Redshift. The bucket permissions are configured to allow all authenticated AWS users read access to the sample data files.

To load the sample data, you must provide authentication for your cluster to access Amazon S3 on your behalf.

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

The COPY commands include a placeholder for the IAM role ARN, as shown in the following example.

copy users from 's3://awssampledbuswest2/tickit/allusers\_pipe.txt'

credentials 'aws\_iam\_role=<iam-role-arn>'

delimiter '|' region 'us-west-2';

To authorize access using an IAM role, replace *<iam-role-arn>* in the CREDENTIALS parameter string with the role ARN for the IAM role you created earlier.

Your COPY command will look similar to the following example.

copy users from 's3://awssampledbuswest2/tickit/allusers\_pipe.txt'

credentials 'aws\_iam\_role=arn:aws:iam::123456789012:role/myRedshiftRole'

delimiter '|' region 'us-west-2';

To load the sample data, replace *<iam-role-arn>* in the following COPY commands with your role ARN. Then run the commands in your SQL client tool.

copy users from 's3://awssampledbuswest2/tickit/allusers\_pipe.txt'

credentials 'aws\_iam\_role=<iam-role-arn>'

delimiter '|' region 'us-west-2';

copy venue from 's3://awssampledbuswest2/tickit/venue\_pipe.txt'

credentials 'aws\_iam\_role=<iam-role-arn>'

delimiter '|' region 'us-west-2';

copy category from 's3://awssampledbuswest2/tickit/category\_pipe.txt'

credentials 'aws\_iam\_role=<iam-role-arn>'

delimiter '|' region 'us-west-2';

copy date from 's3://awssampledbuswest2/tickit/date2008\_pipe.txt'

credentials 'aws\_iam\_role=<iam-role-arn>'

delimiter '|' region 'us-west-2';

copy event from 's3://awssampledbuswest2/tickit/allevents\_pipe.txt'

credentials 'aws\_iam\_role=<iam-role-arn>'

delimiter '|' timeformat 'YYYY-MM-DD HH:MI:SS' region 'us-west-2';

copy listing from 's3://awssampledbuswest2/tickit/listings\_pipe.txt'

credentials 'aws\_iam\_role=<iam-role-arn>'

delimiter '|' region 'us-west-2';

copy sales from 's3://awssampledbuswest2/tickit/sales\_tab.txt'

credentials 'aws\_iam\_role=<iam-role-arn>'

delimiter '\t' timeformat 'MM/DD/YYYY HH:MI:SS' region 'us-west-2';

* + Now try the example queries.

-- 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;

* + You can optionally go the Amazon Redshift console to review the queries you executed. The **Queries** tab shows a list of queries that you executed over a time period you specify. By default, the console displays queries that have executed in the last 24 hours, including currently executing queries.
    - Go to Redshift console.
    - In the cluster list in the right pane, choose examplecluster.
    - Choose the **Queries** tab.

The console displays list of queries you executed.

* + - To view more information about a query, choose the query ID link in the **Query** column or choose the magnifying glass icon.

#### Cleanup

#### 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 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**.