**To create an Amazon Redshift cluster based on a sample dataset**

1. Sign in to the AWS Management Console and open the Amazon Redshift console
2. To create a cluster, do one of the following:
   * On the Amazon Redshift service page, choose **Create cluster**. The Create cluster page appears.
3. In the **Cluster configuration** section, specify a **Cluster identifier**.
4. choose **Free trial** to create a configuration with the dc2.large node type.

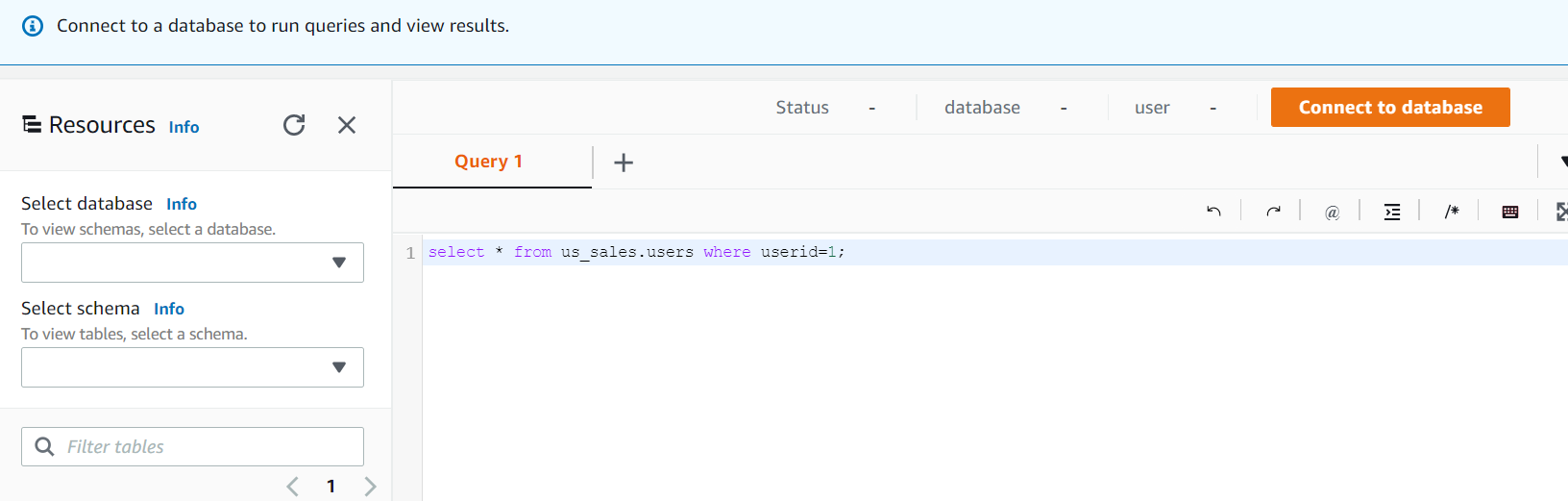

                        Console showing selection of a free trial
                     

1. After you choose your node type, do one of the following:
   * Amazon Redshift loads the sample dataset Tickit to the default dev database and public schema. You can start using the query editor to query data.
   * To bring your own data to your Amazon Redshift cluster, choose **Production**. Then in **Sample data**, choose **Load sample data**. For information about bringing your own data, see [Bringing your own data to Amazon Redshift](https://docs.aws.amazon.com/redshift/latest/gsg/bring-own-data.html).

Amazon Redshift automatically loads the sample dataset to your sample Amazon Redshift cluster.

1. In the **Database configuration** section, specify values for **Admin user name** and **Admin user password**.
   * **Admin user name**: Enter **awsuser**.
   * **Admin user password**: Enter a value for the password.
2. Choose **Create cluster**.
3. The Amazon Redshift query editor, which you can use to query data immediately as Amazon Redshift connects your new cluster to the query editor during cluster creation.
4. You can also choose other SQL client tools that support JDBC or ODBC drivers to work with data in your cluster
5. When Amazon Redshift is creating your Amazon Redshift cluster, it automatically uploads the sample dataset Tickit. Cluster creation might take a few minutes to complete. After creation completes, the cluster status becomes ACTIVE. You can view the sample Tickit tables in the query editor by choosing the dev database and public schema.
6. After the Amazon Redshift cluster is created, in **Connect to Amazon Redshift clusters**, choose **Query data**.
               Screenshot of the console: Query data under Connect to Amazon Redshift clusters
            

The Amazon Redshift query editor appears.



Click on Connect to Database

By default, Amazon Redshift shows the database created during cluster creation in **Select database** and the default schema named public. To view the individual data files of the sample dataset, go to the query editor and choose the dev database and public schema.

Try some example queries in the query editor, as shown following.   
  
-- 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;

**Connect to Redshift Cluster Using JDBC**

The Amazon Redshift JDBC driver uses TCP keepalives by default to prevent connections from timing out. You can specify when the driver should begin sending keepalive packets or simply disable the feature.

The following URL shows how to connect to a Redshift database named *company*using the standard login and password with TCP keep alive turned on and SSL encryption enabled:

**jdbc:redshift://redshift-test2.abcdefg.us-east-1.redshift.amazonaws.com:5439/company?ssl=true&tcpKeepAlive=true**

Notice that the values for the *ssl*and *tcpKeepAlive* parameters have been set to *true.*

**Bringing your own data to Amazon Redshift**

**#1. To create an IAM role for Amazon Redshift**

1. Sign in to the AWS Management Console and open the IAM console
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 that you just created.
11. Copy the **Role ARN** value 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.

**#2. Create a sample Amazon Redshift cluster**

**Follow the Instructions given in page1 to Craete Cluster**

1. Optional) In the **Cluster permissions** section, for **Available IAM roles** choose the IAM role that you previously created, **myRedshiftRole**. Then choose **Associate IAM role**.

If you created your cluster with the **Load sample data** option, associate an IAM role to the cluster.

1. (Optional) In the **Additional configurations** section, turn off **Use defaults** to modify **Network and security**, **Database configuration**, **Maintenance**, **Monitoring**, and **Backup** settings.

**#3: Configure inbound rules for SQL clients**

Later in this tutorial, you access your cluster from within a virtual private cloud (VPC) based on the Amazon VPC service. However, if you use an SQL client from outside your firewall to access the cluster, make sure that you grant inbound access.

**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 Elastic Compute Cloud (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. Find out the correct CIDR/IP for your environment.

**Step 4: Grant access to the query editor and run queries**

1. To use the Amazon Redshift query editor, you need permission. To set access, attach the AmazonRedshiftQueryEditor and AmazonRedshiftReadOnlyAccess IAM policies to the IAM user that you use to access your cluster.
2. On the navigation menu, choose EDITOR, then connect to a database in your cluster.
3. On the Connect to database page, there are two ways to authenticate, namely, Temporary credentials and AWS Secrets Manager. For this tutorial, choose Create a new connection and Temporary credentials, then enter the values that you used when you created the cluster, as follows:
4. Cluster: Choose examplecluster.
   * Database name: Enter dev.
   * Database user: Enter awsuser.
   * Then choose Connect.
5. For Schema, choose public to create a new table based on that schema.
   * Enter the following in the query editor window, and choose Run to create a new table.
   * create table shoes(shoetype varchar (10), color varchar(10));
   * 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');
   * Choose Clear.
7. 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.
     1. Shoe type Color
        1. sandals black
        2. loafers brown
   * Choose Execution to view the run details.
8. Choose Export to download the query results as a file. The supported file formats are CSV, TXT, and HTML.

**Step 5: Load sample data from Amazon S3**

1. 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.
2. To load sample data from Amazon S3
   * **Create tables.**
   * individually copy and run 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);

1. AWS recommend using the COPY command to load large datasets into Amazon Redshift from Amazon S3 or Amazon DynamoDB.
2. Use the myRedshiftRole IAM role that you created
3. 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 1: Create an IAM role.

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. If you are using the Amazon Redshift query editor, individually run the following commands.

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>';

**Step 6: Try example queries using the query editor**

-- 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;

**Create Schema**

create schema <schema-name> authorization <redshift-cluster-user>;

create schema us\_sales authorization awsuser;

**Step 7: Reset your environment**

1. When you have completed this Lab, you 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.
2. To delete a cluster
   * On the navigation menu, choose CLUSTERS to display your list of clusters.
   * Choose the cluster. For Actions, choose Delete. The Delete cluster page appears.
   * 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.

**Connect from External System**

* + Select Cluster 🡪 Actions 🡪 Modify Publicly Accessible Settings
    - Enable

[**ERRORS:**](https://docs.aws.amazon.com/redshift/latest/gsg/rs-gsg-clean-up-tasks.html)

If you get ERROR: The specified S3 prefix 'tickit/allusers\_pipe.txt' does not exist Detail:

Copy S3 URL from S3 and paste