# Table of Contents

Getting started ................................................................................................................................. 1
   Step 1: Set up prerequisites ........................................................................................................ 1
      Sign up for AWS ...................................................................................................................... 2
      Determine firewall rules ......................................................................................................... 2
   Step 2: Create an IAM role ......................................................................................................... 2
      To create an IAM role for Amazon Redshift ........................................................................... 2
   Step 3: Create a cluster .............................................................................................................. 3
   Step 4: Authorize cluster access ............................................................................................... 7
   Step 5: Connect to the cluster .................................................................................................... 9
      Querying a database using the query editor .......................................................................... 9
      Querying a database using a SQL client .................................................................................. 12
   Step 6: Load sample data ......................................................................................................... 16
   Step 7: Find additional resources and reset your environment .................................................. 20
      Where do I go from here? ....................................................................................................... 20
Document history ............................................................................................................................. 23
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.

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:

- Step 1: Set up prerequisites (p. 1)
- Step 2: Create an IAM role (p. 2)
- Step 3: Create a sample Amazon Redshift cluster (p. 3)
- Step 4: Authorize access to the cluster (p. 7)
- Step 5: Connect to the sample cluster and run queries (p. 9)
- Step 6: Load sample data from Amazon S3 (p. 16)
- Step 7: Find additional resources and reset your environment (p. 20)

After you complete this tutorial, you can find more information about Amazon Redshift and next steps in Where do I go from here? (p. 20)

**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. 20) 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 (p. 2)
• Determine firewall rules (p. 2)

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.

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. For more information about credentials and access permissions, see Credentials and access 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. 16).

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. In the **Cluster configuration** section, specify values for **Cluster identifier**, **Node type**, and **Nodes**.

   - **Cluster identifier**: Enter `examplecluster` for this tutorial. This identifier must be unique. The identifier must be from 1–63 characters using valid characters as a - z (lowercase only) and - (hyphen).
   - Choose one of the following methods to size your cluster:

   **Note**

   The following step describes an Amazon Redshift console that is running in an AWS Region that supports RA3 node types. For a list of AWS Regions that support RA3 node
types, see Overview of RA3 node types in the Amazon Redshift Cluster Management Guide.

- If your AWS Region supports RA3 node types, choose either Production or Free trial to answer the question What are you planning to use this cluster for?

  If your organization is eligible, you might be able to create a cluster under the Amazon Redshift free trial program. To do this, choose Free trial to create a configuration with the dc2.large node type. For more information about choosing a free trial, see Amazon Redshift free trial.

- If you don’t know how large to size your cluster, choose Help me choose. Doing this starts a sizing calculator that asks you questions about the size and query characteristics of the data that you plan to store in your data warehouse.

  If you know the required size of your cluster (that is, the node type and number of nodes), choose I’ll choose. Then choose the Node type and number of Nodes to size your cluster for the proof of concept.

  Choose Node type: dc2.large with Nodes: 2 for this tutorial.

5. In the Database configurations section, specify values for Database name (optional), Database port (optional), Master user name, and Master user password. This tutorial uses these values:

- Database name (optional): Enter dev.
- Database port (optional): Enter 5439.
- Master user name: Enter awsuser.
- Master user password: Enter a value for the password.

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

7. Optionally, in the Additional configurations section, turn off Use defaults to modify Network and security, Database configurations, Maintenance, Monitoring, and Backup settings.

8. 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, see 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**.
Step 3: Create a 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.

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.
Step 4: Authorize cluster access

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.

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 (p. 9)
- Querying a database using a SQL client (p. 12)

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
- RA3.4xlarge
- RA3.16xlarge

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 Europe (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 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.

On the Connect to database window, enter the values you used when you created the cluster as follows:

- **Cluster**: Choose examplecluster
- **Database name**: Enter dev
- **Database user**: Enter awsuser
- **Database password**: Enter password that you specified when you created the cluster

Then choose Connect to database.

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 to create a new table.

```sql
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.

<table>
<thead>
<tr>
<th>Shoetype</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>sandals</td>
<td>black</td>
</tr>
<tr>
<td>loafers</td>
<td>brown</td>
</tr>
</tbody>
</table>

9. Choose **Execution** to view the run details.
10. Choose **Export** to download the query results as a file. The supported file formats are CSV, TXT, and HTML.

**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.

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


7. Enter the following command in the Query editor window and choose Run query to add rows to the table.

```sql
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.

```sql
select * from shoes;
```

You should see the following results.

```
showshoes

1 select * from shoes

Use Ctrl + Enter to run query, Ctrl + Space to autocomplete

Run query  Save as  Save  Clear

Query results: Query completed in 0.642 seconds

Showing row(s) 1 - 2

<table>
<thead>
<tr>
<th>shoetype</th>
<th>color</th>
</tr>
</thead>
<tbody>
<tr>
<td>sandals</td>
<td>black</td>
</tr>
<tr>
<td>loafers</td>
<td>brown</td>
</tr>
</tbody>
</table>
```

**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:

- Install SQL client drivers and tools (p. 13)
- To get your connection string (p. 13)
To connect from SQL Workbench/J to your cluster (p. 14)

After you complete this step, you can determine whether you want to load sample data from Amazon S3 in Step 6: Load sample data from Amazon S3 (p. 16) or find more information about Amazon Redshift and reset your environment at Where do I go from here? (p. 20).

**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.

![JDBC URL](image)

**To connect from SQL Workbench/J to your cluster**

This step assumes that 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.

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

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

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

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

```sql
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. 20).
To load sample data

1. Create tables.

If you are using the Amazon Redshift query editor, 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.

```sql
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 sortkey,
    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,
    qtsold 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.
   For information about IAM permissions for the COPY command, see COPY in the Amazon
   Redshift Database Developer Guide.

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. 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 '	' 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  
```
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. 21).

- 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. 22).

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.

![VPC security group image]

4. With the default security group selected, choose the Inbound tab and then choose Edit.

![Edit inbound security group image]

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**.

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

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